THE GENERATION OF ENTITY-RELATIONSHIP DIAGRAMS FROM USER DOCUMENTS

by

DARRELL W. WOELK

B. S., University of Kansas, Lawrence, Kansas, 1971

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements of the degree

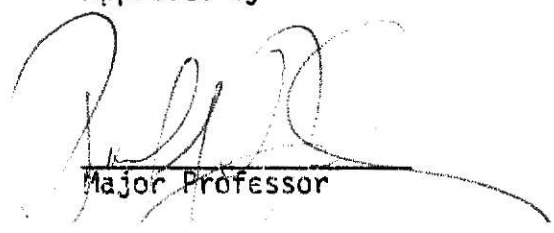
MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhatten, Kansas

1981

Approved by



Major Professor

ACKNOWLEDGEMENTS

I would like to thank Dr. Paul Fisher for his ideas and assistance in the completion of this report.

TABLE OF CONTENTS

List of Figures

1.0 INTRODUCTION

Research in the area of Data Models has been increasing in recent years. The search has been for a model which can capture the full semantic meaning of the data which it represents. Peter Chen has proposed the Entity-Relationship Model as a model which allows a logical view of data to be described at a high level. [1] Chen also proposed a diagramatic technique for exhibiting Entities and Relationships called an Entity-Relationship Diagram. This technique supplies a graphic description of the data model which simplifies the understanding of complex data relationships. Various proposals have been made for the implementation of graphical design aids which will automate the generation and modification of Entity-Relationship Diagrams. [2] [3]

There has also been research into the utilization of User Documents to automate the generation of a data base schema. [4] This paper will propose a system which utilizes User Documents to generate a graphical Document E-R Diagram. The system will then assist the user to interactively manipulate this Document E-R Diagram to generate a true Entity-Relationship Diagram for the data. The system guarantees that all data relationships represented in the Document will be represented in the resulting data model. The system saves the original Document E-R Diagram and will have sufficient information to generate navigation paths to access data for inclusion in Output Documents.

2.0 DATA BASE DESIGN METHODOLOGY

The existence of a good, comprehensive Data Model does not guarantee the ease or accuracy of designing a specific data base. While the model can help conceptualize the data organization, there is still much work required to actually identify the specific data items and insure that all data items are included in the data base. The job of gathering and organizing this data can be simplified through the use of automated data base design tools. These tools may be general in nature or they may be oriented towards a specific data model.

Fisher and Hollist have proposed an integrated data base design methodology which utilizes a design DBMS. [5] The design DBMS manages the information incident to the data base design effort such as Data Item Names, Value Ranges, Descriptions, etc. The steps of the integrated data base design methodology are listed below:

1. Survey the organization.
2. Identify the Entities and Relationships of interest to the organization.
3. Diagram the Entities and their Relationships for each user view.
4. Synthesize the organization view by combining the user views.
5. Specify the mapping of each Relationship.
6. Link the data items to appropriate Entities and Relationships and specify the domains.
7. Isolate the keys for each Entity and Relationship.

8. Translate the resultant organization schema into an actual realization utilizing an hierarchial, network, or relational implementation.

A methodology as described here gives the data base designer two advantages:

1. There is some rigor to the design process so that the designer is always aware of his status and his progress towards the design goals.
2. Automated tools take over many of the bookkeeping functions and allow the designer to concentrate on understanding and organizing the data.

This paper will attempt to add further rigor to this methodology by supplying further automation. The methodology is based on the Entity-Relationship Model and the designer must, in Step 2, identify the Entity Sets and Relationship Sets of interest to the organization. In Step 6, the designer must then decide which Data Items are associated with each Entity Set and Relationship Set. The proposal being made in this paper will allow the designer to utilize the User Documents to partially automate the generation of these Entity Sets and Relationship Sets.

Sections 3, 4, 5 and 6 will present some of the underlying concepts which will be used later in the paper. Section 7 will present the algorithm which is used to map Document Entity-Relationship Diagrams into true Entity-Relationship Diagrams. Section 8 will present an example data base design using the algorithm.

3.0 ENTITY-RELATIONSHIP MODEL

The Entity-Relationship Model takes the view that the real world consists of Entities and Relationships. [6] An Entity is a "thing" which can be distinctly identified. Examples of Entities are an event, a place, or a person. A Relationship is an association among two or more Entities. For example "employee-manager" is a Relationship between two "persons" Entities. Another way of describing Entities and Relationships is to think of them as being analogous to nouns and verbs, respectively in a sentence structure. [7]

An Entity is a member of an Entity Set if it has properties common to other Entities in the Entity Set. For example, a person can be a member of the Entity Set EMPLOYEES or the Entity Set MANAGERS or both. A Relationship Set is a group of Relationships of the same type. There is a Relationship Set between Employees and Departments since Employees work in Departments. Entities and Relationships have properties which can be expressed as Attributes and Values. A Value Set is a group of Values of the same type. An Attribute is a link from an Entity Set (or Relationship Set) to a Value Set. For example, Salary is an Attribute which maps Entities in the Entity Set EMPLOYEES to Values in the Value Set DOLLARS.

There are two forms of Entity Sets, Regular and Weak. In a Regular Entity Set, the Entities can be uniquely identified by the Values of their own Attributes. In a Weak Entity Set, this is not true and a Relationship must also be used for unique identification. For example, DEPENDENTS may be identified by their names. However, unique identification may require the values of the primary key of the employees supporting them. In other words, their Relationships with the employees is part of their uniqueness.

There are also two forms of Relationship Sets, Regular and Weak. If all Entities in a Relationship are members of a Regular Entity Set the Relationship Set is Regular. If any of the Entity Sets are Weak, the Relationship Set is Weak.

Chen also proposed a diagrammatic technique for describing Entity Sets and Relationship Sets. An Entity Set is represented by a rectangular box and a Relationship Set is represented by a diamond-shaped box. Figure 1 presents an example of an Entity-Relationship Diagram. DEPARTMENT and EMPLOYEES are Regular Entity Sets. DEPENDENT is a Weak Entity Set and is, therefore, drawn as a double walled box. The mapping of the Relationship Set between the Entity Sets (1:N, N:1, N:M) is also shown on the E-R Diagram.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

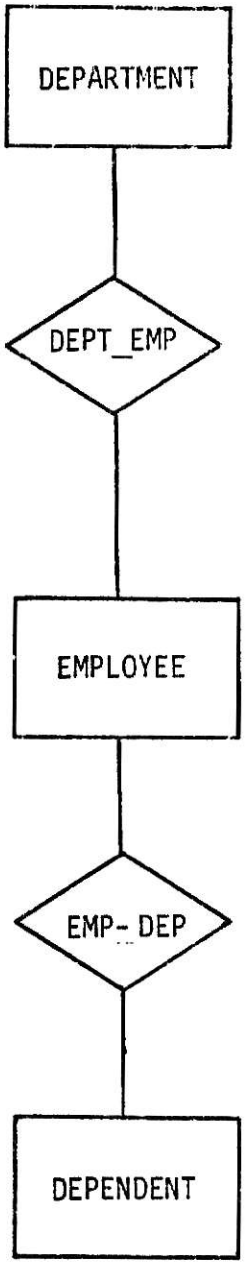


FIGURE 1. E-R Diagram Example

4.0 DOCUMENT ENTITY-RELATIONSHIP DIAGRAM

The most important aspect of this proposed system is the concept of a Document Entity-Relationship Diagram. This concept requires that Documents be thought of as Entities. Each Document Entity has Attributes such as Type (Input/Output), Sender, Receives, etc. However, a Document Entity can also have Attributes which represent Field Names of the data contained in the Document.

For instance, a Document named SALARIES might contain the names and salaries of all employees in a specific department. The Field Names DEPT_NAME, EMP_NAME, and SALARY are, therefore, Attributes of the Document Entity SALARIES. However, EMP_NAME and DEPT_NAME are also Attributes of an Entity such as PERSON or DEPARTMENT. The fact that EMP_NAME and DEPT_NAME exist on the same Document indicates that there is some Relationship between EMP_NAME and DEPT_NAME. This Relationship must be maintained in the final Entity-Relationship Diagram.

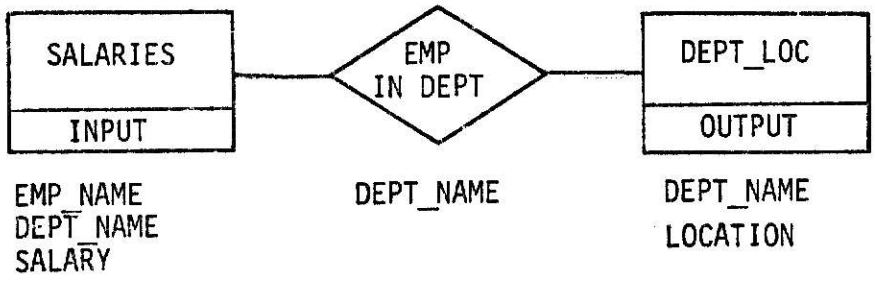
Furthermore, another Document, named DEPT_LOC, might also exist. This Document might contain the locations of all departments in the organization. Since the SALARIES Document and the DEPT_LOC Document both contain the Field Name DEPT_NAME, there is some rationale for saying that there is a Relationship between the two Document Entities.

Figure 2.b is an Entity-Relationship Diagram for the data described above. There are two Entity Sets EMP and DEPT. Each Entity Set has unique attributes which are listed below the Entity box. Figure 2.a is a Document Entity-Relationship Diagram. The Attributes are listed below each Document Entity box.

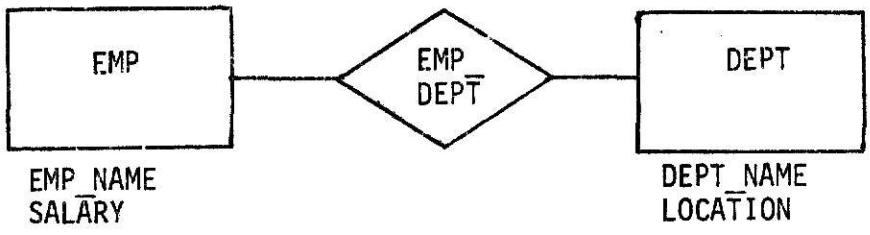
Note that an Attribute can be listed under more than one Document Entity box. This list of Attributes will be referred to as the Document Entity

Data Item List. The Attributes listed under the Relationship diamond-shaped box are the Attributes which are common between the two Document Entities. This list of Attributes will be referred to as the Relationship Common Data Item List. Each Document Entity box contains a label of Input, Resident, or Output to indicate the type of document which it represents.

The proposed system will generate a Document E-R Diagram such as Figure 2.b based on the users *Documents*. The system will then assist the user in translating the Document E-R Diagram into an actual E-R Diagram such as Figure 2.a.



(a)



(b)

FIGURE 2. Comparison of E-R Diagram and Document E-R Diagram

5.0 IDENTIFICATION OF DATA ITEMS

There are some data base design activities which are necessary regardless of the Data Model being utilized. One of these is the identification of the Data Items of the organization. For example, an organization may assign an Employee Number for each employee. This Employee Number must, therefore, be part of the data base regardless of whether it is treated as a field of a Codasyl record called Employee or an Attribute of an Employee Entity.

User Documents provide a good source for Data Item Names. A complete set of User Documents should supply all of the Data Items which the organization uses in its activities. Unfortunately, different User Documents may refer to the same Data Item by different Names. Another problem is two different Data Items being referred to by the same Names on different User Documents. Added to these cases, can be the derived Data Item Names which show up on a User Document but are not actually stored in the data base.

The data base designer must analyze the Data Item Names and generate a complete non-ambiguous list of Data Items for the data base and correctly link these Data Items to the correct Data Item Names in the User Documents. There is definitely a need here for automated assistance for the data base designer.

User Documents can be classified as Input, Resident, or Output. [8] Data Items may occur in one, two or all three of these types of User Documents. Analysis of the existence of a Data Item in each type of User Document will be helpful to the data base designer in understanding the semantics of the organization's data.

For example, a Data Item which exists in an Output Document but not in an Input or Resident Document will be a derived Data Item. The system proposed in this paper will assist the data base designer by graphically showing Data Items which are not represented in all three types of User Documents. The system will also assist in determining different Data Items with the same Names. In addition, it will assist in finding instances of the same Data Item with different names.

6.0 GRAPHICAL REPRESENTATION

The Entity Relationship Model and many of the other data models proposed for logical data base design present information requirements in a graphical representation. Chan and Lochovsky have proposed a graphical data base design aid for the Entity-Relationship Model. Their work serves as a basis for the system proposed here. [9] The Entity-Relationship Diagrams and Document Entity-Relationship Diagrams are effective only when they can be displayed or printed with reasonable clarity. Ideally, this system should be implemented with a graphics terminal. The location and spacing of Entities and Relationships on the screen must not be restricted by a lack of terminal features.

The ideal terminal would be one which allows zooming in on details of the E-R Diagram. A combination of system software and a good terminal should allow the following scenario to take place:

> The data base designer has all of the organization's Document Entities and Relationships displayed on the screen. Each Entity or Relationship symbol may be very small with only abbreviations inside the symbol. As the designer zooms in on an area of the screen, the symbols become larger and at some point the full User Document Name appears in the Entity Box. Zooming in further causes the Data Item Name Lists, Common Data Item Lists, etc. to appear. The user can then move about in the Document Entity-Relationship Diagram looking at other Entities in detail. Zooming in further will cause an example of the actual Document to be displayed.

The use of a Mouse-type cursor, windowing, and special function keys would further enhance the use of the system.

The remainder of this paper will not specify the details of the implementation of the graphical representation. The paper will concentrate on the overall functions and not a specific implementation.

7.0 DOCUMENT E-R DIAGRAM TRANSLATION ALGORITHM

Section 2 reviewed a methodology for data base design. This Section presents an algorithm for automating some of the steps in that methodology. For completeness, the description of the algorithm will encompass all steps of the design phase as described in the methodology in Section 2.

The Document E-R Diagram Translation Algorithm is summarized below. The succeeding Sections will then discuss each step of the algorithm in detail.

1) Data base designer surveys the Organization and enter the data concerning Documents, Data Items, Value Sets, etc. into the system.
2) System generates a Document E-R Diagram.
3) Designer analyzes Data Item Names for duplication and ambiguity with aid of the system.
4) Designer enters a description of each Relationship among Document Entities.
5) Designer indicates the ownership of Data Items which are common to more than one Document.
6) System makes deletions from Document Entity Data Item List for Data Items which are not owned by the Document Entity. Any Data Item deleted from a Document Entity Data Item List is also deleted from any other associated Relationship Common Data Item Lists.
7) System eliminates Document Entities and Relationships which have empty Data Item Lists and Common Data Lists, respectively.

8) Designer manipulates the resulting Entity-Relationship Diagram using the SPLIT, MERGE, SHIFT, MOVE, and CHANGE operations.

9) Designers specifies the mapping (1:N, N:1, N:M) of the relationships among Entity Sets.

10) Designer identifies the Key Attributes for all Entity Sets and Relationship Sets.

7.1 ORGANIZATION SURVEY AND DATA ENTRY

The designer must first collect all of the Documents used by the organization in conducting its business. The collection of these documents also enables the designer to gain an understanding of the organization and its operation. This understanding will be crucial to the successful design of the data base.

Information concerning these User Documents is entered into the design data base where it serves as the beginning of a data dictionary. For each User Document, the following information is entered into the design data base:

User Document Name

Type (Input, Resident, Output)

Description of Purpose of User Document

List of Organizations which receive or generate User Document

Frequency of generation of User Document

The Data Items appearing on each User Document are also entered. The following information is entered for each Data Item:

Data Item Name

Data Item Description

Value Set (Numeric, Alpha, Dollars, Dates, etc.)

Range of Values

Real or Virtual

Usability and Security Constraints

7.2 GENERATION OF DOCUMENT E-R DIAGRAM BY SYSTEM

The system can now automatically generate a Document E-R Diagram as shown in Figure 2.a. The type of each Document (Input, Resident, Output) is displayed with the Document Entity. A Data Item List is displayed for each Document Entity. This is a list of all of the Data Item Names entered for the User Document. A Common Data Item List is displayed with each Relationship. The Common Data Item List includes Data Item Names common to the two Document Entities.

There is also other information associated with each User Document as described in Section 7.1. This information can be requested by positioning the cursor in the Document Entity Box and pressing a special Function Key. This information will be displayed in a separate window. Further information concerning Data Item Names can be displayed by positioning the cursor on the Data Item Name and pressing a special Function Key.

7.3 DATA ITEM NAME ANALYSIS

The designer can now analyze the assignment of Data Item Names through interaction with the system. The goal of this analysis is to associate consistent Data Item Names with Data Items.

Since the system is aware of the type of the User Document (Input, Resident, Output), the designer can request that the system highlight any

Data Item which meets one of the criteria listed below. The highlighting can be implemented by flashing characters, reverse video, or contrasting colors.

1) A Data Item on an Output Document which has no precedent on an Input or Resident Document. This Data Item must be a derived Data Item.
2) A Data Item on an Input Document which has no antecedent on a Resident or Output Document. This might be an Input Data Item which is used to calculate a Resident Data Item.
3) A Data Item on a Resident Document which does not occur on an Input or Output Document. This might be a Resident Data Item calculated from a different Input Data Item. The Resident Data Item might then be used as the basis for a derived Output Data Item.

If one of the three conditions described above is highlighted by the system, the designer can decide if there is a reasonable cause for the condition.

If there is no reasonable cause, there is probably a discrepancy in Data Item Names which is causing the problem. Data Item Names can be modified interactively in the Data Dictionary to create consistent Data Item Names.

The designer can also analyze the Relationships between Document Entities for Data Item Names which are the same but which represent different Data Items. Alias Data Item Names can be interactively created in the Data Dictionary to alleviate this problem.

The designer can also request that the system generate a Document E-R Diagram which uses Value Set Names for generating Relationships instead of using Data Item Names. This will allow the designer to look for Data Item Names which are different but which represent the same Data Items. The Data Dictionary can be modified interactively to solve this problem.

7.4 DESCRIPTION OF RELATIONSHIPS

The designer is then requested by the system to enter a phrase to describe the Relationships among User Documents. These descriptions will serve later as documentation. They also force the designer to "think through" the Relationships and rationalize their existence.

7.5 ESTABLISHMENT OF DATA ITEM OWNERSHIP

At this point the system will have aided the data base designer in eliminating ambiguous and duplicated Data Item Names. The system will save the Document Entity-Relationship Diagram as it is a graphical representation of the relationships among the users Documents. It can graphically display information concerning each Document. It also shows the flow of data through the system. Data Item Names can be traced through Input, Resident, and Output Documents. Furthermore, the Document Entity-Relationship Diagram can indicate which Documents contain common Data Items.

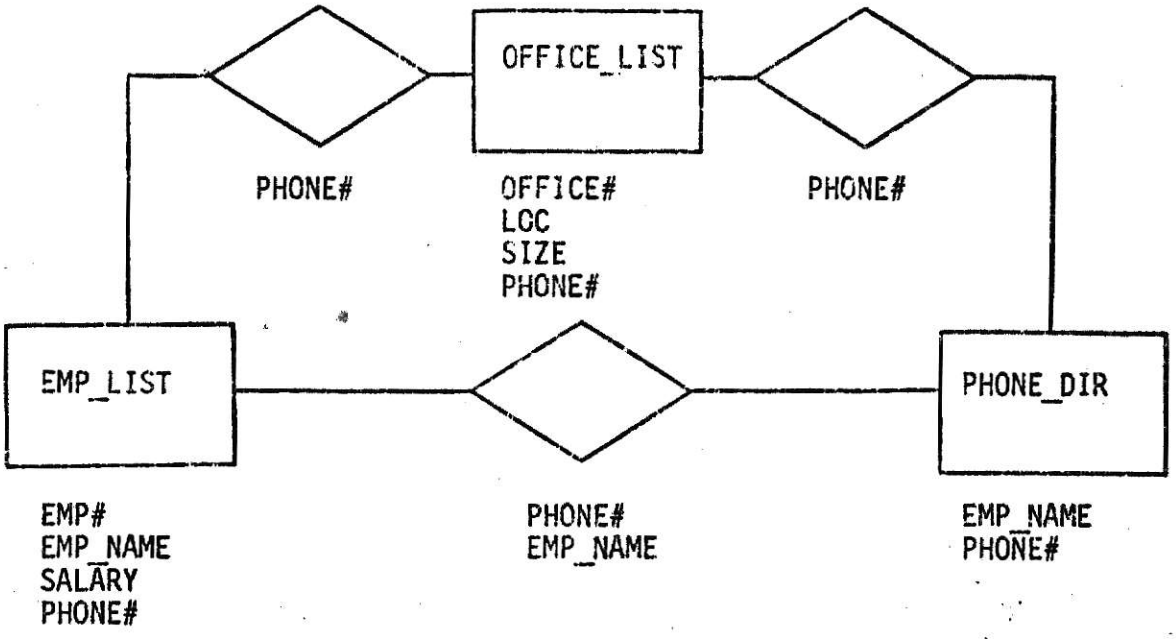
The methodology described in Section 2 requires that the designer define the Entities and Relationships based on his understanding of the organization. This understanding will be based on his interviews with organization

personnel and his analysis of the organization's Documents. The Documents will indicate to him which Data Items are associated with other Data Items. The designer must remember all of these relationships among Data Items to insure that the resulting Entities and Relationships can be utilized to generate Output Documents or receive data from Input Documents. This is a massive amount of information for the designer to remember and mentally manipulate.

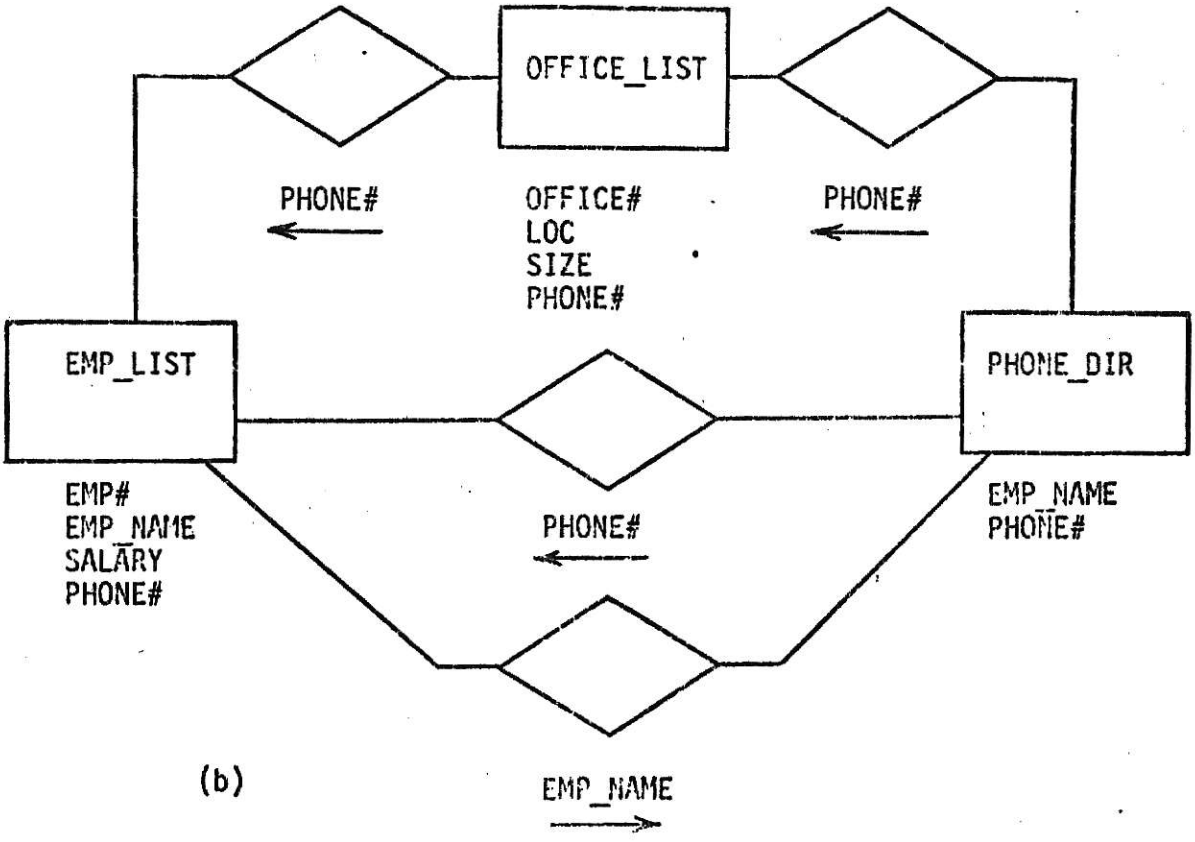
Much of this information, however, is already contained within the system. The missing information is the semantic information which fully describes the Data Items and their meaning within the Documents and within the organization. Some of this information can be entered into the system so that the system can assist the designer.

First, the designer must indicate the ownership of any Data Item Names which are common to any two Document Entities. This is accomplished by requesting that the designer analyze the Common Data Item List accompanying each Relationship and indicate which Document can be described as the owner of the Data Items. The system graphically displays this indicated ownership with an Arrow under the Common Data Item List which points away from the owner Document Entity.

If both Document Entities own some of the Data Items in the Common Data List, two Relationships are created between the two Document Entities. Each Relationship has its own Common Data Item List. For example, in Figure 3.a EMP_LIST, OFFICE_LIST, and PHONE_DIR are three different Documents with Data Item Lists as shown. If the designer indicates that PHONE # is owned by PHONE_DIREC and EMP_NAME is owned by EMP_LIST, the Relationship will be split and the Document E-R Diagram in Figure 3.b will be generated.



(a)



(b)

FIGURE 3. Data Item Ownership Example

Also, note in Figure 3.b that the designer has indicated that PHONE # in the Common Data Item List for the Relationship between EMP_LIST and OFFICE_LIST is owned by OFFICE_LIST. There is no strong rationale for this decision and EMP_LIST might have just as easily been selected as the owner. A decision such as this which can go either way will make little difference when the final design is completed.

7.6 DELETIONS FROM DOCUMENT ENTITY DATA ITEM LISTS

After the designer has indicated the ownership of Data Items, the system can eliminate Data Items in a Document Entity's Data Item Name List which are not owned by the Document Entity. If a Data Item is eliminated from a Data Item List, it must also be eliminated from any other Common Data Item Lists in which it appears. For the example in Figure 3.b this means that PHONE # is deleted from the OFFICE_LIST Data Item List. Therefore, PHONE # is also deleted from the Common Data Item List associated with the Relationship between OFFICE_LIST and EMP_LIST. These deletions are shown in Figure 4.

7.7 ELIMINATION OF DOCUMENT ENTITIES AND RELATIONSHIPS

The system can now automatically eliminate any Document Entity which has an empty Data Item List. Any Relationship with an empty Common Data List is also eliminated. The remaining Document Entities and Relationships represent the actual Entities and Relationships of the organization. For example, in Figure 4 the Relationship betwen OFFICE_LIST and EMP_LIST is eliminated creating the Entity-Relationship Diagram shown in Figure 5. Note that in Figure 5 there are no Common Data Lists or Ownership Arrows.

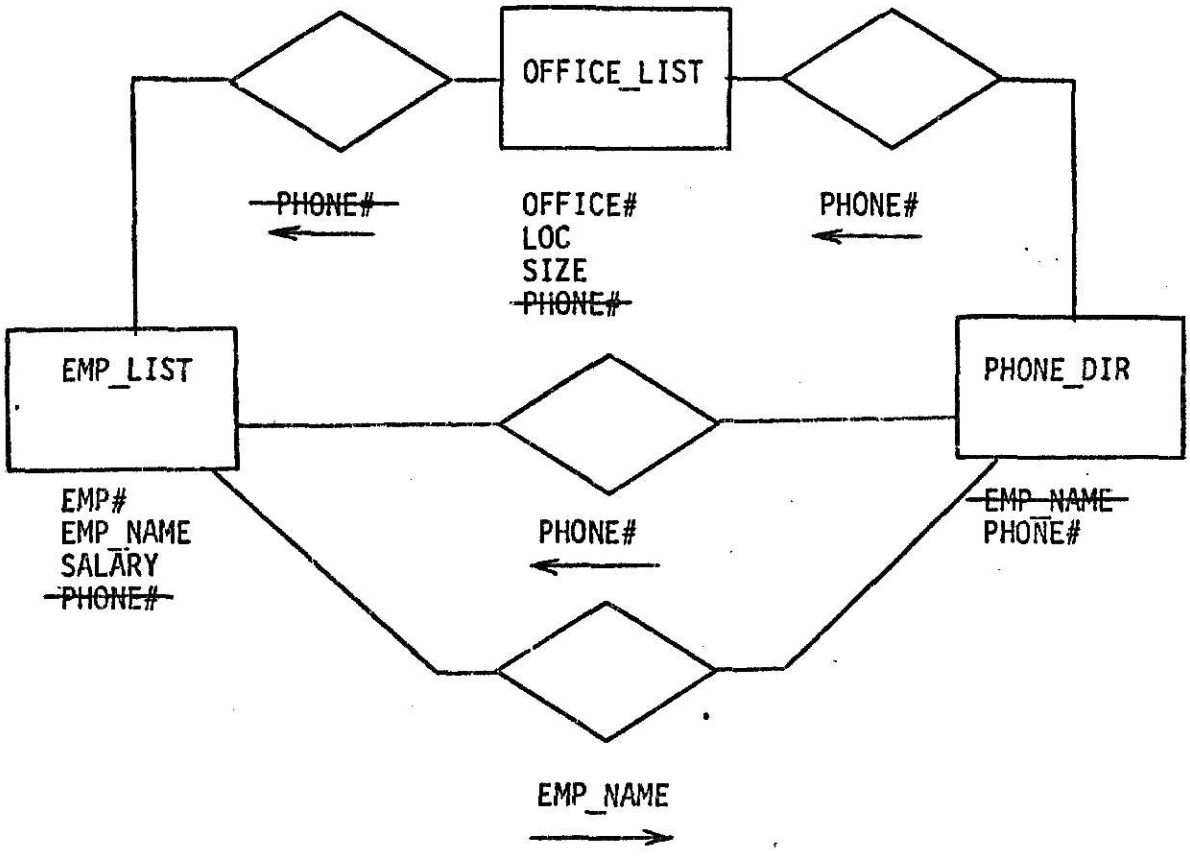


FIGURE 4. Data Item Deletion Example

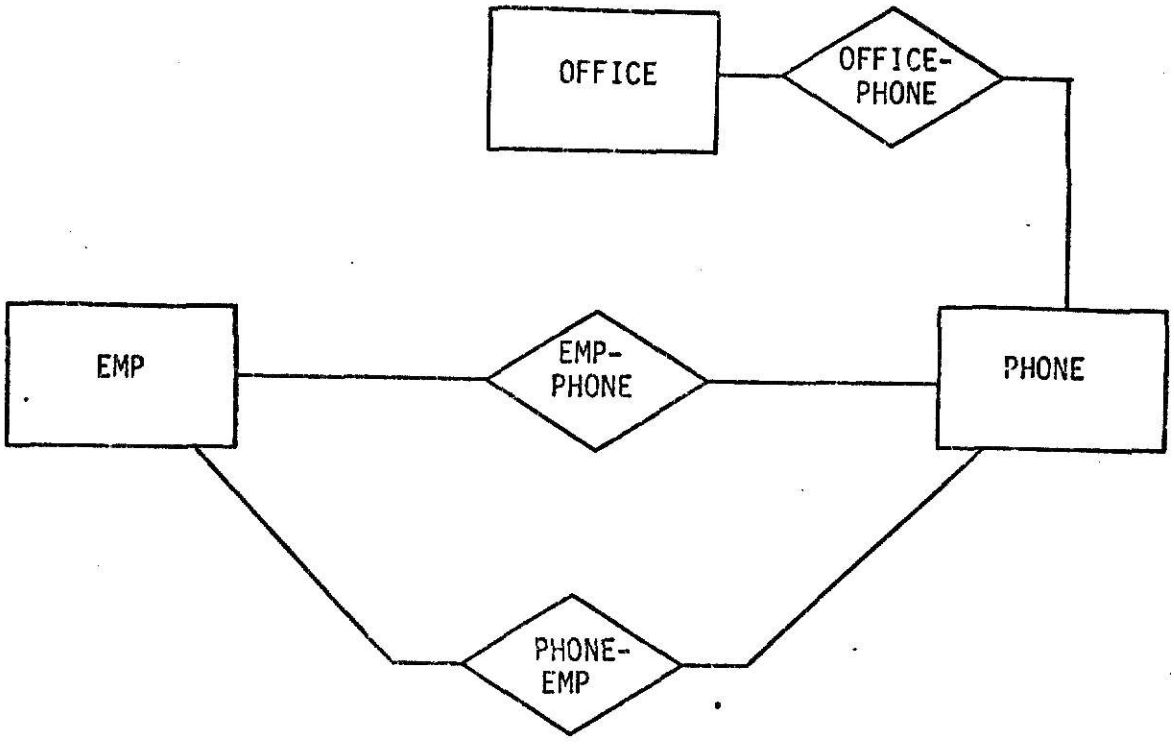


FIGURE 5. Elimination of Document Entities and Relationships

Also, note in Figure 5 that there are two Relationship Sets between EMP and PHONE. These can be merged with the MERGE operation which will be explained in Section 7.8

7.8 MANIPULATION OF ENTITIES AND RELATIONSHIPS

The system has now created an Entity-Relationship Diagram which represents the organization's data and which will support the utilization of the users Documents. This may not, however, be the optimal data organization. The data base designer may recognize semantic implications which simply were not reflected in the Documents. These semantic problems may have been discovered by the designer through discussions with the organization's personnel. The system must provide the designer with tools to manipulate the E-R Diagram without allowing the designer to inadvertently destroy semantic or syntatic information already contained in the displayed E-R Diagram.

Chen suggested five basic types of operations for the manipulation of Entity-Relationship Diagrams (ADD, DELETE, SPLIT, MERGE and SHIFT). [10] The operations are explained below. The system proposed here will only use SPLIT, MERGE, and SHIFT. Two new operations, MOVE and CHANGE will also be implemented. These new operations will be defined later.

7.8.1 CHEN'S E-R DIAGRAM OPERATIONS

The first four operations ADD, DELETE, SPLIT, and MERGE are applicable to Entity Sets, Relationship Sets, Attributes, and Value Sets. The ADD operation is used to add an Entity Set, Relationship Set, Attribute or Value Set to an existing E-R Diagram schema. A Regular Entity Set can be added without linking it to any other Entity Sets. The addition of a Weak Entity

Set requires that the Weak Relationship Set also be defined at the same time. Value Sets can be added without linking them to an Entity Set. An Attribute can be added but it must be linked to an Entity Set and a Value Set.

The DELETE operation is used to delete an Entity Set, Relationship Set, Attribute, or Value Set from an E-R Diagram schema. If an Entity Set is deleted, all Relationship Sets related to the deleted Entity Set are also deleted. Furthermore, all Attributes related to the deleted Entity Sets and Relationship Sets are also deleted. If a Relationship Set is deleted with a DELETE operation, all of its related Attributes are also deleted.

The SPLIT operation is used to split Entity Sets, Relationship Sets, Attributes, and Value Sets into subsets. For example, the Entity Set EMP in Figure 6.a can be split into two Entity Sets. MALE_EMP and FEMALE_EMP in Figure 6.b.

The MERGE operation is the opposite of the SPLIT operation. For example, the Entity Set EMP in Figure 6.a can be obtained by merging the MALE_EMP and FEMALE_EMP Entity Sets shown in Figure 6.b.

The SHIFT operation is used to shift an Entity Set into a Value Set or to shift a Value Set into an Entity Set. This shifting may be necessary as a result of a change in the organization or merely a change in the way the organization is viewed. For example, in Figure 7.a the PLACE where an employee resides is thought of as an Attribute of the EMP Entity Set. Chen uses circles to denote Value Sets such as NAMES_OF_PLACES. In Figure 7.b, it has been decided that PLACE should be thought of as an Entity Set. Therefore, ADDRESS has become a Relationship Set instead of an Attribute. The new Entity Set PLACE has an Attribute called NAME which points to the Value Set NAMES_OF_PLACES.

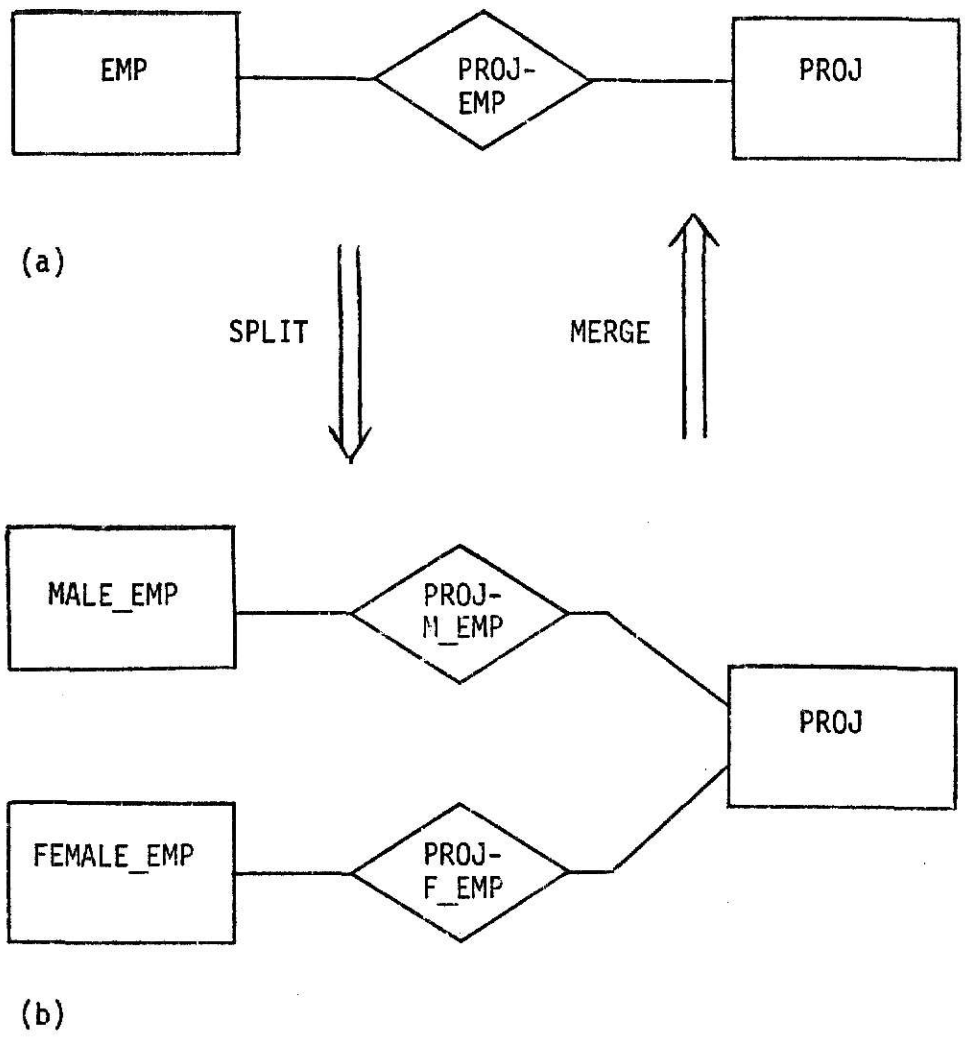


FIGURE 6. Example of SPLIT/MERGE Operation on an Entity Set

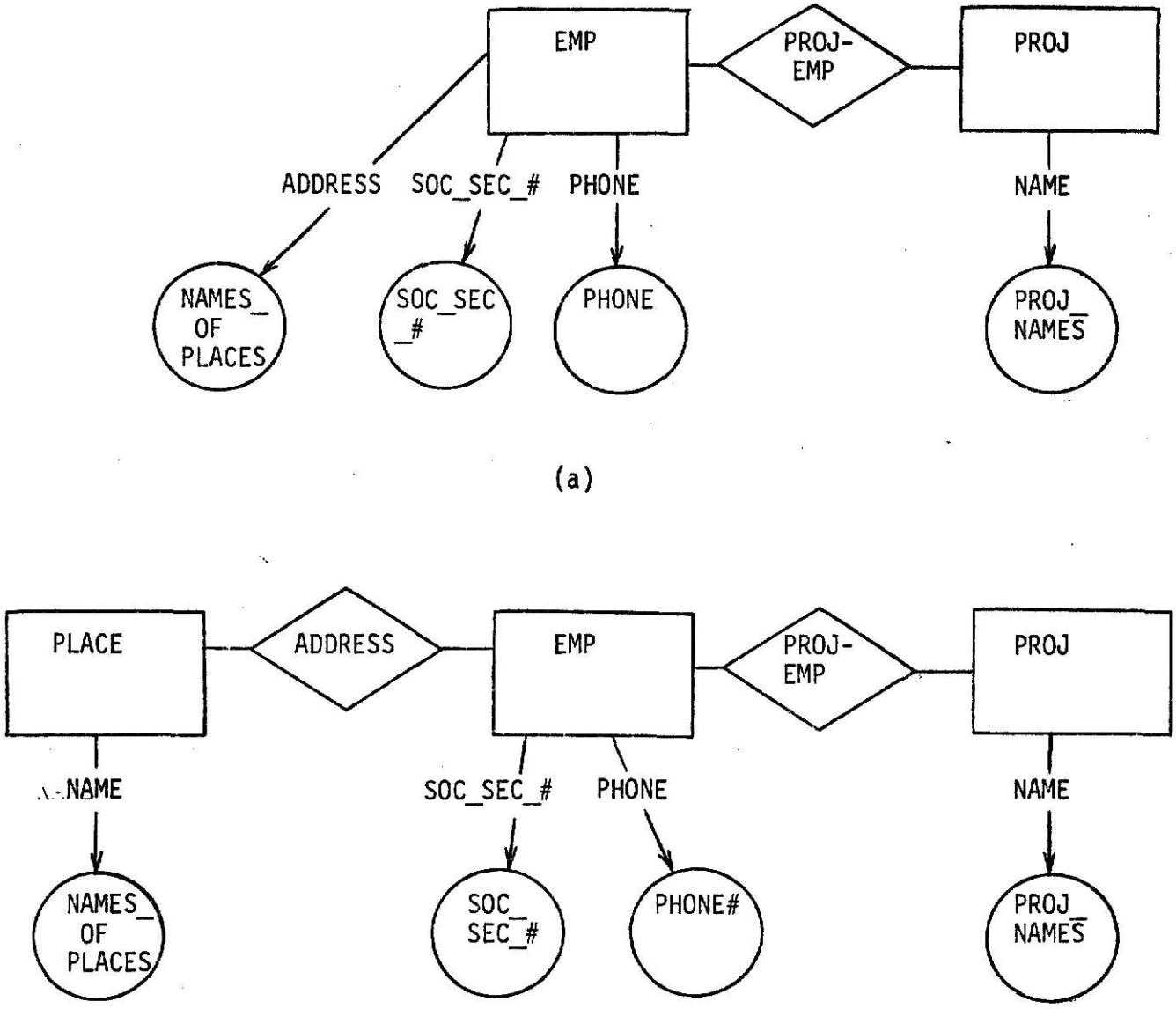


FIGURE 7. Example of SHIFT Operation

7.8.2 PROPOSED E-R DIAGRAM OPERATIONS

In this proposed system the organization's Documents are utilized for defining Entity Sets, Relationship Sets, Attributes, and Value Sets. Therefore, there is no need for the ADD operation. It might be implemented, however, to generalize the functional capabilities of the system. The DELETE operation might also be implemented for completeness but it's use can be very dangerous in this design environment. The deletion of Relationship Sets can easily cause the loss of navigational paths necessary for the generation of Documents.

The SPLIT, MERGE, and SHIFT operations are all necessary and compatible with the proposed system. Special thought must be given to the implementation of these commands, however, since the designer must always interface simultaneously with both the Entity Sets and the Value Sets. For example, in Figure 6.a the Entity Set EMP can be split into MALE_EMP and FEMALE_EMP Entity Sets. However, since Attributes and Value Sets are already associated with the MALE Entity Set the designer must indicate how they are to be duplicated or distributed between the two new Entity Sets.

The system must also prevent the designer from violating any existing relationships through the use of the SHIFT operation. For example, if the PLACE Entity Set in Figure 7.b was also linked to another Entity Set through a Relationship Set, the system should not allow the PLACE Entity Set to be shifted back to its role as a Value Set in Figure 7.a.

There are two new operations which must be added to complete the list of necessary operations. These new operations would normally be carried out with a series of ADD's and DELETE's. The specification of the new operators, MOVE and CHANGE, provides the needed specific functions without utilizing the more dangeous generalized ADD and DELETE operations.

The MOVE operation moves an Attribute and its Value Set from an Entity Set to an associated Relationship Set. This operation is necessary since the first true E-R Diagram which the system generates always has no Attributes associated with the Relationship Sets. For example, the system might generate the E-R Diagram shown in Figure 8.a. If the designer decides that QUANTITY is the quantity of parts for a given project, he can MOVE the QUANTITY Attribute to the PARTS-PROJ Relationship Set as shown in Figure 8.b.

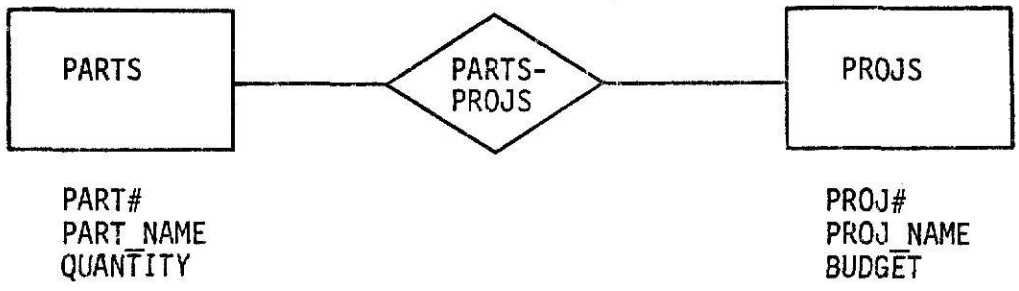
The other new operation needed is the CHANGE operation. This operation changes an Entity Set into a Relationship Set. All of the Attributes of the Entity Set become Attributes of the Relationship Set. This operation is necessary since a Relationship Set with Attributes cannot be generated automatically by the system. Figure 9.a presents an E-R Diagram similar to the one in Figure 8.a. This time, however, the system has generated an Entity Set PARTS_STATUS with one Attribute, Quantity. The designer can use the CHANGE operation to change the Entity Set PARTS_STATUS into the Relationship Set shown in Figure 9.b.

7.9 SPECIFICATION OF MAPPING

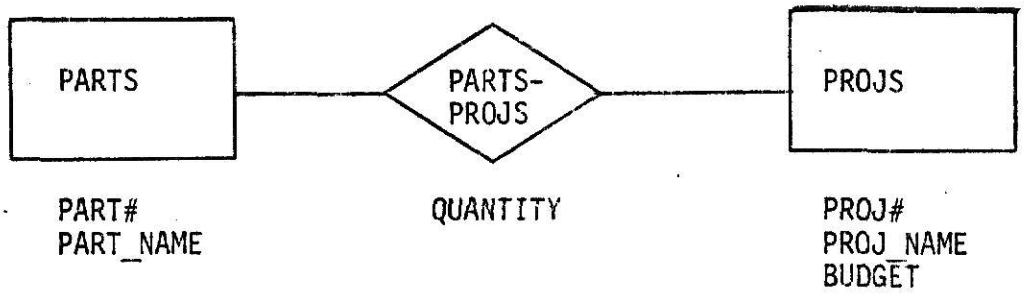
The designer must now specify the mapping of the Relationships established between Entities. This mapping can be 1:N, N:1, or N:M.

7.10 IDENTIFICATION OF KEY ATTRIBUTES

The designer now identifies the Key Attributes for each Entity Set. The Key Attributes for the Relationship Sets are the concatenation of the Key Attributes of the associated Entity Sets. This process of identifying keys will probably lead the designer into further manipulations of the E-R Diagram as described in Section 7.8.

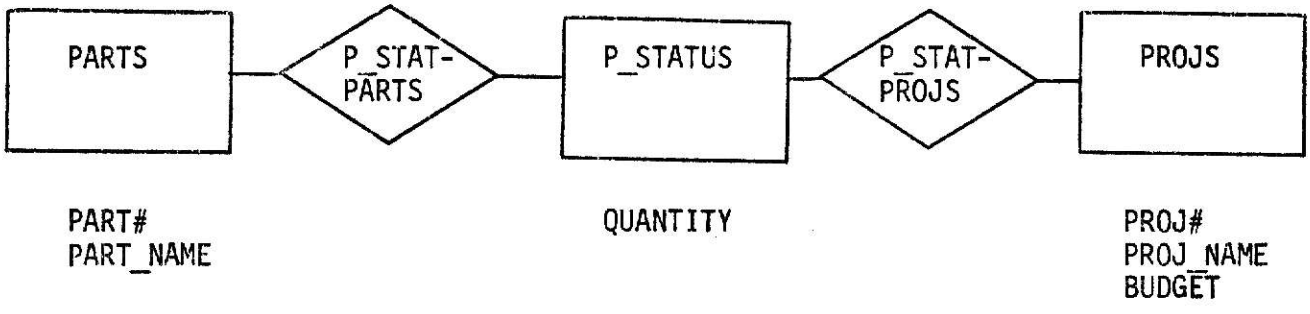


(a)

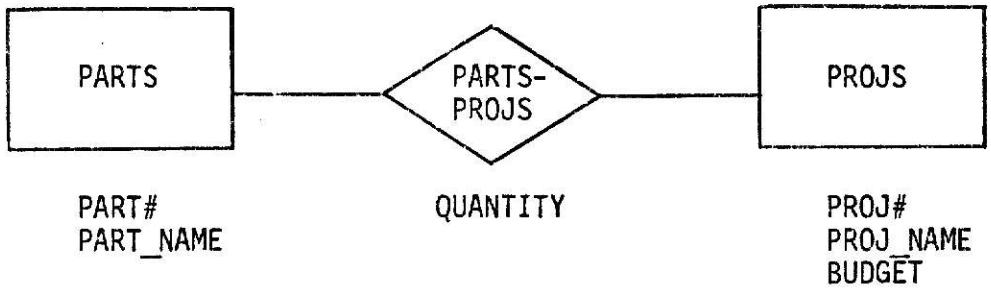


(b)

FIGURE 8. Example of MOVE Operation



(a)



(b)

FIGURE 9. Example of CHANGE Operation

8.0 EXAMPLE DESIGN

This Section will present a complete design following the design steps outlined in Section 7. The goal is to design a system to keep records for maintenance of a fleet of buses. This system was actually implemented using IMS and is currently in use. The following description of the system is taken from a paper by E. L. Lusk. [11]

The essential characteristics of the system are as follows:

1. A company that ran buses for a major city had about ten major garages, each of which managed about 100 buses. Each bus was assigned to one of the garages, but could occasionally be lent to another garage.
2. Some types of work, such as changing the oil, had to be performed periodically on each bus. Other types of repairs would have to be performed when problems were detected by the drivers. Some problems were so severe that a bus would have to be fixed before it could be put back on the street, but others could be deferred for several days if necessary.
3. The data base had to keep track of all work that had been done on a bus in the last month, all deferred repairs that still had to be performed, and all scheduled maintenance (such as oil changes) that would have to be taken care of in the near future.
4. Each garage was staffed for three shifts. The data base would have to keep track of which employees were at work, what jobs they had worked on in the last month, and what jobs they were currently working on. Most of the jobs a person could work on were on a specific bus, but some jobs (such as sweeping up the garage or

washing windows) were not directly related to buses at all. There was a standard amount of time required to complete each job.

5. The jobs which could be performed on a bus were each instances of a general category of job. For example, "air conditioner maintenance" was a job category containing jobs like "install air conditioner" and "replace compressor".
6. Each bus had an identification number. By typing in this number at a terminal, a foreman could examine all of the data stored about the bus. This would include all of the work done on the bus recently, all work scheduled for the bus, and the status of the bus.

Based on the preceding description of the system, the Documents in Appendix A have been hypothesized. A minimum number of Documents are used here in order to limit the graphical representation of the design to a manageable size. Appendix B presents sample screens from various Steps in the designs. The Steps refer to the outlined steps at the beginning of Section 7. Data Item Names which are deleted from Data Item Lists and Common Data Item Lists are shown with lines through them for clarity.

8.1 DISCUSSION OF DOCUMENTS

All of the Documents in Appendix A are Input Documents. No Output Documents or Resident Documents are included so that as much semantic information as possible can be included in this example.

The first Document is the Employee Report. This report lists the employee information by Department Number and Shift (DEPT_NUM and SHIFT). An employee can only work on one Shift but an employee can work in more than one department.

The Department Report lists the department number and description for each department. It also lists the names of all employees in each department. Note that the contents of the first two Documents are similar, but the intent of the Documents is different. The Employee Report is focusing on employees and the Department Report is focusing on Departments.

The Job Report lists the types of jobs and some descriptive information concerning each. Within each job category, the general jobs are described along with their estimated time for completion.

The Vehicle Report lists the vehicles along with descriptive information concerning each. Each vehicle has some association with a department (owned, owned-but-lent, or borrowed). If the vehicle is owned by a department but lent to another department, the DEPT_LENT_TO is also included in the Vehicle Report.

The Maintenance Report lists the maintenance record for each vehicle. For each vehicle, the maintenance record is further broken down into different job status categories (in-progress, scheduled, completed). The severity of the job is also listed (deferrable, non-deferrable).

The Employee Jobs Report lists each employee with their department number and specific jobs on which they have worked. A general job description, GEN_JOB_DESC is included for each specific job worked on by an employee. The estimated time and elapsed time for each specific job are also listed.

The Job Category Frequency Report shows the number of occurrencies of each CATEG_ID during a specified range of dates. A description is included with each CATEG_ID. The START_DATE and END_DATE can be specified each time the report is requested.

8.2 DISCUSSION OF SAMPLE SCREENS

Sample Screen 1 is the Document E-R Diagram generated by the system in Step 2. Note that an implementation of this proposed system would require a fairly intelligent program to position the symbols on the screen. The Data Item Lists are positioned near the Document Entities and the Common Data Item Lists are positioned near the Relationships.

Sample Screen 2 shows the results of Step 3 where the designer has analyzed Data Item Names for duplication and ambiguity with the aid of the system. ELAPSED_TIME, TOTAL_HOURS, and NUM_OF_OCCUR have been identified by the designer as Derived Data Items. START_DATE and END_DATE from the JOB CATEG FREQ REPORT have both been identified by the designer as having the same meaning as the DATE_ENTERED Data Item in the MAINTENANCE REPORT.

Sample Screen 3 shows the screen after Step 5. The designer has entered a description of each Relationship. The designer has also indicated the ownership of each of the Common Data Items. For example, the designer has decided that the DEPT_NUM Data Item is owned by the EMPLOYEE REPORT when the EMPLOYEE REPORT and the EMP_JOBS REPORT share it as a Common Data Item. However, when EMPLOYEE REPORT shares DEPT_NUM with DEPT REPORT, DEPT REPORT is declared the owner.

Sample Screen 4 shows the results of Step 6 where the system has made deletions from Document Entity Data Item Lists and Relationship Common Data Item Name Lists. For example, the Document Entity EMPLOYEE REPORT does not own the Data Item DEPT_NUM. Therefore, DEPT_NUM is deleted from it's Data Item List. DEPT_NUM is also deleted from the Relationship Common Data Item List which EMPLOYEE REPORT shares with EMP_JOBS REPORT.

Sample Screen 5 shows the results after Step 8 where the system has eliminated Document Entities and Relationships from the Diagram.

A number of Document Entities and Relationships were elimintated. For example, the Document Entity JOB CATEG FREQ REPORT was eliminated from Sample Screen 4 since it had no entries left in its Data Item List. Sample Screen 5 shows true Entity Sets and Relationship Sets.

Sample Screen 6 shows the final E-R Diagram generated by the system after Step 10 of the algorithm. The designer has moved the Data Items RELATION_CODE and DEPT_LENT_TO from the VEHICLE Entity Set to the VEHICLE/DEPT Relationship Set with the MOVE operation because they are dependent on both DEPT_NUM and VEH_ID_NUM. The designer has changed the EMP_JOBS Entity Set to a Relationship Set because EMP_JOBS was not really an Entity. The designer has used the SHIFT operation to shift the idea of job category from a Value Set to an Entity Set. There is now an Entity Set called JOB_CATEG which has the Attributes CATEG_ID and CATEG_DESC.

8.3 DISCUSSION OF RESULTS

For this example, the Entity Sets represented in the Entity-Relationship Diagram are very similar to some of the original Documents. This is not unexpected since the Documents in an organization are meant to be used for dispersing or gathering information concerning the Entities and Relationships in the organization. In general, one might expect that there will be more Documents in the organization than there are Entity Sets. Therefore, the algorithm described in this paper can be characterized as one which systematically eliminates Document Entities and Relationships while maintaining the semantic information contained in them. The Document Entities and Relationships which are not eliminated retain this information and represent a possible data model for the organization.

Another general comment concerning this example and the algorithm in general is that there is no direct discussion of functional dependencies during the design process. The designer provides information on functional dependencies as he makes decisions concerning the ownership of Data Items. This type of information is provided by the designer again as mapping (1:N, N:1, N:M) is assigned to Relationship Sets. Finally, the designer provides more information on when primary keys are assigned to Entity Sets.

In fact, Chen claims that the semantics of functional dependencies among data in the Entity-Relationship Model is more easily understood than in the Relational Model. He defines two major types of functional dependencies [12]:

1. Functional dependencies related to description of Entities and Relationships. For example, the Non Key Value Sets associated with an Entity Set will be functionally dependent on the Key Value Sets.
2. Functional dependencies related to Entities in a Relationship. For example, suppose there is a 1:N mapping between DEPTS and EMPLOYEES. An employee can only be a member of one department. Therefore, in the Relationship Set DEPTS_EMPLOYEES, the Value Set DEPT_NO will be functionally dependent on the Value Set EMPLOYEE_NO.

9.0 CONCLUSIONS

The system proposed in this paper is intended as an aid to the data base designer who is utilizing the Entity-Relationship Model as a basis for his design. The Entity-Relationship Model has been proven over the past few years to be a step forward in the area of data base organization and design. It allows the designer to model the data at a logical level without being concerned with the low level structure of the data. This provides a basis for a unified view of data which can then be mapped into alternate low level structures.

Research is continuing in the area of extensions to the Entity-Relationship Model. There is also a great amount of investigation of techniques for utilization of the Entity-Relationship Model in actual data base design. The system described in this paper is an attempt to supply the designer with an automated tool for data base design. The basic premise of the system is that the user's Documents contain information concerning the inherent structure of the data. The goal of the system is assist the designer in identifying Entities and Relationships. The Documents become the basis for the designer's interaction with the system. The system requests information from the designer by asking questions related to the Documents using terminology consistent with the Documents.

One of the major problems in using this "Document approach" is the problem of alias names for Data Items. With a great number of diverse Documents there will be a problem with associating diverse names with the correct Data Items. While this association of names and Data Items must be accomplished eventually, it can be argued that forcing the

designer to complete this task at the onset of the design effort may create frustration and confusion. In a way, it is a contradiction of the underlying idea of the Entity-Relationship Model. This underlying idea is the designer should begin at a high level of abstraction thinking only in terms of Entities and Relationships.

Other major problems may be introduced if the user has a great number of Documents. First of all, this may create a problem of lack of comprehension by the designer. For example, consider a system with 20 Documents which are similar with only slight differences in data content. The initial Document E-R Diagram would be very complex. Its graphical representation might prove impossible to generate. If successfully generated, the Document E-R Diagram might be incomprehensible. The establishment of ownership of Common Data Items would also be confusing for 20 Documents which are very similar since there would be no obvious owner of the Common Data Items.

Another problem concerns the first Entity-Relationship Diagram which the system generates after Step 7. This E-R Diagram represents the data organization but it may also contain some Entity Sets which are confusing. For example, the Entity Set EMP_JOBS in Sample Screen 5 has been created by the system. It is difficult to rationalize the actual existence of an Entity which is "an employee doing a specific job". The system, however, has no basis for making this judgement and creates an Entity Set. The designer must then use his judgement to change EMP_JOBS to a Relationship Set using the CHANGE operation.

Further investigation is needed in these problem areas before a successful implementation of this system can be carried out. The ideas

presented in this paper should provide a framework for that investigation. The concept of using a graphical representation aid to assist the designer still appears to be a sound one. Sufficient abstraction is necessary, however, to prevent the designer from being overwhelmed by the amount of information presented.

REFERENCES

1. Chen, P.S., "The Entity-Relationship Model: Toward a Unified View of Data", ACM Transactions on Database Systems 1, March 1976, p.10.

2. Chan,E.P.F. and Lochovsky,F.H., "A Graphical Data Base Design Aid Using the Entity -Relationship Model", Proc. of the Intl. Conf. on Entity-Reletionship Approach to Systems Analysis and Design, 1979, p.295

3. Hankley,W. and Maryanski,F. "Request for Funds on Interactive Techniques for Design of Data Bases", NSF Grant Proposal. Kansas State University, 1977.

4. Maryanski,F.J., Buser,R.H., Hoflich,M.A., Hunt,W.C., Kusnyer, S.K., and Stevens, T.J. "Automatic Generation of Third Normal Form Relations". Technical Report Number CS 77-21, Computer Science Dept., Kansas State University, Manhatten, Kansas,1977, pp. 1-20.

5. Hollist,J.Penton and Fisher,Paul.S., "An Integrated Data Base Design Methodology".

6. Chen, P.S. p.10.

7. Chen, P.S., "Entity-Relationship Diagrams and English Sentence Structures", Proc. of the Intl. Conf. on Entity-Relationship Approach to Systems Analysis and Design,1979,Abstract only.

8. Fisher,Paul.S, "A New Approach to Data Base Design", Unpublished paper. Kansas State University.

9. Chan,E.P.F., p.295.

10. Chen, P.S., "The Entity-Relationship Model: A Basis for the Enterprise View of Data", Proc AFIPS 1977 NCC, Vol. 46, AFIPS Press, N.J., pp.80-83.

11. Lusk,E.L., Overbeek,R.A. and Parello,B., "A Practical Design Methodology for the Implementation of IMS Databases Using the Entity-Relationship Model", Proc. of ACM_SIGMOD, 1980, p.13.

12. Chen, P.S., "The Entity-Relationship Model: Toward a Unified View of Data", p.28.

BIBLIOGRAPHY

1. Chan,E.P.F. and Lochovsky,F.H., "A Graphical Data Base Design Aid Using the Entity-Relationship Model", Proc. of the Intl. Conf. on Entity-Relationship Approach to Systems Analysis and Design, pp.295-310.

2. Chen,P.S., "The Entity-Relationship Model: Toward a Unified View of Data", ACM Transacions on Database Systems 1, March 1976, pp.9-36.

3. Chen,P.S., "The Entity-Relationship Model: A Basis for the Enterprise View of Data", Proc. AFIPS 1977 NCC, Vol. 46, AFIPS Press, N.J., pp.77-84.

4. Chen,P.S., "Entity-Relationship Diagrams and English Sentence Structures", Proc. of the Intl. Conf. on Entity-Relationship Approach to Systems Analysis and Design,1979.

5. Fisher,Paul.S., "A New Approach to Data Base Design", Unpublished paper. Kansas State University.

6. Hankley,W. and Maryanski,F., "Request for Funds on Interactive Techniques for Design of Data Bases", NSF Grant Proposal. Kansas State University,1977.

7. Hollist,J.Penton and Fisher,Paul.S., "An Integrated Data Base Design Methodology".

8. Lusk,E.L., Overbeek,R.A. and Parello,B., "A Practical Design Methodology for the Implementation of IMS Databases Using the Entity-Relationship Model", Proc. of ACM SIGMOD, 1980, pp. 9-21.

9. Maryanski,F.J., Buser,R.H., Hoflich,M.A., Hunt,W.C., Kusnyer S.K., and Stevens,T.J., "Automatic Generation of Third Normal Form Relations". Technical Report Number 77-21, Computer Science Dept., Kansas State University, Manhatten, Kansas,1977, pp.1-20.

APPENDIX A

BUS MAINTENANCE DATA BASE

USER DOCUMENTS

EMPLOYEE REPORT

DEPT-NUM

5

SHIFT

1

SSN	NAME	ADDRESS
514-48-8692	B. Smith	14 N. Main, Fargo, N.D.

DEPARTMENT REPORT

DEPT-NUM	DEPT-DESC
1	Maintenance

NAME

B. Smith
P. Jones

JOB REPORT

CATEG-ID	CATEG-DESC
1	Air Conditioner Maintenance

GEN-JOB-ID	GEN-JOB-DESC	EST-TIME
52	Replace Compressor	.5
53	Add Freon	.5

VEHICLE REPORT

VEH-ID-NUM	TYPE OF VEH	DEPT-NUM	RELATIONSHIP-CODE	DEPT-LENT-TO
141	Bus	5	Owned	--
142	Truck	5	Owned-But-Lent	4

MAINTENANCE REPORT

VEH-ID-NUM

141

STATUS

Completed

DATE ENTERED	SPEC-JOB-ID	CATEG-ID	CATEG-DESC	GEN-JOB DESC	SEVERITY
1/8/81	1201	1	Air Cond. Maint.	Replace Compressor	Non-Deferrable

EMPLOYEE JOBS REPORT

SSN	NAME	DEPT-NUM
514-47-8662	B. Smith	5

SPEC-JOB-ID	GEN-JOB-DESC	DATE ENTERED	EST TIME	START TIME	STOP TIME	ELAPSED TIME
1201	Replace Comp.	1/8/81	.5	0807	0907	1.0
1314	Replace Comp.	1/10/81	.5	1015	1045	.5
				TOTAL-HOURS		1.5

JOB CATEGORY FREQUENCY REPORT

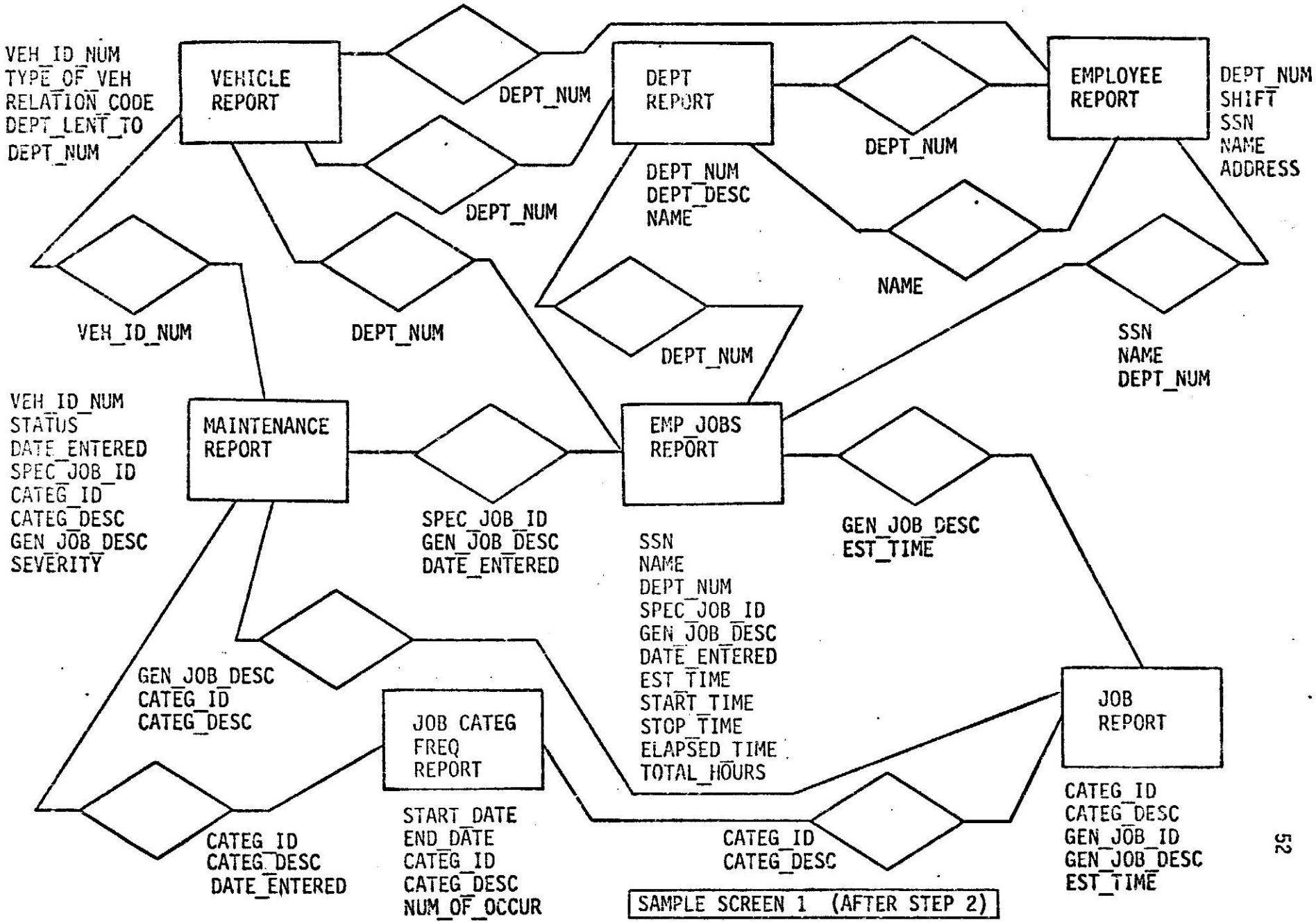
START DATE	END DATE
1/1/81	3/1/81

CATEG ID	CATEG DESC	NUMBER OF OCCURENCES
1	Air Conditioner Maint.	23

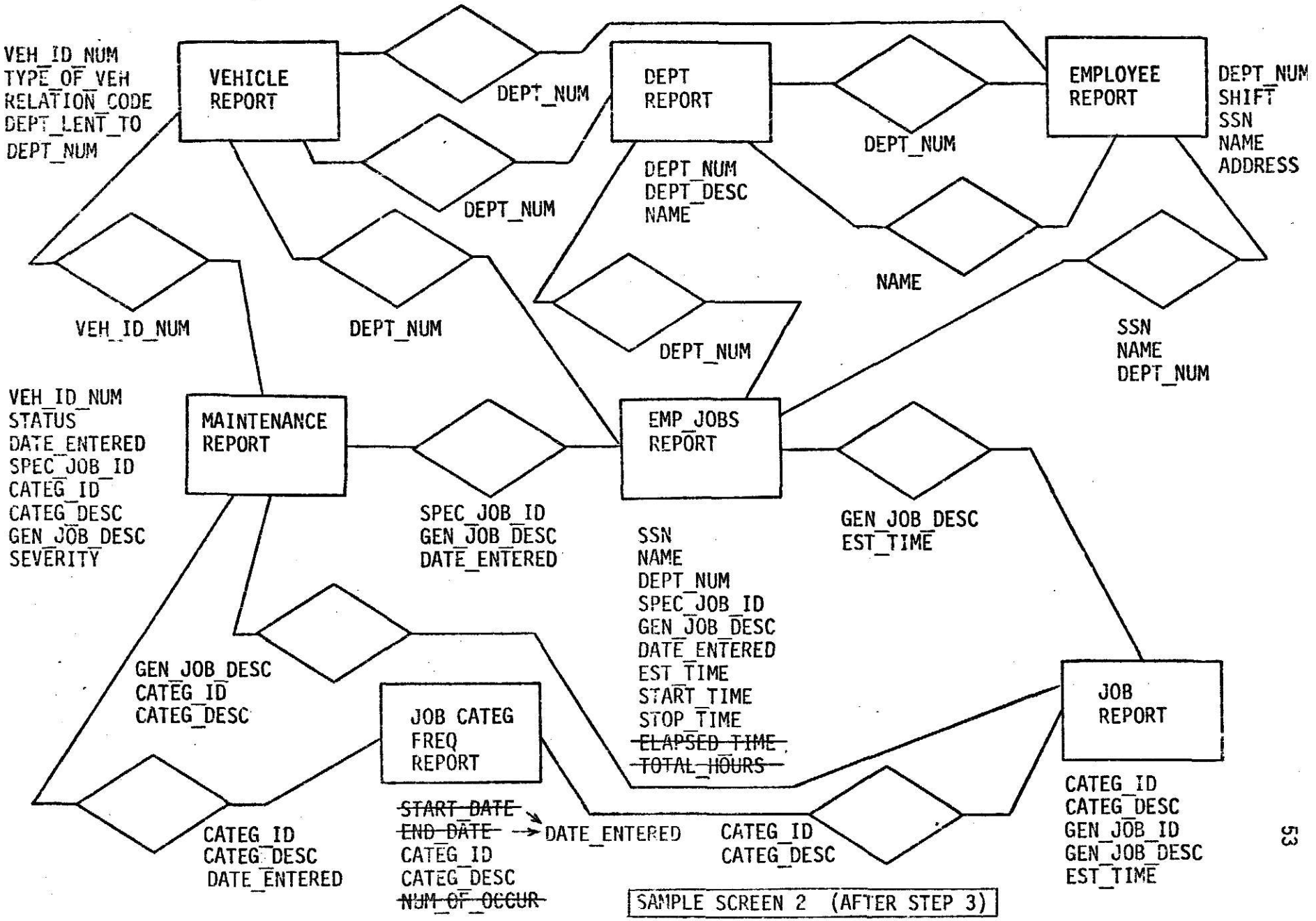
APPENDIX B

BUS MAINTENANCE DATA BASE

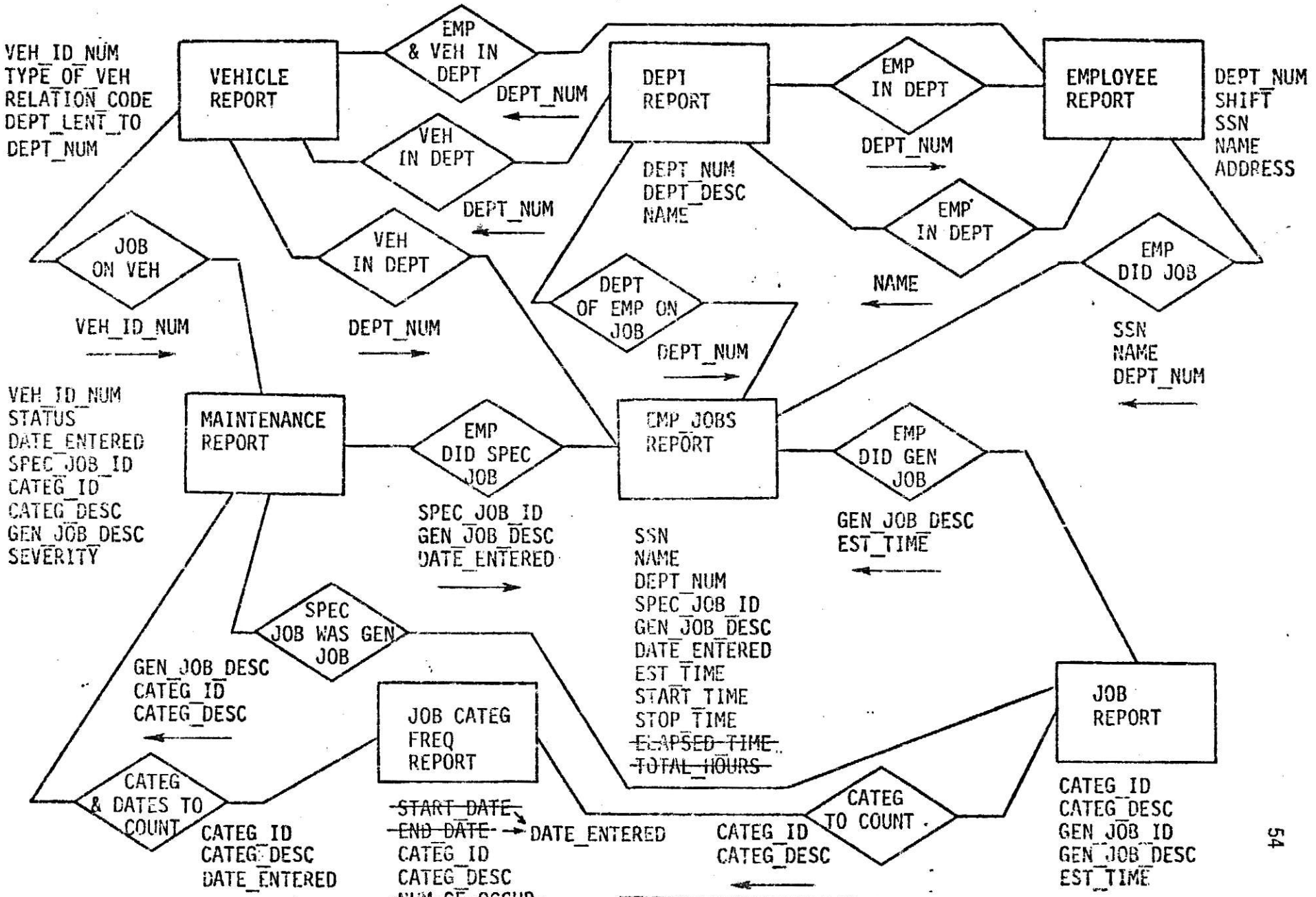
SAMPLE SCREENS



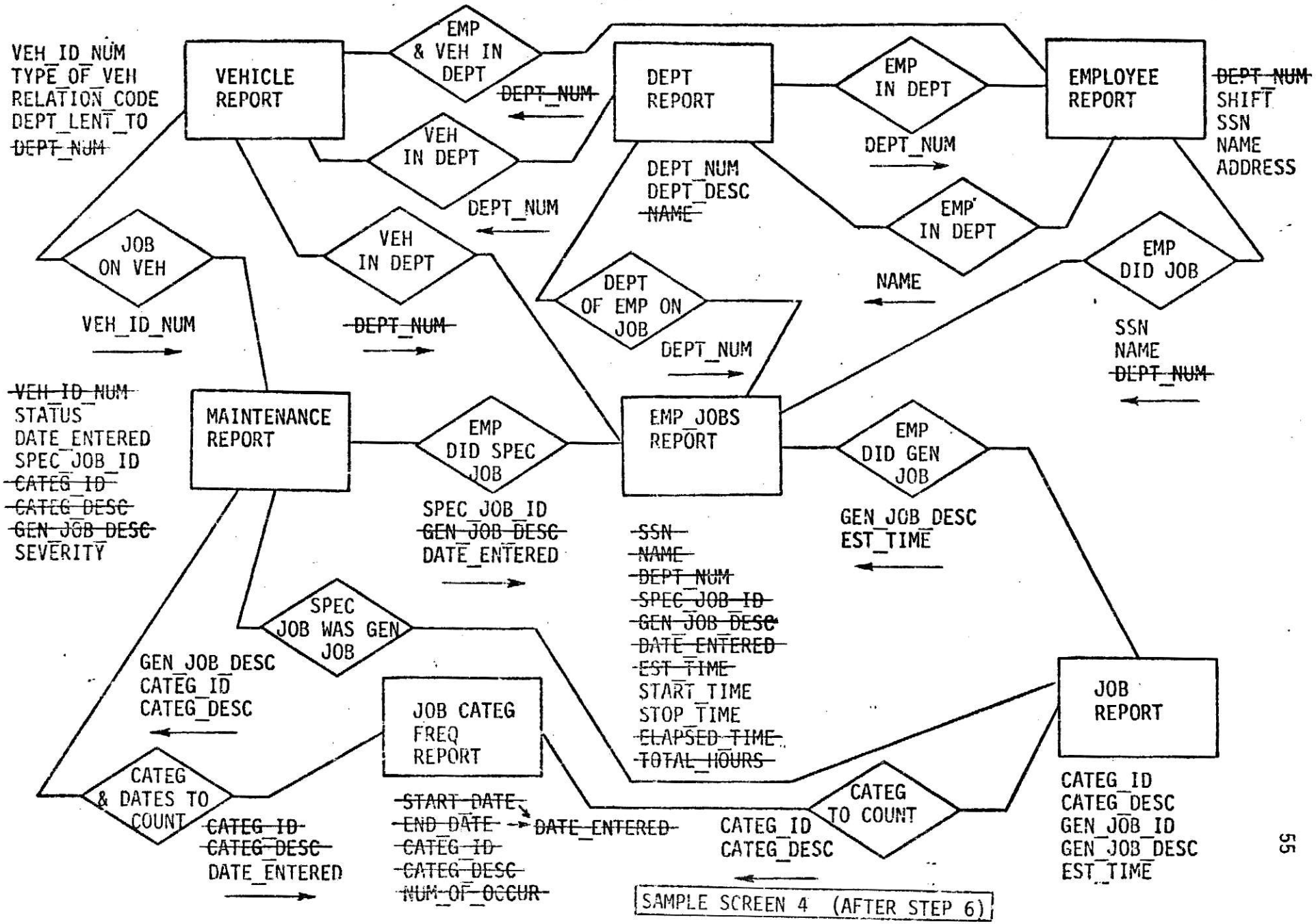
SAMPLE SCREEN 1 (AFTER STEP 2)



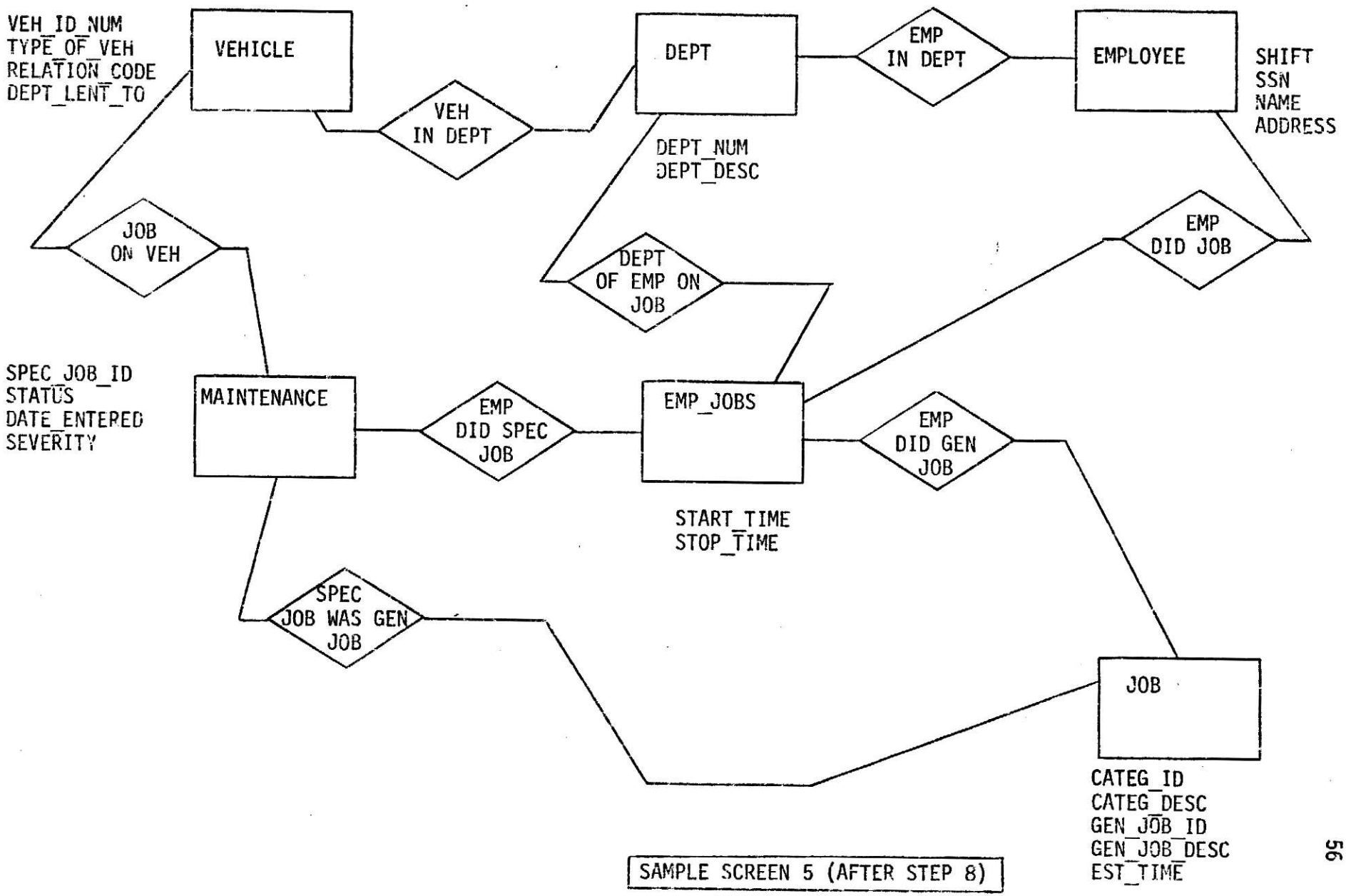
SAMPLE SCREEN 2 (AFTER STEP 3)



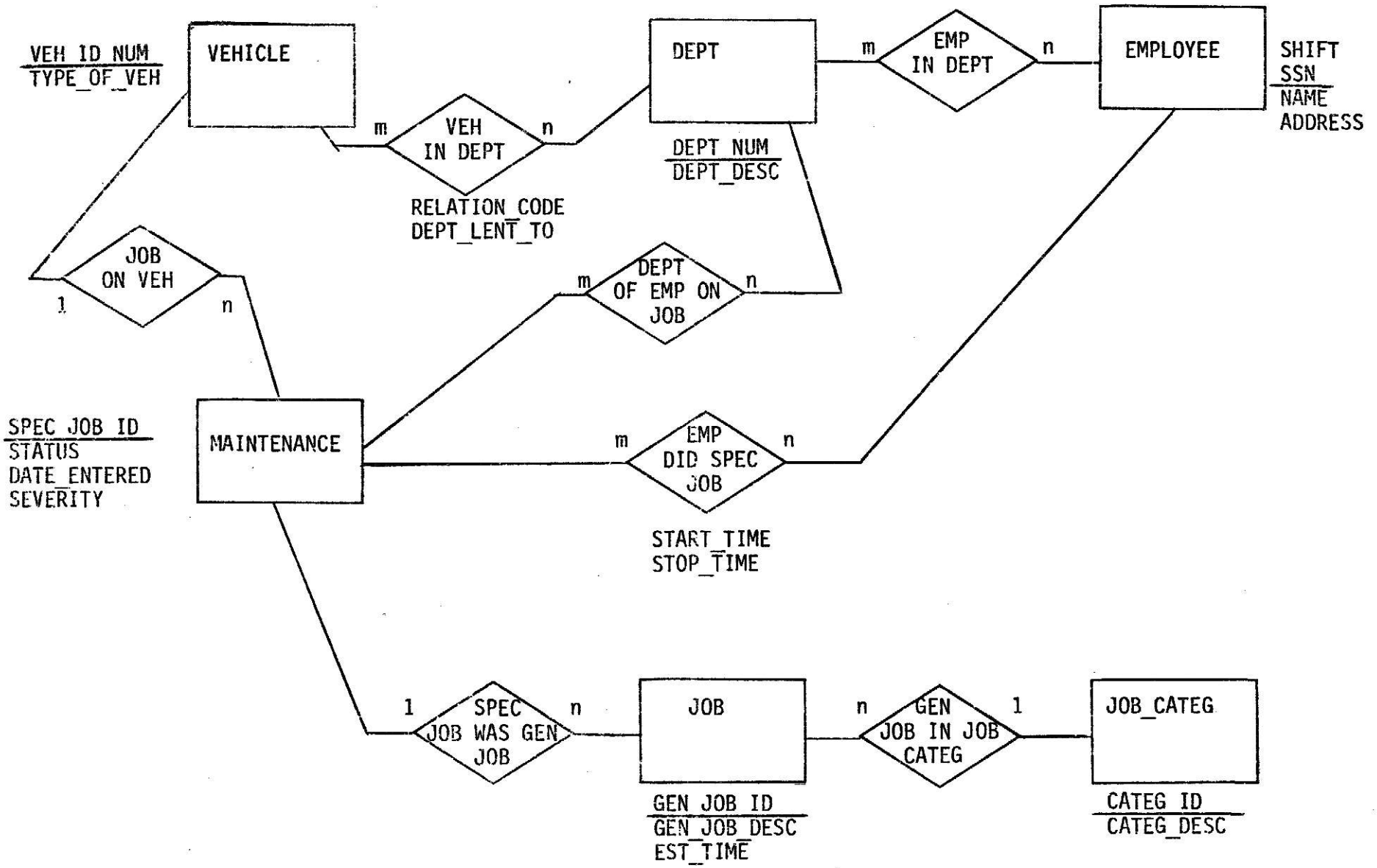
SAMPLE SCREEN 3 (AFTER STEP 5)



SAMPLE SCREEN 4 (AFTER STEP 6)



SAMPLE SCREEN 5 (AFTER STEP 8)



SAMPLE SCREEN 6 (FINAL E-R DIAGRAM)

THE GENERATION OF ENTITY-RELATIONSHIP DIAGRAMS FROM USER DOCUMENTS

by

DARRELL W. WOELK

B. S., University of Kansas, Lawrence, Kansas, 1971

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements of the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhatten, Kansas

1981

The Entity-Relationship Model is a data model which allows a logical view of data to be described at a high level. An Entity-Relationship Diagram is a graphical representation of a data base schema in the Entity-Relationship Model. This report proposes a system which will translate User Documents into Entity-Relationship Diagrams.

User Documents are all of the documents which the user normally utilizes to carry out his business enterprise. The proposed system will allow the user to input Data Item Names from each User Document. A Document Entity-Relationship Diagram will be generated by the system. The system will then assist the user in interactively manipulating the Document Entity-Relationsip Diagram to generate a true Entity-Relationship Diagram. The system guarantees that all data relationships represented in the User Document will be represented in the resulting data model. The system also will have sufficient information to generate navigation paths to access data for inclusion in Output User Documents.

This report will discuss an example data base design utilizing the proposed system. Problems encountered with the design example will be discussed in detail. General problems and conclusions will also be discussed and solutions and enhancements will be proposed.