

# Project Phase 1

CS4.301 Data and Applications

Deadline: 23/10/2023, 11:59 PM

## 1. The Task

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

Please find your assigned TA for the project in the sheet shared on moodle.

*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.

**Note:** State your **assumptions** and **constraints of your** mini-world, especially if it's a hypothetical or unknown mini-world.

## 2. Database Requirements

Please list out the requirements of your database/mini-world in a format similar to COMPANY mini-world covered in the class (Slides 39 & 40 in Lecture 3).

Your mini-world should result in data requirements that have:

- At least five strong entity types
- At least one weak entity with two key attributes
- At least two weak-entity types
- At least five relationship types(which should include cardinality ratios and participation constraints)
- At least one ( $n > 2$ ) degree relationship type.
- Few composite, multi-valued, derived attributes

### Bonus

- Relationship type with the same participating entity type in **distinct roles Eg.** Example: In a COMPANY database SUPERVISION relationships between EMPLOYEE (in the role of supervisor) and EMPLOYEE (in the role of subordinate)

- At least one ( $n > 3$ ) degree relationship type.

**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.**

### 3. Functional Requirements

In your mini-world, you will be required to have applications that operate on the database. Any operation on your database for some useful purpose in your mini-world is a functional requirement.

For instance, let's consider a scenario where you're managing employee records.

These operations are functional requirements:

- **Retrieval Operations (at least one query for each)**
  - **Selection:** "Retrieve a list of all employees who joined the company in the last year."
  - **Projection:** Query to enable the users to search the database by a particular attribute. Example: "Names of all employees in the marketing department."
  - **Aggregate (SUM, MAX, MIN, AVG):** Perform an operation on the data to get the desired output. Example: "The highest monthly sales achieved by any employee."
  - **Search:** Search (partial text match) for entries in an entity, matching for subparts of the entries. Example: Searching for "Man" to find "Manager."
- **Analysis (at least two analysis reports to be generated)**

**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.

Examples:

"Average tenure of employees in each department."

"Success rate of projects undertaken by different departments."
- **Modification Operations (at least one query for each)**
  - **Insertion** of data, check for violations of integrity constraints. "Adding the details of an employee when he joins"
  - **Update** operation. "Update the salary of an employee after promotion"
  - **Delete** operation. "Delete the details of the employee after he leaves the company"

## 4. Submission

One member per team must upload a PDF document (named **<teamnumber>.pdf** - without the “<” and “>”) with the following details:

- A paragraph or two **describing** the mini-world, **purpose** of the database, and **users** of the database.
- Database requirements section
- Functional requirements section

### Bonus

- If the database that you are creating already exists in the real world, mention a few points where you think your database improves upon the one that already exists.

## 5. 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 must store and what functionality you provide on top of it.
- 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**.

**Good Luck :)**