**Assignment 2a**

You are now familiar with multi table database design and design principles aka Normal Forms to create a ‘good’ table structure. In a real life scenario, a database will most definitely consist of more than one table. Each table will store data about one entity of interest. Relationships will exist between the tables. In your first Assignment 1a, you had come up with a real life scenario where a database will work well. You had created a relevant table to store information about an entity.

Continue working with the script you created for Assignment 1a. Ensure it is **well-documented -** include name(s), date and purpose of the script at the head of the script and comment all code sections. **Use meaningful names and consistent naming conventions throughout the script.** The script should execute correctly without any syntax or logical errors or warnings.

Start with the script you created in Assignment 1a. **Add another relevant table to the database now**. I am going to refer to the table you created in Assignment 1a as the ‘old’ table and this table as the ‘new’ table. Do a good job creating this new table and populating it with data. In the next sub assignment, you will need to retrieve data from the two tables, using joins and subqueries. The tables should satisfy the following requirements.

1. The new table should at least have 5 relevant fields. Choose the CORRECT BEST FIT data types and other properties (not null, default etc) depending on the data to be stored in the fields. **Comment the PURPOSE of each field.**
2. Both tables must be in Second Normal Form/2NF. Put the definitions of First Normal Form/1NF and Second Normal Form/2NF in comments and justify that the two tables are in 2NF.
3. The new table must have a foreign key that references a field (preferably primary key) of the old table. You can have more than one foreign keys in the new table, if required. You may ‘reverse’ this requirement. The old table can have a foreign key instead referring to a field (preferably primary key) of the new table. The aim is that the two tables are ‘related’ via foreign key(s). You will not be able to do the following sub assignments if the tables are not ‘related’.
4. Clearly ‘comment’ the type of relationship (1-1, 1-m, m-m) between the two tables.
5. Display the structure of the new table.
6. Populate the table with **relevant** data, at least 10 records. Ensure that you have the set the sql\_mode to “STRICT\_ALL\_TABLES” to catch data input errors. **Important Note: Use literal text data in quotes, numeric data without quotes and date in a consistent format of your choice. You will lose marks if you do not follow this instruction.**
7. Add a SELECT statement to display the records of the new table.

Provide an Entity Relationship Diagram (ERD) to depict the database structure- the entities and the relationship(s) between them. You can use any software program or do it by hand. The ERD must clearly show all tables, all the fields in the tables, clearly labeled primary keys and foreign key(s) and the relationship(s) labeled with the type of relationship (1-1, 1-m, m-m). I have covered all this in my videos starting Week 7. It must have a legend, if you are using colors or symbols to label the fields. I leave the ERD creativity to you but it must be clear, correct and complete. I need you to take a ‘Print Screen’ or a picture (if you drawing by hand) and paste it into a .doc file.

**How to submit**: Name the script Assignment2a*your\_name(s).*sql and the ERD as Assignment2aERD*your\_name(s).*doc. Submit TWO files, not a zipped file for this assignment. I will not accept or mark any zipped files. Use the **Assignment 2a** link in the **Assignment 2** folderin Assignments to submit the two files.Only one member of the group needs to submit and resubmit if required, for all assignment parts constituting Assignment 2.