CHAPTER V

VIEW TRANSLATION

5.1. INTRODUCTION

Many database design methodologies involve describing the database application in terms of a semantic data model, and then translating the resulting schema into a data model of a commercial database management system. We consider in this work a version of the EER model, which extends one of the most popular semantic data models, the ER model. In addition to the basic constructs of the ER model, the EER model used in this work includes the generalization and full aggregation capabilities. Relational DBMSs are currently the most popular commercial database systems available on a wide range of equipments and under various operating systems. Therefore, keeping in view of the today's processing requirements, we proposed in this work a view translation methodology to translate an EER schema into a relational schema. Methods for translating an EER diagram into relational databases have been extensively studied by many researchers [22,27,34,35,74,85,92,121,122,125]. Most of these methods are simply a form of guidelines ,rules of thumb, and/or only consider simple ER-diagrams which do not include the extended constructs, such as weak entities, recursive

A finite set of attributes $U=\{ A_1,\ldots,A_m \}$ is called a universe. A relation-scheme (over U) is a named collection of attributes, $R_i(X_i)$, where R_i is the relation-scheme and X_i U. A relation-schema is a pair (R,F), where R is a set $\{R_1(X_1), R_2(X_2),\ldots, R_m(X_m)\}$ of relation-scheme over U such that $\bigcup_{i=1}^{m} X_i=U$, and F is a set of dependencies over R.

Associated with each attribute A is a finite set Dom(A) called the domain of A. A relation **r** over the relation-schema R(X) is a subset of the cartesian product of domains corresponding to the attributes of R(X). The elements of the relation **r** are called tuples and are denoted by the letter t.

Let R(X) be a relation-scheme associated with the relation **r**, and let W be a subset of X. For any tuple t of **r**, the restriction of t to W is denoted by t[W] and is called W-value of t. The projection of **r** onto W is denoted by $\pi_W(\mathbf{r})$, and generates a relation associated with attribute set W that is equal to $\{t[W] \mid t \in \mathbf{r}\}$.

For relations $\mathbf{r}_i$ and $\mathbf{r}_j$ corresponding to relation-schemes $R_i(X)$ and $R[x_j]$ respectively, the join of $\mathbf{r}_i$ and $\mathbf{r}_j$ is denoted by $\mathbf{r}_i * \mathbf{r}_j$, and generates a relation associated with X_iX_j, that is equal to $\{t \mid t[X_i] \in \mathbf{r}_i$ and $t[X_j] \in \mathbf{r}_i\}$.

Let $R_i(X_i)$ be a relation-scheme associated with relations r_i. A functional dependency(FD) over R_i is a statement Y->Z where $YZ \subset X_i$; Y-> Z is satisfied by r_i iff for every two tuples of r_i, t_1 and t_2, $t_1[Y] = t_2[Y]$ implies $t_1[Z] = t_2[Z]$.

Let $R_i(X_i)$ and $R_j(X_j)$ be two relation-scheme associated with relations r_i and r_j, respectively. An inclusion dependency(IND) is a statement of the form $R_i[Y] \subset R_j[Z]$, where Y and Z are subset of X_i and X_j, respectively; $R_i[Y] \subset R_j[Z]$ is satisfied by r_i and r_j iff $r[Y] \subset r[Z]$. The attributes involved in the left-handed side of an inclusion dependency are called foreign attributes.

Let $R_i(X_i)$ be a relation-scheme associated with the relation r_i. A multivalued dependency(MVD) over R_i is a statement Y->->Z where YZ X_i. Let $W = X_i-YZ$; Y->->Z is satisfied by r_i iff for any two tuples t_1 and t_2 of r_i, such that :

$t_1[X] = t_2[X]$ implies $t_3[X] = t_1[X] = t_2[X]$

and $t_3[Y] = t_1[Y]$ and $t_3[Z] = t_3[Z]$.

Let **r** be a relation over the relation-scheme R(X) and let $(X_1, \ldots, X_m)$ be a set of pairwise distinct subsets of X such that $\bigcup^m_{i=1} X_i = X$. A join dependency (JD) over **r** is a statement of the form $*[X_1, \ldots X_m]$; the relation **r** satisfies the JD $*[_1, \ldots X_m]$ iff $\mathbf{r} = r[X_1]*r[X_2]*\ldots*r[X_m]$.

Let r be a database state associated with schema (R,F). Database state r is said to be F-consistent if it satisfies all the dependencies of F. Given F a dependency f is said to be implied by F if every state that satisfies F also satisfies f. The set of all dependencies implied by F is called the closure of F and is denoted by F^+.

A set of attributes K of a relation-schema $R_i(X_i)$ is said to be a candidate key of R if K $\rightarrow X_i$ (i.e. all attributes of R are functionally dependent on K) and there exits no proper subset K^1 of K such that $K^1 \rightarrow X_i$. Any superset of a key of R_i is called as a superkey of R_i A relation-schema can be associated with several candidate keys from which one primary key is chosen. An attribute which belongs to a candidate key is called a prime attribute. A functional dependency of the form Y->Z is called a trivial dependency if Z is a subset of Y . A non-trivial functional dependency Y->Z is called (i) a key dependency if Y is a superkey of R_i ; (ii) a partial dependency if Y is a proper subset of a candidate key of R_i ; (iii) a transitive dependency if Y is not a subset any of candidate key of R_i.

Some data manipulation anomalies arise due to the presence of data redundancies in relations. One process that minimizes data redundancy and hence attempts to remove data manipulation anomalies is known as normalization. Relational

Normal forms are properties that are defined for relational schemes that guarantee minimal data redundancy. For relational-schemes associated with FDs, the highest normal form is Boyce- Codd Normal Form(BCNF). A relation is in first normal form(INF) if all the attributes are atomic (i.e. the domains for each attributes are non-decomposable). A relation is in second normal form(2NF) if it is in 1NF and there are no partial functional dependencies in the relation. A relation is in third normal form(3NF) if it is in 1Nf and there are no transitive functional dependencies in it. A relation is in BCNF if it is in 1NF and all functional dependencies of the relation are key dependencies.

The concept of Fourth normal form (4NF) is associated with multivalued dependency. A multivalued dependency X->->Y|Z is non-trivial if neither B nor C is an empty set of attributes. A MVD : X->->Y|Z is said to be strong MVD if Y does not functionally dependent on X. A normalized relation **r** is said to be in 4NF if all the MVDS in **r** are consequence of key dependencies of r. The concept of fifth normal form(5NF) is associated with join dependencies. A normalized relation **r** is in 5NF iff every JD in **r** is implied by the candidate keys of r. It has been shown that any 5NF is also in 4NF, any 4NF relation is also BCNF, and any BCNF relation is also in 3NF.

5.3 Well-formed EER (WF-EER) diagrams

In this section we define what is meant by a WF-EER diagram. The objective of such a WF-EER diagram are :

(1) To capture and preserve correctly all semantics of the real world. for example it is semantically incorrect to use an entity as an attribute and a relationship as an entity in a schema.

(2) To ensure that the schema represents minimally the real world requirements. We define a schema to be minimal when no concept can be deleted from the schema without losing information. The following is a list of criteria that causes redundancy in a schema :

(a) Existence of some redundant attributes for entities or relationships. For example, the attribute age is redundant as it can be derived from the attribute DOB (date-of-birth) of a person as illustrated in Figure 5.1

(b) Existence of redundant relationships in the schema. A relationship is redundant when it can be derived from other relationships.

(c) Existence of redundant ISA hierarchies.

The definition of WF-EER diagram proposed below is more refined than the one proposed by Chuang and

Springsteel [34] and also it allows the direct translation of the EER diagram into 5NF relations. We first define what is meant by well formed Entity types and relationships types and then give a definition of WF-EER diagram.

Definition 5.1. An entity E of an EER diagram is said to be well formed (E-WF) if the following properties hold :

(1) No attribute of E is multivalued.

(2) All non-trivial FDs which involve only the attributes of the entity E are implied by its key dependencies.

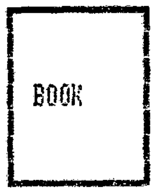
The property (2) ensures that all the relations generated from such entities are in BCNF, and together with property (1) which does not allow any multivalued dependencies among the attributes of the entity also ensures that the relations generated from such entities are in 4NF and 5NF, as there are no strong multivalued and join dependencies exist in the entity.

In figure 5.2 the set of functional dependencies of the entity employee consists of the following dependencies :

E# -> SSN, Name, DOB

SSN -> E#

[NAME,ADD,dob,age,...]
AGE is derivable from DOB
FIGURE 5.1 ENTITY with redundant attribute



[CAT_NO,name,auth_name,auth_add]

Primary key : CAT_NO
Alternative key : [NAME,AUTH_NAME]

FD : AUTH_NAME -> AUTH_ADD

FIGURE 5.2 Entity type which is not Well-Formed

[E#,SSN,name,dob]

Primary key : [E#]
Alternative key : [SSN]

FIGURE 5.3 Well-Formed Entity

Where E# is the identifier of Employee and SSN is a candidate key of it.

Employee is in WF-EER because there are no multivalued attributes of it and there is no other dependencies which only involves attributes of it and is not a key dependency.

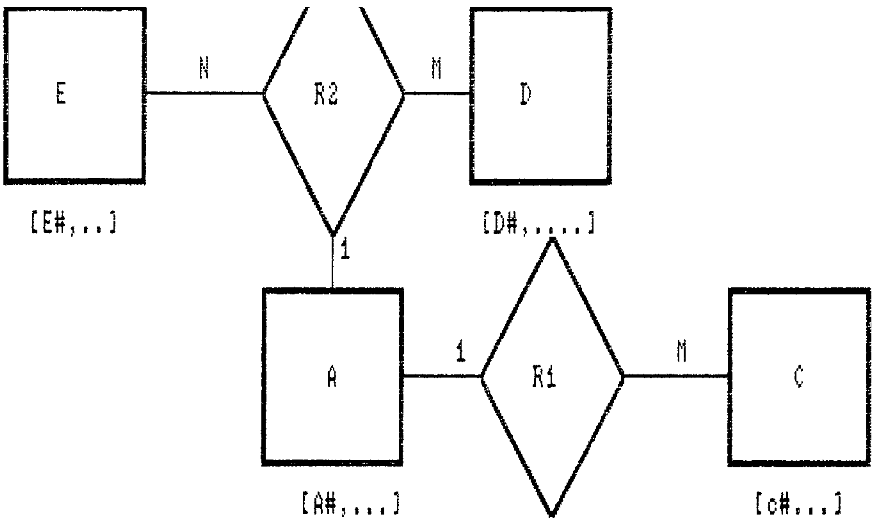
In figure 5.3 the entity book has a dependency Author_Name-> Author_Add, and Author_Name is not a key of the entity Book, so the entity Book is not well formed.

Definition 5.2. A relationship set R of an EER diagram is said to be well formed if the following properties hold for R :

(1) R does not have any multivalued attribute.

(2) All non-trivial FDs which only involve attributes of R and identifiers of entities participating in R are either implied by the set of key dependencies of R or represented by separate relationships in the schema.

For example in Fig.5.4 the relationship R1 is in R-WF, while the relationship R2 is not, because the FD : A#->D# is not represented separately by a relationship. The correct EER schema in which all relationships are in R-WF is shown in Figure 5.5.



Relationship R2 is not in R_WF

FIGURE 5.4

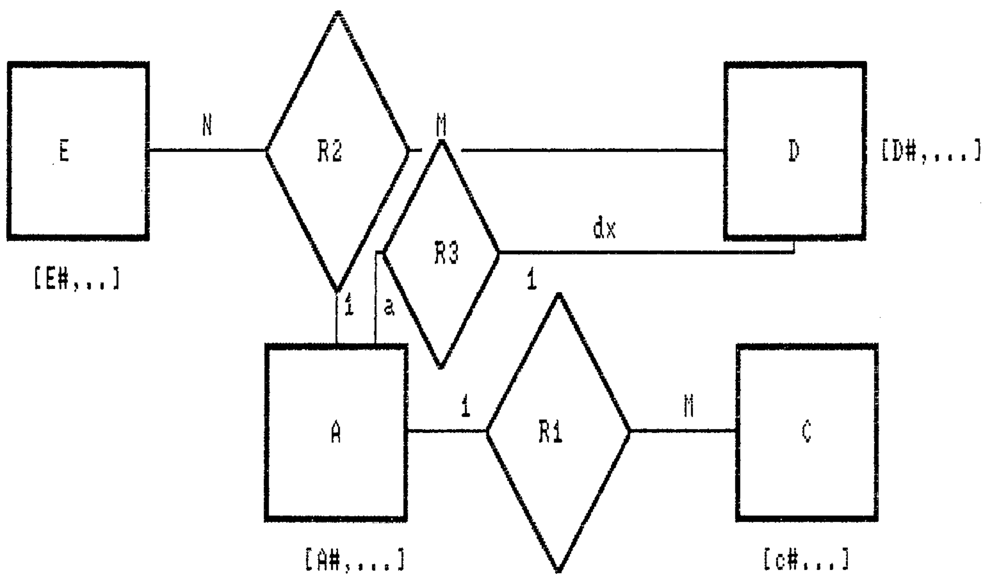


Figure 5.5 All Relationships are in R_WF

Definition 5.3 : An EERD as defined in Chapter-2 is said to be well-formed if it has these following properties: (

1) Each A node has just one connector, i.e. all the attribute names are distinct and of different semantics.

(2) An E node can not have both ISA and ID outgoing edges, and it has an identifier (Primary Key) if either it has an ID edge or no edge to another entity, but not if E_I has an outgoing ISA edge.

(3) Every R-Node has outdegree greater than one, with edges to at least two X-nodes (Where X is an entity or another relationship).

(4) The EERD is connected as a directed graph.

(5) There are no parallel edges between two nodes unless they connect a unary relationship, nor are there unlabelled edges of the form X-> E, for any node X.

(6) There are neither two distinct ISA-path nor two distinct ID-paths from any node E.

(7) The ID/ISA paths are acyclic. This property says that there does not exist contradictory ISA/ID paths and that an entity set will not depend (Via "ID") on itself for identification, nor be defined as a subset of itself.

(8) All E-nodes are in E-WF and all R-nodes are in R-WF.

(9) Every R-node with no associated A-node connectors, Satisfy the following conditions :

R is not equal to the join of any two or three other relationship sets.

Informally speaking, the first property is required in order to confirm to the universal relation assumption. The next five properties (property (2), (3), (4), (5) and (6)) are reasonable to assume for correct and complete schema. Property (7) restricts an entity to be depended on itself or a subset of itself for its identification. The property (8) ensures that all relations generated from the entities are in 5NF, and together with property (9) ensures that all the relations generated from the relationship sets of the EER diagram are in 5NF.

Conversion of an EER diagram into a WF_EER diagram

The basic steps for converting any EER diagram into a WF_EER diagram are as follows :

Step_1. convert all multivalued attributes to entities and add relationships if necessary.

step_2. Assure globally unique attributes for each A-node common to two. Different objects (Entity or Relationship) prefix the label of the object to A. For example :

Entity: Student [Name,...]

Faculty [Name,...]

Both have a common attribute "Name"

After renaming, the entities will become:

Student [Student_Name,...]

Faculty [Faculty_Name,...]

step_3. Convert non E_WF entities to E-WF entities.

By removing all undesirable FDs such as Partial FDs, Transitive Dependencies, and non trivial FDs which are not key Dependencies, and introducing new entities and relationship sets, the non E-WF entity can be converted to E-WF. For example :

Book [Cat_No, Book_Name, Author_Name, Author_Add, Publisher]

kEY : [Cat_No]

FDs : Cat_No->Book_Name,Author_Name,Author_Add,Publisher.

Book_Name, Author_Name -> Cat_No.

Author_Name -> Author_Add

The undesirable FD here is :

Author-Name -> Author_Add

By removing this we have :

Entity : Book [Cat_No, Book_name, publisher]

Entity: Author [Author_Name, Author_Add]

Relationship : written-by [Book, Author].

The entity BOOK is now in E_WF.

Step_4. Convert non R_WF relationship sets to R_WF.

By removing all the undesirable FDs and introducing new relationships and entities the non R_WF relationship set's can be converted to R_WF relationship sets.

Step_5. Remove redundant relationships.

If a relationship set R with no associated attributes which is equal to join of two or three relationship sets clearly generates relations which are not in 4NF and 5NF. Such types of relationship sets should be identified and deleted from the EER Diagram. To detect whether the join of two or three relationship sets is equal to R or not, we require more information about the semantic meaning of the relationship sets which can be provided by the database designer/user.

The other properties of WF-EER diagram can be tested by using the methods described in view modeling and Integration stages.

5.4. Translation of a WF-EER diagram to a relational database

In this section an algorithm is given to translate a well formed EER diagram to a set of relations. All the relations produced by the algorithm are in 5NF. The input to this algorithm is the WF_EERD in terms of a set of nodes (N) and a set of edges (E), which are obtained from the graph representation of the EERD as discussed in chapter-II.

The output of this algorithm is the set (R,K,I), where R is the set of relations together with its attributes, K is the set of key dependencies and I is the set of Inclusion dependencies. The notation Ident(O) is used in the following algorithm to represent the identifier of the object O, and the notation Total_att(X) represents the total set of attributes including the non-key and key-attributes of the object X.

Algorithm : Translating WF_EERD = (N,E) to Fifth Normal Form Relation Schema (R,K,I).

Input : WF_EERD = (**N**,**E**)

Output : (R,K,I)

Begin

Step_1. Initialize R,K,I to empty.

Step_2. Define primary keys and total attributes for entities.

For each e-node E_i in **N**:

(i) Ident (E_i) = Ident (E_j) if E_2 <u>ISA</u> E_j

(ii) Ident (E_i) = Ident (E_j) U key (E_i) if E_i <u>ID</u> E_j.

(iii) Ident (E_i) = key (E_i)

where key (E_i) = primary key of entity E_i

(iv) Total_att (E_i) = Att(E_i) U Ident (E_i)

where Att (E_i) is the set of non-key attributes of the entity E_i.

Step_3. Define primary keys and total attributes of relationship sets.

For each R-node in **N** :

(i) For each edge R->X in **E**:

If the edge is labeled with any Role, then prefix the identifier attributes of X with the Role names.

(ii) If all edges R->X to e-nodes are R-Nodes and are ONE-labeled ;

Then Ident (R) = Ident (X) for first such X

Else Ident (R) = U (Ident (X) | R->X is MANY-labeled).

Total_att(R) = (U Ident(X) | R-> X is in E) U att(R).

Step_4. Define relations.

For each E- or R- node X_i in **N** :

let R_i be the label of X_i;

S_i = Total_att(X_i);

K_i = Ident (X_i);

add $R_i(S_i)$ to the set of relation schemes of R.

add the KD : K_i->S_i to the set K.

Step_5. Define Inclusion dependencies.

For each edge X_i->E_j in E, where X is an E- or R-node :

if R_i and R_j are the respective relation schemes

then add an IND $R_i[K_j]$ $R_j[K_j]$ to the set I.

Step_6. Eliminate 1-attribute schemes and their INDs.

end.

5.5 Summary

In this chapter we first define the concepts of a well formed EER diagram based on the principles of normalization of the relational model, the graph theory and the concepts of the EER model. This definition for a WF_EER diagram gives the necessary and sufficient conditions for ensuring all relations of the corresponding EER diagram are in 5NF. We give a method to convert the EER diagram into a WF-EEr diagram and then an algorithm is described for translating a WF_EER diagram into a set of 5NF relations which confirms to the Universal relational assumptions. This algorithm is implemented in our expert system VMITS.