C<u>∦</u>2

DEPARTMENT OF ELECTRICAL ENGINEERING

CIRCULATING COPY Sea Grant Depository

UNIVERSITY OF WASHINGTON

COLLEGE OF ENGINEERING

SEATTLE, WASHINGTON 98195



+

A LARCE SCALE INFORMATION RETRIEVAL SYSTEM

by

.

Karun Khanna and A. D. C. Holden

Department of Electrical Engineering

Technical Report #156

- .

August 1972

This work was supported by the Washington Sea Grant Program, which is a part of the National Sea Grant Program under Grants NG-1-72 and 1-35320. The Sea Grant Program is supported by the National Oceanic and Atmospheric Administration.

CONTENTS

.

	1.	INTR	ODUCTION
		1.1	Research Objectives
		1.2	Survey of the State of Art in Generalized Data
			Management Systems
		1.3	Definitions of Terminology
	2.	MAJO	R COMPONENTS OF THE SYSTEM
		2.1	General Organization
		2.2	Primary Functions of System Elements
	3.	DESC	RIPTION OF COMMAND LANGUAGE
		3.1	Control Commands
-		3.2	Define Commands
		3.3	Loader Commands
		3.4	Retrieval Commands
		3.5	Update Commands
		3.6	Reporter Commands
	4.	FILE	ORGANISATION OF THE SYSTEM
		4.1	File Structure of the System
		4.2	Description of Tables
			(a) Summarization of table of contents
			(b) Explanation of pointers
		4.3	Analysis of File Structure
	5.	AN EX	KAMPLE OF DATA BASE CREATION AND USAGE
		5,1	Data Collection
		5.2	Data Structure
		5.3	Creation of Data Base
		5.4	Usage of Data Base

· ----

6.	CONCLUSIONS							
	6.1 Conclusions of the research							
	6.2 Recommendations for further research							
7.	BIBLIOGRAPHY							
8.	APPENDIX							

.

. -

.

· .

.

1. INTRODUCTION

1.1 RESEARCH OBJECTIVES

This research has been directed towards the analysis and evaluation of a generalized data base management system, and the design of improved data base creation and updating methods.

One of the problems confronting potential users of generalized systems is the selection of one of these systems from a wide variety of existing systems.

In an attempt to evaluate a system, the feature analysis proposed by the Codasyl Systems Committee [8] provide helpful guidelines.

Though these systems are generalized, in the sense that they provide the necessary file processing capabilities, most of them are biased towards the direction of providing rapid responses to information requests.

Unfortunately this bias has resulted in providing cumbersome techniques for actually creating the data base to hold this large mass of information.

It has been felt by designers of generalized systems that the creation of a data base is a one time procedure and so the facilitation of this procedure has not received adequate attention.

Another limitation of generalized systems dedicated to rapid response is the inability to handle massive changes to the data base, i.e. dynamic updating, in a rapid manner.

The Multi-Access Retrieval System-MARS VI [17] has served as a focal point in this research effort.

The prime reason for considering the MARS VI system is that it is typical of the more recent generalized data management systems. An added reason for considering it has been its availability for experimental use on the existing computer facilities at the U.W. Computer Center. This system is typical of the more recent generalized data management systems. The design philosophy has been directed towards providing:

(a) a reasonably complete set of general purpose data management
 capabilities,

(b) a command language which is simple, easy to learn, not prone to syntax errors and oriented towards non-programmer users,

(c) hierarchical data handling capabilities with multi-level nesting,

(d) internal structures for rapid response to queries and

(e) facilities allowing multi-access in a real-time environment

However, the system has severe input data restrictions and these restrictions hamper the creation of the actual data base though they in no way inhibit the user from easily specifying any complex hierarchical structure to relate the various elements of his data base.

The input format, used to create the data base, requires complex character manipulations which make the cost of creating the data base almost prohibitive, as FORTRAN and COMPASS on the CDC 6000 series are hardly oriented towards character manipulation in data strings. Hence, one of the major objectives of this research has been to modify the data base creation procedure, with a view to facilitating and increasing the flexibility of this procedure, as well as reducing the creation costs substantially.

The following facilities have been incorporated to meet this objective:

1. The syntax of the user language has been expanded to allow the user to create tables, which relate the key fields designated by the user to the actual physical location on the record. [Appendix A]

2. Provisions have been designed to allow the user to create tables which contain the relevant hierarchical association between the designated

key fields. [Appendix A].

The new MARS VI system employs these user created tables along with the definition tables which have been created by specifying the data base structure to validate and load actual data values into the data base.

In the original system the retrieved subsets of the data base were subject only to limited processing capabilities and no provisions had been made to input the retrieved subsets for other more powerful data analysis packages such as SPSS or BMD. [19, 2].

This capability has been provided in the new MARS system and the user is now able to conduct complex analyses of his retrieved data. This provision is important since the use of statistics is a major tool for the management of data resources.

In order to incorporate the facilities described above, additions have been made to syntax of the LOADER and RETRIEVAL modules [3.3, 3.4].

1.2 SURVEY OF THE STATE OF ART IN GENERALIZED DATA MANAGEMENT SYSTEMS -

Current trends in data management techniques are in the direction of development of generalized data management systems, i.e., systems of generalized programs which provide the common data management functions of storing, retrieving and updating data for a wide variety of applications.

There are two classes in generalized data base management systems. .

1. Host-language capabilities

2. Self-contained capabilities

The host-language capabilities are merely new tools for the applications programmer to facilitate the handling of complex data structures. Generally host-language systems offer the capability to define hierarchical structures. The host-language capabilities are merely built upon the facilities of a procedural language such as COBOL, PL/1 or FORTRAN.

As host-language capabilities are primarily within the realm of an applications programmer, the emphasis has been more towards the development of self-contained systems.

In using a generalized data management system with self-contained capabilities the user is actively involved. The user is required to explicitly define a desired data structure. A pre-programmed or built-in processing algorithm automatically allocates space, establishes directories and affords a degree of independence between the generalized programs and the data with which the program interacts. Depending on the degree of independence, programs can accommodate varying degrees of change in the definition and structure of the data without the necessity of modifying the program itself.

Self-contained generalized data base management systems are geared primarily towards the non-programmer. A whole new language is offered, which is non-procedural, i.e., free from the restraints of procedural languages like PL/1 or FORTRAN. The syntax of the language is oriented towards the natural languages and is rather simple to master.

Generally, user interaction is on-line from a remote terminal, but most systems are accessible through the batch mode. Typically, the user will not be involved with the actual creation of the data base but will be formulating and directing his own queries at the data base. Updating of the data base may be yet another user interaction.

The essential features of a generalized data base management system are listed below [8].

1. Data Definition - This permits a user to define a data structure in any arbitrary hierarchy of data elements within logical entries. The

defined data structure is generally different from the <u>storage</u> structure, which is the particular file organization used internally by the system.

2. Data Retrieval - The retrieval function involves making selective searches within a data base. Retrieval requests typically contain a list of data items or logical entries to be outputted depending on certain selected criteria. Various operations such as computations, sorting and reformating may be performed on the extracted data.

3. Data Base Creation - This involves either a built-in processing algorithm for constructing the data base, operating entirely on the user's data input or programming in a conventional sense to create a data input acceptable to the built-in function, provided by the system.

4. Updating the Data Base - This encompasses changing, adding to or deleting values of certain data items or certain logical entries within the data base. However, the data definition or the data structure of the logical entries may not be restructured as this would imply modifying the pre-stored data definition. Update is a process somewhat analogous to retrieval in that some part of the data base must first be selected. In most systems the selection facilities are modelled on those used in the interrogation function. However, once a certain part is selected or updated, it is changed in some defined way rather than displayed in a report.

Réview of some current data management systems - 📪

1. Generalized Information System (GIS) has been developed by IBM for the System 360 and is available both in the batch and interactive environment. The language is procedural with high level operators which are used to specify the processes of file definition, file creation, file updating, file sorting and file interrogation. Hierarchical structuring

of data is permitted. The system requires a fairly large machine and a fair amount of storage, with the batch processing version available on a Model 40 with 128 kilo-bytes of core and the interactive version available on a Model 50 with 312 kilo-bytes of core. Details of the system can be found in the applications manual [10].

2. ISL-1 - Data base definition, organization, creation, maintenance and interrogation are handled by generalized procedures invoked by the user in a non-procedural command type language. The interesting feature of this system is that the definition language for specifying the data structure desired is primarily in a tabular form, whereas the other processes are specified in a string language. Simple or complex hierarchical data structures are allowed. The system is available on an IBM 360/30, 40. Further specifications of the system are available in the Codasyl Systems Committee's report on Generalized Data Base Management Systems [1].

3. Mark IV - This system arose out of batch oriented data processing and still has some features left over from essentially tabulating or card system environment. Its input is by fixed preprinted forms. It does not have good facilities for easily generating new files and extracting data excerpts for report generation. It is available on a fairly wide variety of machines. The language is non-procedural and hierarchical data structures up to nine levels of nesting are allowed. It is suited for commercial information systems and is used fairly widely.

4. Time-shared Data Management System is a general-purpose system designed to permit users to define, generate, correct, maintain and retrieve information from large data bases. The language is non-procedural and is free-form, beginning with command words followed by command objects and

modifiers. The system is oriented towards rapid response to queries and operates in both the batch and the interactive mode. Hierarchical data structures in the form of trees are provided and certain computations can be performed on the retrieval data subsets. The system is operational on an IBM 360/50H, but can also be implemented on a larger configuration under the control of a suitable time-sharing executive. Specifications of the system are available from the Systems Development Corporation [3,4,22].

5. User Language/l System - is available as part of the standard software package for the Spectra 70. The philosophy of the system is to provide easy usage by non-programmers, including those interrogating or updating a data base from a remote terminal.

The batch processing version is available on 70 Model 45, 55 or 60, requiring 131K of memory. The time-sharing environment for both the timesharing and batch processing mode is available on a Spectra 70 model 46.

An interesting feature of this system is the capability provided for storing frequently used and therefore predictable interrogations and updates to the data base, so that these may be invoked by the lowest level of user who need know essentially nothing about the command language. Two technical papers have been published on UL/1 [21].

Although the systems discussed are similar in basic design approach, they vary widely in processing capabilities, language used, user orientation and file structure.

Generalized data base management systems which have built-in capabilities, offer considerably reduced set-up time and a vast reduction of the time required to prepare a new interrogation or an update to the data base. In this respect, such systems make major economies in the use of people's time and can also give a more rapid satisfaction of <u>ad hoc</u>

information requirements [8].

A recent trend in the development of data base management systems is the effort to produce a universal language for the description and manipulation of data stored within a data base. The intent of such a language is to describe a data base independently of any programming language [25].

The use of more than one programming language to access the same data, normally requires data conversion from one format to another. The cost of handling ever increasing volumes of data combined with increasingly complex processing demands have created the urgent need for a method to create and manipulate a data base which is common to all applications but independent of any particular programming language [24].

A new CODASYL Data Description Language Committee (DDLC) has been formed for the development of a common data description language to be developed on the efforts of Codasyl Data Base Task Group. The efforts of this group will probably lead to a comprehensive data manipulation language designed to further improve execution performance and system integrity.

1.3 DEFINITIONS OF TERMINOLOGY -

Definitions of terminology used in describing the MARS VI system are given below to aid in the understanding of the system features.

- a. A user-defined data base consists of sets of logical entries.
- b. A logical entry is a file of information built on the definition structure. The definition structure establishes the structural relationship between data items. Data items are entered in accordance with this definition in the logical entries.

c. A component is an element or node of the definition structure.

1) Element components are tree definition nodes which are assigned data values in the data tree.

2) A repeating group component serves as a parent to components specified as members of that repeating group set, linking all its descendants and thereby allowing the establishment of a hierarchical structure.

3) A repeating group set consists of all components, repeating groups and elements attached to the same repeating group component. The repeating group set is created at the repeating group level plus 1, i.e., if the repeating group component is at level 1, the repeating group set is created at level 2.

d. Level numbers are used to identify the level of hierarchy in the tree structure. The higher the level number, the lower is the level in the tree structure.

e. A multi-record data input file is one which contains different data records linked to each other through one or more components.

f. A data file is the collection of data records into logical units.

g. A data record is a collection of elementary data items.

h. A data item is the basic unit of information that the system stores, retrieves and processes.

i. A data base management system is one in which a controlled body of data is made available in machine-based form for retrieval, processing, and reporting.

2. MAJOR COMPONENTS OF THE SYSTEM

2.1 GENERAL ORGANIZATION -

The MARS VI system uses the modularity concept in its system architecture.

Software has been developed in small, independent and easily controlled modules each with a particular task.

Each of the modules is treated as an overlay. An overlay is a portion of a program written on a file in absolute form and loaded at execution time without relocation, to ensure the fastest possible loading speed as well as reduce the size of the resident loader.

When an overlay is loaded from a file, it is identified by its level number. The level number is a pair of two-digit octal numbers $(0-77_8)$ giving the primary and secondary overlay relationship. The first number is the primary level, the second is the secondary level. An overlay with a nonzero primary level and a zero secondary level (6,0) is a primary overlay. Any overlay with the same primary level and a non-zero secondary level (6,1) is associated with and subordinate to the corresponding primary and is called a secondary overlay. This difference is significant when overlays are loaded. Level (0,0) is reserved for the main overlay which remains in core memory during overlay execution.

The main overlay (0,0) is loaded first. All primary overlays are loaded at the same point immediately following the main overlay. Secondary overlays are loaded immediately following the primary overlay. Loading the next primary overlay destroys the first loaded primary overlay and any associated overlays. Similarly, the loading of a secondary overlay destroys a previously loaded secondary overlay [9]. There are a total of 33 overlays which comprise the MARS system in its entirety.

The overlay organization is

- 1. MAIN OVERLAY (0,0) CONTROL MODULE
- 2. PRIMARY OVERLAY (1,0) DEFINE MODULE

3. PRIMARY OVERLAY (2,0) - LOADER MODULE

4. PRIMARY OVERLAY (3,0) - UPDATE MODULE

5. PRIMARY OVERLAY (4,0) - RETRIEVAL MODULE

6. PRIMARY OVERLAY (6,0) - REPORTER MODULE

The above overlays are typically the ones the user interacts with, while utilizing the system. (See Fig. 2.1.)

2.2 PRIMARY FUNCTIONS OF SYSTEM ELEMENTS -

A. CONTROL MODULE - Communication between the user and the MARS VI system is handled by the control module. Basically the control module has two tasks. One is the supervising of user file operations, while the second entails the sequencing and execution of the other modules.

The system-user communication is handled through the following files

1. COMMAND FILE

2. REPORT FILE

3. MESSAGE FILE

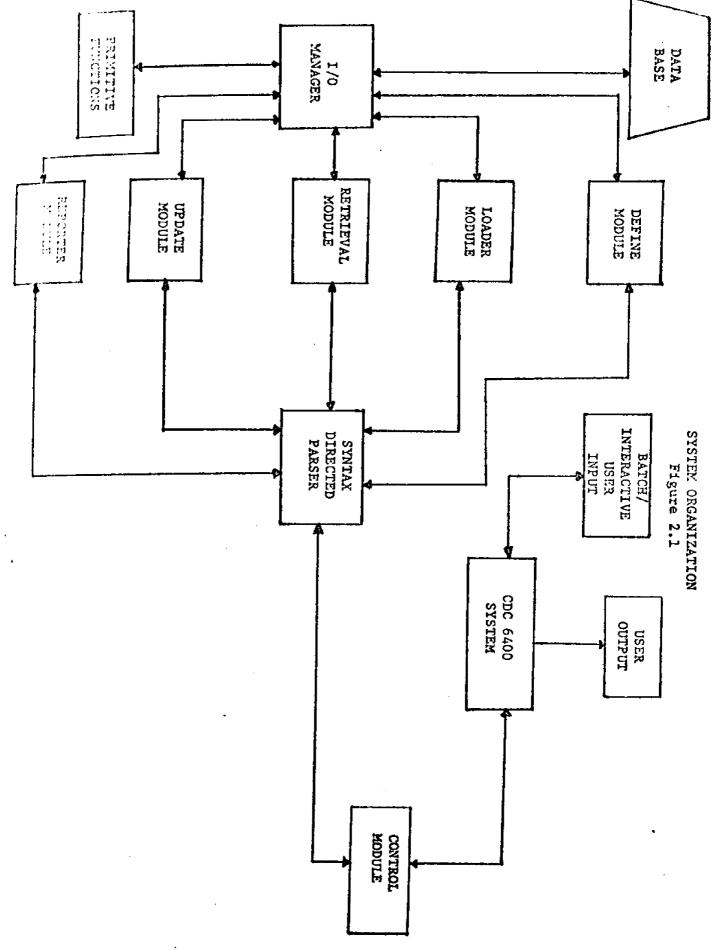
4. DATA FILE

The COMMAND file contains user directives which specify the step-wise processing of the user's data.

The MESSAGE file contains an echo of each user directive as it is encountered on the COMMAND file, as well as error messages due to errors incurred while executing the directives.

The REPORT file contains results obtained by the execution of directives on the COMMAND file.

The DATA file contains the user's data, which can be either in the inverter string format or in a user desired format.



The control module also interacts with the CDC 6400's SCOPE 3.3 operating system.

B. I/O MANAGER - This serves as a housekeeping system designed to relieve other modules from the task of locating and moving data in storage.

The I/O manager interacts with the data base and utilizes the PRIMITIVE FUNCTIONS.

The data base exists as a set of tables with each table being logically divided into partitions. Partition size is variable, but has been set for 256 central memory words at this installation.

A table contains entries with various items and the I/O manager uses the PRIMITIVE FUNCTIONS to store/retrieve an item or several items from a particular table entry in the table.

In addition the PRIMITIVE FUNCTIONS contain routines for converting relative addresses to absolute addresses as well as allocating partition space to the various tables in the Input/Output buffer area of the system.

The I/O manager handles all transfers of tables from external storage to disk and movement of the relevant partitions from disk storage to central memory.

C. SYNTAX-DIRECTED PARSER - The parser uses the syntax tables for parsing user directives.

The DEFINE, LOADER, RETRIEVAL, UPDATE and REPORTER modules each have their own syntax tables. Control module directives can be given in any of the other modules and hence the syntactical structure for all control directives is contained in each of the other modules. The parser uses a scanning routine for obtaining syntactical units. A syntactical unit is recognized as a string of alphanumeric characters from A through Z and O through 9. The syntactical unit is delimited by any CDC character which has a display code greater than 44 octal, i.e., semicolon, + etc. Each of the delimiters, serves as a syntactical unit by itself, except 55 octal which is a blank. Hence, the parser handles user directives with leading and trailing blanks, the blanks only being considered when appearing as delimiters between syntactical units.

D. DEFINE MODULE - This permits a user to describe the contents of the entries he desires in his data base, as well as define the organization in any arbitrary hierarchy of data elements within entries.

The DEFINE module must be used to create a data base before any other module can be called. Creation of a data base here only defines the structure of the data base. Data values are not loaded by the DEFINE module.

The defining feature is divided into two phases, the declaration and the finalizing phase.

During the declaration phase the user issues commands and definition entries, which he can modify at his own discretion.

The finalizing phase makes a data base definition permanent, by creating the definition tables which contain the user-defined component number and names, the type of data associated with each component, as well as the hierarchical relationship between the user-defined components. The data structure of entries in the data base cannot be restructured during the course of this session. A session is defined as the execution of user directives on the command file till an EXIT; directive is encountered, which

terminates the execution of directives and transfers control out of the MAYS VI system and back to the SCOPE supervisory system.

The user is permitted only one data base definition during a single MARS VI session.

E. LOADER MODULE - Actual data values to the data base are assigned by this module, in accordance with the finalized data base definition specified in the DEFINE module.

The LOADER module scans and analyses the input string before accepting it for loading into the data base. Various checkpoint reports at different stages of the loading as well as errors encountered in the input string can be displayed on the MESSAGE file.

The LOADER module can be called several times to add new logical entries to an already existing valued data base.

F. RETRIEVAL MODULE - The RETRIEVAL module permits to qualify, retrieve and output data at any level of data hierarchy from the data base. Relational Operators and Boolean Conditions are applied to data items in order to qualify data sets for retrieval.

Simple processing of data items is also permitted. The following arithmetic computations are performed on data base values

- 1. Average
- 2. Count
- 3. Maximum
- 4. Minimum
- 5. Sum
- 6. Standard deviation

In order to submit subsets of data retrieved to more complex processing, the user can generate an output file which can be then input to any FORTRAN program or statistical package such as SPSS or BMD [19,2].

G. UPDATE NODULE - This provides the capability of editing an existing data base through the use of directives incorporating addition, deletion, insertion and changing of data values.

Data value manipulation can involve entire logical entries, portions of logical entries, data sets or isolated data items.

The UPDATE module can in no way alter the definition structure specified in the DEFINE module.

The UPDATE module allows the user full use of retrieval specifications to qualify data sets and data items for modification.

H. REPORTER MODULE - It accepts a report format description and generates a report based on this description using the subsets of data retrieved by the RETRIEVAL module.

Typically the report format description consists of 1) the report title which is printed at the top of the first page only, 2) column headings which are printed after the title on the first page, and at the top of all subsequent pages, and 3) user-defined vertical and horizontal page spacing.

The MARS system can be activated in a batch processing mode, which includes access from remote teletypes through use of the U.W. TANDEM VERSION 1.1 system. TANDEM is a "conversational job entry" system. The TANDEM user may create and correct jobs at his terminal while enjoying almost instant turnaround. Once ready to run, the job is submitted from the terminal into the normal batch queue of the CDC 6400, and the results will be available

at the terminal (or on a printer) when it has completed execution [28].

Facilities for interactive processing will also be available with the implementation of the SCOPE 3.4 operating system, using its Intercom Fackage. In fact the system design is oriented towards the use of conversational operations from remote teletypes which greatly enhances the use of the simple command language.

3. DESCRIPTION OF COMMAND LANGUAGE

A syntax directed parsing system is used by the CONTROL, DEFINE, LOADER, RETRIEVAL, UPDATE, and REPORTER modules to interpret commands being issued by the user.

A metalanguage is used to describe the syntax, while the semantics are described in English, since no formal semantic language has yet been developed. The metalanguage used is BNF-like (Backus Normal Form or Backus-Naur Form). The notation does not adhere strictly to the formal definition of the metalanguage but is intended to indicate the purpose of each command [18,27].

The syntax rules as defined by the metalanguage specify which sequence of symbols are permissible; the semantic rules specify the meaning in terms of an (interpretive) execution of this string of symbols, i.e., what happens if the command is executed. In addition due to the restrictive expressive power of formal syntax rules, the semantic rules often give a further syntactic description.

The following metalanguage symbols are used in the BNF-like notation adopted:

denotes a metalinguistic variable. Symbols

<....>

themselves. Concatenation is indicated by juxtaposition.

::= is defined as
/ or
... repeats the previous syntactic unit
zero or more times.

[and] denotes choice of symbols. The set of
alternatives is usually stacked vertically.
; denotes end of command, or of a userdirective.

The intent of using a metalanguage to describe the syntax rules is to ensure conciseness, which apparently tends to be incompatible with readability and clarity. Hence a description of the semantics in English, coupled with the syntax description in a metalanguage provides a definition of the command language which is most meaningful and useful to the greatest number of users.

The following description of the commend language attempts to achieve the desired objectives.

3.1 CONTROL COMMANDS -

COMMAND FILE IS <filename>;

The command file contains user directives which describe the processing, in a step-wise fashion to be accomplished with the user's data.

REPORT FILE IS <filename>;

The report file contains results obtained from the execution of user directives on the command file.

MESSAGE FILE IS <filename>;

This contains errors incurred during the execution of the command file.

DATA FILE IS <filename>;

This contains the user's data in the loader input string format, or in a user-specified format.

The default for the command file in INPUT.

The default for the message and report file is OUTPUT.

<filename>::= 1-7 alphanumeric characters

EXIT:

This command is used to terminate the session and control passes over to the next Scope control card.

3.2 DEFINE COMMANDS -

DEFINE:

The DEFINE command is used to call the DEFINE module which is overlay (1,0) into the computer.

NEW DATA BASE <data base name>;

<data base name>::= 1 thru 20 alphanumeric characters

The user assigned data base name serves as an identification for a new data base and the DEFINE module initiates the definition version number to 1, and the data version number to zero.

The first component description in a new declaration must be a level O description, where the component is either an element type or a repeating group. The higher the level number, the lower is the level in the hierarchical structure.

<level 0 component description>::= <component description> (<type 0 description>)

component number>::= any 1-4 digit user defined number

cseparator symbol>::= <system defined separator>/<user-defined separator>
<system defined separator>::=)

cuser-defined separator>::= any legal non-alphanumeric display code character, not occurring in the data string

and excluding the blank, the "," and the ";"

<component name>::= any string of display code characters up to a
maximum of 150, excluding the user-defined

separator symbol and the "," and the ";"

<type 0 description>::= <repeating group type>/<element type>/
<repeating group type> <nulls option>/

< element type> <padding option>

<repeating group type>::= RG/REPEATING GROUP

A repeating group component type serves as a link for subordinate components and hence has no actual data associated with it. Repeating groups may be nested to a depth of 63 levels below the 0 level.

<element type>::= <name>/<date>/<integer number>/<decimal number>/

<exponential number>

<name>::= any string of display code characters up to a maximum of 255
<date>::= mm/dd/yy / mm/dd/yyyy

If the year is in the 20th century then only the last two digits of the year code are needed. the sign

the sign and the decimal point

responential number>::= a string of 20 characters of the form in nEim,

where n is an integer number, E is the base 10

exponent and m is an integer number limited

to a maximum of three digits

mulls option>::= WITH NULLS

spadding option>::= WITH <mm >PERCENT PADDING/WITH <amount> FUTURE

ADDITIONS

<mma>::= any integer number less than 60
<mmount>::= NO/FEW/SOME/MANY where,

NO = 0% FEW = 15% SOME = 30% MANY = 50%

<level n component description>::= <component description>(<type n description);</pre>

0< n ≤ 63

<type n description>::= <repeating group type> IN <parent component number>/

<element type> IN <parent component number>/

<repeating group type> <nulls option>/<element

type> <padding option>

<parent component number>::= any previously declared repeating group com-

ponent number

ical entry>::= <level 0 data set>/<dummy level 0 data set>/<logical entry> <repeating group data set>

beval 0 data set>::= data values that were entered for the defined level 0 elements, if any, for a specific

logical entry

cropeating group data set>::= <level n data set>/<dummy repeating group data set>

<level n data set>::= a single set of actual data values for a repeating

group

<dummy level 0 data set>::= data set at level 0 with no associated data
values

<dummy repeating group data set>::= repeating group data set with no

data values

DESCRIBE;

This allows a user to output his data base definition to the REPORT file.

CHANGE DATA BASE NAME TO <data base name>;

The user can alter the name of his data base by issuing the above command.

CHANGE NAME OF <component number> TO <component name>;

The component name associated with any component number can be changed. CHANGE NUMBER OF <component number₁> TO <component number₂> ; RENUMBER;

This command causes all user-defined component numbers to be renumbered in their current input string sequence order, with the first component number being set to 1, and each succeeding component number being incremented by 1. Other variations of this command are,

RENUMBER STARTING WITH <integer number> AND INCREMENTING BY <integer number>;

RENUMBER STARTING WITH <integer number>;

Each succeeding component number is incremented by 1.

RENUMBER INCREMENTING BY <integer number>;

The starting component number is set to 1 and the succeeding ones are incremented by the specified integer number.

CHANGE TYPE OF <component number> TO <type 0 description>/<type n description>;

This command may be used to change an element component type to another element component type or to a repeating group component, but it cannot be used to change a repeating group component to an element type, because the repeating group has descendants in the hierarchical structure, associated with it.

repeating group component.

DELETE <component number>;

DELETE <component number_> THRU <component number_>;

DELETE <component number> TO END;

The DELETE commands delete single definition input string entries or inclusive group of entries. "TO END" signifies the end of the definition input string entries.

INSERT BEFORE component number>;

Definition input string entry or entries may be inserted before or

ADD TO BOTTOM;

This command informs the module that the input string which follows in to be added to the end of the DEFINE input string.

The REORDER command is issued before the data base definition is finalized and enables the user to dictate the display order, which othervise parallels the DEFINE input string order of component occurrence.

The reordering options are:

REORDER BY NUMBER:

The data base declaration is reordered in an ascending order of the component number occurrences.

REORDER BY LEVEL;

The data base declaration is reordered by the level numbers in an ascending order i.e. all members belonging to the level 0 appear before all members belonging to the level 1 set.

RG REORDER BY REPEATING GROUP;

The data base declaration is reordered by grouping together all members of the same repeating group. All level 0 components are grouped at the start of the input string.

MAP;

This finalizes the definition of the data base. The definition tables are created and the data definition number is set to 1. Once the data base definition has been finalized by issuing the MAP command, the tree structure and the type descriptions associated with the various component names cannot be altered.

1.3 LOADER COMMANDS

LOADER;

The LOADER module which is OVERLAY (2,0) is invoked by the above remand.

ENTRY TERMINATOR IS <entry terminator>;

<entry terminator>::= 1-10 alphanumeric characters except ";"
The entry terminator is used to signal the end of a logical entry.

SEPARATOR IS <separator>;

<separator>::=

any CDC character except A thru Z, O thru 9,

blank or the ";"

The default value is ")"

NOTIFY MESSAGE FILE IF ANY ERRORS OCCUR,

DISPLAY LEVEL O ONLY ; ERRORS ONLY

The ENTIRE ENTRY option lists all accepted and rejected data values for the entire logical entry on the message file.

The LEVEL 0 ONLY option lists all level 0 data values which are accepted and all rejected data values on the message file.

The ERRORS ONLY option produces a list of rejected data values only.

Even one error in the data for a logical entry causes all <u>remaining</u> Upth for that logical entry to be rejected although the remainder of the duta string will also be scanned for further errors.

In general, each data value with its associated component number is printed on a single line; and the appropriate error message is given on the print line following an incorrect data value. All accepted data values and error-free data values which have been rejected are displayed with no error messages. A total count of the number of accepted and rejected tota values is given for each logical entry displayed because of error conditions in that logical entry.

ISSUE REPORT WHEN ALL LEGALITY CHECKED;

The user's data input string is scanned to the end and a checkpoint report consisting of the number of acceptable entries, number of rejected entries, number of values rejected, number of data sets accepted and the number of non-valued data sets accepted is given.

ISSUE REPORT WHEN SELECTION TABLES ARE COMPLETE;

This indicates that a report is to be outputted on to the report file when all data values for all the elements have been merged into the selection tables.

ISSUE REPORT WHEN FINAL SORT IS COMPLETE:

This indicates that a report is to be outputted on to the report file when the data sets have been sorted into data set order and the construction of the retrieval tables can begin.

ISSUE REPORT WHEN LOADING IS COMPLETE:

This indicates that a report is to be outputted onto the report file when the retrieval tables have been constructed and the LOADER module has

, scessfully loaded the input data string.

ISSUE REPORT WHEN ALL CHECKPOINTS OCCUR;

All four checkpoint reports mentioned above are to be outputted at the proper time on the report file.

STOP AFTER
SCAN

This directs the scanning program in the LOADER module to stop scanning the user's input data after the specified integer number of errors or after all the data has been scanned.

SUPPRESS COMMENTS:

The above command will suppress all user comments encountered in the data input string.

DATA FILE is <filename>;

The user's data file is identified by this command.

THE COMMANDS WHICH FOLLOW BELOW HAVE BEEN INTRODUCED IN THE SYNTAX FOR THE NEW MARS VI SYSTEM.

DIRECT FILE INPUT;

This command informs the system that a record description of the user's file follows, which will be used to create the tables which hold this description.

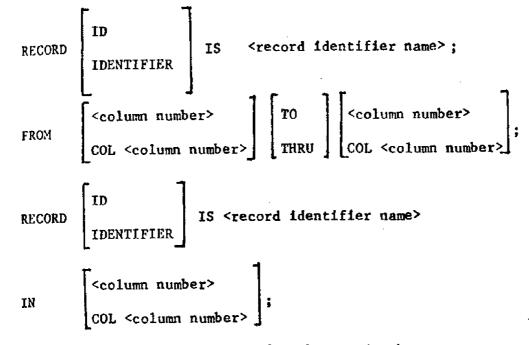
RECORD <number>;

<number>::=

any integer number from 1 to 10. The record number is mandatory only if the user has a multi-record data file.

In the case where this command is not included in the user-directives, the system assumes that the user's data file contains only one type of

record and the record number is assigned a value of 1.



<record identifier name>::= any 1-6 alphanumeric characters
<column number>::= any integer number from 1 to 640

The record identification is only required for a multi-record file, so that the system is able to distinguish between different kinds of records. Variable length user records are allowed and the maximum limit to the length of the user record is 64 words or 1 physical record [5].

KEY COMMON FIELDS FOR THIS LOGICAL ENTRY ARE C <component number>

<compare keys>::= C <component number>,/<compare keys >

 TO
 <column number>

 THRU
 COL <column number>
 .- ponent number> <separator> IN COL <number> ;

The location of the fields on the physical record of the user's data file, related to a particular component number description is made available $\frac{1}{100}$ issuing these commands. If the component number described is the last compenent number description in the user's record, then a semicolon must terminate $\frac{1}{100}$ it command. If other component number descriptions follow, then this particular component number description must <u>not</u> be terminated with a semicolon. The semicolon at the end of a component number description informs the system that the component number description for the user's record is complete and the necessary component description tables can be now created.

If the user has a multi-record data file then the following sequence of steps is necessary to create the various tables:

- 1) A unique record number has to be assigned to each new record.
- A unique record identification is to be associated with each new record.
- The various components occurring in each of the records have to be described.

SCAN;

Once a complete description of the user's data file has been submitted, this command proceeds to check the validity of data values, as well as loads actual data values into the data base, according to the definition structure specified. Hence a valued data base is only created after issuing this command.

SAVE DATA BASE ON <filename>;

Unless this command is issued a permanent copy of the data base is not -ndx. This will save the data base on a magnetic tape. The data base is regident on three files on this particular magnetic tape. The files contain

1. The header information, which includes the data base name, the definition version number and the data version number.

2. The actual data base which consists of,

- a. The Definition tables which have been constructed by the DEFINE module.
- b. The Selection tables which have been constructed by the LOADER module and which can be altered by the UPDATE module.
- c. The Retrieval tables which have been constructed by the LOADER module and can be altered by the UPDATE module.

3. This contains trailer records which signal the end of the data base.

The SAVE DATA BASE command can be issued at any time after the LOADER module has constructed the data base. However, it should be issued prior to issuing the EXIT; command which terminates the particular session.

3.4 RETRIEVAL COMMANDS -

RETRIEVAL;

This command is used to call the RETRIEVAL module, which is OVERLAY (4,0) into central memory. If the session starts with the creation of a data base, use of the DEFINE and LOADER modules must precede the use of the RETRIEVAL module. When the session starts with the use of the RETRIEVAL module, the appropriate data base must be loaded. This is accomplished by issuing the following command:

LOAD DATA BASE FROM <filename>;

From messages from the RETRIEVAL module are sent to the MESSAGE file, wereas the retrieved output is sent to the REPORT file. These files may as specified at any time and if not specifically given are set by default to the on-line printer.

Retrieval requests can have any one of the possible syntactic

PRINT <output collection> WHERE <retrieval specifications>;

PRINT <output collection>;

UNLOAD;

UNLOAD WHERE <retrieval specification>;

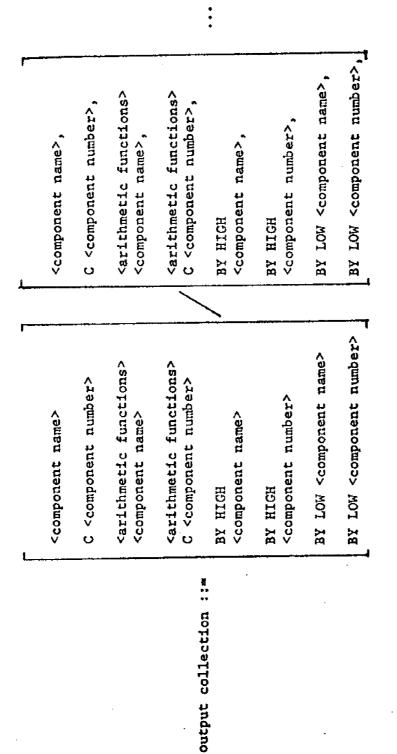
The PRINT request sends output result as desired by the user to the EEPORT file, whereas the UNLOAD request produces output in the LOADER string input format.

QUALIFY ON FILE <filename> <output collection> WHERE <retrieval specifications>;

QUALIFY ON FILE < filename> < output collection>;

THE QUALIFY COMMAND IS A FEATURE OF THE NEW MARS VI SYSTEM.

The qualify command is used to retrieve subsets of data and store them on either a disk file or a magnetic tape identified by the filename specified in the command. This retrieved subset of data can then be processed by any processing program, using this file as its data file.



carithmetic functions>::= COUNT/SUM/AVG/MAX/MIN/SIGMA

The scope of the output is defined by the component type. Components of element type which appear in the output list cause those elements respective values to be output. Components of repeating group type cause the values of all elements in those repeating groups and their subordinate repeating groups to be output.

The arithmetic functions apply only to the <u>element type</u> components, except the COUNT function.

The arithmetic functions perform the following tasks -

 COUNT - Totals the number of occurrences of an element value, repeating groups, data sets or logical entries.

2. MAX - Selects the highest data value assigned to an element. This function can be used with alphabetic or numeric data values. If used with alphabetic values, MAX is equal to the last alphabetic entry in a alphabetic listing.

3. MIN - Selects the lowest data value assigned to an element. This function can be used with alphabetic or numeric data values. If used with alphabetic values, MIN is equal to the first alphabetic entry in an alphabetic listing.

4. SUM - Totals the data values assigned to an element. This function can be used with numeric data values only.

5. AVG - Divides the total of the data values for an element by the total number of occurrences of data values for an element, i.e., AVG = SUM/COUNT. This function can only be used with numeric data values.

6. SIGMA - This computes the standard deviation of data values for an element. SIGMA can only be used with numeric data values.

The BY HIGH and the BY LOW refer to the sorting criteria to be performed. The "BY LOW" sort instruction will generate a normal 6400 display code ordering, except that a blank is always treated as the lowest character value. The "BY HIGH" sort instruction will generate an inverted display code ordering.

The sorting criteria is applicable only to the element type components.

It should be observed, that the term "ENTRY" is used to represent the level zero repeating group.

<retrieval specifications>::= <condition> AND
OR
NOT

<repeating group> HAS <condition>/ <retrieval specifications>

<condition>::= <component name> <relational operator> <user supplied value>/
 C <component number> <relational operator> <user-supplied
 value>/
 <component name> EXISTS/<component name> FAILS/
 C <component number> EXISTS/C<component number>
 FAILS

<relational operator>::= EQ/NE/LT/LE/GT/GE

<user-supplied value>::= data string supplied by the user

The components must be of type element in the "conditions" used in the "retrieval specifications"

The "HAS" clause permits designating a level of hierarchy for retrieval different than that used in the qualifying boolean expression.

.....WHERE SAME;

This expression specifies that the WHERE clause of the immediately preceeding request is to be used as the WHERE clause of this retrieval request.

DITTO WHERE.....;

This expression specifies that the output collection of the immediately preceeding request is to be used as the output collection of this retrieval request.

1.5 UPDATE COMMANDS -

The above command invokes the UPDATA module, which is OVERLAY (3,0). Let us first consider commands which perform single level modification i.e., modifications which affect only one level of data sets in the data base.

ADD <component name>/C <component number>/ENTRY/CO E0 <data string> WHERE <retrieval specifications>;

Within each selected data set, the ADD operation looks at the status of each element specified in the update request. If the element already has a value, no action is taken, otherwise the new user-supplied data string is added to the contents of the data set.

ENTRY or CO specifies the level 0 data set. The components may be of either type element or repeating group.

The "retrieval specifications" are the same as in the retrieval requests.

<data string>::= <data values> <separator> <separator> <entry
 terminator>

If the component type is repeating group then the <u>data values</u> are in the logical entry format.

<logical entry format>::= <data set> <separator> <separator> <entry

terminator>

<data set>::= <repeating group component number> <separator> <data items>/

<data set>

to a repeating group type component.

citata value>::= actual user assigned values to the components in the data base
creations component name>/C <component number>/ENTRY/CO EQ <data string> WHERE
creations pecifications>;

For each selected data set if the element has a value, it is changed to the new value supplied in the data string. If the element has no value, a) action is taken.

The other requirements are similar to the ADD operation.

specifications>;

If the specified component is of element type, then for each selected data set, if that element has a value, the value is removed, otherwise no action occurs.

If the specified component is of type repeating group or ENTRY, then every value is removed from every selected data set.

ASSIGN <component name>/C <component number>/ENTRY/CO EQ <data string> WHERE <retrieval specifications>;

The ASSIGN operation assigns unconditionally new values for each element specified in the update request, for each of the selected data set.

The other requirements are similar to the ADD and CHANGE operations.

Multi-level or tree modifications affect more than one level of data sets, i.e., descendent data sets that lie below the selected data sets, are also modified.

The components can only be of type repeating group in all TREE operations.

REMOVE TREE ENTRY/CO/ <component name>/C <component number> WHERE

This operation removes each selected data set and all of its descendent that set. Each selected data-set is the top-most node of a data tree. After each data tree has been removed, the remaining data trees in the that base are relinked to close the gap at the topmost level created by the removal.

ASSIGN TREE ENTRY/CO/ <component name>/C <component number> EQ <data <tring> WHERE <retrieval specifications>;

This operation performs, for each selected data set, a REMOVE TREE operation and then assigns an entirely new data tree containing the data string for the request. In the data string <u>all level 0 values must be</u> <u>prouped together at the beginning</u> of the data string. The level of assignment in the data string must never return to the first specified repeating group level, but descendents and their descendents can be assigned as many values as desired.

INSERT TREE ENTRY/CO/<component names>/C <component number> EQ <data string> BEFORE <retrieval specifications>;

This operation uses the selected data sets as a guide, never changing the contents of any selected data sets. The data string of new values is unlimited except that only one data set can be inserted at the level of the specified repeating group; the number of descendent data sets from the new topmost parent data set is unlimited.

If the connector word is BEFORE, the new data tree is inserted before each selected data set. If the connector word is AFTER, the new data tree is inserted after each selected data set.

.......PREVIOUS WHERE <retrieval specifications>;

The PREVIOUS option is similar to the DITTO option encountered in the obvial request, except that it pertains to the data string. When this of a is used the previous data string is simply used in this particular the request.

In order to make permanent the update actions executed, the updated in base must be saved on a magnetic tape, by issuing the save data base

3.6 REPORTER COMMANDS -

REPORTER;

This invokes the retrieval module which is overlay (6,0).

A report can only be generated if subsets of data have been retrieved from the data base by use of the RETRIEVAL module. However, if the retrieved data is to be used by the REPORTER module, all retrieval requests should contain the word QUALIFY, instead of the word PRINT. This causes the retrieved output to be structured for input to the RETRIEVAL module and assigns the extracted data to a SCOPE scratch file.

FORMAT;

This informs the module that a report format description is to follow. The report format descriptions of interest to the typical user are:

a. Report Title for Report

b. Column Heading for report

c. Components to be included in the report TITLE IS <1iteral string>;

<literal string>::= any alphanumeric string of display code

characters except the "," or the ";"/, <literal

string>;

This placed the literal string values on the first page of the report.

```
HEADING IS
H <1iteral string>;
```

This places the literal string values beneath the title lines on the first page and at the top of each subsequent report page. Headings are left justified.

 DETAIL IS
 <component name>
 /

 D
 C <component number>
 /

 C <component name>,
 ...;

 C <component number>,
 ...;

FINIS;

This indicates the end of the record format description.

GENERATE;

This causes the report to be printed on the REPORT file under the record format description.

4. FILE ORGANIZATION OF THE SYSTEM

Since the basic purpose of this system is to manage large and complex files of data effectively, the problem of file organization is of paramount importance. The system's file organization is critical in determining its speed of response, use of storage, maintenance of structural relationships among data, ability to manipulate data, and usefulness of output. Users of information systems whose needs are predefinable rarely + systems with complex file structuring capabilities. Serial-access reles on tape or random access storage devices with dictionaries containing x few known key words are adequate enough to meet the demands of such relential users.

However, most growing organizations are already handling their preoffinable needs to their satisfaction and are anxious to give a more theoly response to their nonpredefinable information needs, which usually originate from the higher management echelons.

The inverted file structure used as the basis for the file organization in this system is directed towards such ad hoc queries.

4.1 FILE STRUCTURE OF THE SYSTEM -

In the earlier days of file processing, a file was taken to be a collection of records, each of which was assigned a unique address. These unique addresses were collected into a set called the directory. In order to locate a set of records, having some common key word, all the addresses in the directory were used for bringing in the corresponding records for examination to see if it contained the desired key word. This, of course, constituted uneconomical file processing [11].

The inverted file structure was developed to increase file processing efficiency. In this type of structure, the directory contains a copy of its key words and each of the key words is associated with the addresses of all those records which contain that key word. Hence in order to find a set of records containing a common key word, one has only to scan the directory, and on finding the desired key word all the associated records are available.

In this system, the inverted file structure is totally organized and writed by component value and occurrences of each value. Any value of any ponent can be used as a key for retrieving information. All potential ways are grouped into blocked tables in a manner designed to minimize storage transfer requirements.

The definition tables define only the structure of the data base. After that values have been entered into the data base the data base now consists of the definition tables, the selection tables and the retrieval tables. The tables are partitioned into blocks of 256 central memory words and these partitions are transferred from disk to central memory as required by the system.

4.2 DESCRIPTION OF TABLES -

The table organization is schematically illustrated in Figure 4.1. A summarization of table contents and an explanation of the pointers is appended to clarify the schematic representation.

The definition tables are:

1. CDEFINA AND CDEFINB TABLES - Each entry in these tables corresponds to a component in the logical entry definition as declared by the user. Items in the tables contain properties and attributes for each component and the relationship among components as described by the user. The maximum number of components allowed in a data base is 1023 and the nesting of repeating groups is 63 below the level 0 components.

Hence in essence the CDEFINA table contains the logical entry structure ordered by user definition, whereas the CDEFINB table contains the user-

ethed component number and the type of value, associated with that poment.

GELS TABLE - Each entry in the CELS table corresponds to a component in the logical entry definition. A maximum of 150 characters is allowed in user-defined component names. This table is not ordered as it is a lessed by a relative address pointer from CDEFINB. This relative address pointer is the system-assigned component number, which is used throughat the selection and retrieval tables, rather than the user defined comrement number.

The selection tables are -

1. CVALDIR TABLE - Each entry in the CVALDIR table corresponds to a single (artition in the CVALUES table. For each element in the logical entry Nefinition for which data values have been entered, there are as many CVALDIR entries as the number of CVALUES partitions created for that element. (VALDIR entries need not be ordered, as the set of entries pertaining to a single item are linked. CVALDIR entries indicate the highest value in a CVALUES partition and, thus are used to point to the CVALUES partition where a data value may be found if it exists in the data base for the given element, i.e., it is a pointer to the associated block.

Thus CVALDIR is a page directory to values for each component having a value in the data base; and is accessed by item CPVDIR of the CDEFINE table.

2. CVALUES TABLE - Each unique data value assigned to a given element is represented by one and only one entry in the CVALUES table. Each CVALUES partition contains entries pertaining to only one element. Entries within

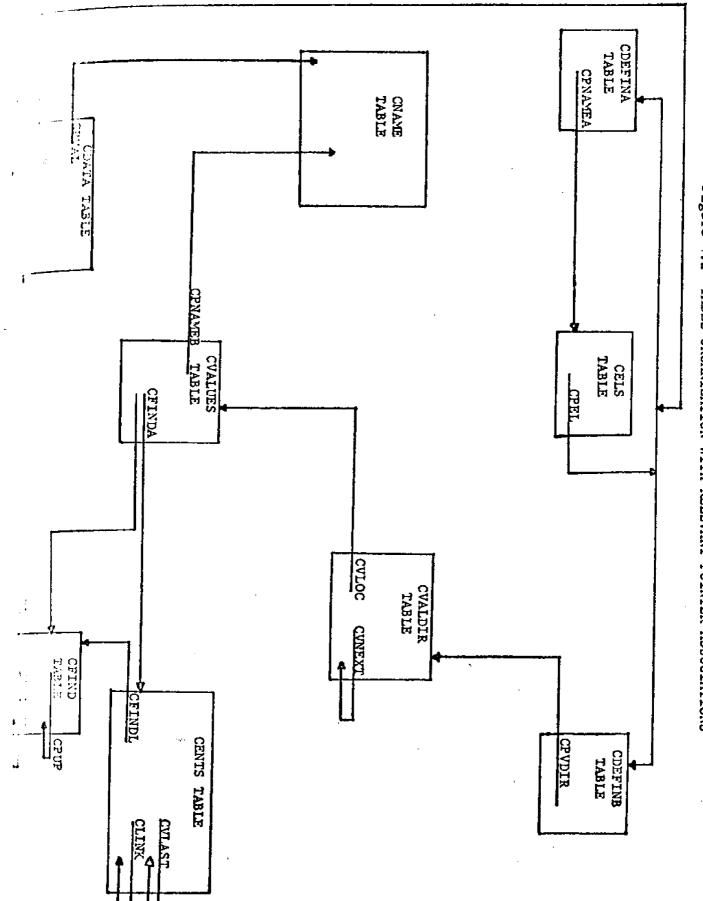


Figure 4.1 TABLE ORGANIZATION WITH RELEVANT POINTER ASSOCIATIONS

,

The partition are ordered by low value first with respect to the total the in table CNAME, but the set of partitions for any given element and not be ordered or consecutive in the CVALUES table. Only the first the characters of each unique value are stored, and there is a pointer to the full literal string in the CNAME table. If one were to attempt a theory search in such a table, the disc transfer time would be excessive. Theore the CVALDIR TABLE is used so that within one disc transfer the theory optimate block of value list can be located and consequently searched. 3. CENTS TABLE - Each value in the CVALUES is cross-indexed to the CENTS table where the occurrences of that value may be found. These occurrences are called "data sets." The first word of each list contains the count of occurrences. The list of set numbers is ordered from low to high. If the value is unique, instead of pointing to the CENTS table, CVALUES points directly to the CFIND table. There is an item in the values list indicating whether or not any value has occurred only once.

4. CNAME TABLE - A CNAME entry exists for each unique data value assigned to an element. If the value was assigned to a specific element in more than one data set, one and only one copy of the value is retained in the CNAME table, and it is pointed to by pointers in the appropriate CDATA table entires. Entries in the CNAME table are used in matching when uniqueness is not assured by the corresponding CVALUES entry and are used in generating report output.

The retrieval tables are:

1. CFIND TABLE - CFIND Table is indexed by the CENTS TABLE. It resolves any hierarchical retrieval problems and locates the entries in the data table.

 \cdots on tains an entry for each data set that exists in CDATA. If no , was were entered for a data set but other data sets exist at a deeper - ical level for the same family of data sets, then a dummy CFIND entry is created for the null parent set. There are four items in CFIND that ure used to accomplish retrieval of hierarchical data (1) a repeating group identifier (RGID), (2) a up pointer (3) right-pointer and (4) a tern-pointer. They have the following functions: the RGID identifies the group with which each CFIND entry (data set) is associated (the Hentifier at level 0 in each logical entry is always zero). The up- and down-pointers are relative pointers that contain positive values. The up-pointer contains the relative location of each data set's senior in the hierarchy tree (lower data set number). Except at level zero, the up-pointer always points up. At level zero, this points to the relative location of the next logical entry. The right-pointer contains the relative location of the next data set-number at the same level (higher data set number). If there are no more data sets in the logical entry at the same level, the right-pointer equals zero. The right-pointer links all data sets at the same level across family boundaries. The down-pointer points to the junior of each data set in the hierarchy tree (higher data set number). If a data set has a junior, it is guaranteed to be in the next higher data set.

The last piece of information in CFIND is a pointer to the location of the associated data set in CDATA. Hence, the CFIND table preserves the vertical and horizontal hierarchical structures within a logical entry and between logical entries.

CDATA TABLE - Each CDATA entry serves as a directory to access actual
values entered for specific elements in a single data set. Every
set in the data base that contains at least one data value has a
A entry (accessed by item CPLOC of the corresponding CFIND entry for
at data set). If the data set is a dummy entry in the CFIND table,
a no CDATA entry exists and item CPLOC=0. The CDATA TABLE contains a
ponent number which is a direct index back to the dictionary. It also
stains a pointer into the CNAME table.

CDATA is constructed at load time in the order in which data arrives. If there is no hierarchical information in the logical entry, then there is only one <u>data set</u> in each entry. That data set is defined as level 0. A data set in CDATA is created for every occurrence of repeating group information at one level. A data set may be thought of as entry within a entry. There is an item in CDATA that indicates when the end of a data set has occurred.

4.2 (a) SUMMARIZATION OF TABLE CONTENTS ---

1. CDEFINA TABLE - Logical entry structure declared by user definition.

- CELS TABLE String of display code characters comprising the user defined component name.
- 3. CDEFINB TABLE User-defined component number and type of value associated with the component.
- 4. CVALDIR TABLE Directory table to values for each component.
- 5. CVALUES TABLE Unique values of components in an ordered and paged fashion, containing truncated or binary value.
- 6. CNAME TABLE Display code string containing full display value; only one unique value for each component. The table is unordered.

- CENTS TABLE Pointers to the different data sets containing the same value for a component. The table is unordered.
- CFIND TABLE Hierarchical structure of data sets and logical entries in the data base. The table is unordered.
 CDATA TABLE - Grouped by data sets and containing existence is
- CDATA TABLE Grouped by data sets and containing pointers to the component number and the value associated with it.

...2 (b) EXPLANATION OF POINTERS -

- . CPNAMEA Pointer to the header word of the CELS table entry which contains the component name for this component.
- CPEL Pointer to the CDEFINA and CDEFINB table entries associated with this component name.
- 3. CPVDIR Pointer to the first CVALDIR entry for an element.
- CVNEXT Pointer to the next CVALDIR entry for the same element.
- 5. CVLOC Pointer to the first word of the relevant CVALUES partition.
- 6. CPNAMEB Pointer to the first word of the CNAME table entry which contains the entire value string in display code associated with this representation in the CVALUES entry.
- CPFINDA Pointer to the first word of the CENTS table if this value occurred more than once for this element; otherwise this is a pointer to the single CFIND table entry in which data set this value occurred.

- CVLAST Pointer to the last item in the entire CENTS entry for this unique data value.
- CLINK Pointer linking all blocks pertaining to multiple occurrences of the same unique data value.
- CFINDL Pointer to the CFIND table entry in which data set this value occurred.
- Pointer to the CFIND entry for the data set at level n-1 with which this level n data set is associated as a repeating group.
- CPRIGHT Pointer to the next data set on the horizontal chain of data sets at level n, which have the same parent data set at level n-1.
- CPDOWN Pointer to the CFIND entry for the data set at level n+1 associated with this level n data set.
- CPLOC Pointer to the beginning of the CDATA table entry for this CFIND entry if any data values have been entered for this data set.
- 15. CPVAL Pointer to the relevant CNAME table entry which in turn contains the actual data value associated with this element.
- 16. CPVAR Pointer to the CDEFINA and CDEFINB table entry for the element in this data set.
- 17. CHOOK Pointer which links a sequence of blocks that comprise
 one CDATA entry.

4.3 ANALYSIS OF FILE STRUCTURE -

As the data base is comprised of various tables, with each table being

(cally partitioned into blocks of 256 central memory words, the size the data base does not necessarily determine the speed of response to a pary, since only a very small portion of the data base needs to be examined in order to determine the entries which qualify for retrieval.

The important factors in determining response time are:

1. Number of Boolean expressions in the WHERE clause

2. Number of unique values occurring per component in each Boolean expression involved in the WHERE clause.

3. Number of occurrences of the desired value in each Boolean expression, i.e., number of occurrences of the desired value in the various bits sets.

In order to achieve this retrieval speed certain concessions have to be granted. The inverted file structure employed requires three times more storage space than that required by the original data, if all components in the data base are level 0 elements. Hence, in order to minimize storage requirements the concept of structuring data in a hierarchical structure must be employed.

The advantages of employing a hierarchical structure are:

1. Hierarchical structure is used to force physical proximity of the various child segments related to the parent segment.

2. A hierarchical structure is used to eliminate the storage of redundant data fields associated with the child structures, i.e., if these are a group of children records all with the same component value, the component value is relegated to the parent repeating group.

However, the use of hierarchical data structures at best allows the data base to occupy the same storage as the original data. But this is not a limiting factor as storage is becoming inexpensive, whereas

time people want to expend in order to get information is becoming re and more important.

Another important consideration of this file structure is its extreme contrivity to ordering. Most of the tables which comprised the data where ordered by component value or set number depending on the inforcation they contained. Hence, dynamic updating of the data base becomes a Hifficult problem. In the case, where only small changes are expected, small amount of space allocated in the selection and retrieval tables, by the use of the nulls and padding feature, provides for rapid updating. weaver, if the changes are massive, dynamic updating in this system cannot be performed very rapidly. In fact massive doses of updating would be tables out in the background mode (batch mode) rather than in the foreground or interactive mode.

Although this particular file organization requires more disk storage than other file organizations, it is particularly suited for rapid retrieval of information for ad hoc queries.

5. AN EXAMPLE OF DATA BASE CREATION AND USAGE

In an attempt to illustrate the potential of the new MARS VI system, the implementation of a fisheries information system for the analysis of eatch and effort data, will now be given.

5.1 DATA COLLECTION - The Washington State Department of Fisheries has been collecting catch and effort information regarding 40 odd species of groundfish off the coast of Washington State, British Columbia and Alaska, since 1953. There are approximately 150,000 records of data collected in the span 1953-1970. Each of these records is an 80 character card image which

Format with a density of 555 characters per inch (cpi). The various fields of information existing in a record are illustrated in Table 5.1 [13].

	Columns	· · · · · · · · · · · · · · · · · · ·			
Year	(1-2)	Last two digits of current year.			
Month/Date	(3-6)	Number of month (1-12) and date of month of landing i.e. when the interview is taken.			
Fort	(7-10)	Col. 7: Primary area i.e. Seattle and Bellingham, which represent two port clusters.			
		Col. 8-10: Secondary area i.e. port of landing within the port cluster.			
Days Out/Fished	(11-14)	Col. 11-12: Total number of days from time vessel leaves home port to day of return.			
		Col. 13-14: Total number of days fished.			
Boat /Class	(15-16)	Col. 15-17: Vessel code number.			
		Col. 18: Vessel class number.			
Fishing Aids	(19-21)	Col. 19-20: Blank: these columns are not presently being used.			
		Col. 21: Vessel fishing aid code for horsepower.			
Species	(22-24)	Code number used to designate a par- ticular species.			
Area	(25-30)	Pacific Marine Fisheries Commission - Col. (25-26).			
		Washington statistical areas - Col. (27-28).			
		Ground areas - Col. (29-30).			
TABLE 5.1					

Record Structure	of	Washington	State	Groundfich	Date
	<u> </u>		_ocace	01000011100	Dara

tal Hours	(31-33)	The total number of hours (to the nearest tenth) fished in each area and at a particular depth.				
.	(34)	Code number which best defines fishing efficiency e.g. effective fishing with adequate catch per effort is designated the code number 1.				
3 3	(35-36)	Number of drags in each area.				
; th	(37-45)	Col. 37-30: Minimum depth fished in each area.				
		Col. 40-42: Maximum depth fished in each area.				
		Col. 43-45: Average depth fished in each area.				
ıtch	(46-59)	Col. 46-52: Total catch of all species at a particular depth.				
		Col. 53-59: Estimated catch by species for a particular depth.				
Lercentage	(60)	Percentage of species in total catch.				
Frice	(61-63)	Price of species in cents per 1b.				
Vet Type	(66-67)	Net type code number.				
llesh	(68-69)	Mesh size to nearest tenth inch without decimal place.				
Limits	(70-71)	Limit restriction on species.				
Swarch Hours	·(72-73)	Actual hours spent on the grounds, running with fish finding gear operating, to the nearest hour.				
	(74-80)	Not presently being used.				

fourments - In essence each record contains the estimated catch in lbs. with reference to a particular area, a particular species, a particular depth, and a particular date. It should be noted that the date for a particular trip is the date of landing and is hence common to all records created for that trip.

TABLE 5.1 (Contd)

•

÷

The data records are divided into year files and within each year the data is sorted in the ascending order according to the following ort order,

1. Date of interview

2. Boat number

3. Washington State area

4. Depth of fishing

5.2 DATA STRUCTURE - In the groundfish interview data one card image represents "a series of drags by a particular boat, for a particular trip, catching any one species at any one depth in any one area." "Depth" is approximately by ten-fathom intervals.

The aggregated data, then, appears in this structure.

	(area - depth)	<i>¶</i> 1	species 1 species 2
Landing 1	(area - depth)	∛ 2	species 1 species 2
	(area - depth)	# 3	species 1 species 2
Landing 2	(area - depth)	#1	species 1 species 2
	(area - depth)	#2	species 1 species 2
	(area - depth)	#3	species 1 species 2
	(area - depth)	# 4	species 1 species 2

L

Landing-specific information includes date of landing, i.e., date interview is conducted, port of landing, boat number, boat class, fishing aids (horsepower of boat), days fished and days at sea.

Area-depth specific information includes area, depth, hours fished, areber of drags, total catch and hours searched.

Species-specific information includes species code, species catch, pertent level of species within total catch, price per pound, net type, mesh size and limits.

In essence each record contains the estimated catch in pounds with reference to a particular area, a aprticular species, a particular depth, and a particular trip (identified by date of landing).

5.3 CREATION OF DATA BASE - Having understood the mode of collection of data as well as the record structure of records in the data file, one can now proceed to define the structure of the data base. In order to illustrate the new MARS VI system, two structures for the data base will be given. The definition structure for the first data base will be constituted of linearly structured entries, whereas the definition structure for the second data base will be constituted of hierarchically structured entries.

The DEFINE section for Data Base I should be: [Appendix D].

DEFINE;

NEW DATA BASE DATA BASE 1;

- 1) DATE OF LANDING (INTEGER NUMBER)
- 4) PORT OF LANDING (INTEGER NUMBER)
- 6) TOTAL DATA SPENT ON TRIP (INTEGER NUMBER)
- 7) DAYS SPENT FISHING (INTEGER NUMBER)

- •) BOAT NUMBER (INTEGER NUMBER)
- 4) BOAT CLASS (INTEGER NUMBER)
- (1) WASHINGTON STATISTICAL AREA (INTEGER NUMBER)
- (5) TOTAL HOURS FISHED (INTEGER NUMBER)
- (1) NUMBER OF DRAGS (INTEGER NUMBER)
- (1) AVERAGE DEPTH OF FISHING (INTEGER NUMBER)
- (1) TOTAL CATCH OF ALL SPECIES (INTEGER NUMBER)
- 11) SPECIES CODE (INTEGER NUMBER)
- 22) TOTAL CATCH OF MAJOR SPECIES (INTEGER NUMBER)

MAP -

The linear structure formed by the above definition of the data base is illustrated in TABLE 5.2.

The actual loading of the data base according to the definition structure constructed above in order to create a valued data base, is accomplished by executing the following command statements: Level 0

Date of Landing Port of Landing Total Days Spent on Trip Days Spent Fishing Boat Number Catch Information Boat Class Wash. Statistical Area Total Hours Fished No. of Drags Ave. Depth of Fishing Total Catch of all Species Species Code Total Catch of Major Species

Linear Structure of Data Base Table 5.2 LOADER;

DATA FILE IS GFISH;

DIRECT FILE INPUT;

- 1) FROM COL 1 TO COL 6
- () IN COL 7
- b) COL 11-COL 12
- s) 15-17
- 3) IN COL 18
- 13) FROM COL 27 THRU COL 28
 - 15) 31 THRU 33
 - 17) COL 35 THRU COL 36
 - 20) 43-45
 - 21) COL 46-COL 52
 - 11) COL 22-COL 24
 - 22) FROM COL 53 to COL 59;

ISSUE REPORT WHEN ALL LEGALITY CHECKED;

SCAN;

The DEFINE section for Data Base II should be: [Appendix G]:

DEFINE;

NEW DATA BASE DATA BASE II;

- 1) DATE OF LANDING (INTEGER NUMBER)
- 4) PORT OF LANDING (INTEGER NUMBER)
 - 6) TOTAL DAYS SPENT ON TRIP (INTEGER NUMBER)
 - 7) DAYS SPENT FISHING (INTEGER NUMBER)
 - 8) BOAT NUMBER (INTEGER NUMBER)
 - 9) BOAT CLASS (INTEGER NUMBER)

MASHINGTON STATISTICAL AREA (INTEGER NUMBER)

TOTAL HOURS FISHED (INTEGER NUMBER)

() NUMBER OF DRAGS (INTEGER NUMBER)

() AVERAGE DEPTH OF FISHING (INTEGER NUMBER)

11) TOTAL CATCH OF ALL SPECIES (INTEGER NUMBER)

CATCH INFORMATION (REPEATING GROUP)

(1) SPECIES CODE (INTEGER NUMBER IN 30)

(1) TOTAL CATCH OF MAJOR SPECIES (INTEGER NUMBER IN 30)

1375

The hierarchical structure formed by the above definition of the data base is illustrated in Table 5.3.

Once the data structure of the data base has been defined one proceeds to actually loading the data values into the data base in order to create a valued data base. This is accomplished below:

LOADER;

DATA FILE IS GFISH;

DIRECT FILE INPUT;

KEY COMMON FIELDS FOR THIS LOGICAL ENTRY ARE C1, C8, C13, C20;

1) FROM COL 1 TO COL 6

4) IN COL 7

6) COL 11-COL 12

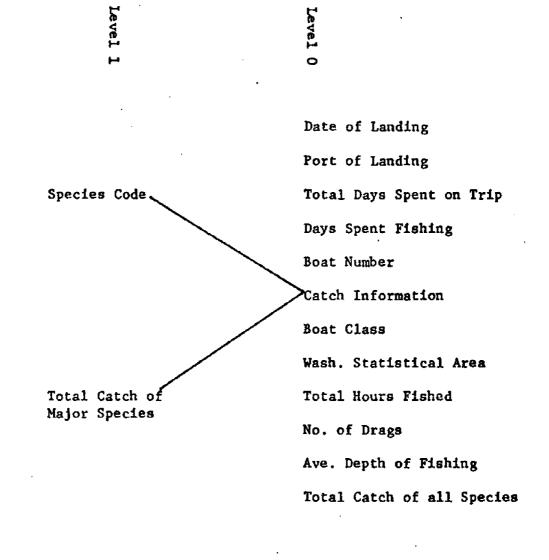
7) COL 13 THRU COL 14

8) 15-17

9) IN COL 18

13) FROM COL 27 THRU COL 28

15) 31 THRU 33



HIERARCHICAL STRUCTURE OF DATA BASE Table 5.3 COL 35 THRU COL 36

43-45

(1) COL 46-COL 52

(1) COL 22-COL 24

::) FROM COL 53 TO COL 59;

155UE REPORT WHEN ALL LEGALITY CHECKED;

SCAR

The key common fields are used to inform the system as to which fields are to be compared in order to identify records belonging to the same logical entry. In the case of this particular data structure,

Cl = Date of landing

C8 = Boat number

C13 = Washington Statistical Area

C20 = Average depth of fishing, are to be compared to achieve hierar-

5.4 USAGE OF DATA BASE - Potential users of the groundfish data base could be fisheries biologist, as well as management personnel from state agencies responsible for regulating and imposing effective controls on the groundfish industry.

The functions of retrieving subsets of data from the data base, conducting statistical analyses on the retrieved data, updating the data base, as well as generating reports would be available to potential users of the system. [Appendix H].

Retrieval requests which might be of interest to users of Data Base I and Data Base II are listed below with suitable comments to enhance the darity of the requests:

PRINT C1, C4, C8, C9, C13, C11, C22 WHERE C1 LT 540101;

The output collection contains the date of landing, port of landing, boat number, boat class, Washington Statistical Area, species code and the total catch of major species for all data of landings which were reported for the year 1953.

PRINT BOAT NUMBER, BOAT CLASS WHERE DATE OF LANDING EXISTS;

The output collection would contain all the boat numbers and their corresponding boat classes which reported even a single landing.

PRINT SUM C22, C11 WHERE C13 EQ O2 AND BOAT CLASS EQ 6;

The output collection would contain the sum of all major species and their species code for all species caught in Washington statistical area 02 by boats belonging to class 6.

PRINT ENTRY WHERE C9 GT 2 OR C13 LE 80 AND SPECIES CODE EQ 254;

The output collection would be a list of all data set(s) in the data base which have boat class greater than 2 or Washington Statistical area less than or equal to 80 and species code equal to 254.

For the hierarchically structured Data Base II it is further possible to issue the following request,

PRINT C1, C11, C22 WHERE CATCH INFORMATION HAS SPECIES CODE EQ 251 OR TOTAL CATCH OF MAJOR SPECIES LT 30000;

The output list would contain the data of landing, species code and total catch of major species for all data set(s) which satisfy the boolean expression in the WHERE clause. The interesting feature of this request is that if a certain data set in a particular repeating group family qualifies for retrieval, then all data sets belonging to that repeating group family also qualify for retrieval.

Other retrieval requests which conform to the syntactical structure precified in the description of the retrieval commands may be specified by the user. Diagnostic messages will occur if the syntactical structure is incorrect or if the WHERE clause is not satisfied [17].

If the user desires to manipulate data values in the data base involving logical entries, portions of logical entries, data sets or isolated data items, the update requests permit him this capability.

Some possible update request for Data Base I and Data Base II are listed below:

ADD BOAT CLASS EQ 3)) END WHERE BOAT NUMBER EQ 126;

The boat number 126 is assigned boat class 3 if that particular data item (i.e. boat class) does not already have a value. If it has a value, no action is taken.

CHANGE C4 EQ 1)) END WHERE C1 EQ 530506 and C8 EQ 199;

The port of landing is assigned the value 1 for the selected data set (s) if that data item already has a particular value. If the data item does not have a value, no action is taken.

REMOVE ENTRY WHERE C8 EQ 99 AND C13 EQ 10;

The entire level 0 data set(s) which satisfy the boolean expression are removed from the data base.

ASSIGN C13 EQ 02)) END WHERE C8 EQ 260;

All selected data sets are altered, whether the particular data item i.e. Washington Statistical Area has a value or not. Hence this command is unlike the ADD and CHANGE operations which are conditional.

The update requests below may be issued only for Data Base II:

ASSIGN CATCH INFORMATION EQ 11)254 22)25000)) END WHERE C11 EQ 254;

As the request specifies catch information which is a repeating group each selected data set is entirely empyied of all values and filled entirely with the values specified in this update request.

REMOVE TREE ENTRY WHERE C9 EQ 9;

This would remove all <u>logical entries</u> which satisfy the boolean expression. In this sense, tree operations affect more than one level of data sets, i.e., they are multi-level operations.

ASSIGN TREE CATCH INFORMATION EQ 11)251 22)15000)) END WHERE C1 LE 530501;

For each selected data set, a REMOVE TREE operation is performed and then an entirely new data tree containing the data string specified in the request is inserted in its place.

INSERT TREE ENTRY EQ 1) 530413 4) 2 6) 10 7)06 8) 151 9) 1 13) 08 15) 010 17) 25 20) 080 21) 55000 11) 256 22) 40000)) END BEFORE C1 EQ 530416;

The user specified logical entry is inserted in the data base before the logical entry (s) with date of landing 16th April 1953. The AFTER clause may be used instead of the BEFORE clause depending on where the insertion of the new logical entry is desired.

In case new logical entries are to be appended to the end of an already existing data base it is more efficient and economical to use the LOADER module. As an example, consider that catch and effort data for 1954 is to be integrated with the 1953 data base in order to develop a comprehensive data base. In order to develop a comprehensive data base which is linearly structured, the procedure outlined below is required:

LOAD DATA BASE FROM L1953;

LOADER;

DATA FILE IS DFISH;

DIRECT FILE INPUT;

- 1) FROM COL 1 TO COL 6
- 4) IN COL 7
- 6) COL 11-COL 12
- 7) COL 13 THRU COL 14
- 8) 15-17
- 9) IN COL 18
- 13) FROM COL 27 THRU COL 28
- 15) 31 THRU 33
- 17) COL 35 THRU COL 36
- 20) 43-45
- 21) COL 46-COL 52
- 11) COL 22-COL 24
- 22) FROM COL 53 TO COL 59;

ISSUE REPORT WHEN ALL LEGALITY CHECKED;

SCAN;

SAVE DATA BASE ON L5354;

EXIT:

It was assumed that the 1953 linearly structured groundfish data base i.e. Data Base I was available on filename L1953; whereas the 1954 groundfish catch and effort data is available on filename DFISH.

The comprehensive linearly structured data base for 1953 and 1954 groundfish data is now saved on a magnetic tape, identified by its filename, L5354. For the purpose of creating a comprehensive hierarchically structured data base, it is assumed that the 1953 hierarchically structured groundfish data base is available on filename H1953, whereas the 1954 groundfish catch and effort data is available on filename DFISH.

The procedure outlined below is required to achieve the desired - objective:

LOAD DATA BASE FROM H1953;

LOADER;

DATA FILE IS DFISH;

DIRECT FILE INPUT;

KEY COMMON FIELDS FOR THIS LOGICAL ENTRY ARE C1, C8, C13, C20;

- 1) FROM COL 1 to COL 6
- 4) IN COL 7
- 6) COL 11-COL 12
- 7) COL 13 THRU COL 14
- 8) 15-17
- 9) IN COL 18
- 13) FROM COL 27 THRU COL 28
- 15) 31 THRU 33
- 17) COL 35 THRU COL 36
- 20) 43-45
- 21) COL 46-COL 52
- 11) COL 22-COL 24
- 22) FROM COL 53 TO COL 59;

ISSUE REPORT WHEN ALL LEGALITY CHECKED;

SCAN;

WE DATA BASE ON H5354;

EXIT;

The comprehensive data base for the 1953-1954 groundfish catch and information data would now be available on filename H5354. It is mandatory that the filename be a magnetic tape, otherwise the data base will not be saved.

In order to generate reports of the retrieved subsets of data, the following procedure should be applied for data retrieved from either Data Base I or Data Base II:

RETRIEVAL;

QUALIFY C1, C9, C13, C21 WHERE C1 EXISTS;

QUALIFY C8, C9 WHERE C1 EXISTS;

REPORTER;

FORMAT;

LINES PER PAGE = 55;

TITLE IS GROUNDFISH 1953 CATCH REPORT;

NEADING IS DATA OF LANDING, BOAT CLASS, WASHINGTON STATISTICAL AREA, TOTAL

CATCH OF ALL SPECIES;

DETAIL IS C1, C9, C13, C21;

FINIS;

FORMAT;

LINES PER PAGE = 40;

TITLE IS GROUNDFISH 1953 BOAT INFORMATION:

HEADING IS BOAT NUMBER, BOAT CLASS;

DETAIL IS C8, C9;

SP 13 IN D1;

FINIS;

ORTER;

UETERATE;

Statistical analyses of the retrieved subsets of data from Data Base I and Data Base II may be performed either in the same session, or saved on a magnetic tape for input to a statistical package or program at a later time. However, the same retrieval request is required in either case.

OUALIFY ON FILE SRETV C1, C13, C21, C22, C11 WHERE C11 EQ 251;

This would retrieve all data set(s) which satisfy the boolean expression and the output collection would be available on filename SRETV, which may be either a disk scratch file or a magnetic tape. The format specifications for the output collection would be in accordance with the format specified by the user's input data file.

6. CONCLUSIONS

6.1 CONCLUSIONS OF THE RESEARCH

In order to compare the performance of the new MARS VI system with the old MARS VI system, two data bases were created using both systems.

Both the data bases used 2101 records from the 1953 groundfish catch and information data. Data Base I was linearly structured and hence was composed of 2101 logical entries, each logical entry having 13 different attributes. Data Base II was hierarchically structured and this hierarchical structuring, was composed of 648 logical entries having 14 different attributes.

The cost of creating the two data bases in the old and the new MARS VI system is summarized in Table 6.1.

The cost of creating a data base using the new MARS VI system can be

· 67

TABLE 6.1

COST OF CREATING DATA BASE

NEW MARS VI SYSTEM	CP = 256.409 SEC	PP = 115.019 SEC	IO = 068.915 SEC	COST = \$35.54	CT = 106.816 SEC	PP = 062.779 SEC	IO = 025.925 SEC	COST = \$14.64	
OLD MARS VI SYSTEM	CP = 351.346 SEC	PP = 065.042 SEC	IO = 079.393 SEC	. COST = \$41.14	CP = 161.324 SEC	PP = 037.373 SEC	IO = 033.278 SEC	COST = \$19.33	
DATA BASE I					DATA BACD	II			

further reduced by decreasing the field length requirements. Presently the new MARS VI system requires a field length of 75000 (octal) central memory words whereas the old MARS VI system requires a field length of 60000 (octal) central memory words. This increase in the field length of the new MARS VI system came about as a result of the new programs added to enhance the old system. However by causing these new programs to be treated as a separate secondary overlay, the field length may be reduced back to 60K (octal) central memory words.

It is expected that the reduction in the field length requirements of the new MARS VI system will cut the costs indicated in Table 6.1 as the job cost formula is,

 $C_{i} = P * [CP_{i} + FL_{i} (PP_{i} + 4 * IO_{i})]$

where

P = job priority (1-4) C_i = the cost of the ith step CP_i = the central processer time used in the ith step FL_i = fraction of the central memory used in the ith step (200,000 octal = 100% of central memory) PP_i = peripheral processor time used during the ith step IO_i = disk channel time used during the ith step. The actual % reductions expected because of the field length reduction from 75K (octal) to 60K (octal) will be: Data Base I -% reduction in cost = 12.075 Data Base II -% reduction in cost = 12.183 This research effort has successfully achieved the following objec-

(a) The severe input data restrictions which caused enormous problems to create the data base have been removed. The user of the \underline{new} MARS VI system does not have to concern himself with the peculiar input required by the <u>old MARS VI system</u>, but merely describes the entries of his file in the new syntax of the LOADER module. The new syntax is easy to understand. Consequently the creation of a data base is simplified.

(b) The introduction of the QUALIFY command in the RETRIEVAL module of the <u>new MARS VI</u> system has given the user a powerful tool to conduct complex analyses of his retrieved data, a feature which was not available in the old MARS VI system.

This feature is significant, as only a few of the current generalized data management systems offer this capability [1].

An area which could not be considered in this research was the quantitative analysis of retrieval and update efficiency.

The reason for this was the unavailability of interactive facilities on the CDC 6400 system. This made it impossible to determine on-line time response which could have served as a useful measure of retrieval and update efficiency.

In order to successfully achieve the research objectives, major changes were necessary to the CONTROL module and the LOADER module, while minor modifications were required in the RETRIEVAL module.

In making these changes it was strongly felt that even with the modularity architecture employed in the MARS VI system, it is a difficult task to incorporate new features. However, as has been demonstrated by this research effort, the original system can be enhanced to provide greater flexibility and ease of usage.

6.2 RECOMMENDATIONS FOR FURTHER RESEARCH -

During the past decade there has been a significant amount of research devoted to the design and construction of natural language questionanswering systems. Simmons' article [26] provides an excellent survey of work in that field. Paralleling the development of natural language question-answering systems has been the development of generalized data management systems.

Unfortunately there has been distinct differences in the scope and orientation of the activities in question-answering and data base management systems. Researchers in the question-answering area have stressed the development techniques to interpret the meaning of questions posed in English. Their main concern has been on translating natural language questions into a form that can be answered by direct retrieval from a file, and small-scale experimental computer models have been built to test these techniques. In a striking contrast, work on data management systems has been aimed at early operational use and has emphasized techniques for efficiently storing and rapidly accessing large quantities of formatted data, while relying on a relatively simple artificial query language for data retrieval purposes.

An interesting system could evolve, if these two developing technologies could be suitable combined to achieve a capability of interrogating large formatted data bases in a subset of English with reasonable on-line response times.

Kellog, in his article [12] presents some evidence for the thesis that, for some kinds of structured data bases, an English subset may be usefully realizable within the not-too-distant future.

The approach which has been suggested is the formal specification of a query language, a similar specification of the grammar for an English subset, and the description of a semantic interpretation procedure that allows questions to be translated into queries that will fetch or compute relevant facts.

Hence, in order to design a system where users may converse with large computer data files in an interactive mode using subsets of ordinary English, one would require the following basic units in the system.

1. A question-processing sub-system to translate English questions into query-language form

2. A generalized data management system like the MARS VI system which would serve as the answering subsystem.

Systems which link together the convenience of phrasing questions in English with the data storage and retrieval system appear to be the developing trend in this area.

A meaningful continuation of this research would be the development of the system proposed above. However, in order to widen the class of translatable questions it is suggested that a significant attempt be made to extend the expressiveness of the query language.

Research directed towards this goal would certainly lead to improved man-machine communication and make it possible for a computer system not only to allow economical representation of and rapid access to facts stored in the system but <u>also</u> assist in the logical analysis of the facts stored within the system.

7. BIBLIOGRAPHY

- 1. <u>A Survey of Generalized Data Base Management Systems</u>. CODASYL Systems Committee, May 1969.
- 2. Biomedical Computer Programs BMD University of California, 1971.
- 3. Bleier, R. E. <u>Treating Hierarchical Data Structures in the SDC Time-Shared Data Management System</u>. Proceedings of the 22nd National Conference, 1967, p. 67, 41-49.
- 4. Bleier, R.E. and Vorhaus, A.H. <u>File Organization in the SDC Time-</u> Shared Data Management System, IFIP, F92 - F97, 1968.
- 5. CDC 6400 Users Guide. University of Washington Computer Center, 1971.
- 6. Dodd, George G. <u>Elements of Data Management Systems</u>. Computing Surveys, 117-133, June 1969.
- 7. Everett, Gerald D. <u>Remote File Management System</u>. University of Texas Computation Center, Austin, Reports TSD 0-6, 1969.
- 8. Feature Analysis of Generalized Data Base Management Systems. CODASYL Systems Communications of the ACM, 1971 14(5), 309-318.
- 9. Fortran Reference Manual. Control Data 6400/6500/6600 Computer System, 1969.
- 10. GIS Application Description Manual. IBM Document, H20-0574.
- 11. Hsiao, D. and Harary F. <u>A Formal System for Information Retrieval from</u> <u>Files</u>. Communications of the ACM, 1970, 13(2), 67-73.
- 12. Kellogg, C.H. <u>On-Line Translation of Natural Language Questions into</u> <u>Artificial Language Queries</u>. SDC Document, SP-2827, 1967.
- Khanna, Karun. Information Storage and Retrieval Mini-system for Washington State Groundfish Data. Information Systems NW02, Norfish, University of Washington, August 1971.
- 14. Khanna, Karun. <u>Report on MARS VI System</u>. Information System, NIO5, Norfish, University of Washington, August 1971.
- Khanna, Karun. <u>A Multi-Access Retrieval System for the CDC 6400</u>. Information Systems, NI17, Norfish, University of Washington, April 1972.
- 16. Khanna K., Buss, J., Reeves, J. <u>Upgrading of the 1953-1966 Washington</u> <u>Groundfish Data</u>. Information System, NW10, Norfish, University of Washington, September 1971.
- 17. MARS VI Multi-Access Retrieval System For Full Inversion. Control Data 6400/6500/6600 Computer System. Reference Manual, 1971.

- Naur, P. <u>Revised Report on the Algorithmic Language ALGOL 60</u>. Communications of the ACM, Jan. 1963, 6(1), 1-17.
- 19. Nie, N.H., Bent, D.H., and Hull, C.H. SPSS Statistical Package for the Social Sciences, 1970.
- . Olle, T.W. MIS: Data Bases. Datamation, Nov. 1970, 47-50.
- 21. Olle, T.W. <u>UL/1: A User Language for Document Applications</u>. Proceedings of the ASIS Conference, Vol. 5, 1968, 151-153.
- 2. Raucher, V.L. and Schwimner, H.S. <u>The Basic Language Specifications for</u> the <u>Time-Shared Data Management System</u>. SDC Document TM-3370, 1967.
- Rangel, R.G. <u>A Design Approach to User Customized Information Systems</u>. FJCC, 1968, 171-177.
- 24. Report of the CODASYL Data Base Task Group. April 1971.
- 25. Schubert, R.F. <u>Basic Concepts in Data Base Management Systems</u>. Datamation, July 1972, 42-47.
- 26. Simmons, R.F. <u>Answering English Questions by Computer: A Survey</u>. Communications of the SCM, 1965, 8(1), 53-70.
- Smith, L. and Golde, H. <u>Programming Language Descriptions</u>. Technical Report No. 69-3-5, Computer Science Groups, University of Washington, March 1969.
- 28. TANDEM Users Guide. University of Washington Computer Center, Seattle, Washington, July 1971.

User defined component description tables for Data Base II Appendix A.

CTI TABLE

Mult1-level flag	40004040040000
User assigned component #	24948985554848 249898554848
Pointer to the beginning of a data set	00004040040000
Length of component	64000000000000000000000000000000000000
System assigned component	1004004001204
Start position of component	22266332285421 2226633228

. .

75

.

End of component table flag	0000000000000
Pointer to the next multi- level entry	vooorogooyoooo
Pointer to the data string table for comparison of multi- level entries	4000000000
System assigned component # of parent repeating group	000000000000000000000000000000000000000
Type Sf component	44444444 4 4 4 4
Level # of component	0000000000000000

CT2 TABLE

76

RUN VERSION	104 JUN 72 C 18115 06/02/72				PAGE 031	
					•	
	PROGRAM NORFSH (AIN, DOUT, OUTPUT) Sectnent tidefrontput)	, TAPE1=01N, TAPE2=00UT, INPUT				
CC033	THERER WIG61, FED	00-100 00-000 000 000 000	11	•••		•
• KEUEDO	• 1 0415 4/ 14 141.141 2°14 14/	2414,242H 9	•		-	
		15				
	6 144,144,141, - 05 757 717 11-12 16-20					
	5 1H 115 1	2714 ;				-
	# 01 OF40 FUNED 10-14 21-25 4 17 4 174 - 21-25	2+1+ 2 2+1+ 2	•			
	2 11-21 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2	29-31				
	5 11 114, 11), 511 110111111, 11		-			
	1ES 22-24	36-13 •		:	•	·
-	1 14 MASH			-		•
: :	14 14 14 14 14 14 14 14 14 14 14 14 14 1					
-	1 I		-	· .		•
	01 C 12 - 10	56-97 •		· ·	•	
		62-66 -	-	-		
•	14 14 14 14 14 14 14 14 14 14 14 14 14 1			•		
-	• 22 SPECIES CATCH 52-59 65-75	+ SZ-69				
•	· (112-12-12-14)	TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT	1		•	
661633	•14 •24[N] 185, 243, 240 6414 440/64 •) 680/	· ·				
060633	PE 40 (5, 10					
60311 608611	4					
	1 + 1974.0 1 + 1974.0	аарир (5), N(7), N(8), N(15),		•		- ,
*****	C SYAC	T + OAYS FISHED	- - 	•	•	•
	LK(14) + N(20) + N(24) + N(25) + 30AT + 3	SPECIES SPECIES		•		-
	24(26) * (12) * (20) * (27) * (32) *	•		•	•	•
•	+ HSV3 (12)NA(12)NA(12)NA	HOURS FISHED				
		AVE DEPTH	•			
•	S.(56)			. ,	•.	•
·		, 2 (70), 2(79),		•		
	(98)X* (98)X* (98)X* (98)X* (98)X* (98)X4	(1) N (39) N (96)				-
	20 FORTHIT (721, 0%) 8 A1, 3%, 34 1, 2%, 3 40 - 100 - 1 - 400, 10	211 · CX · 327 · 14 · CUT · CV · A · CV		· · · · ·	•	
				•		• •
101139 503136	100 10110 (C) 101) EEO					
	131 FJ244T(A10) . E40 FILE2	•		•		٠
				•		,
			-	•		

.

L

ţ

2

•

		•				-																	•		,				•																								
																											•		•																						. ·		
•																																																					
•																																				•		-			_												
:																,				:											•										÷												
							•													-			-															-				•											
					•											٠				•							-				•					۰,						'											
<u>9</u> 9	Ē	29	2	21	59		- C Z 2	ž	2 1	2 2 2		: c 7 3	2 2	C Z	Ê	5	ç	ມ ຊີ		r c z z		2 2	2 9	2 Z	2	Ę.	ş	5	5	D C	29	e Z	C2	Ş	ŝ			ç.	ŝ	ş		; <u>c</u>	Ē	Û.X.	ĊŻ	ΞĒ	25	15	Ş	ę	5	£	
		22	2	23	22	::	-	1	1			1	2	2	2	2	2	2	= ;	24		-		2	2	2	2	23	2		22	2	2	2	23	22	-	7	2	22	2		-	2	2	22	22		2	2	Ξ	3	
0.0		0 C	10	E 1 (* 1	35	- L - L - L	30	e n c	50		5 C 5 C	2	20	2	55	ŝ	0	300	3	5 P 5 P 4 C	2 2 2 2	7 C		6	110	D.	0 i 1 n	0 1	NI (Pi (32	ם ר	ŝ	6	ŝ	0 I V-1	5 C A 1 7 S	20	1	e -		30	2	18	3	20	នះ	20	25	50.0	13	S.,		
		20	Ñ			-		1	16	1	12	10		2	<u>د</u> ،	ລ	ស្ត					ļ	ŝ	ត្				2	2	53	50	1	Ę.	5	2				<u>R</u>	2	55		5	2	2	N 6	÷Ē	20	1	- N	2	Ñ	
0.0	0	5 e	• •	ω,	•	3 6	- 0	» c	,	3 43	• e	. 6	, 0	0	o	0			n i	nφ	×ψ	n y	, in		ro.	e -	0	e i	0	p c	• c	· c	ą	o	а (20	0	•	• 2		ر, د د د	3	5	n 3	0	2 4	0		0	S	3	
1 1 2 4 1 4 4 1 4 4 1 4	1	- - -	1		11 H	14	5 (C 5 (C		, ř	'nř	: x			•	Ŧ	۹,		'n		, v , v , v	ì		, in	Ť	H.	J.	5	-	-		5		ະ້	5	č		1	15	57		-	() () ()	5	ě	σ.	Ċσ	r C	ŝ	ធ	. Un	ů,	ι. Γ	
N N			1.64	C4 8	J D	16	46	•		4 11	- 1		u tu	- N	¢ν	ŵ	(W)	N 1		d v	'n	e n	1	•	N	64	N I	N C	N I	vr	9 14		÷.	N	N 1	V P	1		-	N 1	u n	1 10		Ņ	N	N K	V N	5	• ••	~	N	N	
21175		20	- C1 -	0.		ο.	• -	f 🕶	• •	36	- C3	, c	3 63		0	12	0.	-	•	эc) C	3 C.		പ	0	- 	-1	0,	0 1		. 0) r3	· C>	÷.	ο.				•••	ພະ	5 C	a ∹a	0	U.	12 4	ാറ			• •	•	•	0	
		4 N	1 64 1 44	0 0 6 1				• r		- 17 - 17	2 45 5 17	, . , .	ы N : су	1 A.	~	າ ຄ	њ. г	∾ (~ (un hp	 		• • • •	i Ne S Pro	ŝ	N 1	ы. к. і	n i m i	N 1 M 1	~ •	u N o M) m	ເດຍ : ນາ-	ε. 10	יויי אוריי	<u>с</u> и	, iù 1 U	- 10-	10 I 10 I	nu e Hid	9 U U 9 U			ŝ	N	03 D.	u n u n	* fy * J	ואו; ב-י	- en	N I	∾ ∽	
			-		4.	• •	• •	• •	• Þ	4 9	• •	÷		1	-	17	h (-			•				ĥ	17	5		5		- 1-	÷	**	17	►. } +1 ;			5	5		1.		F.	1	ь. I • С	***		1	5	11	+1	ч	
104	51345	20	20(202	20		• * • *		ŝ			ŝ	20	101	101				5		5	5 U 5 C	50	000	101	5	500	88	20				101	.03			200	5	5	66	20	2 N 5 0	201	100	20		10	10	3	000	2	55	
e4 e4		4 4 9 8	1	 	-1 + -1 +	••	••	• •	4 F . A	4 U A,	• •	• •	• •	-	-	م ب	•1	- 1	~ ,	• •	+ + د +	4 - 1 -3	• ••		7 1	44	ы. Ni	1 ·	- 1 - 1	-* - 6 *	4 F 4 F		i et i Mi	т п	н, м,	-1 F -1 F	, tu 1		т. 61	14 1 N 8	4 4 4 4	4 74 3 N	1 7 1 1 64	т! Ф	۰۳. م	++++ •0-≤0		н-н 1-н	(4 4	1 H 1 H	ብ ይ	5	
<u> </u>	1	12	Î	5;	i f	÷	ì		Ì	Ì	• •					+		-	••		• •	• •	• ••	9	13	5	<u> </u>	<u> </u>	Î	2	ì	2	2	3	9	22	12	Ę	Î			5	Ē	11)	5	22		12	Ş	Ĥ	11	2	
521	11211	121	5	22	4.6) [) [1	12	1	20	2.5	52	33	Ň	ŝ	G j	20	15	1	121	1 25	3.25	5	5.2	S.	2		1	52	23	5	22	i c	5.0	12	522	66	36	12		2	2			26	. .	5	24	23	
1 2 1	51	* *	14			4 - 1			4 4 9 M 9 M		• •	1		1	331	* * * *	2	22	**	4 + 1 -1	• •			1	1 5	1 2 2	1. 1. 2.	1		- 1 - 1 - 1	. .	1	1	1				11			2	•	11	1	+* • •		4 H 1 M 1 M		- 	11	1	11 11 11	,
0.0 N N	N 3	۰ د ا	;;		4 . 1 .	- 0 - C	σ			i.		-1		3	4	\$	Ξ.								ø	i∿ i PD i	N 6 19 7	N (M f	N I	0 F	u N n M	N P	N						•••	. م			÷	÷.	ю.	<u>ю</u> «с	داده	t. C	4	-1 -1 -15	3 71	3 65	
5 6	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	6	÷.	6		ŝ	2	2	ŝ		5		-	÷	•	•	0	•		0 K			•		ŝ	6		តវ	6	6	2	9	â	•	0.4	•	9	٠	-0	¢ i	•	a a	•	ē	-	00	0 10	5		ê	ຣ	ê	
210	710		2		22	1	10				ĥ	1	î	2	2	2							-	11	111	2	21					10	7.0	Î,				2.5	2			12	1					, C	2	1	2	ĉ	
22	5163	2	23	22	3		2	ŝ	2	12	17	1	2	2	2	2	23	2:	1,	12	1	12	1	1	Ţ.	23	2	ہ د	2,	2	2	1	11	Ξ.	1	12	1	1	23	25	12	12	1	2	2	<u>с</u> (1	1.5	2	2	2	23	-	
2 2	, r. , 1,	-	2			-	1	1		1		12	-	1	:	7	-		:	1	: =		1	1	ï	÷.	23		22	::	1		7	4	, i ,		1	-	1;	::		1	Ĩ	2	1			12	1	ž			
55	1/51	12	5.1	22		1	1	3	2	5	: .	5	1	5	2	5	22	n 9		2	:2	: 3	5	5	?</td <td>2</td> <td>2</td> <td>21</td> <td>21</td> <td>1</td> <td></td> <td>13</td> <td>5</td> <td>ini Gi</td> <td>52</td> <td>5 U 5 U</td> <td>: 2</td> <td>5.5</td> <td>53</td> <td>52</td> <td>:5</td> <td>10</td> <td>512</td> <td>5</td> <td>2</td> <td>50</td> <td>33</td> <td>5</td> <td>5</td> <td>5.2</td> <td>5</td> <td>2</td> <td></td>	2	2	21	21	1		13	5	ini Gi	52	5 U 5 U	: 2	5.5	53	52	:5	10	512	5	2	50	33	5	5	5.2	5	2	
1 × 1														173	::									1.1	::3	73						1 - 1	212	\sim				5112		2 A 2 A 2 A 2 A 2 A 2 A 2 A 2 A 2 A 2 A		Ň	5112	\tilde{z}		2		2	22	5112	2.12	N N	
a a	22		23	2 3		1	1		; 2 ()	1	1	2	1	9		9	;;	2:	• •	1	-	: 3		ຊ	= ;		23	2:		1	10	କ	9	9	2:	13	19	9	2:	20	12	-	9	a :	21	29	17	-	=	4	7	A'	

Appendix C. Crantion of Duty Base E withs Low C. 2010 01 Manuary Content of C. 2010 01 Manuary Content of Cont

2011.111 CATCH TE ALL FORTESS (THIERER NUMMER) 11300-115 20051(410020 NUMER) 1213174 CATCH ST MAJDE SOSCIESS(INTEGER NUM958) 」1944年1456月94 それまではすしか。ACA(日前に6528 MUM458) 1517月14日 4月145 をはないではれていてい、MUM168) 1714月14日 55 35960(日本にのごう 4014月2) Gathin Bat Fields Intil at theil BERTAL DAYS STIRE ON THE CLATCES NUMBERS POSTAR SPECE FISHING CLATCAR MUMPERS LCIQCO' Data file 13 Adriu: 155uf fiboyi N424 All Ghigkadinis Orguo: INTERFECTS NUMBED MEM)116 2455 CENTRATEM 20532LVI 212977 65 Evite2/1220222 212927 05 Evite2/12202000 (a) The state of the second state (a) Turkatin dat 1921 [[Till 13 te th WFOSION 3.4F 12/31/73 H495 VI IS 21434 -:.!!.30 10

21.4....

141 Wurden fruitues accenter 1 2733 Murden fruitues accenter attm 2533 Murden fruitues fruitues a Prime fruitues fruitues a Prime fruitues fruitues accenter a Murdes fruitues onta SETS accented a IS LOADER CALL 2101 27 313 21.31.41. איזיה הסשרונונט גסע בובעימו ה 1 לסות. יווייתבע סד טיניטנ צגנטרג גרפן ואוג רוצאבעד ב 21.53.52. 47945 534515163 594 54587 5 4 16766 144454 05 141795 534545 562 7415 552454 1854275 SIH1 205 *172155 * 21.34.10. meage du Jarahie valurs 1014. www.eege du Jarahie valurs 31 715011177 15 01010 0.117

21+3++14+ 42457 004415757 F03 FL34ENT C 7 10141 Num953 OF HULPHUC VALUES F02 THIS ELE4ENT =

בנילאינד. איזאה הקיירדנא ראי בנבאראד ר דסוקן אטאאבע טר טאנראד עבטנג ראת בנבאראד ר

σ נהל בנקיבאד כ נסק דאור פנקיביי 21.54.27. MERCINELETER TOTAL NUMBER OF UNITED

21-34-77. MUTC CAMPLETED FOR ELEMENT C. 13 1014. Numary de Jairys. Values for this Element =

21.54.56. MC265 COMPERTY FOR ELEMENT C 17 1014. NUMPER OF SHIPS VALUES FOR THIS ELEMENT

ŝ

8

7

	· · · · · · · · · · · · · · · · · · ·	f		
MZRGE ROMPLETED FOR ELEMENT C 21 Of infong values for this flement = 410		• •		
FOR ELEVENT C 11 FOR THIS ELEMENT # 2				•
MERCE COMPLETED FOR SLEMENT C 22 OF UNIOUE VALUES FOR THIS ELEMENT = 246			-	
-				
SELECTION TABLES ARE COMPLETE				
	-	1	•	į
Solution and the second s				
LETED de aggeptable cutrizs = de values aggepted = de cutries with gejegte				
0¢ VALUES REJETTED * 0 *********************************			80	
ОР ИЛИ-VALUED DATA SETS ACCEPTED = 0 · · DF VULLS COEATED IN THIS JOB = 0 · · ULLS COEATED IN THIS JOB = 0				
L SITE OF FURTERT DATA BASE - 216 PARTITIONS	· · · · · · · · · · · · · · · · · · ·	I		•
TION VEASION 1 DATA VERSION 1				
20.32/72 SFILE1 FD		۲		
	-			
		••••••		
			•	
新聞		· ·	۰.	
	* 1 * 1			
and the second		₽ * S + ! !		
		1		
		•	-	
i		•		
ne an suit an ann an ann ann ann ann an an airte an an ann an ann an ann ann ann ann an				
	04PLETE 2101 04PLETE 2101 0 2101 0 2101 0 0 0 2101 0 0 0 0 0 0 0 2101 0 0 0 0 0 2101 0 0 0 2101 0 2101 0 2101 0 0 0 2101 0 2101 0 2101 0 2101 0 2101 0 2101 0 2101 0 2101 0 211 0 211 0 211 0 211 0 211 0 212 0 214 0 214 0 215 0 215 0 214 0 214 0 214 0 214 0 214 0 214 0 214	References 2001 AME CAMERS = 2001 S. 4CT-ROLOS = 2001 S. 4CT-ROLOS = 0 S. 4CT-ROLOS = 0 C. 4CLUS = 0 C. 4CLUS = 0 C. 4CLUS = 1001 S. 4CT-ROLOS = 0 C. 4CLUS = 1001 C.	ANG FUTURES = 2141 ANG FUTURES = 2141 FULLION = 0 FULLION = 0 FULL	

Creation of Data Base I using the new MARS VI system. Appendix D.

IIISPECIES COGE(INTEGER NUMBER) 22)IJTEL GATCM OF MAJOR SPECILS(INTEGER NUMBER) ALE A (INTEGER NUMBER) SPECIES (INTEGER NUMBER) HT-GER NUMBER) Fishthg(integer Number) HBER) 11С11 22-ССС 2-21F4См ССС >3 10 СОС 39; SSUE AEPJ4T ММен ALL LEGALITY СМЕСКЕD; COLLATEGEN NUMBERS LATEGER INTEGER HUMDE 4 ° CHO 4 HILGER NUMBE C31 27 THPU COL 26 e bi INTele ATISTICAL VERSICH 3-3F 12/01/70 GROUNDE 20 THAU COL 36 ž FILE IS TIY: ISHING 13 THAU CUL 31 70 × READT ь С 6-CJL 52 1-30L 12 LITEL CATCH C 1 1 3 - 11 11 - 1 7. 14 8 1 ŝ 02 4z SCAN 177 H 3

15.27.31.

ALL LECALITY IS CMLCKLO FOR THIS LOADER CALL Nummer, of acceptagle entries = 2101 Nummer, of values accepted = 27313 Nummer, of values accepted = 27313 Nummer, of values Acuento = 0 Nummer, of values Acuento = 0 Nummer, of values iscluded = 0 Nummer of Non-Valued Data Sets accepted =

GADUVICTISH 332LEV DEFINITION VERSION 1 DATA VERSION

45+47-06+ 06/02/72 54/L (4474 8452 (4) D01453; 54/L/1640 604Pleted East?

Ş

	n an an ann an ann an an an an an an an			t • , ,	: • • ••
745E 001				•	
					•
	,TAPE1=01N,TAPE2=00UT,INPUT, TH 04-11 04-05,07-00,10-11 25-14 ,14/,2°14 , 14 , 14 , 19 , 19 , 20-25 - 21 4 , 21 4 ,			281,6%,381,1441)	•
//////////////////////////////////////	5 6 2	4 8 4 4 6 4 4 8 4 4 5 8 8 5 8 8 6 8 6	69-75 7*14 , 7*14 , 7*14 , * CPOUP , N(7), N(8) 0UT + 0AYS FISH * SPECIES	MOURS FISHED . AVE DEPTH . AV	81,0764R) ,080473
//////////////////////////////////////		11111111111111111111111111111111111111	52-59 52-59 52-59 52-59 52-59 52-59 54 51 51 51 51 51 51 51 51 51 51	CULT:	20 - 00 1 (1 20 1 (20 1 (2 20 1 (20 1 (2 20 1 (2 20 1 (2 2 2 (2 2) (2)
0 21 MAR	PR0584 11495555 114955555 11495555 11495555 1149555 1149555 1149555 114955 114955 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 11495555 114955555 1149555555 11495555555555	11 11 11 11 11 11 11 11 11 11 11 11 11	162 Cat 124 Ca	AN (10) - 10 AN	CALL SU CALL SU CALL SU CALL SU
17////////////////////////////////////			6 0000 0 00000 0 00100 0 00100 0 00000 0 00000 0 0	4 4 8 8 - 4 8 8 	

	I		•	•	ť			.	• ••	Ŀ,		• - ·	г	् १ ुः -		• • •	.			ţ	۰:		i		:	ł.		:		: ·	` 1		ŀ	
ļ			•			•						•	•										•											
	•																	•		÷			•		·									
									•				•		•			3				•												
	20 0 0 12										•	•		-							•													
	- UC							.,					•										•											
	A.							·		•				۰.						•														
						-						•					•							'				-				•		
													•		-					•														
					•										•	•									-			•.	·					
	ĩ		•							, . !		•			•							•	•		,	•								
			,									:					;				· .				•	٠	I							
	•		:		÷			:				•	1		-		:			i	-			•	•	•								
	1		:				• •	ļ		i		1			4			•	-	;	•						•						•	
	•						!	۰.				:	~		:	۰.	· 1	•		;	•				•		ļ				• .		-	
	:		3				•			÷		:	, 93)		•		•					•	•		•		•							
							٠	;					. [X=60,93)				•	•		·					•		:	•						
		÷	. :			•	•	.•		:	•			•			:	-							•		1							
			• •		•							:	(N L IX)				1					•	•									•		
	•		:		·		. ·			n .		Ξ	42).		•		•					•					ł						1	
	•	1					,					3,79								:							2						÷	
	9 1						•			•	•.	1 = 4 1 K = 4	,							!	·				•									
		СИ L	TT)	Î	-		:				•	TK).					ļ			:		•		: •	•								•	
		Call SU92(N(66),N(67),N(68),ODEPTH)	20 N 1		101	51	69 TO 499 1 25144054 .20. 04854160 TO 352	67 17 17		,	~	N)			:		•			•		•	•	•••			ł							
		(68)	1 N (1	RE A		10 12	10 X	0 10	2		46.1	1					•	•					•	••••	:		:	· ·	•		•		•	
	21/2	N.C.	N (7)	Ē	3	8	000	5.00			1 4 4 0	TKet		, .			•		I		_								-					
	17 148 00/02/72	N (6)	4151,	3	2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 -	BOAT	2 P E 4	8	1 J D D		5			ŝ	EE0					1	•	•		;					-	•				
	81	(99)	(2),		0.0	:	0 6	e 1			H				11 66	-	•		-			:		•								:		
	17	92 (N	55	5	2 4	10 10	م ۳		1 H 4 E A 3	9041 A 06 A		641)	122	234	CONTINUE WRITE(2.101)	10	, '		:					•				•	•			•		
	U	105		i Ri	1 SU3	TO 40	10 1		1302 7 8 8 9	1 = 1 0			N 4 4	7 E C Z	ITTHU	N P T	1			:	. •	:		4			• •		•			·	:	
	N 72	S.	333		L L L	ខ្លង	ទដ	. 5 :	3 F 0 9 Y 0	050		i è i			ខំទី		ž Ž		•	:				l i	;				1					
	5 8		251		252	950	151		352 400			2	12 51	19 19 19 19 19 19 19 19 19 19 19 19 19 1	50	101				, ,		1	•.	į.	. 1		ŀ	_	±		•			
	VERSION JUN 72	•							•		•						1		•					i.	•			-	ļ				:	
		2923	003100	01/2	1203	5020	0120	C213	9320 9320	6220	1220	0732	2 9 2 0 1	10242	1025	59201	1026		:	1				i I			:		ţ				:	
	¥ N ¥	00			00	00		, o	88	8		9 0 9 0		00			55		1	, , 1		Ī					:	-	i	• •				
			. 1	£		•	-	21 - A									'	•		-										. .				

•

.

ı.

.

-

Į		i				•		•		•	•				'
							ι.			ı					-
						• .							•		
.`				,			· ·	•			•				
i.															
			.•		•			·		,					
-												•.			
						•									
			,						•						
				•											
			:										•		
											· ·		•		
		•													
								•		•					
i	•											•			
÷.,															
			÷	-											
;		i i	•					-							
							i	-					-		
							:				•				
÷		•				÷				:					
•	•			•		•	1	•					-		
		-	·										-		
	-		•					-							
	1	2	0		9	œ		<u>o</u> '	5		a	2	2	0	
46100		13600	368 J	56500	3500	7500	0 20 20 20	3005	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5		1900	1000	34070	1520	
	*	н,	t.	in.		-	:		ю ·				M3		
2 Ì.)	· .	`c	2	21)	21)	21)	21)	51)	21)	-	21)	51)	21)	(12	
		23	2	A1											
		0 51)	5 21)	N 0							10				
				170							10				
20)175 2		20) 070 21	20) 075 21	201170 2	20) 060	2 0) (62 2	20) 932 2		20)081 2		201073 2	201076 2	201172	20) 020	
20) 175	-	20) 470	20) 075	170	20) 060	2 0) 062	20) 932	201120	201061		20) 073	20)076	7. 203 172	201 050	
20) 175	-	20) 470	20) 075	170	20) 060	2 0) 062		201120	201061		10	20)076	7. 203 172	201 050	
5 17)15 20)175		17111 201070	17110 201075	170	17)05 20)060	17)19 20)062	220(02 50/17 0	17101 201120	17) 29 20) 061		17:03 20:073	17)35 20)076	17)17 201172	17)03 201050	
1045 17715 201175		20) 470	1020 17110 201075	1034 17117 201170	20) 060) 020 17) 19 20) 062	20) 932	002 11)01 201120	201061		20) 073	20)076	7. 203 172	006 17)03 201050	
5 17)15 20)175		17111 201070	17110 201075	170	17)05 20)060	17)19 20)062	220(02 50/17 0	071 (07 T0 (21 20) (5	17) 29 20) 061		17:03 20:073	17)35 20)076	17)17 201172	17)03 201050	
02 121045 12115 201175		02 15)022 17)11 20)070	20 (02 DT (21 020151 CO	07 15)034 17117 20)170	12 15/010 17/05 20/060	15 15)020 17) 10 2 0)062	17 15;216 17;05 20;032	021102 10121 200151 <u>2</u> 1	04 15)056 17129 20)061	•	05 15)006 17)03 20)073	07 151010 17105 201076	07 151043 17117 201172	00 15)066 17)03 20)050	
3)07 15]045 17)15 20)175		3) 02 15) 022 17) 11 20) 970	2)0102 DT(ZT 02015T E0(E	3) 02 72) 034 711 1 50) 170	12 15/010 17/05 20/060	3)15 15)020 17)10 2 0)062	250(02 50(11 9TC(ST	021102 T0121 20015T LT12	4 15)056 17)29 20)061	• •	05 151006 17103 201073	3107 151010 17105 201076	217102.17127 270157 1012	a 15)0C6 17)03 20)050	-
2 13)07 151045 17)15 20)175		4 13)02 15)022 17)11 20)070	4 13)03 15)020 17)10 20)075	0 13107 151034 17117 201170	3 13)12 15)010 17)05 20)060	3 13)15 15)020 17)10 20)062	3 13)17 15)216 17)09 20)032	021(02 10(21 200(51 21(21 £	+ 13) 04 15) 050 17) 29 20) 081	 - -	4 13)05 15)006 17)03 20)073	4 13/07 15/010 17/05 20/076	2.1102.121(17 E+0(51 10)(125	3 13)06 15)066 17)03 20)050	-
6)5 13)07 151345 11)15 50)175		914 13)02 15)022 17)11 20)070	914 13103 151620 17110 201075	918 13107 151034 17117 201170	9)3 13)12 15)010 17)05 20)060	9)3 13)15 15)020 72)10 2 0)062	250(02 50(11 910(51 11(51 5(6	021(02 T0(2T 200(5T 2T(2T 26)	9)4 13)84 15)850 17)29 28)881		914 13105 151006 17103 201073	910102 211010 17105 201076	21102.11117 240(51 10(51 5(6	9)3 13)06 15)0C6 17)03 20)050	-
50 915 13107 151345 17115 201175		45 914 13102 151022 17111 201070	41 914 13103 151020 17110 201075	99 918 13107 151034 17117 201170	33 13)12 15)010 17)05 20)060	3 13)15 15)020 17)10 20)062	250102 5011 15101 11102 501035	021(02 T0(2T 20)(5T 2T(2T 5(6)4 13)04 15)056 17)29 20)081	- - -	4 13)05 15)006 17)03 20)073	14 13107 151010 17105 201076	21102.11117 200151 10121 216 22	32 9)3 13)06 15)0C6 17)03 20)050	-
0 915 13107 151245 17115 201175		914 13)02 15)022 17)11 20)070	914 13103 151620 17110 201075	918 13107 151034 17117 201170	4 913 13112 151010 17105 201060	9)3 13)15 15)020 72)10 2 0)062	4 9)3 13)17 15)210 17)09 20)032	54 913 13117 100151 11107 516 53	9)4 13)84 15)850 17)29 28)881	•	914 13105 151006 17103 201073	910102 211010 17105 201076	21102.11117 240(51 10(51 5(6	2 913 13108 151066 17103 201050	-
3 8)150 9)5 13)07 15)045 17)15 50)175		5 81141 914 13) 02 15) 022 17) 11 20) 070	5 81145 914 13103 151020 17110 201075	5 81999 910 13107 151034 17117 201170	06 b) 24 9)3 13)12 15)010 17)05 20)060	06 0) 24 913 13115 151020 17110 201062	5 8) 24 9)3 13)17 15/016 17/09 20)032	6 8) 24 913 13111 121005 11101 501150	0 0) 6 9)4 13)04 15)056 17)29 20)061		0 8) 6 914 13105 151006 17103 201073	\$ 8) \$ 9)4 \$3107 \$51010 \$7105 201076	0 21725 6)3 73)01 75)042 11/17 20)715	66 81132 913 13198 151066 17/83 201050	
521(02 ST(21 SHC(S1 20(S1 S(6 021(A E0(005 61141 914 13102 151022 17111 201070 0 1 1	05 81141 914 13103 151020 17110 201075 0 0	05 61999 910 13107 151034 17117 201170 3 0	06 b) 24 9)3 13)12 15)010 17)05 20)060	06 0) 24 913 13115 151020 17110 201062	016 61 24 913 13117 151016 17109 201032 0 0 0	166 81 24 913 13117 151002 17101 201120 2	110 6) 6 9)4 13)04 15)056 17)29 20)061 0 1 1		110 6) 6 9)4 13)05 15)006 17)03 20)073 0	7)16 8) 6 9)4 13/07 15/010 17/05 20/076 60 .	0 0 0 11122 0131 13107 151043 17117 201172	66 81132 913 13198 151066 17/83 201050	
52102 51(21 540(51 20)02 13)02 60 21(0 20)2 6	5330 5410 1100 0000	9 7)05 6)141 9)4 13)02 15)022 17)11 20)070 1500 1600	9 7)05 8)141 9)4 13)03 15)020 17)16 20)075 4500 5360	7 7) 05 61999 9) 0 13) 07 15) 034 17) 17 20) 170 5300 9308	9 7)86 8) 24 9)3 13)12 15)010 17)05 20)860 1008 2538	9 7106 81 24 913 13115 151020 17110 201062 6033 530	9 7)96 61 24 9)3 13)17 15)916 17)89 20)932 4980 1580 138	3 7106 81 24 913 13117 151002 17101 201120 3000	3 7 1 10 6) 6 9)4 13) 04 15) 050 17) 29 20) 061 1200 530		17110 8) 6 914 13105 151006 17103 201073 1000	1360 8) 6 9)4 13/07 15/010 17/05 20/076 1360	9 7)06 61132 9)3 13)07 15)043 17)17 ⁷ 20)172 1009 2003	0 7)56 6)132 9)3 13)06 15)066 17)03 20)050 200 320 1006	
52102 51(21 540(51 20)02 13)02 60 21(0 20)2 6		6109 7105 61145 914 13102 151022 17111 201070 11500 1100 1100	5109 7105 81141 914 13103 151020 17110 201075 4500 14900 25300	61 C7 7) 05 61 999 9) 0 13) 07 15) 034 171 17 20) 170 4500 2500 9300	61097106 81 24 913 13112 151010 17105 201060 1300 2530	6109 7106 81 24 913 13115 151020 1710 201062 6033 1333 530	6/09 7/06 6/ 24 9/3 13/17 15/016 17/09 20/032 400 1500 1500	6109 7106 81 24 913 13117 151002 17101 201120 3000	6)13 7)10 0) 6 9)4 13)04 15)056 17)29 20)061 11200 1335	1000 1000 1000 1000	6)13 7)10 8) 6 9)4 13)05 15)006 17)03 20)073 1000	6113 7116 8) 6 914 13/07 15/010 17/05 20/076 1060	12009 21009 12003	6)00 7)56 6)132 9)3 13)06 15)066 17)03 20)050 200 320 1000	
5/1(02 51(21 5%)(51 20(21 5)(6 07)(8 20(2 6))(6 1	2) 15300 2) 1100 2) 30000	11 6109 7105 61141 914 13102 151022 17111 201070 221 11500 221 1100 221 1100	11 6109 7105 81141 914 13103 151020 17110 201075 221 4500 221 25305 221 25305	1 6167 7)05 81999 9)0 13)07 15)034 17117 201170 22) 45300 22) 2540 22] 9300	1 6)09 7)06 8) 24 9)3 13)12 15)010 17)05 20)060 21 1390 21 2530	61697)66 8) 24 913 13)15 15)020 17)10 20)062 91 6003 21 1333	1 6109 7106 61 24 913 13117 151016 17109 201032 221 4000 221 1500 221 1500 221 1500 221 1000	1 5109 7106 81 24 913 13117 151002 17101 201120 21 3000	JI 6913 7110 6) 6 9)4 13)04 15)056 17)29 20)061 221 11200 221 1500 221 535	2) 4066 2) 3640 21 640	t 5113 7110 81 6 914 13105 151006 17103 201073 21 1000	1 6113 7115 81 6 914 13107 151010 17105 201076 21 1060 .	11 6109 7106 81132 913 13107 151043 17117 201172 221 21009 221 13093	L 6)06 7)66 6)132 9)3 13)06 15)066 17)03 20)050 2) 203 2) 320 2) 1000	
521102 51121 540151 20121 516 02119 ED12 6019	14 22) 15300 15 22) 15300 17 22) 30000	411 6109 7105 61141 914 13102 151022 17111 201070 1 221 11500 7 221 1100 1 221 1100	L1 61 09 71 05 81141 914 131 03 151 020 17140 201 075 221 4500 231 24900	J1 64 67 71 05 84999 910 131 67 151 634 17117 201 170 221 45000 221 2506 221 9300	1 4)1 6)09 7)06 8) 24 9)3 13)12 15)010 17)05 20)060 12 221 1300 41 229 2500	.1 6189 7)86 8) 24 9)3 13)15 15)828 17)19 20)862 22) 6693 23) 1333 22) 530)1 6109 7196 6) 24 913 13117 151916 17109 201032 221 4900 221 1500 221 1500 221 1500 221 1500	JI 6109 7166 81 24 913 13117 151002 17101 201120 221 3000	ÞI 6)13 7)10 0) 6 9)4 13)04 15)050 17)29 20)061 72) 11200 22) 1600 27) 1500	22) 4069 22) 3630 22) 8430	4)1 6)13 7)10 8) 8 9)4 13)05 15)006 17)03 20)073 2 22) 1000	4)1 6)13 7)10 8) 6 9)4 13/07 15/010 17/05 20/076 2 22) 1360	3 41 6109 7106 61132 913 13107 151043 17117 201172 67 221 21009 54 221 12093	1 4)1 6)06 7)06 6)132 9)3 13)06 15)006 17)03 20)050 17 22) 200 13 22 320 14 22 1000	
521102 51(21 540(51 20)21 5(6 02)(9 E0(2 60)9 I(4 75/	14 22) 15300 15 22) 15300 17 22) 30000	53 411 6109 7105 61145 914 13102 151022 17111 201070 241 221 11500 207 221 1100 251 221 1100	53 411 5109 7105 81141 914 13103 151020 17110 201075 264 221 4500 241 221 14900 251 271 25300	J1 64 67 71 05 84999 910 131 67 151 634 17117 201 170 221 45000 221 2506 221 9300	1 4)1 6)09 7)06 8) 24 9)3 13)12 15)010 17)05 20)060 12 221 1300 41 229 2500	52 411 6109 7106 81 24 913 13115 151020 1710 201062 251 221 6003 241 221 1330 222 221 530)1 6109 7196 6) 24 913 13117 151916 17109 201032 221 4900 221 1500 221 1500 221 1500 221 1500	JI 6109 7166 81 24 913 13117 151002 17101 201120 221 3000	ÞI 6)13 7)10 0) 6 9)4 13)04 15)050 17)29 20)061 72) 11200 22) 1600 27) 1500	22) 4069 22) 3630 22) 8430	4)1 6)13 7)10 8) 8 9)4 13)05 15)006 17)03 20)073 2 22) 1000	153 411 6115 7115 8) 6 914 13/07 15/010 17/05 20/076 1552 22/ 1360	/53 4)1 6)09 7)06 8)132 9)3 13)07 15)043 17)17 20)172 1207 22) 21009 1254 22) 13093	1 4)1 6)06 7)06 6)132 9)3 13)06 15)006 17)03 20)050 17 22) 200 13 22 320 14 22 1000	
521(02 51(21 540(51 20(21 5(6 02)(0 10) 60(9 1(4 15/11)	11/254 227 15000 11/265 221 15000 11/205 221 1100	13/53 4)1 6)09 7)05 8)141 9)4 13)02 15)022 17)11 20)070 11)241 22) 11500 11)207 22) 1100 11)207 22) 1100	11/53 411 5109 7105 81141 914 13103 151020 17110 201075 111241 221 14900 111241 221 14900	14/53 4)1 6:67 7)05 8:999 9)0 13:07 15:034 17:17 20:170 11:207 22) 45000 11:205 22) 2500 11:254 22) 9000	15/51 411 6109 7106 81 24 913 13112 151010 17105 201060 111202 221 1300 111241 221 2530	(15/57 4)1 6)09 7)06 0) 24 9)3 13)15 15)020 17)10 20)062 11)241 22) 603 11)241 22) 1303 11)242 22) 500	15/57 4)1 6)09 7)06 8) 24 9)3 13)17 15)016 17)09 20)032 111251 221 4000 11126 221 1500 11286 221 1500 11287 221 1000	1/23 4)1 6)09 7)06 8) 24 9)3 13)17 15)002 17)01 20)120	\$4753 4)1 6)13 7)10 6) 6 9)4 13)04 15)056 17)29 20)081 11)253 22) 11200 11)267 22) 1500	11) 252 22) 4000 11) 21 22) 4000 11) 21 22) 4000	16/53 411 6113 7110 81 8 914 13105 151006 17103 201073 111252 221 1000	18/53 4/1 6/12 7/15 8) 6 9)4 13/07 15/010 17/05 20/076 11/252 22) 1360	7/53 411 6109 7106 61132 913 13107 151043 17117 201172 111207 221 21009 	[7/53 4]1 6)06 7)06 6)132 9)3 13)06 15)006 17)03 20)050 11207 22) 200 11231 22) 320 11241 22) 320	
521(02 51(21 540(51 20(21 5(6 02)(0 10) 60(9 1(4 15/11)	11/254 227 15000 11/265 221 15000 11/205 221 1100	13/53 4)1 6)09 7)05 8)141 9)4 13)02 15)022 17)11 20)070 11)241 22) 11500 11)207 22) 1100 11)207 22) 1100	11/53 411 5109 7105 81141 914 13103 151020 17110 201075 111241 221 14900 111241 221 14900	14/53 4)1 6:67 7)05 8:999 9)0 13:07 15:034 17:17 20:170 11:207 22) 45000 11:205 22) 2500 11:254 22) 9000	15/51 411 6109 7106 81 24 913 13112 151010 17105 201060 111202 221 1300 111241 221 2530	(15/57 4)1 6)09 7)06 0) 24 9)3 13)15 15)020 17)10 20)062 11)241 22) 603 11)241 22) 1303 11)242 22) 500	4/15/57 4)1 6109 7106 6) 24 9)3 13117 151016 17109 201032 111251 221 4000 111265 221 1500 111207 221 1500 111207 221 1000	VJ 111254 221 3000 ND	4/15/53 4)1 6)13 7)10 6) 6 9)4 13)04 15)056 17)29 20)081 11/253 22) 11200 11/267 22) 1500 11/221 22) 150	111252221 4069 11124122) 3600 111231221 640	16/53 411 6113 7110 81 8 914 13105 151006 17103 201073 111252 221 1000	18/53 4/1 6/12 7/15 8) 6 9)4 13/07 15/010 17/05 20/076 11/252 22) 1360	7/53 411 6109 7106 61132 913 13107 151043 17117 201172 111207 221 21009 	[7/53 4]1 6)06 7)06 6)132 9)3 13)06 15)006 17)03 20)050 11207 22) 200 11231 22) 320 11241 22) 320	
521(02 51(21 540(51 20(21 5(6 02)(0 10) 60(9 1(4 15/11)	11/254 227 15000 11/265 221 15000 11/205 221 1100	13/53 4)1 6)09 7)05 8)141 9)4 13)02 15)022 17)11 20)070 11)241 22) 11500 11)207 22) 1100 11)207 22) 1100	11/53 411 5109 7105 81141 914 13103 151020 17110 201075 111241 221 14900 111241 221 14900	14/53 4)1 6:67 7)05 8:999 9)0 13:07 15:034 17:17 20:170 11:207 22) 45000 11:205 22) 2500 11:254 22) 9000	15/51 411 6109 7106 81 24 913 13112 151010 17105 201060 111202 221 1300 111241 221 2530	(15/57 4)1 6)09 7)06 0) 24 9)3 13)15 15)020 17)10 20)062 11)241 22) 603 11)241 22) 1303 11)242 22) 500	15/57 4)1 6)09 7)06 8) 24 9)3 13)17 15)016 17)09 20)032 111251 221 4000 11126 221 1500 11286 221 1500 11287 221 1000	VJ 111254 221 3000 ND	4/15/53 4)1 6)13 7)10 6) 6 9)4 13)04 15)056 17)29 20)081 11/253 22) 11200 11/267 22) 1500 11/221 22) 150	11) 252 22) 4000 11) 21 22) 4000 11) 21 22) 4000	16/53 411 6113 7110 81 8 914 13105 151006 17103 201073 111252 221 1000	18/53 4/1 6/12 7/15 8) 6 9)4 13/07 15/010 17/05 20/076 11/252 22) 1360	7/53 411 6109 7106 61132 913 13107 151043 17117 201172 111207 221 21009 	1 4)1 6)06 7)06 6)132 9)3 13)06 15)006 17)03 20)050 17 22) 200 13 22 320 14 22 1000	
521(02 51(21 540(51 20(21 5(6 02)(0 10) 60(9 1(4 15/11)	11/254 227 15000 11/265 221 15000 11/205 221 1100	13/53 4)1 6)09 7)05 8)141 9)4 13)02 15)022 17)11 20)070 11)241 22) 11500 11)207 22) 1100 11)207 22) 1100	13/53 411 6109 7105 81141 914 13103 151020 17110 201075 111249 221 4500 111241 221 14900	14/53 4)1 6:67 7)05 8:999 9)0 13:07 15:034 17:17 20:170 11:207 22) 45000 11:205 22) 2500 11:254 22) 9000	15/51 411 6109 7106 81 24 913 13112 151010 17105 201060 111202 221 1300 111241 221 2530	(15/57 4)1 6)09 7)06 0) 24 9)3 13)15 15)020 17)10 20)062 11)241 22) 603 11)241 22) 1303 11)242 22) 500	74-412/57/57 401 604 7106 60 24 903 13017 150016 1704 200032 00 111251 220 4000 01 111265 221 1500 01 110215 221 1500 01 110215 221 1000	VJ 111254 221 3000 ND	1164/16/53 411 6113 7110 0) 6 9)4 13)04 15)056 17)29 20)081 201 11/253 221 11200 301 11/227 221 1600 301 11/221 221 330 338	0) 11)25222) 4069 0) 11)24122) 3609 0) 11)23122) 640)2/10 104/16/53 4/1 6/12 7/16 8) 6 9)4 13/07 15/010 17/05 20/076 200 11/252 22) 1360	104/17/53 411 6109 7106 61132 913 13107 151043 17117 201172 61 111264 221 12009 61 111254 221 12003	7743	

į ٠i Ì We with the second CENERACI OF BARA BALAN JA Appendix F.

:

	<pre>CFOUNDETEY 1953N: TEVIETERILATERE NUHAEO TEVIETERILATERE NUHAEO TEVIETERILATERE NUHAEO TEVIETERILATERE NUHAEO TEVIETERILAL EVERTUNES NITATISTICAL EVERTUNES NITATISTICAL EVERTUNES NITATISTICAL EVERTUNES NITATISTICAL EVERTES (UNEGER NUHAEO) OSTINIS CHORTENING SAUNASEN NUTATER NUHAEO CHORTENING SAUNASEN CHORTENING SAUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN CONTRESS OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN CONTRESS OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN CONTRESS OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN OF VALUES SAUNASEN NUNASEN NUNASEN OF VALUES SAUNASEN NUNASE</pre>		
<pre>prove the provement and the provement pro</pre>	EV 31E SIG GRUUPTEV 1954MI EV 31E SIG GRUUPTEV 1954MI 10011 00 FUTEVITATIONED 10011 10 FUTEVITATIONED <td></td> <td></td>		
<pre>Maint of Infinition in an end of the infinition of in</pre>	<pre>IDDIE OF NETFORMER WINGS IDDIE OF NETFORM</pre>		
<pre>mont of transmission works mont of transmission wore</pre>	<pre>ADD-FILE Day Tecture NUMPERD TO TITLE Day Tecture Tecture NUMPERD TO TITLE Day Tecture Tecture NUMPERD TO TITLE DAY TECTURE TECTURE TECTURE TO TITLE DAY TETTURE TECTURE TO TITLE DAY TETTURE TECTURE TO TITLE TETTURE TETTURE TO TITLE TETTURE T</pre>		. : !
<pre>May any any any any any any any any any a</pre>	<pre>prove destruction winded prove destructio</pre>		
10.3111 AUTURET STREET AURALIA	<pre>point curstication investo privation for any and any and any and any any and any any and any any and any and any any any and any any and any any and any any and any any any any any any any any any any</pre>		
MUSIC FERSA UNDER ELENTER AUTOR MUSIC SETS TELETER AUTOR MUSIC AUTOR A	1) 100000000000000000000000000000000000		
<pre>site in the interval in the interval interval in the interval interval in the interval interval in the interval int</pre>	1111000000000000000000000000000000000		1
<pre>TINUARS = TANTERER MUNER INTERPORT = TANTERER MUNER INTERPORT = TANTERER MUNER INTERPORT = TANTERE MUNER INTERPORT =</pre>	2111 21111 21111 21111		
<pre>control argin argin</pre>	<pre>2017FFFL GETCH OF LL SETTIGTER WHYER) 2022FFFL GETCH STITTSTSSFEER AUNSER IN 30) 2020FFFL GATCH STITTSTSSFEER AUNSER IN 30) 2020FFL GATCH STITTSTSSFEER AUNSER AUN</pre>		•
<pre>113:FISTSTSTSTSTSTSTSTSTSTSTSTSTSTSTSTSTSTST</pre>	<pre>1111_11_11_11_11_11_11_11_11_11_11_11_1</pre>	-	
<pre>2011.2014_STGD 90-14106 NUMPER IN 30. 2011.2014_STGD 90-14106 NUMPER IN 30. 2012.2014 2014.2014 2014.2014.2014.2014.2014 2014.2014.2014.2014.2014.2014 2014.2014.2014.2014.2014.2014.2014.2014.</pre>	<pre>2010TIL SICH 9F "41,00% SPECIESTIVEGER NUMBER IN 30) 2010TIL SICH 9F "41,00% SPECIESTIVEGER NUMBER IN 30) 2010TIL SICH 9F "41,00% SPECIESTIVEGER NUMBER IN 30) 2010TIL SOUTH IE CHECKPOINTS OCCURT 2010TIL SECTIONS PEROPE EOF RELIECTIONS * 0 ALL LECALITY IS CHECKOF FOR THIS LODDER CAL NUMBER OF VALUES FORTER * 1330 MUMPER OF VALUES FOR THIS ELEVENT * 1445 TOTAL WOMPER OF</pre>	· · · · · · · · · · · · · · · · · · ·	
ZIJICIAL GIATON DE FAJOR SEGIESTIVIEER MUTAN MUT LIJIET SOBSUI LIJIET SOBSUI L	22:10141 CATCH OF ALLOR SPECIES(TYTEGER NUMBER IN JUN- 1015 FILE 10 1014 FILE 10 1015 FILE 10 1014 ONE ENTY TEAHINATOR REDOFE EOF REJECTIONS * 0 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.39. 21.24.30. 21.24.30. 21.24.50.07 FUTES ACTERTS * 10. 21.24.50.07 FUTES ACTERTS * 10. 21.24.50.07 ALLES ACTERTS * 11.30 21.24.50.07 ALLES ACTERTS * 11.30 21.24.50.07 ALLES ACTERTS * 11.30 21.24.50.07 ALLES ACTERTS * 27.40 21.24.50.00 FUTES CONFERTS * 27.40 21.24.50.00 FUTES FUTES * 27.40 21.24.50.00 FUTES * 27.40 21.24.50.00 FUTES * 27.40 21.24.50.00 FUT	•	1
MAY DOK ENTAY TENTIANTUR METODE EUR MAY DOK ENTAY TENTIANTUR METODE EUR ALL LEGALITY TS CACTAT DA THES LEGENT ALL LEGALITY TS CACTAT DA THES LAN ALL LEGALITY THE CACTAT DA THES LAN ALL LEGALITY TO THE CACTAT DA THES LAN ALL LEGALITY TO THE LAN THE	CLUEP CLUEP SSUE RECORT WHY IC PRECEDENTS OCCURS SSUE RECORT WHY IC PRECEDENTS SSUE RECORT WHY IC PRECEDENTS SSUE RECORT WHY IC PRECEDENTS RIVY ONE EWAY TERVILIY IS CHETYCO REUSCITIONS SUG2272 RULUES RIVY ONE EWAY TERVILIY IS CHETYCO REUSCITIONS RIVY ONE EWAY FERTILIES 643 RIVY SECTIONS REUSCITIONS RIVY SECTIONS REUSCINCS RIVY SECTIONS REUSCINCSE	-	
<pre>6351WI WAGY ALL CHECKPOINTS OCCURT # JERHINATCR REFORE EOF # JERHINATCR REFORE EOF #UNASER OF VALUES ACTERTS = #UNASER OF VALUES FOR THIS ELEM #ERCL CONFLETED FOR ELEMENT OF #ERCL FOR FOR FOR FOR ELEMENT OF #ERCL FOR FOR FOR FOR FOR ELEMENT OF #ERCL FOR FOR FOR FOR FOR ELEMENT OF #ERCL FOR FOR FOR FOR FOR FOR ELEMENT OF #ERCL FOR FOR FOR FOR FOR FOR ELEMENT OF FOR FOR FOR FOR FOR FOR FOR FOR FOR</pre>	<pre>6351M: WAGY ALL CHECKPOINTS OCCUR: Y IERHINATOR REFORE EOF INUTOR OF VALUES FOR THIS LO IEGALITY IS CHECKFO FOR THIS LO NUTORE OF VALUES ACTERTS = NUTOR OF VALUES EATTH REJECTS NUTORE OF VALUES FOR THIS ELEMI OF UNITURE VALUES FOR THIS ELEMI AFEACE CONFLETED FOR ELEMIENT C OF UNITURE VALUES FOR THIS ELEMIE OF UNITURE VALUES FOR THIS ELEMIENT OF UNITURE FOR FOR THIS ELEMIENT OF UNITURE VALUES FOR THIS ELEMIENT OF UNITURE FOR FOR THIS FOR THIS ELEMIENT OF UNITURE FOR FOR FOR THIS ELEMIENT OF UNITURE FOR FOR THIS ELEMIENT OF UNITURE FOR FOR THIS ELEMIENT OF UNITURE FOR FOR THIS FOR THIS ELEMIENT OF UNITURE FOR FOR THIS FOR THIS ELEMIENT OF UNITURE FOR FOR THIS FOR THIS FOR THIS FOR THIS FOR THIS FOR THIS FOR TH</pre>		•
WHEN ALL CHECKPOINTS OCCURT T TERMINATOR BEFORE EOF TEGALITY IS CHECKPIARE EAF HUNARE OF VALUES ACCEPTER HUNARE OF VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEROE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM	<pre>WAEN ALL CHECKPOINTS OCCURT Y JERMINATCR REFORE EOF LEGALITY IS CHECKFO FOR THIS LO NUMBER OF VALUES ACCEPTED E NUMBER OF VALUES ACCEPTED E VALUES FOR FLEWER C OF UNITUE FOR FLE ELENIE OF UNITUE FOR FLE FLE ELENIE OF UNITUE FOR FOR FOR FLE ELENIE OF UNITUE FOR FOR FOR FLE ELENIE OF UNITUE FOR FOR FLE ELENIE OF UNITUE FOR FOR FLE FOR FLE FOR FLE FOR FLE FLENIE OF UNITUE FOR FOR FOR FLE FOR FLE FLENIE OF UNITUE FOR FOR FLE FOR FLE FLENIE OF UNITUE FOR FOR FLE FLENIE OF UN</pre>		
<pre>Y JERMINATCR REFORE EOF LEGALITY IS CHEOKEN FOR THIS LO HUNMARG OF VALUES ACCEPTED HUNMARG OF VALUES ACTERTED HUNMARG OF VALUES ACTERTED HUNMARG OF VALUES ACTERTED HUNMARG OF VALUES EXCLUDED HUNMARG OF VALUES EXCLUDED HUNMARG OF VALUES EXCLUDED HUNMARG OF VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HEAGE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEMENT C OF UNTOUE VALUES FOR FOR THIS ELEMENT C OF UNTOUE VALUES FOR FOR FOR THIS ELEMENT C OF UNTOUE VALUES FOR FOR FOR FOR FOR THIS ELEMENT C OF UNTOUE VALUES FOR FOR FOR FOR FOR FOR FOR FOR FOR FOR</pre>	<pre>Y JERMINATCR REFORE EOF LEGALITY IS CHECKED FOR THIS LO HUMPRER OF VALUES ACCEPTED # HUMPRER OF VALUES ECTEPTED # HUMPRER OF VALUES ECTEPTED # HUMPRER OF VALUES EVENLUED D# HUMPRER OF VALUES EVENLUED DATA SETT HUMPRER OF VALUES FOR THIS ELEMI HERE COMPLETED FOR ELEMIT C HERE C</pre>		
<pre>Y TERMINICR EEFORE EOF LEGALITY IS CHEOKEN FOR THIS LO NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES EXCLUSED = NUMMER OF VALUES FOR THIS ELEM HERDE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HERDE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM HERDE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM THIS ELEM THIS ELEMENT COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEMENT C OF UNTOUE VALUES FOR ELEMENT C OF UNTOU</pre>	<pre>Y IERHINITOR EEFORE EOF LEGALITY IS CHECKED FOR THIS LO LEGALITY IS CHECKED FOR THIS LO NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES REALURED = NUMMER OF VALUES REALUSED = NUMMER OF VALUES REALUSED = NUMMER OF VALUES FOR LOSE = NUMMER OF VALUES FOR THIS ELEM MERCE COMPLETED FOR FLEMER C OF UNTOUF VALUES FOR THIS ELEM OF UNTOUE VALUES FOR THIS ELEM OF UNTOUE VALUES FOR THIS ELEM OF UNTOUE VALUES FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM THERE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM THERE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM THERE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEM THERE COMPLETED FOR ELEMENT C OF UNTOUE VALUES FOR THIS ELEMENT C OF UNTOUE VALUES FOR THIS ELEMENT C OF UNTOUE VALUES FOR THIS ELEMENT C OF UNTOUE VALUES FOR THIS FOR THI</pre>		
LEGALITY IS CHEOKEN FOR THIS LO RUNARE OF VALUES ACCEPTED = NUMARE OF VALUES FOR ELEMENT OF NUMARE OF VALUES FOR THIS ELEMENT OF NUMARE OF VALUES FOR ELEMENT OF MERCE COMPLETED FOR ELEMENT OF MERCE COMP	LEGALITY IS CHECKED FOR THIS LO LEGALITY IS CHECKED FOR THIS LO NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES ACCEPTED = NUMMER OF VALUES REJECTED = NUMMER OF VALUES REJECTED = NUMMER OF VALUES REVELUED DATA SET HILMER OF VALUES FOR THIS ELEMENT C OF UNITUDE VALUES FOR THIS ELEMENT C		
LEGALITY IS CHECKEN FOR THIS LO LEGALITY IS CHECKEN FOR THIS LO RUMARER OF VALUES ACCEPTED = NUMARE OF VALUES ACCEPTED = NUMARE OF VALUES ACCEPTED = NUMARE OF VALUES EACLUDED = HEACE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR ELEMENT C OF UNTOUR VALUES FOR ELEMENT C AFERE COMPLETED FOR ELEMENT C AFERE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM VERCE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR ELEMENT C OF UNTOUR VALUES FOR ELEMENT C OF UNTOUR VALUES FOR ELEMENT C	LEGALITY IS CHECKED FOR THIS LO LEGALITY IS CHECKED FOR THIS LO RUNATER OF VALUES ACCEPTED = NUNATER OF VALUES ACCEPTED = NUNATER OF VALUES ACCEPTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES RETERING MERCE COMPLETED FOR ELEMENT C OF UNITUDE VALUES FOR THIS ELEME NUTUUE VALUES FOR THIS ELEME OF UNITUDE VALUES FOR THIS ELEME NUNATER OF FOR THIS ELEMENT C OF UNITUDE VALUES FOR THIS ELEMENT C OF UNITUDE VALUES FOR THIS ELEMENT C OF UNITUDE VALUES FOR THIS ELEMENT C		1
LEGALITY IS CHECKED FOR THIS LO LEGALITY IS CHECKED FOR THIS LO MUNARER OF VALUES ACCERTED = MUNARER OF VALUES ACCERTED = MUNARER OF VALUES REJECTED = NUCARER OF VALUES FOR THIS ELEM MERCE COMPLETED FOR ELEMENT O PERCE COMPLETED FOR ELEMENT O MERCE COMPLETED FOR ELEMENT O MER	LEGALITY IS CHECKED FOR THIS LO LEGALITY IS CHECKED FOR THIS LO MUNASER OF VALUES ACCEPTED = MUNASER OF VALUES ACCEPTED = MUNASER OF VALUES ACCEPTED = MUNASER OF VALUES REJECTED = NUNASER OF VALUES REJECTED = NUNASER OF VALUES REJECTED = NUNASER OF VALUES REST ACTEPTED HISTOR OF VALUES FOR THIS ELEMENT C OF UNITOUE VALUES FOR THIS ELEMENT C		
LEGALITY IS CHECKED FOR THIS LO RUNATER OF VALUES ACCEPTED = NUNATER OF VALUES ACCEPTED = NUNATER OF VALUES ACCEPTED = NUNATER OF VALUES ACCEPTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES REJECTED = NUNATER OF VALUES FOR THIS ELEM NUNATER OF VALUES FOR THIS ELEM NUNATER CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM	LEGALITY IS CHECKED FOR THIS LO HUNNER OF VALUES ACCEPTED # HUNNER OF VALUES ACCEPTED # HUNNER OF VALUES ACCEPTED # HUNNER OF VALUES ACCEPTED # HUNNER OF VALUES RELETED # HUNNER OF VALUES RELETED # HUNNER OF VALUES RELETED # HUNNER OF VALUES RELETED # HUNNER OF VALUES FOR FLEVENF C OF UNIOUS FOR THIS ELEM		
LEGALITY IS CHECKED FARLE FYTRIS - HUHAFR OF VALUES ACTERTET = HUHAFR OF VALUES ACTERTET = HUHAFR OF VALUES REJECTED = NUMARE OF VALUES FOR THIS ELEM HERGE COMFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM HERGE COMFLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM HERGE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM VERTE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM VERTE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM VERTE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM VERTE COMPLETED FOR ELEMENT C OF UNTOUR VALUES FOR THIS ELEM	LEGALITY IS CHECKED FOR THIS LO RUNASER OF ALCEPTARE ENTRIES = NUNASER OF VALUES ACCEPTED = NUNASER OF VALUES ACCEPTED = NUNASER OF VALUES REJECTED = NUNASER OF VALUES FOR THIS ELEMENT OF UNITUDE VALUES FOR THIS FOR THIS ELEMENT OF UNITUDE VALUES FOR THIS FOR THIS ELEMENT OF UNITUDE VALUES FOR THIS FOR THI		
HUNGER OF VALUES ACTERTS ATTRE FYTRIS HUNGER OF VALUES ACTERTS F HUNGER OF VALUES REJECTED = NUCASE OF VALUES REJECTED = NUCASE OF VALUES REJECTED = NUCASE OF VALUES REJUDED = NUCASE OF VALUES REALUDED DATA SET HUNGER CONFLETED FOR FLEVENT C OF UNTOUE VALUES FOR THIS ELEM HERE CONFLETED FOR FLEVENT C OF UNTOUE VALUES FOR THIS ELEM HERE CONFLETED FOR FLEVENT C OF UNTOUE VALUES FOR THIS ELEM HERE CONFLETED FOR ELEVENT C OF UNTOUE VALUES FOR THIS ELEM HERE CONFLETED FOR ELEVENT C OF UNTOUE VALUES FOR THIS ELEM NERGE CONFLETED FOR ELEVENT C OF UNTOUE VALUES FOR THIS ELEM HERE CONFLETED FOR ELEVENT C OF UNTOUE VALUES FOR THIS ELEM VERGE CONFLETED FOR ELEVENT C OF UNTOUE VALUES FOR THIS ELEM	HUNNER OF ACCEPTANT, THE FUTELS - HUNNER OF VALUES ACCEPTED = HUNNER OF VALUES ACCEPTED = HUNNER OF VALUES FUTUDED = HUNNER OF VALUES FUTUDED = HUNNER OF VALUES FUTUDED = HUNNER OF VALUES FUTUDED = HUNNER OF VALUES FOR HIS FLEN HERE CONFLETED FOR FLENFT OF UNTUDE VALUES FOR THIS FLENT OF UNTUDE VALUES FOR THIS FLENT		•
NUMMER OF VALUES ACTERTS # NUMMER OF VALUES ACTERTS # NUMMER OF VALUES REJECT NUMMER OF VALUES REJECT NUMMER OF VALUES REJECT NUMMER OF VALUES RETUDED # NUMMER OF VALUES RETUDED # NUMMER OF VALUES REVILUED # MUMMER OF VALUES FOR THIS ELEMENT OF UNTOUF VALUES FOR THIS ELEMENT AFERE COMPLETED FOR ELEMENT C OF UNTOUF VALUES FOR THIS ELEMENT OF UNTOUF VALUES FOR THIS ELEMENT OF UNTOUF VALUES FOR ELEMENT C OF UNTOUE VALUES FOR ELEMENT C	NUMMER OF VALUES ACCEPTED # NUMMER OF VALUES ALTH REJECT NUMMER OF VALUES REJECT NUMMER OF VALUES REVILUED # NUMMER OF VALUES REVILUED # NUMMER OF VALUES FOR TATA SET NUMMER OF VALUES FOR THIS FLEM MERCE COMPLETED FOR FLEMENT C OF UNTOUR VALUES FOR THIS FLEM VALUES FOR THIS FLEM OF UNTOUR VALUES FOR THIS FLEM VALUES FOR THIS FLEM OF UNTOUR VALUES FOR THIS FLEM OF UNTOUR VALUES FOR THIS FLEM COP UNTOUR VALUES FOR THIS FLEM THERE COMPLETED FOR FLEMENT C OF UNTOUR VALUES FOR THIS FLEM	r	-
MUNARY OF ENTERS NITH ASSUCTED = NUNARY OF VALUES EXJECTED = NUNARY OF VALUES EXJLUED = NUNARY OF VALUES EXJLUED = NUNARY OF VALUES EXTLUED = NUNARY OF VALUES FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C HERE CONFLETED FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C OF UNTOUE VALUES FOR FILSENT C HERE CONFLETED FOR FILSENT C HERE CONFLETED FOR FILSENT C HERE CONFLETED FOR FILSENT C HERE CONFLETED FOR FILSENT C	MUNARY OF ENTERS WITH KAUGU MUNARY OF VALUES REJECTED = NUNARY OF VALUES REJECTED = NUNARY OF VALUES REJECTED = NUNARY OF VALUES FOR THIS ELEMENT C MERCE COMPLETED FOR FLEMENT C		
NUTHING OF VALUES EVALUED DATA SETS ACTERTS NUTHING OF VALUES EVALUED DATA SETS ACTERTS NUTHING OF VALUES EVALUED DATA SETS A NUTHING VALUES FOR THIS ELEMENT C OF UNITOUE VALUES FOR THIS ELEMENT C OF UNITOUE VALUES FOR THIS ELEMENT C AFEGE COMPLETED FOR THIS ELEMENT	NUMMER OF VALUES EVALUED DE MUNALER OF VALUES EVALUED DATA SETS ACTEPTED * NUMMER OF VALUES EVALUED DATA SETS AN UNMERCE COMFLETED FOR FLEMENT OF MISSEE COMFLETED FOR FLEMENT OF UNTOUF VALUES FOR THIS ELEMENT OF VALUES FOR THIS		
NUTASE OF DATA SETS ACCEPTED * NUTASE OF YON-VALUED DATA SETS * NUTASE COMFLETED FOR THIS FLEMENT C MERCE COMPLETED FOR THIS FLEMENT C	NULASE DE DATA SETS ACCEPTED * NULASE DE VALUES DATA SETS A MERCE COMFLETED FOR ELEMENT C DF UNICUE VALUES FOR THIS ELEMENT C MERCE COMFLETED FOR ELEMENT C MERCE COMFLETED FOR ELEMENT C MERCE COMFLETED FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C DF UNICUE VALUES FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C		•
NUTHER OF YON-VALUED DATA SETS A MERCE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR THIS ELEMENT C MERCE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR THIS ELEMENT C	NUTASE OF YON-VALUED DATA SETS A MICAGE COMFLETED FOR ELEMENT C of UNITALE VALUES FOR THIS FLEMENT C MERCE COMFLETED FOR ELEMENT C MERCE COMFLETED FOR ELEMENT C AFEACE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C		
PERSE COMPLETED FOR ELEMENT C PERSE COMPLETED FOR THIS ELEMENT C PERSE COMPLETED FOR ELEMENT C PERSE COMPLETED FOR ELEMENT C PERSE COMPLETED FOR THIS ELEMENT C PERSE COMPLETED FOR ELEMENT C	HERGE COMPLETED FOR ELEMENT C OF UNITULE VALUES FOR THIS ELEMENT C HERGE COMPLETED FOR ELEMENT C MERGE COMPLETED FOR ELEMENT C MERGE COMPLETED FOR THIS ELEMENT C DF UNITULE VALUES FOR THIS ELEMENT C HERGE COMPLETED FOR THIS ELEMENT C HERGE COMPLETED FOR THIS ELEMENT C		
<pre>dF UNITALE VALUES FOG THIS ELEMENT HERE COMPLETED FOG THIS ELEMENT HERE COMPLETED FOR THIS ELEMENT HERE COMPLETED FOR THIS ELEMENT HERE COMPLETED FOR THIS ELEMENT PREACE COMPLETED FOR THIS ELEMENT WEAGE COMPLETED FOR THIS ELEMENT OF UNITUDE VALUES FOR THIS ELEMENT OF UNITUDE VALUES FOR THIS ELEMENT HERE COMPLETED FOR ELEMENT C MERE COMPLETED FOR ELEMENT C M</pre>	<pre>dF UNITALE VALUES FOR THIS ELEMENT HERE COMPLETED FOR ELEMENT C HERE COMPLETED FOR ELEMENT C WERE COMPLETED FOR ELEMENT C WERE COMPLETED FOR ELEMENT C HERE COMPLETED FOR ELEMENT C UNITALE VALUES FOR ELEMENT C WERE COMPLETED FOR CLEMENT C WERE COMPLETED FOR CLEMENT C</pre>		
YETCE COVELETED FOR ELEVENT C OF UNITOUE VALUES FOR THIS ELEVENT WERCE CONFLICT FOR THIS ELEMENT HERCE CONFLICTED FOR LISTENT C DF UNITOUE VALUES FOR THIS ELEMENT WERCE COMPLETED FOR ELEVENT C WERCE COMPLETED FOR ELEVENT C WERCE COMPLETED FOR ELEVENT C WERCE COMPLETED FOR ELEVENT C MERCE COMPLETED FOR THIS ELEMENT	HERE COMPLETED FOR ELEMENT C OF UNTOUF VALUES FOR THIS ELEMENT WERE CONFLITED FOR ELEMENT OF UNIDUE VALUES FOR THIS ELEMENT HERE COMPLETED FOR ELEMENT C UNIDUE VALUES FOR THIS ELEMENT C		;
HERE COMPLETED FOR ELEVENT C HERE COMPLETED FOR ELEVENT C HERE COMPLETED FOR ELEMENT HERE COMPLETED FOR ELEMENT C HERE COMPLETED FOR ELEMENT C WERE COMPLETED FOR ELEMENT C HERE COMPLETED FOR ELEMENT C	HEAGE COMPLETED FOR ELEMENT OF UNTYOUF VALUES FOR THIS ELEMENT WERE THIS ELEMENT OF UNITYOE VALUES FOR ELEMENT C HEAGE COMPLETED FOR ELEMENT C WARNE FOR A THIS ELEMENT C WARNE FOR A THIS ELEMENT C		
WERGE CONFLETED FOR ELEMENT OF THIS ELEMENT HERGE CONFLETED FOR THIS ELEMENT O HERGE COMPLETED FOR ELEMENT O HERGE COMPLETED FOR THIS ELEMENT O MERGE COMPLETED FOR THIS FLEMENT O MERGE COMPLETED FOR THIS FLEMENT O MERGE COMPLETED FOR THIS ELEMENT O	45362 COMPLETED 503 ELEMENT 0 45362 COMPLETED 503 THIS ELEMENT HERE 2019LETED 503 THIS ELEMENT 0 02 UN19UE VALUES 503 CIEMENT 0		
HERGE CONFLETED FOR ELEVENT C HERGE CONFLETED FOR ELEVENT C HERGE CONFLETED FOR ELEVENT C MERGE CONFLETED FOR ELEVENT C	42362 CONFLATED FOR ELEMENT 0 04 UNIOUE VALUES FOR ELEMENT C MERCE COMPLETED FOR ELEMENT C 04 UNIOUE VALUES FOR FUES ELEMENT C		
<pre>gF UNTUGE VALUES FOR ELEMENT C HERAE COMPLETED FOR ELEMENT C of UNTUGE VALUES FOR ELEMENT C of UNTUGE VALUES FOR ELEMENT C of UNTUDE VALUES FOR ELEMENT C ALEAE COMPLETED FOR ELEMENT C OF UNTUUE VALUES FOR THIS FLEMENT C OF UNTUUE VALUES FLE</pre>	DE UNEIUE VALUES FUN ALLE CLEAR C HEREE CONSLETED FOR ELEMENT C DE UNEOUE VALUES FOR FUES ELEMENT DE UNEOUE ATEL EDY ELEMENT D		ł
HERE COMPLETED FOR ELEMENT C OF UNITUE VALUES FOR THIS ELEMENT WERE FOMPLETED FOR THIS ELEMENT OF UNITUE VALUES FOR THIS ELEMENT OF UNITUE VALUES FOR THIS ELEMENT OF UNITUE VALUES FOR ELEMENT C OF UNITUE VALUES FOR ELEMENT C ALEMENT COMPLETED FOR ELEMENT C OF UNITUE VALUES FOR ELEMENT C	HERAE COMPLETED FOR ELEVENT C OF UNTOUS VALUES FOR THIS ELEMENT UNTOUS VALUES FOR SIGNET O		
DF UNITUE VALUES FOR THIS ELEMENT WERGE COMPLETED FOR THIS ELEMENT OF UNITUE VALUES FOR THIS ELEMENT OF UNITUE VALUES FOR THIS ELEMENT OF UNITUE VALUES FOR ELEMENT OF UNITUE VALUES FOR ELEMENT OF UNITUE VALUES FOR THIS ELEMENT OF UNITUE VALUES FOR THIS ELEMENT	OF UNITUR VALUES FOR THIS ELEMENT Lenge only etc. Edt of strent C		.:
Z1.23.C7. YERGE COMPLETED FOR ELEMENT C Total Number of Unitude Values for this flement Z1.Z3.C9. "Fric Completed for element Total Number of Thisus for this element Z1.Z3.11. Mitage Completed for this element Total Number of Unitud Values for this element Total Number of Unitud Values for this element	COMPANY CARA CARA CICKENT	-	
TOTAL NUMBER OF UNITUR VALUES FOR THIS FLEMENT 21.23.69. "FRE COMPLETED FOR ELEMENT Totel Number of Thitue Values for this element 21.23.11. Murte Completed for this element Total Number of Uniture Values for this element Total Number of Uniture Values for this element			:
21.23.69. "FREF COMPLETED FOR ELEMENT C Totel Humber of 1910 ue values for this element 21.23.11. Higher completed for element C 1014. Humber of Union values for this element C 1	TOTAL NUMBER OF UNITUE VALUES FOR THIS FLEM		
TOTAL HUMBER OF THIDLE VALUES FOR THIS ELEMENT 21.29-11. HERE COMPLETED FOR ELEMENT C 1 Total Humber of Unionz Values for this element of	453C5157865 504		
21.23.11. MLRGE COMPLETED FOR ELEMENT C 1 Total Number of Unioniz Values for this element	TOTAL HUMBER OF HAIDUE VALUES FOR		
TOTAL NUMBER OF UNIOUS VALUES FOR THIS ELEMENT Total states for states to the states of the states o	MERGE COMPLETED FOR ELEMENT C 1		
1 1757513 400 0040 0100 UV410	TOTAL NUMBER OF UNITUR VALUES FOR		
Ar 2011 COMPLEINS FUX ELENGEL C P	at at a state state soft strend C 15		;

ï t 1 1 3 ļ 2 į ļ i 1 1 1 1 \$: 1 ł 1 2 • ŧ : ÷ i ł e, ۰Ī ł ł ÷ Ĵ ł 2 į ÷ ł : I 317440 CHARACTERS : ; ; 1 ŝ 1 ï ÷ t 4 1 ; 1 2 ę • ţ į ; į j ; : č ٠ 1 1 ţ i F ţ 124 PARTITIONS ì 1151 t CATA VERSION c ł بر ŝ 172 410 *2 246 j N DATA 9455 ł ł ! 1 ļ ដ្ឋ į ł ł 22UJ4V 5 A CCEP 1 6.43 ٠ H ŧ, . ELEYSNT C 20 ELEYENT 0 22 This Element : 11435% ОF NULLS CPTATED IN THIS . "STAL NUHAR? OF UNIONE VALUES IN Total Size of сиогент рата 9ASE THIS ELEMENT ELEMENT C 2 THIS ELEMENT THIS SLEHENT TH REJECTED ŝ NUN-VALUED CATA SETS 1 i ACTS PTED DEFINITION VERSION ; 02125 SELECTION TABLES ARE COMPLETE 30010 ċ ŝ 505 507 F03 The Soluti 55 55 505 VALUES RE IEV Los. CATA SFT FINAL SOLT IS COMPLETE 21.23.14. 46965 004PLETED --Total Number of Unioue Values HERE COMPLETED OF UNITUE VALUES NERGE COMPLETED 21-23-19- 4436 COMPLETED - TOTAL NUMBER OF DRIDUE VALUES 1 21.29.28. 45145 COMPLETCO - TOTAL RUMACK OF UNIOUE VALUES 577201v SLJTAROS SHIGYOT ---21+29-29-28- 34/ ---Save 0414 9435 04 SFIL2: - Save/Load Coypleted 40 4 Je Hin NUMBER OF NUMPE 0 636498 NUT 31 R 30423 NESSE HSIJONICAS ł 21.25.23. . 59. 63. 13. 21-29-23. ł 71.23.15. - TOTAL NUMMER 21.23.23.23. TOTAL NUMMER ŧ į ; ł 1 ļ į l \Box

Creation of Data Base II using the new MARS VI system. Appendix G.

i

arter da 11 15 1711 arter da 11 15 17111 arter da 11 15 17111 arter da 11 15 17111 arter da 11 15 17111 arter da 11 15 1711111 arter da 11 15 1711111 arter da 11 15 17111	regre har 1757/17 regression 155 FELTY regression 155 FELTY reg	regrow p.M. FZF0//M. File and FZF0//M. File and FZF0//M. File and FZF0//M. File and FZF0//FILE FZF0/FZF0/FZF0/FZF0/FZF0/FZF0/FZF0/FZF0
<pre>error if if</pre>	<pre>mail is rely</pre>	are and the file of the file o
<pre>ment 11.5 #50/ret 10.011 by There services 10.011 by There services 10.011 by There services 10.011 by There services 10.011 by There is a constrained 10.011 by Ther</pre>	<pre>mail 11 F FEST *** F F F F F F F F F F F F F F F F F F</pre>	0 0
<pre>control ast countries source interview interview interview source interview inter</pre>	<pre>control is a sequencies space; is a sequencies space; is a sequencies where; is a sequ</pre>	<pre>representation synthesis works representation of the synthesis representation of the synt</pre>
<pre>rest fragments fragments under fragment during fragments under inst services fragments under inst services fragments under inst under inst under inst under inst under inst under inst under inst under inst under inst under inst under on observer inst under inst under inst under inst under inst under inst under inst under on observer inst under inst under inst under inst under on observer inst under inst under inst under inst under on observer inst under on observer inst unde</pre>	<pre>rest for the former content for the former of the for</pre>	<pre>rest fragment fragment interest worken inter. A interval fragment interest worken inter. A interval fragment interest worken inter. A interval fragment interest interval fragment interest worken interval fragment interest worken interval fragment interest worken interval fragment interval interval interval fragment interval interval fragment interval interval fragment interval fragment interval inter</pre>
<pre>rest of restrict in the rest of r</pre>	<pre>rest of restrict of restrict where where rest provide restrict where rest provide restriction where restriction restriction restriction restriction restriction restriction restriction restriction restriction restriction restriction res</pre>	<pre>1 cpr cpr cpr cpr cpr cpr cpr cpr cpr cpr</pre>
The second secon	A THE WAY AND A THE AND A	<pre>Mark Function Fu</pre>
<pre>main function for the second sec</pre>	<pre>Hand Construction Names Number Hand Construction Names Interferent Number Hand Construction Names Interferent Number Hand Construction Names Number Hand Construction Names Names Hand Construction Names Hand Construction Hand Construction Names Hand Construction Hand Cons</pre>	<pre>Hand Constructions interpreter winden Hand Constructions interpreters winden Hand Constructions and Specific Handles Hand Constructions and Specific Handles Hand Constructions and Specific Handles Hand</pre>
<pre>Main T Convert T Converted Number Main T Convert T Converted Number Main T Convert Converted Number Main T Convert Converted Number Main T Convert Converted Number Main T Convert Set Main T Converted Number Main T Convert Set Main T Converted Number Main T Convert T Converted Number Main T</pre>	The construction of the second	The construction of fragments in protect where the construction of fragments interpret where an under a construction of fragment interpret where an under a construction of fragment interpret where an under a construction of fragment interpret where a construction of fragment interpret where a construction of fragment where a construction of fragment interpret where a construction of fragment where a construction of fragment where a construction of the construction of fragment interpret where a construction of the constru
<pre>11 MAST FOR TATE TATE TATE TATE TATE TATE TATE TAT</pre>	<pre>11 Marging Fight Set Set Set Set Set Set Set Set Set Se</pre>	11000000 Construction According and Accordin
<pre>All content of a start and a sector a water interface a start of start and a sector advance interface interface a water interface interface a water interface interface a start interface interface interface a start interface</pre>	<pre>1101012 CFT 3.45711012012 WAREN 110101 CFT 3.571115 CFT 3.47115 110101 CFT 3.57115 CFT 3.47115 110101 CFT 3.57115 CFT 3.47115 110101 CFT 3.57115 110101 SFT 3.57115 1101011 SFT 3.57115 1101011111111111111111111111111111</pre>	<pre>11071CFECE FORM STATEST WAREN 11071CFECE FORM STATEST WAREN 11071CFECEFEE FORM STATEST WARENT 11071CFECEFEE FORM STATEST WARENT 11071C</pre>
<pre>Minterster start of relations in the second while Rise in the second sections of while Rise in operations is and section in the second section second section second section second section in the second section is and section in the second section section second section second section second section second second se</pre>	<pre>Interform of the restricting works Interform of the restriction works Interform of the restriction works Interform of the restriction works Interform of the restriction works Interform restors for must not an interform restors for must not</pre>	The formation of the sector structure works The sector structure structure works while the sector structure structure works while the sector structure structure structure works while the sector structure structure structure structure structure structure The sector structure s
<pre>reprint Firston of the first system of the reprint for the of the first system of the reprint for the rep</pre>	TITOTIC CIERCE OF ALL'ESTETISTICES NUMERA TITOTIC DELA CONTRETESTINATION TITOTIC DELA CONTRETESTINATION TITOTIC DELA SUBJECTIVA AND CIENCE AND FRANCE TITOTIC DELA SUBJECTIVA AND CIENCE TITOTIC TITOTIC DELA TITOTIC	TITOTIC CIEFT AND
TOPERED RESERTING SAUD TOPERED RESERTING SAUD STORAGE ENTRY SECTEMINETER NUMBER N. 33 STORAGE STORAGE S	<pre>Interface Answer and a secretarian Economy Interface and Secretarian Interface Interface an</pre>	Triprice of Funder 11 and 12 a
Tyroffers cost relative secret where rware	Typercist contractsNumber IN JackTypercist contractsSecretististice Numer N isTerritisticSecretististististististististististististist	Typercist coord functionsTypercist coord functionTypercist coord functionSeries interfect where R is the series interfect R is the series R is the series interfect R is the series R is the ser
<pre>22 TOTAL CATCA OF FALOM SELLENTINGENT MAR CLOCAL ENTRY ARE CLOCALS FOR THIS LOCICAL ENTRY ARE CLOCALS FOR THIS LOCICALS FOR THIS LOCICALS FOR THIS LOCICAL ENTRY ARE CLOCALS FOR THIS LOCICALS FOR T</pre>	TATIONAL CATCH OF MAJOR PREVENTIATION AND TATIONAL TATION	TATIONAL CATCH OF MADOR PARTICULATIONAL DEFETION OF THE TOWN TOWNED TO AN ANY STORY AND CLAIR THAT AND CLAIR TERM COL 1 10 COL 1 41 11 10 COL 1 41 11 11 10 COL 1 41 10 COL 1 41 10 COL 1 41 11 10 COL 1 41 11 10 COL 1 41
<pre>uprec: Tric Figers istar File Figers istar File Figers if File Figers if File Figers if File Figers if File Figers if File Figers if File File File if File File if File File if File File if File File if File File if Fi</pre>	The first fragment of the first and the first and the formation of the first of the	The second secon
HECKECS FRATA HECKECS CERTATA CCEERTATA CCEERCS CCEERS	HEGKEO F ENTRY A HEGKEO F ENTRY F ENTRY F C C C C C F C C C C C F C C C C C F C C C C C C F C C C C C C F C C C C C C C C C C C C C C C C C C C	Н. С.
HECKEOS FOR THTS FOR	44. 6NTAY A HEGKEGCE	и, ситат н ситат
HE CHIRT A HECKEO : FOR THIS LE ENTRIE HITE ENTRIE SELECTEO = CKCLUCED = CKCLUCED = CKSION ON VERSION	Н. С	HL. C. HT. H. C. H. H. C. H. H. C. H. H. C. H. H. H. C. H. H. H. C.
HECKEO I FOR THIS CCEPTEO = CCCEPTEO = CCCEP	HEGKE0 F C C C C C C C C C C C C C C C C C C C	HEGKEO F CKEO F CKEO F CKEO F CKEO F CKEO F K F K F K F K F K F K F K F K
HEGKEG HEGKEG FGKEG FGKEG FGKEG HITH HITH HITH FG FG FG FG HITH FG FG FG FG FG FG FG FG FG FG	HE C K C K C K C K C K C K C K C K C K C	HEG KEG KEG KEG KEG KEG KEG KEG KEG KEG K
HECC FOR FOR FOR FOR FOR FOR FOR FOR FOR FOR	HECKEC HECKEC CCEFT FFC FFC FTT FTT FTT FC FC F F F F F F F F F F F F F	COL 14 COL 14 COL 36 J COL 36 52 52 54 52 54 54 57 57 57 57 57 57 57 57 57 57
HEG KEO 1 HEG KEO 1 FOR THIS I FOR THIS HETH TEO 1 HETH	HECKEO HECKEO FOR I FOR HITH HITH HITH HITH HITH FOR HITH HIT	8) 111 CGL 18 2) 114 CGL 18 2) 170 CJL 27 2) 170 CJL 27 2) 170 CJL 25 2) 150 CJL 25 2) 150 CJL 46-CGL 24 2) 150 CL 46-CGL 26 2) 150 CL 46-CGL 26 20 CL 59 2) 150 CL 46-CGL 26 20 CL 59 2) 150 CL 46-CGL 26 20 CL 59 2) 150 CL 47 2) 150 CL 46-CGL 26 20 CL 59 2) 150 CL 47 2) 190 CG 71 2) 190 CG 71 2) 190 CG 71 2) 190 CG 71 2) 20 CL 100 CG 71 2) 20 CL 100 CG 71 2) 20 CL 100 CG 71 2) 20 CG
HEGKE0 FGKE0 FGKE0 FGKE0 FTHE0 FTHE0 FCCCCCE0 FCCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCCCC0 FCC0 FC0	HECKEO FOR THIS CCE ENTRIS CCEPTEO = SELENTEO = S	HEGKE0 HEGKE0 F C C E C K H I C C C C C C C C C C C C C C C C C C C
HEGKEO FOR FOR FOR THIS CLCE ENTRIE CCC FOR FOR FOR FOR FOR FOR FOR FOR	HEGKEO FOR THIS LEE ENTRIA CORFICE FOR THIS CORFICE SELECTEO SELECTO SELECTEO SELECTO SELECTO	HECKEO C FOR THIS C FOR THIS WITH REJECTEO = WITH REJECTEO = VCCEPTEO =
HEGKEO FOR THIS FOR THIS HCT FTRE HCT FTRE HCT FTE HCT FTE	HECKEO FOR THIS FOR THIS LLE ENTRIE HITH FO HITH FO FO FO CO CO CO CO CO CO CO CO CO C	HEGKE01 HEGKE01 F F CR THTS I F CR THTS HET F F CR T HEC F F TR TE HEC F C F CR TE HEC F C F C F CR TE HEC F C F C F CR TE HEC F C F C F CR TE HEC F C F C F C F C F C F C F C F C F C F
HECKEO FCR FCR FCR FCR FCR FCR FCR FCR	HECKEO: FOR THIS CCEPTEC = CCEPTEC = CCEPTEC = FS ACCEPTEC = FS ACCEPTEC O A T A S ON VERSION	HEGKE01 I FOR I FOR
HECKEO FOR THIS I FOR THIS I FOR THIS CCEPTEO FOR TE CCEPTEO FOR TE FOR THIS CCEPTEO FOR THIS FOR THIS FOR THIS FOR THIS FOR THIS FOR THIS COM THIS C	HECKEO FOR THIS I FOR THIS CCCEPTEO = KITH RE O FOR THIS FOR	HECKEO FOR THIS FOR THIS CCCETCTEO FOR THIS CCCETCTEO FOR TC FOR TC FO
HEGKEO FOR THIS FOR THIS HCTHTRE HC	HECKEO FOR THIS FOR THIS ILE ENTRIE WITH RELEC FOR STOR ON VERSION	HEGKE0; FGKT0; FGKT1 FGKT1 FGKT1 FGKT1 FGCT0 FCCT0
HECKEO: FOR THIS CCE FOR THIS CCE FOR THIS CCE FOR S SELECTEO	HECKE0: I FOR THIS I CCEPTE0 = CCEPTE0 = CCEPTE0 = FISTECTE0 = CCEPTE0	HECKE0: I FOR I FOR
D FOR THIS DLE ENTRIES CGGEPTEC = CGGEPTEC = Selfthee = Selfthee = Securces =	D FOR THIS DILE ENTRIES FITH RELECTED = FITH R	D FOR THIS CLEE ENTRIE CLEE ENTRIE CLEE FUTH CLEE FUTH CLEE FILL FE CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL CLEE FILL FILL CLEE FILL FILL FILL FILL FILL FILL FILL F
D FOR THIS DIE ENTRIES Huither Entries Huither Entries Sejectes - S Sejectes - S Secceptes Second C Second E Second E Se	D FOR THIS DIE ENTRIES HITERED = HITERED = HIT	D FOR THIS 1 FOR THIS 10 E ENTRIES 10 E ENTRIES 10 E ENTRIES 10 M V E A S 10 M V
J FOR THIS JLE ENTRIES ACCEPTED = CCCEPTED =	. FOR THIS ILE ENTRIES ILE ENTRIES FITH FEO = FITH FEO = FEUECTER E FEUECTER E FEUECTER E FEUECTER E FEUECTER E FEUECTER E FEUECTER E FUECTER E FU) FOR THIS DLE ENTRIES TGGEPTED = HITH REJEC = FELICTED = FS ACCEPTE = 160 0ATA S 0ATA S 0ATA S 0A VERSION
0 FOR THIS 0 FOR THIS 0 FOR THIS 10 E ENTRIE 11 H RE 0 = 11 H RE 0 = 11 H RE 0 = 12 C C C C C C 12 C C C C C C C C 12 C C C C C C C C 12 C C C C C C C C C 12 C C C C C C C C C C 12 C C C C C C C C C C C C C C C C C C C	0 50% THIS 0 50% THIS 0 50% THIS 0 50% THIS 0 10% 20% 50% 10% 0 0 10% 20% 510% 0 0 10% 20% 50% 0 0 10% 50% 50% 50% 50% 0 0 10% 50% 50% 50% 50% 50% 0 0 10% 50% 50% 50% 50% 50% 50% 50% 50% 50% 5	J FOR THIS DLE ENTRIES ALTH REJEC = Hith Rejects Frences = Frences
1 FOR THIS 1 FOR THIS 1 E C R THIS 1 E C F THIS 1 THI RE JE 1 E C C C F C S 1 E C C C C C C C C S 1 E C C C C C C S 1 E C C C C C C C C C C C C C C C C C C	1 FOR THIS 1 FOR THIS 1 FOR THIS 1 FOR FIEL 1 FOR	J FOR THIS JLE ENTRIE CCEPTED = Hith Rejected = for the second on version on version
	0.6 6 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ULE ENTIFIC ULE ENTIFIC HITH REJECTED HITH REJEC
WUNDER OF VALUES ACCEPTED = 11330 WUNDER OF VALUES ACCEPTED = 11330 WUNDER OF VALUES REJECTED VALUES = 0 WUNDER OF VALUES EXCLUCED = 0 WUNDER OF VALUES EXCLUCED = 2749 WUNDER OF NON-VALUED DATA SETS ACCEPTED = 0 WUNDER OF NON-VALUED VARIANTI	WUNDER OF VALUES AGGEPTED = 11330 WUNDER OF VALUES AGGEPTED = 11330 WUNDER OF VALUES REJECTEG VALUES = 0 WUNDER OF VALUES EXCLUCED = 0 WUNDER OF VALUES EXCLUCED = 2749 WUNDER OF VALUES EXCLUCED = 2749 MUNDER OF VALUES AGGEPTED = 0 NUNDER OF NON-VALUED DATA SETS AGGEPTED = 0 NUNDER OF N	WUNGER OF VALUES AGGEPTED = 11330 WUNGER OF VALUES AGGEPTED = 11330 WUNGER OF VALUES REJECTED VALUES = 0 WUNGER OF VALUES REJECTED = 0 MUNGER OF NON-VALUED DATA SETS AGGEPTED = 0 HUNGER OF NON-VAL
KUMARER OF ENTRIES WITH REJECTED VALUES = V NUMARER OF VALUES REJECTED = 0 NUMARER OF VALUES ERCOTED = 2749 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED ACCEPTED =	NUMMER OF ENTRIES WITH REJECTED VALUESNUMMER OF VALUES REJECTEDNUMMER OF VALUES REJEDNUMMER OF VALUES REJEDNUMMER OF VALUED DATA SETS ACCEPTEDNUMMER OF NON-VALUED DATA SETS ACCEPTEDNUMMER OF NON-VALUED DATA SETS ACCEPTEDSAVE DATA PASESAVE DATA RESE ON OB1953M:SAVE/LOAD COMPLETEDEXIT:	WUNDER OF ENTRIES WITH REJECTED VALUES FUNNAGER OF VALUES REJECTED = 0 NUNDER OF VALUES REJECTED = 0 NUNDER OF VALUES REJECTED = 2749 NUNDER OF NON-VALUEO DATA SETS ACCEPTED = 0 NUNDERSH 5JHLEV DEFINITION VERSION 1. DATA VERSION 1. SAVE DATA ZASE, 0 08/02/72 SAVE LOAD COMPLETED EXIT: EXIT:
NUMMER OF VALUES REJECTED = 0 NUMMER OF VALUES EXCLUCED = 2749 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 SAVE DATA SETS OF NON-VALUED ACCEPTED = 0 SAVE DATA SETS OF	NUMMER OF VALUES REJECTED = 0 NUMMER OF VALUES REJECTED = 2749 NUMMER OF VALUES CEEPTED = 2749 RUMBER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 SAVE DATA SETS ON URANTING VERSION 1 SAVE DATA SETS ON URANTING VERSION 1 EXIT:	NUMMER OF VALUES REJECTED = 0 NUMMER OF VALUES REJECTED = 2749 NUMMER OF DATA SETS ACCEPTED = 2749 NUMMER OF NON-VALUEO DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUEO DATA SETS ACCEPTED = 0 19.26.35 SAVE DATA 245E ON 08/02/72 SAVE DATA 245E ON 08/02/72 SAVE LOAD COMPLETED EXIT:
NUMMER OF VALUES EXECUTED = 2749 NUMMER OF DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMERY STHLEV DEFINITION VERSION 1 DATA VERSION 1 GROUNDFISH STHLEV DEFINITION VERSION 1 DATA VERSION 1 SAVE DATA 2455 ON 081953M; SAVE/LOAD COMPLETED EXIT:	NUMMER OF VALUES ACCEPTED = 2749 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMERY 53HLEV DEFINITION VERSION 1 DATA VERSION 1 5400HDFISH 53HLEV DEFINITION VERSION 1 DATA VERSION 1 SAVE DATA 2455 ON D81953H1 SAVE/LOAD COMPLETED EXIT:	NUMMER OF VALUES REACCOFTED = 2749 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMER OF NON-VALUED DATA SETS ACCEPTED = 0 NUMMERSH 5JHLEV DEFINITION VERSION 1. DATA VERSION 1 SAVE DATA 2455. 08/02/72 SAVE LOAD COMPLETED EXIT:
RUNGER UT VALUED DATA SETS ACCEPTED = 0 RUNGFISH SIMLEV DEFINITION VERSION - 1 DATA VERSION 1 Groundfish Simlev Drivition Version - 1 Data Version 1 Save Data 2455 on DB1953M; Save/Load Completed	RUNGER OF NON-VALUED DATA SETS ACCEPTED = 0 number of Non-Valued Data Sets Accepted = 0 Groundfish 53hlev definition Version - 1 Data Version - 1 Save Data 2055, 00 081953H: Save/Load Completed Exit;	RUNGER UT VALUED DATA SETS ACCEPTED = 0 RUNGERSH STALEV DEFINITION VERSION 1 DATA VERSION 1 GROUMDFISH STALEV DEFINITION VERSION 1 DATA VERSION 1 SAVE DATA EASE ON OB1953K: SAVE/LOAD COMPLETED EXIT:
GROUNDFISH 5IHLEV DEFINITION VERSION - 1 DATA VEÀSION 1 19.26.35. 0 08/02/72 Save/load completed Exit:	GROUNDFISH 5IHLEV DEFINITION VERSION 1 DATA VERSION 1 19.26.35, 08/02/72 Save data 2656 on 08/02/72 Save/Load completed Exit;	GROUNDFISH 534LEV DEFINITION VERSION 1 DATA VERSION 1 19.26.35. 03/02/72 Save/Load completed Exit:
GROWNDFISH STHLEV DEFINITION VERSION - 1 COMPANY - 19.26.35. 08/02/72 19.26.35. 08/02/72 Save/load completed Exit:	GROWHFISH SHALEV DEFINITION VERSION - 2 COMPLETE 19.26.35. DB/02/72 Save Data 2055 On DB1953M: Save/Load Completed Exit;	GROUNDFISH SAMLEV DEFINITION VERSION - A COMPLETE ON DEVISION - A COMPLETE ON DE1953K; Save/Load Complete O Exit;
19.26.15. 03/02/72 Save Data 2455 on 081953m; Save/Load Completed Exit:	19.26.35. 03/02/72 Save data 2455 oy 081953m; Save/load completed Exit;	19.26.35. 04/02/72 Save Data 2455 O4 0819534; Save/Load completed Exit;
SAVE DATA EASE ON OB1953M; Save/load completed Exit:	SAVE DATA PASE ON OB1953M: Save/load completed Exit:	SAVE DATA ENSE ON UB1953M: SAVELLOAD COMPLETED EXIT:
SAVE/LOAD COMPLETED Exit:	SAVE/LOAD COMPLETED Exit:	SAVE/LOAD COMPLETED Exit;

10 000	E 701			•				1
	010 011 8455 FROM 0	- - - - -		······································				1
11201 1221 1221 1221 1221 1221 1222 12200 12200 12200 12200 12200 12200 12200 12200 12200 12200 12200	ίς. •	2 HHERE C1 L	5304171		-		•	•
11211 12211	111	1	- F					1 1 1
23 4500 25 1400 25 1400 25 25	111251 111251 111251							
120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 120 1200 1200 1200 1200 1200 1200 1200 1200 12000 </td <td>21 450 21 1400 2) 2500</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>: ·</td>	21 450 21 1400 2) 2500							: ·
2017 11207 1209 1200 1200 1200 1100 111 1200	117504655 · · · · · · · · · · · ·	• 						
22 15300 250 413 250 413 250 413 250 413 251 12 271 251 272 250 2999 200 1150 201 1205 201 1205)5 3)87 1)254		4 b a a 14 a 14 a 14 b a 27 #					1
22 100 250413 250413 250413 2012 2012 2012 2012 2012 2020	1)205 1)207 2) 15300,							
1182 1182 1182 1186 1186 1186 1186 1186	2) 350 413							•
11251 1251 1255 1550 1500 1500 1510 1525 1526 1526 1526 1526 1526 1526 1526								i
2) 11500 2) 11500 2) 1100 2) 1100 2) 1100 2) 2500 2) 2500 2) 2500 2] 2500 2] 2500 2] 2500 2] 2500 2] 2500 2] 2500 2] 2500 2] 2] 2] 2] 2] 2] 2] 2] 2] 2] 2] 2] 2] 2	192(1 19212 1207	- - - -	÷	· · · · · · · · · · · · · · · · · · ·	· · · · ·			•
5530414 5530414 1205 1205 1205 1205 1205 1205 1205 1205	2) 11500							ļ
	2) 1100)530414)1		n anna an an Anna Anna Anna Anna An					
1) 207 1) 205 2) 2550 2) 2550 2) 2550 1530 415 12 25 12 25 12 12 12 25 12 12 12 25 12 12 12 12 12 12 12 12 12 12 12 12 12	9.99 1.31.07 1.31.07							-
23 5500 29 550 1530415 12 12 12 12 12 12 12 12 12 12 12 12 12		· · · · · · · · · · · · · · · · · · ·				4 • •	•	:
530415 24 3317 11236 11231 11231 1237		• •		Bille Martin Barner			····	
3317 3317 11296 11236 11231	530 21 25	,		, van men en weer en de ster een de ster de ster een de ste		•	•	;
11296 11231 11281						5 7 1 1 1 1 1 1 1 1		; : 1
	11296 11231 11207	-					· · · · · · · · · · · · · · · · · · ·	•

10. A the manual and a second a second of the second second second second second second second second second s

	1)536415												
								•			-		
1												- - -	ι.
	111255 221 3000 11530415	•	· · ·				• • • •						i
													Į
	2020	1	. k								•		(,
1	221 2000						•			• • •			-
	- N -	A			•	1	: 			t ,	•		
	13155												
	ຊື່ຂອງ						-			•			· .
	-22) 1)5304												
	♦ N	•						 -					1
	- 11)252 221 1000 1}530416								•				: •
	4)1			-	-							2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
1	- 13904										. 		:
1	11)252 12)252 12)252												; 1
	111231												
	- 22) 335		والمتعاربة والمستعرفة والمستعرفة والمستعرفة والمستعرفة والمستعدية والمستعدية والمستعدية والمستعدية والمستعد والمستع والمستعد والمستعد والمست					 					
	134) 						- -	} ~ ~: ₁
	- 8) 6			-				T 					
Į.	11) 252 22) 10 End C	-											
	1/11 8041) 52)5					•	-						
1													

.

,

90[.]

•	Ì	ţ	f !		ļ - 1		ţ	• • •	ļ	1	· !		Į.	• • •		• • •	ŀ	ļ		1	ŗ	•	۱.	ł	ļ	•	!	•
•			• •	• ·	1 -	;	:	· · .	•	•	• .		•	:		:	÷				; ; ;]	:		i.	:	•
;							1	••			;			;			•	, , ,	•	:	:		:		, [ſ	1	
· · ·		!		•		:			;	1	•		1	 t		•	•	1		:	I		:	1	•			
1		 :	-	-	i		1		:							:	•			:	1						:	
				•				_	:					:				1	 		; ; ;		• •	: ; ;		*	1	
•		•							r t				1			•	•			1	ļ		•	ļ		: :	•	
		;	1	•										Ì		!	:	;				. .	i i	1			:	
•	2		1	-		1			ĺ					: • •		•	ţ		ļ		1		;	1		t		
1				- 	1				1 1 1						,			•	 . [.] 					;;;		i	i	
ļ.		ł	1.	1		;			1			(1				1				•					1	-	
i .				1									1	i									• •	;	•	ļ		
	ļ	:		•		1			-					-			-		1 1	•					_	•	•	
		:										1		•		· ·	;	_								: : :	:	
i. I	-	i	•			1								•		i i	1	•		•			•	:		1 . 1 . 4 .	:	
1			ļ	: . :		1	•	-								: f t				:			1	ا م۔ ا		1 1	1 :	
	-	;		•					:					-			-			•	-			•			:	
	ļ		T .	ł				1		•	-)	-				• •			-	-			•	
				: !		۲ ۱		ļ	1. .		ļ.											i i				•		
		;		•	İ	. :		1			1	: ; •			-		•		1 			; ļ	•		-	•		
						•		•	•			i	:			•				: : :			÷				:	
Read and and			-			,			:		•	r				•				•		1	÷					
Ĭ		. 1	; · 1	-				•	•	•	ł	•	:							:		į	ł		1	2	:	
			• •	;	ļ	:		:			: ; ;	•	:	•		1							:		•	•	;	
5		:	:	•	ł	•		1.	ł		ļ.	•			:	-			•	•		. .		•	!	i	•	
ł		- 110 - 26 - 26	; 9 2	۰ میر م		.≈.	°.	1.	520	120		; ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	5 22 2	120	120	110	120		110	194 194 194	- - - - - - - - -	4 7	-	° ۲'	22 5	32	22	
		100	566	6 6 6 8	0.00	2 2 2 2 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3		26	56		364		53	e e	.	F 6 8	2	6 6 6	:20 1	ŝē	96	÷.	- ÷	66	60			

• 1

1

1

1

e)120	01116		01100	91110 915	9120	6) 128 6) 128	9120 915 9120	e) 22 9)5 1, 23	915	02119
•	•	1			,	•	Ì			•
•	1.	ৰ 👘			1		1		1	Į
	! ;	•							: : :	
		•		•	i		•		•	•
2		•		•		1	•			ł
		•	ļ			· •	1	•	•	
•	ľ			,	!	:	1	• • •	I H	
• •					i .	-				-
					1		i	•		
	 				:					ł
·.					i 1 . 1	. – .			}	
· ·. ·		•	1	i ,						
		a New York State (1997) - Anna Anna Anna Anna Anna Anna Anna An	1	•		• • •	I			
, .								-	-	
:		•								
							a management of the second			
•										
						•				1
		1		L	. 		i i			
				•			+ 1 +	1 6 6	i	
				:		:		•	t 1	•
		!		:	1		; 		1 1 1	1
	ł 	•	ł				:		1	ł

	• ·		ļ	(** * 1		1			!.	∦ ₹	•
		, 	•		I						•
		i			i		•		•		
	1,		-		İ	. !					
			5							•	
		1		· ; ! ;	1		i i				
	;	1		·						,	
	1	1	•		1		-		• /		•
i	f 1 1 1 1		• 		· 1					•	•
1	•	· .	· ·			. 1	:		-		
	-				1	,				•	
					į					•	•
	, I						1				÷
		-	•		1	1			•	•	
					1				• •		1
						.					:
		ŀ,	, i			ł					•
					i	-				1	
					i		į			i	-
								1		ļ ,	2
ļ					ł		į	,			•
· ·		. 1				1					•
									•	•	; .
-							-			•	
						:				i	1
					i						.
			•		.		1				1
							i	. 1			1
					i			-		l	
		,			į			. 1		1	
ł	1		·					l		•	-
•	•				Ì	:		_			-
					1			•	•		
		•								! 	
							1			•	
				· .			1				•
			i :			·	•	-		:	
											•
					i					:	
·	l -			; ;						• •	
1	!	: 1						•	l I	•	
1								. 1			
	. ;	•••		•		·		- 1		•	
	•										•
•			-		. i		• :				i
			-		1		•				, ,
_								ا		; 	
1. in 1	02119	5	,		1.5	515		`	, N, N, N		
											222
)		;			:	1	. <u>.</u>	ĺ	1
			-	. •							

9

; ;

į

						「「「「」」」」」「「「」」」」」」」」」」」」」」」」」」」」」」」」」				• • •			C6 EQ 1201			1666							HHERE CO HAS	RRR.
- 02 02	915 51120 81110		 516 22 10	28	 		NO OF RETRIEVAL	PRINT SUM C22+C11 WHERE C13 EQ 02 4ND BOAT CLASS and sofits fore en 360	221 2.7500550000 521 2.75005500000	111251	11755	11/251) OF RETRIEVAL Nitpy HHERE C2	SATISTICO -NONFATAL	UIAGNOSTIG-9%ERE GLAUSE NOT SATISFIED -NONFATAL 	RECUEST 109 Post CLASS 23 1 1 END WHERE BOAT NUMBER ED -	SE LAILTERED	4 ЕА 1 1 ЕМО ННЕЧЕ С1 ЕЛ 531127 АМО С8 • Selected Data Sfis	DATA BASE LNALTERED Reduest	ENTEY HHERE CL EQ 53121 SSTICHE L EC 531213	- BAG ES UNSATISFIED - Hurger Clause and Sa - Erses of the Area	DATA BASE UNALTERED		

	; •	94		
			n (* 1970) 1	
		i i		

1 ì

į

.

5 SELECTED DATA SETS

RE OUT \$1 ...

Linnie

L Z TERMINATED		
REFUEST Refeicval: Dualley Ce.C9 NHERE C1 Exists; END of Refrieval		
REPORTER: FOURAL OR GENERATE - Lines Per Page IS -		.
LUT REAV FOR FCRMAT - TILE IS CROUNDFISH 1953 BOAF INFORM HEADING IS BCAT NO.80AF CLASS DETAIL IS CO.CO: DETAIL IS CO.CO:	ation:	
NERATE +		1
· · · ·		1
		i
an ann ann an an ann ann an dùrachad a' meann a		
		1
	ar de sande i maren e a mart a marte a mart a mart a mart a mart de se	
		i ; ;
		:
		•

..... -All and the second . ! 1 ---------; ĺ ł ł

1

......

÷ . . . 545

1

: .

ļ

-		-		
	I		-	•
•	•	•		

											-					• . *	,				
•-		!	}	1	1	1	ţ	rt jt	• • ••	~ ~		1		t 1771	1 -	1	!	ŀ	! !	r i	;
		•	! :	. .	ţ		I.				I	•		:	,		:	;		: :	
		İ	l			•	•	,				•				ŕ		t	· 1	ь	:
		•		:	i P			;		i		! !	3	:	2	:	i		:	i i	
		•		:			i i			•		· •		;	•	t 1	t t	•	:	t : : ; .	
						1 ,				•	 	; • •		ļ	i	ţ	i į	: / 	•		
		! ; {	1	!		•			I		 	:			•	•			• •		
						1	į		•			·		;					!		
		•	Ι.	1 1	ł		i					i :		:		•	i :	İ.		!	•
-			1	1	ļ	•		:		• - 	•			•	:	: 1		1	•		:
		•	-								I	! . ; .		;		!	1	!	• •		•
·				 1	.									;	•	• • • •	- • •	İ		•	
			· .	1		ł	ł					ļ,						t -			į
				1			ł	i	ĺ	1				!.							1
			1	ł										i .		ļ			•	•	:
		1		1			Ì	ļ	[. 		6 1 1							; . ·
ar t		: :	:	1		;				•		i		1	; ; ;		•		ļ	:	
		ţ	1							: :				1 	1 i 1	1	1	1	•	: '	
			l.	1			i	1		;					:	! .				i .	
				1 		i							•		1				•	-	•
÷	ľ	ļ	Ì	1		ł	ŀ					1			ί.		1				
• • •		}		•						[]	1				1	* : •					r t
		;								!				1	-			i -		• •	-
.`	. -					1	1	•		r .		į				\$ 5 . ·			•	: ;	: : :
	i i					•				i '				i -	1			ļ	•	ł	•
		• •			ľ	:	!	;	:				-		2	1° :	i.		1	•	
		•				-	-			1 1. 1 1.		 		1 · 1	•	•	1				
•		NON	t i	, :	1	-•	1				•			i i	:	: : •	1	1			` .
		- 4 F &	1 .	1	1		i	•		-	• }.	• 	·. ł			!	• •	•	•	۲	
*	1	GROUNDEISH 1953 BOAT INFORMATION	• •		ļ	•	1		•	i 	•			1	!	: .	ł		•		
		T T T	455	633	1	1 P3 P3 E		P) P)	***	3 M M I	4 In In (:	10000 	9 M M I	nn '					•		
	÷	ž	5 5		i 1							;			•	•		į			
	, :	195	BOAT CLASS			1	ŀ				i		;	i . *	•	i	į				1
	ļ	1513		1		:	•	•	• • !		ł	 	:		• •	!	ł	1			¥
•	•	0 N NO	804T N	9 न न	ا. سەمەب	ومع			: e vo vo i	1 • <u>12 1</u> 2		1 N 10 10 1	6 10 3	1 		1	1		•		
	l ·		•	233	4	1	,			112 112 112	4	L	et						•	1 2 1	ļ
	ŧ.	4		4	12	1	i.		1	•	t	• •	ť	۰.	-	1	•	e -	•	•	•