Practical 6 – Relational Database Design and Normalization

In this practical you will learn how to derive a relational model from an (Enhanced) Entity Relationship (ER) Model and to use normalization to validate the relational model. This work forms part of the second phase in database design known as logical database design. The relational model you build during this phase should be independent of any specific DBMS such as MS Access or MySql. The outputs (deliverables) of logical design phase are ***a collection of normalized relations (tables)***. In this practical, you create MSAccess/MySQL tables based on your design and populate the database with some example data. This means, you also learn how to perform physical database design in this practical.

Because, you are by now familiar with the DreamHome domain, we start learning the rules involved in table design using an ER Model for the DreamHome domain as an example. You will then use this experience to design tables for a different database. Please note that you have several ER Models for the DreamHome domain, one of which is developed by you in exercise 1 of practical 4. You could use any of these as a starting point for table design.

When you design databases, in the real world, you do not always build a complete ER before starting table design. (Note: in this practical table design and relational design and logical design are all used interchangeably.) In fact it helps to perform database design iteratively in several cycles interleaving ER Modelling with table design. In the initial cycle you identify the major entities and their relationships which you try to model as tables. In the next cycle, based on the feedback from the table design phase from the first cycle you go back and add more details to the ER Model before making another attempt at table design. You cycle though these ER modelling and table design activities several times until your tables fulfil all the client's and quality requirements (Normalization requirements). Therefore, if you have a partial ER Model from exercise 1 practical 4, you are encouraged to start this practical from this ER Model and incrementally perform both ER Modelling and table design.

Exercise 1
In this exercise, you work with the DreamHome domain. Please select an ER Model that can drive your table design. As stated earlier this ER Model need not be perfect. Keep the *DreamHomeRequirements.rtf* document ready for reference. In fact, any material related to the DreamHome domain such as the *Case Study* will be useful to you while performing table design. Three tasks are performed in logical database design:

1. Designing Tables
2. Linking the Tables Or deciding foreign keys
3. Normalizing the Tables

These three tasks are performed over a number of steps as explained below:

1. In the ER model identify strong entities and for each of them design a table with columns (fields) for the table derived from the attributes associated to that entity. Decide the primary key attribute (or group of attributes). For example, from the DreamHomeER2.doc the entity PropertyForRent can be modelled as a table

   PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent) Primary Key propertyNo
   Now create tables for all the other entities from your ER Model.

2. In the ER model identify 1:* binary relationships and for each of them decide the entity that acts as the parent and the entity that acts as the child. (Note: the entity on the 1 side of the 1:* relationship is always labelled the parent entity). Make sure that both the entities participating in the 1:* relationship have been modelled as tables in the previous step. Post a copy of the primary key of the parent table as the foreign key in the child table. For example, consider the 1:* relationship Staff manages PropertyForRent. We assume that both Staff and PropertyForRent have been modelled as tables

Staff (staffNo, fName, lName, position. Sex, DOB)
Primary Key staffNo
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent)
Primary Key propertyNo

Now because of the 1:* relationship, Staff table is the parent table and PropertyForRent table is the child table. Therefore we redesign PropertyForRent table

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, staffNo)
Primary Key propertyNo
Foreign Key staffNo references Staff(staffNo)

Note that the design of Staff table remains the same. A foreign key is added to the PropertyForRent table creating a link to Staff table. Now create foreign keys due to all other 1:* relationships from your ER Model.

3. In the ER Model identify *:* binary relationships and for each of them create a new table where a copy of the primary key from both the participating entities is posted as foreign keys. Try this yourself on the *:* relationship between PropertyForRent and Client.

4. In the ER Model identify 1:1 binary relationships. In this case, the both entities in the relationship have the same max constraint of 'one'. So we use the min constraint to determine the parent table and child table in the relationship. Use the following rules to model 1:1 relationships with known min constraints:

```
IF Mandatory participation on both sides  THEN Combine
     entities into one relation
ELSE
IF Mandatory participation on one side  THEN
     Post primary key of entity on 'optional' side to act as foreign key in relation representing entity
     on 'mandatory' side
ELSE
IF Optional participation on both sides THEN
     Follow any one of the above methods unless you have additional information about the
     relationship in support of one of them
```

Try this yourself on all the 1:1 relationships in your ER Model.

5. In the ER Model identify complex relationships with degree > 2. In this case, for each of them create a new table where a copy of the primary key from all the participating entities is posted as foreign keys. Try this yourself on the ternary relationship among Staff, Client and Branch.
6. Go to Exercise 2 for normalizing the relations (tables).

Exercise 2

Implement your tables in MSAccess/MySQL and please check if all the relations (tables) in your DreamHome database are in the following normal forms (When you create tables, you must define the integrity enhancement features such as domain types, primary keys and foreign keys.):

Sr. No	Normal Form	Meaning
1	First Normal Form (1NF)	A relation in which intersection of each row and column contains one and only one value.
2	Second Normal Form (2NF)	A relation that is in 1NF and every non-primary-key attribute is fully functionally dependent on the primary key.

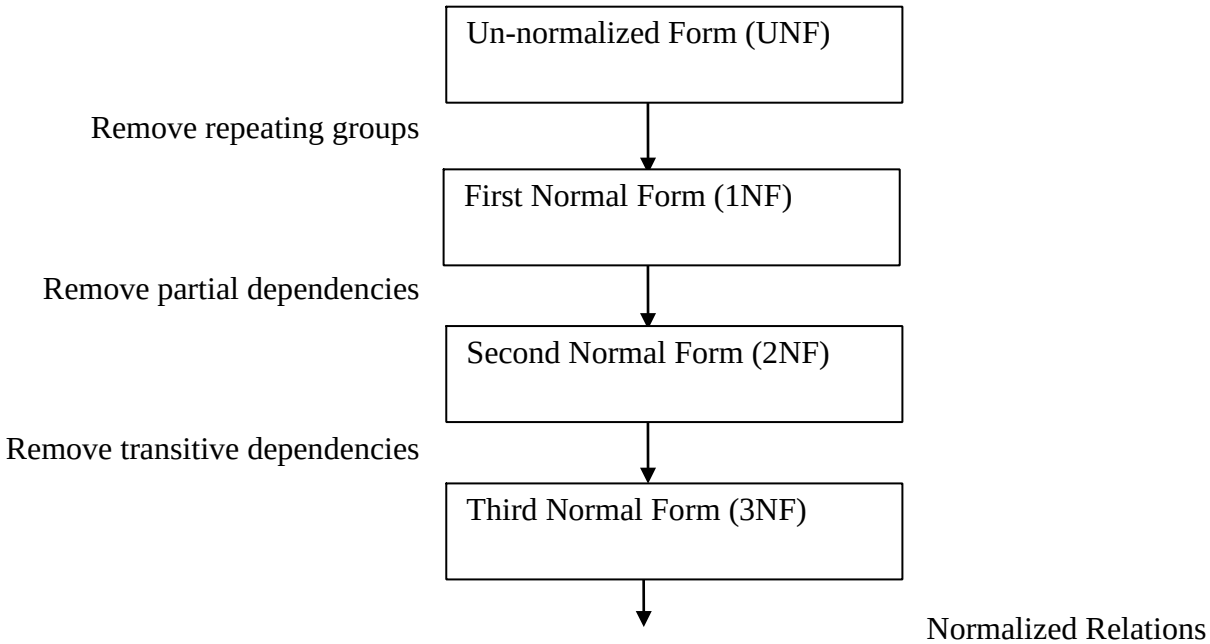
3	Third Normal Form (3NF)	A relation that is in 1NF and 2NF and in which no non-
		primary-key attribute is transitively dependent on the primary key.

Exercise 3

In practical 4, you have designed an (E)ER model for the Equipment Record System (ERS).

(a) Translate this (E)ER design into a relational model using the guidelines described in exercise 1.

(b) Validate your relational model using normalization using the definitions of normal forms from exercise 2. For each relation in your relational model apply the process of normalization as shown below:



Exercise 4

In this exercise you implement the logical design developed in exercise 3 as an MSAccess/MySQL database.

(a) Create MSAccess/MySQL tables based on the normalized relations (tables) obtained from 3(b). When you create tables, you must define the integrity enhancement features such as domain types, primary keys and foreign keys.

(b) Populate the tables created in 3(b) using some example data, four to five rows of data per table.

(c) Write SQL queries for the transactional requirements specified for the ERS system in practical 4. When you write these queries do you feel the need to fine tune your tables? If yes, please go back to practical 4 and make the needed refinements to the EER design and redo the steps 3(a) – 4(c) to obtain improved tables.