**Activity-3**

**1.Document the steps to create logical design.**

All the relationships we have identified are going to be implemented in the database by either creating a new table, or by just creating a new column, or maybe there are some other options we are going to cover here.

So, here are the steps:

**1.Mapping of regular entity types**

For each entity, create a table that includes all of it’s simple attributes. Than, choose the primary key, if it’s composite, than a set of simple attributes will together form the primary key.

**2.Mapping of weak entity types**

For each weak entity, create a table that includes all it’s simple attributes. And include a foreign key points to the primary key of the owner entity, where the foreign key and partial key will be the primary key of the weak entity.

**3.Mapping of 1:1 relationship types**

There are three ways

Foreign Key approach: Choose the primary key from either one of the entities, and make a foreign key in the other entity referencing the primary key of the first one.

Merged relation option: Merge both entities, since every row in any entity will have a corresponding row in the other entity.

Cross-reference or relationship relation option: Create a third table, that has two foreign keys from the primary keys of both entities (overkill!).

**4.Mapping of 1:M relationship types**

Considering the example we have been using, where each employee works for only one department, while a department can have more than one employee.

Now, in order to map this relationship, we add a foreign key in the employee table (many side), which in turn will point to the primary key of the department table (1 side).

**5. Mapping of M:N relationship types**

A good example for this relationship, is every employee can work on one or more project, and every project can have one or more employee involved in this project. So it’s many from both sides.

Now, in order to map this relationship, we create a new table, this table exists only to connect the employee and project tables. You start by adding two foreign keys, each one will point to a primary key of one of the two tables. The two foreign keys together will form the primary key of the new table.

There may be an employee called Adam who works on Project A, and Project B. And another employee who’s working on Project A. But, you can’t have duplicates; meaning, you can’t have two rows with the same employee, and the same project

**6.Mapping of Multi-valued Attributes**

Remember? A multi-valued attribute is a set of different values.

As an example, a department may have different locations, thus it will have different location values for each department.

So, we create a new table that has a foreign key points to the primary key of department, and another column represent the multi-valued attribute (one-to-many relationship). This is done for each multi-valued attribute.

The foreign key and the multi-valued attribute together will form the primary key of the new table.

**7.Mapping of N-ary Relationship Types**

“What if you have a relationship that connects more than two tables?”. It’s almost the same solution as we did in M:N relationship.

Create a new table, that has the foreign keys from the primary keys of all participating entities, the foreign keys together will form the primary keys of the new entity. Also include any simple attribute of that relationship in the new table.