

DRAFT MANUAL

TABULAR DESCRIPTIVE LANGUAGE

January 15, 1962 Technical Report No. 2A2 Thomas B. Glans Burton Grad Felo

INTERNATIONAL BUSINESS MACHINES CORPORATION

White Plains, New York

© 1962 by International Business Machines Corporation

IBM Data Processing Division 200 Mamaroneck Avenue White Plains, New York

January 17, 1962

To: Recipients of the Tabular Descriptive Language Manual

Subject: Corrections and Additions

The manual that you received is incomplete in a number of ways. Included within this package are a cover page, preface, table of contents and an errata sheet containing a list of the errors we have found to date. The corrections are in the same order as the manual itself and therefore can be made easily.

Three sections of the manual have not yet been completed: Chapter 3 which includes two sample problems and their solutions, using the Tabular Descriptive Language; Chapter 2, Example 8, which defines the use of the input-output operations (input-output is described in Chapter 6); and finally, Chapter 7, which defines the way in which the various forms used with TDL are written.

A complete revision of the manual is currently under way. It is anticipated that this will be printed by the 26th of January. These revised manuals will include those portions not currently in your hands and will be sent to you as soon as available.

Your comments, suggestions and recommendations are most welcome and should be submitted through Mr. D. Hradecky, Education Planning, Endicott. Any forms that you require can also be obtained from Education Planning. I hope that you will find the Tabular Descriptive Language a useful tool for system development and documentation.

Yhomas B. Mana

Thomas B. Glans Systems Engineering Services

TBG:mam Enclosures

PREFACE

There is increasing evidence to indicate that decision tables are a useful means of analyzing and portraying systems logic. With this in mind the Tabular Descriptive Language was developed to encourage experiments, using decision tables for a wide variety of business problems. The feedback from the users of the Tabular Descriptive Language will provide a reliable evaluation of the usefulness of decision tables in problem analysis and documentation.

The Tabular Descriptive Language is designed to permit systems engineers to clearly and effectively think through the systems design logic, record this logic, and then communicate it to other systems engineers or programmers. The primary purpose is man-to-man communication, as opposed to man-to-machine communication, for which programming languages are designed. Various levels of system description can be made using the Tabular Descriptive Language because of the hierarchical nature of the language, that is, operands need be specified only to the level of detail necessary for clear communication, and certain procedures need merely be referenced in the decision tables without portraying the explicit logic.

This initial version of the Tabular Descriptive Language will provide a foundation upon which future developments can be built as the needs for expansion are identified. Therefore, the evaluation, comments and recommendations of users will serve to guide future development, so that the full potential of an advanced systems language can be realized within the near future.

i

A word about this manual. It is directed primarily at people who will be experimenting with the language. Chapter 1 presents a quick sketch of the basic ideas of data processing and decision tables, along with the environment in which the Tabular Descriptive Language is to be used. Chapter 2 covers the essentials of the Language, using a set of graded examples to introduce a few new ideas at a time. There are also review questions and answers after each example. Chapter 3 contains two problems worked through in the Tabular Descriptive Language. Chapters 4, 5 and 6 are the reference portion. Chapter 4 defines the structure of the language, Chapter 5 the operands and Chapter 6 the decision table operations. Chapter 7, the Appendix, indicates the mechanics of filling out the various forms needed to describe a system.

We wish to thank Messrs. G. H. Bang, R. N. Barnes, W. L. Kelly, A. A. Lea, and W. M. Selden for their contributions and suggestions in defining the Tabular Descriptive Language, Mr. J. L. Mannasmith for developing the problem solutions of Chapter 3, Mr. L. Weinstein for his development of sample problems and overall review of the language, Miss S. Troy for developing sample problems in Chapter 2 and preparing teaching material and Mr. E. L. Blossom for his assistance in rewriting the material in the reference portion of the manual.

It is expected that improvements will be made to the language as suggestions are received from users. Since the Tabular Descriptive Language is for man-to-man communication there is no implication that

ii

IBM will produce a processor to translate this language to any IBM data processing machine.

Thomas B. Glans

Burton Grad

January 17, 1962

TABLE OF CONTENTS

CHAPTER 1 - A Tabular-form Descriptive Language

A Data Processing System	1.1
Decision Tables	1.3
The Language Concept	1.6
Purpose of the Language	1.7
Structure and Content of the Language	1.7
Operand Descriptions	1.7
Decision Table Descriptions	1.8
Use of TDL	1.13
Solving a Systems Problem	1.14

CHAPTER 2 - Using the Tabular Descriptive Language

Introduction	n.	•	•			•						•	•	2.1
Example 1														2.2
Example 2														2.7
Example 3														2.12
Example 4														2.20
Example 5														2.23
Example 6														2.28
Example 7														2.32
Example 8	(No	ot	p	re	es	er	nt)					

CHAPTER 3 - None

CHAPTER 4 - The Structure of the Language

Nature of Language	•	•	•	•									4.1
Vocabulary and Grammar							•						4.1
Formation of Operands .													4.2
Character Set													4.2
Names													4.2
Special Names													4.3
Literals													4.4
Connotative Term	S												4.4
State Terms													4.4
Descriptive Term	S												4.5
Operator Names													4.5
Formation of Statements													4.5
Condition Statements													4.6
Action Statements .													4.6
Formation of Decision Tab	le	S											4.6
Table Header .													4.10
Condition Stub .													4.10
Condition Entry													4.10
Action Stub													4.11
Action Entry													4.11
Rule Header											0	÷	4.11
Remarks											÷		4.11
Formation of System Descu	rin	oti	on	Ĉ.					2	Č.			4 12
Special Decision Table	2			2									4 12
START				2				<u> </u>	2	Ċ		÷.	4 12
STOP													4 13
Limited and Extended	Er	ntr	v	T	able	S			1				4 13
Open and Closed Table	g	a tub	3	Τc	LOIC	5							4 14
ohorr arra oropour Tapic	5						•		•		•		T. TI

CHAPTER 5 -

Operands

Types of Operands														5.1
Operand Names .														5.3
Information Operar	nds.													5.3
Files														5.3
Records														5.5
Fields														5.6
Association Gro	ups													5.7
Sub-Fields .														5.8
Reference Operand	s.													5.10
Reference List	Fun	ctic	ons											5.10
Value Lists .			La C	-										5.11
Arithmetic Expr	ess	sion	s.											5.12
Logical Express	sion	s.												5.14
Symbolic Expres	ssid	ons												5.16
Constants .			2	2								÷.		5.16
Work Operands .	2.2							÷	÷	÷		2		5.16
Work Records											2	2		5.17
Work Fields	2.2		2	Č.,				÷	÷	÷	÷.			5.18
System Segmenting	Op	era	nds	5.										5.18
Procedure Name	es .	120 AN 22	arease a							÷	2	2		5.18
Decision Table	Nan	nes												5.19
Self-describing Op	era	nds		0									0	5.19
Direct Valued L	iter	als	Ċ.											5.20
Numeric Lite	eral	S.						÷						5.20
Alphameric I	ite	rals	5						Ċ					5.21
Connotative Ter	ms		•••	•	·	•	•	•	•					5.21
State Terms	1110	•	•	•	•	•	•	•	•	•		•	•	5. 22
Descriptive Ter	ms	•	•	•	•	•	•	•	•	·	÷	÷	•	5.23
Value List	1110	•	•	•	•	•	•	•	•	•	•		•	5.23
Rance .	• •	·	•	•	•	•	•	•	•	•		•	•	5 24
roange	• •	•	•	•	٠	•	•	•	•	•	•	•	•	0.01

CHAPTER 6 - Decision Table Operations

Conditions														6.1
Value Relational C	Con	dit	ion	IS		•								6.2
Operations .		•												6.2
Operands														6.3
Comparison Ru	les	5.	•	•	•						•			6.3
Usage					•					•		•		6.4
Field Name														6.5
Value List N	Jan	ne												6.6
Arithmetic 1	Exp	pre	ssi	ion	Na	am	е						•	6.6
Symbolic Ex	pro	ess	sion	n N	Ian	ne								6.7
Constant Na	me													6.8
Work Field	Na	me											•	6.8
Direct Value	ed	Lit	era	al										6.8
Connotative	Te	rm	1.											6.9
Descriptive	Te	rm	1.											6.9
Value List a	nd	Ra	ing	е										6.10
Mechanics														6.10
State Relational C	ond	liti	ons	3										6.12
Operators														6.12
Operands														6.12
Comparison Ru	les	5.												6.13
Usage														6.13
POSITIVE.														6.13
NEGATIVE														6.14
NUMERIC.				2			2							6.15
ALPHABET	IC		0		÷.		<u>.</u>			2			2	6.15
STATUS										2		2	÷	6.15
PRESENT		•		÷.							1		2	6.16
END OF FIL	.F.	÷.	•										÷	6.16
Mechanics						÷.		÷.		Ċ.,	2		1	6.17
Actions														6.18
Assignment Action	ns			•						0			÷	6.18
Operators		•	•	•	•	•	•							6.18
Operands	·			•	•	•	•					•		6.19
Special Bules			•	•	•	•			•	•	•			6.20
Usage	•	•	•		·	•	•	•	•	•	•	•		6 22
Individual V			Acc	im	· nm	on	÷.	•	•	•	•	•	•	6 22
Correspond	ina	T	100	a A	dee	im	nm.	- ont	•	•	•		•	6 23
Two_opener	ing /	Ini	thn	not	ic	AC	sic	rnn	107	t	•	•	•	6 23
Multiple For	cto	21 1	Init	thn	net	ic	Ac	sic	mr	ner	t	•		6 24
Mochanica	CLU	I I	77.7		uet	10	no	DIC	J111.	191	10	•	•	6 26
mechanics								•	•	•				0. 40

Sequence Control	Ac	cti	on	S		•									6.28
Operators															6.28
Operands													•		6.28
Special Rules								٠			•		•		6.28
Usage	•														6.29
GOTO				•				•							6.29
DO															6.30
Mechanics	•														6.32
Input/Output Actions															6.40
Serial-type Files				÷											6.41
Operators		•									•		•		6.41
Operands															6.41
Control Rules	•							•		•					6.42
Usage											•				6.43
READ				٠		•				•					6.43
WRITE	•							•					•		6.44
BEGIN	•														6.44
SORT															6.45
Mechanics															6.46
Random-type Files	5														6.47
Operators	•				•	•		•	•	•	•	•	•		6.48
Operands				•		•		•	•	•					6.48
Control Rules		•	•										•		6.48
Usage		•	•	•		•	•	•		•	•				6.49
READ								•		•					6.49
WRITE				•						•				•	6.50
CLEAR		•			•			•			•				6.50
Mechanics															6.52

TABULAR DESCRIPTIVE LANGUAGE (Technical Report No. 2A2)

ERRATA #1

January 17, 1962

Page	Paragraph	Line	
1.7	1	12	names of files , records
1.8		5	Commonly <u>understood names</u> can be
2.4		9	operand 1, operator, operand 2
2.8	3	1	Using this form the complete
2.10		12	a quantity <u>in</u> the first
2.11		Q 2	What action_is
		Q 3	EQ operat <u>o</u> r
2.17		3	$\underline{B + A} \rightarrow B$
		4	<u>B - A</u> → B
		5	$\underline{B * A} \rightarrow B$
		6	$\underline{B/A} \rightarrow B$
2.18 2.19 2.20 2.21 2.22	2 Ans.1 3	1 9 6 11	Action row 8, in go to table <u>GETX</u> action in row <u>5</u> introduces range 22-24, the
2 23	2	5	be preceded by
2.24	Left	Ser. 4	QUAN SOLD: CT * CUST DISC:WORK A
2.24	1	4	To refer to these fields the user must qualify <u>the field name</u> with the record name WORK A, e.g., QTY ON HAND: WORK A

Errata #1 - 2 -

Page	Paragraph	Line	
2.24 2.25	3 3	2 4	field, AMT EXT_, from For example, serial 6 of
2.25	Fig. 5c	Ser. 4	QUAN SOLD $N + 4$
2.25	Last	4	\ldots serial <u>5</u> of figure 5b, \ldots
2.27		Q 4	delete question 4 and answer 4
2.28	4	4	and GO TO table 016.
2.28	4	7	no identifying marks
2.29		2	LOC NO, DEPT, EMP NO,
2.31	Ans.	3	of numeric <u>valued</u> operands with
	Ans.	5	the bond purchase <u>price</u> is placed.
	Ans.	5	and the table is repeated.
2.32	2	10	DEPT EXP REPT.
		12	is preceded by
2.33	1	4	is preceded by
2.35	2	13	DEPT NO. From
		16	field (the right-most) is in
	3	4	\ldots is in printwheel _ position 64.
5 <mark>.</mark> 6	6	2	as alphameric, <u>the entire field</u> is alphameric.
5.12	1	5	an alphameric <u>literal</u> is enclosed
	3	2	The value of an expression is always
5.21	2	8	All examples in single quotes (').
5.22		1	ZERO_S
		6	zero_s,

Errata #1 - 3 -

.Page	Paragraph	Line	
6.1		2	to be des <u>cr</u> ibed by the
6.3	4	2	(2) High-order zeros are added
6.4	2	3	is performed position <u>by</u> position from
6.6		7	and these operands:
6.8		22	Value_ List Name
		23	Value_ List
6.12	4	5	Delete <u>Record Name</u>
6.13		15	operands <u>are</u> discussed
6.15		13	be used with <u>numeric or</u> alphameric_operands.
6.27		1	The first <u>two</u> forms shown <u>are</u> limited entry, <u>the last four</u> are
	Last		the other <u>six</u> are
6.32	Last		PER ASSOC letter <u>FOR</u> ASSOC letter PER ASSOC letter <u>FOR</u> ASSOC letter
6.46		4	\underline{X} , ., or blank
		5	\underline{X} , ., or blank
		7	\underline{X} , ., or blank
		8	\underline{X} , ., or blank
6.47		2	\underline{X} , ., or blank
6.52		3	\underline{X} , ., or blank
		16	\underline{X} , ., or blank



PRELIMINARY MANUAL

TABULAR DESCRIPTIVE LANGUAGE

January 25, 1962 Technical Report No. 2A5 Thomas B. Glans Burton Grad

INTERNATIONAL BUSINESS MACHINES CORPORATION White Plains, New York

PREFACE

There is increasing evidence to indicate that decision tables are a useful means of analyzing and portraying systems logic. With this in mind the Tabular Descriptive Language was developed to encourage experiments using decision tables for a wide variety of business problems. The feedback from the users of the Tabular Descriptive Language will provide a reliable evaluation of the usefulness of decision tables in problem analysis and documentation.

The Tabular Descriptive Language is designed to permit systems engineers to clearly and effectively think through the systems design logic, record this logic, and then communicate it to other systems engineers or programmers. The primary purpose is man-to-man communication, as opposed to man-to-machine communication, for which programming languages are designed. Various levels of system description can be made using the Tabular Descriptive Language because of the hierarchical nature of the language, that is, operands need be specified only to the level of detail necessary for clear communication, and certain procedures need merely be referenced in the decision tables without portraying the explicit logic.

This initial version of the Tabular Descriptive Language will provide a foundation upon which future developments can be built as the needs for expansion are identified. Therefore, the evaluation, comments and recommendations of users will serve to guide future development, so that the full potential of an advanced systems language can be realized within the near future.

i

A word about this manual. It is directed primarily at people who will be experimenting with the language. Chapter 1 presents a quick sketch of the basic ideas of data processing and decision tables, along with the environment in which the Tabular Descriptive Language is to be used. Chapter 2 covers the essentials of the Language, using a set of graded examples to introduce a few new ideas at a time. There are also review questions and answers after each example. Chapter 3 contains two problems worked through in the Tabular Descriptive Language. Chapters 4, 5 and 6 are the reference portion. Chapter 4 defines the structure of the language, Chapter 5 the operands and Chapter 6 the decision table operations. Chapter 7, the Appendix, indicates the mechanics of filling out the various forms needed to describe a system.

We wish to thank Messrs. G. H. Bang, R. N. Barnes, W. L. Kelly, A. A. Lea, and W. M. Selden for their contributions and suggestions in defining the Tabular Descriptive Language, Mr. J. L. Mannasmith for developing the problem solutions of Chapter 3, Mr. L. Weinstein for his development of sample problems and overall review of the language, Miss S. Troy for developing sample problems in Chapter 2 and preparing teaching material, and Mr. E. L. Blossom for his assistance in rewriting the material in the reference portion of the manual.

It is expected that improvements will be made to the language as suggestions are received from users. Since the Tabular Descriptive Language is for man-to-man communication there is no implication that

ii

IBM will produce a processor to translate this language for any IBM data processing machine.

Since this is a first attempt to formalize the language used for man-to-man communication there have been many technical difficulties. As a result, there may need to be further changes and improvements as feedback is received from users. Please note, therefore, that this is a preliminary manual which will need to be technically modified and rewritten after there has been adequate experience.

> Thomas B. Glans Burton Grad

January 17, 1962

TABLE OF CONTENTS

CHAPTER 1 - A Tabular-form Descriptive Language

A Data Processing System	1.1
Decision Tables	1.3
The Language Concept	1.6
Purpose of the Language	1.7
Structure and Content of the Language	1.7
Operand Descriptions	1.7
Decision Table Descriptions	1.8
Jse of TDL	1.13
Solving a Systems Problem	1.14

CHAPTER 2 - Using the Tabular Descriptive Language

Introduct	ior	1								٠					2.1
Example	1														2.2
Example	2	•	•			•				•					2.8
Example	3		•											•	2.14
Example	4														2.24
Example	5	•	•									•		•	2.29
Example	6							•	•				•	•	2.35
Example	7		•											\sim	2.41
Example	8			•		,	×				•	٠	•	٠	2.47

CHAPTER 3 - Sample Systems Descriptions

Example	1											٠	٠	•	•	3.1
Example	2	•	•	•	٠	•	٠	•	·	٠	•		•	•		3.19

CHAPTER 4 - The Structure of the Language

Vocabulary and Grammer4.1Formation of Operands4.2Character Set4.2Names4.2
Formation of Operands
Character Set
Names 4.2
Special Names 4.3
Literals \ldots 4.4
Connotative Terms $\ldots \ldots \ldots \ldots \ldots 4.4$
State Terms 4.4
Descriptive Terms 4.5
Operator Names 4.5
Formation of Statements 4.5
Condition Statements 4.6
Action Statements 4.6
Formation of Decision Tables 4.6
Table Header 4.9
Condition Stub
Condition Entry 4.9
Action Stub
Action Entry
Rule Header 4.10
Remarks
Formation of System Description
Special Decision Tables
START 4 11
$\frac{111}{5}$
Limited and Extended Entry Tables
Open and Closed Tables 4 13

CHAPTER 5 - Operands

Operand Description Sheet5.2Operand Names5.3Information Operands5.4Files5.4Records5.5Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Self-Describing Operands5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22Descriptive Terms5.22Value List5.21Connotative Terms5.22Value List5.24Descriptive Terms5.22Value List5.24Descriptive Terms5.24	Types of Operands				•				•		•					5.1
Operand Names5.3Information Operands5.4Files5.4Records5.5Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.17Work Operands5.17Work Operands5.17Work Records5.18System Segmenting Operands5.19Self-Describing Operands5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.22Value List5.23Value List5.23Value List5.24Descriptive Terms5.24Descriptive Terms5.24	Operand Description	Sh	lee	t									•	2	•	5.2
Information Operands5.4Files5.4Records5.5Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Self-Describing Operands5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.22Value List5.23Value List5.23Value List5.24Descriptive Terms5.24Descriptive Terms5.24	Operand Names	•			•											5.3
Files5.4Records5.5Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.18Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.22Value List5.24Descriptive Terms5.24	Information Operands	5						÷								5.4
Records5.5Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.18Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.22Value List5.23Value List5.24Descriptive Terms5.24Descriptive Terms5.24	Files															5.4
Fields5.6Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22Descriptive Terms5.23Value List5.24Descriptive Terms5.24	Records															5.5
Association Groups5.7Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Fields															5.6
Sub-Fields5.9Reference Operands5.10Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Deces5.24Descriptive Terms5.24	Association Grou	ps	;		•											5.7
Reference Operands 5.10 Reference List Functions 5.10 Value Lists 5.11 Arithmetic Expressions 5.12 Logical Expressions 5.14 Symbolic Expressions 5.16 Constants 5.17 Work Operands 5.17 Work Records 5.18 System Segmenting Operands 5.19 Decision Table Names 5.19 Self-Describing Operands 5.20 Numeric Literals 5.20 Alphameric Literals 5.21 Connotative Terms 5.22 State Terms 5.23 Value List 5.24	Sub-Fields															5.9
Reference List Functions5.10Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Reference Operands															5.10
Value Lists5.11Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Reference List H	Fur	nct	ion	ns											5.10
Arithmetic Expressions5.12Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.20Literals5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Value Lists															5.11
Logical Expressions5.14Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Numeric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Arithmetic Expr	es	sic	ons	5											5.12
Symbolic Expressions5.16Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Logical Express	ior	ıs													5.14
Constants5.17Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22Descriptive Terms5.23Value List5.24Descriptive Terms5.24	Symbolic Expres	si	ons	5												5.16
Work Operands5.17Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Constants															5.17
Work Records5.18Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.23Value List5.24Descriptive Terms5.24	Work Operands	2														5.17
Work Fields5.18System Segmenting Operands5.18Procedure Names5.19Decision Table Names5.19Self-Describing Operands5.20Literals5.20Numeric Literals5.20Alphameric Literals5.21Connotative Terms5.22State Terms5.22Descriptive Terms5.23Value List5.24Descriptive Terms5.24	Work Records	2														5.18
System Segmenting Operands 5.18 Procedure Names 5.19 Decision Table Names 5.19 Self-Describing Operands 5.20 Literals 5.20 Numeric Literals 5.20 Alphameric Literals 5.21 Connotative Terms 5.22 State Terms 5.23 Value List 5.24	Work Fields .															5.18
Procedure Names 5.19 Decision Table Names 5.19 Self-Describing Operands 5.20 Literals 5.20 Numeric Literals 5.20 Alphameric Literals 5.21 Connotative Terms 5.22 State Terms 5.22 Descriptive Terms 5.23 Value List 5.24	System Segmenting C	pe	ra	nd	S	2									÷.	5.18
Decision Table Names	Procedure Name	S														5.19
Self-Describing Operands 5.20 Literals 5.20 Numeric Literals 5.20 Alphameric Literals 5.21 Connotative Terms 5.22 State Terms 5.22 Descriptive Terms 5.23 Value List 5.24	Decision Table N	Jan	me	s							÷.					5.19
Literals	Self-Describing Oper	ar	ids			ŝ.	÷									5.20
Numeric Literals 5.20 Alphameric Literals 5.21 Connotative Terms 5.22 State Terms 5.22 Descriptive Terms 5.23 Value List 5.24	Literals					2										5.20
Alphameric Literals5.21Connotative Terms5.22State Terms5.22Descriptive Terms5.23Value List5.24Descriptive5.24	Numeric Li	ter	al	s		2	÷			1						5.20
Connotative Terms 5.22 State Terms 5.22 Descriptive Terms 5.23 Value List 5.24	Alphameric	T.i	ite	ra	Is	÷	1		0							5.21
State Terms 5.22 Descriptive Terms 5.23 Value List 5.24	Connotative Terr	ms	:		.~			2		Ĵ.	÷				÷	5.22
Descriptive Terms. 5.23 Value List 5.24	State Terms		·				Ċ									5.22
Value List	Descriptive Ter	ms			Ċ	<u>.</u>		•								5.23
Danga 5.94	Value List										÷	Ĵ				5.24
Ranne	Rance				÷.		Ċ				÷	1				5.24

CHAPTER 6 - Decision Table Operations

Conditions			6.1
Value Relational Conditions			6.2
Operators			6.3
Operands			6.3
Comparison Rules			6.4
Usage			6.5
Field Name			6.5
Reference List Function Name			6.5
Value List Name			6.6
Arithmetic Expression Name .			6.7
Symbolic Expression Name.	2		6.7
Constant Name	2		6.8
Work Field Name	÷		6.8
Literal	÷		6.9
Connotative Term			6.9
Descriptive Term	÷		6.10
Value List and Range	<u>`</u>	• •	6.10
Mechanics			6,11
State Relational Conditions	÷.		6.13
Operators	·		6.13
Operands		• •	6.13
Comparison Rules			6.14
Usare		•••	6.14
POSITIVE	•	• •	6 15
NEGATIVE		•••	6 15
NUMERIC	•	• •	6 16
	•	• •	6 16
	·	• •	6 16
DDEGENT	·	• •	6 17
		• •	6 17
	•	•••	6 18
	•	• •	0.10

(2)					
Actions				6 19	
Assignment Actio	ns		•	6 19	
Operators			8 (* . * 27 (20) (*	6.20	
Operands.				6.21	
Special Rules				6.21	
Usage			N 101 N	6.23	
Individua	ul Value Assi	anment .		6.24	
Correspo	onding Value	Assignme	nt.	6.24	
Two-ope	rand Arithm	etic Assign	iment	6.25	
Multiple	Factor Arith	metic Assi	mment	6.25	
Mechanics			9	6.27	
Sequence Control	Actions.		5 S S	6.29	
Operators .				6.29	
Operands.				6.29	
Special Rules	5			6.30	
Usage · · ·				6.30	
GO TO .				6.30	
DO				6.31	
Mechanics .				6.33	
Input/Output Actions .	300 A 500 A 1			6.35	
Types of Files .				6.36	
Serial Files			• • •	6.36	
Operands .				6.38	
Operators .			• • •	6.38	
READ .				6.39	
WRITE .				6.39	
BEGIN .			• • •	6.41	
SORT .			• • •	6.42	
Random Files .			• • •	6.43	
Operands .				6.43	
Operators .				6.44	
READ .			• • •	6.44	
WRITE			• • •	6.46	
CLEAR				6.48	

CHAPTER 7 - Appendix

File Description Sheet	7.1
Sample Form	7.2
General Headings	7.3
File Description Headings	7.3
Field Description	7.5
Reference Operand Description Sheet	7.11
General Headings	7.11
Sample Form	7.12
Operand Description	7.13
Constants	7.13
Expression	7.13
Value List	7.16
Reference List	7.16
Decision Table Forms	7.20
Sample Form	7.21
Continuation Form	7. 22
Decision Table Header	7.23
Decision Rule Header	7. 25
Table Body	7.27
Report Specification	7.30
Sample Printer Lavout Sheet	7.31
Report Lavout Sheet	7.32
Record Heading	7.33
Record Description	7.35
Edited Fields	7.36
Character Representation	7.37
Insertion Characters	7.37
Replacement	7.42
Field Adjustment	7.43
Operand Summary	7 45
Character Set in Collating Sequence	7 46
Functions	7 47
Condition Summary	7 50
Oreward Combinations (Value Relationals)	7 51
Operand Combinations (Value Relationals)	7 53
File Conditions	7 55
Action Summary	7 56
MOVE and SET Operands	7 57
Input Output Operators	7 59

CHAPTER I

A Tabular-Form Descriptive Language

Suppose you have been assigned to a team that is to set up a data processing system for some application in payroll, or inventory control, or utility billing, or insurance, or even in an area of science or engineering. What do you need to know about data processing in order to use the Tabular-form Descriptive Language (TDL) on such a job?

To begin with, we can mention a few areas of knowledge that are not needed. You need no knowledge of electronics. You need no knowledge of mathematics beyond high school algebra (unless, of course, the problem itself is mathematical). With the Tabular-form Descriptive Language you do not even need a detailed knowledge of how any particular computer system works. However, you do need to know certain facts about data processing, and eventually, if you work with the subject for a while, you will pick up certain detailed facts about particular computers - but you do not need these now. For now, the general ideas which you should have are discussed below.

A data processing system is composed functionally of five parts, as shown in Figure 1. The input section accepts information "from the outside," and converts it into the electronic form in which it is manipulated and stored internally. Externally, information is typically recorded on punched cards, punched paper tape, or magnetic tape. In some applications, printed characters can be read directly. Presently-used business machines cannot recognize handwriting or speech. The output section of a computer has the obvious function of converting from the internal representation to some convenient external form, such as printing, punched cards, magnetic tape, punched paper tape, or a variety of specialized media. Though the speeds of all

This introductory information is reprinted from the IBM Commercial Translator Manual; it also seems appropriate for this teaching text. "Tabular-form Descriptive Language" has been substituted for "Commercial Translator" throughout. these devices are much greater than those of manual devices, they are still generally quite slow compared to speeds of internal electronic manipulation. The kind and number of input and output devices naturally depends on the particular machine and its application.

The storage section of a computer serves two important purposes. The obvious function is to hold the data on which we wish to operate. A function less obvious to the newcomer is to hold coded instructions which we place there to specify the procedure we wish to follow. A collection of such instructions is called a program. There are usually two types of storage. One type, though very fast, is of limited capacity and quite expensive; it is called main storage. What is frequently termed auxiliary storage can hold much more information, but is substantially slower.



Figure 1. Schematic representation of the functional parts of a computer.

The last two sections of a computer are called the logicalarithmetic section and the control section. The actual data processing is done in the logical-arithmetic section, and the control section is needed to decode and interpret the instructions in storage.

A most important feature of modern data processing machines is the way instructions are held in main storage right along with the data. For this reason we speak of a stored-program machine. The machine is therefore able to bring in new instructions when needed so that extremely large and complex problems can be solved.

The instructions which a data processing machine can execute naturally vary from one machine to another, but they can still be grouped into general categories. One group is used for arithmetic operations, another for making the elementary "decisions" of which a data processing machine is capable. Still another group covers input/output operations and a fourth group carries out miscellaneous control functions which are required because of the way the machine operates. Most individual operations are quite elementary, requiring a large number of them be combined properly in order to carry out a meaningful data processing task. This work, which follows the complete definition of the processing task, is called programming.

Data processing requires an extremely precise statement of the problem. We must not say "less than 30" if we mean "less than or equal to 30." There is no way we can say, "make sure the data looks reasonable;" if we want to check the validity of data, we must specify exactly what tests are to be made on it.

With electronic data processing, we are required to detail our procedures in advance to a degree not found in other methods. If we were asking a clerk to do a job, we might end by saying, "and if you run into anything you don't know how to handle, call me and we'll figure out what to do." In order to do a similar thing with a data processing machine, it is necessary first to define precisely what constitutes an exception, and then to write a procedure to handle it.

Decision Tables

Of the various activities that go into setting up a data processing procedure for a computer, the hardest is obtaining this precise definition of exactly what is to be done under all combinations of circumstances. Every step must be specified. The conditions under which each action is to be performed must be enumerated. The exceptions to normal processing must be identified. Necessary sequences of operations must be precisely indicated.

Determining what is required of the computer system is called <u>analysis</u>; deciding just how to go about meeting these requirements is the area of system <u>design</u>; communicating the chosen procedures to the computer is called <u>programming</u>. In each of these areas a <u>language</u> is needed for defining the data processing procedures. Ideally, a language form or structure should be suitable both for man-to-man documentation, and for man-to-machine communication.

Many languages are used for these purposes. Procedures are often communicated to the machine in a form closely resembling the language of its own instructions, but this imposes a heavy and unnecessary burden on the person writing the procedures, since human language and machine language are quite different. Flow charts are widely used for man-to-man communication, but they can become confusing in complex situations, and the flow chart language is not suitable for communication with the machine. Logical equations are sometimes used, but they do not display relationships in as graphical a form as we might wish, and furthermore most system designers do not find them to be a comfortable form of expression.

<u>Decision Tables</u> seem to hold promise of meeting all of these needs of a computer language. Decision tables provide a graphical

1.4

representation of complex decision procedures in a way that is easy to visualize and understand. They show alternatives and exceptions much more explicitly than other languages. They present relationships among variables clearly. They show the necessary sequences of conditions and actions in an unambiguous manner. The language of decision tables can be used equally effectively for system analysis, procedure design, and computer coding. Thus, a computer procedure written as a set of decision tables is, to a large extent, its own documentation.

There is a growing body of experience to indicate that these claims are justified. Those who have used decision tables for man-tomachine work say that programming is much faster and that program checkout time is significantly reduced. They say that the use of tables leads to greater accuracy and completeness in problem formulation. They say that program maintenance is simpler, and that a program written in tabular form is indeed a powerful communication and documentation device.

For all these reasons it has seemed worth while to develop a Tabular-form Descriptive Language; i.e., a language using decision table format to structure the communication of procedures and systems solutions.

To our knowledge this is the first attempt to describe a man-toman language with the kind of precision that computer languages possess,

1.5

while retaining the ability to describe the system logic rather than the details of a particular solution.

The Language Concept

In developing such a formal language there are three major considerations: Purpose, Structure and Content.

<u>Purpose</u>: The purpose of TDL is to permit systems engineers to clearly and effectively think through the system design logic, record this logic, and then communicate⁽it to other systems engineers or programmers. It is not a machine-level or programmer-level language and it is not intended to be implemented automatically. TDL is expected to provide a clear statement of the solution logic so that a good programmer can prepare an efficient program. TDL must be usable at various levels of detail for all types of business data processing. It must, therefore, have features suitable to the various kinds of problems found in business systems: report preparation, decision making, arithmetic processes, file maintenance, etc.

With this broad a purpose, it is evident that the initial TDL can only establish a foundation; extensive work will be required to add extra features as they are identified, and clarify aspects for situations not currently visualized. Much, therefore, depends on those who use TDL at this experimental stage. Their evaluation, comments, and recommendations will serve to guide future development so that the full potential of an advanced systems language can be realized. <u>Structure and Content</u>: The structure of the language sets up two classes of information:

- (1) Operand Descriptions to enable the systems engineer to define the meaning and use of various field names, constants, etc.; and
- (2) Decision tables to describe the data transformations.

While operand descriptions are common to many current machineoriented languages like Autocoder and COBOL, the attempt here has been to allow a quite simple and natural way for identifying and specifying operand names only to the level of detail absolutely necessary for clear communication. The only real requirement is that names of files, records, fields, etc. be identified to avoid their inadvertent reuse for other operands. The balance of the descriptive material is only needed for actual communication to the programmer who has to layout and code the job for a particular machine.

To indicate how the idea and nature of the operand descriptions permit a high degree of flexibility and convenience, let's review a few of the features.

> Virtually any convenient name can be used for an item of data. Common names can be used in many records and distinguished by simply indicating the record name.

Names can represent discrete functions of one or two arguments — such as, airplane fare between two points.

Names can represent the value of complex arithmetic formulas.

- Names can represent Boolean-algebra type logical expressions.
- Names can represent lines of information resulting from putting together the values from different fields.
- Commonly accepted words can be used without definition, such as Branch Office Manager's name, current month, etc.
- Work fields which can be created and changed at will do not require formal definition.

The second basic element of the structure of the language is the one of decision tables. To understand this, let's look at the simple table shown in Figure 2.

	Rule 1	Rule 2	X//X	Rule 30
age	25-34	25-34	SIX	≥65
health	"excellent"	"excellent"	X/X.	"poor"
section of country	"east"	"west"	XX	"west"
rate per \$1000	1.57	1.72	S/X	5.92
policy limit	200,000	200,000	\$/A	20,000

Figure 2. An example of a decision table.

The first decision rule can be paraphrased: <u>If</u> age is between 25 and 34 inclusive, <u>and</u> health is excellent, <u>and</u> section of country is east, <u>then</u> rate per thousand dollars is 1.57 <u>and</u> policy limit is 200,000. The underlined words are implied by the table layout. The quote marks are used to differentiate non-numeric values from variable names. The other rules are alternatives to this one, so that logically, it doesn't matter which rule is examined first; only one rule can be satisfied in a single pass through this decision table.

The information in Figure 2 is shown in an exploded view in Figure 4, to show more clearly the parts of a table and the terms that are used to describe them. The double lines serve as demarcation: CONDITIONS are shown above the horizontal double line, ACTIONS below; the STUB is to the left of the vertical double line, ENTRIES are to the right. Each vertical combination of conditions and actions is called a RULE. The essential nomenclature is completed by adding at the top of the table a title section, called a TABLE HEADER, and by adding a RULE HEADER over the entries.

Tables may be used in a slightly different way to state decision logic, as shown in Figure 3.

	Rule No.	1	2	3	4
credit limit ok		Y	N	N	N
pay experience favorable			Y	N	N
special clearance obtained				Y	N
approve order		X	X	x	
return order to sales					Х

Figure 3. A limited entry table

1.9



ENTRY



1.10

The first rule (columns 1 and 2) is read: <u>If</u> credit limit ok, <u>then</u> approve order. Again, the underlined words are implied by the form. You may note here that the form of the individual condition or action is somewhat different. In a limited entry condition or action, the entire condition or action must be written in the stub; the entry is <u>limited</u> to asserting, reversing, or ignoring a condition or executing an action. In contrast, an <u>extended entry</u> form (as in Figure 2) has part of the condition or action <u>extended</u> directly into the entry. Both forms may be used within one table, but any one condition or action row must be of just one form.

You may note from this example that the basic concept of a single rule in a table is quite straightforward, being based on the "if. . . then. . . " relationship. If A = B, and C is greater than 5, and . . . then assign the value 7 to X, and GO TO Table 10. If the conditions in rule 1 are not met, then try rule 2. If none of the rules succeeds, then something is wrong. If we have considered all significant possibilities, and we want to indicate, "Go ahead anyhow with a special routine," then we indicate "ELSE" in the last rule; otherwise, we provide for an automatic error routine. The flow chart in Figure 5 shows schematically the way in which a table is "executed." In practice the actual solution technique may vary, but the logical result remains the same.



Figure 5. Schematic representation of the sequence of tests and actions in executing a table.

1.12

For user convenience decision tables may be referred to by name or identification code. Groups of tables are referenced by a procedure name. Until the job is actually turned over for programming many of these tables and procedures may remain undefined. They are simply an indication that such a routine will be supplied. This also offers the opportunity of using named tables to define a particular routine which is used many times.

Use of TDL

The general principles of decision tables have been applied to the development of a Tabular-form Descriptive Language (TDL) for manto-man communication. As such, TDL is not a processable language, but it is intended to be precise and comprehensive. Certain operations which only have meaning for computer coding have been omitted and will have to be added by the programmer. The primary goal of TDL is to replace flow charts and narrative description of problem solutions and decision logic. It is not intended to replace block diagrams used to show physical relations or essentially pure referential processing work. But where there are significant conditions and branches, and where there are complex decisions involving many factors TDL permits more rapid, accurate, effective solution formulation and description with attendant savings in computer programming and machine space and running time.

TDL uses decision tables to describe the decision logic involved in a systems problem. The tabular form is essential to decision tables.
It provides a means of briefly, unequivocally and completely describing the requirements of an information processing system in a language which is easy to write and read. Because the language is also rigorously defined and the forms on which the tables are described have an effective structure, brevity is possible. The language content and structure enable the user to describe complex, logical solutions with a relatively small amount of writing. The nature of decision tables induce the user to make complete descriptions of the information processing requirements. As a corollary to this advantage is the fact that what is written in tabular language may be readily checked for completeness and correctness.

Who then will use TDL? Primarily the systems engineer. The systems engineer performs the function of systems analysis and development. While he may later act as a programmer, when he is doing systems analysis he is concerned with the system on a level above that of programming a computer. The system with which a systems engineer is concerned is some segment of a company's total information processing. It is the function of the systems engineer to analyze the requirements and to develop the necessary detail to describe the information processing system.

While no one seems to know just how to go about solving a particular data processing problem, the following steps always seem to come up:

- 1. Describe input and output fields, records, and files;
- 2. Describe printed reports, card and tape content;
- 3. Define constants, expressions, lists; and
- 4. Determine transformation logic to indicate preparation of outputs as a function of inputs.

There may be substantial iteration in these steps, but they have worked well enough in the past to serve as an organizing device for the Tabularform Descriptive Language.

The systems engineer will describe the data which is input to or output from the system and the processing of the data which must be accomplished. Data processing consists variously of recording data without change, validating data to show that it is correct, operating on data based on certain conditions, and manipulating data arithmetically. It is obvious that a description of such processing does not require explicit computer instructions or computer hardware or that a particular computer be used; it's only necessary that the systems engineer describe fully what the data is and what is to be done to the data, in a language which is as close as possible to the language in which he thinks about the problem and its solution.

As the systems engineer determines what the inputs and outputs of the system which he is developing will be he may describe these using the operand description facilities of the Tabular-form Descriptive Language. As the systems engineer determines what processing must be done on the information in his system, he will describe this using decision tables.

Summary

This then is a general introduction to TDL, a Tabular-form Descriptive Language. Intended for precise man-to-man communication it is based on separating the operand definition from the transformation logic. The transformation logic is expressed in a tabular form for convenience, completeness, and clarity.

The balance of the TDL manual explains the details of the language and the forms used for describing systems solution in the language.

The next two chapters present a teaching text description of the language with a number of sample problems. The remainder of the manual provides a reference description of the language elements and forms. 1

CHAPTER 2

Using the Tabular Descriptive Language

The purpose of language is effective communication. Consider:

John is big

as a language statement. It satisfies the requirements of a communication cycle in that a completely defined thought is transmitted by symbols over a medium (paper) to a receiver (the reader) who understands it.

For convenience, the format should be standardized, although the symbolism may vary. Now consider:

Jean est gros

which has the same meaning as the earlier statement, except that the symbolism is slightly different. The format, however, remains exactly the same in that the leftmost something is related to the rightmost something by the something in the middle.

The Tabular Descriptive Language (TDL) likewise promotes effective communication from sender to receiver by conveying defined thoughts in symbolized language statements through a standard format. This chapter is addressed to an explanation and description of the TDL standard format.

In treating the subject, the examples and text will simply be illustrative of its potential, since a complete analysis of all the applications,

TYPE CODE	EQ	7								2								3							L
SET MODEL CODE	EQ	1	c	N	Y	τ	B	L	1	1	H	R	D	T	0	P	1	1	S	T	A	W	G	N	1

Figure	1.	Decision	Table,	Example	1.

situations, and problem solving approaches which the language is capable of handling is too extensive for coverage here. Consequently, this chapter should be read thoroughly to acquire a general understanding of language techniques before proceeding to the more rigorous treatment of the language itself in later chapters.

Concepts of the language are introduced through a series of examples which grow progressively more difficult; the associated narrative points out and emphasizes the rules of format. Periodically a group of questions is inserted which permit the reader to test his understanding of the material he has just studied. In this manner, the reader may proceed to succeeding pages at his own pace.

Example 1 (all figures will appear on left facing pages and will be repeated to coincide with the narrative) is an excerpt from one form used in TDL, showing a simplified version of a <u>decision table</u>. It might have applied to a situation where it was necessary to convert code classifications of automobile styles to an understandable abbreviation. The table reads, in part:

IF the type code of the car body	i	s	1
THEN the model	i	s	a convertible
	0	r	
IF the type code of the car body	i	s	2
THEN the model	i	s	a hardtop

2.2

etc.

TYPE CODE	EQ	1					14			2								3							
SET MODEL CODE	EQ	~	c	2	Y	т	B	L	1	1	H	R	D	Τ	0	P	1	~	S	T	A	w	G	N	1

Figure	1.	Decision	Table,	Example	1.

Each of these statements is known as a <u>rule</u>, which indicates the relationship between a <u>condition</u> and an <u>action</u> in tabular language.

IF a condition or set of conditions is satisfied,

THEN the corresponding action(s) are executed.

A rule further is composed of statements located in the 4 major areas of a decision table: condition stub, condition entry, action stub, and action entry, each one separated from the others by horizontal and vertical double lines.

> Decision Table Structure



This sketch locates and identifies those major areas in a decision table where information may be assigned for later reference.

The number of rules included in a decision table is limited only by the requirements of the system. The condition stub and the action stub are common to all the rules, but condition and action entries vary from one rule to the next. In example 1, there is only one condition and one action for each of three distinct rules. In general, a decision table will involve a series of conditions with a related series of actions for each rule.

TYPE CODE	EQ	1					×			2								3						Γ	
SET MODEL CODE	EQ	~	c	2	Y	Т	B	2	1	1	H	R	D	Τ	0	P	1	~	S	T	A	w	G	K	, /

Figure 1.	Decision	Table,	Example	1.
-----------	----------	--------	---------	----

Each statement in a decision table is constructed from a series of components of two different types: <u>operators</u> and <u>operands</u>. Each has a distinct function depending upon the context of the statement. The method of their relationship in a condition row for a single rule is:

operand 1, operator, operand 2

The relationship in an action row is:

operator, operand 1, connector, operand 2.

To show this relationship, rule 1 (rules are numbered consecutively starting with the first entry column after the vertical double line) is written as follows:



Operands in TDL are of many forms. All have the common property of representing something and of being the subject or object of some statement. Consequently, they can be considered the equivalent of the noun in English grammar rules. With the same reasoning, the operator may be assumed to take the place of the verb. A condition statement, then, can be considered in the form:

noun (operand) verb (operator) noun (operand) In rule 1, TYPE CODE and MODEL are operands which are the names of certain fields. These operands are known as <u>field names</u> and

TYPE CODE	EQ	7								2								3							
SET MODEL CODE	EQ	`	C	N	Y	τ	B	L	1	1	H	R	D	T	0	P	1	~	S	T	A	W	G	N	1

Figure	1.	Decision	Table,	Example	1.

represent data variables. These operands come under the general category of <u>named operands</u>, and identify information that is peculiar to the system. In the same rule, the symbols 1 and 'CNVTBL' are also operands. However, these operands are known as <u>literals</u> since they represent literally that which they say, and are classified under the general category of <u>self-describing operands</u>. These may be used universally for all systems without additional definition since they directly communicate the value of the operand.

The distinction between a named operand and a self-describing operand is important, since one implies a variable value and the other a fixed value. Therefore, a method must be devised for differentiating between them. This is accomplished by the restriction that a named operand must start with a letter. Literals, when they are <u>alphameric</u> (as 'CNVTBL') are enclosed in quotation marks, so that they are not mistaken for named operands. <u>Numeric literals</u> need not be enclosed in quotation marks as they cannot be mistaken for field names.

In this table, two kinds of operators are displayed, one each in condition and action statements:

- The first, written as EQ, is referred to as a <u>condition</u> <u>operator</u>. Functionally, it relates the field name TYPE CODE to the numeric literals, 1, 2, or 3 in the 3 rules and therefore acts as a <u>relational operator</u>.
- o The second, recognized by the SET... EQ... notation is an action statement. It performs the function of des-

2.5

TYPE CODE	EQ	1					•			2								3							L
SET MODEL CODE	EQ		C	N	Y	τ	в	L	1	1	H	R	D	Т	0	P	1	1	S	Т	A	W	G	N	1
																						1			

Tranto T. Doorore Topic, Drottipic T.	Figure	1.	Decision	Table,	Example	1.
---------------------------------------	--------	----	----------	--------	---------	----

cribing actions to be followed if the conditions of a rule are satisfied. For this reason, it is more specifically known as an <u>assignment operator</u>.

In this case, the action is to transfer one of the abbreviations ('CNVTBL', 'HRDTOP', 'STAWGN') to the field named MODEL CODE, depending upon the rule satisfied.

The manner in which a decision table is manipulated consists of investigating individual rules until the conditions for a rule are satisfied. Either rule 1 may be satisfied or rule 2, or rule 3, but not more than one at a particular time.

QUESTIONS

- 1. How do you read the 3 rules in Example 1 using English grammar?
- 2. What happens if rule 3 is considered before rule 2 in Example 1?
- 3. What is the difference between a literal and a field name? How is the difference shown in a table?
- 4. What does the action statement SET...EQ... call for in this example?

TYPE CODE	EQ	1								2								3							
SET MODEL CODE	EQ		c	N	Y	Т	B	L	1	1	H	R	D	Т	0	P	1	~	S	T	A	W	G	N	1

Figure 1. Decision Table, Example 1.	Figure	1.	Decision	Table,	Example	1.
--------------------------------------	--------	----	----------	--------	---------	----

ANSWERS

 Rule 1: If the type code of the car body is 1, then the model is a convertible.

Rule 2: If the type code of the car body is 2, then the model is a hardtop.

Rule 3: If the type code of the car body is 3, then the model is a station wagon.

- 2. There is no change in the results. The rules may be investigated in any order, regardless of position in the table.
- 3. The field name represents a variable that changes according to the information placed in it. A literal is its own value. Alphameric literal names, which could be mistaken for field names are differentiated in the table by enclosing them within quotation marks (' ').
- The SET...EQ... action moves an abbreviation, which is a literal that describes the car body, to the field named MODEL CODE.

CISION TABLE HOLT 11.1.1.0. 19.4.1.1.1.1.0. 19.4.1.1.1.1.0. 19.4.1.1.1.1.0.1.1.1.1.1.1.1.1.1.1.1.1.1.1
BLE TYPE DEELAND2 OP OP OPERAND2 OP OP OPERAND2 OP OP
OPEN 00 1 1



Example 2 shows the complete TDL Decision Table form. Here, the situation deals with a proxy report, especially that part dealing with the selection of stockholder groups.

The heading, SYSTEM NAME, is filled in with the particular name, PROXY REPORT. A system description consists of a group of decision tables, each of which is assigned a name and an identification. The entire system description defines a complete information processing task. A decision table defines a segment of that analysis. In this case, the TABLE NAME is SEL STKVALUE GRPS and the TABLE IDENTIFICATION (usually a 3 digit number) is 005, but the table is a part of the overall system called PROXY REPORT.

A system description, composed of many decision tables, must have some method of indicating the sequence of going from one table to the next. This is performed through the use of the operator GO TO in the last action row. The operand for this operator is the table name KEY REPORT. (The table identification could also have been used as the operand.) It is assumed that some previous table or tables have been completed for this system and the last action of the previous table considered was GO TO SEL STKVALUE GRPS, the name of this table. Likewise, after this table is completed, the last action called for is a transfer of control to the table named KEY REPORT. At the termination of the processing, transfer of control is made to a table named STOP.

2.8

- M	F	~	34 23			TI	-	<u>B-</u>	
DATE	CY	DP " OPERAND	16 16 30 31 33 38				SROKER'	SOKER STOCK	
HOLT	Naucasa	OP OF OPERAND2 C	04 10 11 12 15 14 19 16 17	22 03	2		BANKI	ANN STOCK 88	
DECISION TABLE		OP CORRAND 2	0 01 01 04 05 06 01 00 0		· · · · · · · · · · · · · · · · · · ·		T NOIVI	INDI NI DU AL STOCK	
PROCEDURE NAME	TABLE TYPE	OP OPERAND2	OPEN	EQ	GR 100000		EQ	to	KEY REPORT
SYSTEM NAME PROXY REPORT	TABLE NAME	OPER OPERAND 1	SEL STKVALUE GRPS	TYPE STKHLOR CODE	SHARES		SET TYPE STKHLDER	MOVE STKYALUE	GO TO
	TANK	SER	8	4	~		Ю	3	5

Figure 2. Decision Table, Example 2.

-¥

This table is classified as an <u>open table</u>. All open tables must be entered by a GO TO operator and must contain a GO TO operator as the last action of the table, the only exception to this rule being the STOP table.

This example illustrates another difference in format as compared to Example 1. In the previous example, the second operand of all conditions and actions were written in the entry portion of each rule, a fact that classifies it as an <u>extended entry table</u>.

Where the complete condition is written in the stub of the table, and the affirmation or negation of the condition is indicated by a Y or an N in the rule column, the table is referred to as <u>limited entry</u>. It is possible that a rule does not require the condition. In this case, the column has no entry. For example, the second condition row for the three rules of this table could be roughly translated:

Rule 1. the number of shares is more than 100000

Rule 2. (regardless of the number of shares)

Rule 3. the number of shares is not more than 100000 The limited entry format may also apply to actions. In this case, the complete action is written in the action stub, and is to be executed if an X is placed in the column for that rule. If the action is not pertinent to a rule (not to be executed), the entry for that rule is left blank. In this example, all three rules require the action GO TO KEY REPORT. Since

	L	EVETEM NAME	PROCEDURE NAME	TREM	ANALVST	DATE	_
		PROXY REDORT	J	DECICION TABLE	HOLT	112/62	PAG
	AALL I	I maken in according	and a rector				
	2	TABLE NAME	IABLE - YYS				+
	888	OPER OPERAND 1	OP OPERAND2	OP OPERAND 2	OP 10111111111111111111	17116 19 201 21 22 82 12	12
*	8	S SEL STKVALUE GRPS	OPEN				-
	4	TYPE STHHLOR CODE	EQ	CU III	02	03	Н
	~	SHARES	GR 100000			2	-
	M	SET TYPE STICHLDER	EQ	/ AIGW T.	PBANK	NGROKER!	-1
	3	MOVE STKYALUE	4	TNDI NI DU AL STOCK	BAIN STOCK	BRDKER STOCK A	+
				TU UONA	BROUNT	MOUNT	-
	5	60 TO	KEY REPORT	×		X	+
							+
							+
							-
							r
							+
							+
							T
							F
							-
							Ť
							+
							1
							+
							+
							Ţ
							+
							+
	10	SEMADICE.					
	-	VERYPORTS.					ĩ
							I

Figure 2. Decision Table, Example 2.

this action applies for all three rules, an X is written in the action entry for each rule. To qualify as either limited or extended entry, a decision table must follow the arrangement rules for each consistently.

A third tabular format is known as <u>mixed entry</u>. Example 2 is a mixed entry table. That is, some of the conditions and actions are in extended entry form while others are in limited entry form. The only restriction in the use of mixed entry is that a single condition row or action row must be consistent. It must be either all limited entry or all extended entry. The two cannot be intermixed in <u>one</u> condition or action row.

The condition part of rule 1 asks: ... "Is the TYPE STKHLDR CODE equal to 01 and are the number of shares greater than (GR) 100000?"... The operator GR is another relational operator. In order for the second condition to be satisfied, the question "Are the number of shares owned greater than 100000?" must be answered yes in rule 1. The full list of relational operators, with their abbreviations, is given below:

LR	less than (lesser)	LE	less than or equal to
GR	greater than	GE	greater than or equal to
EQ	equal to	NE	not equal to

These relational operators could be expressed symbolically (e.g.,>,=), but the alphabetic notation will be followed throughout this manual.

2.10

	X	Г	~	31				1	L i	1			1	T	\mathbf{t}	I	í a		1	1		1			1
DATE	112/62 1	IENCY	2 OP " OPERAND 2	10 17 15 19 30 31 33 35 34	03	2	P.B.ROKER	MOUNT DIDOM H-	X																
ANALYST	HOLT	LAREQU	OP OF OPERAND	0 0 10 11 13 19 14 19	02		BANK	ANDUNT THOMA																	
IBM	DECISION TABLE		OP COERAND 2	01 02 03 04 06 06 01 00	1		T NON /	PUDU NT DU HE STOC																	
PROCEDURE NAME		TABLE TYPE	OP OPERAND2	OPEN	EQ	GR 100000	EQ	2	KEY REPORT																
SYSTEM NAME	PROXY REPORT	TABLE NAME	OPER OPERAND 1	SEL STKVALUE GRPS	TYPE STRALOR CODE	SHARES	SET TYPE STICHLDER	LINKE JINYHLUE	GO TO				-											EMARKS:	
		TABLE	Na Sa	8	T	~	m =		5											Γ				RE	

Figure 2. Decision Table, Example 2.

Rule 2 reads:

IF the type stockholder code is 2

THEN the type of stockholder is a bank

and

move the stock value to bank stock amount

and

the table KEY REPORT is entered.

This rule introduces the MOVE operator with the connective TO which works as an <u>assignment operator</u> just like the SET...EQ... action. It moves the quantity specified by the first operand (STKVALUE) to the field named in the second operand (BANK STOCK AMOUNT). The differences between the SET and MOVE assignment operators may be stated as follows: SET...EQ... is used when one of several quantities in the second operand field is to be transferred to the first; MOVE... TO... is used when a quantity in the first operand field is to be transferred to one of several fields named by the second operand.

In the action row containing the MOVE operator, the entries are INDIVIDUAL STOCK AMOUNT, BANK STOCK AMOUNT, BROKER STOCK AMOUNT. If the names do not conveniently fit a line, they may be hyphenated or broken where most convenient. They are then continued on the second line of the rule, as illustrated in this example.

-		EVETEM NAME	PROCEDURE NAME	IBM	ANALVST	DATE	_
-		Down or other		9	110.1	110100	DAG
		TROXY REPORT	J	J DECISION TABLE	HOLI	1 1 1 6 1	NMA.
and the same	TABLE	TABLE NAME	TABLE TYPE		US BAS	NCY NO	
Sanifation of the local division of the loca	SSR	OPER OPERAND 1	OP OPERAND2	OP " OPERAND 2	OP OF OPERAND	OP OPERAND 2	~
in the second				01 02 03 04 06 06 07 08	09 10 11 12 15 14 19 1	17 18 19 20 21 22 25 24	21
¥	8	SEL STKVALUE GRPS	OPEN				+
	T	TYPE STHALOR CODE	EQ	8	02	03	+
	~	SHARES	GR 100000	~		2	-
distant.							F .
	3	SET TYPE STKHLDER	EQ	/ I/IGN I	PBANK'	PGROKER'	-
	+	MOVE STKYALUE	4	INDINI DUAL STOCK	BANK STOCK	BROKIER STOCK A-	-
				PNDU NT	ANOUNT ANO	MOUNT	_
	5	GO TO	KEY REPORT	X	X	X	4
	*						_
							-
							_
							r .
							T
							-+
							_
							1
							F
							1
							_
							4
							_
							-
1	æ	REMARKS:					



QUESTIONS

- 1. Read rule 3
- 2. What action is called for by the MOVE...TO... action in rule 3?
- 3. How does the action of the MOVE...TO... compare with the action of the SET...EQ...?
- 4. In rule 1, if the number of shares is equal to 100000, are the conditions of the rule satisfied? Why or why not?
- 5. What are the requirements of an open table?

PAG	IE	~					.		I
112/62		02 OP " OPERAND 2	4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	03	2	halankaa '	BRDNER STOCK A-	NOUNT	
HOLT	FREQUE	OP OP OPERANC	et Ar et 91 11 01 60 0	02		NRIANK'	K BANK STOCK	ANDONT	
TECICION TABLE		OP OPERAND 2		CCT I	X	T NOIV /	INDINI DUAL STOCI	PNDU NT	
PROCEDURE NAME	TABLE TYPE	OP OPERAND2	OPEN	EQ	GR 100000	EQ	10		
SYSTEM NAME	TABLE NAME	OPER OPERAND 1	SEL STKVALUE GRPS	TYPE STHHLOR CODE	SHARES	SET TYPE STKHLDER	MOVE STKYALUE		EMARKS:
0	1	24	8	1	×	M	-		1 a

Figure 2. Decision Table, Example 2.

ANSWERS

- If the type number of the stockholder code is 03, and if the number of shares is not greater than 100000, <u>then</u> the type of stockholder is a broker, and broker stock amount receives the stock value, and another table called KEY REPORT is entered.
- The stock value is placed in the data field broker stock amount.
- 3. Their action is opposite. MOVE transfers a quantity from the first field to one of several fields named by the second operand, while SET transfers quantities from one of the possible second operand fields to a field named by the first operand.
- No. The condition used with the relational operator is that the number of shares must be greater than 100000. Equal to this number does not satisfy the condition of the rule.
- An open table is entered by a GO TO operator, and must conclude with a GO TO operator.

	TABLE	TABLE	NAME	TAS	LE TYPE			T				T			5	R	EQ.	UE	N	CY	T	T			Т	Т	T	T		П	T
	SER	OPER	OPERAND 1	OP	OPERAND 2	0	P	(PE	RAP	D 2	10	P	08	0	PER	LAN	DZ	0	P	0	ER	ANI	0 2		0	> 04	OP	RA	ND :	L
*	005	SALES EX	TENSION	1 0	PEN	04	01 0	2 04	04	06	67 00	04	10	11 1	2 1	1	18	16	17	18	19 3	0 21	32	23	24 3	5 5	4 1	20	19	80 3	32
	1	INVENTORY T	RANSACTIONS	ISEN	D OF FILE	N						N							N			T				1					
	2	ITEM NO	: SALES	EQITE	MNOINVEN	Y						Y							N							Τ		T			
	3	GTY SOLD)	LE QT	ON HAND	Y		-			_	N										1			_	-	-				
	4	SET SALE	S QTY SHIP	EQ			a	TY	50	LD		+		ary	0	VI	IAN	D			+	+			+	+	+	+	Η	-	+
	5	SET GTY	BACKORDER					T				EC		ath	S	DLI										T	T				
												-		(TD)	0	N	HAN	Þ											1		
	6	SET QTY	ISSUE			+	a	٢Y	30	D		+		qTh	0	N	AN	0				T					T	T			
	7	MOVE UNI	T PRICE	TOS	ALES AMT	X	+	Ŧ			-	X			+	+	-	-			-	-	-		+	Ŧ	Ŧ	F	\square	+	F
	8	SET QTY	ON HAND			-	G	ΓY	50	LD		EG		ZER	0						1						T				
	9	GOTO				TR	ANS	IN	-		_	TR	AN	31	N				GE	IX	1	-	-		Ď	HI	PM	ENT	R	ELE	ASE

Figure	3a.	Decision	Table,	Example	3.

t

Example 3 is concerned with an inventory and billing system. It applies to that part of the system which processes sales transactions against inventory, checks the quantity sold against the inventory record to establish the quantity to be shipped or back ordered, applies a price to the quantity shipped to extend the sales amount, and updates the inventory record by the quantity shipped in the sales transaction.

The first condition of the first rule is:

IS INVENTORY TRANSACTIONS AT END OF FILE ...? This illustrates the use of a state <u>conditional operator</u> and a <u>state term</u>. The state conditional operator relates a state term to an operand name of something that varies. There are two such operators, IS and NT (is NoT). In this example, the first operand is the file name, INVENTORY TRANSACTIONS, and the operand which is related to it by the operator, IS, is the state term END OF FILE. This entry indicates that when the answer is yes, there are no more records available. The END OF FILE operand may also be abbreviated EOF.

Prior to discussing the remainder of this example, it is necessary to digress and discuss some of the types of named operands: <u>files</u>, <u>records</u>, and <u>fields</u>.

 A file is a collection of one or more records and has three characteristics:

(1) medium - e.g., card, disk, etc.

- (2) directional type input, output, retained, intermediate.
- (3) access type serial, random

TABLE	TABLE	NAME	TAP	LE TYPE		1								F	RE	Q	UE	N	CY	•]	T	T				Τ		T	T	T
SER	OPER	OPERAND 1	OP	OPERAND2	OP	81	0	PE	RAN	D2	0	p '	58	OP	ER.	AN	DZ	0	P°	0	PER	AN	DI	2	0	P	10	PER	ANC	2
					01 0	01 01	104	10	04	07 08	04	10 1	1 12	115	14	15	16	17	16	14	10 1	1 23	29	14	25	2.6	17 3	28 1	9 30	31 3
005	SALES EX	TENSION	0	PEN																-				-						
1	INVENTORY"	TRANSACTIONS	IS EN	D OF FILE	N						N							N							Y					
2	ITEM NO	: SALES	EQITE	MNOSINVEN	Y						Y							N												
3	QTY SOL	D	LE QT	Y ON HAND	Y	+			_	-	N	-						_		-	-	-						-	-	
4	SET SAL	ES OTY SHIP	EQ			GT	Y	30	LD	+		G	TY	UN	H	AN	D			1	+	+			1	+	+	+	+	++
5	SET GTY	BACKORDER	1			T					EQ	G	TY	50	LD													T	T	
											-	6	ΨY	10	H	AN	P													
6	SET QTY	ISSUE			+	QT	Y	50	D		+	G	TY	ON	H	An	D				T					T		T	T	
7	MOVE UN	T PRICE	TO S	ALES AMT	X	-	-			+	×	-	-	-	F				-	-	+	-	-	F		-	+	-	Ŧ	
8	SET QTY	ON HAND			-	GT	Y	50	LD		EQ	7	ER	0														+	+	
9	GOTO	·			TRA	NS	IN				TR	AN	she	N				GE	TX						SH	IPI	MEN	TY	REL	EAS

Figure 3a. Decision Table, Example 3.

SYI INV	ENTORY, BILLING]		FILE	DE	SCF	M	ION				FRANC	ST	DATE 1/2/6	2	PAGE	OF	PAGES
FILE	NAME	DIF	2 T	YPE		ME	DIUM		R	RECO	ORI	NAME		ABB	AVERA	NO OF	RECOR	IDS
INVE	NTORY TRANSACTIONS	IN	-			SE	R-C	ARD	1	SAL	.E.	s		-	80	00	10	000
NO	FIELD NAME	MUMANDAC	BIGN		TH	OCCI	AVG	MAX	ASSOC	ALLOC TO	TROUGHCE	RELATIVE	END	V	ALUE/P		₹KS	
1 2 3	ITEM NO QTY SOLD	ZZ	1)+	7 4				1										
4 5	OTY BACK ORDER SALES AMT	NN	+ V	4				1 1		-						_		
			-	-	-	2		~			-	-					-	

Figure 3b. File Description, Example 3.

SYS	ENTORY, BILLING]		FILE	DE	IB		ION				FRANC	ST	DATE	-()	PAGE	OF	PAGES
FILE	NAME	DIS	2.7	YPE		ME	DIUM		R	ECO	R	NAME		ABB	AVIRA	NO OF	RECO	RDS
SER	FIELD NAME	IN INCOME	Sign	LEN	TH	OCCI	URRE	NCES	1 A5802	Z ASSOC	SEGUENC	RELATIVE	LOCATION		ALUE/F	REMAI	RKS	200
1233450700	ITEM NO UNIT PRICE OLD BAL ON HAND QTY RECEIPTS QTY ISSUES QTY CN HAND QTY ON ORDER QTY MINIMUM ITEM DESCR	ZZZZZZZ	U++++++	7 3.2 5 5 5 5 5 5 5 5 20				111111111111111111111111111111111111111								****		

Figure 3c. File Description, Example 3.

- A <u>record</u> is a collection of one or more fields. It
 assumes the attributes of the file to which it belongs. A
 File Description form should be completed for every
 type of record, listing all the fields which make up the
 record.
- A <u>field</u> is a named piece of information with value and can be part of a record.

File names, record names, and field names are operands in a decision table. On figure 3b, the file description of a record of the file named INVENTORY TRANSACTIONS is shown. This form describes the record, SALES, of that file. The record is composed of five fields. The file name, INVENTORY TRANSACTIONS, the record name, SALES, and the field names, ITEM NO, QTY SOLD, SALES QTY SHIP, QTY BACK ORDER, and SALES AMT, may all be used as operands.

In figure 3c, a file description is shown for the record, INVENTORY, which is part of the file, INVENTORY MASTER. This record has nine fields. The headings for figures 3b and 3c are explained as follows:

<u>System Name</u>: The name of the system of which the file description is a part. In this example, both forms are part of the system INVENTORY, BILLING. The decision table is also part of this system.

File Name: The name of the file, as described above.

TABLE	TABLE	NAME	TA	LE TYPE							Т	Т			FR	28	ရပ်	EN	ve	Ý					T	T	T		Т		
SER	OPER	OPERAND 1	OP	OPERAND 2	OP	81	0	PE	RAH	D 2.		OP	OL	4	OP	RA	ND	2	OP	63	OPE	RA	ND	2		OF	3 04	OP	ERA	ND	2
-					01 0	01 01	04	04	04	07 06	0	10	11	12	15	14	15 1	6 1	7 11	19	10	21	22	25 3	4 2	5 2		7 21	1 19	80	31 33
005	SALES EX	TENSION	0	PEN														L		1	1								1		
1	INVENTORY	RANSACTIONS	13 EN	D OF FILE	N						N							h	1)	1					
2	ITEM NO	: SALES	EQITE	MNOUNVEN	Y						Y	1						1	4												
3	GTY SOLI	>	LE QT	ON HAND	Y	+	+			-	N	4		_		-	+	+	+	-	-		-	+	+	\downarrow	+	+	-		-
4	SET SALE	S OTY SHIP	EQ			GT	Y	30	LD				QT	Y	ON	H	WD				-			+	+	+	+	1	+	\square	+
5	SET GTY	BACKORDER									E	Q	QT	Y	50	D		T	T	T											
						-					-	1	GT	Y	ON	H/	AND										T				
6	SET QTY	ISSUE			+	ar	Y	50	D		4	-	QT	Y	ON	H	AND											T			
7	MOVE UNI	TPRICE	TOS	ALES AMT	Х	-					Þ	4		_			-	Ŧ	T	F	F			-	Ŧ	+	T	Ŧ	F	\Box	+
	* SALES	aty ship		C. C							1						_			-			_								
8	SET QTY	ON HAND			-	GT	Y	50	LD		E	9	ZE	RO			_	1					_								
9	GOTO				TRA	NS	IN				T	RAN	B	N		4	+	6	ED	1	-		-	-	Þ	HI	PM	EN	TR	FL	EAST
			-					-	-	-	-	-				-	_	+	-	-			_			+	+	+	+	-	

Figure 3a. Decision Table, Example 3.

SYS INV	ENTORY, BILLING]		FILE	DE	SCR		ION				FRANK	5T 215	DATE	2/6	2	PAGE	OF	PAGES
FILE	NAME	DIS	R.T	YPE		ME	DIUM		P	RECO	OR	D NAME		AB	8	AVERA	NO OF	PEAK	DS
INVE	NTORY TRANSACTIONS	JN	1		_	SE	R-C	ARD	1	SAL	Ē.	s			-	80	000	10	000
SER NO	FIELD NAME	Manifest	BIGN		TH	OCC	AVG	MAX	VSBOX	To	SHONENCE	RELATIVE	LOCATION		VA	LUE/F		RKS	
123	ITEM NO QTY SOLD SALES GTY SHIP	N Z Z	11++	74				1 1											
4 5	OTY BACK ORDER SALES AMT	NN	+ V	4				1 1							-	_			
L		1	+	-	-	h		m		1	-	-		_	~				-

Figure 3D. File Description, Example	Figure	3b.	File	Description,	Example	3.
--------------------------------------	--------	-----	------	--------------	---------	----

SYS	ENTORY, BILLING	3	1	FILE	DE	IB	M	ION				FRANC	.IS	(DATE //2/62		PAGE	OF	PAGES
FILE	NAME	DIR	1.7	YPE		ME	DIUM	1	R	ECC	R	O NAME		-	ABS	AVERA	NO OF	RECO	DS
INI	ENTORY MASTER	RE	TA	INED	2	RA	N-D	ISK	Ĩ	NV	E	NTORY			INVEN	4,0	00	5,0	000
SER	FIELD NAME	MUMARY.		LEN	TH	OCCI	AVG	MAX	ASBOC	ASSOC TO	TROUGHCE	RELATIVE	LOCATION		VAI	LUE/R		RKS	
1	ITEM NO	N	V	7				1											
3	OLD BAL ON HAND	NN	+	3.2	+	-	-	1	+		-	-						-	
4	QTY RECEIPTS	N	+	5				1						_					
6	QTY ON HAND	N	+	5				1			-							_	
7	QTY ON ORDER	NN	++	5	-			1	-		-			_					
9	ITEM DESCR	A		20	1			1									-		

Figure 3c. File Description, Example 3.

<u>Dir. Type</u>: One file is IN, the other RETAINED. <u>Medium</u>: The medium on which the file is contained. INVENTORY TRANSACTIONS is a serial card file, and INVENTORY MASTER is a random disk file.

Record Name: The name of that record which is described by the form. In these cases, SALES and INVENTORY.

<u>Abb</u>: This entry allows for the abbreviation of long record names. Such abbreviations may be used as an operand rather than the full record name.

<u>Average Records:</u> The average number of the record described on the sheet.

<u>Peak Records</u>: The peak number of the record on the sheet.

After this information is inserted, the field names for the record described are then listed in the field name column. The fields need not be described in any particular order. For each field, the specification must be made as to

- (1) whether the field is alphameric or numeric (A or N),
- (2) the sign possibilities for numeric fields (+, -, E, U)

TABLI	TABLE	NAME	TAS	LE TYPE		1	Γ		T		Γ			1	R	RQ	UE	N	CY										T		Γ
SER	OPER	OPERAND 1	OP	OPERAND 2	OP	-	0	PER	RAN	D2	0	P	08	0	PE	RAN	102	0	P°	0	PER	LAN	D	2	0	P°	10	PER	AN	2 2	
	-				01 0	01 01	04	94	26	07 08	09	10	11	12 1	5 14	6 13	16	17	18	14	20 1	1 17	1 23	14	25	24	17	18 1	9 30	n	32
* 005	SALES EXT	TENSION	0	PEN																*				•							Γ
1	INVENTORY T	RANSACTIONS	13 EN	D OF FILE	N						N							N							Y						
2	ITEM NO	: SALES	EQ ITE	MNOUNVEN	Y						Y							N													
3	GTY SOLD)	LE QTY	ON HAND	Y	-				-	N		-	-	+	-				+	+	-						-	+		
4	SET SALE	S OTY SHIP	EQ		+	QT	Y	30	LD	+	+	\square	art	10	N	HAN	D			+	+	+	+		H		1	+	+	\vdash	F
5	SET GTY	BACKORDER									EG		aTh	13	OL	D	T											T	T		
							Γ				-	1	QT Y	10	N	HAN	ND												T		
6	SET OTY	ISSUE			+	ar	Y	50	D		+		QT	10	N	HAI	D						T						T		
7	MOVE UNI	T PRICE	TOS	ALES AMT	Х	F	-		-	-	X	F	-	-	T	T	F	F		Ŧ	-	+	-	F		Π	-	-	Ŧ	P	F
8	SET QTY	ON HAND			-	GT	Y	50	LD		EQ		ZE	20														1	T		
9	GOTO				TRA	NS	IN		-	_	TR	AN	31	N	+	-	-	GE	TX	-	+	-			SH	IP	ME	TIN	RE	En	5E
-			-				-		-	-	-			_	-	-	-	-	-	-	-	-	_			-	-	-	+	-	

Figure 3a. Decision Table, Example 3.

SYS INV	ENTORY, BILLING]		FILE	DE	IB	M	ION				FRANK	5T 15	DATE	2/6	2	PAGE	OF	PAGES
FILE	NAME	DI	RT	YPE		ME	DIUM		R	ECO	OR	NAME		AB	8	AVERA	NO OF	RECOR	DS
INVE	NTORY TRANSACTIONS	JN				SE	R - C	ARD	1	SAL	Ē.	s			-	80	000	10	000
SER NO	FIELD NAME	and	S.an		TH	OCCU	AVG	MAX	A5800	ASSOC TO	STORESCES	RELATIVE	LOCATION		VA	LUE/		KS	
1 2 3	ITEM NO QTY SOLD SALES GTY SHIP	N Z Z	1/ +	7 4				1											
4 5	OTY BACK ORDER SALES AMT	N	+ V	4				1											
		1	1	-	1-	n		~				-			~		-	-	

Figure 3b.	File	Description,	Example	3.
------------	------	--------------	---------	----

SYS	ENTORY, BILLING]		FILE	DE	IB	M	ION				FRANC	IS IS	DATE 1/2/62		PAGE	OF	PAGES
FILE	NAME	DIR	LT.	YPE		ME	DIUM	i.	R	ECO	R	NAME		ABB	AVERA	NO OF	RECO	DS
INI	ENTORY MASTER	RE	TA	INED	>	RA	N-D	ISK	Ï	NV	E.	NTORY		INVEN	4,0	00	5,0	200
SER	CIELD NAME	15		LEN	TH	occi	URRE	NCES	1		1013	RELATIVE	LOCATION		115 /0			
NO	FILLD NAME	1 A	N.	MAX	VAR	MIN	AVG	MAX	20	000	ENCI	BEGIN	END	VAL	LUE/R	EMAN	KS	
1	ITEM NO	N	U	7	-			1		F	-							
2	UNIT PRICE	N	+	3.2				1										
3	OLD BAL ON HAND	N	+	5				1										
4	QTY RECEIPTS	N	+	5				1										
5	GTY ISSUES	N	+	5				11										
6	QTY ON HAND	N	+	5				1										
7	QTY ON ORDER	N	+	5				1										
8	QTY MINIMUM	N	+	5				1										
9	ITEM DESCR	A		20	1			1		-						1		

Figure 3c. File Description, Example 3.

- (3) the maximum total length for that field (and whether the field is of fixed length or variable length) and finally,
- (4) the maximum number of times that each field will appear on each record.

Serial 5 on figure 3b illustrates these points. The field name is SALES AMT which is a numeric value (N) of length 4.2. The number 4.2 indicates four integer and two decimal positions. If only an integer value is specified, then it represents the total number of characters. The field name SALES AMT occurs only one time per record.

Serial 9 on figure 3c also amplifies the discussion. The field name is ITEM DESCR, which has an alphameric value. Therefore, no sign possibilities need be noted. The maximum length of the field is 20, but it is variable. This field occurs only one time per record.

The second condition of the first rule of the decision table asks: Is the ITEM NO of the SALES record the same as the ITEM NO of the INVENTORY record...? This is a case where field names are qualified by their respective record names. In other words, the field name is given (ITEM NO) and then followed by a colon (:) and the record name (SALES). The field names are qualified because each of the two types of records contain identical field names. If the field names of a record are unique (never used in any other part of the system) then the name may be used without any qualification. For example, the third condition of rule 1 is:

2.17
| TABLE NA | ME | TAS | LE TYPE | | | | | | | | | | F
 | R
 | Q | UE | NC
 | Y |
 | | |
 | T | | | |
 | T | Γ |
|----------------|---|---|--|--|---|---|---|--|--|---|---|--

--
---|--|--|---|--
--
---|--|---
---|--
--|--|---|--|--|--|
| OPER OP | ERAND 1 | OP | OPERAND2 | 05 | 2 81 | 0 | PER | RAN | D2 | 0 | P | | OF
 | ER
 | AN | 02 | 0
 | > 01 | OP
 | ERA | MI | 2
 | | OP | 04 | OPE | RAN
 | 10 2 | - |
| | | | | 91 | 01 01 | 04 | 08 | 04 | 07 08 | 09 | 10 | 11 1 | 2 15
 | 14
 | 15 | 16 | 17 1
 | 6 1 | 9 20
 | 14 0 | 23 | 23 1
 | 4 21 | 5 1 | \$ 17 | 18 | 19
 | 10 11 | 132 |
| SALES EXTEN | SION | 0 | PEN | | | | | | | | | |
 |
 | | |
 | | 1
 | | | | | | | | | | | | | |
 | | | | |
 | | Γ |
| INVENTORY TRAN | SACTIONS | 13 EN | D OF FILE | N | | | | | | 4 | | |
 |
 | | | N
 | |
 | | | | | | | | | | | | | |
 | Y | 1 | | |
 | | T |
| ITEM NO: 5 | ALES | EQ ITE | MNOWNVEN | Y | | | | | | Y | | |
 |
 | | | N
 | |
 | | | | |
 | | | | |
 | | |
| GTY SOLD | | LE QT | ON HAND | Y | - | | | | | N | | + | -
 |
 | | | _
 | | -
 | | | | |
 | | | | |
 | | |
| SET SALES O | TY SHIP | EQ | | | GT | Y | 30 | LD | + | | 6 | πh | DA
 | 1 +
 | AN | D | -
 | + | +-
 | | | +
 | $^{+}$ | + | + | |
 | + | + |
| SET OTY BAC | KORDER | | | | | T | | | | EG | K | πY | SC
 | LD
 | | |
 | |
 | | | | |
 | T | | | |
 | - | |
| | | | | | _ | | | | | - | .* | YU | 01
 | 1
 | AN | D | 1-1-1
 | |
 | | | | |
 | T | | | |
 | | T |
| SET OTY 139 | UE | | | + | ar | Y | 30 | D | | + | | TY | 0
 | 1 1
 | AN | D |
 | | T
 | | | | |
 | Т | | T | |
 | | T |
| MOVE UNIT PR | ICE | TOS | ALES AMT | Х | - | - | | - | - | X | - | T | -
 | F
 | F | - | -
 | - | Ŧ
 | - | | -
 | Ŧ | - | + | - | -
 | + | F |
| SET QTY ON | HAND | | | - | GT | Y | 50 | LD | | EQ | 7 | ER | 0
 |
 | | |
 | |
 | | |
 | + | - | 1 | - | -
 | + | + |
| GOTO | | | | TR | INS | IN | | | | TR | AN | 5 11 | N
 |
 | | | GET
 | X |
 | | |
 | 51 | HIT | PM | NT | R
 | LEP | BE |
| | SET QTY ISO
MOYE UNIT PR
SET QTY ON I
SET QTY ON I | ABLE NAME
OPER OPERAND 1
SALES EXTENSION
INVENTORY TRANSACTIONS
ITEM NO: SALES
GTY SOLD
SET SALES GTY SHIP
SET GTY ISSUE
MOVE UNIT PRICE
* SALES GTY SHIP
SET GTY ON HAND
GOTO | TABLE NAME TABLE OPER OPERAND 1 OP SALES EXTENSION O INVENTORY TRANSACTIONS IS EN ITEM NO: SALES EQ ITEL GTY SOLD LE GTY SET SALES QTY SHIP EQ SET QTY ISSUE MOYE UNIT PRICE MOYE UNIT PRICE TO S * SALES QTY SHIP SET QTY ON HAND GOTO GOTO | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 SALES EXTENSION OPEN INVENTORY TRANSACTIONS IS END OF FILE ITEM NO: SALES EQ ITEM NO: INVEN GTY SOLD LE GTY ON HAND SET SALES QTY SHIP EQ SET QTY ISSUE MOYE UNIT PRICE MOYE UNIT PRICE TO SALES AMT * SALES QTY SHIP SET QTY ON HAND GOTO GOTO | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OI SALES EXTENSION OPEN ITEM NO: SALES IS END OF FILE N ITEM NO: SALES EQ ITEM NOINVEN Y GTY SOLD LE GTY ON HAND Y SET SALES GTY SHIP EQ SET QTY ISSUE + MOYE UNIT PRICE TO SALES AMT X SALES GTY SHIP SET QTY ON HAND SET QTY ON HAND - GOTO TR | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP SALES EXTENSION OPEN Image: Sales state s | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP SALES EXTENSION OPEN Image: Color of the colo | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OPER INVENTORY TRANSACTIONS IS END OF FILE N IS IS IS INVENTORY TRANSACTIONS IS END OF FILE N IS IS IS ITEM NO: SALES EQ. ITEM NO:INVEN Y IS IS GTY SOLD LE GTY ON HAND Y IS SET SALES QTY SHIP EQ QTY SO SET QTY ISSUE + QTY SO MOYE UNIT PRICE TO SALES AMT X * SALES QTY SHIP IN IN SET QTY ON HAND - QTY SO GOTO TR ANS IN | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OPERAND 2 SALES EXTENSION OPEN Improvement of ost | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 SALES EXTENSION OPEN et oil | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP Image: Strength of the strength | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP SALES EXTENSION OPEN Image: Contract of the contract | TABLE NAME TABLE TYPE OPER OPERAND 2 OPERAND 2 OPERAND 2 SALES EXTENSION OPEN Image: Constraint of the constraint | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP <td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP</td> <td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OPERAND 2 OP OPERAND 2 OPERAND 2</td> <td>TABLE NAME TABLE TYPE Image: Constraint of the state of the s</td> <td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP</td> <td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP <td< td=""><td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE Image: Transaction of the second secon</td><td>TABLE NAME TABLE TYPE OPERAND 2 OP BAND 2 N N D OP BAND 2 N D D<!--</td--><td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND2 OP OP OPERAND2 OP OP OPERAND2 OP <t< td=""><td>TABLE NAME TABLE TYPE PERCUENCY OPER OPERAND 1 OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERADIC OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE IPREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERIOD OPERAND 2 OP OP</td></t<><td>TABLE NAME TABLE TYPE PREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP OPERAND 2 OP OPE</td></td></td></td<></td> | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OPERAND 2 OP OPERAND 2 OPERAND 2 | TABLE NAME TABLE TYPE Image: Constraint of the state of the s | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP <td< td=""><td>TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE Image: Transaction of the second secon</td><td>TABLE NAME TABLE TYPE OPERAND 2 OP BAND 2 N N D OP BAND 2 N D D<!--</td--><td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND2 OP OP OPERAND2 OP OP OPERAND2 OP <t< td=""><td>TABLE NAME TABLE TYPE PERCUENCY OPER OPERAND 1 OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERADIC OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE IPREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERIOD OPERAND 2 OP OP</td></t<><td>TABLE NAME TABLE TYPE PREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP OPERAND 2 OP OPE</td></td></td></td<> | TABLE NAME TABLE TYPE OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP< | TABLE NAME TABLE TYPE Image: Transaction of the second secon | TABLE NAME TABLE TYPE OPERAND 2 OP BAND 2 N N D OP BAND 2 N D D </td <td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND 2 OP <</td> <td>TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND2 OP OP OPERAND2 OP OP OPERAND2 OP <t< td=""><td>TABLE NAME TABLE TYPE PERCUENCY OPER OPERAND 1 OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERADIC OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE IPREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERIOD OPERAND 2 OP OP</td></t<><td>TABLE NAME TABLE TYPE PREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP OPERAND 2 OP OPE</td></td> | TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND 2 OP < | TABLE NAME TABLE TYPE PERCUCENCY OPER OPERAND 1 OP OPERAND2 OP OP OPERAND2 OP OP OPERAND2 OP OP <t< td=""><td>TABLE NAME TABLE TYPE PERCUENCY OPER OPERAND 1 OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERADIC OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP <</td><td>TABLE NAME TABLE TYPE IPREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OP OPERAND 2 OP OP<</td><td>TABLE NAME TABLE TYPE PERIOD OPERAND 2 OP OP</td></t<> <td>TABLE NAME TABLE TYPE PREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP OPERAND 2 OP OPE</td> | TABLE NAME TABLE TYPE PERCUENCY OPER OPERAND 1 OP OPERAND 2 OP OP< | TABLE NAME TABLE TYPE PERADIC OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP < | TABLE NAME TABLE TYPE IPREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OP OPERAND 2 OP OP< | TABLE NAME TABLE TYPE PERIOD OPERAND 2 OP OP | TABLE NAME TABLE TYPE PREQUENCY OPER OPERAND 1 OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OPERAND 2 OP OP OPERAND 2 OP OPE |

Figure 3a. Decision Table, Example 3.

SYS INV	SYSTEM NAME			FILE	DE	SCR		ION				FRANK	эт 2/5	DATE 1/2/6	2	PAGE	OF	PAGES
FILE	NAME	DIF	2 T	YPE		ME	DIUM		P	REC	OR			ABB	AVERA	NO OF	RECOR	DS
INVE	NVENTORY TRANSACTION			_		SE	R-C	ARD	1	SAL	.E.	s		-	80	000	10	000
SER	FIELD NAME				TH	OCC	AVG	MAX	A5804	To	SECUENCE	RELATIVE	LOCATION	V	LUE/		KS	
1	1 ITEM NO 2 QTY SOLD			14				1										
24	3 SALES GTY SHIP			4				1	E	t						_	_	
5	5 SALES AMT			4.2	-	~		1	L			-						

Figure 3b.	File	Description,	Example	3.
------------	------	--------------	---------	----

SY	ENTORY, BILLING]	1	FILE	DE	IB	M	ION			[FRANC	IS IS	DATE //2/62		PAGE	OF	PAGES
FILE	NAME IENTORY MASTER	RE	TA	NED	>	ME	DIUM	1515	R	ECO	En	NAME		ABB	AVILAN 4,0	10 OF	PLAK	RDS
SER NO	FIELD NAME	MUMADEL	SIGN		TH	OCCI	AVG	MAX	ASSOC	ASSOC	ENGUENCI	RELATIVE	LOCATION	VAI	LUE/R	EMAR	RKS	
1232507000	ITEM NO UNIT PRICE OLD BAL ON HAND QTY RECEIPTS QTY ISSUES QTY CN HAND QTY ON ORDER QTY MINIMUM USEM DESCO	2222222	U++++++	7 3.2 5 5 5 5 5 5 5 5 5 5				1 1 1 1 1 1 1 1 1										

Figure 3c. File Description, Example 3.

QTY SOLD is less than or equal to the QTY ON HAND...

Each of these operands is a field of the records SALES and INVENTORY, respectively. However, they are unique within the system and need not be qualified. Since ITEM NO is a field name in both records and therefore lacks uniqueness, it has to be qualified by being linked to the name of the record where it is contained.

The decision table in figure 3 illustrates the use of the SET and MOVE operators in two operand arithmetic and in multi-operand arithmetic. Action row 6 for rules 1 and 2 is stated as follows:

rule 1: SET the Quantity of Issue equal to the Quantity of Issue plus the Quantity Sold...

rule 2: SET the Quantity of Issue equal to the Quantity of Issue plus the Quantity on Hand...

This is the SET operator with two operand arithmetic. Two operand arithmetic may be used for addition, subtraction, multiplication, and division. The results of these arithmetic operations for two fields, A and B, are summarized below:

OPERATOR	OPERAND 1	CONNECTOR	OPERAND 2	Result is
SET	А	+	В	A ← A + B
SET	А	-	В	A 🖛 A - B
SET	A	*	B	A -A * B
SET	А	1	В	A 🛶 A / B

TABLE	TABLE NAME	TAS	LE TYPE							Γ	Π		F	RI	Q	UE	N	CY	T	Т	T							T	T
SER	OPER OPERAND 1	OP	OPERAND2	OF	81	0	PE	RAN	D2	0	P	05	OP	EP	AN	02	0	P°	0	PER	AN	D	2	0	P°	10	PER	AND	2
		-		01 0	01 01	04	05	06	07 05	04	10	11 1	2 15	14	13	16	17	16	19 1	10 21	22	25	114	25	26	17	28 3	9 30	31 3
* 005	SALES EXTENSION	C	PEN																1										
1	INVENTORY TRANSACTIONS	IS EN	D OF FILE	N						¥							N							Y					
2	ITEM NO: SALES	EQ ITE	MNOWNYEN	Y						Y		_	1				N		_	-	1						_		
3	GTY SOLD	LE GT	Y ON HAND	Y	+				-	N		+	-	-					+	+	-					-	4	+	
4	SET SALES OTY SHIP	EQ			GT	Y	30	LD	-			GTY	UN	1+	AN	D			1	+	1			H	+		+	+	H
5	SET GTY BACK ORDER									EC		OT Y	50	LC							1								
										-	1	QT Y	0	1	AN	P													T
6	SET QTY ISSUE	1		+	an	Y	50	D		+		QTY	01	1	An	D													
7	MOVE UNIT PRICE	TOS	ALES AMT	X	+	-	-		+	×		-	-	F	F	-			+	Ŧ	F	-		Η	-	-	Ŧ	Ŧ	H
8	SET QTY ON HAND			-	GT	Y	50	LD		EG	2	ZER	0															T	T
9	GOTO			TR	NS	IN				TR	AN	51	N				GE	TX	_		-			SH	IP	MED	NT	REI	EAS
-		-					-	-	-	-				-	-			-	-	_	-			-	-	-	+	-	

Figure 3a. Decision Table, Example 3.

SYS INV	SYSTEM NAME			FILE	DE	SCR		ION				FRANK	5T	DATE 1/2/6	2	PAGE	OF	PAGES
FILE	NAME	DIS	1T	YPE		ME	DIUM		R	RECO	OR			ABB	AVERA	NO OF	RECOR	DS
INVE	NTORY TRANSACTIONS	IN	_	_	_	SE	R-C/	ARD	1	SAL	Ĕ.	s		-	80	000	10	000
SER NO	FIELD NAME				TH	OCCU	AVG	MAX	A5800	ASSOC TO	SEOVENCE	RELATIVE	LOCATION	V	ALUE/		KS	10.00
1 2 3	1 ITEM NO 2 QTY SOLD 3 SALES GTY SHIP			7 4				1										-
4 5	3 SALES GTY SHIP 4 QTY BACK ORDER 5 SALES AMT		+	4 4.2				1 1										
		L	-	-	L	~		~			F	-				-	-	

Figure 3b. File Description, Example 3.

SYS	ENTORY, BILLING]		FILE	DE	IB	M	ION				FRANC	т 15	DATE //2/62		PAGE	OF	PAGES
FILE	NAME	DIR	T	YPE		ME	DIUM		R	ECO	R	NAME		ASS	AVERA	NO OF	RECO	DS
INV	ENTORY MASTER	ĥE	TA	INED	2	RA	N-D	ISK	Ĩ	NV	Er	VTORY		INVEN	4,0	00	5,0	000
SER	FIELD NAME	No.	1	LEN	тн	occi	URRE	NCES	A	-1	1000	RELATIVE	LOCATION	1/01	115 /0	ELAN	NC	
NO	PIELO NAME	1	x	MAX	VAR	MIN	AVG	MAX	305	200	ENCE	BEGIN	END	VAL	LUE/R	EMAR	(KS	
1	ITEM NO	N	U	7				1			-							
2	UNIT PRICE	ITEM NO NU						1										
3	OLD BAL ON HAND	N	+	5				1										
4	QTY RECEIPTS	N	+	5				I									_	
5	GTY ISSUES	N	+	5				1										
6	GTY CN HAND			5				1										
7	QTY ON ORDER	N	+	5				1										
8	QTY MINIMUM	N	+	5				1										
9	ITEM DESCR	A		20	~			1	-	-			_					

Figure 3c. File Description, Example 3.

A is always the receiving field. B is added to, subtracted from, multiplied by, or divided into A before the result is placed in A. A similar display could be constructed for MOVE:

OPERATOR	OPERAND 1	CONNECTOR	OPERAND 2	Result is
MOVE	А	+	В	B + A → B
MOVE	А	-	В	B - A → B
MOVE	А	*	В	B * A→B
MOVE	А	1	В	B∕A→B

B is always the receiving field. A is added to, subtracted from, multiplied by or divided into B before the result is placed in B. Two operand arithmetic may be used in either limited entry or extended entry form.

Action row 7 of rules 1 and 2 illustrates the MOVE operator in multioperand arithmetic. That part of rule 1 reads:

... MOVE the UNIT PRICE times the SALES QuanTitY SHIP to SALES AMounT... The results of multi-operand arithmetic for three fields, A, B, and C, are summarized below.

	TABLE	TABLE	NAME	TAS	LE TYPE							Г			1	FR	EG	UE	N	CY	- 1		T	Г					Т	T		
	SUR	OPER	OPERAND 1	OP	OPERAND 2	OF	81	0	PER	NAS	D2	C	P	08	0	PE	RAI	NDZ	0	P°	0	PER	AN	D 2	2	0	P°	OF	ER	ANC	2 2	_
						01	01 07	04	08	06	07 08	04	10	11	12	5 1	4 11	16	17	16	19	20 3	1 23	1 23	24	25	2.6	17 7	18 1	0 80	ग	32
*	005	SALES EXTE	ENSION	0	PEN																1											
	1	INVENTORY TR	ANSACTIONS	IS EN	D OF FILE	N						N							N							Y						
	2	ITEM NO:	SALES	EQITE	MNOWNVEN	Y						Y							N				1									
	3	GTY SOLD		LE QTY	ON HAND	Y	-				-	N		-	-	+	+	+		_	+	+	-	-			-	4	+	-		_
	4	SET SALES	OTY SHIP	EQ			QT	Y	30	LD	+			ar	10	N	HA	ND		-		+	1	+				+	+	+		
- 1	5	SET OTY E	ACKORDER									EG		ath	1 3	01	D											T	T	T		
							-				-	-		ar v	1 0	N	HA	ND														
	6	SET QTY	ISSUE			+	ar	Y	50	D		+		QT	10	N	HA	ND												T		
	7	MOVE UNIT	PRICE	TO S	ALES AMT	X						X																				
		* SALES Q	TY SHIP																													
	8	SET QTY O	N HAND			-	GT	Y	50	LD		EQ		ZE	05																	
	9	GOTO				TR	NS	IN				TR	AN	SI	N	-		-	GE	IX	_					SH	PI	MER	T	REI	EA	SE
	-									-	-	-				_	-	-	-	-	-		-			-	-	+	+	-		

Figure 3a. Decision Table, Example 3.

SYN INV	SYSTEM NAME			FILE	DE	SCF		ION				FRANK	5T	DATE 1/2/6	2	PAGE	OF	PAGES
FILE	NAME	DIF	R T	YPE		ME	DIUM		P	REC	OR	NAME		ABB	AVERA	NO OF	PEAK	DS
INVE	NVENTORY TRANSACTICA					SE	R-C	ARD	1	SAL	.E.	s		-	80	00	10	000
SER NO	FIELD NAME				TH	OCC	AVG	MAX	ASSOC	To	12MINORS	RELATIVE	LOCATION	V	LUE/R		KS	1001000
1 2 3	1 ITEM NO 2 QTY SOLD 3 SALES GTY SHIP			7				1										
4 5	3 SALES GTY SHIP 4 QTY BACK ORDER 5 SALES AMT			4				1	L		E							
-	5 SALES AMT			I.F.	-	~		-					-			-	~	

Figure 3b. File Description, Examp	le	3	
------------------------------------	----	---	--

SYS	ENTORY, BILLING]	1	FILE	DE	IB	M	ION				FRANC	в т .15	DATE 1/2/62		PAGE	OF	PAGES
FILE	NAME IENTORY MASTER	DIA	TA	THE		ME	DIUM	ISK	R	ECO	F	NAME		ABS	AVILAN	NO OF	PEAK	abs
SER	FIELD NAME	Manager.	5.61	LEN	TH	OCCI	AVG	MAX	A5 805	ALL OT	SECUENCE	RELATIVE	LOCATION	VAI	LUE/R	EMAR	RKS	200
123350700	ITEM NO UNIT PRICE OLD BAL ON HAND QTY RECEIPTS QTY ISSUES QTY CN HAND QTY ON ORDER QTY MINIMUM	2222222	U+ + + + + + +	7 3.2 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5														

Figure 3c. File Description, Example 3.

2.	20	

2	OPERATOR	OPERAND 1	CONNECTOR	OPERAND 2	Result is
	MOVE	А	TO	С	A+B→C
		+ B			
	MOVE	А	TO	C	A - B→C
		- B			
	MOVE	А	TO	С	A * B→C
		* B			
	MOVE	А	TO	С	A∕B→C
		/ B			

OPERATOR	OPERAND 1	CONNECTOR	OPERAND 2	Result is
SET	А	EQ	В	A ← B + C
		+	С	
SET	А	EQ	В	A ←B - C
		-	С	
SET	А	EQ	В	A←B * C
		*	С	
SET	А	EQ	В	A←B/C
		1	С	

TABLE	TABLE N	ME	TAS	LE TYPE		- 81					T	T			FR	R	ရပ်	EN	ve	Ý			1					Π	Ι	T	Γ
SER	OPER O	PERAND 1	OP	OPERAND2	0	P	0	PE	RAN	D 2		OP	-		OPE	IR/	ND	2	OP	~	OPE	RA	ND	2	C	P'	0	PER	NAS	02	
-					01	01 01	04	08	96	07 ol	0	10	11	12	13	14	15 1	6 1	7 10	19	20	21 2	13 3	3 24	25	14	27	28 7	93	0 31	32
* 005	SALES EXTEN	SION	0	PEN						_										1					1					1	
1	INVENTORY TRAI	SACTIONS	13 EN	D OF FILE	N					_	4							N	1						Y						
2	ITEM NO: :	SALES	EQ ITE!	MNOWNVEN	Y	_					Y	1		_				2	1												
3	QTY SOLD		LE QTY	ON HAND	Y	+				-	N	1				-	+	+	+	+			+	+	+				+	+	-
4	SET SALES	OTY SHIP	EQ			QT	Y	30	LD		+	-	gr	N	ON	H	WD	,	1	1	-		+	+	+			H	+	+	t
5	SET GTY BA	CKORDER									E	0	GT	Y	50	LD		T													T
											-	- 1	QT.	Y	ON	H	AND											Π			
6	SET QTY IS	SUE			+	an	Y	50	D		4	-	QT	Y	ON	H	AND							Τ				T			T
7	MOVE UNIT P	RICE	TO S	ALES AMT	X						>	$\langle $						T							Γ						Γ
	* SALES OTY	SHIP																													Г
8	SET QTY ON	HAND			-	GI	Y	50	LD		E	Q	ZE	RO											T						Γ
9	GOTO				TR	ANS	IN				T	RAN	B	IN				Ģ	ED	1					SH	IP	ME	NT	RE	LEP	BE
										_	-					_		-	-	-						-	-	-	-	1	

Figure 3a. Decision Table, Example 3.

SY	ENTORY, BILLING]		FILE	DE	SCR	M	ION			1	FRANK	5T	DATE 1/2/6;	٤	PAGE	OF	PAGES
FILE	NAME	DI	2.1	YPE		ME	DIUM		P	ECO	OR	D NAME		ABS	AVERA	NO OF	RECOR	DS
INVE	NTORY TRANSACTIONS	IIN				SE	R-C	ARD	1	SAL	.E.	s		-	80	00	10	000
SER NO	FIELD NAME	Manufact.	S.av		TH	OCCI	AVG	MAX	A580C	ASSOC TO	STONENCE	RELATIVE	LOCATION	VA	LUE/R		RS	
1 2 3	ITEM NO QTY SOLD	NZ	11	7				1										
4 5	OTY BACK ORDER	NN	+	4	-			1										
		L	F	-	-	-		~			-	-				-	~	

Figure 3b. File Description, Example 3.

SYS	TEM NAME	1				IB	M				1	ANALYS	т	DATE]		
TIAN	ENTORY, BILLING	1		FILE	DE	SCR	IPT	ION				FRANC	15	12/62		PAGE	OF	PAGES
FILE	NAME	DIR	1.T	VPE	-	ME	DIUM	l	R	ECO	R	D NAME		ASS	AVERAG	10 OF	RECO	RDS
INV	ENTORY MASTER	RE	TA	INED	2	RA	N-D	ISK	Ï	NV	E	NTORY		INVEN	4,0	00	5,0	000
SER	FIELD NAME	MUNICIPAL AND	Sida	LEN	TH	OCCI	AVG	MAX	A5800	Yo	SASASHC	RELATIVE	LOCATION	VAI	LUE/R	EMAR	RKS	
1	ITEM NO	N	1U	7	-	-		I	1		-				-			
2	UNIT PRICE	N	+	3.2	1			1										
3	OLD BAL ON HAND	N	+	5				11	1									
4	QTY RECEIPTS	N	+	5				11										
5	GTY ISSUES	N	+	5				11										
6	QTY ON HAND	N	+	5				1										
7	QTY ON ORDER	N	+	5				1										
8	QTY MINIMUM	IN	+	5				11										
9	ITEM DESCR	A	L	20	~			1									311	

Figure 3c. File Description, Example 3.

More than three operands can be used in multi-operand arithmetic.

MOVE	А	ТО	D	means	A*B*C→D
	*B				
	* C				
SET	А	EQ	В		$A \leftarrow B + C + D + E$
		+	С		
		+	D		
		+	Е		

Two division connectives cannot be used in a single action, although addition, subtraction and multiplication connectives can be used more than once.

Action row 8 in rule 2 illustrates the use of a <u>connotative term</u>. This is an operand whose value is directly implied by the meaning of its symbols. This action of rule 2 is read:

... SET the QUANTITY ON HAND to ZERO...

Here, ZERO is a <u>connotative term</u> which represents a string of zeros. The number of zeros is determined by the size of the field which is to be set to zero. In this case QTY ON HAND is a five position field, so five zeros are moved to QTY ON HAND. Other connotative terms are: BLANK, NINE, ALL 'X'. In the last term, X represents any symbol or combination of symbols. For example, ALL '7' would produce 7777...., ALL 'ABC' would produce ABCABCABC...

TANLI	TABLE NAME	TABLE TYPE								Τ	T	T		F	R	Q	ÚΕ	N	CY	-		Τ			Т	Т	Τ		Т	T	T	Г
SER	OPER OPERAND	OP OPERAND	2	OP	-84	0	PE	RAP	4D 2	T	OP	08		OP	EP	AN	02	0	P	" 0	PE	RA	NC	2	T	0	Por	OF	PER	AN	0 2	
-				4 01	07	04	04	06	07 0	1 0	9 10	111	112	15	14	15	16	17	18	19	20	21	22	35 3	4 3	15 1	24 3	7 7	28 2	9 3	0 31	31
* 005	SALES EXTENSION	OPEN															1								•							Г
1	INVENTORY TRANSACTIONS	S ISEND OF FILE	1	1						1								N							N	Y						
2	ITEM NO: SALES	EQ ITEM NOUNVEN	N	1						1	1							N														
3	GTY SOLD	LE GTY ON HAND	1	-						1	1		-						_	-	-	-	-	-	+	+	4	-	+	+		
4	SET SALES OTY SHI	PEQ		+	QT	Y	30	LD		1	1	QT	Y	ON	H	AN	D				-	1	+	+	+	+	+	+	+	+	+	t
5	SET GTY BACK ORDER	2								12	Q	QT	r	50	LD										T	T		T	T		T	
										-	-	Ø	Y	ON	H	AN	Þ								T				T		T	
6	SET OTY ISSUE		-	-	ar	Y	50	LD		H	F	QT	Y	ON	H	AN	D						Т		T	T	T	T	T	T	T	
7	MOVE UNIT PRICE	TO SALES AMT	Þ	<	-					P		F	-	-			F		_			-	-	-	Ŧ	-	-	+	+	+	T	F
-	* SALES OTY SHIP			-	-	-	-	-		+	-	-		+	-	-	-		_	-	-	+	-	+	+	-	_	_	-	+	-	_
8	SET QTY ON HAND		-	-	GI	Y	50	LD		E	Q	ZE	RC		1		-		_	-	_	_	-	-	-	-	-	-				
9	GOTO		T	RAN	S	N	-		-	T	RAN	B	10	1	-	-	-	6E	IX	-	-	-	-	+	Þ	HI	PM	E	TP	RE	LEI	6E
-					-	-	-	-		-	-	1		-	-	-	-	-	-	-	-	-	_	-	-		+	+	-	+	-	1

Figure 3a. Decision Table, Example 3.

SYS	ENTORY, BILLING]		FILE	DE	SCR		ION				FRANK	5T 215	DATE	2/6	2	PAGE	OF	PAGES
FILE	NAME	DIF	R.T	YPE		ME	DIUM		R	ECO	OR	NAME		AB	8	AVER	NO OF	RECOR	IDS .
INVE	NTORY TRANSACTIONS	NL			_	SE	R-C	ARD	1	SAL	.E.	s			-	8	000	10	000
SER NO	FIELD NAME	MUMANA	BIGN		TH	OCCI	AVG	MAX	A880C	TO	SROVENCE	RELATIVE	LOCATION		VA	LUE/	REMAR	RKS	
1	ITEM NO QTY SOLD	22	11	1 4				1											
3	OTY BACK ORDER	N	++	4				1	-	-	-								
5	SALES AMT	N	V	4.2	-	~		1			E				~		_		

Figure 3D. File Description, Exa	nple	3.
----------------------------------	------	----

SYS	ENTORY, BILLING]		FILE	DE	SCR	M	ION				FRANC	IS	DATE		PAGE	OF	PAGES
FILE	NAME	DIR	T	YPE		ME	DIUM	1	R	ECO	R	D NAME		ASS	AVERAC	10 OF	RECO	RDS
INV	ENTORY MASTER	RE	TA	INED	2	RA	N-D	ISK	Ĩ	NV	E.	NTORY		INVEN	4,0	00	5,0	200
SER	FIELD NAME	MURAPHY.	SIGN	LEN	TH	OCCI	AVG	MAX	A5 504	A480C	ERCORNET	RELATIVE	LOCATION	VA	LUE/R	EMAR	RKS	
1	ITEM NO	N	U	7	1			I	1	F								
2	UNIT PRICE	N	+	3.2				1										
3	OLD BAL ON HAND	N	+	5				1										
4	QTY RECEIPTS	N	+	5				I										
5	GTY ISSUES	N	+	5				11										
6	QTY ON HAND	N	+	5				1										
7	QTY ON ORDER	N	+	5				1										
8	QTY MINIMUM	N	+	5				1										
9	ITEM DESCR	A		20	1			1	-	-						-		

Figure 3c. File Description, Example 3.

QUESTIONS

1.	Read the	3rd and	4th r	ules.			
2.	What are	the rela	ations	ships a	among files,	records,	and fields?
3.	How may	a field	be un	iquely	qualified?		
4.	What is t	he mean	ing o	f the f	following?		
	a)	MOVE	А	TO	D		
			+ B				
			+ C				
	b)	SET	А	EQ	В		
				+	С		

5. What is the meaning of the statement:

SET A EQ NINES

TANLE	TABLE NAME	TABLE TYPE	1	T							T			1	FR		ąį	E	N	27	r					1		Ι	T		Γ	T	Γ	Γ
SER	OPER OPERAND	OP OPERAND		OP	01	0	PE	RAI	ND 2		0	•	DE	0	PE	RA	NC	2	0	P	" (OP	ER	AN	D	2	10	SP	•*	DP	ER/	ANC	20	
			0	01	07	04	04	04	07 0	08	09 1	01	11	2	5 1	4	5	16	17	18	19	24	21	27	31	124	25	124	127	11	119	10	111	32
* 005	SALES EXTENSION	OPEN									_						1									1.								
1	INVENTORY TRANSACTIONS	S IS END OF FILE	N								4								N								Y							
2	ITEM NO: SALES	EQ ITEM NOUNVEN	Y	1							Y					1	-		N	_		L												
3	QTY SOLD	LE GTY ON HAND	Y	-						_	N	+	+	-	+	+	-		_	_				-	1	-	-	+	+	+	-	+	╞	
4	SET SALES OTY SHI	PEQ	t	1	QT	Y	30	LD				6	πh	t	N	H	N	2	-			t		t	t			t	t	+	+	+	+	t
5	SET GTY BACK ORDER	2	T	T							EQ	6	πh	1	01	D							Γ								T	T		
					-						-	(U	0	N	H	N	>										L	1		T			
6	SET QTY ISSUE		+	1	ar	Y	50	LD			+	G	T	C	N	H	AN													T	Г	T		
7	MOVE UNIT PRICE	TO SALES AMT	X	-		-	-	-		-	×	+	+	+	+	+			-	_	-	-	F	F	-		F	F	F	F	F	F	F	F
8	SET GTY ON HAND		1-	-	GT	Y	50	LD			EQ	7	ER	0														T	T		T	T		
9	GOTO		T	AN	S	IN				_	TR	INC	5 1	N	+	+	-	_	GE	IX				-			5H	IP	M	EN'	TF	(LI	EP	5E
					-	-	-	-		_	_	1			-	-	_	-	-	-	-			1			-	+	+	+	+	+-	-	

Figure 3a. Decision Table, Example 3.

SYS INV	ENTORY, BILLING]		FILE	DE	IB		ION				FRANK	ST LIS	DATE 1/2/6	2	PAGE	OF	PAGES
FILE	NAME	DIF	R T	YPE		ME	DIUM		10	REC	OR	NAME		ABB	AVERA	NO OF	RECOR	DS
INVE	NTORY TRANSACTIONS	NL				SE	R-C	ARD		SAL	E.	s		-	80	000	10	000
SER NO	FIELD NAME	and and age of	BIGN		TH	OCC	AVG	MAX	A\$504	TO	15wandes	RELATIVE	LOCATION	VA	LUE/		KS	
1	ATY SOLD	NZ	11+	7				1								1		
3 4	OTY BACK ORDER	N	* +	4				1		+								
5	SALES AMT	N	V	4.2		~		1				-						

Figure 3b. File Description, Example	3.
--------------------------------------	----

SYS	ETEM NAME]		FILE	DE	IB	M	ION			1	FRANC	ST IS	DATE //2/62		PAGE	OF	PAGES
FILE	NAME	Dis	T	YPE		ME	DIUM	1	R	ECO	DR	NAME		A58	AVERA	10 05	RECOS	DS
INV	ENTORY MASTER	RE	TA	INED	2	RA	N-D	ISK	Ĩ	NV	E. (VTORY		INVEN	4,0	00	5,0	000
SER	FIELD NAME	MURADOV.	BIGN	LEN	TH	OCCI	AVG	MAX	A\$ 904	ASSOC.	CECULARIA CE	RELATIVE	LOCATION	VAI	LUE/R	EMAR	RKS	
1	ITEM NO	N	U	17	1			I	F	F	-							
2	UNIT PRICE	N	+	3.2				1	1									
3	OLD BAL ON HAND	N	+	5			1	1										
4	QTY RECEIPTS	N	+	5				I										
5	GTY ISSUES	N	+	5				11										
6	QTY ON HAND	N	+	5				1										
7	QTY ON ORDER	N	+	5				1									-	
8	QTY MINIMUM	N	+	5				11							_		_	
9	ITEM DESCR	A	L	20	~			1	-									-

Figure 3c. File Description, Example 3.

ANSWERS

- Rule 3: <u>If</u> it is not the end of the transaction file, and the item number of the sales record is not the same as the item number of the inventory record, <u>then</u> go to table GETX.
 Rule 4: If there are no more records in the transaction file, <u>then</u> go to the table named SHIPMENT RELEASE.
- A file is a collection of one or more records.
 A record is a collection of one or more fields.
- By specifying the field name, placing a colon after it, and then specifying the particular record name.
- a) This is multi-operand arithmetic with the MOVE operator. The final result is that

D will contain A + B + C

b) This is multi-operand arithmetic with the SET operator.The final result is that

A will be replaced by B + C

 The field A is to be replaced with the value 9999....; the number of nines will be determined by the size of A.

	10	TABLE NAME	TABLE TYPE	Т	T									1	FFL	RQ	UE	N	CY	•					Ι		1				1
	SER	OPER OPERAND	1 OP OPERAND		OP	34	OP	TERA	ND 1	2	0	Þ '	35	0	PER	LAN	02	C	P	" (PE	RA	M	2		OP	04	7676	RA	10 7	4
	-				4 0	12 00	04	04 04	07	00	09	10	11 1	1	8 14	15	16	17	18	19	30	21	22	48 2	4 2	18 14	£7	28	1.9	80 1	118
*	0.37	DEDUCTION VALU	D OPEN	6	da	SE									1	1_															
	1	TYPE DED CODE	EQ DED CODELL	TY	IY	1										1													-		
	2	TYPE DED CODE	EQ 13	N	YL	dL.													6									1			E
			22-24										1															1			
	3	BAL AMT	GE DED AM	T	y	4				_			-	+	-			_				_		_	T	+	1	-		-	
	4	GD TO	PAY BOLL CAL	Ŋ	1	(Ħ					1	1	1	1	t						-		1	1	\pm	t			+	t
	5	GO TO	ERROR	-	1	X				_		-	-	+	1				_			_								1	1
	-	the second		-		-	_	-	-	-						-	^	-	-	-	-	-	-	-	Т	-	-	-	-	-	in the second

Figure 4a.	Decision	Table,	Example	4.
------------	----------	--------	---------	----

SYS	STEM NAME		IBM	ANALYST	DATE		
PA	YROLL	REFERENCE	PERAND DESCRIPTION	HOLT	1/3/62	PAGE	OF PAGES
965A 140	NAME	TYPE		VAL	UE		
1	DED CODE LIST	VLIST	3, 7, 13, 18, 22 -	26			

Figure 4b. Reference Operand Description, Example 4.

Example 4 covers the part of a payroll system which checks to see if a certain type of deduction is valid.

The decision table figure 4a is written entirely in limited entry form. In preparing decision tables for programmers the systems engineer should make an estimate of the frequency that a rule will be executed. For this example, the analyst has estimated the conditions for rule 1 will be satisfied 60% of the time and 35% of the time for rule 2. This information is recorded at the top of the column of entries for each rule in either the extended entry or the limited entry table.

In the first 3 examples of this chapter, it is assumed that the several rules exhaust all possible combinations of conditions which can ever occur in these tables. Therefore, failure to satisfy at least one set of conditions indicates that there is an error in logic or in data. However, occasions will arise when the table is not intended to explicitly describe all possible combinations of conditions, or where it is impractical or impossible to do so. In this event, some method must be provided to recognize unaccounted for conditions without inferring that an error in logic or data exists. This is covered by the special unconditional rule ELSE (E), which provides for action control when none of the other rules can be executed. In example 4, the ELSE rule will occur about 5% of the time (100 - 60 - 35 = 5). This rule will execute action row 5 which provides for an error table to be entered.

2.24

TABLE	TAD	E NAME	TAI	LE TYPE		Π			T					1		学校	E Q	UE	N	CY								1			1
SER	OPER	OPERAND 1	OP	OPERAND2	0	P		OP	ERA	ND	2	0	P	37	C	NUE	ELA.	102	0	Pai	0	PER	AN	D	2	0	¢ 34	OP	ERA	ND	2
	and the second				el	93	03	04 0	4 86	67	08	09	10	11	12	18 1	6 15	116	17	18	0 1	10 1	1 2.2	125	24	2.5	5.41 2	7 2	11.9	30	31]
037	DEDUC	TION VALID		OPEN	60	35	E									1	1												1.		
1	TYPE	DED CODE	EQI	ED CODELIST	Y	Y												1													1
2	TYPE	DED CODE	EQ	13	N	Y			_																			1			
				22-24												1												1			
3	BAL	AMT	GE	DED AMT		Y									-	-															
4	GD TO		PAY	BOLL CALC	X	X	-	-	+				+	+	+	+	+			+	+	+	+	+		\vdash	+	+	+	\mathbb{H}	+
5	GO TI	0	ER	ROR			X							-					_		1	1		L			1				
-					-			_	-	-	-		-		-		+	1		-+	÷	+-	+	-	-	1	-	-	+	-	

Figure 4a.	Decision	Table,	Example	4.
------------	----------	--------	---------	----

EVE	TEM NAME		IBM	ANALYST	DATE		
PA	ROLL	REFERENCE C	PERAND DESCRIPTION	HOLT	1/3/62	PAGE	OF PAGE
96A 140	NAME	TYPE		VAL	UE		
	DED CODE LIST	VLIST	3, 7, 13, 18, 22 -	26			

Figure 4b. Reference Operand Description, Example 4.

On the limited entry table, the ELSE rule is indicated by the placement of an E in the column where the frequency of the rule is usually written; in a mixed or extended entry table, the word ELSE is written in the appropriate place for the rule to which it is assigned. The ELSE rule is always considered last when "examining" a table. There can, of course, be no condition entries in an ELSE rule.

Rule 1 of this example is read as follows:

If the type of deduction code is on the deduction code list,

and

the type of deduction code is not 13, 22, 23, or 24,

then proceed to payroll calculation table.

With this rule, two kinds of value list operands are displayed simultaneously. The first DED CODE LIST, is a <u>value list name</u> which represents a specific collection of one or more values or ranges of values. It is not a field, nor does it appear on a record or file. It may only be used in a conditional statement and must be defined on a Reference Operand Description form (figure 4b) before being used.

The Reference Operand Description form is used for the description of constants, expressions, reference lists, and value lists. In figure 4b, the system name, which is the same as the system name for the decision table, is placed in the upper left-hand corner. Each listing is then assigned an individual number. For this example, there is only one

1	18-1	TAD	E NAME	TA	BLE TYPE					T	Π			T		14	REC	NUC	N	CY	·		T							T	1
	RR	APER	OPERAND 1	OP	OPERAND 2	0	P	14	OP	SRA.	ND 2		OP	3	a.	OP	TRA	NDZ	0	P	0	PER	AN	D 2		09	04	3PE	RAN	ID 2	
E	-					-	03	03	04 0	4 06	07	80	09 10	0 1	1 112	15	14 1	5 16	17	18	10 3	10 1	1 2 2	1 49	3.4	3.5 5	* 27	128	193	to m	11
*0	37	DEDU	TION VALID		OPEN	60	35	E			11																1			1	
	1	TYPE	DED CODE	EQI	DED CODELIST	Y	Y											1									1				1
	2	TYPE	DED CODE	EQ	13	N	Y																1					1			T
E					22-24																							1		T	
E	3	BAL	AMT	GE	DED AMT		У					_																	1	T	L
ŀ	4	GD TO	1	PAY	BOLL CALC	X	X		1	+	+	-		t	\pm		+	+	+		+	+	+		-	-	+		+	+	+
	5	GO T	0	ER	ROR			X							1																Γ
	-	-				-	-	-	_	-		-					-+-	-	-	-	-	+	+	-	-		-	1-	-		·h-

Figure 4a.	Decision	Table,	Example	4.
------------	----------	--------	---------	----

EVE	TEM NAME		IBM	ANALYST	DATE				
PA	ROLL	REFERENCE C	PERAND DESCRIPTION	HOLT	1/3/62	PAGE	OF	PAGES	
865A 240	NAME	TYPE	1	VAL	UE				
L	DED CODE LIST	D CODE HIST V HIST 3,7, 13, 18, 22-26							

Figure 4b. Reference Operand Description, Example 4.

listing, the DED CODE LIST. The type entry is V LIST, and the values are as shown. Each value is either a single entry value (e.g., 3, 7, etc.) or a range of values for which the top and bottom limits are stated (e.g., 22-26). All individual values or ranges of values are separated by a comma which denotes an exclusive <u>or</u> associated with the value list. In other words, the value is 3, <u>or</u> 7, <u>or</u> ... <u>or</u> 22-26. Consequently, to satisfy the first condition of rule 1, the code number must be one of the values, or within the range of values given in the list.

The second kind of value list operand is demonstrated in the second condition of rule 1:

TYPE DED CODE EQ 13 22-24

The operand <u>value list (13, 22-24)</u> reveals that a list of literal values can be used directly as the operand in a condition statement without specifying the list in a Reference Operand Description form. A range (22-24) is permitted, just as with the value list name.

If TYPE DED CODE is 13 in example 4, the condition of row 2, rule 1 is not satisfied (N in the entry) and rule 2 is satisfied (a Y in the entry).

2.26

1.0	TABLE	BMAN	TA	BLE TYPE					ГТ		T				FR	EQ.	UE	N	Y	T							T		
STR	CPER	OPERAND 1	OP	OPERAND 2	0	P	H.	OPS	RAN	ED 2		OP.	33		OFE	FLAN	102	0	pav	OP	ERA	ND	2	C	\$P°	406	10.0	ND	2
and the second					61	93	63	04 04	86	67 0	6 0	01 0	111	11	18 1	4 1B	16	17	10 11	20	21	32	28 24	25	5.41	27 3	0129	80	M] !
0,37	DEDUCTI	ON VALID		OPEN	60	35	E									1													
1	TYPE DE	D CODE	EQI	DED CODELIST	Y	Y		T			Ι													1		-			1
2	TYPE DE	D CODE	EQ	13	N	Y																				1			
				22-24							1							-											
3	BAL A	MT	GE	DED AMT		Y													T	L			-						
4	GD TO		PAY	BOLL CALC	X	X	1	+		+	+	+	+		-	+	+		+	-	-	$\left \right $	+	+		+	+		
5	GO TO		ER	ROR			X						1	L															
-	Carde Colorest Colorest Colorest				-	-+		_		-	-+-					+-	-		-		-		+	-		-	+	-	

Figure 4a.	Decision	Table,	Example	4.
------------	----------	--------	---------	----

EYS	TEM NAME	1	IBM	ANALYST	DATE			
PA	KROLL	REFERENCE O	PERAND DESCRIPTION	HOLT	1/3/62	PAGE	OF	PAGES
140 045	NAME	TYPE		VAL	UE			
1	DED CODE LIST	VLIST	3, 7, 13, 18, 22 -	26				

Figure 4b. Reference Operand Description, Example 4.

QUESTIONS

- 1. How are a list of values indicated in TDL?
- 2. What do the rules of this table indicate as far as frequency of execution is concerned?
- 3. What is the effect of the ELSE rule?
- If the logic required that a code type not be in the range
 72 103, how could this be handled?
- 5. If a value list has a range of 0 53, how may a code type be investigated to ascertain that it is not in the range of 12 - 26, but that it is on the list?

2.0	TAD	E NAME	TA	LE TYPE							1	1			FR	RQ	UE	NC	Y				T			1			1
STR	OPER	OPERAND 1	OP	OPERAND 2	0	P	84	OPE	RAN	102	1	QP	3L	4	RE	RAN	02	0	998	OP	RA	ND	2	0	¢°	OP	ERA	ND 2	1
					10	03	03	04 06	86	e7 of	0	10	11	12	18 14	13	16	17 1	0 11	30	21	\$2 8	3 24	25	5.6 1	17 21	129	30 N	1]3
037	DEDUC	TION VALID		OPEN	60	35	E									1													
1	TYPE	DED CODE	EQI	ED CODELIST	Y	Y				and a		T							T										L
2	TYPE	DED CODE	EQ	13	N	Y																				1			
				22-24							1				1														T
3	BAL	AMT	GE	DED AMT		Y				-	I				_				T										-
4	GO TO		PAY	BOLL CALC	X	X		1		-	t	t			1	1	Η	1	+	-		+	+	+	+	+			+
5	GOT	0	ER	ROR			X					1					\Box												T
10				the second day of the second day is a	-	-		_			-	-				+			-		-		-	-	-		-		-

Figure 4a.	Decision	Table,	Example 4	
------------	----------	--------	-----------	--

PAGE	OF PAGE
	IFA@5

Figure 4b. Reference Operand Description, Example 4.

2

ANSWERS

- By giving the list of values a name, and describing them as a list on a Reference Operand Description form, or by introducing a list of literal values directly in the table.
- 2. The analyst who prepared this decisive table has estimated that rule 1 will be executed almost twice as often as rule 2, but the two rules still do not cover all the possibilities that may occur. In 5% of the cases, rule 3 will be executed.
- The conditions for this rule are automatically satisfied and the actions executed. The ELSE rule is considered only when the conditions of none of the other rules are satisfied.
- 4. A range literal might be used, as there is no obvious requirement that the code types need be part of a defined list. Therefore, a statement as:

TYPE CODE NE 72-103 satisfies the requirement.

5. The reference operand form list has the value list described on it. Therefore, the conditions may be stated in two steps:

1) Is the type code on the value list?

2) Is the type code in the range 0 - 11?

27 - 53?

OR

1) Is the type code on the value list?

2) Is the type code not in the range 12 - 26?

	141	TABLE NAME	TABLE TYPE			17								•		=		Q	5	N	57	1								T		Γ		
	的政策	OPER OPERAND 1	OP OPERANDS	0	P		0	-		ND	2	•	-	- 02		OP	ER.	AN	DZ	0	P		PE	RA	NI	2		0	P **	OP	ER	NNC	2	
					05	ð1	64	04	94	en	66	61	18	41	12	15	14	19	16	17	16	14	10	21	11	23	24	10	1.0 1	7 3	1 3	130	N	32
淋	009	EXTEND DETAIL	OPEN	10	50	20	_								_								_					_	_	-				
	1	CUST NO: CUST HED NG	EQ CUST NO: WORKA	Y	Y	N									-																			
1	21	DISCOUNT (TTEM NO:CUST	15 PRESENT	Y	N	•																												
		TRAN, QUAN SOID: CUST																										I						
		TRAN)																									Т	Т	Т	Т	Т	Г		
				F	Ħ	-	-	=		=		F	-	=	-			-	=	-	-		-	-	-		=	-	+	+	+	F	F	F
	3	WOVE CUST TRAN	TO CUST THIN	Y	V																							1	+	+	1	T		
	L I	SET ANT EXT LISTING	FO UNIT POVE : CT	Ŷ	X																							1	+	+	+	T		Н
		CALL FULL BALLING LAGE	X OINN COID! CT	-	Μ	-			-					H												-	-	-	+	+	+	+		Н
		194.50 (III - II - 10 - 10 - 10 - 10 - 10 - 10 -	X CUST DISC WORK	H	H	-		\vdash	-			+-	\vdash	+	-			-	-		-		-		1	-	+	+	+	+	+	+	H	Н
	-		A		H					+		t					H				-				1		+	+	+	$^{+}$	+	+	\vdash	Н
	K	SET ONLY EXT. CUSTINI	Y DISCOUNT IT IN NO	V					-	+	-	-			-				-		-			-	-	-	-	+	+	+	+	+		Н
	-	THE HEITERT CUSTING	(HAN SOLD CT)									\vdash			-														+	$^{+}$	+	+		Н
	6	SET WORK TO THE	+AMTEXT: CUSTINU	X	Y	•																								T		T		
	7	WRITE	CUST INV	X	Y																											T		
	9	MOVE WORK TOTAL	TO OMT BULL TOT : TO-			X	-																									T		
	-4.	The month in the	TAL BILLING			-																								T		T		
	3	WRITE	Total BILLING	•	•	X																								T		T		
	IC	MOVE 0	TO WORK TOTAL	•	•	X																												
1	11	MOVE CUST HEDNG	TO WORK A			X																		1						Τ				
	12	GOTO	GET REC	X	X	X																												
				-			-		-		-			-	-	-			_	_	_		_	_	_	_	-	4	-	-	-	+	1	-

Figure 5a. Decision Table, Example 5.

Example 5 is part of the procedure Invoice Preparation in the system called Customer Billing, and is concerned with the calculation of an amount for extended sales, the calculation of a discount value for the extended sales amount, and the writing of a customer invoice.

Condition row 1 in rule 1, reads:

... Is the customer number on the record named customer heading the same as the customer number on work record A...?

WORK A is an illustration of a <u>work operand</u>, which is used for the temporary holding of either records or fields. The names of these operands must always be preceded by the word WORK.

A work operand always takes on the characteristics of the field or record which is placed in it, except for the name. It is assumed that there is sufficient space to contain all work records and work fields. The work operand will retain the value last placed in it, until something else is placed in it.

In this example, a record has been previously placed in the work record. A field in this record (not shown) is now being compared for correspondence with a field value of another record abbreviated CUST HEDNG (also not shown). Action row 11 of rule 3, demonstrates the placing of this record in the work record in preparation for the next pass through this table. This portion of rule 3 reads as follows:

2.29

21	TABLE NAME	TABLE TYPE				[-		Q	1	2	CY	1			T		T							
081	I OPER OPERAND 1	OP OPERANDE	C	9	-	0	P6	RAI	ND	٤	0	•	97	(PPI	EP.	ANG	22	0	Þ,	• •	PE	RA	ND	2		0	P**	OP	ER	NNC	2	
-			-	01	63	04	04	86	07	-	6	10	11	13.	13	14	15	16	17	18	14	30	25	43 1	19 1	14 13	-	20 2	12	1 2 9	30	-	1
* 02	PLEXTEND DETAIL	OPEN	20	50	20				-				-	_		-	_	_	_	_	-	-	-	+	-	+	-	+	+	+	+	⊢	_
L	CUST NO: CUST HED NG	EQ CLIST NO: WORKA	Y	V	N					-			4	_	-	_		_	_	-	-	-	4	+	_	-	4	-	+	+	+	\square	_
2	DISCOUNT (TTEM NO:CUST	15 PRESENT	Y	N	•										_																		
	T'RAN, GUAN SOID: CUST																		1.														
	TRAN)			Π									I													Т							
F			F	F		F	F		F			-	-	-		-	-	-	-	-	-	-	+	+	+	+	7	+	Ŧ	Ŧ	F	F	=
13	MOVE CUST TRAN	TO CUST THIN	X	Y																				+	1	+	+	+	T	T	T	H	
4	SET AMT EXT: CUSTINU	EQUNIT PRICE : CT	Y	X																				T				T	T	T	T		
1		Y OHAN SOLD: C.T.	r																					T	1		1	1	T	T	T	H	Π
		A CUST DISC WORK						-					-										+	+	+		+	1	+	+	\top	H	-
F		A																						+	1	1	1		T	1	T	П	
5	SET AMY EXT: CUST INV	H DISCOUNT(ITEM NOCT	X	•	•																								T	T	T		
		(QUAN SOLD:CT)																	-			_											
6	SET WORK TO TAL	+AMTEXT: CUSTINU	X	X	•																				1								
17	WRITE	CUST INV	X	X																													
19	MOVE WORK TOTAL	TO AMT BILL TOT : TO			X																												
		TAL BILLING																															
3	WRITE	TOTAL BILLING	·	•	X							-																	T	T			
110	MOVE O	TO WORK TOTAL	•	•	X																		1										
11	MOVE CUST HEDNG	TO WORK A			X																									1			
15	GOTO	GET REC	X	X	X			\mathbf{r}^{2}																	1				1				
				-		-			-	-			_	-	-	_	-	-	-	_	_	_	-	_	-	+	4	_	-	-	+	-+	-

Figure 5a. Decision Table, Example 5.

...move the record named customer heading to the work record named A... When the contents of a record are moved into a work record, the work record contains all the fields, and also the same field names as they appeared in the original record. The record then exists in identical form in two different places. To refer to these fields, the user must qualify the field name with the record name, WORK A, e. g., QTY ON HAND: WORK A.

A work operand may be either a <u>work record</u> or a <u>work field</u>. Action row 6 of rule 1 is read as follows:

... the work field, WORK TOTAL, is incremented by the value of the field, AMT EXT, from the record named CUST INV...

A value has previously been placed in the work field. It is now replaced by this value plus some other value. The remainder of the work record does not change as a result of this action.

The movement of one record to another is different. When records are assigned to other records, only those fields with corresponding names are assigned. To illustrate this, consider two records A and B. They are defined:

Record A, contains fields named Q, R, S, T, U, V, and W.

Record B, contains fields named Q, R, X, Y, Z, S, and T. With the operator MOVE in the statement

MOVE A TO B,

the values of corresponding fields, Q, R, S, and T of record A, are moved

- 1	2914	TABLE NAME	TABLE TYPE	E											1	1		ЩG	ų	Ę٢	IC	Y			1										
	5892	OPER OPSRAND 1	OP OPERAND #	0	p '	24	0	PS	RAI	ND	٤	1	0p	0.		05	PET	LAP	0		OP	-00	0	PE	RA	ND	2		0	P°	*0	PER	AAS	10 1	L
1	-			-	01	03	-04	10	84	47	(48	84	10	11	17	11	114	11	14	1	11	T TP		10	11	32	13	84	ÅS.	1.0	143	10 1	10	0 3	33
*	0.0	ETTEND DETAIL	OPEN	50	50	20								_			-	-	-	1	1	1	-	-	_	_		_		_	_	_	-		_
1	1	CUST NO: CUST HILD NG	ED CAST NO: WORKA	Y.	V	M										_				1	1	1													
1	2	DISCOUNT (TTEM NO:CUST	15 PRESENT	Y	N								1			1_		1	1	1	1	1													
	1	T'RAN, GUAN SOID: CUST											1							1					1										
Į	_	TRAN)																-	1	L	-	1				_						1	-	-	T
	10	WOVE CUST TRAN	TO CUST TAIL	V	V		-	-	F	-	-	F	-	F	-	F	F	F	Ŧ	Ŧ	F	Ŧ	Ŧ	-	+	-	-	-		-		Ŧ	Ŧ	Ŧ	Ŧ
1	N.	GET OMTENT CISTINU	EA UNIT POVE : AT		2					-		\vdash	+-	+	+	+	+	+	+	+	t	+	Ť	+	+	+	-			-	t	+	+	+	+
ł	7	DEL FIFILEAL GOLLAN	V ONALL CALDER T	1	1	-	-	-	-	-		+	+	+	+	+	+	+	+-	t	+-	+	+	+	+	+	-			-	+	+	+	+	+
ł	-		- GURIN SULL: CI		H	-	-	\vdash	-	\vdash	+	ł	+	+	+	+	+-	+	+	+	+	+	+	+	+	+	-	-		-	+	+	+	+	+
	-		A DISC. WORK		H			+	\vdash			\vdash	t	t	+	t	$^{+}$	t	t	t	t	t	+	+	+	+	1				+	+	+	+	+
1	FI	SET ANT ELT: CUST TNV	A DISCOUNT/ITEM NOCT	Y							1			T	T	T	T	T	T	T	T	T	T	T								T	T	T	T
1			(JUAN SOLD:CT)							1				T	1			T	1	T														T	T
1	6	GET WORK TO THE	+AMTEXT: CUSTINU	X	18								T		1	T		T	T	T	Т	T		Τ										T	T
1	7	WRITE	EUST INV	X	X								Г	Г	T	Т	T	Т	Τ	Τ	T	Τ	T		Т									1	T
Î	8	MOVE WORK TOTAL	TO AMT BILL TOT : TO			X							T	Г	Т		1	Т	T	T	T	Т	T	T								T	Τ	T	
1			TAL BILLING													Τ	T		1	Ι	L	Τ													L
1	3	WRITE	TOTAL BILLING	+		χ							Γ	Τ			T	T	Т	Τ	ſ	T	1	T									T	T	T
Į	IC	MOVE O	TO WORK TOTAL	•		X													1	1	1														
l	11	MOVE CUST HEDNG	TO WORK A			X		1.									1					1										1			
1	12	GOTO	GET REG	X	X	X					1																				1				
1	-			-	-	-	-		-	-	-		1	-	1	-	+-	1	+-	+	+	+	1	-	-	-	-	-	-	-	-	+	+	+	+

Figure 5a. Decision Table, Example 5.

to record B. Fields X, Y, and Z of record B are unchanged. The original record A is also unchanged. A similar operation can be performed with the SET operator. Note that the corresponding fields of A and B are moved regardless of their position on the record. This illustrates the fact that a file description is made up without regard to the positions of the fields. Action row 3,8, and 11 illustrate assignment of records:

Condition row 2 (on figure 5a) of rule 1 reads:

... Is the discount (defined as a function of the item number and quantity of that item sold according to a record named CUST TRAN) present...?

This condition row introduces an operand which has not been previously described, a <u>reference list function</u>. A reference list serves a purpose similar to that of a matrix or table of values. Each reference list function (like discount) is a function of either one or two arguments (two in our example: item number and quantity sold). These arguments may be considered as the vertical and horizontal axes of a tabular array, as shown below:

	15	6	UANTITY	
	ITEM NOS.	Col. 1 1-30	Col. 2 31-100	Col. 3 101-1000
Row 1	00000 -22040	+.98	+.95	+.88
Row 2	50000 -79999	+.97	+.94	+.87

2.31

1	TAMA F	TABLE NAME	TABLE TYPE	Г	Γ					Π				T		-		Q	E	N	Y	T	Τ	Т		Т		T	T	T	T		
	出版政	OPER OPERAND 1	OP OPERAND2	4	P	201	0	PE	RAS	ND 2	2	0	P	01	C	PE	RA	M	2	0	P	0	PE	RA	ND	2		0	> **	OP	ERA	ND	2
	-				81	03	04	9	04	87	95	09	10	11	2	18 1	14	15	16	17	16	19	20	21	\$2	25	54	15 1	10 2	/ 21	19	20	31 31
*	000	EXTEND DETAIL	DPEN	30	50	29							_	-	_	_	-	_	_		_	_	-		_	_		_	_	-	1		-
	1	CUST NO: CUST HED NG	EQ CUST NO: WORKA	Y	Y	N														_													
	2	DISCOUNT (ITEM NO: CUST	15 PRESENT	Y	N	•																											
		TRAN, QUAN SOLD: CUST													1	T						1			1						L		
		TRAN)			T										1													T		T	1	Π	
		- Child I		F	+	1	=	=	=		-		=	+	4	==	+	=	=	=	=	=	=	+	=	+	-	+	+	Ŧ	F	F	=
	3	MOVE CUST TREN	TO CUST TNIV	Y	Y		t			H			+	+	t	+	+	1	1	1		1	+	1		+	1	+	+	+	1		
	4	SET AMT EXT (USTINU	EQUINT PRES CT	Ŷ	X			T	-					+	+	+	+		-1	-			1		1	1		1	+	T	\top		
		LE T THE BALLOUS LADER	Y OUAN COID: CT		10		-	H						+	+	+	+	-	-	1	-	-	+	1	-	-		+		+	1	11	
	-		Y CINET DICC IMORY		+	-	\vdash	-	-	\vdash	-		+	+	+	+	+	-	-	-	+	-+	+	+	+	+	-	+	+	+	+		
			A CUST DISC. WORK	-	+	-	-	+			-		+	+	+	+	+	-	-	-	-+	-+	-+	+	+	+	-	-		+	+	H	
	-	The second states in the second states of	A		+-	-	-	+	-	\vdash	-		+	+	+	+	+	-	-	-	-	-+	+	-	+	-+	-	-	-	+	+	++	
1	2	SET AMIT ELT: CUST INV	+ DISCOUNTLITEM NOC	X	ŀ	ŀ.	-	-	-	\square	-		+	+	+	+	-	-	-	-	-	-	+	+	-	-+	-	+	+	+-	+	H	
	-		QUAN SOLD:CT)		+	-	-	-	-	\square	_		-	+	+	+	+	-	-	-	-	-	\rightarrow	+	+	+	-	+	+	+	+-	-	
	6	SET WORK TO TAL	+AMTEXT: CUSTINU	X	X	•		-			_		-	-+-	+	+	+	-+	-	-	_	-	+	-	4	+	-	+	+	+	+		-+-
	7	WRITE	CUST INV	X	X										_		_						_		_	_		_			1		
	8	MOVE WORK TOTAL	TO AMT BILL TOT : TO			X							<u> </u>																				
- 1			TAL BILLING																														
	91	WRITE	TOTAL BILLING			X									T		T							T				1		T	T		
1	10.	MOVE O	TO WORK TOTAL	1.	1.	X																		1				T		T	T		
- 1	11	MOVE CUST HEDNE	TO WORK A			X									1		T																
1	12	GOTO	GET REC	X	X	X																						T			1		
1	-			1	1			-	_	-	-		-	+	+	-	+	-	1	-	-	-	-+	4	-	-	-	-	-	-	-	-	-

Figure 5a.	Decision	Table,	Example	5.
------------	----------	--------	---------	----

SYS	STEM NAME STOMER BILLING	REFERENCE OF	BM DESCRIPTION	ANALYST FRANCIS	DATE	101	PAGE OF	PAGES
569 NO	NAME	TYPE		VALU	E			
1	QUANTITY DISLOWTS	REF LIST	ITEM RANGE	QUAN	RANGE	PIS	COUNT	
	DISCOUNFLITEM RAME	FUNK NAME	00000 - 2:2040	1 - 3	30	+ .	98	
	QUAN RANGE]	(2,3)	00000 - 22090	31 -	100	+.	95	
			00000 - 22040	101 -1	000	+ .1	88	
			50000 - 799999	1	30	+ .	97	
			50000 - 79999	31 - 1	00	+ . *	94	
			50000 - 79999	101 -	1000	7.8	7	
								~

Figure 5b. Reference Operand Description, Example 5.

SYS	TOMER BILLING			FILE	DE	SCRI	PTI	ON			F	FRANC	15	DATE 12/28/6	,	PAGE	OF	PAGES
FILE	NAME	DIR	L T	YPE		MEDI	IUM		R	ECO	RC	NAME		ABB	AVERA	NO OF	PEAK	RDS .
CUS	T DATA	I	N			SER	CAR	0	10	Cus	T	TRAN		CT	1,00	0	3,0	000
都在段 NO	FIELD NAME	ALPHA MIRMAN	-	LEN	TH	OCCUP MIN A	RREN	CES	ASBOX	ASANC TO		RELATIVE	END	V	ALUE/P	EMAR	KS	
1	CUST NO TYPE REC. CO	NN	VU	62				1			21							
3	ITEM NO	N	U	5				1										
5	UNIT PRICE	N	4	2.2			+	1			-		2	~~~~				_

Figure 5c. File Description, Example 5.

This tabular array can be described as a reference list and written on a Reference Operand Description form, shown partially on figure 5b. Here, the two axes of the original table are expressed in list form. For example, serial 6 of figure 5b expresses the information contained in row 2, column 2 of the matrix above. The quantity range is 31 thru 100, the range of item numbers 50000 thru 79999, and the function of these two arguments, discount quantity, is +.94.

When a range of values is used as an argument in a reference list, both the top and bottom limits of the range should be given. Arguments may be single entry constant values, ranges of values, or the names of value lists. If the name of a value list is given, the values represented by that name should be separately defined on a Reference Operand Description form as noted in example 4.

Reference lists cannot be modified by the actions of a decision table. Using condition row 2 as the illustration, specific entries within a list are referred to in this form:

DISCOUNT (ITEM NO: CUST TRAN, QUAN SOLD: CUST TRAN) Here variables 1 and 2 are field names on the input record CUST TRAN. They are not unique names and consequently are qualified in the normal manner. If they were unique names, the reference of condition row 2 would be:

DISCOUNT (ITEM NO, QUAN SOLD).

1	10	TABLE NAME	TABLE TYPE	T	T									T		-	28	Q	JE	N	24	·	Τ	Τ						T	T	T		1
	陶瓷	OPER OPERAND !	OP OPERAND2	1	QC	81	0	-	RAI	-	2	0	P	08	4	OPI		ANO	2	0	P°	0	PE	RA	NC	2		0	P	OP	ER/	ND	2	
. F				E	61	03	04	10	06	et.	8	04	10	11	12	18	14	18	16	17	16	14	20	21	82	25	54	25	10 1	7 2	8 29	10	31 2	5
* j	000	EXTEND DETAIL	OPEN	BO	50	29					_		_	_	_		_		_		_	_	_		_		_	_	_	_				_
	1	CUST NO: CUST HED NG	EQ CUST NO: WORKA	Y	Y	N																_												
	2	DISCOUNT (ITEM NO: CUST	15 PRESENT	Y	N	•																												
ſ		TRAN, QUAN SOLD: CUST					Γ															T			T				Т		T			
r		TRANI																													1	T	T	1
F		LUSING /		ŧ	+	-	-	=			-		=	-			=	-			-	=	=	=	=	-	=	=	+	+	+	=	==	4
ł	3	MOVE CUST TRON	TO CUST TAIL		V		-	-					+	+		-	-					+	+	+	+	-	+	+	+	+	+	+		1
ľ	U I	SET ONT EXT OUSTIN	EQUALT POVE 'AT	ÎŶ	1x		t	+	+		-		-	+	-		1		-			-	-	+	-	-1		+	+	+	+	+		1
F	7	DET MPTEATOUTING	NOUND FRAE OF	₽^	+	F-	+	+	+	-	-	-	+	+	-	-	-	-	-	-	-	-+	+	-+	-+	-	+	-+	-+-	+	+	+	i t	-
ŀ	-		-X GUAN SOLD: C.I	1	+	-	-	+-	+	-	-	-	+	-+	-		-	-	_	-	-	-+	+	+	-	-	+	-+	+	+	+	+	++	-
ŀ	-		* CUST DISC : WORK	4	+-	-	-	+-	+		-		-	+	-	-	-	-	-	-	-	-+	-	-+	-	-	-	-	+	+	+	+-	++	-
1	_		A		1								_	_	_				_		_	_	_	_	_	_			_	1	1		\vdash	4
L	5	SET AMT EXT: CUSTINV	+ DISCOUNT(ITEM NOC	X	•	•																												
L			QUAN SOLD: CT)	L											_													1						
E	6	SET WORK TO TRL	+AMTEXT: CUSTINV	X	X	•	1																											
	7	WRITE	CUST INV	X	X			Γ						Т															T	T	T			
r	9	MOVE WORK TOTAL	TO AMT BULL TOT ' TO		1.	X					-		-																	T	T	1		1
h			TAL DILLING	t	1	10	1	1	-				-	-+	-	-	-						-	-		-	-1		+	+	+	+		
ŀ	0	MOITE	TAL BILLING	-	+	1	+-	-	-		-		+	+	-		-			-	-	+	-	+	-	-	-	-	+	+	+	+	-	-
t	2	WRITE	IQIAL BILLING	₩÷	+·	1	+	-	+				-+	+	-	-	-	-	-	-	-	+	-+	-+	-	-	-	-+	+	+	+	┿	++	-
F	10.	MOVE 0	10 WORK TOTAL	ŀ		A	+	-	+		-		-	+	-	-	-	-	-	-	-	+	+	-+	-+	-	+	+	+	+	+	+-	++	-
F	11	MOVE CUST HEDNG	D WORK A	ŀ	*	A	-	-	-	-	_		-	-	-		-	-	-		-	-	-	+	-	-	-	-+	+	+	+	+	++-	-
	12	GO TO	GET REC	II X	X	X	-	-	-		_		-	_	-		_	-	_	-	_	-	-	-	-	-	-	_	+	+	+	-		_
	-	and the second s			1	-	-	-	_	-	-		-	-+	-	-	-	-	_	-	-	-+	-+	-+		-	-	-	-	-	-	-		1

Figure 5a. Decision Table, Example 5.

SYS CU	STEM NAME STOMER BILLING	REFERENCE OF	BM DESCRIPTION	FRANCIS	DATE	101	PAGE OF	PAGES
869 NO	NAME	TYPE		VALU	E			
1	QUANTITY DISLOUTS	REF LIST	ITEM RANGE	QUAN	RANGE	PIS	WUNT	
	DISCOUNTLIFEMRANCE	FUNK NAME	00000 - 22040	1 -	30	+.	98	
	QUAN RANGE]	(2,3)	00000 - 22090	31 -	100	+.	95	
			00000 - 22040	101 -1	000	+ .1	88	
			50000 - 79999	1	30	+ .	97	
			50000 - 79999	31 - 1	100	+ . *	94	
			50000 - 79999	101 -	1000	7.8	7	
-				-				~

Figure 5b. Reference Operand Description, Example 5.

SYS	TOMER BILLING		1	FILE	DES	BM CRIPT	ION				FRANC	5T 15	DATE 12/28/6	/	PAGE	OF	PAGES
FILE	NAME	DIR	t.T	YPE		MEDIUM	-	R	ECO	MR			ABS	AVERA	NO OF	PEAK	RDS
CUS	T DATA	I	N			SER-CA	RD	10	Cus	T	TRAN		CT	1,00	0	3,0	000
SER NO	FIELD NAME	ALPRIA MURRINE	8+0X	LENT		CCURRE	MAX	ASSOL	ASAPC TO	12mmon	RELATIVE	END	V	ALUE/P		KS	
1 2 3	CUST NO TYPE REC. CO	ZZZ	U U U U	625	-		1			21							
4 5	QUAN SOLD	NN	* *	4	1		1		H								

Figure 5c. File Description, Example 5.

To record information externally, an input/output operator WRITE is used, as noted in action row 9. The operand (e.g., TOTAL BILLING) always appears in the second operand position of the action. For limited entry tables like Figure 5a, this would be in the action stub; for extended entry, it is located in the entry area of the rule.

The remainder of the decision table shown in Figure 5a is selfexplanatory, except for a new notation in the condition and action entries. This is the placement of a period (.) in the entry for a rule when that condition or action entry is not pertinent. This is a convenience which signifies that the analyst has considered the action or condition of that row, and has decided that it does not apply to the rule. A period has exactly the same meaning as the blank entry.

QUESTIONS

1.	What is a work operand? What are its characteristics?
2.	What happens when one record is moved to another record?
3.	How is a specific entry of a reference list referred to in a decision table?
4.	What is the operation specified in action row 4 of the decision table shown in Figure 5a?
5.	What is the operation specified in action row 10 of the decision table shown in Figure 5a?

TAM.	TABLE NAME	TABLE	TYPE			Т	T	T	Π		T			-	284	QU	E	NC	Ý	T	T				T	T		T	TT	
SAR	OPER OPERAND I	OP (OPERAND2	0	P	54	OP	BRA	ND 2	.	OP	04		OPI	IR.A	NO	2	OF	93	OP	ER.	ANI	D 2		0	p et (PE	RAN	02	
				61	01	63	04 9	6 06	07.0	0 0	9 10	11	12	18	14	10 1	6	71		110	2 21	22	23	14	15	4 E7	28	29 3	1111	51
000	ETTEND DETAIL	OPI	EN	30	50	20				_					_	_				1										
1	CUST NO: CUST HER NG	EQUIST	NO: WORKA	Y	V	N																								
2	DISCOUNT (ITEM NO: CUST	15 PRE	SENT	Y	N	•											T													
	TRAN, QUAN SOID CUST								T											T									TT	
	TRANI				H	-	-	1	Ħ	+	-				+	+	1	+	+	+	+				-	1		-	++	-
	1.511.01.7					=	+	+	++	=	+			=	=	=	+	=	+	+	+			=	=	+	+	F	++	=
17	MOVE CUST TRON	TOCUET	TAIN	V	V		+	+	++	+	+-	+	-	-	+	+	+	+	+	+	+		-		+	+		-+-	++	-
1×	LAST ONT SYT ALST THU	EQ UNIT	DOUCESCT	t	2	-	+	+	+	+	+	+		+	-	+	-+	+	+	+	+	-			+	+	+-	++	++	-
4	DET HOTELLUSILOU	EGUMIT	PRIE	A.	A	-	+	+	++	+	+	+	-	-	+	+	+	+	+	+	+	-	-	+	+		-	++	++	-
-		* QUAN	SOLD: CT			-	_	+	+	+	-	\vdash	-		-	+	+	+	+	+	+	-	_	-	-+	+	-		++	-
		* CUST	DISC : MORK			_	-	-	++	-	-		_		-	+	4	-	+	+	-				-	-	-	\vdash	++	_
	1	A																												
5	SET AMIT EXT: CUST INV	+ DISCOUN	UTUTEMNOCT	X	•					Т							Т													
		QUANS	OLD:CT)							T							Т		T						1				TT	
6	SET WORK TO THE	+ANT ET	T:CUSTINU	X	X	•		1	TT	T	1					T	T		1							T	T	T	T	
7	WARTE	CUST	INIV	X	X			1	TT	1									1		T					T		T	TT	
10	MOVE WORK TOTAL	TOOMT D	US TOT TO-	1	-	V	-	+		+				H	1	+	+	+	+	+	+				1	+	1			-
10	LIGVE NORA IVIAL	TAL	11.101.10	-	-	4	+	+	++	+	-	+		-	+	+	+	+	+	+	+	-	-	-	+	+	1		++	-
-	11/0/20	THP R	ILLINGT_		-	-		+-	┢╍┿	-		+-+		-	-	+	+	+	+	+-	+			-					++	-
19	WHITE	IDTAL B	ILLING	Ŀ	•	X	+	-	+	+		\vdash	-	-	+	+	+	+	+	+	+		-		+		+	+	++	-
10	MOVE O	TO WOR	TOTAL		·	X	_	-	+	-	-		_		-	-	4	+	-	-	+		-	_	-	-	-	+	++	_
11	MOVE CUST HEDNG	TO WOR	KA			X.	_	_		-	-		_		_	_	_	_	-	-	-				-	-	-	\vdash		-
112	GOTO	GET	REG	X	XI	X										1														_
	1				_	-	-	-	-	+	-	-	-	-	-	-	-		+	+	+			-	-	-	-	-	++	-

Figure 5a. Decision Table, Example 5.

SYS	STEM NAME	REFERENCE OF	BM DESCRIPTION	ANALYST FRANCIS	DATE 12/25/01	PAGE OF PAGES
SER NO	NAME	TYPE		VALU	E	_
7	QUANTITY DISCOUNTS DISCOUNT (ITEM RANGE QUAN RANGE)	REF 1157 FURK NAME (2,3)	17EM RANGE 00000 - 22040 00000 - 22040 00000 - 22040 50000 - 79999 50000 - 79999 50000 - 79999	QUAN 1 - 31 - 1Q1 - 131 - 131 - 131 - 131 - 131 - 131 - 131 - 131 - 141 - 151 -	RANGE 30 000 30 30 00 1000 7	DISCOUNT + .98 + .95 + .88 + .97 + .94 87

Figure 5b. Reference Operand Description, Example 5.

SYS	STEM NAME			FILE D	ESCR		ION				FRANC	IS	DATE 12/28/6	/	PAGE	OF	PAGES
FILE	NAME	DIR	T	YPE	ME	DIUN		1	ECO	NRC	NAME		ABB	AVERA		PEAK	DS
CUS	T DATA	I	N		SE	R-CA	RD	1	Cus	T	TRAN		CT	1,00	0	3,0	000
SEM NO	FIELD NAME	ALPHA ALPHANA	SIGN	LENTH MAX W	OCC	AVG	MAX	ASSOC	ASANC TO	CEQUENCE	RELATIVE	END	V	ALUE/F		RKS	
1.	CUST NO TYPE REC. CO	NN	VV	62	-		1			21							
2 2	QUAN SOLD	N	+	4	-		1			_							
	UNIT PRICE	N	2	44	to		1-		1	-	-	2			-		

Figure 5c. File Description, Example 5.

ANSWERS

- 1. A work operand is an operand used for the temporary storage of either records or fields. It has the characteristics of the last item assigned to it.
- 2. An assignment operator used with records specifies the movement of corresponding fields of the sending record to the receiving record. The only exception to this operation is the movement of a record to a work record. In this case, the entire record is moved to the work record. The work record is then a carbon copy of the sending record.
- 3. A specific entry of a reference list is referred by writing the function name, and enclosing in parentheses the variables which correspond to arguments. If necessary, the variables are qualified in the usual manner.
- 4. This is an example of multi-operand arithmetic using the SET operator with arithmetic connectors. The operation performed is:

The unit price, from the record CT, is multiplied by the value of the quantity sold from the same record, which in turn is multiplied by the value of the customer discount. The customer discount was derived previously from the reference list and moved to the WORK A record.

5. This is an example of moving a literal to a work field. Since the work field takes on the value of the field which is last placed there, WORK TOTAL now contains the single digit zero.

1	ID	TABLE NAME	TABLE TYPE	Г				Т	T	T	Т			F	RE	ວບ່	N	CY										
	自行民	OPER OPERAND 1	OP OPERAND 2	9	P	81	0	PER	LANI	02	10	P	08	OP	ERA	ND2	10	P	00	ERA	NC	2	10	P	10	PER	AN	02
*	OU	DED AND VELS	(const	12	02	03	04	08	06 4	21	09	110	11 1	2 13	14	5 18	17	16	14 2	0 21	22	15 1.	25	2.0	17	28 3	9 30	1 11 0
	C.L.	CADE . RE	OVEN	-	-			-+	-	+	11	-		+	\vdash	-	35		_			-	10					
1		LUDE . DS	EQ	-	1	-		-	-	-	1×	Past		-		-	X						'A	1				
1	2	SET EMP IDENTIC	EO EMO	┢	+			+	+	+	-	+	-	+	+	+	-		+	-		-				-	-	TT
- 1	3	SET BOND BAL . DIRCHARE	EG BONID BON BC	1	+			+	-+	+	+5	+	-	+	+ +	-	X	-		+ +	-	-	K	-		-	+	++
		LEAD LOAD TOTAL HISS.	+ BOALD DED	t	+			+	+	+	1			+	+	+		-		+	-	+	+×	-		+	+	++
1	4	SET STR BAL: PURCHASE	LO STK BAL . BS	T	-			+	-	+	+		+	+	+	+	×		-			+	tx	1	\vdash	-+-	+	++
1			+ STK DED	L				1			1			1	H	-			1				f	1		+	+	++
	5	Do									bi	2					01	3		Π			bi	2				
[6	Do								T	T			T									61	3				
	7	GO TO		Da.	3						CI	6				_	CI	6					þi	6	П	-	1	\square
	012	BOND DIMULACE TECT	0.00-0	1.	-		+	+	+	+	-		-	+	\vdash	-	\square	-	+		-	+	┝	-	\vdash	+	+	++
	1	BOND RAL DUDGUASE	CH()3#L/	Ec		80	10	d	120	1	- Part	25	-	+-		+		-	+	+ +	-	+	t	-	H	+	+	+
1		DUNC DAL, TURCHASE		T -		AS	E I	PRI	C		+	\square	-	+		+		+	+		+	+	+	1	\square	+	+	$^{++}$
1	2	SET BOND PRICE	EQ	T		80	NO	PL	RE	- 14	1																	
						42	E	PR	CE																			
1	3	SET CODE : PURCHASE	EQ			`a'					T																	T
1	4	SET BOND BAL PUR HASE	•			ec.	ND	Ph	RC	H -																		
						ACE		PRI	CE																			
1	5	DO	PREPARE P.O.	X																								
	6	DO	012	X			-	-	-	-	1		-	-		-		_	-		_	_			\square	-	+	+
1	013	STH BIOGHASE THET	CLASED	20	E	\square	+	+	+	+	+	H	-	+	\vdash	+	\vdash	+	+		-	+	+	-	H	+	+	++
		STK BAL DIRCHOSE	LE MET MALLE	1V	-		1	+	+	+	+		1	+		+			+	11	-	+	+	\vdash	\square	+	+	++
	2	SET STK PRICE	TO MET VALUE				-	+	+	+	1	\square	-	+		-		+	+		-	+	t		\top	+	+	++
1	3	SET STR BOL DUP WASE	- MET VALUE	X				1	1	+				1					1	Ħ		-				+	+	Ħ
1	4	SET CODE : PURCHASE	EQ 'S'	X			1	-					-						-	11	1	-	\top			T	T	T
1	5	PO	PREPARE P.O.	X				1	1	-				T					-	П						T	T	T
	6	DO	013	X					1																	T		
				1					1																			

Figure 6a. Decision Tables, Example 6.

FILE DESCRIPTION									ţ	JONE	5	1/3/62		PAGE	OF	PAGES
DIR. TYPE			MEDIUM SER- CARD			RECORD NAME			ASS	SB AVERA		NO OF RECORDS				
IN						BOND STOCK DED				(DED	BS	50	5000 700		000	
State MAX VAR		OCCURRENCES			ASSOC TO ASSOC			RELATIVE	END	V	REMARKS					
22222222	C + + + + + C	20 52 32 32 4.2 3 1 3				111111111										•
	RIH MANA RZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	R H MANAGAN A ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ	DIR. TYPE IN MAX A 2C N U 6 N + 32 N	DIR. TYPE IN NAX VAR A 2C / N U 6 N + 32 N + 32 N + 32 N + 42 N + 32 N + 42 N + 32 N + 32	DIR. TYPE MEI IN SE IN SE MAX VAR MIN A $2C$ / N U 6 N + 32 N + 32 N + 4.2 N + 32 N + 4.2 N + 32 N + 32	DIR. TYPE MEDIUM IN SER-CA MAX VAR MIN AVS A $2C$ / N U 6 N + 32 N + 33 N + 32 N + 32	DIR TYPE MEDIUM IN SER-CARD MAX VAR MIN AVS MAX A $2C$ / 1 N U 6 N + 32 1 N + 3	DIR TYPE MEDIUM R IN SER-CARD 8 MAX VAR MIN AV5 MAX A $2C$ / 1 N 4 32 1 N $+$ 32	DIR TYPE MEDIUM RECO IN SER-CARD BON MAX VAR MIN AVS MAX A $2C$ / 1 N U 6 N $+ 32$ 1 N $+ 32$	DIR TYPE MEDIUM RECORD IN SER-CARD BOND MAX VAR MIN AVS MAX A $2C$ / 1 N U 6 N $+ 32$ 1 N $+ $	DIR TYPE MEDIUM RECORD NAME IN SER-CARD BOND STOCH MAX VAR MIN AVS MAX A $2C$ / 1 N U 6 N $+ 32$ 1 N $+ 3$	DIR TYPE MEDIUM RECORD NAME IN SER-CARD BOND STOCK DED	DIR. TYPEMEDIUMRECORD NAMEABBINSER-CARDBONDSTOCK DEDBSINSER-CARDBONDSTOCK DEDBSINILENTHOCCURRENCESInternet in the second	DIR TYPE MEDIUM RECORD NAME ASS IN SER-CARD BOND STOCK DED BS 50 WALVERMENT AND MAX SERVICES TO RELATIVE LOCATION MAX VAR MIN AVG MAX SEVER BEGIN END N U 6 1 N + 32 1 N	DIR. TYPEMEDIUMRECORD NAMEABSNO OFINSER-CARDBONDSTOCK DEDBS5000NLENTHOCCURRENCES $\frac{1}{2}$ $\frac{1}$	DIR. TYPEMEDIUMRECORD NAMEABSNO OF RECOIDINSER-CARDBONDSTOCK DEDBS500070NLENTHOCCURRENCES $\frac{1}{20}$

Figure 6b. File Description 1, Example 6.

PA	TEM NAME		FILE DESCRIPTION									ANALYS	ST IES	DATE 1/3/62		PAGE	OF	PAGES	
FILE	DIF	DIR TYPE					MEDIUM SER-CARD			ORI	NAME		A88	AVIEN	10 05	RECOR	DS		
SER NO	FIELD NAME	N-MAR	Sidn	LEN	TH	OCC	AVG	MAX	ASBOK	TO	PHENODER	RELATIVE	LOCATION		ALUE /R	REMARKS			
123450	EMP IDENTIE BOND BAL BOND PRICE STK BAL STK PRICE CODE	ANNNA	+ + + +	3.2 3.2 3.2 3.2 1.2 3.2 1.2 1		0000	in to in	11111							,				

Example 6 treats a stock and bond purchase procedure, which should be easy to follow upon careful examination of the three tables.

The situation described in tables 011, 012, and 013 of figure 6a, is a deduction computation for stocks and bonds. It is assumed that prior to the execution of these tables, a record has been read that contains a code to indicate employee participation as follows:

N Neither bond nor stock

B Bond

S Stock

A Bond and stock

The record also contains the various fields shown on the File Description form, figure 6b.

Table 011 of figure 6a tests for participation in either or both plans. Rule 2 reads:

If The employee is buying bonds

then various employee information is placed in a field named employee identification

and

his previous bond balance plus the amount of this

bond deduction is placed in the field named bond balance

and

DO table 012

and

GO TO table 016
of the local division in which the local division in which the local division in which the local division is not the local division of the local division in the local division	TABLE NAME	TABLE TYPE						Т		Т	Γ	П	1	FRE	iqu	EN	ic	Ý								T	П
IER	OPER OPERAND 1	OP OPERAND2	10	9	81	0	PER	LAN	02	1	OP	08	0	PER	AND	2	OP	61	OPE	RA	ND	2	10	P°	OP	ERAN	ND :
			-	01	01	04	08	04	07 of	09	110	11	121	5 14	115 1		7 10	14	120	11	22 2	3 14	25	2.6	17 20	119	80 3
11	DED ANALYSIS	OPEN	40							1	5					3	\$						10				
1	CODE : BS	EQ	X							В	1					Ŀ	ź						'A'				
	CETT				_	-	-	-	-			\square	-	-	\square	+		-			-	+		\square	-	+	
2	SET ENPIDENTIC	EQEMP			-	-	-	-	-	X	-	\downarrow	-	+	++	12	4	+			+	+	X	-	-	++	\vdash
2	SET BOND BAL: PURCHASE	EQ BOND BAL: BS							_	X				_		-		1				-	X				
-		+ BOND DED					_		_																		
41	SET STK BAL: PURCHASE	LO STK BAL . BS														X		1					X	-			
_		+ STK DED																									
5	Do									b	2			T	IT	b	13						01	2			
6	Do									T	T				T	T	T						61	3		T	
7	GO TO		n	8				T		6	16				\square	1	16	1					hi	6			
			F			1	-	+	1	ř	12	H	+	-	$^{++}$	T	15	+			-	1	1	*	-	++	\square
212	BOND DIMONASE TEST	CLOSED	40			+	+	+	-	6	50		+	+	++	+	+	+		-	+	+		+	+	++	H
1	ROND RAL ! DIRCHASE		K.F		BOL	10	a	12		-	HAE.	+	+	+	++	+	+				+	+		\uparrow	+	++	
-+	LACON LINE TURNINASE		T C			and	00			+	+	+	+	+	++	+	+	+			+	+		+	+	++	H
2	SET BOND DOLOT	EA			22	E	TK	IC S	-	+	+	\vdash	+	+	++	+	+	+		+	+	+		+	+	++	H
~	SET BONG PRICE	EG			102	00	P	RE	H -	+	+	+	+	+	+	+	+	+	-	-	+	+		+	+	++	H
3	SET CODE : PURCHASE	EQ		-	12	-	PK	5	-	+	+	H	+	+		+	+	+		-	+	+		+	+	++	H
4	SET BOND BAL DUDCHOSE		1		ant		2	10	- 14	T	+	H		+	H	+	+	+			-	-			-	11	
-			1		ACT	X.	DO	0	-	+		H	+	1	1 +	+	+	1				+			+	++	
5	Do	DEF DADE DO				1	-	-	-	+		H	+	-	++	+	+	+			+	+			+	++	
21	DO	012	Ŷ		-	-	+	+	+	+	+	+	+	+	++	+	+	+			+	+		+	+	++	H
-	~~~~	U.A.	r			+	-	+	+	+	+	H	+	+	+	+	+	+			+	+		+	+	++	H
213	STK PURCHASE TEST	CLOSED	30	E					-	1						+	1	1				1				T	
1	STK BAL: PURCHASE	GE MKT VALUE	Y							Т						T								T		T	Π
2	SET STK PRICE	EQ MKT VALUE	X												TT	T											
3	SET STK BOL DUPCHASE	- MKT VALUE	X							T						T						T				T	
4	SET CODE : PURCHASE	EQ 'S'	X														T					1		T	-	T	
5	DO	PREPARE P.O.	X		1				1	1			1	1	t t	1	1	T			1	1			-	\square	H
6	DO	013	X			1				T				1		T	T							T		TT	T
-		w. w			1	+	+	+	+	+		-	+	-		+	+	1			-	+			+	++	H
RE	MARKS: END SE TO	IN LOC NO'BE	DE E	7.	P	5	5	- 10	DA	10	R	<	EM	PI	NAU	16	. P	5	5	-	-	-			_		_
N	AFT VALUE AF (DAYS HIGH PRIC	F 4	5	A V	15	-	0	I E	DI	CE	3	12		and in	A. Sect.	-						_				
		Figure 6a	Г	0	ai	ci	Or		Ta	hl	00	,	ਸ਼ਾ	vo	mr	le	F										
-		I Igure da.	M		011	51		*	- a	NT.		,,	، لنظ	nd.	TTF	10			_	_		-					
SYS	STEM NAME	200		-					H	AN/	ALY	31	-		-	UA	E			_		+		_		-	
0.				_																					-		

FILE	NAME	DIR	N	TPE	-	SE	DIUM	20	a	Cal	2	STOCH	DED	BS	AVERAGE	PEAK
0	BUCTION		-14	-	-	SE	K. CA	- FD	10	QN	U	51000	DEU	1 23	1 2000	1000
SER	EIGLD NAME	No.		LEN	тн	OCCI	URRE	NCES	18	-	10085	RELATIVE	LOCATION	V	ALLIE PENA	PKS
NO	FIELD MARIE	No.	ž	MAX	VAR	MIN	AVG	MAX	8	ĥ	51	BEGIN	END		ALOLY REITIN	
1	EMP NAME	A		20	1			1								
2	EMP NO	N	U	6				1			1					
3	BOND BAL	N	+	32				1								
4	BOND DED	N	+	32				1.								
5	BOND PURCHASE PRICE	N	+	32				1								
6	STK BAL	N	+	4.2				1								
7	SIK DED	N	+	32				1								
8	CODE	A		1		-		1								
9	DEPT	N	U	3				1								
10	LCC NO	N	U	3				1				-				
-		+	-	-		-	-	-	-	-	-					

Figure 6b. File Description 1, Example 6.

SYS	ROLL				0	IB	M				F	ANALYS	ST	DATE		1		
				FILE	DE	:50	RIPI	ION			L	JON	45	1/3/62		PAGE	OF	PAGES
FILE	NAME	DIS	R T	YPE		ME	DIUN		R	ECO	OR			A88	AVERA	NO OF	RECOR	IDS
Co	R DED BAL	0	Л			SE	R.CA	RD	11	VR	CH	ASE			100	00	15	00
HO	FIELD NAME	and the second	5.04	LEN	TH	OCC	URRE	NCES	A580	100	SEGUEN	RELATIVE	LOCATION	V	ALUE /R	EMAR	RKS	
1	EMP IDENTIF	A	ŧ	3.2	-			1	f	f	-	BEGIN	ENU		1			
2	BOND BAL	N	+	3.2		0	.5	1										
4	STK BAL	N	+	3,2	+	0	5	1	+		\square							
5	STK PRICE	N	Ŧ	3.2		0	.5	1										
6	CODE	A		1				1			_							

In the remarks section of the decision table it is noted that the EMP is a <u>symbolic expression</u> name (denoted by SE). In the table, this name has no identifying marks to indicate that it is such an expression. A symbolic expression relates various symbolic values much like an arithmetic expression relates numeric values. JOIN is the only operator; it describes the stringing together of two or more fields to form a new field. When-ever expressions like this are used in decision tables, they are immediately evaluated, and re-evaluated every time they appear.

In this example, the fields LOC NO, DEPT, EMP NO, and EMP NAME are qualified as being part of the record (see figure 6b) which is abbreviated BS. Since these are separate fields, it is not possible to say whether or not they are physically adjacent to one another. In order to be able to use them as a single value, the analyst describes the joining of the appropriate fields. If the expression is used only on a single Decision Table form, it need only be defined in the remarks, but if the expression is used throughout the system, it should be defined on the Reference Operand Description form. In either case, it is defined by writing the expression name, the type SE (symbolic expression), and the operator JOIN. The fields to be joined are then placed in parenthesis after the operator. As usual, the fields are qualified if they are not unique.

The JOIN operator may be used to form symbolic expressions by the joining of fields, literals, constants, descriptive terms, work field names, and reference list function names.

2.36

D BAL: BS	K 6 90	01	01 0		1 06	ND 2	2 08	01 19		1 12		14 15	102	0735	P 01	001	21 1	ND 23 2	2 24	0 25 10	20	°OF	B 14	ANC	5
D BAL : BS DED AL : BS	K 8 9	01	01 0		06	07	et	19	10 1	1 12	18	14 15	16	17 35	16 1	0 20	21	23 2	3 24	25 10	2.6	27 1	0 14	30	-
D BAL : BS DED AL : BS	× 8							B	+	+				35	+	+	+	+	+	10 'A	-	+	t	+	⊢
DBAL: BS			-	+			_	B						5			1 1		1.1	'A]		- 1-			
DBAL: BS			-	-				and the second					-		-	-	-	_	-	ينغن		-	-	+	╞
DBAL: BS		7			-	1			T				T								-	4	+	+	╀
DED			-		1			X						X	- 25					X	$ \rightarrow$	-	-	+	+
DED ALIBS								×												X		4	-	-	1
AL BS	12-12		1	1	T								1									-	+	+	+
COR L MAN				1										k						X		-	-	-	+
				1	1									Ι								-	_	-	1
			1		1			01	2					bi	3					bi	2				
			1	+	+			~1	-					Γ		1				DI	3				+
	-	=	-	+				01	6					1	6					bi	6				
		-		+	+				*	1			1	1											
	4.5		-			+		E.	Se	1	1													-	+
	CE		BO	10	OL 1	-								Γ		-									1
			4	- 0	2011																				
		H	22	10	DET		-			-	1		1												
				-	040	F				-	1			1											
			1			-	-			+	+			\top						T					1
		\square	B	-	1						+			t											
					P C	-				+	+	11	+												
20		\square	<u>a</u> C.	- 1	Reis	E E	-				+		+	+					1	Γ					T
P.C.	÷			+	+	+	+		H	-	-	$^{++}$	+	+						T					T
	r	\vdash		+	+	+	+			-	+	+	+	+			1		-	T					T
	5	-		+	+	+		\vdash	H	+	-	$^{++}$	+	\mathbf{t}						T					T
2	20	<u> </u>		+	+	+				+		++	+	t			1			T		\square			T
VALUE	L.	H		+	+	+	+		\vdash		-	$^{++}$	+	+		-	1			T					Τ
VALUE	Ь÷	-		+	+	+	+			-	+		+	1			1			T	T				
VAUE	E.	-		+	+	+	+	+		+		+	+	+	+	+	1	++	1	T					T
- 0 -	1÷	-		+	+	+	-	-		+	-	++	+	+		-	1	tt	1	T	T				T
E P.O.	K	-		+	+	+	+	+		+	-	++	+	+			1	H	-	T	T			T	T
	X.	-		+	+	+	+	-		+	-	++	+	+	+		+	+	-	1	T	T			T
		1														-		and the second s	the second se				_	_	-
	P.O. VALUE VALUE VALUE VALUE	P.O. X VALUE X VALUE X VALUE X VALUE X VALUE X	22 3 35 5E 5E 7 7 7 7 7 7 7 7 7 7 7 7 7	Dil 5 3:5 5 5:6 80 Ab 90 Ab	D2 5 35 56 80 ND A5E 6 80 ND A2E 8 80 ND A2E 8 80 ND A2E 9 30 C VALUE X VALUE X VALUE X VALUE X VALUE X VALUE X VALUE X VALUE X	D2 5 35 56 BOND PUT ASE PRIS 60 ND PUT ASE PRIS 80 ND PUT ASE PRIS 80 ND PUT ASE PRIS 90 ND PUT 90 ND PUT	22 5 35 56 66 80 ND PUR CH ASE 6RICE NO ND PUR CH AZE PRICE 18 80 ND PUR CH AZE PRICE 18 1 80 ND PUR CH AZE PRICE 18 1 10 10 10 10 10 10 10 10 10	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	$\begin{array}{c c c c c c c c c c c c c c c c c c c $					

Figure 6a. Decision Tables, Example 6.

SYS	YROLL		F	FILE	DE	SCR		ION				ANALYS JONE	T .5	DATE 1/3/62		PAGE	OF	PAGES
	NAME	DIR	T	PE		ME	DIUM		R	ECO	R		1	ASS	AVERA	NO OF	PEAK	RDS
DI	DUCTION	I	N			SE	R- CA	RD	8	ON	Q	STOCH	DED	BS	50	00	170	200
SER	FIELD NAME	MUNICAL	BIGN	LEN	TH	OCCU	AVG	MAX	A 5806	ASSOC TO	SEQUENCE	RELATIVE	END	v	ALUE/P	REMAI	RKS	
1237567007	EMP NAME EMP NO RUND BAL BOND DED ROND PURCHASE PRICE STK BAL STK DED CODE DEP T	AZZZZZAZ		20 6 32 32 32 42 32 42 32 1 32				1 1 1 1 1 1 1 1 1									(4.4).44	•

Figure 6b. File Description 1, Example 6.

PA	TEM NAME			FILE	DE	SCF		ION			E	ANALYS	ST €Ş	DATE 1/3/62		PAGE	OF	PAGES
FILE	NAME	DIR	ιT	YPE		ME	DIUM		R	ECO	R	NAME		A88	AVERA	NO OF	RECO	RDS
CU	R DED BAL	a	π			SE	R-CA	RD	P	UR	CH	ASE			100	00	15	00
SER NO	FIELD NAME	Manager,	5.0N		TH	OCC	AVG	MAX	A580	A4BOC TO	STORENCE	RELATIVE	END	V	ALUE/P		RKS	
12345	EMP IDENTIF BOND BAL BOND PRICE STK BAL STK PRICE	AZZZZ	+++++	3.2 3.2 3.2 4.2 3.2		0000	5555	11111						1				

The next to last action of rule 2 in table 011 is: DO 012. DO is a new operator which calls for executing the table specified, and then returning to the action which follows the DO in the particular rule. This corresponds closely to the linkage with a closed subroutine in normal programming. It relinquishes control to the identified table, and then returns when that table has been executed.

The type of table which is entered by a DO operator is a <u>closed table</u>. If a closed table contains a GO TO operator the return will be broken for that rule. Both open and closed tables may contain DO operators.

Rules pertaining to open and closed tables can be summarized in this chart:

	open table	closed table
entered by	GO TO	DO
exited by	GO TO	(automatic return to calling table)
may contain	DO, and GO TO	DO

The use of the automatic return for a closed table is shown in table 012. This table is entered by the DO operator of rules 2 or 4 of table 011. There are two rules for table 012; rule 1 is:

If the bond balance is greater than or equal to (GE) the bond purchase price,

then the bond purchase price is placed in the bond price, a 'B' is

2.0	TABLE NAME	TABLE TYPE												FR	EQ	UE	N	-1	1	1	1	1	-	-	- 04	-			-
SR	OPER OPERAND 1	OP OPERAND2	0	P *		OP	ER	ND	2	0	P	OR	0	PE	RAN	102	0	P	0	PER	ANI	02		0	2411	OP	ERA	ND	2
_	the second se		91	02	01 0	04 0	05 0	6 01	7 08	09	10	11	12	13 1	4 15	16	17	16	19	10 1	22	4.2		1	-	-	-	-	-
211	DED ANALYSIS	OPEN	40	-	-	_	-	-	-	15	\square	-	+	-	+	+	35	-	+	+	+	\vdash	-+'	10	+	+	+	+	+
1	CODE : BS	ÈQ	'N	-	-	+	-	+	-	B		-	-	-	-	+	X	-	+	-	+	\vdash	Ħ	4	+	+	+	Ħ	F
-	CCT		\vdash	+	+	+	+	+	+	V	+	-	+	+	+	+	X		1	+	1		t	x	1				
2	SET EMP IDENTIE	FORMD		+	+	+	+	+	+	t2		+	+	+	+	+	r^						T	X	T		T		
2	SET BOND BAL: PURCHASE	EQ BOND BAL: 85		-	+	+	+	+-	+	ŕ	+	-	+	+	+	+				+			T	1	T				
4	SET STY BAL DID CHASE	FOSTE BAL BE				+	+	+		1				1			x	1						X					
-	SET SIR DIL, PURCHASE	+ STK DED																	_	-	-		4	-	+	+	+	+	ł
5	DO								-	01	2	_		-	+	-	01	3	-		-	-	H	21	2	+	+	+	┝
6	Do						-	+	-	-		-	-	-+	+	-	-		-	+	+	+-	F	21	2	+	+	+	t
7	GO TO		2	3		-	-	-	+	01	6		-	-	+	+	K1	6	+	+	+	+	ł	21	6	+	+	+	t
0.2	Daily Distance There	0.000	35		-	+	+	+	+	6	50			+	+		1			+	1	t							t
UIL	BOND PURCHASE LEST	CHUSED	KE		BO	10	d	10 10	1	-	a de la																		
_	BOND BAL: PURCHASE		40		AS	F	PRI	0 =		1						1													I
2	SET BOND PRICE	EQ			80	NO	PL	RC	H -											-	-	+	\square		-	+	+	+	+
					A>	E	PR	CE		1	-				-	+	+	-		+	-	+	+	-		+	+	+	+
3	SET CODE : PURCHASE	EQ			B			-	-	-	-				-	+	+	-		-	+	+-	+	-	\vdash	+	+	+	$^{+}$
4	SET BOND BAL, PURCHASE	•			80	NO	PN	JRC	H -	+	+				1	+	+	-		-	+	+	H	-	\vdash	+	+	+	$^{+}$
					ACE	1	PR	CE	-	+	-	-			-	+	+	-		-+	+	+	+	\vdash	\vdash	+	+	+	+
5	DO	PREPARE P.O.	X					-		+	-				-	+	+	+		-	+	+	+	\vdash		+	+	+	+
6	DO	012	X			_		-	+	+	+	-		-	-	+	+	+		-	+	+	$\left \right $	H		+	+	+	1
017	STU DIGUISE TOCT	CLOSED	30	E			H	+	+	+						1	T										1	T	1
P	STY DALLOUGHASE	KE MET VALUE	Y	-				1													1						1	+	4
1	SET STY DOUCE	TO MET VALUE	K					1		T																	-	+	4
12	SET STE PALE	- MET VALUE	X					1		T																	_	-	4
H	SET ONE DE HASE	EQ 'S'	X																			-				-	+	+	4
1	DO	PREPARE P.O.	X													-	1	-	-		-	+	+		+	+	+	+	+
TL	DO	013	X																-		-	+	+	-	-		-	+	4
1-6	100	12.0	H							T		1											1					_	

Figure 6a. Decision Tables, Example 6.

SYS	YROLL		1	TILE	DE	SCR	M.	ION			[ANALYS JONE	T 5	DATE 1/3/62		PAGE	OF	PAGES
CH E	NAME	DIR	T	PE		ME	DIUM		R	ECO	R	D NAME	2	ABB	AVERA	NO OF	PEAK	RDS
DE	DUCTION	Ī	N			SEI	R- CA	RD	8	ON	ð	STOCH	(DED	BS	50	00	7	200
SER	FIELD NAME	MUBRERNE	SIGN		TH	OCCU	AVG	MAX	A\$804	ASSOC TO	SEQUENCE	RELATIVE	END	V	ALUE/F	REMA	RKS	
1	EMP NAME	AN	U	20	1			1										a
34	BOND BAL	N	+	32				1	+-			-						
567	STK BAL	NNN	+++++	4.2				1	-		F							
8	CODE DEPT	AN	U	1				1	+							- -	101152	
10	LCC NO	N	U	3		1	-	1	+	t	+							

Figure 6b. File Description 1, Example 6.

SYS	ROLL		1	FILE	DE	IB	M	ION			E	ANALYS	e s	DATE		PAGE	OF	PAGES
FILE	NAME	DIR	LT	YPE	an inc	ME	DIUM		R	ECC	ORD	NAME		A88	AVERA	NO OF	RECOR	RDS
CU	R DED BAL	a	π			SE	R-CA	RD	P	UR	CH	ASE			100	00	15	00
SER HO	FIELD NAME	ALPHA MUNICIPS	5.GN		TH	OCC	AVG	MAX	A\$806	TO	SEQUENCE	RELATIVE	LOCATION	V	ALUE / P	REMAR	RKS	
123456	EMP DENTIF BOND BAL BOND PRICE STK BAL STK PRICE CODE	A Z Z Z Z Z	+ + + +	3.2 3.2 3.2 4.2 3.2 4.2 3.2		0000	5555	111111										

placed in the purchase code, subtract the purchase price from the bond balance, prepare a purchase order, and do this table again.

The second time the table is entered, this time by its own DO operator, the actions of rule 1 are executed again if the condition is still satisfied. When the bond balance is not greater than or equal to the bond purchase price, then the conditions of the rule are not satisfied, and the ELSE rule is called. The ELSE rule has no actions, so the table is complete and control is returned to the original table (table 011), and to the action following the first DO action.

Rule 3 of table 011 specifies the logic for a man who participates in the stock purchase plan only, and rule 4 covers the employee who is in both plans.

Table 013 differs from table 012 in that the current market value of the stock is used in place of the fixed purchase price of the bond. The current market value is developed through the use of an <u>arithmetic</u> <u>expression (AE) explained at the base of the form in the remarks section,</u> as was the symbolic expression. An arithmetic expression is a description of a combination of operands and arithmetic operators which has a numeric value when evaluated. The operands can be numeric fields, numeric constants, numeric literals, numeric work fields and reference list function names. The five operators used in arithmetic expressions are: + addition, - subtraction, * multiplication, / division, ** exponentiation.

2.38

2.0	TABLE NAME	TAS	LE TYPE						T					1	FR	RQ	UE	N	Y												
BER	OPER OPERAND 1	OP	OPERAND 2	0	P	81	0	PER	AND	20	0	P	01	0	PE	RAN	DZ	0	P°	00	EF	LAN	D	2	0	P	OF	ER	AN	D	2
011		Leon I		-	01	03	04	06 (04 0	7 08	09	10	11	12	11 1	+ 15	16	17	16	19 3	0 1	1 2:	2 23	24	25	2.6 1	7 3	8 14	9 30	0 3	5
11	DED ANELYSIS	OPEN		40				+	+	+	115	\square	-	+	+	+-		39	+	+	+	+	+		10	\vdash	+	+	+	+	-
-	CODELIBS	EQ		M		-	-	+	+	+	B	\vdash	-	-	-	+	-	X	+	+	+	+	+		A	+	+	+	+	+	-
2	SET EMP IDENTIE	EO ER	np						1	1	X				+	1		x		+	t	1	1		Х		+	T	+	1	-
3	SET BOND BAL: PURCHASE	EQ BO	ND BAL : BS								X										Т				X	T	Т	Т	T	Т	ĺ
		+ BCN	DED								T																			T	Î
4	SET STK BAL: PURCHASE	LO STK	BAL BS															X							X						
		+ STK	DED											_																1	
5	DO										01	2						01	3						01	2					_
6	Do																								Q1	3					
7	GO TO			22	3			_	-	-	01	6			-	-		CI	6	-	+	-	+		01	6	+	+	+	+	
012	BOND PURCHASE TEST	CLOSE	0	35			H	+	+	+	E	Se	-	-	+	+-			+	+	+	+	+	+		H	+	+	+	+	•
1	BOND BAL: PURCHASE			GE		BO	ND	PA	RE	H						+					t		T				T	T	1	T	'
						As	Æ	PRI	CE																			T		T	į
2	SET BOND PRICE	EQ				80	NO	PL	RC	н -																					
						AZ	E	PR	CE																						
3	SET CODE : PURCHASE	EQ				`B'																									
4	SET BOND BAL PUR HASE	•				80	NO	PN	RC	H -						1															
						ACE		PRI	CE							1											-	-	-		
5	00	PREPAR	RE P.O.	X				-		_				_	_	-				_	_	_					_	_	_	_	1
6	Do	012		X		-	-	-	+	+	+		-	+	+	-	-		-	+	+	+	+		-	\vdash	+	+	+	+	
013	STK PURCHASE TEST	CLOSE	0	30	E			+	+	+	t	H		+	+	+			+	+	+	+	+	H	-	H	+	+	t	+	
1	STK BAL: PURCHASE	GE MK	TVALUE	Y																									T	T	į
2	SET STK PRICE	EQ MK	T VALUE	X																											
3	SET STK BAL: PURCHASE	- MK	TVALLE	X																											
4	SET CODE : PURCHASE	EQ 15	.*	X																											
5	PO	PREPA	RE P.O.	X																											1
6	DO	013		X																							-	1		-	
																															,

Figure 6a. Decision Tables, Example 6.

SYS PA	YROLL		1	FILE	DE	SCR		ION				JONE	T :5	DATE 1/3/62		PAGE	OF	PAGES
FILE	NAME	DIR	ι. T	YPE	_	ME	DIUM		R	ECC	R	NAME		ABB	AVERA	NO OF	RECOR	RDS
DI	DUCTION	1	N		_	SE	R- CA	RD	18	ON	ð	STOCH	(DED	BS	50	00	70	000
SER	FIELD NAME	AL PHA	\$10N	LEN	TH	OCCU	AVG	MAX	A\$806	AABOC TO	3 2MB/UDB	RELATIVE	END	V	ALUE/F		RKS	
123756789	EMP NAME EMP NO ROND BAL BOND DED BOND PURCHASE PRICE STK BAL STK DED CODE DEP T	AZZZZZZA Z	D++++ D	20 6 32 32 32 4.2 32 1 3 2												-		

Figure 6b. File Description 1, Example 6.

SYS	STEM NAME					IB	M				F	ANALYS	ST	DATE		1		
	· // () // · · · · · · · · · · · · · · · · ·			FILE	DE	SCI	RIPT	ION			L	JON	IES .	1/3/62		PAGE	OF	PAGES
FILE	NAME	DIF	LT	YPE		ME	DIUM	ř.	R	ECO	ORD	NAME		ABS	-	NO OF	RECOR	DS
Co	PR DED BAL		π			SE	R-CA	RD	P	VR	CH	ASE			100	0	15	00
NO	FIELD NAME	thumultus	Bigh		TH	OCC	AVG	MAX	ASSOC	4680C	ROMENCI		LOCATION	1	ALUE /R	EMAR	RKS	
1234	EMP IDENTIF BOND BAL BOND PRICE STK BAL	ANNN	++++	3.2 3.2 3.2 4.2		000	555	1 1 1 1										
5	STK PRICE	A	+	3.2		0	.5	1										

If an arithmetic expression is described in the remarks section of a table, it is a local arithmetic expression. When it is used throughout the system, it should be described on a Reference Operand Description form. In figure 6a, the local arithmetic expression is the market stock value. It is defined as: The day's high price for the stock, plus the day's low price for the stock; the sum divided by two.

QUESTIONS

- 1. What are the differences between an open table and a closed table?
- 2. What is the definition of a symbolic expression? How is it used?
- 3. What is the definition of an arithmetic expression? Where is it defined in TDL?
- 4. What is the sequence of actions for rule 4 of table 011 of this example?
- 5. In table 012, read both rules.

2	۲	TABLE NAME	TABLE TYPE								Γ			5	R	QL	EP	1c	Ý									
66	震	OPER OPERAND 1	OP OPERAND 2	0	P	H	0	PER	AND	2	0	P	0%	0	PER	AND	2	OP	01	OPE	RA	D a	-	0	·**	PER	AN	D Z
5		550	60x -1	81	01	03	04	04 (0 40	7 08	09	10	11 1	12 1	\$ 14	15	6 1	7 11	19	10	21 3	2 23	24	25 2	\$ 17	28 1	9 30	0 51
14	4	DED ANELYSIS	OPEN	40	\vdash	-+	-	+	+	+	113	$\left \right $	+	+	+	++	-12	3	+	-	+	+	\vdash	10	+	++	+	++
	4	CODELINS	EQ	N		-	+	+	+	+	B		+	+	+	+	÷	5	+		-	+	\vdash	A	+	++	+	+
1	e	SET ENP IDENTIE	EO EMP		H	+	+	+	+	+	x	H	+	+	+	t t	5	d	+		-	+	H	x	+	H	+	Ħ
	3	SET BOND BAL . PURCHASE	EN BONID BAL . BS							1	X	Π				$\uparrow \uparrow$	T	T						X				T
			+ BOAND DED						1	1	1	H	+		1	$^{++}$	+	1	T	1		T		1		T	T	T
1	41	SET STK BAL : PURCHASE	LO STK BAL : BS						1	1	1	H	-		1	†	5	1	T					x	1		1	11
	1	MANUAL MANUAL CONTRACTOR	A STK DED		\square					1	1	H		+	1	$^{++}$	Ť	1	+			1		1	1		-	11
	5	DO								1	h	5		1	+	+	1		1					71 2	2		T	
	6	Do		1	H	-		+	+	+	1	1	+		+	$^{++}$	۴	+	+		+	+		01	5	H	+	++
	7	GO TO		h	x					1	h	6	-	-	1	\uparrow	t	114	1					21			T	
	1	M.T. 1 M				-	-	1	+	1	P'	1	+	+	+	++	T	4	+		1	+	П	4	-	\square	+	++
01	12	BOND PURCHASE TEST	CLOSED	35		1		1	+	1	E.	SE		+	1		+	1	\uparrow			T		1		tt	+	+
r	1	BOND BAL ! PURCHASE	have be added and	6E		BO	ja	P	18 1	4	1			1	1	11	+	+				-				П	1	
						AL	-	120 1	0 =		T	\square					1											
	2	SET BOND PRICE	E (2		П	82	10	PL	Rr	4-	t	H		+	+	+	+	+	+			+	П			T	+	++
						1	-	DR	CE	1	\top				1		1		T							T		
	3	SET CODE : PURCHASE	EQ			'n	1			1	t	H		+	1	T	+	+					П			T	T	\top
	4	SET BOALD BAL PURCHASE				BOL	vd	PA	IRC	H -							T		T				П				T	
						ACE	1	PR	CE		1					i	T									T		
	5	Do	PREPARE P.O.	X				1	1		1												Π				T	
17	6	DO	012	X						T	Г						T		T				Π					
Г									T								T								T	T		
0	13	STK PURCHASE TEST	CLOSED	30	E												Т	T	T									
Г	1	STK BAL: PURCHASE	GE MKT VALUE	Y																								
	2	SET STK PRICE	EQ MKT VALUE	X																								
	3	SET STK BOL PURCHASE	- MKT VALUE	X							T					T												
	41	SET CODE : PURCHASE	EQ 'S'	X												П	T		T									
	5	Po	PREPARE P.O.	X																								
	6	DO	013	X			T		T							IT	T											
											T	T	Т		1		Т	Т	T									

MET VALUE AE (DAYS HIGH PRICE + DAYS LOW PRICE)/2

Figure 6a. Decision Tables, Example 6.

SYS	YROLL		1	FILE	DE	IB	M	ION			[ANALYS JON C	T (5	DATE 1/3/62		PAGE	OF	PAGES
FILE	NAME	DIR	λ. T	YPE	-	ME	DIUM		R	ECO	R	NAME	A	ABB	AVERA	NO OF	RECO	RDS
D	DUCTION	3	ĒN		_	SE	R- CA	RD	в	ON	ð	STOCH	K DED	BS	50	00	1.7	000
NO	FIELD NAME	MLMARM.		LEN	TH	OCCU	AVG	MAX	ASSOL	ASSOC TO	STOUGHCE	RELATIVE	END	V	ALUE/F	REMAI	RKS	
123750	EMP NAME EMP NO BOND BAL BOND DED BOND PURCHASE PRICE STK BAL	AZZZZZ	U + + + +	20 6 32 32 32 4.2	/			1 1 1 1 1										
789	SIK DED CODE DEPT	NANN	+ U U	32 1 3 3				1 1 1		-						1		

Figure 6b. File Description 1, Example 6.

SY!	STEM NAME					IB	M					ANALYS	ST	DATE]		
1.6	in Callena			FILE	DE	SCF	RIPT	ION			1	JON	IE S	1/3/62		PAGE	OF	PAGES
FILE	NAME	DIF	L T	YPE		ME	DIUM		R	RECO	OR			ABB	-	10 05	RECOR	DS
Cu	R DED BAL	0	π		-	SE	R-CA	RD	ſ	PUR	CH	ASE			100	0	15	00
SER HO	FIELD NAME	NLIVIA NLIVIA	81.0N		TH	OCC	AVG	MAX	A8804	10 To	SEOVENC	RELATIVE	LOCATION	V	ALUE/R	EMAR	RKS	
123456	EMP IDENTIF BOND BAL BOND PRICE STK BAL STK PRICE CODE	2222A	+++++	3.2 3.2 3.2 3.2 1.2 3.2 1.2 1		0000	5-12-12-12-12-12-12-12-12-12-12-12-12-12-	111111										

ANSWERS

- 1. An open table is entered by a GO TO operator and terminates with a GO TO operator. The closed table is entered by a DO operator and returns automatically to the action after the DO action of the same rule of the table from which it was called.
- A symbolic expression causes a stringing together of two or more fields to form a new field.
- 3. An arithmetic expression is a combination of numeric operands with arithmetic operators. This expression has a numeric value when evaluated. It may be defined either in the Remarks section of the tables or on a Reference Operand Description form.
- 4. If the conditions of the rule are satisfied, the actions are carried out to the DO action. The first DO action calls table 012, and the return is to rule 4 for the next DO action. This DO action calls table 013; return is to rule 4 again for execution of the GO TO 016 action.
- 5. a) <u>If</u> the bond balance is greater than or equal to the bond purchase price, <u>then</u> the bond purchase price is placed in the bond price, a 'B' is placed in the code of the purchase record, the bond purchase price is subtracted from the bond balance, a purchase order is prepared, and the table is repeated.

b) If the condition of the first rule is not satisfied, the ELSE rule is called, the table is completed and return is to the next action of the rule in the table from which it came.

TLE NAME	SYSTEM E	XPENSE DIS	T. ANALYST	<u>3.P.M.</u>	DATE 11/28/	GI PAGE.	0F
		MONTHLY	EXPENSE D	ISTRIBUTION	REPORT	REP DATE	PAGE
		NO	ACCT	INVOICE NO	INVO/	CE INVO E AMOU	ICE
DEPT DETAIL - V		***	NNNNN	AA-NNNN	ZN ZN	NN 22,222	NNCR
LEDGER TOT-V	A.R.W	GEN LEDG	ER NN			2,222,222	. NNCR
							+
END REPORT-V					L	AST PAGE	

Figure 7a. Report Layout Sheet, Example 7.

The rules of format for preparing reports in Tabular Descriptive Language are described in the context of their application to the following forms:

- (1) Report Layout sheet, Figure 7a
- (2) File Description forms, Figures 7b, d, and e
- (3) Decision Tables, Figure 7c.

Example 7 involves the compilation and printing of a Monthly Expense Distribution Report in the system EXPENSE DIST. The collection of records needed to make up the report are indicated in the left hand margin of the Report Layout sheet. In this case, the records LEDGER TOT and DEPT DETAIL are part of the file DEPT EXP REPT, which represents a complete report description.

Information at the very top of the layout is filled in by the analyst:

SYSTEM NAME ANALYST (name) DATE (the sheet is prepared) PAGE (number of number)

MONTHLY EXPENSE DISTRIBUTION REPORT, DEPT NO, ACCT NO, INVOICE NO, INVOICE DATE, and INVOICE AMOUNT are headings. Constant information that is to be printed in the same location on every page of the report is identified by the record name CONSTANT and so marked in the left hand margin of the Report Layout sheet. Every page will carry this exact same information, printed in the same position,

FILE NAME	SYSTEM	EXPENSE DIS	<u>T.</u> ANALYST	<u>3.P.M.</u>	DATE	11/28/61	PAGEOF	
8 CONSTANT		MONTHLY	EXPENSE D	ISTRIBUTION	REPO	RT REP	DATE PAGE N NNNN NNNN	
<u>8</u>		DEPT	ACCT	INVOICE NO		INVOICE DATE	INVOICE AMOUNT	
2 DEPT DETAIL - V		NNNN	NNNNN	A A - N N N N		ZN ZN NN	22,222.NNC1	2
2 LEDGER TOT-V	ARK	GEN LEDG	SER NN			z,	\$22,222.NNCF	¢.
								a.
				14				
ē.								¥
2 FAID REPART-V					2	LAST	PAGE	
4 CONSTANT						LINI		

Figure 7a. Report Layout Sheet, Example 7.

automatically and without instructions. CONSTANT is prefixed by a number (e.g., 8) to state the number of lines that comprise the record. No File Description form is required, since no supplemental information is needed.

The representative date and page number fields are considered part of this record even though their values will change; the programmer has the responsibility for inserting sequential page numbers and the current date on each page.

Record names (e.g., DEPT DETAIL) are suffixed with a V or F which designate vertical placement of the final printed line on the form. V means that printing may occur at any position on the page, except where the space has been pre-empted by the higher priority CONSTANT. F indicates that the position of the final printed line is fixed at that vertical position on the page. Since the term CONSTANT denotes fixed position, no suffix is needed.

The number preceding record name indicates the number of lines required for that record. Reading from the Report Layout, DEPT DETAIL and LEDGER TOT each require two lines. Here, even though just one line is printed, the blank line will be skipped before the next record is written. The number includes the actual number of printed lines as well as the lines not to be printed.

Any record described by the analyst may be composed entirely of pre-assigned information or some combination of pre-assigned and variable information.

2.42

	SYSTEM	EXPENSE DIS	ST. ANALYST	R.P.M.	DATE I	1/28/61	PAGE_	_0F
ILE NAME								
CONSTANT								
CONSTANT								
		MONTHLY	EXPENSE DIS	TRIBUTION	REPORT	REP	DATE	PAGE
						AAA	AN RNNN	KNNN
		DEPT	ACCT	INVOICE	1	NVOICE	INVO	CE
		NO	NO	NO		DATE	AMOU	NT.
CPT BETAK -V		MMMM	NNNNN	AA-NNNN		IN ZN NN	77 772	NNCR
ALL I AL IME - F		AAAA	AAAAA	an anna			,	ALC: NO.
							4	
LEDGER TOT-V	長王坊	GEN LEDI	GER NN			2	, 222, 222	NNCR
	1.00							
	1							
								- 1
						11 - 67-54-5		
END REPORT-V						LAST	PAGE	
and a state of the state of								
CONSTANT	-							
CONSTANT								

Figure 7a. Report Layout Sheet, Example 7.

1.3

The record END REPORT is an example of pre-assigned information. This is a record consisting of two lines, one of which is blank and the other contains the words LAST PAGE. The LAST PAGE designation will print at least five lines from the bottom (four CONSTANT). On the Report Layout sheet, the instruction in the left margin specifies exactly what information will appear and its precise position vertically.

Variable information on a record is specified by indicating where it is to appear within the record with the use of special editing characters. Some of these are described below, but others are given in the reference section of the manual.

CHARACTER	MEANING AND USE
N	An N indicates that the character position will always contain a numeric character.
•	A decimal or period will occur where indicated.
А	An A means that the character position can contain any alphameric character.
Z	The character Z specifies zero suppression of the indicated characters. This is the process of replacing unwanted left-hand zeros by blanks.
3	A comma will be inserted only if the value is large enough to have non-zeros to the left of it.
CR	The symbol CR appears at the right end of the field description and occupies two charac- ter positions. When the value of the field is negative, this symbol will be placed at the right end of the item.

SYS	ETEM NAME]	1	FILE	DES	CRIP	TION				ANALYS	ят 1.		DATE ///20/0	61	PAGE	OF	PAGES
FILE	NAME	DIR	T	PE		MEDIUN	4	P	ECO	DR	D NAME		_	A88	AVERA	NO OF	PEAK	RDS
DE	PT EXP BEPT	C.	T		4	SER- PR	1001		VE	P	DETA	IL		DD	18,5	200	25	000
58R	PIPI D MALAP	HE AL	-	LENT	H O	CCURRI	NCES	N.	12	SHOW	RELATIVE	LOCATION					auc.	
NO	FIELD NAME	A CONTRACTOR	1	MAX	AR M	IN AVE	MAX	202	248	- units	BEGIN	END		VA	LUE/	EMA	RKS	
1	DEPT NO	N	u	4			1			T	1	13						
2	ALCT NO	N	u	5			1/	Γ	Ι	Γ.	1	25						
3	GEN LEOGER NO	N	4	2			1			L	1	22	R	CET NO				
4	SUB LEOGER NO	N	u	3			1				1	25	A	KT NO				4
5	INV NO TYPE	A		2			1				1.	32						
6	ENV NO	N	4	4			1		T		1	37						
7	INV DATE	N	u	6			1	1			1	51						
8	SAV MO	N	u	2	- 2		1				1	45	II	IN DATE				
9	ENV DAY	N	4	2			1	1			1	48	IA	IV DATE		_		
10	ENV YR	N	4	2			1				1	51	TA	IV DATE				
11	INV AMT	N	É	52			1			F	1	64						
-		2	-		-	-	-	+	+-	+-		1		~		-		

Figure 7b. File Description 1, Example 7.

SYS E)	PENSE DIET.]	f	FILE	DE	SCR	M.	ION			E	ANALYS	T PM	DATE ///28/	161	PAGE	OF	PAGES
FILE	NAME	DIR	T	PE		ME	DIUM		R	ECO	R	NAME		ASS	AYER	NO OF	PEAK	RDS
EX	PENSE FILE	7	N	_		SER	- MI	S TAPE	E	X	CE.	NSE DI	ETAIL	EP	18	000	25	1000
980		E.		LENT	н	OCCL	IRREI	NCES	24	12	noas	RELATIVE	LOCATION	·	ILLE /	DEMAN	DVC	
NO	FIELD NAME	LINK.	2	MAX	VAR	MIN	AVG	MAX	8	000	3,2943	BEGIN	END	VA	ALUE/	REIMAN	KK5	
1	DEPT NO	N	u	4				1			3							
2	ACCT NO	N	U	5				1										
З	GEN LEDGER NO	N	u	2			1	1			11	1	2	P.CCT NO	2			
af.	SUB LEDGER NO	N	4	3				1			2	3	5	ACCT NO				
5	INV NO TYPE	A		2				1										
6	INY NO	N	U	4				1									_	
T	INV AMT	N	E	5.2				1										
8	INV DATE	N	U	6				1							-			
9	INY MAG	N	4	2				1				1	2	INV DATI	Ę		_	
10	INY DAY	11	U	2				1				3	4	INV DAT.	E			
11	INV YR	H	4	1				1				5	6	INV DAT	E		-	
~		-	-	-			-		-	-	1				~			

Figure 7d. File Description 2, Example 7.

SYS	STEM NAME PENSE DIST.]		FILE	DE	SCF	M	ION				ANALYS	т У	DATE 11/28/6-1	,	PAGE	OF	PAGES
FILE	RAME	DIR	R T	YPE		ME	DIUM	1	R	EC.	OR	D NAME		ABB	AVERAG	0 05 1	PEAK	05
DE	PTEXP REPT	0	4	7-		56	R-P	RINT	- 1	EL	00	ER To	27	14T	7	0	90	7
SER	FIELD NAME	aradia.c	Sidh	LEN	VAR	OCCI	AVG	MAX	A\$50%	ASSOL.	SEONENCE	RELATIVE	END	VAI	LUE/R	EMAR	KS	
50 20	SEN LEDGER NO	NH	UE	7.2				1				1	22				_	
-		-	-			-			-	-			-			_		

Figure 7e. File Description 3, Example 7.

All of these editing characters are displayed in Figure 7a. For example, the record DEPT DETAIL consists of several fields. The first of these is a four character numeric field, the second a five character numeric field, etc.

A File Description form (Figure 7b for DEPT DETAIL output, 7d for EXPENSE DETAIL input, and 7e for LEDGER TOT) is prepared for each record in the file to report positions with field names. Preassigned information cannot be included through the File Description. For each field, the position of the character located farthest to the right is specified in the END column of RELATIVE LOCATION; the BEGIN column indicates which line of the record contains the field. If there may be confusion, the beginning position may also be shown:

INVOICE NO 1, 34 37

The first field of Figure 7b to be printed is DEPT NO. From the File Description, this field is a numeric, unsigned field of four characters, printed on the first line of the record, with the last character in position 13. Another example is shown with the field INV AMT. This is a numeric field, which can be either positive or negative, and has a length of five integer and two decimal characters. The field is written on the first line of the record and the character farthest to the right is in position 64. On the Report Layout sheet INVOICE AMOUNT has a field length of 11 characters counting commas, periods, and the final CR. This difference in length between the Report Layout sheet and the File

	TABLE	TABLE NAME	TABLE TYP	增											1	FR	RG.	UE	N	CY	-		T									
	SSR	OPER OPERAND 1	OP OPER	AND2	0	P	54	OPS	RAP	ND A	2	0	p>	OR	C	Pa	RAP	102	0	P	° C	PE	RA	ND	2	C	P°	OF	ER	ANG	202	
	-				10	01	01	04 05	06	07	08	09	101	11	12	18	4 11	16	17	18	10	101	At 1	11 1	3 20	25	2.0 1	17 1	1 24	30	31	3.
* *	005	GET INPUT	OPEN		49	1							_[
		EXPENSE FILE	IS EOF	_	N	Y	_	-			_		-	_		-	-				_	-	-	-	-		_	+	-	+	-	-
	2	WRITE	LEDGER T	OT		X							1			+	1							1				1	1		t	
	3	WRITE	END REPO	DRT		X												T														
	4	GOTO	STOP			X												1				1				1.				1		
	5	READ	EXPENSE	FILE	X												T	1					T		1				T	T		
	6	GO TO	REPT PR	EP	X									-	_	1	1	F		_				+				-	+	1	F	_
÷ŧ	DOC	REPT PREP	OPEN		99	1					-		-	-		+	+	+				-	-		+ ·			-	1	+	+	
	1	GEN LEDGER NOS ED	EQ GEN LEDGER	NO: LT	Y	N	_		_		_		-	_	_	-	_	-			-	_	_		-	-		-	-	-	-	_
	2	WRITE	LEDGER TOT	-		×										+	+		F		-	-	+	+				+	+	t	t	
	3	MOVE EXPENSE DETAIL	TO LEDGER T	TO		X																								T		
	4	MOVE EXPENSE DETAIL	TO DEPT DE	TAIL	X	X											Τ											T	T	T	T	
	5	WRITE	DEPT DE	TAU	X	X										T	1	T									Т	Т	T	Т		
	6	SET INLY TOT LEDG	+ INV AMT	ED	X	1																		T							T	
	7	GC TO	GET INF	UT	X	X		-										T	Γ					1		1				T	T	
	-			-		-	~		-				-	-	-	-		+-		-	-	-	-	-	+	1	-	-	-	-	-	-

Figure 7c. Decision Ta	bles, Example 7.
------------------------	------------------

Description sheet (5.2) indicates that editing characters such as commas and decimal points, are not counted in the total length of the field.

The decision tables shown in Figure 7c reveal the operation of writing the report described in the previous text. Rule 1 of table 006 is a typical example of the sequence of operations. The condition of rule 1 requires that the value of the fields GEN LEDGER NO of records ED and LT are identical. When such is the case, the record EXPENSE DETAIL is moved to the record DEPT DETAIL. This is the output record which is to be printed on the output report. The next action of rule 1 is to actually write the lines for DEPT DETAIL. The remainder of the decision table shows the sequence of making up and printing the output record.

QUESTIONS

- 1. What are the requirements for the description of the output records on the File Description forms?
- 2. What are the meanings of the characters Z, A, and N on the Report Layout sheet?
- 3. What is a CONSTANT record?
- 4. On the Report Layout sheet, how is a record identified?
- 5. Read rule 2 of table 005.

	TANLE	TABLE MARAN	TABLE TYPE											I	FF	28	QU	EP	ic.	Ý									Т	T	
	SSR	OPER OFERAND 1	OP OPERAND2	0	P	81	0	ERA	AND	2	0	5	02	6	OP	120	PID	2	OP	01	OPT	RA	ANIE	2	C	P°	10	PER	AN	D 2	
	-		7	01	01	08	04	06 0	6 07	60	09	10	11	12	13	14	19 1	6 1	7 10	10	120	121	23	13 20	35	20	27	28 2	9 3	11	132
窄 冲	005	GET INPUT	<u>LOPEN</u>	49	11				1		1_	_1						1													
	freak	EXPENSE FILE	IS EOF	N	Y			-	-		-	-	_	_		~	-	+	-	1	-		_				-	_	+	-	
	2	WRITE	LEDGER TOT	t	X			+	+			H	-	-		-	+	\dagger	+	+	+		-	+		H	+	+	+	+	
	3	WRITE	FND REPORT		X												1	T	1									T	T	T	
	4	GOTO	STOP	t	X			1								-		1	\top	1	1-						-	1	+	+	+
	5	READ	EXPENSE FILE	X					T									t										+	t		\mathbf{T}
	6	GO TO	REPT PREP	Х				-	-			1				1	+	1	-	F					_	\square	4	+	1	F	
* #	DOC	REPT PREP	OPEN	99	1			-	+				-	\neg	-	-+	+	+	+-	+	-	H				+	-	+		+	\vdash
	1	GEN LEDGER NO: ED	EQ GEN LEDGER NO: LT	Y	N				1					_			_	L	-						-			-			
	2	WRITE	LEDGER TOT	╟	X	-		+	+		H	+		-	+	+	+	+	+	+			+			\vdash	+	+	+	+	t
	3	MOVE EXPENSE DETAIL	TO LEDGER TOT		X											Ť		T	T	1			1				1		Í	1	Π
	4	MOVE EXPENSE DETAIL	TO DEPT DETAIL	X	X											T	1	T	1	\square	\square						1	T		T	Н
	5	WRITE	DEPT DETAIL	X	X													Î	T	T									1	T	
	6	SET INV TOT LEDG	+ INV AMT; ED	X							1							T					T								\square
	7	GGTO	GET INPUT	IX	X				T									T	T										T	T	
	-	CALLER STREET,		-		~	-	-	+	-		-	-	-	-		-	-	-	-	-		-	-	-	-	-	-	-		1

Figure	7c.	Decision	Tables,	Example	7.

ANSWERS

- 1. The output records are described on the File Description forms and should contain all the variable fields of the record, plus a description in the column labelled RELATIVE LOCATION. This description must describe the line of the record in which the field is written, and the position of the right hand character of the field.
- 2. Z, A, and N are used for editing. The characters A and N specify alphameric and numeric data respectively, while the Z indicates zero suppression.
- A CONSTANT record is one which describes those parts of a report which reappear in exactly the same position from page to page with the same information.
- 4. A record name is indicated with a prefix and a suffix in the left column on the first line where its format is described. The prefix indicates the number of lines to the record, and the suffix indicates whether the record is in a fixed vertical position or if it may vary in vertical position.
- 5. If EXPENSE FILE is at end of file, write the ledger total, write the end report, and go to the STOP table.

		SYSTEM NAME	PROCEDURE NAME				D.	av.				1	N	ALY	ST				D	AT	E							
	- 0	BILLING	INVOJCE DETAIL] 0	EC	151	ON	TAS	BLE	E	F	RA	N	c I	\$	_		1	14	1	1			PAGE	0		PAG	NE'S
7.84	<u></u>	TABLE NAME	TABLE TYPE	-	Г	—	T	T	T	П	П	T	T	-	RE	où	N	EV	T	Т	Т	Т		T-		T	T	-
88	2	OPER OPERAND 1	OP OPERAND2	0	P	105	00	ERA	MD	2	0	p *	R.	OP	ER	ND2	1 0	P	0	PER	AN	D	2	OP	-05	ERA	NO	2
F			A 44 1 4 4	-	40	01	04 0	-	6 07	05	09	10 1	1 13	115	14	15 16	17	18	14 1	10 1	1 31	115	14	15 20	17	1 14	30	n
07	4	LNVBICE	OPEN-FJAST	_	1	-		+	+-		\vdash	+	+	+	4	-	1		-	+	+	-				_		
L-	4	DO	NEXT ITEM	X	1		-	+	-			-	-	-	\square	-			-	-	-	1		_	_	_		
	4	SET LOC ADD	EQ ISEM FORMULA	×				+	-			-	+	-		_	1		_	-	-	1		_		-		_
L	4	Do	NEXT INVEN	x				_	1			-				-	1		_	-		-				-		
		SET GTY ON HAND	- OLY SOLD	x				1	1										_	1								
1		MENE PRICE EXT	TO NET PRICE	X								1	1															
	1	NOVE ITEM	TO ITEM LINE	×																								
1_	1	Da	ENVOLCE HEADER	A				1				-																
	1	Go The	071	ĸ																								
L.	1											_		1					-			1						
1	-				-	-	\square	_	+			-	+	-		-	1		-	+	-	-				-		
9		NEXT ITEM	CLOSED	1	99	-	+	+	+	-		+	+	+			+		-+	+	+	+	-			-	+ +	-
-	1	ITEM CARD	IS EGF	Y	N	-		-	-	-	\vdash	+	+	+-	+ +	-	+		-+	+	+	+-				-	+	-
1	-	READ	ITEM CARP	-	X	-	\vdash	+	+	+	\vdash	+	+	+			+	-		+	+-	+-			-	+	-	-
⊢	-	Do-	STO-P	×	-	-	+	+	+	+	+	+	+	+		+	┝	-	+	+	+	+	+		+	-	+	-
5	3	NEXT TUVEN	CLOSED	50	50	1		+	+	1		+	+	t	1	1	t			Ť	+	t	\vdash			+		
P	1	TTEN NA: ITEM	FA ITEA NOT	Y	IN	1		T	1	T				-		T	T					1	T			-		
1	1		MASTER	T	1			1	1				1	1		-	T			1	T		1		11			
r=	=	SET IAC ADD	+ /	╞	ty	-	F	-	-	-			T				T											
T		READ	INVEN MASTER		X									T			T					T	1					
	1		PER LOC ADD		1												T											
F		Do-	013		×												1											
L								-	-			-	1	-		-							1			_		
07	24	INVOICE HEADER	CLOSED	190	10	1		-	+-	1.		-	+	1		-	1	-		-	-	-	1			-	1	
	_	CUST NOV. ITCH	EQ CUST NOT NEADE	4	N	-		-	-	-		+	+	+	1	-	+		_	-+	+	+	-			-	+	-
		WEITE	ITEM LINE	X							r	-	-	-						-	-	1				-	-	-
	1	DØ.	TOTAL CALL		X	1		+	+	1.		-	+	-	1	_	1	-		1	-	+	1		-	-	1	-
	1	DO-	PROCEDURE		×	-		+	-	1		+	+	+			1	-	-+	+	+-	+	1	+ +	1.1	-	1	-
	1		HEADER PREP					_	-	1_		1	-	-		-	1			-	-	-		1		-	-	
1		00-	074		X							1								_								1
	RE	EMARKS:						_										_	_		_							

Figure 8a. Decision Table, Example 8.

SYS	STEM NAME St. L IM G]	I	FILE	DE	SCR	M	ION			[FRAA	T /CTS	DATE 1/4/	62	PAGE	OF	PAGES
FILE	NAME	DIR	T	rpe		ME	DIUN	100	R	SCC.	-	NAME		A28	AVAR	NO 05	LPCAR	200
SER	FIELD NAME	- ALTUN	N.ON	LEN	TH	OCC	AVE	MAN	1 13904	Atros	SEAGENCE	RELATIVE	LOCATION		VALUE/	REMA	RKS	
12345	CARP CODE CUST NOT ITEM NOT OTY SOLD UNIT PRICE	2222	レリリナナ	156342				1 1 1 1			2							

Figure 8b. File Description 1, Example 8.

SYS	TEM NAME BILLING]	1	FILE	DE	SCR	M	ION				ANALYS E.R.	AHCSS	DATE	4 /6	2	DAGE	OF	PAGES
FILS	NALVE VEN MASTER	DIA	1 1	IPE IA JA	ED	RA		DESK	8	ECO	A	A STE	K.	ABI	6	19.	30 GH	PEAK 12	000
SHA HO	FIELD NAME	International States	81.5N	MAX	TH	OCC	AVG	MAX	ASSOC	TO	STUDIOSE	RELATIVE	END		VAL	LUE/F	EMA	RKS	
1 23 75	EFEM NO CLASS OF ITEM STOCK LOC DESCRIPTION GTY ON MAND	ARRA A	+	63 63 3	~														

Figure 8	. File	Description	2,	Example	8.
----------	--------	-------------	----	---------	----

Example 8 demonstrates a number of approaches for reading and writing records in files and explains the various ways of correcting files, depending on their classification. The procedure outlined in this example is part of the system BILLING and includes (1) reading item cards in a file, (2) getting an inventory master, and (3) proceeding with invoice preparation.

The first action in the open table 071 shown in Figure 8a is:

DO NEXT ITEM

which is the title of table 072. Control is temporarily transferred to this table through the DO operator.

Table 071 differs from earlier samples in that no conditions have been written into its structure. Whenever this happens, actions are executed upon entry from another table, and the arrangement is referred to as an <u>unconditional table</u>.

In closed table 072, the first action:

ITEM CARD IS EOF

tests whether the item card file is at the end of the file. This action is used only with a special kind of file, one in which records appear one after the other as in a deck of punched cards or a magnetic tape. When a file is to be treated in this way, it is known as a <u>serial file</u>.

A detailed summary of the serial file ITEM CARD is supplied in the File Description form, Figure 8b. Across the header essential

1	SYSTEM NAME	PROCE	DURE N	AME			1.0	W.			A	NALY	ST			DA	TE			_			
-	BILLING	INYO	ICE DE	TAIL	DE	CIS	ION	TAB	LE	[FRA	NCI	5		1	14	162		5	MGE	OF	PAG	SE!
100	TABLE NAME	TAB	LE TYPE	r I	T	T	TT	T	П		TT	F	RIOL	EN	cv	T	TT	T	TT	TT	11	-	r
SR	OPER OPERAND I	OP	OPERA	NDZ	OF	101	OP	ERAI	ND 2		OP 6	OP	ERAND	2 0	OP SI	OP	ERA	ND 2	ż t	000	OPE	PANO	2
	Provide the party	1 0.00	1.0		@1 4	10 50	5 04 0	4 04	87 0	1	09 10 11	12 18	14 18	6 17	18	4 110	AT	12 85	11 1	S 10	17 18	14 30	T
74	LNYEACE	SPEI	N-1-1	RST	-	-	++	+	+	+	-+-+-	++		+	++		++	-	1	++		-	-
+	Do	NEXT	ITEN	1	X	+	++	+	\vdash	-	++		++	+	++	-	++	-	++	++		-	-
+	SET LOC ADD	EQ II	EMFor	90LA	×	+	++	+	\vdash	-	++	++-	-+-+	-+-	++	-	+-+	+-	++	++	+		
-+	Do	NEX	TIN	EN	x	+	+		+	-	-+-+-			+-	+-+	-	+	+	++	++			-
+	SET GTY ON HAND	- a	rr 50	10	x	+	++	-	+	-+		4	++	+			44	-	1	++		-	-
	NEVE PRICE EXT	To	NETP	KICE	x	+	++	-	\vdash	-	++			-	++	-		-	++	++		-	1
	YOVE ITEM	To ;	LIEM L	INE	×	+	++	-	++	-		++	-+-+	+	++	_	+	-	++	-			-
-+	DC	ENVa	CE HE	ADER	A	-	++	-	+	-				-	+ +		+		\downarrow	++			-
	Go Th	07	1		×	+	++	-	+	-+	-+			+		_	\downarrow	-		++	_	-	
-+					-	+	++	-	+	-		++-		-	++		++					-	ļ.,
-+		L			-	+	++	-	+			++	11	-		-	1	-		++		-	L
12	NEXT ITEM	CLO	SED		19	9	11	-		-				-	+-+	-	++	-	1	+-		+	1
-	ITEM CARD	35	EGF	_	71	N.	1	-		_				+	++		++	+-		++		-	1-
-	READ	IT	M CAP	0	1	K	++	-	++	-		++		+	++				1		+	-	1
-	Do-	STO	- P		×	-	+ +	-	\vdash	-	++	+ +		-		1	+ +	-				-	Ļ.
-	and a set of the second state of the second s				-	+		-	\square	-				+	+-+		+ +		1			-	1
25	NEXT INVEN	CLO	SED		50	0		-	\vdash	-	-++-			_			++	_	\downarrow	-	-	-	1
_	ITEM NO: ITEM	EAI	EN NO	_	YI	N.		-	\downarrow	-				-	11	1	\downarrow	-		11		-	1
1		MA	STER		1	-		-	\downarrow	_	-			-	1	_	++	_	\downarrow			-	L
	SET LOC ADD	+ 1			T	×		-	\square					-	\downarrow	-		-	1				1
4	READ	INVE	W MA.	SYER		×			\vdash	_	++			-		-	+ +	-	\downarrow				1
1		PER	LOG A	PA	_	_		-	+	_		++		-			1-1	-	\downarrow			_	1
_	20-	01	3		3	4	+	-	\vdash	_				-	1	-	1	-	1		_	_	1
-	and a second				_	-		-	\downarrow	-		1		-	1	-	1	-	11		_	_	1
74	INVOICE HEADER	C6	DSED		901	0	++	-	1					+	+	-		-	+ +			-	∔
-	CUSENER. ITCH	FAGL	IST NO:	HEADS	14	1		-	+	_		++-		+	++		++	+	++	-		-	-
-	KELTE	ITTEN	1LIN.	E	X	1	1							-	+ +	-	1	-	\downarrow			-	Ł
_	DØ	TOTA	1L CA	44	1	X	11	-	\downarrow	_	-++	++-		1	+ +	+	11		14		_		1
_	DO	PROC	EDURE	-	-	×	++	-	\downarrow	-	++	+ +-		+	+ +	-+	+ +		+ +		4		ł
1		HEAD	ER PR	EP				-	\square					-		-	\downarrow	_	\downarrow	-	-	-	1
1	0.0	07	4			×		_						_		_				_			1
RE	MARKS:				_					_													_
		Fiq	Jure	8a.	E)ec	isi	on	Т	21	ole,	Exa	amp	le	8.				_				-
SY:	STEM NAME] ,	FILE D	ESC	RIF	TI	ON			Ľ	FRA.	ST	5	D	1/9	1/1	.2		P	AGE	OF	PA	Gi
SY	STEM NAME] ,	FILE D	ESCR	RIF	TI	ON	1		Ľ	FRA	ST	5	D	1/ 9	124	.2		P NO	05	OF	PA	Gi
FILE	STEM NAME 8-4 4 IN G		FILE D	ESCI	DIU	N	ON	125	sco	HD	FRA.	st NS I	5		1/ 9 A31	1 Z 4 8	2	AVIL	NO	05	OF	PA RD1	G
FILE	STEM NAME 844 IN G NAME TEM SARP	DIR T	FILE D	ESC:			ON	128	sco T	HD	HAME	ST NCI	5		1/ 9 A31		- 2	AVIL	NO SC.	05 05	OF RECO	PA RD1	G
FILE	STEM NAME SEALING NAME TEM CARP FIELD NAME	D K S sien	FILE D			TI NI CEN	ON ARD CES MAX	125	CO I	H P Stacsact	HAME	ST ALC T ELOCAT	5 (10N D		17 9 A21	1 Z 4	- 2 /AL	AVIL 52 UE/	RE	OS D MAR	OF LPGAR	PA RDS	
FILE	STEM NAME	D R R Norman	FILE D	ESC:			ON ARD CES MAX	120 15900	CO In Anno	H P RRUSNEL	HAME	ST NC J ELOCAT	5 (10N D		17 9 A31	1	- 2 /AL	AVOU 52 UE/	RE	os p MAP	OF LPGAN 6	PA RDS	
FILE	STEM NAME SALING NAME TEM CARP FIELD NAME CARP CERE	DR H men U	FILE D				ON ARD CES MAX	125	CO IN ALOC	HD Careson 1	HAME	ST A'S J ELOCAT	5 (104 D		1/ 9 A21	• / • •	/AL	AVAU &	RE	AGE OF MAR	OF RECO PRAN	PA RDI 000	
FILL I	STEM NAME STEM NAME TEM CARP FIELD NAME CARP CERE CUST NG	DR H sten UU	FILE D			TI NI C	ON ARD CES MAN /	125	CO IS Assoc	HD P Russell Z	ANALY FRA NAME 1-1 RELATIV BEGIN	ST A'C J ELOCAT	5 (10N D		1/ 9 A21	× / 4	/AL	AVAU 52 UE/	RE	AGE OF MAR	OF RECO LPGAN C	PA RDS 000	
FILE I	STEM NAME STALING NAME TEM CARP FIELD NAME CARP CORE CUST NOT ITEN NOT	DIR T N san U U	FILE D			TI NI EN	ON ARD CES MAX /	125905	CO IN Atroc	HD G Russell Z	ANALY FRA NAME 1-1 RELATIV BEGIN	ST NSE ELOCAT	5 (101) D		A31 A31	4 / 4 B	/AL	AVOU 52 UE/	RE	AGE OF MAR	OF RECO POAR KS	PA 803	
FILLI EN 1234	STEM NAME STALING NAME TEM CARP FIELD NAME CARP CORE CUST NOT ITEN NOT QTY SOLD	DIR T X Year U U U V	FILE D			DTI M - C	ON ARD CES MAX / /	125900	A A A A A A A A A A A A A A A A A A A	H P Russell V	ANALY FRA NAME 1-1 RELATIV BEGIN	ST MS & ELOCAL	5 (10)1 9		A21	× / 4	/AL	AVOU S	RE	AGE OF MAR	OF RECO LPGAN C	PA 805 000	
	22 22	SYSTEM NAME BILLING BILLING TABLE NAME IN OPERADD 1 THE VOICE SET LOCADD DO SET LOCADD DO SET LOCADD MENE PAICE EXT NOVE LIEM DC GGT THE MEXT ITEM IFM CORD READ DO READ SET LOCADD READ DC NEXT INVEN ITEM NO: ITEM SET LOCADD READ DC NOVELEE HEADER CUST NO. ITEM WILLE DC READ DC READ DC DC READ DC READ DC READ DC DC READ DC READ DC READ DC DC DC READ	SYSTEM NAME PROCE BILLING INVO I TABLE NAME TAB COPER OPERAND 1 OP TO INVOICE DPER DOC NEXT SET LOC ADD GA IT DOC NEXT SET ATY ON HAND - Q NEVE RAIGE EXT TO NOVE JIEM TO GO TH OT COMP ALENT JIEM CLO IFM NB: ITEM CLO IFM NB: ITEM CLO ITEM NB: ITEM CLO INVOIC ADD FI READ INVO DOC QT MINIORICE HEADER CLA SUST NOR ITEM EQ IN NOVE JIEM SCOLOR DOC QT REMARKS: FIG	SYSTEM NAME PROCEDURE N BILLING INVOICE DE TABLE NAME TABLE TYPE DOBE OPERAND 1 OP OPERA TABLE NAME OPERAND 1 OP OPERA DO OPERA DO OPERA DO NEXT ITEN SET LOC ADD GA LICHFOR DO NEXT INV SET ATY ON HAND - QIY SO NEVE LICH TO JET M NOVE LICH TO JET M NOVE LICH TO JET M NOVE LICH OT ON ALL DO STOP DO SIGP 23 NEXT INVEN CLOSED ITEM NO: ITEM GA ITEN NA SET LOC ADD F NEXT INVEN CLOSED ITEM NO: ITEM CLOSED DO OT 3 NEXT INVEN CLOSED DO TO TAL CA DO TAL CA DO TO TAL CA DO TAL CALONER PARA DO TO TAL CA	SYSTEM NAME PROCEDURE NAME BILLING INVOICE DETAIL TABLE NAME TABLE TYPE BORE OPBRAND 1 OP OPERAND2 TO INVOICE BRAND 1 OP OPERAND2 TO INVOICE DETAIND DO NEXT INVEM SET LOC ADD GA LIEM FORMULA DO NEXT INVEM SET ATY ON HAND - QIY SOLD MENE PRICE EXT TO NET PRICE MOVE JIEM TO JIEM LIME DO ENVOICE HEADER GA TH OT ICH CLOSED IFEM CORD 3S EGF ACAD JIEM CARD DO SIGP 23 NEXT INVEM CLOSED ITEM NO: ITEM EQ ITEM NO MASTER SET LOC ADD F I READ INVEN MASTER DE QT 3 MINOPIESE HEADER CLOSED DO OT 3 MINOPIESE HEADER CLOSED LIEM NO: ITEM EQ ITEM NO MASTER SET LOC ADD F I READ INVEN MASTER DE QT 3 MINOPIESE HEADER CLOSED CUST NOT TOTAL CALL DO TOTAL CALL DO TOTAL CALL DO TOTAL CALL DO TY REMARKS: FIGURE 88.	SYSTEM NAME PROCEDURE NAME BILLING INVOICE DETAIL DE BILLING INVOICE DETAIL DE MILLE NAME TABLE TYPE BORGE OPBRANDI OP OPERANDZ OF TABLE NAME TABLE TYPE BORGE OPBRANDI OP OPERANDZ OF TABLE NAME TABLE TYPE DOT NEXT ITEM X SET LOC ADD FAILENFORMUM X DO NEXT INVEN X SET ATY ON HAD - OIT SOLD X MOVE ITEM TO ITEM LIME X DO NEXT INVEN TO ITEM LIME X DO LIFM OF ATEM CLOSED IS IFEM CORD IS EGF YI READ ITEM CLOSED IS ITEM NO: ITEM CLOSED IS INVENTINGEN CLOSED IS ITEM NO: ITEM CLOSED IS INVEN MASTER DO OT SIGN P X MEXT INVEN CLOSED IS ITEM NO: ITEM CLOSED IS INVEN MASTER DO OT SIGN P X MASTER INVEN MASTER DO OT SIGN P X MASTER INVEN MASTER DO OT SIGN P X MEXT INVEN CLOSED IS INVEN MASTER DO OT SIGN P X MASTER INVEN MASTER DO TOTAL CALSED IS INVEN MASTER X DO TOTAL CALSED IS REMARKS: FIGURE 88. D	SVSTEM NAME PROCEDURE NAME BILLING INVOICE DETAIL DECIS TABLE NAME TABLE TYPE IN OPBR OPBRAND 1 OP OPERAND 2 OP TO INVOICE DETAIL OP TO INVOICE DETAIL OP TO INVOICE DETAIL OP SET LOC ADD GO LIEM FORMULLX SET LOC ADD GO LIEM FORMULLX SET LOC ADD GO LIEM FORMULLX SET ATY ON HAND - QIY SOLD X MENT PRICE EXT TO MET PRICE X MOVE I TEM TO SIGH LIME X DO INVOICE HEADER X GO TH IFM CLOSED 199 IFM SERD IS EOF Y N READ SIGH VIEN CLOSED 199 SET LOC ADD FORM X DO SET LOC ADD INTEN CARD X DO SIGP X NEXT INVEN CLOSED 55 SO STEM NO: SIGH FORM Y N MASSER SET LOC ADD FO SIGH NASSER NEXT INVEN CLOSED 55 SO STEM NO: SIGH FO NASSER NEXT INVEN CLOSED 55 SO STEM NO: SIGH SON MASSER NEXT INVEN CLOSED 55 SO STEM NO: SIGH SON MASSER NEXT INVER CLOSED 70 N NEXT INVEN CLOSED 70 N NASSER 70 N NEXT INVEN CLOSED 70 N NASSER 70 N NEXT INVEN CLOSED 70 N N NEXT INVEN CLOSED 70 N N NEXT INVEN CLOSED 70 N N N N N N N N N N N N N N	SYSTEM NAME PROCEDURE NAME IB BILLING INVOICE DETAIL DECISION MITABLE NAME TABLE TYPE DECISION MITABLE NAME OPBRANDI OP OPERANDI OP OPERANDI DC MEXI I TEM OP OPERANDI OP OPERANDI DC MEXI I TEM KITTEM KITTEM SET ACCADD FAILES EXT TO NET PROVA KITEM DC MEXI INVEN CLOSED KITEM DC ENGLIES SICP XITEM DC SICP XITEM KITEM DC SICP XIE XIE SICP SICP XIE XIE	SYSTEM NAME PROCEDURE NAME BILLING INVOICE DETAIL DECISION TABLE TABLE NAME DOBRANDI OP OPERANDI OP OPERANDI OP OPERANDI OP DCT NEXTIENT NOT NEXTIENT NOT NEXTIENT DCT NEXTIENT NOT NEXTIENT NOT NEXTIENT DCT NEXTIENT NOT NEXTIENT NOT NEXTIENT DCT INVALUE DCT INVALUE NEXTIENT TO NET CARP NEXTIENT CLOSED ITEM NB: ITEM CARP NEXTIENT SIGP NEXTIENT	SYSTEM NAME PROCEDURE NAME IBM BILLING INVOISE DETAIL DECISION TABLE MARKE TABLE MAME TABLE TYPE DECISION TABLE MORRANDIO OP OPERANDI OP OPERANDI OP MORRANDIO OP OPERANDIO OP OPERANDIO OP OPERANDIO MORIA SET LOC MEXIITEM TABLE TYPE I I SET ACA ADD CA SIGE CA DD K I I MOVELLIEM TITEM CLOSED 199 I I I MEXII INVEN CLOSED 199 I I I MEXIII INVEN CLOSED SIGE SIGE I I MEXIII INVEN CLOSED SIGE <td>SYSTEM NAME PROCEDURE NAME IBM BILLING INVOICE DETAIL DECISION TABLE BIDGER OPERANDI OP OPERANDI OP OPERANDI OP OPERANDI Important of the state of the state</td> <td>SYSTEM NAME PROCEDURE NAME IBM A BILLLING INVOICE DETAIL DECISION TABLE A BIDDELSION TABLE NAME TABLE NAME TABLE TYPE DECISION TABLE A BIDDELSION DEFENDE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DECISION TABLE A BIDDELSION DEFENDE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE A INDEXT INTERACT MEXT INTERACT X INDEXT INTERACT X</td> <td>SYSTEM NAME PROCEDURE NAME IBM. ANALY BILLING INKGISS DETAIL DECISION TABLE FRANCI Status OPERANDI OP OPERANDZ OP OPERANDZ OP OP OP Status OPERANDI OP OPERANDZ OP <thop< th=""> OP OP</thop<></td> <td>SYSTEM NAME PROCEDURE NAME IBM ANALYST BILLING INNOISE DETAIL DECISION TABLE ANALYST STABLE NAME TABLE NAME TABLE TYPE Image: State of the state of</td> <td>SYSTEM NAME PROCEDURE NAME ISBM ANALYST BILLING INVOICE DETAIL DECISION TABLE ANALYST BILLING ANAME TABLE NAME TABLE TYPE Image: State of the stat</td> <td>SYSTEM NAME PROCEDURE NAME JJDM. ANALYST BILLING INVRICE DETAIL DECISION TABLE FRANCIS / STABLE NAME TABLE TYPE FRANCISS / / / STABLE NAME OPBRAND 1 OP OPBRAND 2 OPBRAND 2 OP OPBRAND 2 OPBR</td> <td>SYSTEM NAME PROCEDURE NAME ISM. ANALYST DX STLLING INVROICE DETAIL DECISION TABLE FRANCISS I/// STLLING INVROICE DETAIL DECISION TABLE FRANCISS I//// STLLING OPBRANDI OP OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY</td> <td>SYSTEM NAME PROCEDURE NAME ANALYST DATE BLLLING INNOTE CETAL DECISION TABLE FRANCES I////C2 International Internation International Internation International Inte</td> <td>SYSTEM NAME PROCEDURE NAME IBM ANALYST DATE BILLSING EXACTS CETAL DECISION TABLE FRAVETS 1/x/62 BILLSING TABLE TABLE TYPE Decision Table FRAVETS 1/x/62 BILDENA OPERANDI OP OPERANDI OP OPERANDI DP OPERANDI DP OPERANDI IN DOR OPERANDI OP OPERANDI OP OPERANDI OP OPERANDIC OP TABLE PREVISION IN DOR MEXTINE BILLSING OPERANDIC OP TABLE OPERANDIC OP TABLE INTERVISION IN DOR MEXTINE BILLSING ANALYST DATE INTERVISION INTERVISION DOR MEXTINE OPERANDIC OP OPERANDIC OPERANDIC INTERVISION DOR MEXTINE ANALYST INTERVISION X INTERVISION INTERVISION DOR MEXTINE OPERANDIC OPERANDIC INTERVISION INTERVISION INTERVISION DOR MEXTINE TAGE EXTINE TAGE EXTINE INTERVISION INTERVISION INTERVISION INTERVISION INCONSTRUCT MEXTINE<</td> <td>SVETEM NAMEIBMANALYETDATESTALLINGINVOICE DETAILDECISION TABLE<math>FRANCISDATESTABLE NAMETABLE TYPEDECISION TABLE<math>FRANCISDATESTABLE NAMETABLE TYPEDECISION TABLE<math>FRANCISDATESTABLE NAMETABLE TYPEDECISION TABLE<math>FRANCISDATESTABLE NAMETABLE TYPEDECISION TABLE$FRANCISDATESTABLE NAMEOPBERANDIOPOPBERANDIDECISION TABLESTALLING ACCESPENTERSPENTERSPENTERSPENTERDCMGRIT ITEMKSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTDOTALE STATEXSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STEPXSPENTERSPENTERDCDOTALE STEMSPENTERSPENTER<$</math></math></math></math></td> <td>Image: State NamePROCEDURE NAMEImage: State TypeState TypeTrades TypeDecision Table$FRAVETS$$I/V/4s$Trades NameTrades TypeDecision Table$FRAVETS$$I/V/4s$Trades NameTrades TypeDecision Table$FRAVETS$$I/V/4s$Trades NameDecision Table$PRENVERS$$I/V/4s$ProcessorTrades NameDecision Table$PRENVERS$$I/V/4s$ProcessorTrades NameDecision Table$PRENVERS$$I/V/4s$ProcessorTrades NameDecision Table$PRENVERS$$I/V/4s$$I/V/4s$Trades NameDecision Table$PRENVERS$$I/V/4s$$I/V/4s$Trades NameDecision Table$PRENVERS$$I/V/4s$$I/V/4s$Decision TablePRENVERS$I/V/4s$$I/V/4s$$I/V/4s$SETLOC ADDPRENVERS$I/V/4s$$I/V/4s$Decision TablePRENVERS$I/V/4s$$I/V/4s$SETLOC ADDPRENVERS$I/V/4s$Decision TableReference$I/V/4s$SETLOC ADD$I/V/4s$Decision Table$I/V/4s$Decision Table$I/V/4s$SETLOC ADD$I/V/4s$Decision Table$I/V/4s$Decision Table$I/V/4s$Decision Table$I/V/4s$Decision Table$I/V/4s$</td> <td>SYSTEM NAME PROCEDURE NAME DBM BILLING INVOICE CETAL Decision TABLE FRANCIS DATE BITALING TABLE TING TABLE TING Decision TABLE FRANCIS DATE BITABLE NAME TABLE TING TABLE TING Decision TABLE PREQUENCY DATE BITABLE NAME TABLE TING OP OPERANDI OP OPERANDI OP DC OPERANDI OP OPERANDI OP OPERANDI OP OPERANDI OP DC MERIT TEM TABLE TING ALTER TABLE TABLE TING OP OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI<td>IBM IBM BYSTEM NAME Investe Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table</td></td>	SYSTEM NAME PROCEDURE NAME IBM BILLING INVOICE DETAIL DECISION TABLE BIDGER OPERANDI OP OPERANDI OP OPERANDI OP OPERANDI Important of the state	SYSTEM NAME PROCEDURE NAME IBM A BILLLING INVOICE DETAIL DECISION TABLE A BIDDELSION TABLE NAME TABLE NAME TABLE TYPE DECISION TABLE A BIDDELSION DEFENDE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DECISION TABLE A BIDDELSION DEFENDE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE OPBRAND 1 OP OPBRAND 2 OP OPBRAND 2 OP DE BIDDELSION TABLE A INDEXT INTERACT MEXT INTERACT X INDEXT INTERACT X	SYSTEM NAME PROCEDURE NAME IBM. ANALY BILLING INKGISS DETAIL DECISION TABLE FRANCI Status OPERANDI OP OPERANDZ OP OPERANDZ OP OP OP Status OPERANDI OP OPERANDZ OP OP <thop< th=""> OP OP</thop<>	SYSTEM NAME PROCEDURE NAME IBM ANALYST BILLING INNOISE DETAIL DECISION TABLE ANALYST STABLE NAME TABLE NAME TABLE TYPE Image: State of the state of	SYSTEM NAME PROCEDURE NAME ISBM ANALYST BILLING INVOICE DETAIL DECISION TABLE ANALYST BILLING ANAME TABLE NAME TABLE TYPE Image: State of the stat	SYSTEM NAME PROCEDURE NAME JJDM. ANALYST BILLING INVRICE DETAIL DECISION TABLE FRANCIS / STABLE NAME TABLE TYPE FRANCISS / / / STABLE NAME OPBRAND 1 OP OPBRAND 2 OPBRAND 2 OP OPBRAND 2 OPBR	SYSTEM NAME PROCEDURE NAME ISM. ANALYST DX STLLING INVROICE DETAIL DECISION TABLE FRANCISS I/// STLLING INVROICE DETAIL DECISION TABLE FRANCISS I//// STLLING OPBRANDI OP OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY OPARITY	SYSTEM NAME PROCEDURE NAME ANALYST DATE BLLLING INNOTE CETAL DECISION TABLE FRANCES I////C2 International Internation International Internation International Inte	SYSTEM NAME PROCEDURE NAME IBM ANALYST DATE BILLSING EXACTS CETAL DECISION TABLE FRAVETS 1/x/62 BILLSING TABLE TABLE TYPE Decision Table FRAVETS 1/x/62 BILDENA OPERANDI OP OPERANDI OP OPERANDI DP OPERANDI DP OPERANDI IN DOR OPERANDI OP OPERANDI OP OPERANDI OP OPERANDIC OP TABLE PREVISION IN DOR MEXTINE BILLSING OPERANDIC OP TABLE OPERANDIC OP TABLE INTERVISION IN DOR MEXTINE BILLSING ANALYST DATE INTERVISION INTERVISION DOR MEXTINE OPERANDIC OP OPERANDIC OPERANDIC INTERVISION DOR MEXTINE ANALYST INTERVISION X INTERVISION INTERVISION DOR MEXTINE OPERANDIC OPERANDIC INTERVISION INTERVISION INTERVISION DOR MEXTINE TAGE EXTINE TAGE EXTINE INTERVISION INTERVISION INTERVISION INTERVISION INCONSTRUCT MEXTINE<	SVETEM NAMEIBMANALYETDATESTALLINGINVOICE DETAILDECISION TABLE $FRANCISDATESTABLE NAMETABLE TYPEDECISION TABLEFRANCISDATESTABLE NAMETABLE TYPEDECISION TABLEFRANCISDATESTABLE NAMETABLE TYPEDECISION TABLEFRANCISDATESTABLE NAMETABLE TYPEDECISION TABLEFRANCISDATESTABLE NAMEOPBERANDIOPOPBERANDIDECISION TABLESTALLING ACCESPENTERSPENTERSPENTERSPENTERDCMGRIT ITEMKSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCMGRIT ITEM FAMMUAXSPENTERSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTDOTALE STATEXSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STATESPENTERSPENTERSPENTERDCDOTALE STEPXSPENTERSPENTERDCDOTALE STEMSPENTERSPENTER<$	Image: State NamePROCEDURE NAMEImage: State TypeState TypeTrades TypeDecision Table $FRAVETS$ $I/V/4s$ Trades NameTrades TypeDecision Table $FRAVETS$ $I/V/4s$ Trades NameTrades TypeDecision Table $FRAVETS$ $I/V/4s$ Trades NameDecision Table $PRENVERS$ $I/V/4s$ ProcessorTrades NameDecision Table $PRENVERS$ $I/V/4s$ ProcessorTrades NameDecision Table $PRENVERS$ $I/V/4s$ ProcessorTrades NameDecision Table $PRENVERS$ $I/V/4s$ $I/V/4s$ Trades NameDecision Table $PRENVERS$ $I/V/4s$ $I/V/4s$ Trades NameDecision Table $PRENVERS$ $I/V/4s$ $I/V/4s$ Decision TablePRENVERS $I/V/4s$ $I/V/4s$ $I/V/4s$ SETLOC ADDPRENVERS $I/V/4s$ $I/V/4s$ Decision TablePRENVERS $I/V/4s$ $I/V/4s$ SETLOC ADDPRENVERS $I/V/4s$ Decision TableReference $I/V/4s$ SETLOC ADD $I/V/4s$ Decision Table $I/V/4s$ Decision Table $I/V/4s$ SETLOC ADD $I/V/4s$ Decision Table $I/V/4s$ Decision Table $I/V/4s$ Decision Table $I/V/4s$ Decision Table $I/V/4s$	SYSTEM NAME PROCEDURE NAME DBM BILLING INVOICE CETAL Decision TABLE FRANCIS DATE BITALING TABLE TING TABLE TING Decision TABLE FRANCIS DATE BITABLE NAME TABLE TING TABLE TING Decision TABLE PREQUENCY DATE BITABLE NAME TABLE TING OP OPERANDI OP OPERANDI OP DC OPERANDI OP OPERANDI OP OPERANDI OP OPERANDI OP DC MERIT TEM TABLE TING ALTER TABLE TABLE TING OP OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI OPERANDI <td>IBM IBM BYSTEM NAME Investe Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table</td>	IBM IBM BYSTEM NAME Investe Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table BY TABLE NAME Table TYPE Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table Image: Status of Original Decision Table

Figure 8b. File Description 1, Example 8.

SYS	TEM NAME]	1	FILE	DE	SCR	M	ION			-	MALYS ER.	T	DATE 1/4/	6 2	DAGE	OF	PAGES
FILS	NAME VVEN MASTER	DIR	1 1	VPE FA IA	PED	ME		DESK	2	<u>sc</u>		ASTE	e	A86	19.	80 GM	PECO	000
58 R HO	FIELD NAME	ALPIA A	8181	MAX	TH	OCC.	AVG	MAX	A5906	10 N4101	SCHEPOSE	RELATIVE	LOCATION	V	ALUE/F	REMA	RKS	
1 23 7	STEM NOT CLASS OF ITEM STOCK LOG DESCRIPTION	NARA	4	63 65	~			1										
5	GTY ON MANE	N	+	6	-	-	-	1		-					~		~	4

Figure 8c. File Description 2, Example 8.

descriptive data concerning the file is entered:

File Name	-	ITEM CARD
Dir. Type	-	INPUT
Medium	-	SER-CARD
Record Name	-	ITEM
		etc.

The EOF (END OF FILE) notation looks ahead to see whether there are any more records to process before reaching the end of the file. Y signifies there is no next record; if this condition is not checked at this point, and further operations are called for (e.g., READ ITEM CARD), they will be ignored if the end of file has been reached. An N denotes there is at least one more record.

Table 072, rule 2 calls for the execution of: READ ITEM CARD which arranges for the storage in memory of values for the 5 fields in the record as they are described in Figure 8b. When the EOF is reached, the DO STOP action provides for the completion of certain closing routines.

The operator READ is always used for providing input from a serial file. The operand is always a file name, and the next record in the file is read into memory.

Control now returns to table 071 to get the inventory record for the item number on the item card. In this cycle from the initial DO action in table 071, through the steps in table 072 and back to 071 again, table 071 assumes the role of master coordinator for several closed tables. Basically, it calls on these tables for a series of actions, then asks for

L		SYSTEM NAME	PROCEDURE NAME			-	1.15		ę.	[A	NA	LYS	T			_	DA	TE							
L		BILLING	INVOJCE DETAIL] 0	EC	ISI	ON	TAI	BLE		F	R.A	N C	IS			_	11	41	63	-		PAGE	4	36	Pia	GES
F	444.5	TABLE NAME	TABLE TYPE	-	Г	Г	TT	T	T		T	T	Т	FR	EO	UE	NC	N	T	ГТ	Т	T	1	Т	TT	T	T
1	HR	OPER OPERAND I	OP OPERAND2	0	10	-	05	16 Rd	DIN	2	OF	1 61	£	OPE	RAN	ID2	OF	45	OPI	RA	ND	2	02	66	Pas	AN	22
E	-		and the set of the set	-	42	01	194		6 07	05	09 1	0 111	12	15 1	4 15	16	17 1	0 10	110	A1	12 4	5 14	1.4 1.4	187	115	\$ 30	TR
P	Tre	INVERCE	B-PEN-FIRST		1											1											
	_	DO	NEXT ITEM	X														_						1			
		SET LOC ADD	ER ITEM FORMULA	×			11																	1	11		
L		Do	NEXT INVEN	X																							
L		SET GTY ON HAND	- QTY SOLD	X				1							1						T	T		T		1	T
ſ		NEVE PRICE EXT	TO NET PRICE	x								1	1		T			Т			Т			T	TT	1	
Г	1	NOVE ITEM	TO ITEM LINE	×								T						1			T	1	IT	T	IT		
ſ	1	DO	ENVOICE HEADER	A																T				1	TT	1	
ſ	1	Go TN	071	K	T	1							T					1	1					T		1	T
ľ	Ì				1	1									1			-	1						T		T
ſ					1								T		1						T			1			1
I,	017	NEXT ITEM	CLOSED	1	99	1						T	Т		1				T	1				T	TT	1	T
	1	ITEM CHRD	35 EB.F	7	N	1													1		1						
ľ	1	READ	ITTEN CARP		X.					1			_					-			_			_			
-	-1	Do-	STO-P	×	-	+	++	+	+			+	-	+-+	+	-	-	+	+	++	+	+	++	+-	\vdash		+
	713	NEXT INVEN	CLOSED	50	50			+	1					ti	1	1		+	1			1		1			
ľ	1	ITEM NO: ITEM	FA ITEM NUT	Y	N	1										T			1			T		T			1
ľ	1	The set of the bad and the Barton and the set of the se	MASTER		1	1			T				T						1					1			T
r	1	SET LAC ADD	+ /		1×	-		-	-	-			1	TT													
Г	- 1	READ	JNVEN MASVER		X														1								1
Г	1		PER LOG ADD		1																			1			
-	-	DO	013		×	-		-	+	-		-	-		-	1		+	-		-	-		-		-	-
	071	INVOICE HEADER	CLOSED	90	10		H	+	+			+	t		-			1	t		+	+	H	1		1	1
ſ		EUST NOV. ITCH	EQ CUST NO: NEADO	-1	N	1			T									T									
F	-	WEITE	ITEM LINE	X	F	-		1		-	-	T							T								
ſ	1	DØ-	TOTAL CALL		X																						
ľ	1	DO-	PROCEDURE		×							1							1								
ſ			HEADER PREP																								
۲	-	0/9-	074		X	1	TT	1							1	1			T				1	T			1

Figure 8a. Decision Table, Example 8.

SYS	STEM NAME SLLING]	F	ILE	DES	CR	NI IPT	ION			E	FRAA	T (C 7 5	DATE 1/4/6	.2	P4:58	OF	PAGES
FILE	NAME TEM CARP	018	TN	PR		MED	L-C	ARD	R	sco T)H(HAME		A38	AVILLA	NO 05	LPCAR	200
SER	FIELD NAME	PERMIT	10.1	LEM	TH C	DCCU HIN	AVE	MAN	15906	10 No.	12457083	RELATIVE	LOCATION END	V	ALUE/I	REMA	RKS	
12345	CHRP CODE CUST NOT ITEM NOT OTY SOLD UNIT PRICE	NNNN NN	540++	- 5-6 3 2							1							

Figure 8b. File Description 1, Example 8.

545	TEM NAME]	ş	FILE	DE	SCR	M	ION				AHALYS F.R.	T MACS S	DATE	62	PAGE	OF	PAGE
FILS	LE NAME INVEN MASTER		17	ide La Ia	150	RA	DIUM	PESK	8	ECO	130	ASTE	<i>K</i> .	884	19.	ND 08 55 0:00	PECOI PEAK	000
58 R HO	FIELD NAME	Which are a straight	\$15X	MAX	TH	OCC	AVG	HAX	YOBBY	TO	SCHROOSE	RELATIVE	END	1	ALUE/	REMAI	RKS	
- 25	EFEM NO CLASS OF ITEM STOCK LOC DESCRIPTION GTY ON KAND	NARRA	4	1973 4 5 · 1	~													

Figure 8c. File Description 2, Example 8.

return of control to issue further instructions. The cycle may be repeated as often as necessary, depending on the complexity of the procedure.

To get the inventory record for the item number on the item card, the file INVEN MASTER is required. An examination of its File Description form in Figure 8c reveals that this file is on disk and is a retained random disk file. As such, it requires different treatment.

A <u>retained file</u> is one which is used over and over in a system; it exists as an input at some time during the processing cycle and still exists at the end of that cycle.

In a <u>random file</u>, as differentiated from a serial file, reading and writing are performed on the basis of location or information content. A random file has no beginning or end, and each record is viewed as it turns up in the loop. New data may be added to a record while it is in memory, but this new information is superimposed on the previously recorded data, so the older data is lost unless explicitly recorded elsewhere. This provides a way to correct the file automatically.

For the current example, information will be obtained by address. The field LOC ADD is given a value from an arithmetic expression called ITEM FORMULA.

The DO NEXT INVEN action calls for table 073; the first condition: ITEM NO: ITEM EQ ITEM NO: MASTER

2.49

-	SYSTEM NAME	PROCEDURE NAME			-		W.			-	-	N	ALY	ST	8	DATE //4/62				- J								
-	DILLING	INVOICE DETAIL	J	DEC	151	ON	TA			FI	RA	AV I	c I	s		L	1	1.	1	62			944	5E	OF	F	AG	E f
TABLE	TABLE NAME	TABLE TYPE	T	T		TT	T	T		T	T	Т	F	RE	QUI	N	EV	Т	Т	Т	T	T		T	-1-	T	T	-
SER	OPER OPERAND I	OP OPERANDE	1	90	34	OP	ERA	MO	2	OP	0		OP	ERA	NDZ	10	Pa	0	PE	RAN	ab	2	0	000	00	TDA	MEN	2
07	TALWEICE	D-DEN-E-AST	-	42	SR	04 4	-	6 07	95	09 10	011	1 12	115	14	18 16	17	10	101	201	41 2	2 25	114	2.8	20)1	7 1	120	301	
11	DIT	UPIEN FARST		+	-	-+-	+	+-	\square		+	+	+	-+	+	-	+	-	-+-	-	+	-	14	_	1		_	
	C	NEXT STEM	4	-	-	-+-	+	+		-	+	+	-	$\left + \right $	-	-	++		+	-	+	-	-+		_	1	-	
-	SET AGE AUD	ER LIEM FORMULA	×	+		-+	+	+	\square	-	+	+	+		-	-	+	-	_		-			-	-		_	
	DO	NEXT INVEN	×	-		-	-	+		-	+	-	-		-	+	-+	-+		+	-	1		-	_		_	
-	SET WIY ON HAND	- arr solo	Ă	-			+	+		-	4	1	1	\vdash		Ļ			-	-	-	1		1	_			į,
<u> </u>	NEVE PAICE EXT	TO NET PRICE	X	-			-	+		-	+	+	+		-		1.1	-	_	-	-							
	HOVE ITEM	TO ITEM LINE	×	-			+	-		-	+	+	1	1	-	1	\square	_	-	_				-				Ĵ
	Da	ENVOLCE HEADER	*	-			-	-		-	-	-	+_		-			4	-	-	-	1		_	-			
-	Gg Thy	071	×				-	-		-	_	-	1		_				1	_				1				ĺ
				-	-		+	+		+	-	+	-			-		+	+	+	+	-		-	-	$\left \right $	-	
012	NEXT J.TEM	CLOSED	7	99			+			-		+		\vdash	+			+	+	+	+		•	+	+	H	-	
	ITEM CHR.D	35 EGF	4	IN			T						\top		-	t-	1-1	1		t	T	1	-+	1	+		-	1
	READ	ITEM CARD	-	X			-	1	-									1	1	1	1	1		-	1-		1	Ì
	Do-	STO-P	×								T							1		1							1	-
178	NEXT TUVEN	CLOSED	K.	6	-	+	+	+		+	+	+	+	+	-	-	+	-	+	+	+	+	\vdash	-	+	\square	+	2
ALL .	TTELS ALS. ITENS	ED ITEA MAY	1	N			+				+	+		+	+	-	+			+					+	-		
	1	ALASTER	+	1		+	+	+-		-	+	+-	-	t t	-	H	+	+	+	+	+	-	+	+	+	+ +	-+	
	SET IAL AND	- MARINE	-				+	+		+	+	+	+	++		\vdash		+	+	+	+		\vdash	+	+	+	+	1
	READ	IN VEN MASTER	-	12	- 1	-	+	1		-	+	+	+	$\left(+ \right)$	-	t		+	-	+	+	-	+	-+	+	+-+		
	LAST F. A.G.	DER LAC ADD		1	-	-1-	1			-	+	+-	1		-1-	t-		+	-	-	+	+	H	-+	+	++		-
	Do	013		×			1			1	T		1		1			1				E						
074	INVOTEDE LEANED	CLOSED	90	10			+	+		-	+	1	-	-	+-	-		+	-	+	+	+	\vdash	-		-	-	
-	CUSE NOV TTEN	EQ CUST NA MEMO	4	N			1	1-1		+	+	+	1		-	1-	1-1	+	-+-	+	+-	1		+	-			1
	U.LTTE	TTEMITALE	V	F	-	-	+	-			+	1	1		-	t	+	+	-+	+	+			+	+	\mathbf{H}	-	i
	DØ	TOTAL CALL	1	X	-	1	+			-	+	1	1		-	-	++	Ŧ	-	-	+	+	-+	-+	+		+	1
	De	PROCEDURE	-	x	-		+	\mathbf{H}			t	+	+		-	1		+	+	+	+	+		-+-	+	+	-	
	- ME-MC-	HEADER PREP	1	1	-		+	H		+	t	1	1			1	1	- 1	+		+-	+		-+	+-	-		-
	00	074	-	×			+	+		-	+	-		-	+			+	-	-	-	+	H	+	+	1	+	
25	MARKS		-				-	-			-	-	-			-	-	-	-	_	-	-	-	-		-	1	

Figure 8a. Decision Table, Example 8.

SYS	STEM NAME S-L L IN G-]	ł	FILE	DE	IB.	M	ION			E	FRAM	IT ICTS	DAT	E 14/62		PAGE	OF	PAGES
FILE	NAME VEM CARD	DIR 3	N	/Pr		ME	DIUM	ARD	R	acc J	040 7	NAME		-	28	AVARA	40 05 1 4	PEAK	200
SER	FIELD NAME	N.SWAX	101	MAX	TH	MIN	ANG	MAX		ALAOC YC	SEQUENCE	RELATIVE	LOCATION		VAL	UE/R	EMAR	RKS	
12345	CHRP CODE CUST NOT ITEM NOT QTY SOLD UNIT PRICE	N N N N N	いひしたや	1 57 6 39 2				 			2								

Figure 8b. File Description 1, Example 8.

SYS	etem name BILLING]	1	FILE	DE	IB.	M UPT	ION				AHALYS E.R.	ANCS S	DATE	62	PAGE	OF	PAGES
FILS	NAME WEN MASTER	DIR	2. TY	IPE	1ED	RA	DIUM	DESK	9	ECO	DRO	ASTE	K	A88	AMILE 10	NO 68 88	PEAK	000
SER HO	FIELD NAME	ALBUA	RS:4	MAX	TH	OCC	AVG	MAX	ASSOC	ASBOL TO	SECONENCE	RELATIVE	LOCATION		VALUE /	REMAI	RKS	
1 23 75	FIELD NAME FIELD NAME CLASS OF ITER STOCK LOG DESCRIPTION QTY ON HANG		4	63 65 2	2			1 1 1										

Figure 8c. File Description 2, Example 8.

This is unlikely to be true the first time through, unless the same item number shows up twice in a row for a single customer; the N path (rule 2) will probably be satisfied first. In rule 2, LOC ADD is incremented by 1, then the action READ INVEN MASTER PER LOC ADD is executed. Table 073 is repeated again. When the Y condition is satisfied, control returns to table 071. Incrementing of LOC ADD is simply one way of chaining if the first address is incorrect.

The READ operator for a random file is somewhat different in construction than in a serial file:

Operand 2 Connector Location Name READ _____ PER _____

Here the operator obtains the record at the location. If no record is there, the READ is not executed.

By definition, a random retained file is automatically updated with modifications at the same location.

In table 071, a number of actions now take place:

- 1. Quantity on hand is decremented (reduced) by the quantity sold.
- 2. Price extension (an arithmetic formula name) is moved to net price.

3. The record item is moved to the record item line.

After these actions, a DO INVOICE HEADER appears, which is once more the signal to call on a closed table, this time 074, for execution.

	SYSTEM NAME	PROCEDURE NAME			1	R	Ņ			1	AN	AL	YS'	r		_	_	D	AT	E	_		-		-			
-	DILLING	INVOICE DETAI	50	EC	ISI	DN T	ABI	LE	1	RI	94	C J	rs		_		1	14	4	1	_	-	12A	6E	0	NF	Pla	NGR
TAN	TABLE NAME	TABLE TYPE	T	1			Π		T		T		RI	Q	JE	NC	Y	T	T	T	T	T		17			Т	Т
\$ 88	OPER OPERAND 1	OP OPERAND	0	P	-	OPE	RAN	10 2	0	P	84.	0	PER	AN	20	0	, 61	Q.	18	AN	10	2	0	P	0	PE	MAS	0:
07	TNWBSCE	A-PEN-ETACT	-	400	108	04 86	-	07 00	09	10 1	11 1	1 11	10	15	16	17 1		0 2	0 4	1 21	2 85	14	18	10	\$7	15	19 3	0 8
	Der	NEXT TYEN		+	+		+	+	+	+	+	+	+-	-	+	+	+	+	+	+	+	+		-	-	+	+	÷
	SFT LOC ADD	GO TTEM Friddel		+	+	-	H	-	+	+	+	+	+	H	+	+	-+-	+	+	+	+	+-	-		-	-+	+	+
	DA:	NEVT TAVEN	Ť	+	+	-	+	-	+	+	+	+	+		-	+	+	+	+	+	+	+	-	\vdash	-		+	-
	SET OTY AN HAWD	- DEX SALD	T.	1	+	+		-	+	++	+	+	+	1	-	+	+	+	-	+	+	+	-	-	-	-	+	+
-	WANTE PATAS EVE	To VIES DATE	÷÷	+	+	+	H	+	+		-	+	+	+-+	+	+	+		+	+-	+	+			-	-	+	+
	MANE TTEM	TO TOULTUS	- A	+	+		+	+	+	+	+	+	+	-	-	+	+	+	+	+	+-	+		\vdash	-	+	+	+
-	DOT LIGH	FALLES AFARE	Ê	+-	+-+	+	+	+	+	+	+	+	+-	-		+	+	+	+	+	+	+		-	-	\vdash	+	+
	C = Te	071		1	+ +	+		-	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	-	-	+	+	+
-	1 (39 /37		-	+	+ +		-	+	+-	+	+	+	+	1-	-	+	+	+	+	+	+	+	-	+	-			+
			-	+	+	-		+	+	+	+	+	+	1		+	+	-	+	+	+	+	1-	-	-		+	+
	ALC T T T CLA	CLASED		00	+	+		-	+-	++	+	+	+	-		+	+	+	+	+	+	+	-	-		\vdash	+	+
911	NEXI LIEN	TELOSCO	ť	177		+		+	+	+	+	+	+	+-	-	+	-+	+	+	+	+	+-	+	-	-	++	+	+
100	ALEFT CARD	42 697	17	C		-		-	-	+	+	+	+-	+		+	+	+	+	+	+-	+-	+	-	-	\vdash	+	+
-	105.00	LIEM CARP	1	1×	+	+		-	+	+	+	+	+-	+	-	-	+	+	+	+	+-	+	+	-	-	+	+	+
-	100-	2100	1	-	++		+	-	+	++	+	+-		+-		-	+	+	+	+	-	+	1-		-	-	-	+
-			1	-	+	-	+	-	+	++	+	+	+	-	-	-	-	+	+	+	+	+	+	-	-		+	+
11	WENT INVEN	CLOSED	130	150	-		-		-	++	-	-	+	-		-+			-	+		+		-	-		+	
	TTEM NO: ITEM	EQ ITEM NOT	HY	N	-	+	+	-	+	+	+	+	+-	-	-	-	+	+	+	+	-	+-	1	-		-	+	+
_		MASTER		1		-	-		-	++	+	+	+-	-	-	-	-	-	+	+	-	+	-	-	-		-	+
-	LSET LOC ADD	± /	+	X	-			-	+-	\vdash	-	-	-	-	-	-	+	+	+	-	+	-	+	-	-		+	+
-	BEAD	JNVEN MASTER	-	X	-		++	-	+-	++	+-	+	+-	+	-	-	-	+	+	+	+	+	1	1	-		-	+
-		PER LOC ADD	#	+				-	+	$\left \right $	+	-	-	-		-	-+	-	+	-	-	+	1	-	-		-	-
	Do	013	-	×	+	_	-	-	+	\square		-	-	-	_	_	+	+	-	-	1	1	1	_	-		-	-+
_			1	1		-			-	\square	i	-	-	-	_		-		+	_	+	1		-	1		-	-
2	INVOICE HEADER	CLOSED	190	10		_		-		\square	-	-	+	1			-	-	+		-	1		1			_	4
	CUST NOV. ITCH	EQ CUST NO: NEM	54	N	1	_			-	1	+	+	1				-	-	-	+	1	-	1	1	-		-	
-	KILITE	ITTEM LINE	X			1			Ť				1						1		1		1					
	00	TOTAL CALL		X		1			1.		1	1											1				_	
	DO-	PROCEDURE		×															I	1			1		1.0			1
		HEADER PREP	1																1				T					
	0.0			14	a 1	_				- T	_	_	_						_	_		_	-A			-		
_	100	074		10					_	1	_		1	L.									Ĺ				Í	1
R	EMARKS:	074							1_		-	_		L.					1				1				1	
R	YSTEM NAME	Figure 8a.	BA		ec:	isio	on	T	2.b)	.e,) F	Ex	ar	nŗ			3. TE						1					1
R 5	YSTEM NAME B 54 4 IM G	Figure 8a.	BA	Da	ec:	isio	on	T	2b)	e,	F	Ex	ar	nŗ). TE		6	2			PA	6.8		OF	A	AG
R	YSTEM NAME B 54 4 IM G	Figure 8a.	BA RI		ec:	isio	on	T	2b)	e,	F YS IA	Ex r	ar	np). TE		6	2			Pa	6.8		OF	A	AG
RST	YSTEM NAME BS44 IN G	Figure 8a.	BA CR1	Da		isio	on	T		e,	F YST LAV	Ex	ar	nŗ). E			2	AVA	N	DA4		Rie	OF	A	AG
R 55 1 1	I DOT EMARKS: YSTEM NAME BS44 IM G LE NAME I TEM SARP	Figure 8a.	BA CR1 MEDI			isio DN R0	on	T		e,	J YST IM	Ex r	a1	mp			3. 19	Z	6		AVAU	No. 51	PA4	30	RE	OF CO CAN 6	A	40
R 5 1	I DOT EMARKS: YSTEM NAME BS44 IM G LE NAME I TEM CARD	Figure 8a.				isio N Rp	on Res			IAL:	I IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	T	ar				3. 19		6	2	AVA	No. St.	PA	30	REL	OF CO CAN	A	AG
R S FIL	YSTEM NAME BS44 IM G LE NAME ITEM CARP	Figure 8a.	BA CRI AEDI			ISIO N RQ		Tro Contraction of the second	AAA AAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA		I I I	EX T C	ar 5	mp). E			2	AVAL 52	No. SI	PA	56 30 1A1	RK	OF CO CAN S	A	AG
R S FI	YSTEM NAME SILL IM G LE NAME ITEM CARD FIELD NAME	Figure 8a. File DESC			TIC CA ENC	ISIO N ES	2500 A3500	Tro	AM AM RE BI	.e,	YS	EX T C	ar). TE				AYOU SY	No. SI	PAU DO DO D	5.8 9.6 1.4 h	RK	OF CON S	A 802	40
R S F E	YSTEM NAME SILL IN G LE NAME ITEM CARP FIELD NAME I CARP CORE	Figure 8a. File DESC				ISION RD ES AX	ncon.	CO Pri Anne	AAA (D N (CEA)	IAL:	I I I I I I I I I I I I I I I I I I I	T C L OC EI	a1				3. 12 12 12 12 12 12 12 12 12 12 12 12 12		<u>6</u>	2	AXOU SY JE/	NRI RI	PAGE DE N	68 05	R	OF COAR S	A	49
R S T F	I DOT EMARKS: YSTEM NAME BELL IN G LE NAME I TEM CARP FIELD NAME I CARP CODE CUST NOT	Figure 8a. File Desc			TIC	ISION RD ES AX 1	on.	Tro	AM REA REA	IAL	I I	EX T C L OCL EI	a1				3. 721				AVOLU 52 JE/	No. Sec.	Page 1	68 0F	RK	OF CON CON CON	A 802	
R S T FI SES TRA	YSTEM NAME BELL IN G LE NAME ITEM CARP FIELD NAME CARP CERE CUST NOT S ITEM NOT	Figure 8a. File Desc OIR TYPE XN V V V V V V V V V V V V V V V V V V				ISION RD ES AX ////	on.	Tro	AAA BI	IAL	I VEI	EX T C	ar 5				3. 121			2	AXMU Sy JE/	NR I	PAV 500 EN	658 05	R	OF COM S	A 803	AG
R 5 1 FI 1 1 1 2 1 4	YSTEM NAME BELLING LE NAME ITEM CARP R FIELD NAME CARP SERE CUST NOT S ITEM NOT CARP SERE	Figure 8a.			CA ENCC	isi0 DN <u>Rp</u> ES AX ///////////////////////////////////	on.	CO In Action	AM REA BI		YS YS	T C L O C L	ar 5				3. 20		<u></u>	2	AXMU SY JE/	S I	PA4	6.8 05	RK	OF CON S	A 803	
R S T FILLEN 12145	YSTEM NAME BELLING ITEM CARD FIELD NAME CARD SEDE CLIST NOT S ITEM NOT GITY SOLD UNIT PRICE	Figure 8a. File Desc OIR TYPE XN FILE DESC OIR TYPE XN V XN XN V XN XN XN XN XN XN XN XN XN XN			TIC SIM	ISIO N Rp ES AX 1 1 1 1	on.		AAA REE BI	IAL:	YS'	T C L O C L O C					3. 121		6 VA	2	ANDU SY JE/	N	PA4	68 05	RK	OF CO SAM	A 01	
R 5 1 FI 1 1 1 1 2 1 4 5	YSTEM NAME BELLING LE NAME ITEM CARD R FIELD NAME CARD CARD CARD CARD CUST NO S ITEM NO CUST NO CUST NO CUST NO CUST NO CUST NO CUST NO CUST NO CUST NO CUST NO	Figure 8a.				ISICON RD ES AX ///////////////////////////////////	on.	Tro	AAA RE BI	e, AAM	YS'		ar 55				3. 20		6	2	AXOU 52 JE/	Ser /R	PACEN EN	658 05	R	OF COAL S	A 802	
R 5 1 1 1 1 2 1 1 2 1 1 2 1 1 2 1 2 1 2 1	YSTEM NAME BELL IN G LE NAME I TEM CARP R FIELD NAME CARP CARP CARP CARP CUST NOT CUST NOT	Figure 8a. File Desc OIR TYPE XN FILE DESC MAX YAR NC Y Y Y Y Y Y Y Y Y Y Y Y Y				ISION RQ ES AX ///////////////////////////////////	on Res SCI		AM	e,	VEIN 1,		ar sar				3. 20				AND	New Sector	DAG DAG	6.8 0F	RK	OF COME	A 01	
R S T FI 140 121 45	YSTEM NAME BELLING	Figure 8a. FILE DESC OIR TYPE TRN FILE DESC Figure 8b. FILE DESC				N N N N	on asses		AM	e, ALY ALY ALY	YS VEI	EX					3. 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.	3.	6 ····································		ANDU SY	New Sector	EN	658 05 1A1	RK	OF CONS S	A 01	
	YSTEM NAME BELLING	Figure 8a. FILE DESC OIR TYPE XN FILE DESC MAX VAR NC / NC				ISION Rp ES AN ///////////////////////////////////	on assessed	Tr		ALY ALLY	YS VEI	EX T C EI					3. 24	3.	4 · · · · · · · · · · · · · · · · · · ·		ANDU SY JE/		EN	GAE AL	RK	OF COAL S	PA PA	
	YSTEM MAME BELLING STEM NAME ITEM CARP R FIELD NAME CARP CEDE CLUST NOT SITEM NAME BILLING S NAME	Figure 8a. FILE DESC OIR TYPE XN FILE DESC MAX VAR NC / NC				ISION Rp ES AN ///////////////////////////////////		Tr		ALY AND	F N N N N N N N N N N N N N N N N N N N	EX C					2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1	3.	5 VA		AXOU SY		EN	658 0F 4A1	RK	OF COR S	Pa P	
	YSTEM MAME BELLING STEM NAME ITEM CARP R FIELD NAME CARP CEDE CLUST NOT SITEM NAME BILLING SNAME CARP MASTER	Figure 8a. FILE DESC OUR TYPE XN FILE DESC OUR TYPE XN Figure 8b. FILE DESC DUR TYPE MR TYPE TYPE MR TYPE MR TYPE TYPE TYPE TYPE T				ISION Rp ES AN // // // // // //		Tr		e, RAD ALY ALY	I I,	EX CA EI					3. 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.1 2.	3.	5		AND US		EN	658 0F 4A1	RK	OF CON S	P4	
	YSTEM MAME BELLING STEM NAME ITEM CARP R FIELD NAME CARP CEDE CLUST NOT SITEM NAME BILLING SNAME TALEN MASTER	Figure 8a. FILE DESC OUR TYPE TN FILE DESC OUR TYPE MAX VAR NC NC NC NC NC NC NC NC NC NC				ISION Rp ES AN // // // // // //		Tree of the second seco	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	e, ALY ALY ALY	I,						2.8 E E	3.			AVAU SY JE/		PACE PO	6.8 0F	RK	OF COA S	P4	

Figure 8c. File Description 2, Example 8.

+-+

1 1

1 1

1 ____

1 EFEM NOT NH 6 2 CLASS OF ITEM A 3 3 STOCK LOG A 6 4 DESCRIPTION A 25 5 GTY ON MANE N+ 6

25 1

N+ 6

1

This table checks to see if the customer number on the item record is the same as the customer number on the header. If they are, then an item line is written (ITEM LINE is a record name on a serial file which is appropriately described on a Record Layout sheet and a File Description form). If they are not equal, then these actions are executed:

DO TOTAL CALC

DO PROCEDURE HEADER PREP

DO 074 (i.e., repeat this table)

The next time through the table the condition will be satisfied and the action WRITE ITEM LINE is executed.

Control then reverts to table 071 to continue the master cycle. The WRITE actions, designating output, are essentially the same in application whether for random or serial files except on random files a location must be specified.

The DO PROCEDURE HEADER PREP action brings up a <u>procedure</u> <u>name</u> for the first time. A procedure is one or more decision tables which together form a distinct and logical segment of an information processing system. The procedure name must start with an alphameric character and be unique. In this series of tables, for example, the procedure name is INVOICE DETAIL. The first table in a procedure must be marked as FIRST under TABLE TYPE as in table 071.

-	1 12 1 2 1 2	PROCEDURE NAME			-	10	av.			L		AN	AL	YS.	r		_	_	D	AT	E	_	_	-		_		_	
	DILLING	INVOICE DETAI	1	EC	151	ON	TA	BLE	E	F	RI	₹ ₩	c.	rs	_	_	-	1	14	16	62		_	19A	6E	0	36	Pa	G
10	TABLE NAME	TABLE TYPE	T	Г	Γ	IT	T	T	Г	Г	Π	Т		RI	iQ	UE	NO	Y	T	T	T	T	T	Γ	T	Г	T	T	Т
SIR	OPER OPERAND 1	OP OPERAND	2 (P	294	0	-	AND	2	0	P	60	0	PER	AN	SQ	0	3 92	OP	ER	AA	D I	2	C	20	0.0	PE	RAN	0
	Talwartet	A-PEAL - E - AC-		40	108	94	04 0	16 07	að.	04	10	11 1	211	5 14	18	16	17 1	0	4 1	0 A	(a	2 25	14	4.4	64	187	19	20 0	0
07/	2440454	OFEN FIRST	-	+	+-	++	+	+-	+	+	+	+	+	+-	-	-	+	+	+	+	+	+	+	+	-		\square	+	÷
-	00	NEXT STEM	-	+	+	-	+	+	+	+	+	+	+	+	-	\vdash	+	-+-	+		+-	+	+	-		-	+	+	+
-	SET LOC ADD	FA LIEM FORMU	4 ×	+	-	-+	+	+	+	+	+	+	+	+-	-	-+	+	+	+		+	+	+	-		-		+	+
-	Do	NEXT INVEN	X	+	-	\vdash	+	+-	+	+	\vdash	+	+	+-	-		-	-	+		+	+	+	-	-		+	-	4
_	SET GTY ON HAND	- ary solo	X	-	-		+	+-	+-	+	\square	1	+	-	-		+	+	+-	+	+	-	-		-		\square	-	+
	NEVE PRICE EXT	TO NET PRIC	FX	-	1		+	-	-			+	_	-			-	+	-	-	-	-	1	1				-	1
_	HOVE ITEM	TO ITEM LINA	X	-	-		_		1		\square	_		_			1	_	_	_	_	_		1				1	1
_	DO	INVOICE HEADE	e A							1			_	_				-	_				1						1
	Go Ty	071	K		-		_	-	1	1				1					1	_									1
				1											1				1.										1
																					1	T		1.					Ι
017	NEXT ITEM	CLOSED	11	199	1					1					Ī				1	T	T	T		1		1	TÌ		1
-	ITEM CARD	35 EGF	14	IN	1		1		1									1		T	1	T	1	1	1	1			1
at read	READ	ITEM CARD	1	X	-	H		1	1	1				1						T	1	1	1	1	-				1
-	Do	STOP	×	1			1		1			1	-	-	1			+	1	T	1	1	1	1	1	1	H	+	1
-	- ME V		1	1	1	++	+	+	1			+	+	+	-		+	+	T	+	+	1	1	1	1	1-	H	-	1
ALC:N	NEWT TUNEN	CLASED	10	te	1	H	+	-	1	1	1	+	+	1-	1		-	+	-	Ť	+	1	1	1	+-	1	11	+	1
XII.	TECH ALD TTCH	GO TICI IN	15	12	1	1	-	+	+	+	+	+	-			-	-		-	+	+		+	-	ŕ-	1	1-1	+	-
	LANSIN NOT ALEM	LOASTER	-	11	+-	+	-	+	+	1	+	+		1-	1			+	+	+	+	+	+-	1-	1	1-	+	+	+
-	51	MASIRE	-	-	-	++	+	-	-	-	++	+	+	+-	1		+	+	+	+	+	+	+	+	-	+	++	+	1
-	PET LOC ADD	T I I I I I I I I I I I I I I I I I I I		1×	-	++	+	+	+	+-	+	+	+	+-	+-	\vdash	-	+	+	+	+	+	-	+	+	-	i l	-	+
	04AQ	Dea The MASTER	1	1×	1-	+	+	+	+	+	++	+	+	+	+	-	-	+	+	+	+	+	+	1	+	-	+-		+
-		PER LOG ADD	+	1	+	+	+	+-	+	+	++	-	+	-	+-		-	-	+	+	+		+	+-	+	+-	+ +		-
	26-	013	-	×	+-	+	+	+	+-	+			+	+	+		-	+	+	+	+	+	-	+	-	-		-	-
			1	+	-		-	+	+	-		+		-	+		+	+	-	+	-	+	-	-	+	-	-	-	-
074	INVOICOE HEADER	CLOSED	190	10		\vdash	-	+-	+	-	-	-	-	+	-		-	+	+	+	-	-	+-	-	-	-	-	-	-
-	CUSY NER. TTEM	EQ GUST NO: NER	1	N	-		-	-	-	1	++	-	+		+	-		+	-	+	+		-	+-	-	+-		-	-
_	Kis I. TE	ITEM LINE	X	1	-	1	-	-	1	1		_	-	1	-		-	-	+	+	-	+	+	-	-	+-	-	-	-
	.0.0	TOTAL CALL	1	X	1		-	+	+	1		4	-	1	-		-	-	1	+	-	+	+-		-			-	-
	Der	PROCEDURE	-	×	-	1	-	+	+	+-	+ +		+	-	-		-+	-	-+	+	+	+-	+	+		-	+ -	-	-
-	ALL ACTION AND ADDRESS AND ADDRESS	A	11	1		11				1.1	r 1		_	-	-			-						1					
-		HEADER PREP	-	1						+-	+ +	_	-	1.00								+	+-	+	-	+	+	-	
R	D.O- EMARKS:	NEADER PREP 074	1	×				1	1	1_		1	_	_						1.			1	1		1			1
R	STEM NAME	Figure 8a	BA		30	is	ion	n	T'a	b)	.e,	J J	- X T	ar	nŗ			3. TE		1				1		1			
R	STEM NAME	Figure 8a	BA	D	3C	ist	ion	n	I Te	[] []	.e,	j YS	T	an	nŗ			3. TE		<u> </u>	2		-	PA	68		OF	A	~
R	STEM NAME	Figure 8a	BA CRI		9C	is:	ion		T'e	[b]	.e,	1	T	a1	nŗ). TE		<u> </u>	2			PA	64	R.5	OF	A	×
R	STEM NAME 8-4.4 IM G 8 NAME	Figure 8a	BA CRI	D	3 3 5 7 11 7	ist	ion	1	L Te		.e,	J YS IA	T		nŗ			E 28			2	AVAL	N. C.	P/4	68	Rip	OF	A	~
R	STEM NAME STEM NAME SALLING E NAME TEM CARP	Figure 8a	CRI MED SEI	DX P		is	ion	250	Ta Ino Ino	L)	.e,] YS }A/	T T T	a1				28 28		<u> </u>	2	AVAL	5	PA co	68	R	OF GAN 6	A	
R ST	STEM NAME STEM NAME STEM NAME STALLING E NAME TEM CARP	Figure 8a Figure 8a FILE DES				IS: ON ARD		n	I Second H D	LD]	ALLATI	YS IA	T	an ario ND	mř.			28		4	2	AXAL SY JE/	Si Si	PA	68 05	RK	OF COAR C	A	
R ST SEA	STEM NAME STEM NAME SEALING E NAME TEM CARP FIELD NAME	Figure 8a Figure 8a FILE DES		DX P IUN RAI		is: ON NRD			Therease of the second		ALLATI	I NEI	T	(a)	nk.			288 128		4		AVORE ST.	I I I I I I I I I I I I I I I I I I I	PA	68 0F	RK	OF GAA	A	
R ST FIL	STEM NAME EMARKS: STEM NAME SIGALING E NAME TEM CARP FIELD NAME CARP CERE CUST NOT	Figure 8a Figure 8a FILE DES DIR TYPE				IS ON RECON			H Instant	D N CEA	.e,	I YS	E 22	(a)				28		6		AXAL 47 JE/	S i	PA co EN	68 05	R	OF CO CO CO CO CO CO CO CO CO CO CO CO CO	A A D S	
R ST FIL SEAD	STEM NAME EMARKS: STEM NAME SILLING E NAME TEM CARP FIELD NAME CARP CORE CUST NOT	Figure 8a File DES DIR TYPE XN KU S NU S			C A	IS ON		1	I I I I I I I I I I I I I I I I I I I		IAL	J YS IA NEI	T T LOC.	(2.1 (2.1) (nk			28		4	2	ANDRE /		PA	668 0F	R	OF GAME (S	A 000	
R 57 FIL 5800 1234	STEM NAME EMARKS: STEM NAME STALING E NAME TEM CARP FIELD NAME CARP CORE CUST NG ITEM NO DIV SOLD	Figure 8a File DES OR TYPE 3N NU 1 NU 1 NU 5 NU 5 NU 5 NU 5			TIC U	IS: ON ARD		250	N ISSUE TO ST	D AN CEA	.e, Re, LAT	YS YS NE N	T	(2) (2) (2) (2) (2) (2) (2) (2) (2) (2)				28			2	AXAL ST	S i	PA co EN	6.8 05	RK	OF 6 (S	A	
R 57 FIL 580 12345	STEM NAME EMARKS: STEM NAME SALLING E NAME TEM CARP FIELD NAME CARP CORE CUST NOT LITELY NOT LITELY NOT DITY SOLD (MIT DRICE	Figure 8a File DES DIR TYPE 3N FILE DES DIR TYPE 3N N U / MAX VAR M N U / N U				IS: ON RES AAN		1	I I I I I I I I I I I I I I I I I I I	AAA REE BI	LATI	I VS						28		6	2	AXAL 4/		PA co EN	68 05	RK	OF CO CO CO CO CO CO CO CO CO CO CO CO CO	A 8.05	
R 5 1 FIL 7 5ENO 12345	STEM NAME EMARKS: STEM NAME SALLING E NAME TEM CARP FIELD NAME CLUST NOT LITEM NOT LITEM NOT LITEM NOT LITEM SOLD UNIT PRICE	Figure 8a Figure 8a FILE DES DIR TYPE 3N FILE DES				IS ON ARD			Instant of the second s	AN PERFECT	I AM	I YS		(a)				28		6	2	AXAU SY JE/	S ··	PA	68 0F	R	OF CO CO CO CO CO CO CO CO CO CO CO CO CO	A A D S 0 O	
R 5 7 FIL 7 50 1 21 3 4 5 1	STEM NAME EMARKS: STEM NAME SALLING E NAME TEM CARD FIELD NAME CUST NG ITEN NG UNIT PRICE	Figure 8a File Des DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES FILE DES	BRACERI SEL			is: ON REES		n f			LATI	J YS MA NE N 1,	T T LOC EI	an ATIO ND				28		4	2	AXORE SY JE/		EN	668 05	R	OF GAN G	A	- I 0
R 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	STEM NAME STEM NAME STEM NAME STEM CARD FIELD NAME CARD CODE CUST NO ITEN NO UNIT PRICE	Figure 8a File Des DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES DIR TYPE XN FILE DES FILE DES	BACRI CRI SEL					n Take		AAA REE BI	LATT	I VEI		ario Ario Ex				28 28	3.		2	AVAL 4 JE/		PA EN	658 05 AA	RK	OF CO GAN CS	A D1	X Io

FILS NAME	DIA	T	TPE		ME	DIUM		1	ECO)RI	NAME		A88	NO OR	PEAK
INVEN MASTER	15	27	AIA	1.ED	RA	N -	PESK	Γ		A	ASTE	K.		19,000	12,000
SUA PELD MANE	AL.P		LEN	TH	occ	JARE	NCES	481	1	NOVE	RELATIVE	LOCATION	V	ALLE REMA	RKS
NO FIELD MAME	an.c	ž	MAX	VAR	MIN	AVG	MAX	8	1°°	ENG.	REGIN	END			
1 STEM NOS	N	¥	6				1								
2 CLASS OF ITEM	A		3				1		1_	Į					
3 STOCK 60G	19		6				11								
4 DESCRIPTION	R		25	1			1								
5 QTY ON KAHD	N	-+	6				1								-
	1	_			-			-	-	-					

Γ.

Figure 8c. File Description 2, Example 8.

QUESTIONS

- What is the significance of EOF for files? Is it equally applicable to random and serial files?
- 2. How does the READ operator differ in construction between serial and random files?
- 3. What does DO PROCEDURE accomplish?
- 4. In what ways is an unconditional table different from other tables?
- What is the relationship of table 071 to 072, 073 and 074?
 Trace the pattern of movement from one table to another.

I		SYSTEM	ANAME	PROCE	DURE N	AME				0.	V.		L		AN	AL	ST				DA	TE					_		_
l	-	BILL	ING	INVO	DISE D	TAIL	Jo	ECI	510	N T	AB	LE	L	FR	AN	CI	5			1	14	163	-	_	94	6E	OF	PA	GES
F	TABLE	TABL	E NAME	TAI	LE TYP	E	1		T	T		П	Т	T	П	F	REC	DUE	NO	EV	T	TT		-	T	T	11	-	T
ł	SER	OP-SR	OPERAND 1	OP	OPERA	ND 2	10	P	*	OPE	RAP	ND Z	1	OP	64	OP	ERA	NDZ	0	Por	OP	ERA	ND	2	10	200	OPT	PAN	02
I				10.00	1.0		e 1	44	31 0	4 04	86	87 0	4	10	11	2 15	14	18 16	17	10 1	120	41	32	15 1.	1.5	2.0 13	7 15	10 30	T
1	PTA	TUR	6454	OPE	N-1-1	AST		\vdash	-	+-		++	+	-	\vdash	+	++	+	\vdash	-	-	++	+	-	+-+	-		-	-
ŀ		Dor		MEXI	ITE	M	×	\square	-	+		\vdash	+	+	\vdash	-	++		\square	-+-	-	1-1	+			-			+
ł	-	SET	LOC ADD	GR I	TEM FO	BOULA	X		-+	+	-	\vdash	+	+	\vdash	-		-	\square		-		-+	-	1	-	+	+	+-
ł		Do	4 -	NEX	TIN	IEN	X	\square	+	+			+			-	++	-		-+-			+	+					+-
ł	-	SAF	GTY ON HAND	2 - 0	ITY SO	LD	X	+	-	+	-	+	+	-	14	+	+	+		-		+	+	+	+ +	+		-	+
ł		MEVE	PAICE EXT	To	NETP	RICH	K	$\left \right $	+	-	-	\vdash	+	-	-+			+					-+	+	1 -	-			+
ł	-	NOVE	J. TEM	To	ITEM L	INI	×	$\left \right $	+	+	-	++	+	+	\vdash	+	+-+	-		-	-	+	+	-		-		-+-	+
ł		DO		LENVa	ICE HE	ADER	1	+	+	+	-	++	+	-	1	-+-	++	+-			+-	+ +	-+	+	-	-+-	+-		+
ł	-	Gg Th	t	07	1		K	+ +	+	+	-	+	+	+	+	+	+	+	+		-	+ +	-	+		+			+
ł							⊢	+	+	+	+	+	+	+-	\vdash	-+-	++	+-	\vdash	+		++	+	+	+-	+	+	+	+
1	-	ALCUT	TELL	101	-1/20		1,	00	-	+	-	+	+	+	+	+	++	+	+	-		+	+	+			+	+	+
1	911	TT-	1.1019	75	FAF		15	177	-	+	1-	+	+	+	+	+	++	-			+	++	+		+	-	+	+	+-
	TIPE I	ALS IT	0	#2	in car	50	F	5	-	-	+	H	-	-	+	+-	++	+	H	-	+	++	+	-	+	-	1	+	+
1		DA	K	1 STA	-P	ae_	×	19	-	+	1		+	-	H	+	1-+	-			-	++	-	+	1	-	-	+	t
1	-	Le c		1000		- 1741-	1º	1	-	+	1	H	+	-	H		1+	-		+	1	++	-	+	1	+			+
ł	-	NANT	Turfer	CL	05.50		1 as	6	-	+			+	-	H	-	1	1				† †	+	-				+	+
1	KIL.	TTEN	ALR. ITTM	EAT	TE-A AIN	-	T _v	N	-	+	-	H	+		H		++	-			-	-	-	T	1			+	+
1		Se String	N	NA A	STER		#	1	1	+	1	++	+	-	Ħ	-	11	-	H	1	+		+	-	1	T	-	1	t
ŧ		SEF	101 400	1+	RA PIC	-	₽	ł.	-	-	+	++	-	1	H	1	11				-			1					t
Ì		READ	A later the Children	INV	EN MA	STER	t	X		1			1		Ħ	-	1								1				T
1		- LAND J LOR	6	PER	LOC A	00	1	1 I		1	T				\square				1		-						1	T	T
1		Do-		01	13		T	×			T	T						1	1		1			1	T				T
1				1			I			1.											T			1	T				
1	074	INVO	ZEE HEADER	1 cl	OSED		190	10																1					
1		CUST	NOR. TTEM	EQ C	UST MO:	NEADS	1	N	-	_	1		-	-		-		_				-		_	1		-	4	-
Ţ		Nit I.T	Ę	ITTE	MLIN	Ē	X		_				1			-	1				-	-		-	1		-	4	-
l		DO	A)	TOT	AL CA	44	1	X			-		-		+ +		1.		1		+	-		-	-				+
Į		DO-		PROC	EDURE	5	4	×	_	-	1	\downarrow	+			+	+ +				+	-		-	+ .		4		+
ł				NEAL	DER PR	EP	1		-	-	-	\square	1	-		-		-			+	-		-	1			1	+
Į,		00-		07	4	_	1	X	_						1	_		1			_	1			_		_		1
Į	R	emarks	:	_				_	_																				_
	SY	STEM 1	NAME	Fi	gure	8a.	RI	De A PT	ici	si	on	T	a.b	anal FR	,]	T	am	plo	DA	8. TE		. 2			PA	58	OF	P	AG
ſ				True T		1.					100	10.00	10		15								_	,	10 0	1 20	esco	RDS	
+	F4L	B NAME	6400	UR.T	1	10	EUI EI	2		2 51	140	7.	7.5	0.4	-					- 26			AV	Ly C	60		POAL	001	5
1		I BAY	C ASEM	10 21 -	LENTY	Inc	C110	205	MC	3	1	2		DELAT	TWE	1004	TION	1	_					1					
	NO	FIELD	NAME	-Sur -	MAX	AR A41	NA	YC	174/	AN	SPOK	To	12KS	BEG	IN	EA	10	1	-	_	1	AL	UE	/R	EM	AR	KS		_
ľ	1	CAR	CEDE	NU	1		1	_		1			T												1				
	2	CUS	TNO	NI	151	T	1		T	1			1					1				_		_					
1	3	TITE	19 No-	NU	61		1			1			T						-	_	_				_			_	
- 8				1.1	1	-			T				1					1	-			_	-						
1	4	10Th	1 2010	N +	- 3					1				_									_						_

Figure 8b. File Description 1, Example 8.

SYS	TEM NAME	1				13	N.				1	AHALYS	т	DATE]		
1	BILLING	ļ	1	FILE	DE	SCR	IPT	ION			I	F.R.	ANCS S	1/4/	62	DAGE	OF	PAGE
FILS	NALIE	DIR	2 1	YPE		ME	DIUM		5	ECO	DEL	NAME		A88	AMICA	NO OF	PEAK	205
TA	VEN MASTER	4	227	A IA	+ED	RA	N -	PESK	T		A	ASTE	K.		19,	000	12	000
SER HO	FIELD NAME	MARCHING MARCHING	NGIG	LEN	TH	OCC.	AVG	MAN	A6804	To	SUNBOOK	RELATIVE	END	,	ALUE /	REMA	RKS	
1	STEM NO CLASS OF ITEM	NA	H	63			-	1	T									
3	STOCK LOC DESCRIPTION	RA	1	6 25	1			1	-	+								
5	GTY ON HANG		1+	6	-		-	1		+							-	-

Figure 8c. File Description 2, Example 8.

- 1. EOF (END OF FILE) has meaning only for serial files. A random file, which acts like an endless conveyor belt, has no end of file.
- 2. For a serial file, the form is:

READ file name

For a random file, the form is:

READ file name PER location name

Specification of location is necessary for a random file.

- 3. The DO PROCEDURE, as used here, avoids specifying the procedure at this time. The DO PROCEDURE normally states that another series of tables will be called upon to carry out a separate procedure within the system.
- 4. An unconditional table has no conditions. The table is entered from another table and actions are carried out as displayed.
- 5. The looping pattern for the four tables in INVOICE DETAIL:

TABLE	071	to	072	
	072	to	071	
	071	action		
	071	to	073	
	073	to	073,	etc.
	073	to	071	
	071	action		
	071	action		
	071	action		
	071	to	074	
	074	to	074,	etc.
	074	to	071	

etc.

2.53
CHAPTER 3

Sample Systems Descriptions

Two systems are described using the Tabular Descriptive Language. The first, CREDIT RESERVE, is a complete description. It is written in the fullest detail possible using the Tabular Descriptive Language. The second system, POLICY FILE MAINT., is a partial system description. This example illustrates the hierarchical use of TDL by showing various levels of detail. The example may be taken to represent the status of a system description at a point in time prior to the completion of the documentation of the detailed analysis.

Example 1

SYSTEM:

Credit Reserve

PURPOSE:

The purpose of the Credit Reserve System is to establish adequate reserves for insurance issued to lendors (banks, credit unions, etc.) in the amount of and for the duration of loans which they have made.

SCOPE:

This system deals only with those policies for which the full premium is paid in advance.

DESCRIPTION:

There are two types of loans:

- 1) <u>Declining</u> Balance Loans Loans for which there is periodic repayment during the term of the loan.
- Level Balance Leans Leans which are repayed in one lump sum at the end of the term of the lean.

In the case of a <u>Declining</u> loan, the liability of the insurance company decreases during the term of the loan. The liability of the insurance company remains fixed, however, during the term of a Level loan.

Reserves are established and maintained for each policy. Policies are issued to lendors. Since a lendor may make many different loans, which he wants insured under one policy, loans whose term and expiration date are the same must be summarized and amortized together. When file maintenance is done, this function will be performed and the output report "Monthly Reserve List" is produced. A policy may cover loans which are of both <u>Level</u> and <u>Declining</u>. On the output report, totals will be produced for each type of loan within a policy and for each policy.

Input "Activity" records will contain information on additional loans to be covered by a policy, and information on new policies.

SYSTEM	NAME
CREDIT	RESERVE

IBM FILE DESCRIPTION

ANALYST	DATE			
M.K.W.	1/18/62	PAGE	OF	PAGES

FILE	NAME	DIR	יד .	YPE	10000	ME	NUIC		R	ECC	ORD	NAME		ABB	NO OF	PEAK
ACT	IVITY FILE	IN	1			SEF	MAG	TAPE	A	CT	IV	ITY	N-011-40	ACT	10,000	16,000
SER		NUM	ŝ.	LEN	гн	OCCL	IRREN	ICES	AS	-12	SEQUI	RELATIVE	LOCATION	N		DKC
NO	FIELD NAME	EBIC HA	z	MAX	VAR	MIN	AVG	MAX	200	000	ENCE	BEGIN	END	V	ALUE/ REIMA	IKKS
1	POLICY NO	N	U	1				1			1					
2	LOAN CODE	N	U	1				1			2.					
3	EXP DATE	N	U	3				1			3					
4	TERM	N	12	2				1			4					
5	AMT INS CHGE	N	E	7.2				1	-							
		1	1													
		+	-		$\left - \right $			+			\square					
		1					_									
-		+	-								-					
		1					-									
	÷	-+	-						_	-				+		
	1															
		+	-			-					-					
		-	-													•
		+											R			+
		+	-						-	-	-					
		-	-			-			_	-	-					
		-														
REN	ARKS:													A		
			-								-					

SYSTEM	NAME
CREDIT	RESERVE

IBM FILE DESCRIPTION

ANALYST	DATE	
M.K.W.	1/18/62	PAGE OF PAGES

FILE	NAME	DIR	. T	PE		ME	NUM		R	ECC	RD	NAME		ABB	ł	NO OF	PEAK
MAS	TER FILE	RET	ALA	JEC		SER	MAG	TAPE	M	AS	TE	R		MF		50,000	70,000
		E >	1 10	ICAN	ru I	000	IDDEN	1055	>	>	SE	DEI ATIVE	I DCATION			,	
NO	FIELD NAME	MERIC	IGN	MAX	VAR	MIN	AVG	MAX	5500	SSOC To	DUENCE	BEGIN	END		VAL	UE/REMA	RKS
1	POLICY NO	N	U	7				1			1						
2	LOAN CODE	N	11	i				1			2						
3	EXP DATE	N	U	3				1			3						
4	TERM	N	U	2				1			4						
5	AMT INS	N	E	7.2				-1									
						÷											
		-							_								
		-			_												
		-	-	-					_		-					A	
		-							_		-						
					-						_						
		+							_		\rightarrow				-		
		-	-								-						
		+	-						-	-	-						
-		+			$ \rightarrow $						\rightarrow						
		+							_		-						
		+							-		+						
			-						_	\rightarrow	+						
		+			\vdash				_		+						
		+							-	-	+						
		-	-							+	+						
										-	+						
	- in the second s									+	+			-			
		+			+					+	+				_		
REN	ARKS:	_									-						
															141		
	and the second sec								-								

	IBM						INTER	MATIONAL BUSIN 3M 1403 5	PACING (CHART			
	LINE DESCRIPTION		FIELD HEADINGS/	WORD MARKS				0 11	es per incu				
	SYSTEM: CREDIT	RESERVE											
	ANALYST: M.K.W.										11111111		
	PAGE: 3 OF												
	GLUE	FILE NAME: MONTHLY	0 1001416171010	011012141512121	2 001121212151512	3 00112121212121212	8 a 1 1 2 2 4 5	278901234	5 RETROUT	3/4/5/4/7/8/9/	7	0123456789	012345678
	23456 789 10 11 12	I CONTRACT	A 10 10 10 100	010040710	0 7 0 0 4 0 7 1 0 0	10742102	2 4 5 0 0	2	-	200			2
Image: constraint of the constrated of the constraint of the constraint of the constraint of the	0	2 CONTRACTOR											
	2 3	3	POL	I CY LOAN	TAX	ERM AMON	NT A	TNUOR	ANDONA	AMA	TNUS		
	00	4	QN	CODE	DATE	0816	INAL ICA	URBENT	IN FUR	CE RE	SERVE		
	0	11 0441 044 10	N N N	N NNNN	NNN X	ZZZZ	- 2222	2222222222	222222	- NEZ	NZZZZZE		
	6 7	1											
	8	8											
	9	9											
	•												
	0 0	12											
	2 1	13								•			
	3 14	14							-				
	0	15											
	. 16	11											
	•	18											
	•	19											
	@ 19	20									+		
	20 2	21											
	1 22	22											
	2 23	23											
	24	25											
	25 2	26											
	6 27	27											
	28	28											
	3 4 5 6 7 8 9 0 11 2 2 9	1											
	Ø 30	31											
	31	32											
	32 3	33											
	3 34	34											
100000000000000000000000000000000	.35	38											
小田市 小田 小田 小田	36	37											
・ ・	37 3	36						-		-			
····································	8 3	36								-			
····································	9 40	4											
····································	0												
11EV.TOT F 米 11EV.TOT F 12E	42	14											
内容 D	43	11EV TAT # 4	*	LEVE		22.22	223322 -	222222222	TELEST	R - NEZA	NEEZZZZZ		
1 1	0	I DEC TOT-E 4	**	DECL	NI NG	2222	- 22222	222222222	ESERSE F	Z- NZZ	1223332N		
	Ø	I DOL. TOT-F 46	***	POLI	XD	2222	- 22222	532323233	22222	Z- NEZ	NEZZER		
	6	N CONSTRNT 41											
	0 0	4											
	8 4	4											
	9 (20								a a la	a la	alalatalatalatalata	initial al a la la la la la la la

SYSTEM	NAME
CREDIT	RESERVE

IBM FILE DESCRIPTION

ANALYST	DATE	
M, K. W.	1/18/62	PAGE OF 4 PAGES

FILE	NAME	DIR	. T)	PE		ME	MUIC	1	R	ECC	RC	NAME		ABB		NO OF	RECORDS
MON	THLY RES LIST	OL	IT			SER	PRIN	ED MAT	L	0A	N	DET		LD	ł	50,000	70,000
SER NO	FIELD NAME	NUMBRIC	SIGN	LEN	TH	OCCL	AVG	ICES MAX	ASSOC	ASSOC To	SEQUENCE	RELATIVE	LOCATION		VALU		RKS
	EXP DATE	N	U	3				1]	24				
2	TERM	N	U	2				1)	29			10 C	
3	AMT ORIG	N	U	7.2				1	_			1	42				
4	AMT CUR	N	E	7.2			-	1			_		54		_		
5	AMT RES	N	E	7.2				1					11				
6	AMT IN FORCE	N	U	7.2	\vdash			1		_	-		65				
0	POLICY NO	N	12									1	10				
5	LOAN CODE	N	0					_			-		15				
		+															
		+															
		1									-						
					+												
									_								
		-									-						
		+-							_	-	\neg						
		-									-						
			\vdash						-	-	-						
		+							-	-	-						
		+							-		-						
		1															
										1							
									_								
REM	IARKS:		_							_							
									-		1376	_					

SYSTEM NAME CREDIT RESERVE

IBM FILE DESCRIPTION

ANALYST	DATE	
M.K.W.	1/18/62	PAGE 2 OF 4 PAGES

FILE	NAME	DIR	TY.	PE		ME	NUIC		R	ECO	RD	NAME		ABB	AVERAGE	PEAK
MON	THLY RES LIST	01	JT			SER	PRINT	TED MAT	L	EV	1 -	TOT		LT	2000	2400
SER	FIELD NAME	NUMERIC	SIGN		VAR	OCCU	AVG	MAX	ASSOC	ASSOC	SEQUENCE	RELATIVE	END	V	ALUE/REMA	RKS
12	AMT ORIG TOT	NN	UE	7.2								1	42 54			
3	AMT RES TOT	N	EU	7.2					-			 	77 65			
	· · · · · ·															
		-														
		_														
		-							-							
_																
_																
		_	F													
REN	MARKS:									_						

SYSTEM NAME CREDIT RESERVE

IBM FILE DESCRIPTION

ANALYST	DATE	
M.K.W.	1/18/62	PAGE 3 OF 4 PAGES

ω. 8

FILE	NAME	DIR	L. T.	PE		ME	DIUM		R	ECO	RD	NAME		ABB	NO OF	PEAK
TON	ITHLY RES LIST	OU	T			SER	R. PRINT	EDMAT	ID	EC		TOT		DT	2000	2400
IER NO	FIELD NAME	NUMERIC	SIGN		TH	OCCU	AVG	MAX	ASSOC	ASSOC TO	SEQUENCE	RELATIVE	LOCATION	VA	LUE/REMA	RKS
	AMT ORIG TOT AMT OUR TOT AMT CUR TOT AMT IN FOR TOT			MAX 7.2 7.2 7.2 7.2							339		END 42 64 77 65			
REI	MARKS:			4						,						

SYSTEM	NAME	
--------	------	--

CREDIT RESERVE

IBM FILE DESCRIPTION

ANALYST	DAT	FE		
M.K.W.	1	118	162	PAGE 4 OF 4 PAGES

FILE	NAME	DIR	. T	PE		ME	NUM		R	ECO	RD	NAME		ABB	NO OF	PEAK
MOI	NTHLY RES LIST	OL	JT			SER	PRINT	EDMAT	F	POL		TOT		PT	1000	1200
SER	FIELD NAME	NUMERIC	SIGN		TH	OCCL	AVG	MAX	ASSOC	ASSOC To	SEQUENCE	BEGIN	END	VA	LUE/REMA	RKS
	AMT ORIG TOT AMT OUR TOT AMT RES TOT AMT IN FOR TOT			7.2 7.2 7.2 7.2 7.2												
REN	ARKS:		1								4					

SYSTEM	NAME
OREDIT	RESERUE

IBM FILE DESCRIPTION

ANALYST	DATE	1			
MiKIW.	1/18/	62.	PAGE	OF	PAGES

FILE	NAME	DIR	2. T	YPE		ME	DIUM		R	ECC	DRD	NAME		ABB		NO C	F RECORDS
INP	UT HOLD	INT	TER	2		SE	P. CO	RE	F	tor	-D	REC		HR		1	1
		12	T	1	-	0.00			1.	T	1 12	-					
SER	FIELD NAME	UMER	SIGN	LEN	TH	OCCI	JRRE	ILAAN	ASSO	10	EQUEN	RELATIVE	LOCATION		VAI	UE/REM	ARKS
		6	-	MAA	a week	INTIN	AVG	MMA	10	-	¥.	BEGIN	ENU			1	
	EXPDATE	N	IV	3					-								
2	TERM	N	10	2				1	-								
	AMT ORIG	N	IU	17.2			<u> </u>		1					3	_		
4	AMT CUR	N	E	7.2			L	1					İ				
5	AMT RES	N	E	1.2				1	1								
5	AMT IN FORCE	N	U	7.2				1	1								
7	POLICY NO	N	10	7			1	1									
8	LOAN CODE	N	10	1				.1	-								
9	AMT INS	N	E	7.2		-		1							-		
									L								
						22234423244											
																NM	
												Man array					
		T															
									-								
		_	1														
			1							T							
															10		
			1					i âl									
			1														
			1								1		1				
		7-	1								1					1	
		1	T			-					1						
		-	1-								-+						
REN	ARKS:	_			las sond	1						i					
				1												2	

CRE	STEM NAME DIT RESERVE	REFERENCE O	PERAND DESCRIPTION MIK.W. 1/18/62 PAGE OF PAGES
SER	NAME	TYPE	VALUE
	LEVEL.	CONIS	
CAR	DECLINING	CONS	0
	CONTROL FORMULY	44	VOLICY NOT ACT, LOAN CODE ACT, EXP DATE ACT, TERM ACT)
ru	CONTROL ME	25	(POLICY NOTINE, LOAN CODE (ME, EXP. DATE: MF, IEKM MF)
	MANTROP HOPO	2	C KOLICE MO. HK, LOHN CODE, HK, EAF DALE, HK, LEKHLAR
7			
	and a second		

	SYSTEM	NAME	PROCI	EDURE NAME			-	D.	IAF				A	NA	LY	ST				DA	TE								
CR	EDIT RESE	RVE			ונ	DEC	ISIC	NC.	TAE	BLE	E	M.	K. 1	N.		_		1	118	162				PAG	E (OF	PA	GES	5
TABLE	TABLE	NAME	TA	BLE TYPE	6	Г		T	T		T	Т		T	F	280	DUE	N	NY.	T	П	-	T	П	T	11	T	Т	
ID	0052	OPERAND 1	OP	OPERAND2	1	29	81	OP	ERA	MO	21	0	P°	2	OP	ERA	ND2	0	Por	OPI	ERA	ND	2	00	04	PE	ANO	22	-
Pill Pill	or on	01 0111100 1	1.01		U	101	08	C+ 0	Elos	07	08 0	100	10 1	1 12	115	14 1	5 16	17	18 19	120	21	22 2	3 24	25 2	5 27	28	19 30	31	32
001	START		OPE	.N																									
1	READ		ACT	IVITY FILE	X			T		T	T			1												П	T	Т	
2	READ		MAS	TER FILE	X										Π		1	T								\square		T	
3	GOTO		00	2	X									1					1	-			1		1	\square		1	
2	0010			<u> </u>	ľ	T		+	+	+	-	+	-	+		+		\square	-		+	-	+		+	++	+	+	
						+		-+-	+	+	-	+	-	+-	+	+	-	\vdash		+-	+	+	+		+	+	+	+	
-			-		╟	1		+	+	+	+	+	-	+	+	+	-	++	+	+	\vdash	+	+		+	+	+	+	-
			-		1	+		+	+	++	-	20	+	+	+	+	+	\vdash	-	+	++	+	+		+	+	+	+	-
202	INIT HOLD	D REC	OPE	N	15	-	\vdash	+	+	+	- 7	15	+	+	\vdash	+	-	\vdash	-	-	++	+	+	\vdash	+	$\left \right $	+	+	\square
_1	CONTROL	ACTIVITY	LR CO	NTROL MF	ľ			-	_		/	N	_			_			_				_				-		
					F		_	-	-		-	-		-		_			-			_	-						
2	SET HOLD	D REC	EQ			AC	TIN	11	Y			1	MAS-	TER															
3	MOVE AMT	INS CHEE: ACT	TO	1	1	AM	TK	UR	HA	S		4	amt	OR	IG	HR			_				1					1	
4	MOVE AMT	INS CHEE! ACT	TO AM	T INS ! HR	X																								
5	READ					AC	TIN	JIT	YF	IL	E	N	MAST	FR	F	LE									T				
6	COTO		00	2 *	Y		T		T		5	1		T	T											T	1		
	100 10			2					+		-1'	-	+	1		+			-	+	\square	+			+	++	+		-
					╟		-	+	+	1-1	+	+	-	+		+	-		+	+	++	+	+		+	++	+	\vdash	-
					╟─		+	-+-	+-	+	+	+	+	+		+	+	+	+	+-	\vdash	+	+		+	++	+	H	-
					╟─		-	+	+-	+	+	+	+	+	\vdash	+	+	\vdash	+	+	++	+	+		+	\vdash	+	\vdash	-
-					۱.			+	+	+		+	+	+	\vdash	+	+	-	_	+	\vdash	+	+	-	+	\vdash	+	\vdash	-
					₩		-	-		+	-+-	+	-	-	\vdash	-	+		-	-		-	-	-	-	\vdash	-	\square	_
_			I		₽_		-	_	+	+	_	-	_	-	\square	-	-		_	-		_		_	-	\vdash	-		_
			L		삝			_	-	+	1	-	_																
																								_					
			1						1	\square									-	1					\square			Π	
			1		1			+	+	+	+	+	+			+	+		+			+		-			+	\square	
							-	+	+	++	+	+	+	H		+	+	+	+		-	+	\square	-			+	H	-
					-	\vdash	\rightarrow	+	+	++	+	+	+	+	+	+	+	+	+	+	+	+	+	-	+		+	\vdash	-
-	Chan Dire:				1						_			-													_		-
1	EMAKKS.						-			_															-				_

	SYSTEM	NAME	PROCE	DURE NAME			22	DI					AN	AI	YS	ST					DA	TE									
CR	EDIT RES	ERVE			DE	CIS	510	N T	ABL	E	M	K	. W	1.					11	18	16	2				PAG	EI	OF	2P	AGE	S
ABLE	TARIE	NAME	TAN	ALE TYPE		T	1	T	T	T	T		1	1	-	RI	Die	e's	1	5	T					1	1	1	П	-	T
1D	1ABCC	ODEDAND 1	00	ADEDAND 2	0	0	1	006	DAM	02	+	1	20	-	DE	DA	NO.		20	01	000	10/		1 2	-	-	04	000		10.2	1
	oren	WY BIGHIOLY 1	108	CP SPORTO S	01	02 0	0 1 50	4 05	05	07 08	109	110	111	12	13 1	41	5 11	5 17	118	119	20	21	22	23	24	25 2	6 27	128	29	30 3	i T
20	CREATE C	DUTFUT	1 OP	ÊN	10					1	43							4	3							2		T			T
1	CONTROL	HOLD	EQ CON	JTROL ACTIVITY	Y	T					N							N								N					T
2	LOAN COD	DE : HOLD REC	FQ LI	LVEL			T	1			N							Y	1							N					Τ
3	EXP DAT	E: HOLD REC	LE #C	WPPENT DATE							N							N	1							Y					T
-			-			-		1			T	1			-			-			1										T
4	SET AMIT	CUR:HR	HOMT	INS CHGE ACT	X						Τ					T		T													T
5	SET AMT I	NS: HR	HAMT	HS CHOSEACT	X											Τ	T														Τ
6	18EAD		ACT	NITY FILE	X						T																				T
9	SET AMT	IN FORCE : HR	EQ									DE	eu	1 1	VF	OR		T	120	elv	IN	F	OR			r	EC	- 1	NI	OR	T
8	SET PINT	RESSHR	EQ					T				DE	CL	Rh	is				LE	V	RE	5				C	EKI	F	ES		T
4	SET LOAN	J DET	EO H	OLD REC							X							X	T	T						X					T
10	WRITE		L	DAN DET		T	1		T		X		1		1			X	-	T						X					1
11	LINDVE AM-	CORIGSHR	+	*		1			T		T	an	F	012	15	rb	T	T	140	n Tr	he	1G	T	OT	:	A	mt	DR	IG	TO	7
								1			T	DI				1	T		L	F						E	T				T
12	MOVE AMT	COP 2 H R	4-			T	T				1	Am	FT	C	18	TO) i = r	T	AM	T	ru	R	TO	T:	LT	F	MT	CU	RT	TOT	
13	MOVE AMT	IN FORCE ; HR	+		. 1		T				T	An	F	M	14	R	1	T	FAI	NT	IN	F	OR			A	MT	IN	FE	512	T
			1			1		1				To	T :	DT		T	-	T	Tr	F.	LT					1	OT	: 01			T
14	MOVE AMIT	RESSHR	4				Ť	T			1	200	4	RE	c. -	1017	r	71	111	TT	102	<	TO	T:	LT	10	m	RES	7	77 -	1
15	SET AMAT	ORIG THR	1600	CODI HR		1	1		TT		X	1	T			1		X	1	T		1				XI			T	1	T
16	MOVE HOL	D REG	1 10 10	ASTER			T				X	T			1	1		X		1									\square		T
17	WRITE		In	ASTER		1	T	T		T	X	1			T			X		1									\square	1	T
18	GO TO		1	and the stand of the		003	7.	1	T Î	1	T	la	4					1	0	14						0	04	\top	T		T
								-	TT		T	Ta		Í		1	1	1	T	1	1	1					1				Ť
	1				T	1	1	1	\square		T	1		1	1	1	1	1	1	T	1					1	+		\square	+	t
								1			\top				1	T	1	+	+	T	1	1				-	-	+	++	+	1
			1			T	T	1			1					1	1		1	1	1					1	-	1	\square	1	1
			1		11		1	T			1	1						T	1	1						-		-	\square	1	1
			1			-	1	1	\square	1	+	1					+	+	+	1	1					+	+		++	+	1
			1			+	-†-	+		-	+-	1		1	1	+	+	+	+	+	1				+	+	+	+	H	+	+
			1			1	+					1				+			1	1	1					+	1	1	\square	-	+
						+	1	+			+	+-		-		+	+	+	+	+	1			-		+	+	+		+	+
R	EMARKS:	DECL IN EDE	2 05	AMT ORIC	• 4	0	×I	EVD	00	TE	• 11	2	# c	110	DE	1-		0	- #	5	1-	E 10	~ -	10	-		-	-			-
		DECL DES	- ME	AMT IN F	DPC	5.	HD	×	EVI	DE	170	14	0 -	HA	NO	<u></u>	17	DA	17	1	X	EKI	11.	HK	75	-					-
		LEV INLEAD	AL	ANT ONL	140	2.4	DIM	T	(1) 11	De i	10	1.11	~	- (UK)	KEA		Un	IE	-	A	.0	00	2	0						_
		IEV DEC		AMT IN	- A	C E	0 1 1	0¥	(=	XP	DAT	· .	NO		Hp.	120	C. /		0.		- 5	-).	K /	20	0	75				-	_
	4 ²		- AC		LAN	e la	. []	1)-i-					1.1	2.6		- AL	EN		anti				1/2		9	5					-

*	
1 CONTROL ACTIVITY 2 MASTER FILE 3 ACTIVITY FILE 3 ACTIVITY FILE 3 ACTIVITY FILE 5 SET AMT ORIG : HR 5 SET AMT CUR : HR 6 SET AMT CUR : HR 7 SET HOLD REC. 8 READ	CREDIT RESERVE
OPEN OPEN LR CONTROL MF IS EOF IS EOF EQ EQ EQ AMT INS CHGE: ACT EQ	TABLE TYPE
OP OPERAND 2 5 01 01 04 05 06 07 00 Y Y N N ZERO ACTI NS CHGE: ACTI NI TY FILE	DECISION TABLE
op or opERAND2 10 11 12 13 14 19 16 172 11 12 13 14 19 16 172 11 12 13 14 19 16 172 11 12 13 14 19 16 172 11 12 13 14 19 16 172 11 12 13 14 19 16 173 11 12 13 14 19 16 174 11 12 13 14 19 16 174 11 12 13 14 19 16 174 11 12 13 14 19 16 174 11 13 14 15 16 16 174 11 13 14 15 16 16 174 <t< td=""><td>M.K.W.</td></t<>	M.K.W.
OP OPERAND 2 17 18 19 201 21 22 23 1 18 19 201 21 22 23 1 18 19 201 21 22 23 1 18 21 23 24 24 24 1 2 2 24 24 24 24 24 1 2 2 26 24	1/18/62
OP "OPERANO 7 25 27 28 29 30 7 26 27 28 29 30 7 26 27 28 29 30 7 20 30 7 30 7 30 30 7 30 7 30 7 30 7 30 7	PAGE OF 2.PA

	SYSTEM NAME			LBN		At	ALYST		DATE		
CR	EDIT RESERVE		DECISION THE	BLE C	ONTINUATION	M. 1	(.W.	1/1	8/62	PAGE	2 OF 2. PAGE
TABLE					EDEOL	ENCY				1	
SER	OP 05 OPERAND2	OP	OPERAND 2	OP	OPERAND 2	OP	OPERAND 2	90	OPERAND 2	OP	OPERAND
002	2									-	
1	F N										
2	N N										
2	Y										
2	APPLICATION CONTRACTOR STATE						H				
IJ											
5											
7											
0	151/10 500										
0	LEV IN FOR										
à	LEV KED								and the second second	+	
10	X										
11	A CHAT OTHER TOTAL										
11	AMI OKIG TOT:					+					
12										-	
14	ANT CUR TOT : LI										
12	AMIT IN FOR									1.	
	10T: LT							-			
14	AMT RES TOT:LT									-	
15	X					-				-	
16										-	
17										-	
18	004										
-										+	
_										-	
											101 N
_											
										-	
										-	
_									4	-	
_						-		1		-	
	11 01/01										
RE	MARKS:										

		SYSTEM NAME			AFG7	Cardio -	ANALYST	DATE		
	CR	EDIT RESERVE		DECISION TA	BLE C	ONTINUATION	M.K.W.	1/18/62	22	BE 20F 2 PAGES
	TABLE					FREQU	JENCY		_	
	SER	OP AS OPERAND2	8	OPERAND 2	OP	OPERAND 2	OP OPERAND 2	OP OPERANO	22	OPERAND 2
*	8	-								
~	-									
	2	Y							_	
	W	Ϋ́								
							a se estas de antica en la construcción de la construcción de la construcción de la construcción de la constru La construcción de la construcción d	A second s		and the second se
	F.									
	5								H	
									_	
	6									
	0-				T				-	
									+	
	10	STOP								
									-	
	T								+	
				Inc. A course of several severa						
							and the second s			
					Ī				+	
		Contraction of the second s		SANDAL SECTION AND SANDAL AND AND AND AND AND			and the first balance of the state of the st	and the second second set is the same second s	_	-
						and a set of the set o				
									+	
			1						+	
									+	
									\square	
	RE	MARKS:								
	Τ						The second			
		the second se		AND DESCRIPTION OF A DE		A second se	the second			

1	SYSTEM	NAME	PROCE	DURE NAME	1			Da	V.		1		AN	ALY	Sĩ				DAT	"Е						
CR	EDIT B	ESERVE] 0	ECI	SIC	N T	ABI	LE	M	1.K.	W.				1/	18	62			5	PAGE	OF	PA	AGE
TABLE	TABL	ENAME	TAB	LE TYPE		П	T		ГТ	T	Г	TT	T	FF	REQ	UE	NC	Y	Г	1	TT	Т		1	TT	T
SER	OPER	OPERAND 1	OP	OPERAND2	0	P	ਸ	OPE	RAN	102	10	p	35	OPE	RAN	102	OP	01	OPE	RAN	D 2		OP	OP	RAN	02
			·		0	02	031	04 05	06	07 08	09	10	11 12	13	14 13	116	17 18	0:1	201	21 27	23	24 2	5 26	27 28	29 3	0 31
006	STOP		OPEN	L	50	50	-				+-	1	_	+-+		+		-			++	+		_	\vdash	+
1	POLICY M	Voift	LQZ	ERO	Υ.	1.	-	-		_	_		-		-		-	1						_		
2	MOVE :	ZERO	TO PO	JUCK NO. PT	1	X					-		-		-	1	-+-	-		-					\square	+
3	00			005	<u> </u>	X	+	+	$\left \right $		+	+	+	+	+			+-	$\left + \right $	+	+	+	++		++	+
007	CLEAR	Lĩ	CLOS	ED			1	1			+			1-1						T		1				
1	SET AMT O	DRIG TOT : LT	EQ Z	ERO	X		-	1_			1-		-	1	_	11	4				1+	ļ_		_	\square	_
2	SET AMT	CUR TOT: LT	EQ Z	ERO	X		_				1	-+	-+-	+-+				_	14		++	_			\square	4_
3	SET AMT	RES TOT: LT	EQ Z	ERO	X	1				_ <u>i</u> _	1		1					-	1_1					_		_
4	SET AMT	IN FOR TOT: LT	EQ Z	ERO	X	$\left \cdot \right $			+		+-	$\left \right $		+-+		+	+	+	++	+	$\left\{ + \right\}$	+		-	$\left \right $	+
007	CLEAR	DC.	CLOS	ED	-									1								1		-		1
.1	SETANAT	ORIG TOT OC	EQ Z	ERQ	LX_		-	-			1						_	1				4		_		+
21	SEI AMT	CUR TOT DC	EQ Z	ERO	X		_	-	\downarrow		\vdash	\vdash	1	\vdash			-				1-1	_		-+		-
3	SET AMT	RES TOTION	FV Z	ERO	X		1		1-1	_	1			++		11		+		-		-		_		1
4	SET APT	IN FOR TOT DC	19.7	ERD	X		+		\vdash					++		$\left \right $		+	\vdash	-	┼─┼	+	++			+
009	CLEAR 1	27	CLOSE	D			_	1					1	1	-					1	Ħ	+				
1	SET AMT	OPIG TOTIPT	10 23	20	X.		_			-			-	11		1		-		-						-
2	SET AMT	CUR TOT PI	EQZ	F.8.0	X		_				1		1	11				-				1				
3	SET AMT	RES TOT PT	EQZ	EPQ	X		1	-		_	1		1		_			1		_						1
4	SET AMI	IN FOR TOT PT	50.21	EP ()	Х		+	-	\vdash	+	-			++				+		-	$\left \right $	+		+		-
									t									t		T		1		1		1
						$\left \right $	+							++		$\left \right $				+-	+	+	++	+	$\left \right $	+
1						1-1	-			1	1-	Ť	+-	++		11	+	1	++	+	++	+	++	-+-		+
							+		++	+	1	H	+	++		++	-+-	-		+	++	+	++	-		+
					-				†-†		1-			++		+		+		+	++	+	++	+	-	1-
						1-1	+		1-1		1-1	+	+	++	-+-		-+-	1-		+	+	+	++			+
					-	\vdash		-+-	+-+	+	1	+	+	++		+	+	+		+	+	+	++	+		+-
Dr	- AADVO							_					_	L_L				1	Laund	_					<u> </u>	1
RE	EMIARNS.													_			_					-		-		
_						<u>.</u>					_	_	_		_				_			_		-		_

	4	۰,	1
٠			
٤.	_	_	ı
г			

Example 2

SYSTEM:

Policy File Maintenance

PURPOSE:

The purpose of this system is to perform all file maintenance functions required for the POLICY Master File. SCOPE:

This system describes the file maintenance of the POLICY Master File for <u>all</u> policies issued by the company.

DESCRIPTION:

For any record in the POLICY Master File, file maintenance may be required for one or both of the following reasons:

- 1) There are input transactions affecting master records.
- Data in the index record satisfies conditions which establish a need for certain processing to be performed, e.g., billing date, anniversary, nonpayment.

In addition to an updated POLICY Master File, the system will produce other outputs for other systems. All files are not described in this example.

As stated earlier, this is not a complete system description. The documentation shows the various levels of decisions, but only a single path is covered at the next lower level. After the Housekeeping procedure, Table 002 determines the various types of activity, Table 032 shows the next level for decision for one type of activity (Automatic changes) by determining the class of change (Billing, Late Pay, Nonpayment, etc.), Table 050 shows the logic for one class (Nonpayment). Tables 082 and 083 define the Procedure to determine Cash Value called for by the DO PROCEDURE CASH VALUE action of Serial 7 Table 050.

Light shading is used where subsequent tables will further define the decision logic.

SYSTEM	NAME	
POLICY E	IE MA	INT

IBM FILE DESCRIPTION

ANALYST	DAT	re				
L. J. T.	1	18	162	PAGE	OF	PAGES

FILE	NAME	DIR	T	YPE		ME	NUIC		R	ECO	R	NAME			ABE	8	AVERAGE	PEAK
POL	JCY MF	RET	TAI	NEC		RA	NDD	ISK	IP	OL	IC	Y INFO			P.	I	200,000	275,000
		E>	1	Lear		000	IDDEN	1050	Te		S	DEL ATINE	I DC ATION					
NO	FIELD NAME	INCOMA	IGN	MAY	VAR	MIN	AVG	MAX	5800	1000	DIENC	REGIN	END			VAL	UE/REMAR	RKS
		1	=						F		m							
	POLICY NO	IN	19	6	-				-	\vdash					-			
2	POLICY IYPE	N	10	1	\vdash			<u> </u>		$\left \right $	-							
3	PLAN CODE	N	D	10	-				+-					01.0	. 1	2		
4	INT RATE	IN	U	<u> </u>	\vdash				+	\vdash	-			PLA	N	ODE		
3	MAT AGE	N	W.	12					\vdash			2	2	- PLA	N.	CODE		
6	155 AGE	IN	U	1 h	<u> </u>				+									
1	SEX	IN	V-	0					-		-							
X	NAME LAST INS	A		8				1									-	
9	LAST TRAN DATE	N	U	5					+		_		0					
10	DAY LAST TRAN	-IN	U	12					-				2	LAS	T	TRAN	DATE	
11	MO LAST TRAN	N	V.	12	$\left \right $				-		_	3	4	LAD	1	TRAN	DATE	
1X	PR LAST TRAN	IN	V	1.							-	2	5	LAS	1	TRAN	DATE	
12	AMT INS	IN	μ.	6	\vdash				-		-							
14	SUE STD CD	A	-						-									
15	KEINS CD	A		1	\vdash			_(-							
16	DATE ISSUE	IN	1U	6				_1	-			-	21	DA				
17	MID DATE ISS	-IN	HY.	6					-	-		3	4	DA	IE	1550	E	
18	VK DATE ISS	IN	10	E.	$\left \right $			1	-	+	-	5	0	DH	E	1550	E	
19	PHID TO DATE	N	14	4					-		-	1	2	Day	0 7			
20	MO PAID TO	N	19	12	+					\vdash	-	7	F-	PAL		O DA		
K.L	TR PAID TO	IN	12	1 h						\vdash	-	2	- 4	141	D	$O U_1$	ATE	
22	STATUS	N	10	2	$\left \right $				-		-				-			
62	NON FORF STAT	IN	14	5		~				1						^	\wedge	
24		1	F	1	K	-				-	-	$\sim $		/	\sim	\sim		
		+	\vdash				1	-/-	\vdash	-	-		$\setminus A$				\sim	\checkmark
		+	-							+	-		~		_			-
		+							\vdash	-	-							
									-	+				_				
		-								+								
REN	ARKS:	_	<u> </u>							_								
																	с	

SYSTEM NAME	
POLICY FILE MAI	NT

IBM FILE DESCRIPTION

ANALYST	DATE			
L. J. T.	1/18/62	PAGE	OF	PAGES

CASH VAL DIV RATE RETAINED SER FIELD NAME Image: Second s	D RAND DISK ENTH OCCURRENCES IAX VAR MIN AVG MAX 3 ! 2 !	CASH VAL	CV 150 300 VALUE/REMARKS
SER FIELD NAME Image: Series of the series	ENTH OCCURRENCES	ASSOC RELATIVE LOCATION	VALUE / REMARKS
I PLAN CODE NU 3 R ISS AGE NU 2 3 MAT AGE NU 2 4 POL DUR NU 2 5 DIV FACTOR NU 2 6 CASH VAL FACT NU 1 7 PD CODE NU 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 1		
	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		
REMARKS:			

	in l	r		Г
	PAGES	and description. It is the second second		
	GE OF			
	8			
	162		271	
	DATE		1,2311	
		ALUE	84, 37	
	J. J.	>	2-133	and the state of the state
1	N		276, 2	and the second second second
	RIPTIO		272,	-
1	ND DES		9.2.71	
Contraction of the second	CPERA		152	
	8.85 532 753 053	202	27	
1	K			1
	TAIA		1317	
	ANIE LE M	AME	TION	And in the second second
	CY EL		EXCED	and a statement of
	POL :	NON		

THE REPORT

į

AVA.

1

÷

ł

Т

1

1

		TONE			
SYSTEM NAME	PROCEDURE NAME		ANALYST	DATE	
POLICY FILE MAINT		DECISION TABLE	L. S. T.	1/18/62	PAGE OF PAGES
TABLE NAME	TABLE TYPE		FREQUE	NCY	
SER OPER OPERAND 1	OP OPERAND2	OP OPERAND 2	OP OF OPERAND2	OP OPERAND 2	OP "OPERAND 2
		et 02 08 04 05 06 07 08	09 10 11 12 15 14 15 16	17 15 19 20121 23 25 24	4 25 20 27 28 29 20 31 22
* OOI START	OPEN				
1 00	PROCEDURE				
	HOUSEKEEPING				
2160 TO	ACTIVITY DETER				
	MINATION				
OOR ACTIVITY DETERMINATION	OPEN			83	2
1 #CURRENT DATS #	EQ HIT DATE IN-				
	DEX				
& POLICY NOT TRANS	ED POLICY NOTIN-			N	
	DEX 0				
ANALY AND A DESCRIPTION OF A DESCRIPTION					and the surplus of the surplus of the
2 0.0	SET UD DOONE-				
07 0					
	DURE				A CONTRACTOR OF A CONTRACTOR O
4 GO TO		Builton Handrike	EXTERNAL CHAN-	NEXT POLICY	DRIO RI TY
		CHANCER	GES		
REMARKS:					

*

*

POL			9					
STORT E	LICY FILE MAINT		DECISION TABLE	L.J. T.	1/18/62	PAGE	OF P	NGES
0ć	TABLE NAME	TABLE TYPE		FREQUE	NC.		E	F
	OPER OPERAND 1	OP OPERAND2	OP SH OPERAND 2	OP OF OPERAND2	OP 02 OPERAND 2	00 °40	DERAN	02
32	AUTOMATIC CHANKES	OPEN	88 2 1 2 5 E	8 09 10 11 12 13 14 15 16	17 15 19 20 21 22 25 2	25 26 27	28 29 2	雨
-	STATUS	15 BILLING						F
2	STATUS	IS LATE PAY	Y S					E
M	STATUS	IS NON PAYMENT						
7	STATUS	IS ANNIVERSARY						
5	57.47.05	IS CHANGE	X					
Ĩ	Social Social and S							t
9	60 70	BILLING ROUTINE	X					
5-	G-0 TO	LATE PAY ROUTHE	X					
X	60 TO	NON PAYMENT	X					
		ROUTINE						
6	60 10	ANNIVERSARY	XX				E	Þ
-		ROUTINE						
0	GO TO	CHANGE ROUTINE						
11	GO TO	ERROR ROUTINE	X					
1								_
1							-	
							_	
-								
-								
T		-						+
t							-	+
+							+	+
+						-	+	+
T							+	+
-						_	_	
20	MARKS: NOU PAYNER I	XIEXP PAID TO	DATE L= (#CL	RRENT DATE#-59	2 JAND PLAN CC	DDE.P	L	
	EXCE	FLION LIST						

*

	OF 2 PAGES		DERAND 2	28 29 30 31 33									VE COMPANY	AAAAA			TERM					
Г	PAGE 1		2 0P °4	24 25 26 27	15	2	N		EQ 1 00	X	~	< >	an Salt				BPL					
DATE	118/62	1	P of OPERAND	15 19 20 21 22 23					100				all a star				IN FORMATION					
ALYST	1	FREQUEN	OPERAND2 0	2 13 14 15 16 17	20	N	Z		LR		X.		CANC MAN				AUTOMATIC					
AN	1414		2 OP OR	08 09 10 11 1	5	2	X				>	<	12 1 12 - 2		X		MAKE	NH ON				
Mai	DECISION TABLE		OP OPERAND	of 02 03 04 05 06 07	0	>							1 Michael Contract	A A A			LAPSE					
PROCEDURE NAME		TABLE TYPE	OP OPERAND2		OPEN	LRI	EQ AUTOMATIC PRE-	MIUM LOAN		LR 35	CO NO DAIO TO	THE VO DAID TO	DOALEDHOE	CASH WALVE	ATTAINED AGE	CANC						
SYSTEM NAME	JCY FILE MAINT	TABLE NAME	OPER OPERAND 1		NON PAYMENT ROUTINE	WORK POLICY DURATION	NON FORF STAT ? PI		PLAN CODE; PI	YR DATE ISS: PI	TET WANT WA AN	PET MORE NO DO	TAN WAR IN TU	and the fight of the	bo		GO TO					
	B	TABLE	125		20	-	2		M	7	Y	2	OF		00		0		11			-

	100 1 00 1 00 1 000	MOR I OF '2 PAGES		OP OPERAND 2																						+				
1000	1/10/10	11×102 14		OP OPERAND 2																										
AMALVET	T T T	Pr 31 1.	ENCY	OP OPERAND 2																										
TEM T	0	SLE CONTINUATION L	FREQU	0PO7 OPERAND 2	20	N	Z		EQ 160	260			X	×	X and the second	A Contraction of the second			INFORMATION	SHEET				~						
Γ	T	DECISION TAR	1	0PO6 OPERAND 2	- M	N	N		EQ 101-159	161-259	261-376				and the second s	All and a second se			LAPSE											
SUPPORT MANAG	UTSIEM NAME	ICY FILE MAINT		OP OS OPERAND2	34	N	N		EQ 100			N	X	×	and the second s	ALL STATISTICS			APL TERM				*					LARKS:		
L	id	DA I	TONIC	SER	050		~	- -	0			4	5	9			0	•	9								-	REM		

			WHAT I				
	SYSTEM NAME	PROCEDURE NAME		ANALYST	DATE		
8	LICY FILE MAINT	CASH VALVE	DECISION TABLE	ドンゴ	1/18/62	PAGE OI	F PAGES
TABLE	TABLE NAME	TABLE TYPE		FREOUE	NCY		
A REAL	OPER OPERAND 1	OP OPERAND2	OP SI OPERAND 2	OP OPERAND2	OP " OPERAND 2	00 00	PERAND 2
			or 02 08 64 05 06 07 05	09 10 11 12 12 19 10 00	17 16 19 30 21 22 28 24	25 26 27	28 29 30 21 32
012	DATE DIFF CALC	CLOSED FIRST	Ho I I	45 1 1 1	0	ŝ	
-	WORK MO PD	VS-CHO DATE	K.C.	GRA	Bo		
15,		155: PT					
0	STATUS: PI	EQ FULL PD	N I I I I I I N	N	N	λ I	
7	SET WORK MO	EQ	4 4	MORK NO PD	8		
Ĺ			-MD DATE ILSSIP	T - MO DATE ESSEPT			
-			HWDRK MO PD				
V	SET WORK YR	COA	- K-	MOP X YR PD	MORK VR PD		
			-YR DATE ISSAP	E - TR DATE ISSPT	- YR DATE ISS: PI		
			+WORK YR PO				
9	SET WORK YR X	FO WORK YR	X				
		+ 1					
5	DEAD	CASH VDI BIU DATE	>				
1	Citized and Citized						
		LEK POC					
		CONTROL					
20	DO	083	×	X	×		
		PER ASSOC B					
0	SET AMT POL CSH		EQ IN TE RP	EO INTERP	FO WORK FACT 1		
					* ANT INS: PE		
0	RO	EIRI BO DOUT				×	
œ	EMARKS: INTERP 8	E (CWORK FACT 2.	-WORK FACT 1)%	WORK MO/12 + WC	RK FACTI) * AMT	INS: PI	
L							

1			k	1	-			11		_	-	-	T	-	-		-	-	-	-	-	-	-	-	-	-		_	_	-	_		_	_	-	_
	NGES	E	02								+	+	+			+	+	+	+	+	+	+	+	t	1			_	_							
•	đ		ERAN								+	1	1		4	1	1	+	+	+	+	+	+	t	E					_						
	5		0000					Ħ			+	\pm	+		1	+	+	+	+	+		+	+													
	PAGE	H	90							+	+	+	+	-	+	+	+	+	+	+	+	+	+										-			
		\square	2		-			\mathbb{H}		4	+	+	+	-	\neg	7	+	+	-	Ŧ	Ŧ	+	+	-	F	F			_			_	_			
m	~	P	SAND					Ħ		\neg	7	+	1	7	7	4	+	+	+	1	1	1	1	1												
V	162		1300					Ħ		+	1	+	+			+	1	+	1	1	1	+	+													
1	1/18	3	so do									+					+			\pm	+									_			-			
+	-	N	02			-	\square	$\ $		+	+	+	+	+	+	+	+	+	+	+	-		+	+	-	_	_	_	_		-	-	-			
-		G	PANI			_					+	+	+		-	-	-	-	-	+	1	+	1	1									_			
F		-K-	840					Ħ			1	1	1				1	1	+	+	+	1	+													
Z	H h	Н	00					Н					\pm			-	+	+	+	+	+							_			-		-			
	i	H	do	5				$\ $	-	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	-			_	_		_	-	-			
	ш	P	205		\square					7	-	+	+	-	7	+	1	1	-	1	+	+	+	-												
0	ABL		RAN							-	+	1	+	1		1	1	1	+	1	+	1	1	t												0
i	NO	H	NO IS					\parallel			+	\pm	+				+	+		+	+	-						_			+		-			
	ECISI	H	5 0	00	z	-	2			X	+	+	+	-	+	+	+	+	+	+	+	-	+	+	-	_			-	_	-	-	-			
-	Б П		õ	01	X	Z			Х		-							1		1																
CEDURE NAME	SH VALUE	ABLE TYPE	OPERAND 2	LOSED	ZERO	POL DUR	POL DUR		CASH VAL FACT	CASH VAL FACI																										
PRO	S	_	1 00	0	EØ.	EQ	S		EO	EQ			-		-	-		+				+	-	•				3	_	_		-	-			
SYSTEM NAME	LICY FILE MAINT	TABLE NAME	OPER OPERAND	GET FACTORS	WORK MO	WORK YR	WORK YR X		SET WORK FACT I	SET WORK FACT 2																								EMARKS:		
- 1	0	10	R	2	-	a	M		T	5		T	T			T	T	T	T	T	T		T	T										2		

*

CHAPTER 4

The Structure of the Language

Chapters 4, 5, and 6 are the reference portion of this manual. The structure of the Tabular Descriptive Language is described in this chapter. Chapter 5 describes the formation and use of operands; and Chapter 6 describes the operators used in the TDL system.

The Tabular Descriptive Language allows systems to be developed and described in a convenient manner. As a documentation device it provides a rigorous statement of the logic of an information processing system that can be easily understood by others. It is a tool for the systems analyst, and as such is capable of describing complex decision logic in a concise and unambiguous way. The systems analyst describes systems at various levels of detail; thus, the language must be hierarchical in nature, permitting both high-level and detailed descriptions of a system. The Tabular Descriptive Language has been designed to serve these functions.

Any language requires a vocabulary and a grammar. The vocabulary of the Tabular Descriptive Language is the operands and operators, together with the character set used to form them. The grammar is the rules used to form logical statements of conditions and actions. The construction (vocabulary and grammar) is described.

Formation of Operands

A set of characters is used to construct words. These words are in turn used to construct operand names, operators, indicate particular values, etc. The characters used in TDL are based on the character set of the IBM 1401 and 1410. The collating sequence is covered in Chapter 7.

Alphabetic	(A - Z)	
Numeric	(0 - 9)	
Punctuation	blank . () ; , : ! ?	
Other Special Characters	(+ \$ * - / % = " ' ¢	#
	ダくまる ⑩ > √ キ)	

Zero is assumed to be an unsigned value; thus zero is neither positive nor negative for numeric comparison.

Names

Names are used to identify the various operands used in the system. With a few exceptions (see below), names are developed by the systems analyst to represent the elements upon which the operators work, (e.g., files, records, fields, expressions, lists, etc.). Names are constructed as follows:

The first character must be alphabetic; the characters following

the first can be alphabetic, numeric, any of these special characters (# \$ % ¢ . @ \triangle), or a blank. The blank is represented by an underscore (_) when its meaning might be confused. Names can be of any length convenient to the user. For example:

SALES \$	GROSS_PAY
GROSS %	SALARY
YEAR TO DATE	INVENTORY AMOUNT 1961 AVG.

The same name can be used for fields of data appearing in different records or files by appending a record name to insure that no ambiguity exists. Certain other names may also be qualified as explained in Chapter 5. When a name is qualified at a higher level, the name is followed by a colon (:) and then the qualifying name. When names are unique, no such qualification is needed. Examples of qualified names:

> GROSS AMOUNT : INVENTORY GROSS AMOUNT : TRANSACTION YEAR TO DATE AMOUNT : TOTAL LINE NUMBER_OF_SEATS : INPUT SEGMENT

Special Names

In addition to names developed by the user to identify operands, another class of operands, which is essentially self-descriptive, is used. These have special significance.

Literals - The user may state the value itself in a decision table rather than a name which represents the value. Numeric and alphameric literals are permitted. Numeric literals start with a numeral (0 - 9), a sign (+ , -) or a decimal point (.) and are thus distinguishable from names, which start with a letter. Alphameric literals must be enclosed in quotation marks to differentiate them from operand names. Examples of numeric and alphameric literals are:

> +3.1416 'SEDAN' 14 'OUT_OF_STOCK' .03125 -.03125

'123 - 24/649'

Connotative Terms - A word which connotes its value.

BLANK, ZERO, NINE, ALL 'any alphameric literal' State Terms - A word which indicates a particular state of a variable.

For example:

POSITIVE, NEGATIVE, NUMERIC, ALPHABETIC

PRESENT, END OF FILE, STATUS

Descriptive Terms - A word which describes its value

CURRENT DATE #, # BRANCH OFFICE ADDRESS # None of these special names need to be described in the operand description portion of the system documentation. Their names alone are sufficient to identify them. These self-describing operands are defined fully in Chapter 5.

Operator Names

There are various operators used in forming statements in TDL. These names are preselected and have a defined meaning. Examples are:

> EQ NE MOVE SET READ WRITE SORT

Formation of Statements

Two types of statements are made in decision tables -- condition statements and action statements. These statements are formed by the proper placement of operators (generally verbs) and operands (nouns). The construction of a condition statement is:

	Operand 1	Operator	Opera	and 2
Examples:	YEAR TO DA	TE FICA	GE	150.00
		TOTAL	· IS	POSITIVE
	ID: I	MASTER	EQ	ID : TRANSACTION
	DEDU	JCTIONS	EQ	BLANK
		STATUS	IS	NON PAYMENT

The construction of an action statement is:

	Operator Operand 1	Connector	_ Operand 2
or			
	Operator		Operand 2
Examples:	MOVE AMOUNT TO	TOTAL	
	SET YEAR TO DAT	E FICA + FICA	
	READ	MASTER	
	DO	PROCEDURE (GROSS TO NET

Formation of Decision Tables

A decision table is formed from sets of condition and action statements. The elements which make up a decision table are discussed with reference to this diagram.

Table Header	Rule Header
Condition Stub	Condition Entry
Action Stub	Action Entry
	Remarks

A <u>condition</u> consists of a condition stub with one horizontally associated condition entry. An <u>action</u> consists of an action stub with one horizontally associated action entry. A <u>decision</u> rule consists of the conditions and actions associated vertically in an entry column in conjunction with the stub. A decision rule can be characterized as (where the underlined words are implied):

If condition 1 is true and condition 2 is true and

condition n is true then execute action 1 and then execute

action 2 and then.... execute action m.

A specific combination of conditions associated with a decision rule must be logically unique. Thus, decision rules can be examined in any order since only one rule can be satisfied for one pass through the table.

Figure 4-1 is a sample decision table form. The various elements will be briefly explained. There is a more detailed description in Chapter 7.
-	STOLEM	PRATOR E	PROCE	DOKE INMALE			-		0		- 1	_		21.0	a state						-			_					_	
	and the second second second second	State of Control of Co			1 ന	ECH	510	N T	AP		ľ									-					P	AGE	OF	P	AG	E
(meres					50	001	510	10 1	~	lay lin				_	_	RU	JLI	5	HE	A	DE	R	_	_	-					
LD	TABLE	NAME	TAS	LE TYPE			Т								F	R	iqi	JEI	NC	Y					Т			TT		
RR	OPER	OPERAND 1	09	OPERAND2	0	P	-	OPE	RAP	ND 2	2	01	0	OB.	OF	ER	ANC	2	OF	000	OP	ER	ANI	2		OP	OP	FRAN	10	2
			101	01 01 01 0 0	10	02 0	0110	4 05	06	07	08	09 11	101	111	2 13	14	115	16 1	17 1	8 1	12	0 21	122	25	24 2	5 26	27 21	29	50 3	n
		TABLE	HE	ADER		T	T	T					Т	T				Т	T		Т	T			T			TT	Т	
T		and which is the disk disk day, we wanted it is not	1					1	1			-†	1			1			T		1		1							-
1					-		+	+	1	+		-+	+	+	1	+		-	-†-	+	+	+	+	1	-	+	-	++	+	-
-+						-+	+	-1-	+	-+	-	+	-+	+	+	+	++	-+	+	+	+	+	+		+			++	+	-
							+	+	-		-	+	-+	-	-	-	\vdash	+	+		+		-		+		-+-	+-+	-+-	-
-		CONDITION	ST	UB					IC	0	N	DI	I	TI	10	N		上	N	T 1	<u> </u>	Y			_		_	\downarrow	-	
																										1				
-			1					T	T							1														
1			1				+		1	\square		\uparrow	+	1	1	1		1	-	1	+	1	1		-			$^{++}$	+	1
+					-	++	+	+	1-		-	-	+		+	+	++	+	+	+	+		1		+	++		+++	+	
-			<u> </u>			-			+		-	-+	-+	+	+	+-	\vdash	+		-	-	+	-			-		++		-
4			L				-+-	-	-		1	-+	_	-	-	-		_	-	_	-	-	-		-	-		++	-	
T	CHOLON AND AND AND AND AND AND AND AND AND AN	APPERVISE SCHOOL	A REAL PROPERTY	Contraction of the local division of the loc					1				T			1					1									1
- and			1				Т	T											T									TT		1
	territe a fait de camp in définientement						+	-	1-	H					+	+			-	+	+	-							+	
1						\vdash	+		+			-	+	-	+	+	+		+	+	+	+	+		-			++	+	1
			<u> </u>			++	-+-	+	+-	\vdash		-	-	+	+		+	+	+	+	+	+	+		-		-	++	+	
-						+	+	_	-			_	_	_	_	-	\square	-	_	_	+	+	_		_		_	+	+	
					1																									
-		ACTION	S'T	UB								A	C	TI		N	ri	E	N'	T]]	2 3	Y								
7		A REAL PROPERTY OF THE PARTY	1						1												1	-					-	T T	+	-
1			+			+	+		+				+	+	+	+-	++	-+	-	+	+-	-	+		+	+	-	++	+	•
ł					⊩	\vdash	+	-	+			-	+		+	+	+ +	+	+	+	+	+-	-					++	+	-
4						\vdash	_	_	-		_	-	4		+	+			-	_	-	_	-				_	+	_	
1									1																		_			
1																												IT		
and and			1		1	1		1								1					1	T		1	-			11	+	-
Ż			1		H	++	+	+	+	+			+	-	+	+-	+	-	+	+	+	+	+	H	-+-	+	-	++	+	•
-					₩—	\vdash		-+-	+	-		-	-	+	+	+	+-+	\rightarrow		-+-	+		-		+			++	-	
-					₽	+		-	-					+	-	-	+	-	4	-+			-			1-		+	+	-
4							_		1_				_	_		1			_		1	_				_			-	
					1			1								1									1					
1	64.01 <u>0</u> 0		1			TT	T						1		T	1		1	T	-	T		Γ		T	1		TT	T	Ĩ
			1			+	+	-	+				+		+-	1-		1	-+		+	1	+		+			++	1	ľ
-1			+		₿	+-+	+		+			-	-		+	+-		-+	-+	+	+	+-	+	-	+			++	+	i.
-			ļ		₩	+	+		+				-	-	+	-	+	-		-	+	+	ļ				-	++	_	
_1	-	-	1		1			1	_	1			_	1		1_	1_1	- 1					_		1		_		_	
RE	MARKS:																													
					_										-						-			-						1
					-											_								_	_					-

d

*

	1-7 erupia	4 X
1		
MARKS:	ACIJON	TABLE NAME OPER OPERAND I TABLE CONDITION
	STUB	TASLE TYPE OP OPERAND2 HEADER STUB
		DECISION TABLE
		NDITTION BORNAUS
		NCY NCY NT R 2 NT R 2
		PAGE OF PAGES

8.₽

<u>Table Header</u>: The table is identified by both a name and an identification code. These may be referred to in the same or other decision tables. The header is also used to indicate whether the current table is "open" or "closed" (described later in this chapter).

More than a single table can be written on a page--each table being given a unique name and identification. Two asterisks are placed in the margin next to the table header to indicate the start of a new table. <u>Condition Stub</u>: The condition stub may contain the entire condition statement (limited entry form), or it may contain just the first operand or the first operand and the operator (extended entry form).

<u>Condition Entry</u>: The condition entry may contain the second operand or it may contain the second operand and the operator (extended entry form). In other cases the condition entry may contain either Y, N, . or blank (limited entry form). Y is to be interpreted as an assertion of the condition statement made in the stub. N is interpreted as an assertion of the negative of the condition stated in the stub. For example, if the condition stub says A = B and there is an N in the associated condition entry, then we are examining the assertion that A does not equal B. The . or blank can be interpreted two ways. The basic interpretation is that this condition is not pertinent. The second interpretation is that this condition is impossible. In the last decision rule, all the condition entries may have periods or blanks, indicating an unconditional rule. The word ELSE is written in the rule header area. The actions in that particular rule are executed if no other rule is satisfied.

<u>Action Stub</u>: The action stub can contain the entire statement, just the operator, the operator and first operand, or the operator, the first operand and the connector.

Action Entry: The action entry can contain the second operand, or the connector and the second operand, or the only operand, if it is a one-operand statement. In other cases it contains an X, which means "execute", or a blank or period (.) which means ignore or do not execute (limited entry form). Actions are executed in the order written. A GO TO action must be the last action for every rule, except in closed decision tables and the special decision table named STOP.

<u>Rule Header</u>: For each decision rule there is a header which indicates specifically the rule number and may show its frequency of occurrence. Rule number does not specify the order of consideration of the rules, but rather is simply a means for identifying each rule. Frequency indicates the relative number of times a rule will be executed. As such, it need not be exact, but suggests a possible ordering for most efficient solution of the decision table. Frequency is used to indicate the per cent of time that a particular set of conditions is likely to succeed. This means that the sum of all the frequencies should add up to 100.

<u>Remarks</u>: The remarks area provides an opportunity for making local definitions and descriptions; for example, arithmetic, logical and symbolic expressions can be defined when they apply only to the decision tables on that page. Comments to aid others in understanding the description are also useful.

Formation of System Description

An entire system is described by two types of operand description forms, standard printer spacing charts, and decision table forms. The File Description form describes files, records, and fields. The Reference Operand Description form describes expressions (arithmetic, symbolic and logical), value and reference lists, and constants. The printer chart specifies printed report formats. The Decision Table form (and a continuation form) specify the logic of the system. The amount of detail in preparing these descriptions depends upon the level of system specification desired.

Each type of record is defined on a separate File Description. The form and content of the File Description and Reference Operand sheets is described in Chapters 5 and 7; sample sheets are included at the end of this chapter (figures 4-2, 4-3).

A description of the system logic is given through the use of many decision tables, suitably connected to demonstrate logical flow. Two special decision tables are used (START, STOP) for beginning and ending the processing. In addition, there are two table formats used (limited entry tables and extended or mixed entry tables), and two major classes of tables (open, closed).

Special Decision Tables

Two special decision tables always appear in each system documented by the systems analyst. The first table to be executed in a system is labeled as the START table. The word "START" will be used for the table name. The last table to be executed within a system is labeled as the STOP table. Both the START and STOP tables are OPEN tables. There is no action operator with the name "STOP." If the systems analyst wishes to stop all processing, he can do so by transferring control with a GO TO action to the STOP table.

The START and STOP tables can be in limited or extended entry form.

Limited and Extended Entry Tables

In a limited entry table all operators and operands for both condition and action statements are written in the stub area (left of the vertical double line). The condition entries are Y for yes, N for no, and period or blank for not pertinent. Action entries are X for execute, and period or blank for do not execute. Since the entry is limited to these items, the narrow rule columns are used to specify the combinations of conditions and actions. Thus, a maximum of 32 rules can appear on a Decision Table Sheet.

In extended entry tables, the condition entry can contain the second operand <u>or</u> the operator and the second operand. The action entry can contain the second operand <u>or</u> the connector and the second operand. In an extended entry table the wide rule areas (each composed of 8 narrow columns) are used. Thus, only four rules can appear on a Decision Table sheet. If more rules are needed a Decision Table Continuation sheet is used for the additional rules. The extended entry table form may include both limited and extended statements (in different rows). These are called mixed entry tables.

Open and Closed Tables

An OPEN table is one to which control is transferred by means of a GO TO action. A CLOSED table is entered only by a DO action which also controls the return to the originating table. An OPEN decision table may contain one or more DO action operators, but the last action of each rule should be a GO TO. A CLOSED decision table may contain one or more DO action operators, but generally will have no GO TO action operators. After the CLOSED decision table has been executed, control is automatically returned to the originating table, to the action statement following the DO action operator which called the CLOSED table. If a CLOSED table has a GO TO in any rule this will effectively destroy the return path for that rule. The table header specifies whether a table is CLOSED or OPEN.

The following chart summarizes this information on OPEN and CLOSED decision tables.

Table Type	May Contain	Entered By	Must Contain
Open	DO, GO TO	GO TO	* GO TO
Closed	DO, GO TO	DO	

OPEN AND CLOSED TABLES

* As last statement of each rule, except STOP Table.

This chapter has described the structure of the Tabular Descriptive Language, its vocabulary and grammar. The details of particular operand and operator specifications are given in the next two chapters along with additional information and illustrations of how they are used.

FIELD NAME	ENTH	_						
ALPHA MUMURIC LEED NAME	# LENTH	MEDIU	X	RECO	RD NAME		ABB	NO OF RECOR
	Z MAX VAR	OCCURF MIN AV	LENCES	ASSOC TO ASSOC	RELATIVE	LOCATION	A	NLUE/REMARKS
								-
		-						
				1				
			-					
	_			_				
			-	+				
				+				
		-	-			T		
				F				
				+				
MARKS:								

4.15

	_																4	. 1	6	
	PAGE OF PAGES																			
	DATE	ω																-		-
	ANALYST	VALU																		
70	ERAND DESCRIPTION																			
	EFERENCE OF	TYPE																		
	TEM NAME	NAME																		
	SYS	SER	PR		Π			 T				T	T				1			

CHAPTER 5

Operands

In order to document a logical solution to a problem, a means must be provided to identify and refer to data associated with that problem. In the Tabular Descriptive Language this means if provided by operands. An operand serves a function similar to that of a noun in an ordinary English language sentence. Operands, used in conjunction with operators (see Chapter 6), allow the various conditions and desired actions associated with the problem to be expressed in a simple, concise form.

<u>Types of Operands</u>: Operands in TDL are divided into two major classifications: Named Operands and Self-Describing Operands. Named Operands identify data associated with a particular information processing system. Their names do not indicate actual values, and therefore, they must be defined in the system in which they are used. Self-Describing Operands do not need to be described for each system and can be used universally, as actual values are specified.

Named Operands are classified in four major types: Information, Reference, Work, and System Segmenting Operands. The Operands used in the TDL System are as follows:

Named Operands

Information Operands

File Record Field

Reference Operands

Reference List Function Value List Arithmetic Expression Logical Expression Symbolic Expression Constant

Work Operands

Work Record Work Field

System Segmenting Operands

Procedure Decision Table

Self-Describing Operands

Literals Connotative Terms State Terms Descriptive Terms Value List Range

<u>Operand Description Sheets</u>: Three forms are used to describe operands: File Description, Report Description, and Reference Operand Description. The File Description form describes files, records, and fields within records. The Report Description is a standard printer layout sheet that describes both fixed and variable information appearing on a printed report. The Reference Operand Description describes reference lists, value lists, all types of expressions, and constants. Work operands and System Segmenting Operands are defined by their use in decision tables.

<u>Operand Names</u>: The name of an operand is composed of a group of characters. The first character of a name must be alphabetic, but the remaining characters can be any of the letters (A - Z), the numerals (0 - 9), or the special characters (#, \$, @, %, ¢, Δ , decimal point, or a blank). A name should be selected that is meaningful for the data described. The length of a name is not limited, but should be kept to a minimum without sacrificing readability or comprehension. Here are examples of operand names:

> STOCK NO. QUAN ON HAND DISCOUNT % DEPT NO. T 127

There are special rules for naming work operands and selfdescribing operands.

INFORMATION OPERANDS

Files

A file is a uniquely named group of one or more types of records. It has two other fundamental properties that must be described: medium and directional type. Medium is the method of recording information within a file. Examples are magnetic tape, paper tape, punched card, printed report, disk, inquiry station, core memory, transmission line, etc. While the medium of a file must be described, no specific input/ output or storage device is named. For example, the medium of a file can be indicated as punched card, but a particular card reading is not specified.

A file can be specified as having one of eight directional types: serial-input, serial-output, serial-intermediate, serial-retained, random-input, random-output, random-intermediate, or randomretained. The term "serial" indicates that only the next record of a file is available (e.g., tape or card). "Input" and "output" indicate whether the information comes into the system or is provided to other systems. "Intermediate" indicates that a file is used exclusively within a system, and is not retained for future use (a temporary file). Retained indicates that a file is available initially, but must be maintained for future operations either inside or outside the system (e.g., a master file). The system description does not imply that the actual program must keep files physically separate, or that a file cannot be split into two or more files. For example, a systems analyst can indicate that the medium for a file is magnetic tape. The programmer can place this file, and one or more other similar files, on the same reel of magnetic tape; or he can split the file, placing some of its records on one reel of tape, and the remaining records on another reel of tape. This need not concern the systems analyst, nor will it invalidate the system description.

For each input, retained, or intermediate file it is assumed that there is a single area in memory for each file large enough to contain the largest record of a file. For output files, it is assumed that an area has been reserved in memory that is sufficient to store one of <u>each</u> of the record types at the same time.

Records

A record is a uniquely named group of fields which belong to a file. The systems analyst's description of a record should not be considered to be a record layout, for he is describing the content of the record, not its format. A record may be read into or written from a system by use of the READ and WRITE operators (see Chapter 6).

Each record in a file is described on a separate File Description form. Thus, the description of a file is a collection of record descriptions. Printed reports, which are considered to be files, are described on a Report Description form, in addition to a File Description form.

However, the order in which fields appear on a line of a Report Description form is the order in which they are to appear on the actual report. The record corresponds to one or more printed lines. For any record except lines on a printed report, the programmer can arrange the fields of a record in the most efficient order.

All fields should be explicitly noted in describing a record. Because only one record from each input file is available at any one time, a means is provided to save such records for subsequent use. Either a work record can be created or an intermediate file can be used. When identical field names and descriptions apply to two different records, the File Description of one record can refer to the other record which has already been defined.

Fields

A field is a named item of information that has a value and is part of a record. A sub-field is a part os a field. Fields and sub-fields have the same characteristics and are both normally referred to as fields.

Because a field has value, the relative position of its characters is significant. A field value can be all numeric, all alphabetic, all special characters, or a combination of these. A field in either an intermediate or output record can have a pre-assigned value. The systems analyst must indicate this pre-assigned value.

When describing a field, the systems analyst should indicate the maximum size it can attain and whether its length can vary. If a

numeric field contains a decimal part, the systems analyst should describe its maximum decimal length.

The sign of a pre-assigned numeric field should be indicated. Also, the sign possibilities should be indicated for all other numeric fields. The systems analyst should indicate that a numeric field will always be zero or greater (+), zero or less (-), either positive or negative (E for either), or will be unsigned (U).

There should be no inconsistency between a field and any sub-fields which are part of it. A sub-field can be defined as part of a numeric field that is signed and contains integral and decimal parts. If this is done, the sub-field must be a continuous part of the field, and must have the same sign possibilities as the field. If a sub-field contains integer and decimal parts, its decimal position must be consistent with the position of the decimal point in the field.

The same field can occur more than once in a signle record. Such a multiple-occurring field can be assigned a single name, permitting the systems analyst to describe, just once in a decision table, the processing for each separate occurrence of the field.

Association Groups: To facilitate the preparation of decision tables for multiple occurrence fields, all fields on the record having a logical relationship to each other can be grouped and identified by an Association Group code.

CIELD MANAE	MUM	10	LENT	ГН	000	IRRE	NCES	2K
FIELD NAME	4RIC NA	ž	MAX	VAR	MIN	AVG	MAX	506
PART NO	A		07				01	
DESCRIPTION	A		25				01	
LOCATION CODE	N	U	02		01	05	10	A
LOCATION NAME	A		15		01	05	10	A
CLASS CODE	W	10	01		01	01	03	B
QTYON HAND	N	U	06		01	01	03	B
PRICE CODE	N	U	01		01	01	04	C
UNIT PRICE	W	+	04.2		01	01	04	C

In the preceding illustration there are three Association Group codes. CLASS CODE and QTY ON HAND are in the same group (B) because quantity on hand is always recorded for each class code.

Associated To: In addition, Association Group codes are utilized to indicate the logical hierarchy of the individual Association Groups to each other.

CICLO MANAC	NUM ALP	i	LENT	TH	occi	JRREN	ICES.	1 PS	18
FIELD NAME	BRIC.	N.	MAX	VAR	MIN	AVG	MAX	305	000
PARTNO	A		07				01	1-	
DESCRIPTION	A		25				01		
LOCATION GODE	N	U	02		21	05	10	A	
LOCATION NAME	A		15		01	05	10	A	1_
CLASS GODE	N	U	01	1	01	0!	03	B	A
ATY ON HAND	W	0	06	[01	01	03	B	
PRICECODE	N	U	04	1	31	01	04	C	
UNITPRICE	W	+	04.2		01	01	04	C	-

The group of fields in Association Group B are recorded a maximum 3 times for each group of fields in Association Group A. Similarly, the fields in Group A occur a maximum of 10 times for each record.

Sub-Fields: If a systems analyst wishes to refer to only part of a field, he must establish a sub-field and name it according to the rules for naming operands. For example, the six-digit, unsigned numeric field DATE can be divided into four sub-fields. One sub-field, defined as occupying the two left-most positions of the field, can be called DAY. A second sub-field MONTH can be defined as the third and fourth positions of the field. A third sub-field YEAR can be defined as the fifth and sixth positions of the field. A fourth sub-field MONTH AND YEAR DATE can be defined as the third through the sixth characters of the field.

One or more sub-fields can be specified for a field by indicating the positions of the field that form each sub-field. To do this, the positions of the field are numbered from left to right beginning with the number 1. To specify a sub-field, the positions of the field that form the sub-field are indicated in the File Description.

Even though one or more sub-fields are specified for a field, subfields for all parts of the field are not required. For example, only the first four positions of a ten-position field might be specified as a sub-field.

Sub-field names are unique within a record type. A sub-field can occur only once within a field. Thus, a field name need not be added to a sub-field name to qualify it. However, it may be necessary to append

a record name to a sub-field name when that sub-field occurs in more than one type of record.

Sub-fields can overlap one another within a field. A sub-field, however, cannot be part of one field and part of another field. A field can be an entire record and be divided into as many levels of sub-fields as needed.

REFERENCE OPERANDS

Reference operands have values defined through an operand description. These values will be present or will be computed at the time the operand is referred to in a decision table. Reference operands are described on a Reference Operand Description form.

Reference List Functions

A reference list is similar to a matrix or table of values. Each reference list function is related to one or two arguments. These arguments may be thought of as the vertical and horizontal axes of a tabular array. A reference list provides the systems engineer with a convenient form for describing a discrete set of values as functions of other values. An insurance rate table can be a reference list, with the function, rate, being dependent on the values of the two arguments, age and health.

Each reference list function value must be an explicit constant value. The argument values may be constants, ranges, or even names of value lists. Alphameric values are enclosed within guotation marks. Numeric values do not require quotes. If a range is used as an argument, the upper and lower limits of the range are separated by a hyphen. If any part of a range (upper or lower limit) is alphameric both values must be enclosed within quotes. The collating sequence specified for TDL determines whether a particular value falls within the range. Reference lists cannot be modified by action within decision tables.

To refer to a specific entry within a reference list this form is used:

Reference List Function Name (argument 1, argument 2). The reference list function name is the name of the function on the list. Argument 1 and argument 2 are designations of operands which may match the argement values of the list. Thus, a request for the rate for age 32 in the eastern section of the country can be specified as: RATE (32, "EAST"). If the rate to be determined is for the age and section of the country on an input record called APPLICATION, the form would be:

RATE (AGE: APPLICATION, SECTION OF COUNTRY: APPLICATION)
Value Lists

A value list is a group of two or more values or ranges of values that are identified by a single unique name. For example, a list of cities in which a company has warehouses can be called WAREHOUSE CITY. A field of an input record, with the name SHIPPING CITY, can be compared to the cities identified by the name WAREHOUSE CITY by the condition statement: SHIPPING CITY EQ WAREHOUSE CITY. If the specific value of the input field is "Albuquerque", the value list named WAREHOUSE CITY is examined to determine whether it contains the value "Albuquerque". A value list can only be used in condition statements; it has no use in action statements.

The values or ranges of values appearing on a value list can be either numeric or alphameric. Each value or range of values is separated by a comma. The upper and lower limits of a range are separated by a hyphen. A numeric value can be written directly but an alphameric value is enclosed within quotation marks. For consistency, if one part of a range is alphameric, then both parts should be in quotes. When ranges of values appear on a value list, the collating sequence determines whether any specific value falls within one of the ranges specified.

An example of a value list is:

+4, -6.2, "RR" - "3A"

A condition statement comparing a field with this value list will result in the reply "yes" if the field value is +4, -6.2, or in the range from RR through 3A.

Arithmetic Expressions

An arithmetic expression is a named operand whose value is an arithmetic function of one or more operands. The value of an expression is always calculated at the time of its use in a decision table using current values of the operands forming the expression. Each time the expression name appears in a decision table, its value is recalculated. An arithmetic expression must be given a unique name which will represent a numeric value. There is no practical limit to the number of operands and operators that may appear in an arithmetic expression. These numeric operands can be fields, constants, literals, reference list functions, descriptive and work fields. The five operators used in the expression are:

- + Addition
- Subtraction
- * Multiplication
- / Division
- ** Exponentiation

In addition to these five basic operators used in arithmetic expressions, special mathematical functions like SIN, COS, TAN, SUM, NEG, POS, and ABS may be used. These are explained in Chapter 7.

Some examples of arithmetic expressions follow:

C is the expression name for A + B, indicating that the sum of operands A and B produce the value represented by operand C.

D represents A + B / C, indicating that the result of adding operand A to the quotient of B and C is the value represented by operand D.

C represents A ** B, indicating that the value of A to the Bth power is represented by operand C.

An arithmetic expression is evaluated from left to right, using this hierarchy of the five basic operators: first--exponentiation, second--

multiplication and division, and third--addition and subtraction. The second example shows the quotient would be formed first, then the sum.

Parentheses can be used to establish the order and effect of any operator when the normal hierarchy is not desired. Operations within parentheses are always evaluated first. If there are parentheses within parentheses, the inner most operations are carried out first, then the next until effectively the parentheses are cleared and the normal hierarchy applies. An equal number of left and right parentheses must be used. As indicated above -- A + B / C would indicate that B is first divided by C and then added to A -- however, (A + B) / C would cause A to be added to B and the sum divided by C. All operands, operators, functions, and parentheses can be separated by one or more blanks to make an expression more readable.

If an arithmetic expression is only to be used for one decision table it is called a local arithmetic expression and may be described in the remarks portion of that table. Its name need only be unique for that decision table. Local arithmetic expressions have the same characteristics as other arithmetic expressions.

Logical Expressions

A Logical expression is an ordered combination of operands and operators that can be evaluated to be either true or false. Any operand, except the name of another expression (arithmetic, logical, or symbolic), can be used in a logical expression. The operators that can be used in a logical expression are:

1. The three Logical Operators: AND, OR and NOT

2. The six Relational Operators:

LR < lesser than

EQ = equal to

GR > greater than

LE \leq lesser than or equal to

NE /= not equal to

GE >= greater than or equal to

3. The five Arithmetic Operators: + - * / **

4. The two State Operators: IS and NT

Examples of logical expressions are:

MARITAL STATUS = 1 OR MARITAL STATUS = 2 OR MARITAL STATUS = 4 - 7

PAY EXPERIENCE = "OK" AND BALANCE LE 10000.00

The logical operator OR is interpreted to mean either or both; that is, it is inclusive. The operator AND means that the two conditions joined by the AND must both be true for the logical expression to be true. The operator NOT means the negation of a condition. For example, NOT (A AND B) means if either A is not true or B is not true then the expression is satisfied. The expression NOT (A OR B) means both A and B must be not true for the expression to be satisfied. Parentheses can be used effectively to indicate meaning and improve readability of logical expressions. If parentheses are used, an equal number of left and right parentheses are required. The hierarchy of the logical operators is: AND then OR then NOT. For example, A AND B OR C is interpreted as (A AND B) OR C.

Great caution should be exercised in the use of logical expressions since they can readily lead to errors. It is suggested that decision tables normally be used instead of such expressions. Even where a logical expression seems necessary or convenient, it may be helpful to work out the logic with a decision table. Local logical expressions may be used through definition in the remarks portion of a decision table.

Symbolic Expressions

A symbolic expression is a means for operating on values with other than arithmetic operators. The Reference Operand Description can be used to specify that ITEM IDENTIFICATION is the name for JOIN (LINE CODE, STOCK NUMBER), which means that the value of ITEM IDENTIFICATION is determined by joining the value of LINE CODE and STOCK NUMBER, in that order. Any number of fields, constants, literals, and reference list functions can be joined in a symbolic expression; the operands are written after the operator JOIN within parentheses in the order in which the values are to be put together; the operands are separated by commas. Local symbolic expressions may be described in the remarks portion of a decision table.

Constants

A constant is a uniquely named operand whose value cannot be changed by operation of the system. There is no limit to the size of a constant. Constants are described on a Reference Operand Description sheet. A constant may represent a signed or unsigned integer or decimal number or an alphameric value.

WORK OPERANDS

Work operands are used for the temporary storage of a value or a set of values (fields or records). Each work operand must have a different name but all of them are preceded by the letters WORK. No formal definition is required for either of the two types of work operands: work records and work fields.

Other than its name, a work operand assumes all the characteristics of any operand that is placed in it; these characteristics remain unchanged until a new operand is placed in it. Thus, each work operand always contains that which was placed in it most recently by an action in a decision table. For example, at one time a work operand may contain a ten-character numeric field; at a later time it may contain a fifteencharacter alphameric field. If a systems engineer desires to delete the contents of a particular work operand, he may do so by placing blanks in that work operand.

Care should be exercised by the systems engineer in the use of work operands. He must be careful not to destroy data that will be required at a later time. It is assumed that there will be sufficient space to contain the contents of all work records and work fields.

Work Records

A work record will contain as many fields as the record which is moved into it. When the contents of a record are moved into a work record, the work record will contain all the field values (and names) associated with the original record. These fields will have the same names as when they appeared in the original record. For example, if a record named STOCK STATUS is moved into a work record named WORK RECORD 1, each field in the work record gets the same name except it must be qualified by WORK RECORD 1. The systems engineer may refer to a field as: STOCK NUMBER: WORK RECORD 1. If later TRANSACTION is moved to WORK RECORD 1 then the fields in TRANSACTION can be referenced by their name qualified by WORK RECORD 1.

Work Fields

Each work field will assume the characteristics and the value of what is moved to it. These characteristics and value will persist until something else is placed in the work field. Each work field name will be unique and begin with the letters WORK. Work fields are not defined in any operand description.

SYSTEM SEGMENTING OPERANDS

Two types of operands are defined on Decision Table sheets and need not be defined elsewhere. These operand names are procedure names and decision table names or identifications. Their primary purpose is to allow a system to be segmented into logical portions. These tables and procedures can then be referred to from other decision tables to provide sequence control.

Procedure Names

A procedure is a group of one or more decision tables that form a logical segment of an information processing system. The procedure need not be developed to refer to it in a decision table. This permits a system specification to be developed at a level above detail analysis. A payroll system could call for procedures to determine FICA deduction, Federal Withholding Tax, etc., assuming that the detailed procedures would be specified at a later date.

The first table of a procedure is indicated by writing FIRST in the Table Type section of the Decision Table form. A procedure may be open or closed depending upon how it will be used in the system. A procedure name must begin with a letter and must always be preceded by the word PROCEDURE, when it is referred to. No two procedure names can be identical. A procedure name should be indicative of the functions performed in the decision tables which comprise it.

Decision Table Names

A decision table is a logical association of conditions and actions. Since it usually takes many decision tables to make up a system, it is necessary to be able to refer from one decision table to another. This is done by using a decision table name or a decision table identification. A decision table name is assigned using the rules for name construction and must be unique within a system. Decision table identifications are three characters in length with the first character numeric; the other two characters may be either numeric or alphabetic. Each decision table will be assigned a different identification.

SELF-DESCRIBING OPERANDS

Self-describing operands can be used universally without being specifically defined for a particular system. There are six types of self-describing operands: literals, connotative terms, state terms, descriptive terms, value lists, and ranges. These permit direct consideration of factor values instead of creating names and then referring to their values indirectly. They are specially useful in handling relationships or standards which are not apt to change.

Literals

A literal designates its own value rather than being the name of a value. There are two types of literals: numeric literals and alphameric literals.

A numeric literal may be a signed or unsigned fixed point number or a signed or unsigned floating point number. There are no restrictions on the length of a literal.

A fixed point number may consist of all integers, which may be preceded by a sign. Unsigned values are assumed positive if used in arithmetic. Examples of integers are: + 42 - 689

7254

A fixed point number may also consist of integer and decimal digits. It may be preceded by a plus or minus sign; unsigned, the value is assumed positive. A decimal point is shown between the integer and decimal part of the number. Examples are:

> + 42.6 - 68.253 0.567 .793

A floating point literal is constructed like a fixed point literal, except that it also contains an exponent, representing a power of ten. This exponent can be indicated as shown in these examples:

Floating Point Notation	Interpretation
14F + 7	14×10^{7}
11.7F - 3	11.7 x 10 ⁻³
+ 14.269F21	+ 14.269 x 10 ²¹
- 11F - 15	- 11 x 10 ⁻¹⁵

An <u>alphameric literal</u> consists of any combination of numeric, alphabetic, or special characters. To differentiate it from a name or number it is enclosed within quotation marks. A blank is a value character, designated by an underline. A single blank between two sets of characters can be left as a space. Quote marks in literals may lead to confusion so these should be handled by constant names and definitions. Examples of alphameric literals are:

> 'JANE DOE___SECRETARY' '+ - * / **' '369A4' '+ 12.57'

Connotative Terms

A connotative term is one whose value is directly implied by its name. The following is a list of the acceptable connotative terms:

ZERO or ZEROS BLANK or BLANKS NINE or NINES

ALL 'any alphameric literal'

The first three connotative terms represent respectively one or more zeros, one or more blanks, and one or more nines. The connotative term ALL is used to represent repetitions of the same character or group of characters. The alphameric literal may be any character or group of characters; it is enclosed in quotes. ALL '7' would represent one or more 7's. ALL 'ABC' would represent one or more repetitions of ABC, such as ABCABCAB.

State Terms

A state term represents a characteristic or class of values. The

POSITIVE or POS NEGATIVE or NEG NUMERIC or NUM ALPHABETIC or ALPHA STATUS PRESENT END OF FILE or EOF

The state term POSITIVE relates to all numeric values greater than zero, NEGATIVE to all numeric values lesser than zero. NUMERIC pertains to a field containing only numbers or blanks; ALPHABETIC fields may contain only letters or blanks.

The state term STATUS can be related only to a logical expression name. It indicates that the value of the associated logical expression is true or not true depending upon the operator used.

Descriptive Terms

A descriptive term indicates by means of its commonly understood meaning the value represented by that name. It is always enclosed within number symbols (#). Examples are:

CURRENT YEAR

MONTH END CLOSING DATE

Descriptive terms represent values which change. In this sense they differ from connotative terms whose values are always the same. A

descriptive term is not defined by the systems engineer since he can assume that its correct value will be provided by the programmer where required.

Value List

A value list consists of two or more literals or ranges. A value list can be used directly in a decision table as an operand. The same rules apply to noting values in a value list as when used in a Reference Operand Description. However, for convenience, each value may be put on a separate line without commas in between; similarly a range may be put on a single line.

An example:

42

Another example:

"MIA" "SFO" "LAG" "LAX" "JOL"

Range

A range is similar in use to a value list, except that it contains but

a single range of literal values. The upper and lower limits of the range are separated by a hyphen. Ranges are indicated as a distinct type of self-describing operand because, unlike most value lists, they can be written on a single line of a decision table. The rules for writing each individual literal (upper or lower limit) of a range operand are the same as those for writing any other type of literal except that if one boundary of the range is alphameric both must be enclosed in quotes. The range is considered inclusive at both ends.

CHAPTER 6

Decision Table Operations

Operands are the nouns of a language; operators are the verbs. Operators combined with operands permit dynamic procedures to be described by the systems analyst.

Operators are generally intended to work at a system analysis level rather than at a programming level. Thus, certain assumptions and statements will be made concerning automatic processes that may in fact be quite complex for a particular computer.

Only actions can change the values of operands. These actions are usually taken only if specified conditions are satisfied. Two major types of operations provide for these basic capabilities: testing a set of conditions and executing resultant actions.

Conditions -- to interrogate values and states

Actions -- to change values, provide input/output and control sequence

Because pyramiding of all conditions within a single decision table makes a system description unwieldy, a means is provided to describe systems using many tables. Action operators are available to control the sequence of table execution.

Conditions

Conditions can be determined to be either true or false. Conditions are the means for asking questions; as such they can be used to control
the actions which are to be executed.

A condition is true or false, depending on its operator and the value of its operands. Each individual condition should be stated as simply as possible to avoid inaccuracy and difficulty of interpretation; complex interactions among conditions are shown through position in a decision table. Those within the same rule bear a logical AND relationship to one another. Individual rules express an OR relationship.

There are two major categories of conditions: Value Relational and State Relational. Each of these is discussed under the headings:

OPERATORS

OPERANDS

COMPARISON RULES

USAGE

MECHANICS

Value Relational Conditions

All value relational conditions are constructed as follows: form: Operand 1 Condition Operator Operand 2 example: QUANTITY ON HAND LE 1000 read as: IF QUANTITY ON HAND is less than or equal to 1000

OPERATORS

The following operators may be used in a value relational condition:

Symbolic <u>Mnemonic</u>		Read As	
=	EQ	is EQual to	
7	GR	is GreateR than	
>=	GE	is Greater than or Equal to	
<	LR	is LesseR than	
ζ=	LE	is Lesser than or Equal to	
/=	NE	is Not Equal to	

An operator can be written in symbolic or mnemonic representation.

OPERANDS

These operands can be used either as operand 1 or operand 2 of a value relational condition:

Field Name Reference List Function Name Value List Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term Value List Range

COMPARISON RULES

Since an operand may represent either a numeric or an alphameric value of different lengths it is necessary to state what happens under various circumstances. Value relational conditions usually require that both operands be numeric or both be alphameric.

<u>Numeric</u>: If both operands are numeric, comparison follows normal algebraic rules: (1) Decimal points are aligned, (2) High-order zeros are added to the shorter integer operand, (3) Low-order zeros are added to the shorter decimal operand, (4) All negative values are less than zero, (5) All positive values are greater than zero, (6) For negative values larger absolute magnitude is considered smaller value, (7) Comparisons are permitted between fixed and floating point numbers by considering floating point number to be converted to fixed point notation, (8) Integers are treated as decimal numbers with the decimal point to the right of the least significant position.

<u>Alphameric</u>: If both operands are alphameric, the comparison follows usual practice: (1) Line up left-most positions, (2) Add low order blanks to shorter operand (3) Comparison is performed position by position from left to right, using character collating sequence, (4) The first unequal character position determines the result of the comparison.

<u>Mixed</u>: Comparisons between numeric and alphameric operands are permitted and are interpreted on this basis: (1) The rules of alphameric comparison apply. The numeric value is considered an alphameric operand, (2) The sign and decimal point (if any) of the numeric value are not considered in the comparison, (3) Low order blanks are added to the shorter operand, (4) Floating point numbers are converted to decimal notation before comparison.

USAGE

Special combination rules apply to the use of the operands and operators in value relational conditions.

<u>Field Name</u>: A Field Name can be used with any value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

A Field Name may be used only with the EQ and NE operators and these operands:

> Value List Name Value List Range

<u>Reference List Function Name</u>: A Reference List Function Name can be used with any other value relational operator and these operands: Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

It can be used only with EQ and NE operators and these operands:

Value List Name Value List Range

If the arguments of a Reference List do not appear in the reference list the comparison will fail regardless of what value it is compared with. For example,

	PAY ('CARPENTER', 7)	LE	4.50
and	PAY ('CARPENTER', 7)	GR	4.50

will both fail if the Reference List skips from ('CARPENTER', 6) to

('CARPENTER', 8).

Value List Name: A Value List Name can be used only with the

EQ and NE operators and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term The EQ comparison is considered true if any item in the Value List matches the value of the other operand. The NE comparison is considered true if no item in the Value List matches the value of the other operand.

<u>Arithmetic Expression Name</u>: An Arithmetic Expression Name can be used with any value relational operator and these operands:

> Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

It can be used only with EQ and NE operators and these operands:

Value List Name Value List Range

The value of the arithmetic expression is determined dynamically at the point of usage and redetermined each time it is used. Since an arithmetic expression always develops a numeric value, this should be considered when selecting the other operand. If it is alphameric, the results may be in error.

<u>Symbolic Expression Name</u>: A Symbolic Expression Name can be used with any value relational operator and these operands: Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

It can be used only with EQ and NE and these operands:

Value List Name Value List Range

The value of a symbolic expression is determined dynamically at the point at which it is used and redetermined each time it is used. Since the value can be either an unsigned integer or an alphameric value, the comparison rules are determined by the other operand. When compared with another symbolic expression, the rules for alphameric comparison will apply.

<u>Constant Name</u>: A Constant Name can be used with any value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Work Field Name Descriptive Term

It can be used only with EQ and NE with a Value List Name.

<u>Work Field Name:</u> A Work Field Name can be used with any value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

It can be used only with EQ and NE and these operands:

Value List Name Value List Range

A Work Field can only take on value that has been assigned to it; therefore, a comparison with a Work Field Name which has never been assigned any value will be treated as an undefined comparison and will fail every type of test.

Literal: A Literal can be used with any value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Work Field Name Descriptive Term

It can be used only with EQ and NE with a Value List Name. The value type is interpreted as written. An unsigned integer is handled in accordance with the other operand's type.

<u>Connotative Term</u>: A Connotative Term can be used with any value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name (unsigned integer only) Symbolic Expression Name Work Field Name Descriptive Term

It can be used only with EQ and NE with a Value List Name. The length and value type are interpreted by the other operand.

Descriptive Term: A Descriptive Term can be used with any

value relational operator and these operands:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

It can be used only with EQ and NE and these operands:

Value List Name Value List Range

A Descriptive Term takes on the form of the operand to which it is compared. For example,

MONTHLY CLOSING DATE # EQ 12/31/61

or # MONTHLY CLOSING DATE # EQ DEC 31, 1961

are both acceptable.

<u>Value List and Range</u>: A Value List or Range can be used directly in the decision table only with the EQ and NE operators and these operands: Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Work Field Name Descriptive Term

The EQ comparison for a Value List is true if any item in the Value List matches the value of the other operand. The NE comparison is true if no item in the Value List matches the value of the other operand.

The EQ comparison for a Range is true if the other operand is greater than or equal to the lower bound of the Range while less than or equal to the upper bound. Unsigned integer ranges are interpreted as alphameric or numeric, depending upon the other operand value type.

MECHANICS

A value relational condition can be expressed in either limited or extended entry form. These are the permitted arrangements:

ST	JB	ENTRY
Operand 1	EQ. = Operand 2 NE, /= GR. > LR. < GE. >= LE. <=	Y, N, . or blank
Operand 1	EQ, = NE, /= GR, > LR, < GE, >= LE, <=	Operand 2, or blank
Operand 1		EQ, = Operand 2, or blank NE, /= GR, > LR, < GE, > = LE, < =
Operand 1	VS Operand 2	EQ. = , or blank NE, /= GR, > LR, < GE, >= LE, < =

The first form shown is limited entry with Y meaning yes (or true), N meaning no (or false) and ". " or blank meaning not pertinent or impossible. The second form is extended entry using a common value relational operator for all rules expressed in the entry columns. The third form is extended entry with different relational operators for each rule. In both of these ". " or blank means not pertinent. The fourth form is a special type of limited entry where VS (interpreted as Versus) is used to represent the specific value relational operator

6.12

in each rule. Each rule is read as though the operator in that rule were substituted for the VS.

State Relational Conditions

All state relational conditions are constructed as follows:

form:	Operand 1	Condition ope	rator	Operand 2
example:	AVAILABLE BA	ALANCE	IS	POSITIVE
read as:	IF AVAILABLE	BALANCE IS	POSITIVE	

OPERATORS

Either of two state relational operators can be used in a state relational condition:

Mnemonic	Read as	
IS	IS	
NT	is NoT	

There are no symbolic representations for these operators. The mnemonic representatives should be used.

OPERANDS

In a state relational condition, operands 1 and 2 may be interchanged, but readability is often impaired. In general, operand 2 should be the state term, however, operand 1 or operand 2 can be any of these operands: File Name Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Work Field Name

When Operand 2 is one of these state terms:

POSITIVE NEGATIVE NUMERIC ALPHABETIC PRESENT END OF FILE or EOF

Operand 1 can be the State Term STATUS if Operand 2 is a Logical Expression Name.

COMPARISON RULES

Comparison rules are determined by the particular State Term used in the state relational condition. Generally, an incorrect match of value class (e.g., alphameric value against POSITIVE) results in failing the test.

USAGE

The rules for usage of the various operands are discussed for each permissible State Term. Multiple occurring Field Names may be used wherever a Field Name can be used except that they must only appear in a closed decision table under association control from a preceding DO statement. <u>Positive</u>: The State Term POSITIVE can only be used with numeric valued operands. An operand is POSITIVE if its algebraic value is greater than zero. The following operands can be used when they represent a numeric value:

Field Name (per Operand Description)

Reference List Function Name (per Operand Description)

Arithmetic Expression Name

Symbolic Expression Name (if composed only of numeric valued operands)

Work Field Name (per description or characteristics of previous field value which entered the work field)

All unsigned numeric values except zero will be considered POSITIVE. If a Reference List Function Name is not defined for the arguments indicated, the comparison will fail.

<u>Negative</u>: The State Term NEGATIVE can only be used with numeric valued operands. An operand is NEGATIVE if its algebraic value is less than zero. The following operands can be used when they represent a numeric value:

Field Name (per Operand Description)

Reference List Function Name (per Operand Description)

Arithmetic Expression Name

Symbolic Expression Name (if composed only of numeric valued operands)

Work Field Name (per description or characteristics of previous field value which entered the work field)

Unsigned numeric values are considered positive and will fail a test for NEGATIVE. If a Reference List Function Name is not defined for the arguments indicated, the comparison will fail.

<u>Numeric</u>: The State Term NUMERIC can be used with operands defined as numeric or alphameric. An operand is NUMERIC if all of its characters are from the set 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, or blank. The following operands can be used:

> Field Name Reference List Function Name Symbolic Expression Name Work Field Name

If a Reference List Function Name is not defined for the arguments indicated, the comparison will fail.

<u>Alphabetic</u>: The State Term ALPHABETIC can only be used with alphamerically defined operands. An operand is ALPHABETIC if all of its characters are letters (A, B, C, ..., X, Y, Z) or blank. The following operands can be used where they represent alphameric values.

Field Name (per Operand Description)

Reference List Function Name (per Operand Description)

Symbolic Expression Name (if composed only of alphameric valued operands)

Work Field Name (per description or characteristics of previous field values which entered the work field)

<u>Status:</u> The State Term STATUS can be used with a Logical Expression Name only. STATUS is written in the Operand 1 position. The logical expression name is written as Operand 2. For example:

STATUS IS MARRIED

The comparison is satisfied if the value of the logical expression MARRIED is true. It is not satisfied if the value of the logical expression is false.

<u>Present</u>: The State Term PRESENT can be used with a Reference List Function Name only. If the function is defined for the arguments given, then the function is considered PRESENT. If the function is not defined for these arguments, the function is not considered PRESENT. PRESENT can be used to prevent missing factors in a reference list from destroying a legitimate calculation; it is also useful to check existence of special circumstances. For example,

SPECIAL FARE ('BOSTON', 'NEWARK') IS PRESENT might be used to determine if a regular mileage-based fare should be used or if there is a special fare agreement.

End of File: The State Term END OF FILE can be used with a File Name only. File name IS END OF FILE is satisfied when there are no more records available on the file. If another record is available, the test will fail. This State condition can be used for serial files only. Also, END OF FILE (can be abbreviated EOF) is only pertinent to input, retained and intermediate files when they are being treated as input to the system. For example,

TRANSACTION IS EOF

will only be satisfied when there are no more records in the TRANSACTION file.

MECHANICS

A state relational condition can be expressed in either limited or extended entry form. These are the permitted arrangements:

ST	UB	ENTRY
Operand 1	IS POSITIVE NT NEGATIVE NUMERIC ALPHABETIC PRESENT END OF FILE EOF	Y, N, . or blank
STATUS	IS Operand 2 NT (Logical Ex- pression Name)	Y, N, . or blank
Operand 1	IS NT	POSITIVE , or blank NEGATIVE NUMERIC ALPHABETIC PRESENT END OF FILE EOF
STATUS	IS NT	Operand 2 ,. or blank (Logical Expression Name)
Operand 1		IS POSITIVE ,. or blank NT NEGATIVE NUMERIC ALPHABETIC PRESENT END OF FILE EOF
STATUS		IS Operand 2 ,. or blank NT (Logical Expression Name)

The first two forms shown are limited entry. The remaining forms are extended entry.

Actions

Actions change values, specify input/output, and control the sequence within a set of decision tables. Within a single table, each action is executed in the order written, and must be logically completed before a subsequent action is initiated.

Actions are stated as commands, and are indicated by their position in a decision table. The three major categories of actions are: assignment, sequence control, and input/output. These are discussed under the headings:

> OPERATORS OPERANDS SPECIAL RULES USAGE MECHANICS

Assignment Actions

The construction of an assignment action is:

form:	Action	operator	Operand 1	Action Conr	nector Operand 2
example:	MOVE	EMPI	OYEE NAM	e to	CHECK NAME
read as:	Assign name.	the value	of employee	name to the	field called check

The following operators can be used in assignment actions:

MOVE

SET

With MOVE these action connectors can be used:

<u>Symbolic</u>	Mnemonic	Read as
->	TO	inTO
+	PL	PLus operand 2 into operand 2
-	SU	SUbtracted from operand 2 into operand 2
*	MU	MUltiplied by operand 2 into operand 2
. /	DI	DIvided into operand 2 into operand 2

With MOVE, Operand 1 specifies the sending value and Operand 2 specifies the receiving area.

With SET these action connectors can be used:

Symbolic	Mnemonic	Read as
-	EQ	EQual to
+	PL	PLus operand 2 into operand 1
-	SU	SUbtract operand 2 into operand 1
*	MU	MUltiplied by operand 2 into operand 1
1	DI	DIvided by operand 2 into operand 1

With SET, Operand 1 specifies the receiving area and Operand 2 specifies the sending value. With both MOVE and SET, either the

symbolic or mnemonic representations of the action connectors can be used.

OPERANDS

These operands can be used as the sending operand (Operand 1 with MOVE or Operand 2 with SET).

Record Name Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Record Name Work Field Name Literal Connotative Term Descriptive Term Special Arithmetic Expression

These operands can be used as the receiving operand (Operand 2 with MOVE and Operand 1 with SET).

Record Name Field Name Work Record Name Work Field Name

SPECIAL RULES

Since an operand may represent either numeric or alphameric values of different lengths, it is necessary to state what happens under various circumstances. Usually, both operands are numeric or both alphabetic. <u>Numeric</u>: If both operands are numeric, the assignment of value is controlled by the receiving operand. When both operands have the same decimal and integer length, the function of the MOVE and SET operators is apparent. If the sending and receiving operands have different decimal and integer lengths, the sending field is rounded from the right or zeros are added to the right until the decimal lengths correspond, and is truncated from the left or zeros are added to the left until the integer lengths correspond.

If the sending operand is in floating-point notation, but the receiving operand is fixed point, the sending operand is first converted to fixed point. When both operands are in floating-point notation, the floating point format is retained. If the number of positions in the exponent and/or mantissa are different, the rules for length stated for fixed point apply. If a fixed point operand is moved into a floating point operand, it is modified to retain the most significant positions and is then converted to floating point.

On arithmetic MOVE or SET actions, all sending operands are operated upon in fixed or floating as per receiving operand, and the result after the arithmetic operations is adjusted to fit the length and decimal restrictions of the receiving operand.

<u>Alphameric:</u> If both operands are alphameric, the assignment of value is controlled by the receiving operand. When operands are of the same length, the function of the MOVE and SET operators is apparent.

If the sending operand is shorter, it is left justified when moved into the receiving operand, and low-order blanks are added to fill the receiving operand. If the sending operand is longer before the MOVE, it is truncated from the right until its length is the same as the receiving operand.

<u>Mixed</u>: Alphameric valued operands can be moved into numeric valued operands only if the actual characters moved are all numeric. In this case, the sending operand is treated as an unsigned integer and the rules for numeric assignment are followed. Numeric valued operands which are unsigned integers can be moved into alphameric valued operands and the rules for alphameric assignment are followed.

When records are assigned to other records, <u>only</u> fields with corresponding names are moved. Each field moved follows the rules that apply for individual field assignment.

If the receiving operand is a report Field Name that is to be edited, it is assigned on a non-edited basis. The editing occurs when the WRITE command is executed.

USAGE

A number of special rules apply to the selection of specific operands to be used with various connectors. Therefore, specific usage is discussed under four categories: Individual value assignment, corresponding value assignment, two-operand arithmetic assignment, and

6.23

multiple-factor arithmetic assignment.

Individual Value Assignment: This is the basic operation which assigns one value to a Field Name or Work Field Name. The sending operand can be:

> Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Connotative Term Descriptive Term

The four forms are:

MOVE sending operand TO receiving operand MOVE sending operand \rightarrow receiving operand SET receiving operand EQ sending operand SET receiving operand \leftarrow sending operand

If a Reference List Function Name is undefined for the argument values specified, then no assignment takes place; that is, the receiving operand is not changed.

Assignment to a Work Field Name makes the work field take on

all the properties of the sending field, except its name.

The sending operand is never affected by the assignment.

<u>Corresponding Value Assignment</u>: This operation assigns values whose Field Names correspond from one record to another. The sending operand and the receiving operand must be a Record Name.

The four forms are:

MOVE sending operand TO receiving operand MOVE sending operand \rightarrow receiving operand SET receiving operand EQ sending operand SET receiving operand \leftarrow sending operand The only fields assigned are those whose names are the same in both records. All rules for individual value assignment apply. The sending record is not affected by the assignment.

Note: If the receiving operand is a Work Record, <u>all</u> fields in the sending record are assigned to the work record and their original names are retained. The original names can be used, but they must be qualified by the Work Record Name.

<u>Two-Operand Arithmetic Assignment</u>: This operation provides for two-operand arithmetic when a specific receiving operand is to be increased, decreased, etc. The sending operand can be:

> Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Descriptive Term

The receiving operand can be:

Field Name Work Field Name

MOVEA+BmeansA plusBintoBMOVEA-BmeansBminusAintoBMOVEA/BmeansAtimesBintoBMOVEA/BmeansBdivided byAintoBSETA+BmeansAplusBintoASETA-BmeansAminusBintoASETA*BmeansAtimesBintoASETA/BmeansAdivided byBintoA

Multiple Factor Arithmetic Assignment: This operation provides

greater facility in arithmetic operations without using arithmetic

Work Field Name. The sending operand can only be a Special

Arithmetic Expression.

The forms are:

MOVE sending operand TO receiving operand MOVE sending operand --> receiving operand SET receiving operand EQ sending operand SET receiving operand --> sending operand

The factors in a special arithmetic expression can be:

Field Name Reference List Function Name Arithmetic Expression Name Symbolic Expression Name Constant Name Work Field Name Literal Descriptive Term

The receiving operand can be:

Field Name Work Field Name

Any reasonable number of factors can be used with special

arithmetic connectors between each of the factors. The rules are:

Connector is + (plus)
Connector is - (minus)
Connector is * (times)
Connector is / (divided by)--only a single division permitted
Connector is ** (exponentiation)--only a single exponentiation permitted

Parentheses cannot be used. The computation takes place from top to bottom except that exponentiation always takes place first, then multiplication and division, then addition and subtraction. An assignment action can be expressed in either limited or extended entry form. The following arrangements are permitted for two-operand arithmetic.

		STUB			ENTRY
I	MOVE	Operand 1	TO + - * /	Operand 2	X,. or blank
S	SET	Operand 1	EQ + - *	Operand 2	X,. or blank
I	MOVE	Operand 1	TO + - * ∕		Operand 2,. or blank
ŝ	SET	Operand 1	EQ + - * /	ξ.	Operand 2 , . or blank
I	MÖVE	Operand 1			TO Operand 2,. or blank + - * /
\$	SET	Operand 1			EQ Operand 2,. or blank + - * /

The first two forms shown are limited entry, the others are extended entry.

With multiple factor arithmetic, the sending operand is more complex, but the assignment operations are limited to a basic MOVE <u>sending operand</u> TO <u>receiving operand</u>, or SET <u>receiving operand</u> EQ <u>sending operand</u>. These can also appear in limited or extended entry form as shown.

		STUB	ENTRY		
	MOVE + - ±	factor a factor b factor c factor n	ТО	Operand 2	X,. or blank
	SET	Operand 1	EQ + : +!	factor a factor b factor c factor n	X,. or blank
	MOVE * *	factor a factor b factor c	TO		Operand 2,. or blank
	SET	Operand 1	EQ		factor a ,. or blank * factor b * factor c
	MOVE	factor a factor b			TO Operand 2,. or blank
	SET	Operand 1			EQ factor a , . or blank / factor b
	MOVE **	factor a factor b			TO Operand 2,. or blank
0	SET	Operand 1			EQ factor a , or blank ** factor b
	the second se	THE OWNER AND INCOME.		and the second se	

6.28

The first two forms are limited entry, the other six are extended entry.

Sequence Control Actions

Sequence control actions are constructed as follows:

form:	Action Operator	Operand 2
example:	GO TO	UPDATE

read as: Transfer control to the decision table called UPDATE. Notice that there is only one operand, and this always appears in the Operand 2 position (see Mechanics).

OPERATORS

There are two sequence control operators:

GO TO

DO

These operators are used to transfer control to a specified decision table or procedure. GO TO transfers control permanently. DO transfers control, but control is later returned to the originating table.

OPERANDS

The following operands can be used:

PROCEDURE Procedure Name Decision Table Name Decision Table Identification

SPECIAL RULES

The last action in each rule of an open decision table except table STOP must be a GO TO. A table can GO TO or DO itself. All tables are entered at the beginning; therefore a specific rule or action cannot be selected by a GO TO or DO action.

USAGE

GO TO and DO provide the means for a complex logical problem to be subdivided into manageable segments. Theoretically, they can be eliminated by writing just one decision table for a job. However, experience indicates that effective system analysis and description occurs when the problem is subdivided into smaller parts.

<u>GO TO Sequence Control Action</u>: The GO TO action can only be the last action in a decision table rule. There must be a GO TO action in each rule of an open table, which transfers control to the table or procedure specified. A rule of a closed table may have a GO TO action but this is usually not the case, as the return path to the originating table is destroyed.

The three operands that can be used are:

PROCEDURE Procedure Name Decision Table Name Decision Table Identification

The first table of a procedure is entered when control is transferred via the GO TO action. The first table of a procedure is identified by

writing FIRST in the Table Type area of the Decision Table form. The initial table of the procedure must be an open table.

DO Sequence Control Action: A DO action can appear in either an open or a closed decision table. It always transfers control to a closed table or procedure, but retains the return path to the originating table. After carrying out the logic in the closed table referenced, control is transferred back to the originating table to the action following the DO. A table may DO itself; there may be any number of levels in a series of DO tables; that is, table 1 may DO table 2 which may DO table 3, etc. Eventually the DO path is retraced to the original table, unless there has been an intervening GO TO action which destroys the return path. Three operands may be used with DO:

> PROCEDURE Procedure Name Decision Table Name Decision Table Identification

If a procedure is referenced, the first table in the procedure is entered and it must be a closed table. In fact, all tables in a procedure to be referred to by a DO must be closed. The FIRST table then may call upon other closed tables which make up the procedure. No GO TO actions may appear in the procedure unless it is desired to leave the procedure without returning.

Association control provides a means of handling multiple recurring fields or associations of fields.

6.31

Operand 2 can have either of two forms:

Decision Table Name or Identification

PER ASSOC I

PER ASSOC J

etc.

I and J represent non-related association codes indicated in the File Description for multiple recurring fields.

Decision Table Name or Identification

PER ASSOC I

PER ASSOC J

etc.

I and J represent related association codes indicated in the File Description for multiple recurring fields.

In the first form (PER ... PER), the association values indicated are all set to 1 to start with. The first time through the specified closed table, the first of the association fields is used for conditions and actions. After execution of the table, and when there are more occurrences of all the fields in the associations, association values are incremented by 1 (set equal to 2 the second time around). The next time through the same closed table, the second occurrence of the association fields is used. This continues until one of the associations runs out of recurring fields. This is a very powerful format allowing many independent fields or groups of fields to be varied simultaneously. However, it can lead to errors if not closely analyzed. Association values can be directly analyzed in conditions in a table just as any other Field Name, but then, values cannot be changed by MOVE or SET. A table can be repeated for each value of the association until completion, but particular conditions can inhibit actions except under controlled circumstances.

The second form (PER ... FOR) provides hierarchy control of associations. If Assoc J varies within Assoc I, then J is incremented and exhausted for I = 1; I is then advanced to 2, J reset to 1, and the procedure continued until Assoc I is exhausted.

MECHANICS

The operand always appears in the Operand 2 position, which is the second column of the stub or the entry column. Procedure Name is always preceded by the word PROCEDURE. Since a Decision Table Name starts with a letter and a Decision Table Identification starts with a number, these can easily be differentiated. Either limited or extended entry form can be used. Illustrations are as follows:

	STUB	ENTRY
GO TO	PROCEDURE procedure name decision table name decision table identification	X, . or blank
DO	PROCEDURE procedure name decision table name decision table identification	X, . or blank
GO TO		PROCEDURE procedure name , . or blank decision table name decision table identification
DO		PROCEDURE procedure name , . or blank decision table name decision table identification
DO	decision table identification or name PER ASSOC letter	X, . or blank
DO	decision table identification or name PER ASSOC letter FOR ASSOC letter	X, . or blank
DO		decision table identification or name
		PER ASSOC letter , . or blank PER ASSOC letter
DO		decision table identification or name, . or blank PER ASSOC letter FOR ASSOC letter PER ASSOC letter FOR ASSOC letter FOR ASSOC letter

The first and third forms are limited entry and the second and fourth

forms are extended entry.

Input/Output Actions

Of all the operations required in a descriptive language those relating to input and output are the most complex and cause the greatest difficulty. The reasons for this include:

- a) Current input and output devices have certain limitations and conventions not directly comparable to manual filing and recording systems.
- b) The concept of "memory" as different from input/output devices has no analogy in manual data processing.
- c) Users seem less willing to accept constraints and artificial mechanics when dealing with records and files than with fields and character positions.

Thus, the input/output portion of any language is a compromise between unrestricted generality with resultant implementation inaccuracies and constrained mechanical manipulation with loss of logical clarity. The compromise chosen here considers the capabilities and structure of various input and output devices, but allows the user to think about these in the most convenient way. Although certain internal or memory records may be used for intermediate information, they are not involved in input/output actions.

Types of Files

Files are classified in two ways, by the function they serve, and by the type of access to them. Functionally, a file can be input only, output only, intermediate (input and output), and retained (input and output).

Two types of access are allowed for files, serial and random. For a specific file, the type of access depends largely on the medium on which the file is written. A serial file allows access to only the next physical record on the file, and is represented by many common media (e.g., punched cards, magnetic tape, printed reports, punched paper tape, and Teleprocessing lines). A random file allows access to any record of the file, but each such record must be specified. Typical media for random files are magnetic drum and disk. Under certain conditions, a specific medium can be used for both serial and random files. For example, a magnetic drum can be read serially by sequential addressing; and a magnetic tape can effectively be read randomly by searching for a specific record (rejecting all other records).

The input/output action operators, and the functional types of files with which they can be used, are discussed separately for serial and random files.

Serial Files

Consider a serial file as a roll of film, each frame holding one record. Only one frame can be viewed or exposed at a time.

An input file is a roll of exposed film. The roll is read serially from the first to the last frame.

An output file is a roll of unexposed film. A frame at a time can be exposed starting with the first frame. A whole series of frames (records), either wholly or partially ready for recording, can be available at a time.

An intermediate file is similar to an output file, but after a number of frames are exposed (and cut off), it can be used as an input. After use as an input, it can be saved, or it can be destroyed and a new roll given that same name. While being exposed, an intermediate file is treated just like an output; while being viewed, it is just like an input.

The retained file is somewhat more sophisticated. As each frame comes into viewing position, it is cut off the roll. It then can be added to a separate output roll or discarded. New frames can be added to the output roll or old frames modified before being re-exposed. After reading a retained file the input roll is destroyed and only the output roll retained. This in turn may be further read and written.

The only condition meaningful for a serial-type file is:

File Name IS END OF FILE

If END OF FILE is not checked and a READ is given beyond the last record, the READ operation does not occur.
Sequence is maintained for an input file as specified on the File Description. Input files cannot be scried. An output file is presumed to be in the sequence specified for the data from which it was developed, unless a different sequence is specified through the SORT operator. This also applies to intermediate files. A retained file is presumed to be in the sequence specified and this sequence cannot be changed. If no sequence is specified, then none can be obtained operationally. OPERANDS

Record names and file names are the principal operands used with the input/output operators for serial files. However, symbolic expression names and field names are used with the SORT operator.

OPERATORS

Four input/output operators are used with serial files:

READ provides for obtaining information from a file by making one record from that file available in memory for processing.

WRITE provides for recording information on a file by placing one record from memory in that file. The connector ON can be used with WRITE.

BEGIN enables one to start over again to READ or WRITE a particular file. SORT rearranges the records on a file according to

particular keys on those records.

READ (Serial File)

The READ operator is used in the form:

READ operand

The operand must be a file name. This action is used to read the next record of an input, intermediate, or retained file into a memory area that is large enough for the largest record of the file.

Canton Ca					ST	UB	ENTRY
x	X X X R		READ	READ File Nam		X, . or blank	
X		x	X	READ			File Name, . or blank

The first form shown is limited entry, and the second form is extended entry.

WRITE (Serial File)

The WRITE operator can be used in two forms:

WRITE			operand
WRITE	operand 1	ON	operand 2

The first form requires that the operand be a record name for an output or intermediate file. It can be a record name or a file name for a retained file. WRITE cannot be used with an input file. The second form requires that operand 1 be a record name from some other file (including internal files), and that operand 2 be a file name. This form can only be used for a retained file. A WRITE places the record selected in the next record area on the file. For output and intermediate files, there is one memory area for each type of record while these files are being written. There is only one common record area for each retained file.

 Midiu	UT TO	Ladi	REFERENCE	AL OHUT		ENTRY
	X	X	X	WRITE	Record Name	X, . or blank
			x	WRITE	File Name	X, . or blank
			x	WRITE Record Name	ON File Name	X, . or blank
			x	WRITE Record Name	ON	File Name, . or blank
			x	WRITE Record Name		ON File Name, . or blank
	X	X	X	WRITE		Record Name, . or blank
			X	WRITE	3	File Name, . or blank
			1			

The first three forms shown are limited entry. The remaining forms are extended entry.

BEGIN (Serial File)

The BEGIN operator is used in the form:

BEGIN operand

EN I

The operand must be a file name. This action can only be used with input, intermediate, and retained files. For an input BEGIN means "prepare to read the first record again". For an intermediate file it means "return to the first record position". If the next action on the file is to be a READ, the file is considered an input file; if the next action is a WRITE, it is considered an output file. For a retained file, BEGIN means "return to the first record of the newly constituted file". If BEGIN is used while processing a retained file, the portion of the file that has not been read is assumed to be destroyed.

June -		ERNEDIA	ATMEN .		
19/		1 27	5.	IUB	ENTRY
x	X	x	BEGIN	File Name	X, . or blank
x	X	x	BEGIN		File Name, . or blank

The first form shown is limited entry. The second form is extended entry.

SORT (Serial File)

The SORT operator is used in the form:

SORT operand

The operand is:

file name

PER key name

A.

The key name can be a symbolic expression name or a field name common to all records of that file. The file is always sorted into ascending order according to the defined collating sequence. The file name must be an output or intermediate file. An intermediate file must first be operated on by the BEGIN operator to insure that the file is ready for reading.

		5/53/ -53/	STUB	ENTRY
2	x	SORT	File Name PER Symbolic Expression Name	X, . or blank
2	x	SORT	File Name PER Field Name	X, . or blank
2	x	SORT		File Name, . or blank PER Symbolic Expressio Name
2	x	SORT		File Name, . or blank PER Field Name

extended entry.

Random Files

Random files have the same functional classification as serial files:

Input	 read only	
Output	 write only	
Intermediate	 a temporary fil	е
Retained	 a permanent fil	e

Consider a random file as a group of film slides that are in a box containing a series of slots, each slot identified by a location. Each slot holds one slide (or frame). A slide must be raised to be examined; to raise another slide the one just raised must be lowered. Changes can be made to the raised slide, just as though it were only partially exposed. New slides can be inserted into empty slots, or be inserted into occupied slots by removing (and destroying) the previous slide in that slot. A particular slide can be destroyed by clearing the slot in which it is stored.

There is no meaning to END OF FILE for a random file because of the media usually used by such files.

OPERANDS

Record names and file names are the principal operands used with the input/output operators for random files. However, logical expression names and location names are used for controlling the READ and WRITE operators.

OPERATORS

Three input/output operators are used with random files:

READ provides for obtaining a specific record from a file.

WRITE provides for inserting additional records in a file. The connector ON can be used with WRITE.

CLEAR is used to remove unwanted records from a file.

READ (Random File)

The READ operator is used in the form:

READ operand

The operand can be either:

file name PER location name

or

file name PER logical expression name

READ

file name

PER location name

obtains the record at the location specified by location name. If there is no record at that location, the read operation is not executed.

READ

file name

PER logical expression name

obtains the first record for which the logical expression value is true. This operation is defined on a circular basis. It starts searching from the location of the previous record that was read, and continues through the end of the physical file and around again to the same record. If there is no record that satisfies the conditions of the logical expression the READ does not take place.

READ can be used with input, intermediate, or retained files. There is just one memory area for the file. A READ temporarily removes the record from the file. For an input file, it is automatically returned in its original location just prior to the subsequent READ of that file. For an intermediate file and a retained file, the record is returned to its original location, but it may have been modified.

/	151	MEDI.	MEL PALL		
Ladin	nilino Har	PER PER	N. S.	STUB	ENTRY
x	X	x	READ	File Name PER Location Name	X, . or blank
x	x	x	READ	File Name PER Logical Expression Name	X, . or blank
X	x	x	READ		File Name, . or blank PER Location Name
x	x	x	READ		File Name, . or blank PER Logical Expression Name

The first two forms shown are limited entry. The remaining forms are extended entry.

WRITE (Random Files)

The WRITE operator can be used in two forms:

WRITE		operand
WRITE	operand 1 ON	operand 2

In the first form, the operand can only be a record name for an output file. For an intermediate or retained file, the operand can be a record name or a file name. The second form can be used only for intermediate and retained files. Operand 1 must be a record name, and operand 2 a file name.

The operand in the first form has this construction:

file name (or record name) PER location name

file name (or record name) PER logical expression name

The same construction is used for operand 2 in the second form, except that a record name cannot be used.

After WRITE the record in memory is not affected.

				EI.		6.47
	/	1	18			
		15	E	L'EL		
	5/	Ê/	ET/	TE .		1
14	<u>/ô</u>	14	1 2	STU:	В	ENTRY
	x	х	X	WRITE	Record Name PER Location Name	X, . or blank
		X	X	WRITE	File Name PER Location Name	X, . or blank
	Х	X	X	WRITE	Record Name PER Logical Expression Name	X, . or blank
		X	Х	WRITE	File Name PER Logical Expression Name	X, . or blank
		х	X	WRITE Record Name	ON File Name	-
		х	х	WRITE Record Name	PER Location Name ON File Name	X, . or blank
					PER Logical Expression Name	X, . or blank
		х	х	WRITE Record Name	ON	File Name, . or blank
		Х	х	WRITE Record Name	ON	File Name, . or blank PER Logical Expression Name
	X	x	x	WRITE		Record Name PER Location Name
		X	х	WRITE	1	File Name PER Location Name
	x	X	x	WRITE		Record Name PER Logical Expression Name
		X	x	WRITE		File Name PER Logical Expression Name
		Χ	X	WRITE Record Name		ON File Name, . or blank PER Location Name
		Х	x	WRITE Record Name		ON File Name, . or blank PER Logical Expression Name

The first six forms shown are limited entry. The remaining forms are extended entry.

CLEAR (Random File)

The CLEAR operator is used in the form:

CLEAR operand

The operand is:

file name

CLEAR can be used only with intermediate or retained files to remove an unwanted record from the file. An intermediate file is assumed to be cleared prior to the start of the process. CLEAR prevents the last record read from being returned to the area from which it was read.



The first form shown is limited entry. The second form is extended entry.

CHAPTER 7

Appendix

The appendix provides detailed explanations on filling out the various forms plus compact summaries of certain features of the language (e.g., which operands may be used with which operators, effect of different value class on conditions and actions, etc.).

The categories are as follows:

File Description Sheet

Reference Operand Description Sheet

Decision Table Form

Report Layout Sheet

Operand Summary

Condition Summary

Action Summary

FILE DESCRIPTION SHEET (See Figure 7-1)

The File Description Sheet is used to describe files, records, and fields. Each type of record within a file is described on a separate File Description Sheet. In addition certain information about the file itself is placed on each sheet. The complete description of the file is contained on the several sheets used to describe the records which make up the file.

SYSTEM	NAME	
	0	



ANALYST	DATE	
Ø	3	PAGE OF PAGES

FILE NAME		DIR. TYPE MEDIUM						RECORD NAME						ABB NO OF RECORDS			
	6)		Ø				\odot \bigcirc								D	(2)	(13)
600		s 3	10	IEND	ru I	Accumptions				>	10	DEI ATIVE	INCATION			1	
NO	FIELD NAME	MERIC	IGN	MAX	VAR	MIN	AVG	MAX	5500	To	UENCS	BEGIN	END		VALUE/REMARKS		
		1	-		-				-		-						
(IH)	(5)	16	M	(8)	19	20	20	22	23	24	23	20	2D		C	28)	
			_														
		+							_								
		+	-						-	-	$\left \right $						
		+			-												
		1															
			-		-				-		\vdash						
									-	\vdash	Н						
		+	-								\vdash						
		+	\vdash							\vdash	\square						
														_			
		+	-		-						Η						
		+	-														
		+	-		-												
		+	-		-	-			-		\square						
		+	-								\square						
DEL																	
REN	ARKS: (29)										-	_					
	_										_	-					·
	-IGI 7																

General Headings

At the top of the sheet the systems engineer will indicate:

- (1) the system name
- (2) his own name
- (3) the date of preparation or revision
- (4) a serial assignment of page number, and
- (5) the total number of pages which should normally equal the number of records names in the file; it will be greater if any record takes more than one page.

File Description Headings

- (6) The file name is shown
- (7) For directional type the systems engineer will indicate:

IN, OUT, RET (for retained), or INTER (for intermediate). IN represents a file used only for input; OUT represents output only; retained (RET) represents a file retained for reuse; intermediate (INTER) represents a file originated and used by the system but not available for subsequent use.

Medium covers two entries. (8) The first relates to whether the file is to be considered in a serial (SER) or random (RAN) manner. Serial files can be read and written one record after the other without need to control location. Random files are read and written per location or content. Serial files have an explicit End of File indication after the last record; random files do not. Serial Files cannot be interrogated without being read (except for End of File); random files can. Serial retained and intermediate files must be written to be preserved; random files do not.

(9) The second aspect of medium is related to the specific medium on which the file will appear. Suitable entries include simple abbreviations for:

Punched Card (PC)

8 Channel Punched Paper Tape (8 PAP TAPE)

Magnetic Tape (MAG TAPE)

8 1/2 x 11 Printed Continuous Form (8 1/2 x 11 CONT) 1301 Disk

Inquiry Station (INQ)

(10) A unique record name is inserted together with (11) a suitable abbreviation which must also be unique. Abbreviations are normally used for field name qualifiers to reduce writing. (12) The average number of this type of record in the file is noted and (13) the peak number expected. Other volume information may be shown under remarks (29).

Record Continuation

If there is a need to use more than one sheet for a single record the general heading (1 through 5) should be written, along with the file name (6) and (10) the record name with the letters CONT in parentheses; e.g., CUSTOMER RECORD (CONT).

Field Description

The fields which may appear on the record need not be described in any particular order. There is no implication that the reference order establishes relative position of these fields. Sub-fields do have positional implication as do fields on printed report records.

If all field descriptions (14-28) of a record are identical to another record in the same file or even a different file, enter the general heading (1-5) and the file description heading (6-13). (14) Enter Serial Number as 001 and (15) the words SAME AS followed by the record name of the File Description Sheet that contains the appropriate field descriptions.

<u>SERIAL NUMBER</u> (14) A serial number is assigned to each field or subfield. Serial numbers will begin with 1 or 001 and continue as far as required. Serial numbers can begin with 1 for each record description since they are only unique within a record description. If additional lines are required for any field the serial number is not changed; it may therefore be omitted on the subsequent lines.

<u>FIELD NAME</u> (15) Under field name the name of each field will be inserted. While there is no limit on the length of a field name, it should, for convenience, be kept brief and mnemonic. Lengthy names may use more than one line with the break being shown by a hyphen if it is in the middle of a word. Names for sub-fields will also be placed in this column. Descriptive information about a field will be written on the first line for that field. <u>ALPHA/NUMERIC</u> (16) In the alpha/numeric column the field is indicated as: N for numeric meaning that the value is always composed of the digits 0 through 9, A for alphameric meaning that any characters, including letters, numbers, and special symbols may be used; or F for floating point numbers. A numeric field may represent an integer or decimal and is acceptable for arithmetic manipulation; similarly for a floating point field. Alphameric fields while they may only have numeric characters in them at any given time are not suitable for arithmetic since neither sign nor decimal point are defined.

SIGN (17) Under the sign column will be indicated for each numeric and floating point field its sign possibilities. A field which is always zero or positive will have a plus sign (+). A field which is always zero or negative will be marked by a minus sign (-). If a field may be both positive or negative an E (for either) is inserted. Unsigned numeric fields, which are treated as positive in arithmetic are marked as U. Alphameric fields have the sign column left blank. The sign possibilities for floating point numbers will always relate to the mantissa. <u>LENGTH</u> -- The systems engineer will indicate (18) under MAX the maximum length for the value of the field. For a number (either N or F) the maximum integer length will be noted, then a decimal point, and finally the maximum decimal length. The sum of these indicates the number of significant positions. For all numbers the sign, decimal point, plus the floating point notation (including the F and the sign and

integer which follows) will <u>not</u> be considered in determining the length of the field (e.g., +16 has a MAX of 2, -17.350 has a MAX of 2.3, +27.1F-3.5 has a MAX of 2.1). For an alphameric field its maximum length will be indicated. (19) If the length of a field may vary considerably (like in name and address) this will be indicated by placing a check mark in the VAR column.

<u>OCCURRENCES</u> -- Certain fields may occur more than once in a record. If a field always appears exactly once per record a 1 is placed under MAX (22). If it is sometimes absent then this is shown by placing a 0 in MIN (20) and the proper average under AVG (21). A field which always occurs a fixed number of times greater than one, will have that number inserted in all three columns. If the number of times that a field appears on a record varies, the systems engineer will indicate the minimum, average, and maximum number of times that the field occurs on the record. Sub-fields will be blank in the occurrence columns because they can occur only once per field. Occurrences are always in terms of the "association to" as explained next.

<u>ASSOCIATION</u> -- On multiple occurring fields there is a need to be able to keep track of which particular field is being handled and to know which other fields occur in association with this field. For these reasons an association code letter is inserted under (23) ASSOC to indicate mutually dependent fields. Association code is blank for sub-fields because they have the same association as the parent field. Fields which occur a

maximum of once per record may not belong to an association. Association codes may be any letter and need only be unique for a record since they can be qualified by record name when used. All associated fields in a record will carry the same association code.

This one level association scheme works conveniently for fields whose occurrences only vary per the basic record. Items per invoice might well be one association including item number, item name, and item price, while taxes per invoice might be another independent association incorporating tax type and tax amount for federal, state, and local taxes.

A more complex problem is how to handle associations which are nested. For instance, there might be many operations to make a particular product; each operation might call on many different fixtures and tools; certain fixtures and tools might call for special instructions. To take care of this, each association or field which doesn't vary directly with the record is related to the controlling association by noting the controlling association code under (24) ASSOC TO. The following example may clarify how these columns are used:

	00	ccurrence	1.0000	Assoc.	
Field Name	Min.	Avg.	Max.	ASSOC.	To
Part Number			1		
Operation Number	1	5	25	A	
Operation Description	1	5	25	A	
Operation Time Standard	1	5	25	A	
Tool or Fixture Number	0	2	5	B	А
Tool Location	0	2	5	В	A
Special Instructions	0	.2	1	C	В

Occurrences for items in an association must be identical. This means that association A which occurs 1 to 25 times per the record consists of operation number, operation description, and operation time standard. Association B consists of two fields, tool or fixture number and tool location; these occur 0 to 5 times per operation (hence 0-125 times per record). Special instructions, Association C, is associated to and occurs 0 to 1 time for each tool (hence will occur 0 to 125 times per **record)**.

<u>SEQUENCE</u> -- If a file is to be considered as ordered, the systems engineer will indicate in (25) SEQUENCE the fields on each record which are used to sequence the file and the relative order of these fields. Numbers are indicated in the column, 1 for major, 2 for intermediate, etc. If any records in a file are ordered all must be ordered. A common named field must be included in each record to indicate the overall sequence of the file. For input and retained files sequence may be used. If used for intermediate or output files this can be nothing but informative to the reader. The analyst must still indicate through a SORT operation the particular sequence he wishes if this is different from the sequence in which it was developed. Files which are not considered sequenced have this column blank; sequence may still be introduced operationally by using SORT for intermediate and output files. Additional sequencing information may be indicated in the remarks section (29).

RELATIVE LOCATION -- There are two uses for relative location:

to indicate the relation of a sub-field to its parent field, and to show the line of the record and beginning and ending locations for fields on printed reports.

Each field which has sub-fields must have a length which doesn't vary (e.g., Length: VAR is blank). This allows positional reference to the field by the sub-fields. The sub-field has indicated the relative beginning position (of the referenced field) in (26) BEGIN and the ending position in (27) END. The referenced field name is noted in (28) the value/remarks area. An example may help explain:

	Length	Relative	Location	
<u>Field Name</u>	_Max	Begin	End	Value/Remarks
DATE	8			
MONTH	2	1	2	DATE
DAY	2	4	5	DATE
YEAR	2	7	8	DATE

The other use of relative position handles actual print position on a printed report. With the Report Layout sheet filled in the analyst must relate each field name to the print positions on the Report Layout sheet and the line of the record. Hence (26) the line of the record is shown for each field and (27) the ending print position of the field. If there is a possibility of misunderstanding the beginning print position, enter the beginning print position in (26) BEGIN after the line number of the record and separated from the line number by a comma; e.g., 2, 42 47. <u>VALUE/REMARKS</u> (28) The value/remarks column will be used for several purposes. If a file is an output or intermediate and the systems engineer wants to describe an actual value which a particular field will always have, he may do this by writing the value as a literal. If a subfield is described on the line, the name of the field of which the sub-field is a part will be noted. If values are assigned to sub-fields, the value will follow the field name separated by a comma. This column may also be used by the systems engineer for any brief remarks which he desires to make about any field such as format, possible values, accuracy, etc. <u>REMARKS</u> (29) At the bottom of the File Description sheet any remarks concerning the record or file may be made: information on sequencing; variations in record size or number of records; source of input or destination of outputs.

REFERENCE OPERAND DESCRIPTION SHEET (see Figure 7-2)

The Reference Operand Description Sheet is used to describe constants, expressions, value lists, and reference lists. None of these operands can have their values changed by the decision tables.

General Headings

At the top of each sheet will be indicated (1) the name of the system, (2) the name of the systems engineer or analyst, (3) the date the sheet is prepared or revised and (4) the page number. After completion, (5) the total number of pages of the Reference Operand Description sheets is shown for control purposes.

PAGE(T)OF(5)PAGES			
DATE	Ш		
ANALYST	VALU		,
BAN RAND DESCRIPTION			- ノインシー
LI REFERENCE OPE	TYPE		
EM NAME	NAME	6	
SYS	NON		

7.12

Operand Description

(6) A different serial number is used for each operand. These should start with 1 and continue sequentially as far as needed. The serial number may be a 1 - 3 digit integer.

Because the use of the form varies considerably for different types of operands, the form will be discussed under an appropriate heading for that operand.

If an operand requires continuation beyond the initial page the general headings (1 - 5) should be repeated, the serial number (6) also repeated, and (7) the name of the operand with the letters CONT in parentheses written after it.

CONSTANTS

(7) The constant name will be written in the name column. Names can be continued to more than one line if needed; if this requires breaking in the middle of a word then a hyphen should be inserted. (8) Under type is written CONSTANT or CON. (9) In the value column will be indicated the value of the constant exactly as it is to be used. If the value is numeric it is to be shown directly as with any literal. Alphameric values are to be enclosed in quotes. There is no length limit for constants.

EXPRESSION

(7) The name assigned to an expression will be written in the name column.(8) Under type will be written a mnemonic code to indicate the

type of expression:

Mnemonic Code	Interpretation
ARITH EXP or AE	Arithmetic expression
SYMB EXP or SE	Symbolic expression
LOG EXP or LE	Logical expression

(9) The expressions themselves will be written in the value column, using as many lines as are required; an operand should not be hyphenated. An arithmetic expression will be written using the arithmetic operators, functions, parentheses, and operands as allowed. Individual factors in the arithmetic expression may be set off by blanks to aid readability.

(9) For symbolic expressions the value column will also contain the expression. The Symbolic Operator name (currently only JOIN is available) will be followed by a left parenthesis; then the operands whose values will be put together to create the value for the expression will be written. Each of these operands will be separated by commas and written in the order they are to appear in the resultant value. The expression is terminated with a right parenthesis.

Suppose IDENT NO were given as the symbolic expression name for the expression JOIN (DEPT NO, '-', DIV NO, '/', MAN NO). If the current values of the operands are:

> DEPT NO = 726 DIV NO = 14 MAN NO = 123456

the value for IDENT NO would be 726-14/123456.

(9) A logical expression will be written in the value column using the operands, relational operators, logical operators, arithmetic operators, and parentheses required. Blanks will be used to insure readability. In constructing logical expressions note the following:

- The simplest form consists of a two operand condition like MARITAL STATUS = 1 PAY AMOUNT = 24.00
- 2) Following generally the same rules for constructing conditions, each of these operands may be more complex. Note the use of THRU instead of the hyphen for ranges.

For example, BOARDING CITY EQ "ATL", "MIA", "LAG" TAXABLE INCOME EQ 4000.00 THRU 7999.99

3) Arithmetic expressions may be introduced for either operand following the same rules as those for constructing arithmetic expressions. An example is:

CREDIT BALANCE + NEW ORDER AMOUNT GR (CREDIT CLASS * CREDIT RATE) + CREDIT BASE

4) Similarly, special arithmetic <u>functions</u> may be used:

SUM (OPERATING INCOME, INVESTMENT INCOME, TAX RETURNS) LE (# DOW JONES IND AVERAGE CURRENT YEAR # / # DOW JONES IND AVERAGE LAST YEAR #) * INVESTMENT BASE

Symbolic expressions may be used:

JOIN (NAME, ', ', RANK) = "JOSEPH JONES, 1ST LT"

5) Finally, in even more complex situations multiple conditions may need to be considered. This is done by using OR, AND, or NOT. NOT should be used very infrequently since there is a full set of relational operators. AND means both, and OR means either one or both. Examples include:

RATE (HEALTH, AGE) GR 4.25 AND POLICY REQUEST AMOUNT GE 25000 STATUS IS SENIOR OR AGE GR 23

VALUE LIST

(7) The name of the value list will be inserted in the name column. (8) The analyst will write VALUE LIST or VLIST under type. (9) In the value column the literal values comprising the list will be noted separated by commas. Each entry may be a single entry value or a range. The upper and lower limits of a range will be separated by a hyphen; the range is considered inclusive at both ends. The range itself will be separated from other entries by a comma. Numeric values are listed directly while alphameric values are enclosed in quotes. If either the upper or lower limit of a range is alphameric, both are individually enclosed in quotes. Values may continue on succeeding lines. Values need not be in any sequence. Alphameric and numeric values may be mixed in a list.

NAME	TYPE	VALUE
RECEIVING CITY	VALUE LIST	1, 3, 7, 9-11, 13, 16, 87, 2
SHIPPING CODE	VALUE LIST	"ATL", "GEO", "NOL", "4A" - "4H", 181 - 189, "LAG"

REFERENCE LIST

(7) The name of the reference list is given under name. This will usually contain some word like table or matrix to indicate the nature of the operand.(8) Under type the words REF LIST or RLIST is entered.

(9) On the top line the name of the first argument is shown - this name need not be unique since it is never referenced. If no obvious name suggests itself then ARGUMENT 1 or a suitable abbreviation should be used. A column should be drawn with this argument name as the title. The second argument name should be entered also on the top line and a column drawn. Finally the function name should be entered; this must be a unique name (or will be when qualified by the reference list name). A third column is drawn. (7) On the second line under name the reference list function name is written with the two argument names in parentheses following it. (8) Under type FUNC NAME is entered followed by a left parenthesis, then the number of values of the first argument, a comma, then the number of values of the second argument, and a right parenthesis.

If the actual values are not to be inserted until later the different values of each argument may be shown in the appropriate columns with the note in the third column saying "all combinations" or something expressive of how long the list will be. If the entries are to be inserted, each legitimate combination of argument 1 and argument 2 should be made with the appropriate value for the function. Figure 7-3 gives an example of such a reference list filled in.

SYS	TEM NAME	11	BM	AN	ALYST	DATE	1		
		REFERENCE OPE	RAND DESC	RIPTION			PAGE	OF	PAGES
		(PAR'	FIAL)						
NO	NAME	TYPE			VALUE				
	INS RATE MATERY	REF LIST	AGE	SECTION O	F COUNTRY	RATE	1		
	RATE (AGE, SEC-	FUNC NAME	0-9	"EAST	211	0.20			
	TION OF COUN-	(6, 4)	ſ	"WES"	T''	0.20			
	TRY)		1	"SOUT	rH"	0.15			
			5	"NOR	TH"	0.10			
		1	10-19	"EAS	וורן	0.35			
			/	"WES	Γ"	0.30			
				"SOU"	CH"	0.28			
			V	"NOR	TH"	0.20			
			20-29	"EAS	וויק	0.55			
			1	"WES	T ⁱⁿ	0.50			
				"SOU	CH"	0.45			
			J	"NOR	ΓH"	0.25			
			30-39	"EAST	ויק	0.80			
			("WES	L.I.	0.80			
				"SOU"	CH"	0.60			
			10°	"NOR	TH"	0.40			
			40-49	"EAST	C11	1.10			
			("WES	Г"	1.05			
_				"SOU	CH"	0.95			
			J	"NOR	TH"	0.80			
			50-59	"EAS	Cu .	1.55			
			("WES	Γ"	1.50			
				"SOU	rh"	1.30			
			×	"NOR	TH"	1.10			
_									
								_	_
			1		19				

Figure 7-3

7.18

SUMMARY OF ENTRIES FOR NAME (7), TYPE (8) AND VALUE (9)

OPERAND DESCRIPTIO	N	NAME (7)	TYPE (8)	VALUE (9)
Constant		Constant Name	CONSTANT or CON	Literal value of constant
		Arithmetic Expression Name	ARITH EXP or AE	Actual arithmetic expression
Expressions	$\left\{ \right.$	Symbolic Expression Name	SYMB EXP or SE	Actual symbolic expression
	l	Logical Expression Name	LOG EXP or LE	Actual logical expression
Value List		Value List Name	VALUE LIST or V. LIST	Literal values
Reference List		Reference List Name and Function Name (Argument 1, Argument 2)	REF LIST or R LIST and FUNC NAME (Count-Argu- ment 1 values, Count-Argu- ment 2 values)	ArgumentArgumentFunction12NameNameNameandandandandArgumentArgument12ValuesValues

An abbreviated version would look this way:

Name	Type	Value		
SPECIAL FARE TABLE	REF LIST	BOARDING CITY	DESTINATION	FARE
FARE (BOARD- ING CITY, DESTINATION CITY)	FUNC NAME (5, 3)	"ATLANTA" "BOSTON" "MONTREAL" "WASHINGTON" "MIAMI"	"NEW YORK" "PHILADEL- PHIA" "CHICAGO"	ALL COM- BINA- TIONS

DECISION TABLE FORMS (See Figures 7-4 and 7-5)

Decision table forms are used to describe the procedure logic of the system. The Decision Table Continuation sheet is used for horizontal continuation of a decision table. Extended entry or mixed entry tables will often require continuation since only 4 rules are provided on the Decision Table sheet. Since limited entry tables have 32 rules provided no special continuation form has been prepared. It is recommended that a limited entry decision table generally not exceed 32 rules; rather, the original table should be split into 2 or more tables. However, the Continuation sheet can be used for limited entry tables by drawing the proper vertical lines.

The type of data entered on the Continuation sheet is the same as on the Decision Table sheet. Therefore, the index numbers (circled numbers on figures 7-4 and 7-5) are the same and the entries will not be discussed separately except where noted.

		32			11		П	TT	Т	П	ТТ	ТТ		TT	
PAGEROF (EPAGES	OP * OPERAND 2	4 25 26 27 28 29 50 31													
DATE (3)	DP 00 OPERAND 2	116 19 20 21 22 23 2													
ANALYST	PREQUEN	10 11 12 15 14 15 16 17													
	P ** OPERAND 2 OF	02 02 04 06 06 07 08 09 1	D (D)												
PROCEDURE NAME	TABLE TYPE	(j) (j) (j)	6 6												
YSTEM NAME	TABLE NAME	Ø	(J) (J)											ARKS: 20	
N N	TAN-E	E	(3)		+	+	++	+		 ++	++	++	++	REN	

* *

		_	R			_	 _	 _	_	_	_	_	 _	_	_	_	 	_	_	_	-	_	 _	 	_	_	_	_	
24	PARE (FOF (6) PARES		OPERAND 2		(13)																								
DATE	0		OP OPERAND 2		00 (19)	A																							
ANALYST	8	ENCY	OP OPERAND 2		(চ) (ব)																								G-/
BM	LE CONTINUATION	FREQU	OP OPERAND 2		(1) (1)	6.4 6.4																+							a rigure
	DECISION TAB		P OPERAND 2		60 0																								
SYSTEM NAME	0	TABLE	SER OP OPERAND2 4	(12) (12)	(13) (14) (15) (1																				REMARKS:				

7.22

* *

General Heading

At the top of each sheet will be indicated: (1) the name of the system, (2) the name of the systems engineer, (3) the date the sheet is prepared or revised, (4) the page number, and (5) the total number of pages for this decision table. Each page of a decision table is numbered consecutively starting with 1. (6) On the Decision Table sheet enter the procedure name if the table is part of a procedure; otherwise leave blank. A single page may not contain tables from two different procedures, nor may it contain some tables which are part of a particular procedure and some which are not.

Decision Table Header

The first line of each decision table will be indicated by placing two asterisks (* *) in the left hand margin of the sheet.

On the first line of each decision table enter (7) the table identification, (8) the name of the table, (9) the table type (OPEN or CLOSED); if the decision table is the first table of a procedure (10) enter FIRST. The table identification is a 3 character field; the first position must be numeric to distinguish it from a table name, but the other two positions may be letters or numbers. The identification must be unique within the system as must table name.

If a table requires continuation to another sheet, the general headings (1-6) are inserted, as is (7) table identification and (8) table name; (9) CONT should be inserted under table type. More than one decision table may be described on the same sheet by placing two asterisks on the margin of the new table header line. Page 2 is used for the second vertical page, then page 3, etc.

If horizontal continuation is needed to handle additional decision rules then a Decision Table Continuation sheet is used. The general headings are inserted (1-5) with the page number of the first horizontal continuation sheet following the page number of the last vertical decision table sheet. An illustration may help.

Dec Table Form	Continuation Form	Continuation Form
page 1 of 9	page 4 of 9	page 7 of 9
Ser 1 - 10	Ser 1 - 10	Ser 1 - 10
rules 1 - 4	rules 5 - 10	rules 11 - 16
Dec Table Form	Continuation Form	Continuation Form
page 2 of 9	page 5 of 9	page 8 of 9
Ser 11 - 20	Ser 11 - 20	Ser 11 - 20
rules 1 - 4	rules 5 - 10	rules 11 - 16
Dec Table Form	Continuation Form	Continuation Form
page 3 of 9	page 6 of 9	page 9 of 9
Ser 21 - 30	Ser 21 – 30	Ser 21 - 30
rules 1 - 4	rules 5 – 10	rules 11 - 16

Figure 7 - 6

There are certain special tables. The initial table of the system is called START. Control will always begin at that point. START must be on open table. There must also be one STOP table per system. This provides for ending the procedure. The STOP table must also be an open table. In every procedure there must be a first table to which control will be transferred when entering the procedure. The first table of a procedure must be open if the procedure is entered by a GO TO action. If the procedure is entered by a DO action then the FIRST table must be closed. Since the DO is interpreted as applying to the whole procedure the return outside the procedure is retained even if a GO TO is encountered. Hence, within a procedure, a GO TO in a closed table does not destroy the return path to the table which called the procedure.

Let's review the GO TO and DO actions relating to open and closed tables. Whenever a DO is encountered a return point is set up in a push down list to bring control back to the action following the DO action. If further DO actions are encountered additional return points are placed one on top of the other in the push down list. A DO can only call on a closed table. A closed table must have at least one decision rule with no GO TO action. If there is a GO TO in a closed table the entire contents of the push down list are destroyed (only within the current procedure, however). An open table may only be referenced by a GO TO action, and must have an explicit GO TO for each decision rule (except within a procedure when a decision rule without a GO TO will return control to the table calling on the procedure).

Decision Rule Header

(11) Rule numbers are preprinted on the Decision Table sheet. The rule number for extended entry form is at the top of the row; it is at the bottom of the row for limited entry form. If a Decision Table Continuation
sheet is used, the rule numbers must be entered in the same relative position as on the Decision Table sheet.

(12) Frequency information can be inserted on the first line of each rule by noting a 2 digit number to indicate the percent of times that each rule will be satisfied. The percents should add up to 100.

OPER OPERAND 1	OP OPERAND 2	RU 01	1LE 02	-
		80	20	← Frequency
CREDIT LIMIT	EQ 'LIMITED'	Y	N	

If 80% of the customers had limited credit, it would be expected that 80% of the time rule 01 would be satisfied. Rule 02 would then be satisfied 20% of the time. For extended entry form the percentage will be entered in the left-most part of the rule column.

If all possible combinations of conditions are not written and it is expected that the data to be processed will not always satisfy the explicit rules, an ELSE rule will be entered. The ELSE rule will be the first rule column to the right of the last written rule. For extended entry, enter (12) the word ELSE in the left-most part of the column. The letter E will be entered (12) for the ELSE rule on limited entry form. The frequency of the ELSE rule is presumed to be 100 minus the sum of the percentages of the other rules.

Table Body

The balance of the decision table is called the table body. The stub is separated from the entry by a preprinted, vertical double line. The condition statements will be separated from the action statements by drawing a double horizontal line after the last condition statement.

Condition	Condition
Stub	Entry
Action	Action
Stub	Entry

(13) <u>Serial Number</u> - A 1 to 3 digit serial number is assigned to each condition and action row in the table. Serial number is only unique for a single decision table. If a horizontal continuation is necessary, the serial numbers will be repeated for the corresponding rows of the continuation sheet. If multiple lines are required for a single statement, the serial number need not be indicated on the subsequent lines. A single heavy line will be drawn under each row when multiple lines are used. <u>Conditions</u> - For each condition row there will be (13) a serial number assigned. This number can be used for reference purposes, but does not indicate the order of examination. All condition serial numbers should be smaller than any action serial number. (15) The first operand name or value is shown under operand 1. If the name requires continuation it will be split where a blank actually occurs or else a hyphen will be used at the end of the operand 1 space. In limited entry form (16) the operator and (17) the second operand will be shown in the stub except when VS (versus) is entered. (18) In the entry area on the first line of each condition enter Y (yes or true), N (no or false), period or blank (not pertinent or impossible) for each rule. If VS (versus) is entered in the stub, any relational operator may be shown in the condition entry, e.g., LE, GR, <, etc. These act just as though they were substituted for the VS.

In extended entry form the operator may appear in the stub (16) or the entry (18). The operator normally will appear in the stub area (16) if it is common for the row. If it varies, it must appear in the entry area (18). Operand 2 always appears in the entry area (19). <u>Actions</u> - For all action rows, the action operator (14) will be entered in the stub. Operand 1 (15) is entered for action operators SET, MOVE and WRITE (when the WRITE operator is used with the connector ON).

	(14)	(15)	(16)	(17)
Limited entry example	WRITE	RECORD	ON	FILE

The other action operators do not require an operand 1.

The operators SET, MOVE and the two operand form of WRITE use connectors. The connectors are entered as shown in Figure 7-7.

In extended entry form operand 2 (19) is entered in the action entry area on the first line of the action statement. If the action is not to be taken, leave the entry area (18, 19) blank.

DESCRIPTION	OPERATOR	CONNECTORS	CONNECTOR LOCATION
LIMITED ENTRY			
Output	WRITE	ON	(16)
Value assignment	SET	EQ or ←	(16)
Two operand arithmetic	SET	+, -, * or /	(16)
Multiple operand arithmetic	SET	EQ or ←	(16)
		+, -, *, / or **	(16)
Value assignment	MOVE	TO or ->	(16)
Two operand arithmetic	MOVE	+ - * or /	(16)
Multiple operand arithmetic	MOVE	TO or →	(16)
₽°		+, -, *, / or **	(14)
EXTENDED ENTRY			
Output	WRITE	ON	(16) or (18)
Value assignment	SET	EQ or ←	(16) or (18)
Two operand arithmetic	SET	+, -, * or /	(16) or (18)
Multiple operand arithmetic	SET	EQ or ←	(16) or (18)
		+, -, *, / or **	(18)
Value assignment	MOVE	TO or →	(16) or (18)
Two operand arithmetic	MOVE	+, -, * or /	(16) or (18)
Multiple operand arithmetic	MOVE	TO or →	(16) or (18)
		+, -, *, / or **	(14)

Figure 7-7

An operand may be continued on as many lines as required. An operand should not be split between pages of the decision table. Either split at a blank or insert a hyphen.

(20) <u>Remarks</u> - At the bottom of the sheet, space is provided to permit comments to aid others in understanding the decision tables entered on the page, e.g., the desired length of the result of an arithmetic expression. If appropriate, reference may be made to the serial number of the condition or action. If more than one decision table is on the page the remarks should reference the table.

Any expression, (arithmetic, logical, or symbolic) that is referred to only by the decision tables on the page may be defined here instead of on the Reference Operand Description sheet. This is called a local expression. The name of the expression is written, then the type of expression (AE, LE, SE), and the expression itself following the same rules as noted for the Reference Operand Description sheet.

Appropriate local abbreviations may also be indicated under Remarks by indicating the abbreviation, then ABB, then the full name or value.

REPORT SPECIFICATION (See Figure 7-8)

Report format and content are described by using a Report Layout sheet (Figure 7-8) and one or more File Description sheets.

The Report Layout sheet will be used to specify:

- 1. The position of records and fields
- 2. The size of records and fields

N U.S		-	H	105		-	=							-	-									+	-			SHT		1
X24-	EE	H	E	578					-		1				-		-	-						-	-			DEF 6"		
RIN	HH	H	H	22	===						++	++	Ħ		-		==	+						-		3		RM		
	HH	· H	FI.	33	111	1				FF	1	1			-	1						=	Ħ	-		4		0 %	11"	
	F F	F	F		H	==				FF	11	++	+		-	T	-			11	11	11	Ħ	-	Ħ	- Territ		33%	5	
	HH	H	H	0									H		-			_						-	F	and		50	. 3	1
	EE	E	E	61		-							-										H	-	-			ALY 33	1.511	
	EE	E	B	45														_						-	F	5	2	NON		
	88	Η	H	53		-									-									-	F	10		33	-	
	A A	F	F	10		-			_	FF			-	-	-				H				-		H	RRIJ		8		
	HH	-	H	8											-			-				-	-	-	F	CAL		5	_	
	BB	H	Η	10											-	-								-	F	0 1		E.	475	1
8	HH	H	H	2 2										-				-				1 23		-	H	SPE	ba	WII 33%	18	
	HH	H	H	23		-		-							-			-			1				-	AL-	1 10	KM.	lax.	Ľ
-	HH	H	H	0		-									_			_								DI	1.5	04 <	~	
	HH	E	H	00											-		-							+		han	tst		-	
	HH	H	H	10											+											2	1	10		
	HH	H	Ħ	0 4							++				+			-								0	p	ITS I	191	
	HH	H	H	64								++			+		-						Ħ	+		- uo	sche	MEN	-	
-		-	1	0		_		-				++			-		-	-								5	b	IRE!	ici.	
	HH	H	Ħ	00		-									+	11		-			#					a d	Pe -	Ast	ġ.	1
	HH	H	Ħ	0		-					++		1.1					-					Ħ	-				ME	ical	1
	HH	E	Ħ	00 4									11										H			ł –		lori	/ert	1
	HH	H	Ħ	2									11		-			-					Ħ	1			_	-	-	Ľ
-	HH	-	-	0		_							11		-		-	-												_
	HH	F	E	8							11		++		+							11	#					++	1	ţ
	HH	E	H	10		-						++			1													++		ŧ
	HH	H	F	V 4							++				-								Ħ					++		ŧ
	HH	F	F	12							11	11	11						Ħ				Ħ	1				++		ŧ
F		-	-	0		_					++		11	_	-		_				11			1			-	++	-	\$
101	F F	F	F	1 00								11			-		_	-		=			H	+				++	-	ŧ
AR	F F	F	F	10								11					-	-					\square	1			-	++	-	ŧ
H	H H	-	F	340										-				-		11			Ħ					++		ŧ
8	H H	F	F	1 23					-			1+			+		-	-		11			Ħ					Ħ	-	ŧ
NIN	-8-8		-	0		-			_					-	-		-	-					++	-			-	++		4
er CI	F F	F	F	18											-		_						11	-				++	-	Ŧ
PA SS	F F	H	-	10											-		_			11								+1	10	t
Sine	H H	F	F	50 4											-															-
03 03	H F	F	H	12							++	-			+						11		H	-			-	Ŧ.		-
14			-11	0									-	-	-		-	-						-			-	-		-
M	F F	-	-	7.8										-	-		-				-			-				-	00 1	-
IBI	H H	-	F	10										-	-	++							H	-				1	0	-
INI	H F	F	F	34		-									-		-	-					H	-					ň	-
	FI FI	F	F	12		-						11		-	-									-				F	Eg.	-
-			-	0									-	-	-								-	-				-	14	-
	E E	F	F	7.8											-			-						-				-		-
	E E	E	F	CD 107								H			-									-				H	1.1	ŝ
	BE	E	H	4		-								-	-		-	-						-		-			-	Ŧ
		E	-	12												-		-				-		-		-	-		-	Ŧ
-				0.6											-			-		-				-					-	+
		F	B	78											-			-						-					-	Ŧ
	BF	E	E	2 10																			H		H				-	f
	BF	F	F	4 5												-			-			-						-		f
	EF	F	F	12										-								-		-					-	f
ł			-	0							-	-	-	-	-				-	-				-				+	-	f
	HE	E	H	78	++-					H													H	-					-	\$
KS			H	- 10										-			-			-										1
MAR		H	H	3.4				-					-	-							-			-						1
020	E	E	E	12																				-					-	-
OM				0	+									-	-						++			-					-	-
NGS	EE	E	B	7.8																										-
ADIA	EE	F	F	0 50	2			-													-							+	-	-
H		I A	H	4															-										-	+
HELE	H F	H	F	12																						-		++	+	1
-				4	N m	+ 5	-	-	0 -	-	2 4 4	1 40 1	- 00		-	1 17 7	10 10	2.0	00 0		2 10 1	12 1	E	2 9	0	1 2 2	3	2 20 1	2 9	1
					TT	T	11	11	11	1	11	1				1 1 1		1 1	11	11	11	11	TT		TT		TT	11	T	1
			\$	R	71																									
			E E	96	à								1														11			
			PA	15																										
	00	0	6	2	TT		11	11-		11	11	=				+			2	1	++	++	11	-	H			11	-	1
			0	W 4	+1						11	++							3	++	++	++		-	11	++		++	-	1
			5	5	++								1						5		11				11			+		1
Z	BMI		0																					•					•	1
Ē	N La		Ð	0 1	+-	-				H		+	-		H				8		++		+	-	11	-		+	-	+
=			1.10	10									-						110	T	IT		IT	T	IT		IT.		1	1
Scell	EM EM	est	9	14															10											J

[

- 3. The pre-assigned content of records
- 4. The editing of variable fields.

A separate File Description sheet will be prepared for each record of a report which contains one or more fields which will vary in value. The report is considered a file and the report name the file name. The File Description sheet is used to:

- List and relate the names of the variable fields to the characters in the printer layout sheet which describe the position, size and editing of the fields.
- 2. Specify the unedited size of the variable fields.

The systems engineer will specify in decision tables the conditions for assigning different values to variable fields and the conditions for printing non-constant records. Many housekeeping type details are assumed to be taken care of automatically and hence need not be procedurally specified by the analyst.

REPORT LAYOUT SHEET (See Figure 7-8)

General Heading

(1) The system name, (2) analyst name, (3) date of preparation or revision, (4) page number, (5) total number of pages for the report, and(6) the report name (file name) are entered.

Figure 7-8 is one of many printer layout sheets that may be used for Report Layout. If another layout sheet is used, the general headings may appear elsewhere on the sheet. A heavy rectangle is drawn on the printer layout sheet to show the relative size of the desired report page.

Record Heading

Each different line, or inseparable, continuous group of print lines, may be considered a record. When a record consists of more than one print line, all lines are printed when the WRITE action occurs for the record.

For operational control of variable spacing between printed lines, one or more blank lines may be designated as a record and written when desired.

<u>Line Count</u> (7) Enter a 1 or 2 digit number to indicate the number of print lines in the record. The line count will immediately precede the record name.

<u>Record Name</u> (8) The name of the record is entered on the first line that is used to describe the record. To continue a record name on a subsequent line, either break at a blank between words or enter a hyphen in the last character position. A record composed of fixed data fields and written in the same position on every page is called a constant record. All such records will be given the record name CONSTANT. A constant record may contain commonly understood variable fields. Presently Page Number and Date are the only variable fields that may be included in constant records. It is presumed that the programmer will supply the proper page number and date on each page of the report. Other fields will be included if the need arises. Writing of CONSTANT records is not required. It is assumed that the programmer will make the necessary arrangements.

(9) <u>Record Position</u>

Records, when printed, may always appear in the same page position (Fixed) or they may appear anywhere on the page (Variable) except where CONSTANT records appear. To indicate print positioning, enter a code immediately after the record name, separated from the record name by a hyphen.

Code	Meaning
V	Variable - The record will be printed in the next available non- <u>constant</u> position.
F	Fixed - The record will be printed on the lines corresponding to the lines on which it is described.
2 Digit Number	Fixed - The record will start on the line specified by the numeric code. The code may not specify lines occupied by CONSTANT records.

Normally, the record which is to be fixed in position (Code F) will be described on the lines on which it is to be printed. When two or more fixed records must appear on the same or overlapping lines on different pages, the number code is used. If the fixed position on the current page has already been passed, it will be printed on the next page at the designated position.

(10) <u>Record Description</u>

A record may be composed of all variable value fields, all fixed value fields, or a combination of both. Except for pre-printed information, fields will be described in the body (10) of the Report Layout sheet. Pre-printed information will be shown by including a sample pre-printed form (or a picture thereof). The relative vertical and horizontal position for each field within a record will be shown on the Report Layout sheet. <u>Fixed Value Fields</u> Fields that always have the same value are called fixed value fields. These will be indicated by writing their values in the proper positions on the Report Layout sheet and not referencing these positions in a File Description sheet.

<u>Variable Value Fields</u> Fields whose values change are called variable value fields. Their values must be supplied by the decision table procedure. Only variable value fields will be described on a File Description sheet.

If the same field can appear in a record more than once, it will be shown on the Report Layout sheet each time it can appear. The field will be described on the File Description sheet in the same manner as any other field except that all possible line numbers and print positions will be entered on separate lines in the Begin and End columns respectively. Variable value fields are shown by proper use of the symbols N or A; N representing numeric valued fields and A for alphameric. <u>Edited Fields</u> Various characters are used on the Report Layout sheet to show the editing of variable value fields. The following describes the use of these characters.

There are four types of editing characters used on the Report Layout sheet:

- 1. Character Representation
- 2. Field Adjustment
- 3. Replacement
- 4. Insertion

When the field description on the File Description sheet (unedited field) differs from the edited field description on the Report Layout sheet (edited) field, the edited description will control. If the edited field is longer, zeros will be substituted for the missing high order integer and low order decimal positions, for numeric fields. Low order blanks will be added to alphameric fields. Decimal points will be aligned for numeric fields. If a decimal point is not specified, it is assumed to be immediately to the right of the right-most digit. For alphameric fields, if the edited field is shorter than the unedited field, the high order values will be printed and the extra low order positions dropped. For numeric fields, if the edited field is shorter, the high order decimal positions will be rounded and printed and the low order integer positions will be printed.

Character Representation

There are two characters used to indicate the character representation mode, A and N. A character will be printed for each of these, unless the character in the source field is blank.

CHARAC-	MEANING	UNEDITED	DESCRIP-	PRINTED
TER		VALUE	TION	VALUE
N	the character position	0164	NNNN	0164
	is always numeric	03040	NNNN	3040
	(0-9) or blank	24	NNN	024
A	The character position may be any character (0-9, A-Z or a special symbol).	ABCD ABCD ABCD	АААА ААА ААААА	ABCD ABC ABCD_

Insertion Characters

Insertion characters add to the length of the printed field. They do not by themselves replace any character that would otherwise be printed. There are four types of insertions: special, floating, conditional, and normal.

<u>Special</u> - There is only one special insertion character. This is the decimal point (.).

CHARAC-	MEANING	UNEDITED	DESCRIP-	PRINTED
TER		VALUE	TION	VALUE
	Decimal point. For numeric fields the decimal points will be aligned. If a decimal point is not indicated, it is specified by the File Description sheet.	00123 00.123 00.127 .134	NNN N. NN N. NN N. NN	123 0.12 0.13 0.13

<u>Floating Insertions</u> - All floating insertion characters suppress lead zeros and lead blanks. The first floating insertion character will be placed next to the high order position of the edited field description and continued to the right for as many positions as are to be suppressed. On floating insertions, a minimum of two insertion characters must be indicated. These are three floating insertion characters:

\$

+

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
\$	Dollar sign. The dollar sign will be printed in the right- most character position that is suppressed	123 012 _12 000 012 000	\$\$NN \$\$NN \$\$NN \$\$NN \$\$NN \$\$\$N \$\$\$\$ \$\$\$\$	\$123 \$12 \$12 \$00 \$ \$12
+	The plus sign causes a plus sign to be placed in the right- most position in which suppression occurs if the field is <u>not negative</u> A minus sign will be placed in the right- most position in which suppression occurs if the field is negative.	+123 012 -001 000	++NN ++NN ++NN ++NN	+123 +12 -01. +00.

7.38

7.39

CHARAC-	MEANING	UNEDITED	DESCRIP-	PRINTED
TER		VALUE	TION	VALUE
-	The minus sign causes a minus sign to be placed in the rightmost position in which suppression occurs providing the field is negative. If the field is <u>not negative</u> , insertion is not made made but lead zeros and blanks are suppressed.	-123 +123 -012 +012 -000	NN NN NN 	-123 123 -12 12

<u>Conditional Insertions</u> - The conditional insertion characters are inserted in the exact position shown, only if one or more conditions are satisfied. There are five conditional insertion characters:

> + -CR DB

7.	40	

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
+	A plus sign in the high order or low order position of the edited field descrip- tion will cause the placement of a minus sign in the specified location providing the field is negative. If the field is <u>not</u> negative a plus sign will be placed in the specified position.	-123 +123 123 -001	NNN+ + NNN NNN+ +NNN	123- +123 123+ -001
-	When a minus sign is the high order or low order position of the edited field description, it will cause the placement of a minus sign in the indicated position. If the field is not negative no char- acter will be printed in that position	-123 +123 -123 -001 -001	-NNN -NNN NNN- -NNN NNN-	-123 123 123- -001 001-
CR	When these char- acters are the low order position of the edited field descrip- tion, CR will be printed if the field is negative. If the field is not negative, the characters will not be printed.	+123 -123	NNNCR NNNCR	123 123CR

7.41

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
DB	If the field is negative CR is printed; if the value is positive DB is printed; if the value is zero nothing is printed.	+123 -123 123	NNNDB NNNDB	123DB 123CR
,	A comma will be printed in the pos- ition indicated only if there are non-blank char- acters preceding.	1234 010 1030	NN, NNN NN, NNN NNN, NNN	1,234 010 1,030

<u>Normal Insertions</u> - Insertions will be made in the character position specified by the placement of the insertion character. Any numbers or special character may be inserted.

CHARAC-	MEANING	UNEDITED	DESCRIP-	PRINTED
TER		VALUE	TION	VALUE
0 - 9 Special Symbols	The character specified will be inserted exactly as shown in the field. It may not be the leftmost or rightmost character.	123062 123062 312.34	NN-NN-NN NN/NN/NN N@-\$NN. NN	12-30-62 12/30/62 3@-\$12.34

<u>Replacement</u> - Replacement characters are used to change the value of a field. They never add length to an edited field, because they will only replace existing characters. There are two types of replacement characters; Conditional and Normal.

<u>Conditional</u> - Conditional replacement characters specify the replacement of high order zeros or blanks of a field. There are two conditional replacement characters:

Z

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
*	Asterisk replace- ment. Replaces leading zeros or blanks of a field with asterisks.	12345 00123 00100 00000 01000 00000 100 00	***NN ***NN ***NN ***NN **** **** ***NN ***NN ***NN ****	12345 **123 **100 ***00 *1000 ***** **100 ***00 *****
Z	Suppress lead zeros for numeric fields ONLY. This character is used when it is desired to replace zeros that appear in the leftmost positions of a field with blanks.	12345 00123 00100 00000 00100 00000	ZZZNN ZZZNN ZZZNN ZZZZZ ZZZ Z	12345 123 100 00 100

<u>Normal Replacement -</u> Replacement characters replace the value of whatever character appears in the field with themselves. There are two normal replacement characters:

O (stands for zero)

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
0	Zero Replacement for numeric fields only. The place- ment of zero in an edited field will re- place the corres- ponding character in the unedited field with a zero.	12345 12345 12345 12345	OOONN NNOOO OONN NNOO	00045 12000 0045 2300
B	Blank Replacement. The placing of B in a field is the same as zero except the character is re- placed with a blank instead of a zero.	12345 12345 12345 12345 ABCDE ABCDE	BBBNN NNBBB BBNN NNBB BBBAA AAABB	45 12_45 23_2 DE ABC

В

<u>Field Adjustment:</u> - The field adjustment characters are used to specify the truncation or rounding of a field. Where they are written they specify the location of the rightmost character of the field. There are two field adjustment characters:

Т

R

17			A
1		4	4
	٠	-	-

CHARAC- TER	MEANING	UNEDITED VALUE	DESCRIP- TION	PRINTED VALUE
Т	Truncate. When the unedited value is longer than the edited description and the high order positions are to be printed, a T is placed in the right- most position. The fields will be left justified.	12345 112.345 12.3	NNNT NN. NT NNT	1234 12.34 12
R	Round for numeric fields only. Similar to Truncate except 5 is added to the right- most position to be dropped, the carry, if any, is made, and the field is truncated. The fields will be left justified.	12345 12345 123. 45 456. 78	NNNR NNR NNN. R NN. R	1235 123 123.5 56.8

If decimal points are aligned and the edited decimal position is shorter,

rounding will occur unless truncating is specified.

OPERAND SUMMARY

The permissible operands are listed in Figure 7-9. The character set in proper collating sequence is listed in Figure 7-10. The character set listing includes the character names and information concerning the use of the various characters.

OPERAND	WHERE DEFINED	WHERE USED
Information Operands File Record Field	File Description Report Layout	READ, WRITE, SORT, BEGIN, CLEAR, State relational WRITE, MOVE, SET MOVE, SET, value relationals
<u>Reference Operands</u> Reference List Function Value List Constant	Reference Operand Description	MOVE, SET, relationals value relationals MOVE, SET, value relationals
Arithmetic Expression Logical Expression Symbolic Expression	'Reference Operand Description Remarks in Decision Table	MOVE, SET, relationals MOVE, SET, relationals MOVE, SET, relationals
Segmenting Operands		
Procedure Decision Table	Decision Table Headers	GO TO, DO GO TO, DO
Work Operands		
Work Record Work Field		WRITE, MOVE, SET MOVE, SET, relationals
Self-Describing Operands		
Literals Connotative Terms State Terms Descriptive Terms Value List Range		MOVE, SET, value relationals MOVE, SET, value relationals State relationals MOVE, SET, value relationals value relationals value relationals

CHARACTER SET in COLLATING SEQUENCE

Character	Name	Use
	Blank	other than first position in operand names; to set off operands from operators
	Period or Decimal Point	to show decimal position in number
Ħ	Lozenge	
(Left parenthesis	for delimiting portions of expressions
<	Less than	relational operator
≢ `	Group mark	
+	Plus sign	for addition; for positive numeric literals
\$	Dollar sign	for editing on reports; for other than first position of operand names
*	Asterisk	for multiplication; for editing on reports
)	Right parenthesis	to delimit positions of expressions
;	Semicolon	
Δ	Delta	for other than first position of operand names
· -	Minus sign or Hyphen	for subtraction; for continuation in the middle of a word; for negative numeric literals
1	Slash	for division
3	Comma	for editing on reports
%	Percent sign	for other than first position of operand names
= .	Equal	relational operator; with SET operator

Character	Name	Use
T	Single quotation mark	to set off alphameric literals
н	Double quotation mark	
¢	Cents sign	for other than first position of operand names
#	Number sign	for other than first position of operand names; to set off descrip-tive terms
@	At sign	for other than first position of operand names
:	Colon	between qualified names
>	Greater than	relational operator
	Radical	
?	Question Mark	
A to I	Alphabetic Characters	for operand names
1	Exclamation Point	
J to R	Alphabetic Characters	for operand names
7	Record Mark	
S to Z	Alphabetic Characters	for operand names
0 to 9	Numeric Characters	for other than first position of operand names

Figure 7-10 (cont.)

Functions

Special mathematical functions may be used in arithmetic expressions. A function consists of a function name and, in parentheses, one or more arguments. The arguments are field names, literals, arithmetic expressions, or functions.

If a multiple occurring field is used in an arithmetic expression, the arithmetic expression may be referred to only in a closed table. The rules that apply to the DO statement for handling multiple occurring fields are applicable as though the arithmetic expression is part of the closed decision table.

The following functions are explicitly defined with values as accurate as any other in the expression or as accurate as the receiving or comparing field in the decision table. In the case of comparing two arithmetic expressions each consisting only of functions, the result will be carried to the same number of significant positions as the argument value.

Mnemonic Code	Name	Form
SIN	Sine*	SIN (arg 1)
COS	Cosine*	COS (arg 1)
TAN	Tangent*	TAN (arg 1)

* all values of arguments are expressed in degrees

Examples: SIN (ANGLE) COS (RATE (HEALTH, AGE)) TAN (LENGTH 1 - LENGTH 2) SIN (36.2) COS (SIN (B)) TAN (WORK ANGLE)

Mnemonic Code	Name	Form
ABS	Absolute	ABS (arg 1)
NEG	Negative	NEG (arg 1)
	The sign of the argument will be reversed. A unary minus.	

Examples:

ABS (QTY ORDERED - QTY DELIVERED)

A + NEG (B)

Mnemonic Code	Name	Form
SUM	Summation	SUM (arg 1, arg 2,
	Only field names, work field names, literals, reference list function names, and functions may be used.	

Examples:

SUM (A, B, C, D) The result is A + B + C + D

SUM (Sin (A), COS (B)) The result is SIN (A) + COS (B)

Other functions may be added later as experience indicates their need

and value. For instance, these arithmetic functions are under consideration:

MAX (arg 1, arg 2, ..., arg n) -- picks the largest value of the arguments.

MIN (arg 1, arg 2, ..., arg n) -- picks the smallest value of the arguments. AVG (arg 1, arg 2, ..., arg n)

-- adds the argument values and divides by the number of arguments.

REM (arg 1, arg 2)

-- the remainder after dividing argument 1 by argument 2.

INT (arg 1, arg 2)

-- the integer value from dividing argument 1 by argument 2.

Other logical and symbolic functions are also being examined. Recommendations as to the need for such functions is requested.

CONDITION SUMMARY

Certain operands may not be used together in condition statements. Also, certain of these operand pairs may only be used with the equal and not equal operators. The rules are summarized in Figure 7-11 with examples shown in Figure 7-12.

When the State Relational Operators (IS and NT) are used in a condition statement, one of the operands must be a Named Operand and the other must be a State Term. The permitted combinations are shown in Figure 7-13 with examples shown in Figure 7-14.

Operand	Combinations	with	Value	Relational	Operators
---------	--------------	------	-------	------------	-----------

Operand 2 May Be If Operand 1 is	Field Name	Value List Name	Arith. Expr. Name	Symb. Expr. Name	Ref. List Func. Name	Const. Name	Work Field Name	Literal	Connot. Term	Desc. Term	Value List (Range
Field Name	X	X*	X	x	x	x	X	X	X	x	X*	Χ*
Value List Name	X*		X*	X*	X*	X*	Χ*	X*	X*	X*		
Arithmetic Expression Name	X	X*	X	X	X	X	X	X	X	x	X*	Χ*
Symbolic Expression Name	x	X*	X	X	X	X	X	X	X	X	X*	X*
Reference List Function Name	X	X*	X	X	X	X	X	X	X	X	X*	Χ*
Constant Name	X	X*	X	X	X		X			X		
Work Field Name	X	X*	X	X	X	X	X	x	X	X	X*	X*
Literal	X	X*	X	X	X		X			X		
Connotative Term	X	X*	X	X	X		X			X		
Descriptive Term	X	X*	X	X	X	X	X	X	X	X	X*	X*
Value List	X*		Χ*	X*	X*		X*			X*		
Range	X*		X*	X*	X*		X*			Χ*		

Note: X means any operator permitted. X* means only the EQ and NE Value Relational Operators may be used. Blank means that the operand pair is not legitimate.

			DECISION TABLE			PAGE OF P	AGES
11							
TOT	TABLE NAME	TABLE TYPE	OD O	OB OR ODEDAND?	NCY DEFEAND 3	COCDANN	- 6
NRK N	OFER OFERO	Crementer 10	o: 02 03 04 05 06 07 08	09 10 11 12 15 14 15 16	17 18 19 20 21 22 25 24	25 26 27 28 29 3	0 31 33
00	CITY CODE	EQ 04 06	XM				
TT		04 13-19					
R	CITY CODE	EQ	13 - 19	21 - 30	41-57		
003	OPEN BALANCE		GR + 1250	EQ + 12 50	LR + 17,50		
10	EMBLOYE E NAME	GR	NDAM "	W10 "	"HOIN		
205	EMPLOYEE NAME		10 TO 11	"JOH "	Min u		
900	CUSTOMER IDENT	NE BLANK	N				
20	DATE REQUESTED: CUSTOMER ORDER	GR # CURRENT DATE#	N N N				
8	WORK ABC	LR OVERTIME PAY (XDR CLASS) HOURS WORKED)	> • · · ·				
50	FORMULA 172		LR 1275	•	LR 1275		
010	STRING HEADER		EQ "ABC-257"	EQ * BQL-373*			++-
10	DIAMETER: REQUIREMENT SPECIFICATION	18 bi	N				
CIZ	BOARDING CITY	EQ SABER CITY	N N · X				
R	MARKS:						

7.52

Figure 7-12

Operands with State Relational Operators

	ALC: NOT THE OWNER OF THE OWNER OWNER OF THE OWNER OWN						
State Terms Operand Names	POSI- TIVE	NEGA- TIVE	NUMER- IC	ALPHA- BETIC	PRES- ENT	END OF FILE (EOF)	STATUS
File Name					X *	X**	
Field Name	X	X	X	X			
Reference List Function Name	X	X	x	X	X		
Arithmetic Expression Name	x	X					
Symbolic Expression Name	x	x	x	x			
Work Field Name	x	X	x	x			
Logical Expression Name							x

NOTE: In general, State Terms may be either operand 1 or operand 2. Readability is improved if the State Terms, except STATUS, are operand 2. STATUS reads best as operand 1.

X means the combination is permitted blank means the combination is not permitted

* random only

** serial only

Figure 7-13

	SYSTEM NAME	PROCEDURE NAME	IBM	ANALYST	DATE	1	
			DECISION TABLE			PAGE OF PI	NGES
TALE	TABLE WAME	TABLE TYPE		FREQUE	NCY		E
SSR	OFER OFERAND !	OP OPERAND2	OP " OPERAND 2	OP OF OPERAND2	OP OPERAND 2	OP * OPERAN	02
			of [02 08 04 04 05 06 07 08	09 10 11 12 15 14 15 16	17 16 19 20 21 22 25 24	25 26 27 28 29 3	0 31 32
100	NVENTORY	IS PRESENT	XN				
0200	MANTITY ON HAND	S	Positive	N EGATIVE			
150	DENT CODE I MATERIAL		IS NUMERIC	IS ALPHARETIC			
R	STATUS	15	M ARRI ED	SINGLE	DINORCED		
No.	STATUS	IS SWITCH A ON	Y - N				
32	TRAWSACTION	IS EOF	XN				
18	NITHERAWAL		IS EDF	NT LEGE			
1							
1							
1							-
ſ							T
Ē							
							-
1							
1							
							$\frac{1}{1}$
3	MARKS:						T
							Ι
					*		

Figure 7-14

7.54

*

There are two permissable condition statements that may contain a File Name as one of the operands. One statement pertains exclusively to serial type files, while the other pertains exclusively to random type files.

Serial Files - It is presumed that a serial file has a detectable end beyond which data is not recorded. The statement used to detect this condition has the following appearance:

file name

$$\left\{ {}^{\rm IS}_{\rm NT} \right\}$$

END OF FILE

The IS condition is true if the last record in the file has already been read. The NT condition is true if the last record in the file has not yet been read. Random Files - Because of the characteristics of random files, it is sometimes necessary to determine if specific data is recorded in the file. The statement used to make this determination may have the following forms:

file nam e PER logical expression name	$\left\{ {}^{\rm IS}_{\rm NT} \right\}$	PRESENT
file name PER location name	{ IS NT	PRESENT
file name PER logical expression name PER location name	$\left\{ {}^{\mathrm{IS}}_{\mathrm{NT}} \right\}$	PRESENT

The IS condition of the first form is true if any location is found whose

values satisfy the specified logical expression. If no location is found, the NT condition is true. The IS condition of the second form is true if the specified location exists, otherwise the NT condition is true.

The logical expression may be any logical expression permissable in TDL. The location name may be:

Field Name

Reference List Function Name

Arithmetic Expression Name

Symbolic Expression Name

Constant Name

Work Field Name

Literal

ACTION SUMMARY

The following action operators are used in TDL:

SET	WRITE
MOVE	BEGIN
DO	CLEAR
GOTO	SORT
READ	

With SET and MOVE only certain operands can be used. These are listed in Figure 7-15.

Operands	with	Assignment	Operators
----------	------	------------	-----------

	MOVE AN	D SET					
Sending Operand	Receiving Operand						
	Record Name	Field Name	Work Record Name	Work Field Name			
Record Name	*		*				
Field Name		X		X			
Reference List Function Name		X		X			
Arithmetic Expression Name		X		Х			
Symbolic Expression Name		X		X			
Constant Name		X		X			
Work Record Name	*		*				
Work Field Name		X		X			
Literal		X		X			
Connotative Term	X	X	X	Х			
Descriptive Term		X		X			

X = Individual value assignment.

* = Corresponding value assignment.

NOTE: With MOVE, operand 1 is sending operand, operand 2 is receiving operand.

With SET, operand 2 is sending operand, operand 1 is receiving operand.

Figure 7-15

DO and GO TO are used only with a Decision Table Name or identification numbers, or with a Procedure Name.

The use of input/output operators is summarized in Figure 7-16.

Input Output Operators

File Type		Serial	L			Rai	ndom	
Directional Type	Input	Output	Inter.	Ret'd.	Input	Output	Inter.	Ret'd.
Sequence Controlled by File Description Sheet	X		-	X				
Number of Records Available in Memory	1	a	b	1	1	a	1	1
Record is Deleted By				\odot				d
SORT		X	X					
READ	X		X	X	Х		Х	Х
WRITE - Record Name		X	x	x		X	x	x
WRITE - File Name				X			x	X
WRITE - Record Name ON File Name				X			x	X
CLEAR	5						X	X
BEGIN	X		X	X				

a 1 of each type

(b) 1 for READ, 1 of each type for WRITE

C READ no WRITE

(d) CLEAR

7.59