Project Phase I - Requirements Document

Consider a mini-world of your choice and come up with requirements for database design, and functional requirements for database operations.

A mini-world is a set of users and how they will use the database you design—for example, a school examination system used by students, teachers, and more.

Note: You cannot use any of the mini-worlds provided in the group assignments.

Refer to temp.docx for the template of the submission.

Clearly state

1. The purpose,
2. Users,
3. And the applications of the mini-world.

Database Requirements:

1. At least five entity types,
2. At least two weak-entity types,
3. At least one entity with two primary key attributes.
4. At least one n>3 relationship type,
5. At least one subclass,
6. At least two of each composite, multi-valued, derived attribute.
7. Relationship with cardinality constraints.

Bonus :

- At least one entity type of each with candidate key, super key, alternate key.
- for the relationships with degree n where n > 2 , explain how it can be modeled differently as n binary relationships and weak entities.
- Relationship type with the same participating entity type in distinct roles.
    - Example: In a COMPANY database SUPERVISION relationships between EMPLOYEE (in the role of supervisor) and EMPLOYEE (in the role of subordinate)

Functional Requirements:

In your mini-world, you will be required to have applications that operate on the database. The application will query and/or update the database.
Example: Check for a student's grades from last year. These operations are functional requirements.

The functional requirements to meet are:

1. Retrieval :
    a. At least one query function for each:
        i. ***Selection:*** Example: "Retrieve complete data tuples of students belonging to UG2K20".
        ii. ***Projection:*** Query to enable the users to search the database by a particular attribute. Example: "Names of all courses with ≥ 50 students.
        iii. ***Aggregate:*** Function (SUM, MAX, MIN, AVG). Example: "Maximum score secured in the midterm exam".
        iv. ***Search:*** Search (partial text match) for entries in an entity, matching for subparts of the entries. Example: "APP" will match with "APPLE".
    b. Analysis : At least two analysis reports to be generated.
        i. Example: "Number of students that scored above average in the midterm exam in all courses".
        ii. Note: We expect that these reports convey something about the relationship between entities and are not simple selection operations from a single entity. To do so, you have to use the Join operator.

2. Modification :
    a. At least two loading or insertions of data, check for violations of integrity constraints.
    b. At least three modify or update operations.
    c. At least one delete operation.

The functional requirements must make at least one change to the data requirements.

There will be a small weight for mini-worlds that are unique with a very specific purpose and specialized applications. For such mini-worlds above requirements can be relaxed.

Submission:

One member per team must upload a PDF document (named as <teamname>.pdf - without the '>' & '<') with the following details:

1. A paragraph or two describing the mini-world, purpose of the database, and users of the database. What will the users do with the database?
2. Database requirements section
    a. Relationship between entities
3. Functional Requirements section

Points to Remember:

As always, it's good to get an early start on the assignment! Try keeping the following points in mind when you get to work:

- The requirements document is simply a description of what your database has to store and what functionality you provide on top of it.
    - Example: If your database requirements describe the relations between students, their grades, courses, and instructors, the functional requirements describe what operations should be possible - such as looking for a report card.
- Specify constraints that your data may have. If an attribute has a specific, non-trivial domain, please specify the domain. If there are cardinality constraints, participation constraints, for instance, if an object of one entity can be related to only one other object of another entity type, specify. Example: an employee can belong to only one department.
- You may not understand some of the concepts like join, aggregate, etc. as of now, do not worry about that. You will know them well enough by the time you have to build the app.
- You will build your ER models and your final program based on these sets of requirements. Hence, try to think about this in detail and keep in mind that this cannot be changed drastically later.
- Check for data redundancy, data redundancy occurs when the same piece of data is stored in two or more separate places.
- There is no need for an ER diagram in this document and if you make ER diagrams that will not be graded.
- **Plagiarism is strictly prohibited.** Even if your requirements are not perfect, the emphasis is on development of a sound logic. If there is the slightest doubt that you are not clear about the requirements and it's implementation, you will be heavily penalized.