Experiment No.: 02

Title: To Map EER diagram drawn in experiment no.1 to relational model.

Batch:SY-IT(B3) **Roll No.:16010423076** **Experiment No.: 02**

Aim: To Map EER diagram drawn in experiment no.1 to relational model.

Resources needed: MS-office

Theory:

The relational model uses collection of tables to represent both data and the relationships among those data. Each table has multiple columns and each column has a unique name. The relational model is an example of record-based model. Each table contains records of a particular type. The columns of the table correspond to the attributes of the record type. The relational model is the most widely used data model.

~~Procedure / Approach /Algorithm / Activity Diagram:~~

Steps for Reducing EER model into relational model

1) Any strong entity set E having attributes a1, a2,…,an is reduced into a relation schema called E with n distinct attributes i.e. a separate relation with name E and n distinct columns.

2) Any weak entity set A having attributes a1, a2,..n and a strong entity set B on which A depends, having primary key attributes as b1, b2, …, bn is reduced into a relation schema called A with one attribute for each member of set
{ a1, a2,…, an} U {b1, b2, ……. , bn}

3) Any relationship set R having a1,a2,…,an as a set of attributes formed by union of the primary keys of each of the entity sets participating in R and b1, b2,….,bn as set of descriptive attributes is reduced into a relation schema called R with one attribute for each member of the set {a1, a2, …. ,an} U {b1, b2, …., bn}

Primary key of relationship set is decided as follows

For **binary many to many relationships** the union of primary key attributes from the participating entity sets is primary key.

For **binary one to one relationship set** the primary key of either of the participating entity set can be chosen as the primary key.

For **binary many to one or one to many relationship set** the primary key of the entity set on the many side of the relationship set serves as the primary key.

For **n-ary relationship sets without any arrows on its edges**, union of the primary key attributes of participating entity sets is a primary key.

For **n-ary relationship sets with an arrows on one of its edges**, union of the primary key attributes of participating entity sets is a primary key.

To remove redundancy we generally make separate relation schema for many to many relationship set with primary key and other attributes as mentioned above.

For one to one we combine relation schema of relationship set with relation schema of either sides of entity sets relation schema.

For one to many and many to one we combine relation schema of relationship set with relation schema of entity set on many side entity set.

We don't make separate relation schema for identifying relationship set.Every composite attribute A having subparts a1, a2,…,an is represented by separate column for each subpart in relation schema of the associated entity set.

For **multivalued attribute** separate schema is form having columns as attributes of primary key of associated entity set and a column for multivalued attribute

For **overlapping generalization/specialization** create separate relation schemas for higher level as well as lower level entity sets.

Also include the foreign key constraint in lower level entityset for the primary key attributes of higher level entity set.

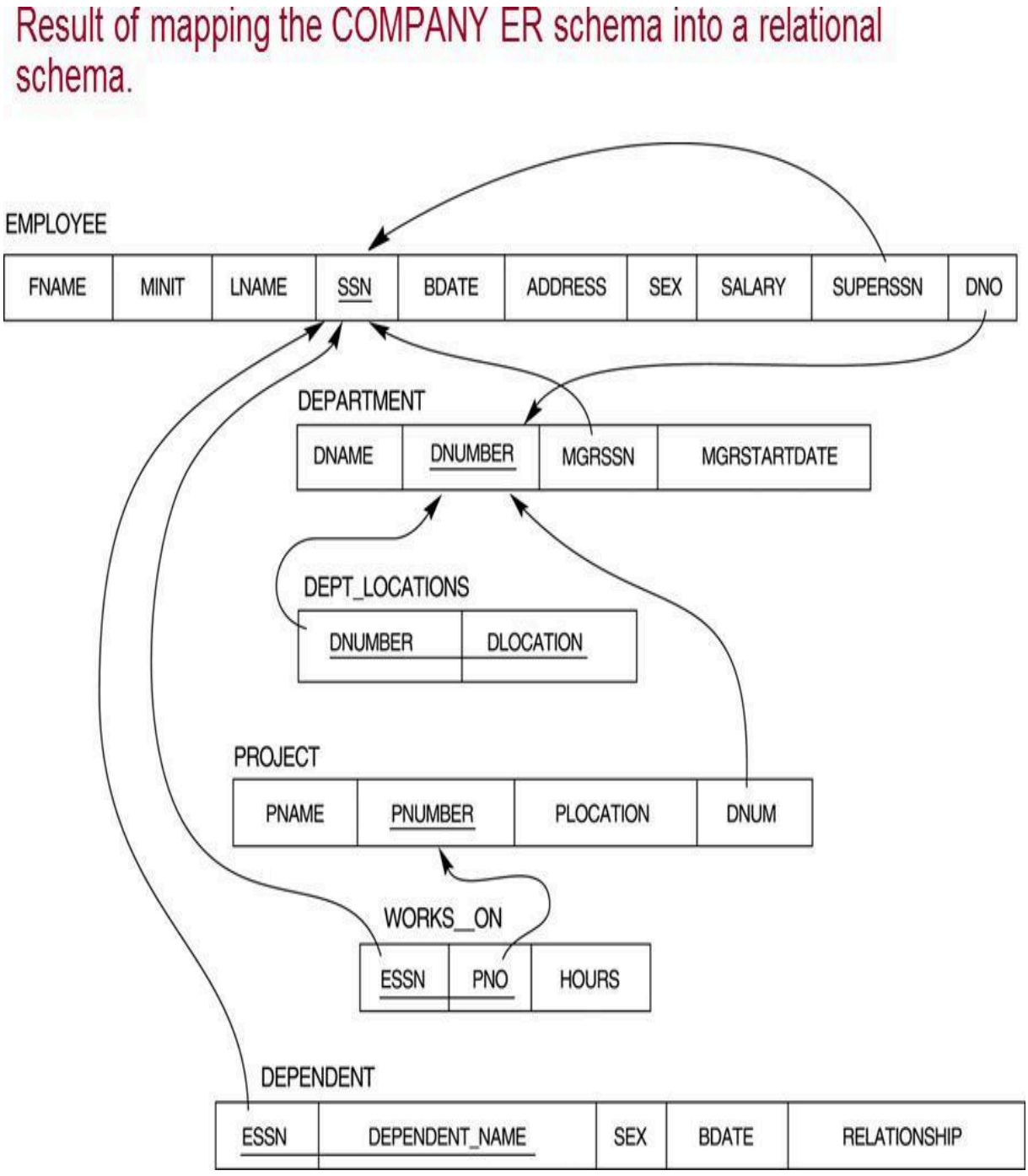
For **disjoint generalization/specialization** create separate relation schemas only for every lower level entity set(higher level entity set's attributes are inherited so add columns for same) and not for higher level entity set.

No separate relation is required to represent the **aggregation** the relation created from the defining relationship is used instead (design schema for relationship set treated as entity set carefully)

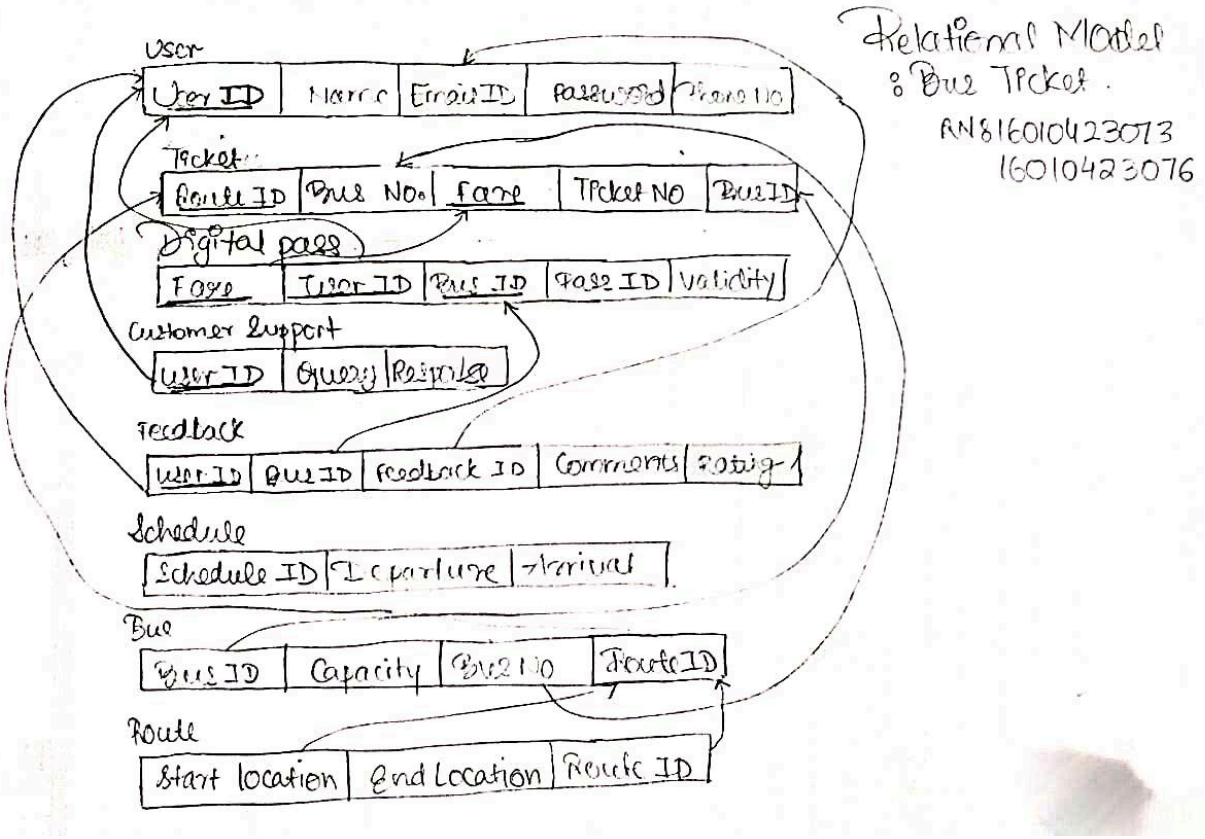
Results: (Document printout/handwritten)

1. Relational model

Example:



Outcomes:



Questions:

Q1 Explain generalization and specialization with example

Generalization: Imagine you're in a school and you have different types of people—students, teachers, and staff. To make things simpler, you could create a general category called "People" that includes everyone in the school. Generalization is about creating a broad category that can cover multiple specific instances. In database terms, it's like creating a higher-level entity that generalizes the properties of several more specific entities.

Example: If you have separate tables for "Employees" and "Managers," you could generalize

them into one table called "Staff" that includes both employees and managers, with a field to specify the role.

Specialization: Now, let's say you want to break down "People" into more specific groups like "Students" and "Teachers" because each group has unique details. Specialization is about breaking a general category into more specific ones to capture unique characteristics.

Example: From the "Staff" table, you could specialize it into "Employees" and "Managers" to manage their specific roles and details more accurately.

Q2 what is physical and logical data independence in DBMS.

Logical Data Independence: This means that changes to the logical structure of the database (like the tables and relationships) don't affect the application programs or the way data is accessed. For instance, if you decide to add a new column to a table or change the table structure, the way you access and use that data through your application shouldn't need to change.

Example: Suppose you add a "Date of Birth" column to a "Students" table. With logical data independence, your application should still work fine without needing to change how it reads student data.

Physical Data Independence: This refers to changes in the physical storage of the data (like how data is stored on disk) not affecting the logical schema of the database. In other words, you can change how the data is stored without changing how you access or view the data.

Example: If you decide to move data from one type of storage device to another or reorganize the way data is stored for performance reasons, the logical structure and queries you use to access the data should remain the same.

Conclusion:

I learned how to translate complex entity relationships into a structured format for databases. I also gained insights into generalization and specialization, understanding how to abstract and detail information effectively.

Reference books:

1. Elmasri and Navathe, "Fundamentals of Database Systems", 6th Edition, Pearson Education
2. Korth, Slberchatz,Sudarshan, :"Database System Concepts", 6th Edition, McGraw – Hill