1. Using **CREATE TABLE** statements, you are required to create at least **3 tables**, ensuring there is **at least one many-to-many (m <-> n) relationship**, and to define integrity constraints for these tables:

- Key constraints: **unique**, **primary**
- Constraints on **column values**
- Constraints on **record values**
- **Foreign key** constraints

You should also **associate comments (descriptions)** to both tables and columns.
When choosing your problem domain, consider that these tables will need to be **modified later by adding columns with values of a user-defined type**.

Additionally, at least **one table should include a numeric column** that can be used in later requirements. For the defined tables, you are required to create **relevant indexes**, and to **provide comments explaining your choices**. (**1.5 points**)

2. You are required to use **statements** for **inserting**, **updating**, and **deleting** data in the defined tables. You should observe how the previously defined **constraints are enforced** by attempting to execute statements that **violate those constraints**. (**1 point**)

3. You are required to use **system views** to retrieve information about: the **defined tables**, the **tables you have access to**, the **defined constraints**, the **constructed indexes**, the **columns** of a specific table (**name**, **type**). (**0.5 points**)

4. Using the tables created earlier, you are required to implement a **procedure** that determines, for a **numeric column**, **up to p% of records** from each category that fall **in the middle of the value range** (**50 - p/2% <-> 50 + p/2%**), sorted **in descending order** by the numeric value.
(**p** is a parameter of the procedure).

Example:

- If there is a **products** table with the columns **price** and **category**, and for **category 1** there are **10 products** with prices between 10 and 100, and **p is 20%**, then you will display **2 products**, from positions **50 and 60**. (**2.5 points**)

5. You are required to create a **view** that provides the **list of procedures** (**schema name**, **procedure name**) that the **current user** has access to. (**0.5 points**)

6. You are required to create a **procedure** that takes as parameters a **schema name** and a **procedure name**, and returns the **source code text** with which the procedure was defined, in the **original line order**. (**1 point**)

7. You are required to **use a programming environment** (**PHP, Java, C#, Python**) to implement components that fulfill the requirements above (**display table data**, **display data from a view**, **execute a procedure**). (**2 points**)

All the used instructions should be included in **command files (scripts)** so they can be **re-executed and tested**.

Deadline: 4 calendar weeks (**March 25th**)