

Databases

Laboratory work 4

Prpared by: Izbassar Assylzhan

Task 1.

- a. What are the main phases in the database design? What is done on each development phase?

There are 4 database design phases.

- The initial phase of database design is to characterize fully the data needs of the prospective database users. The database designer needs to interact extensively with domain experts and users to carry out this task. The outcome of this phase is a specification of user requirements. While there are techniques for diagrammatically representing user requirements, in this chapter we restrict ourselves to textual descriptions of user requirements.
- Next, the designer chooses a data model and, by applying the concepts of the chosen data model, translates these requirements into a conceptual schema of the database. The schema developed at this conceptual-design phase provides a de- tailed overview of the enterprise. The entity-relationship model, which we study in the rest of this chapter, is typically used to represent the conceptual design. Stated in terms of the entity-relationship model, the conceptual schema specifies the entities that are represented in the database, the attributes of the entities, the relation- ships among the entities, and constraints on the entities and relationships. Typically, the conceptual-design phase results in the creation of an entity-relationship diagram that provides a graphic representation of the schema. The designer reviews the schema to confirm that all data requirements are indeed satisfied and are not in conflict with one another. She can also examine the design to remove any redundant features. Her focus at this point is on describing the data and their relationships, rather than on specifying physical storage details.
- A fully developed conceptual schema also indicates the functional requirements of the enterprise. In a specification of functional requirements, users describe the kinds of operations (or transactions) that will be performed on the data. Example operations include modifying or updating data, searching for and retrieving specific data, and deleting data. At this stage of conceptual design, the designer can review the schema to ensure that it meets functional requirements.
- The process of moving from an abstract data model to the implementation of the database proceeds in two final design phases.

- In the logical-design phase, the designer maps the high-level conceptual schema onto the implementation data model of the database system that will be used. The implementation data model is typically the relational data model, and this step typically consists of mapping the conceptual schema defined using the entity-relationship model into a relation schema.
- Finally, the designer uses the resulting system-specific database schema in the subsequent physical-design phase, in which the physical features of the database are specified. These features include the form of file organization and choice of index structures.

b. What is the entity-relationship (**ER**) **data model**?

The **entity-relationship (E-R) data model** was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database.

An E-R diagram can express the overall logical structure of a database graphically. E-R diagrams are simple and clear—qualities that may well account in large part for the widespread use of the E-R model.

Task 2.

a. We can see the student entity in Figure 1.

| Student | |
|-----------|---------------------------------|
| PK | <u>ID</u> |
| | name (first_name, last_name) |
| | tot_cred |
| FK | dept_name |
| | date_of_birth |
| | [phone_number] |
| | age() |

Figure 1 – Student entity

- b. In Figure 2, we can see entities of “University”, “Course”, “Dormitory”, “Teacher”, “Office of the Registrar”.

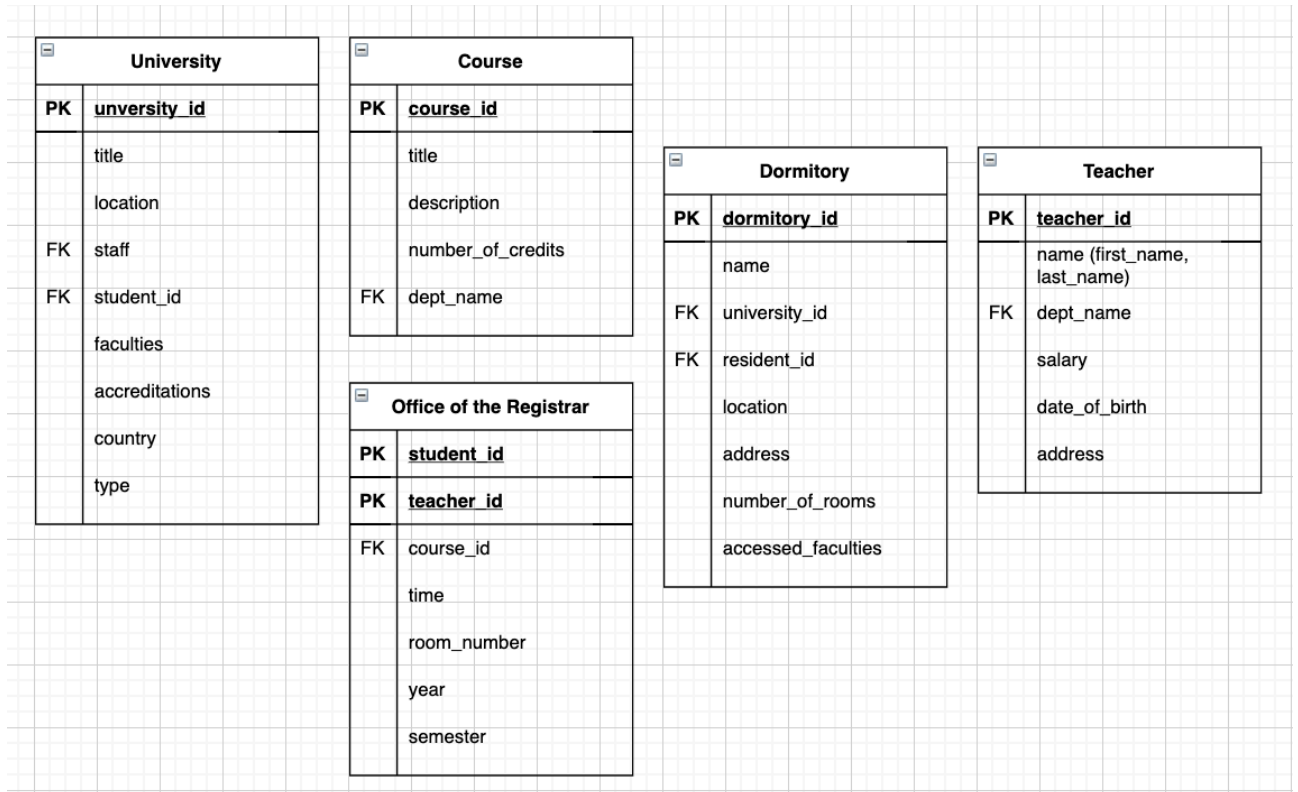


Figure 2 - Entities of “University”, “Course”, “Dormitory”, “Teacher”, “Office of the Registrar”

Task 3.

For a binary relationship set R between entity sets A and B , the mapping cardinality must be one of the following:

- **One-to-one.** An entity in A is associated with at most one entity in B , and an entity in B is associated with at most one entity in A . Example in Figure 3.

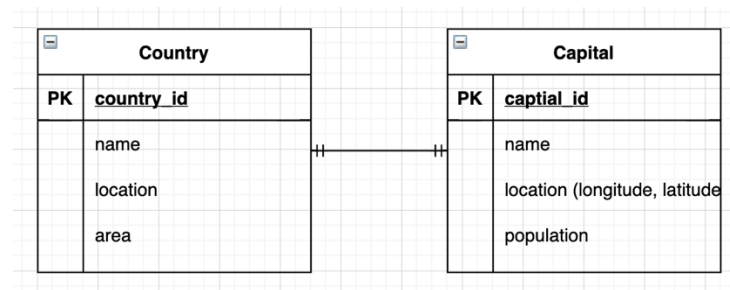


Figure 3 – Example of one-to-one relationship.

- **One-to-many.** An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A. Example in Figure 4.

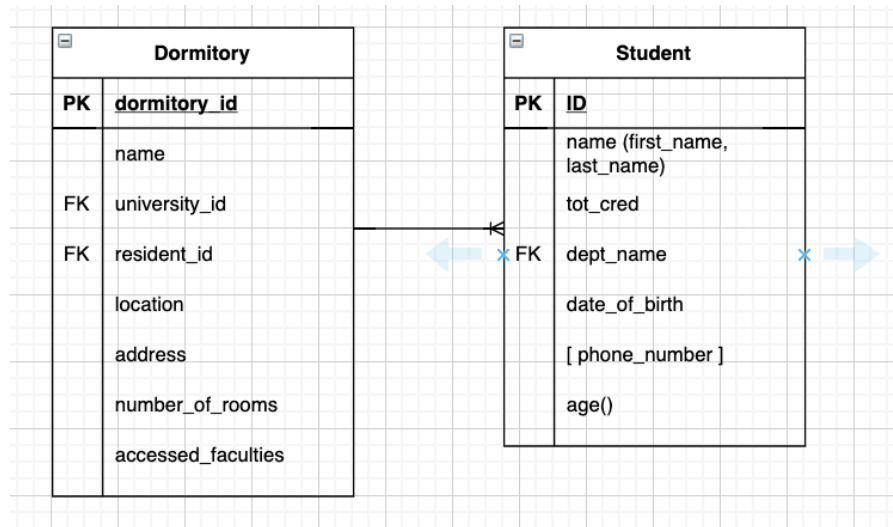


Figure 4 – Example of one-to-many relationship

- **Many-to-one.** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A. Example in Figure 5.

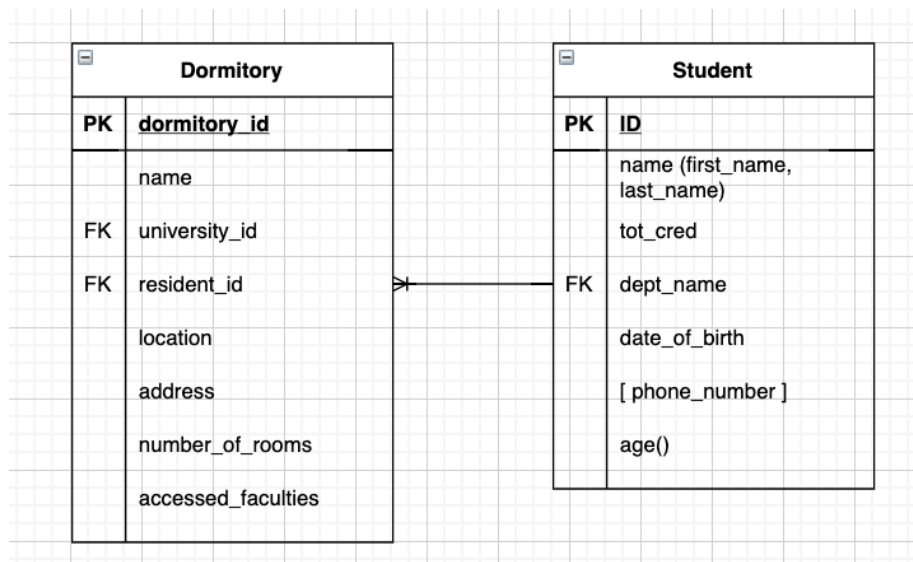


Figure 5 – Example of many-to-one relationship

- **Many-to-many.** An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A. Example in Figure 6.

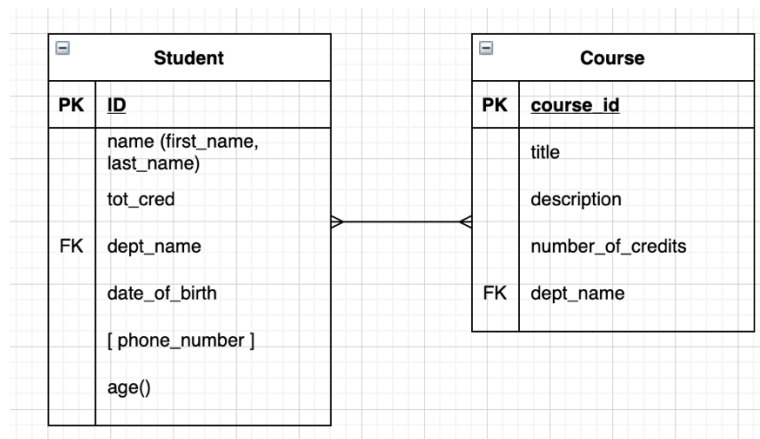


Figure 6 – Example of many-to-many relationship

Task 4.

In Figure 7 we can see ER data model with entities in second task.

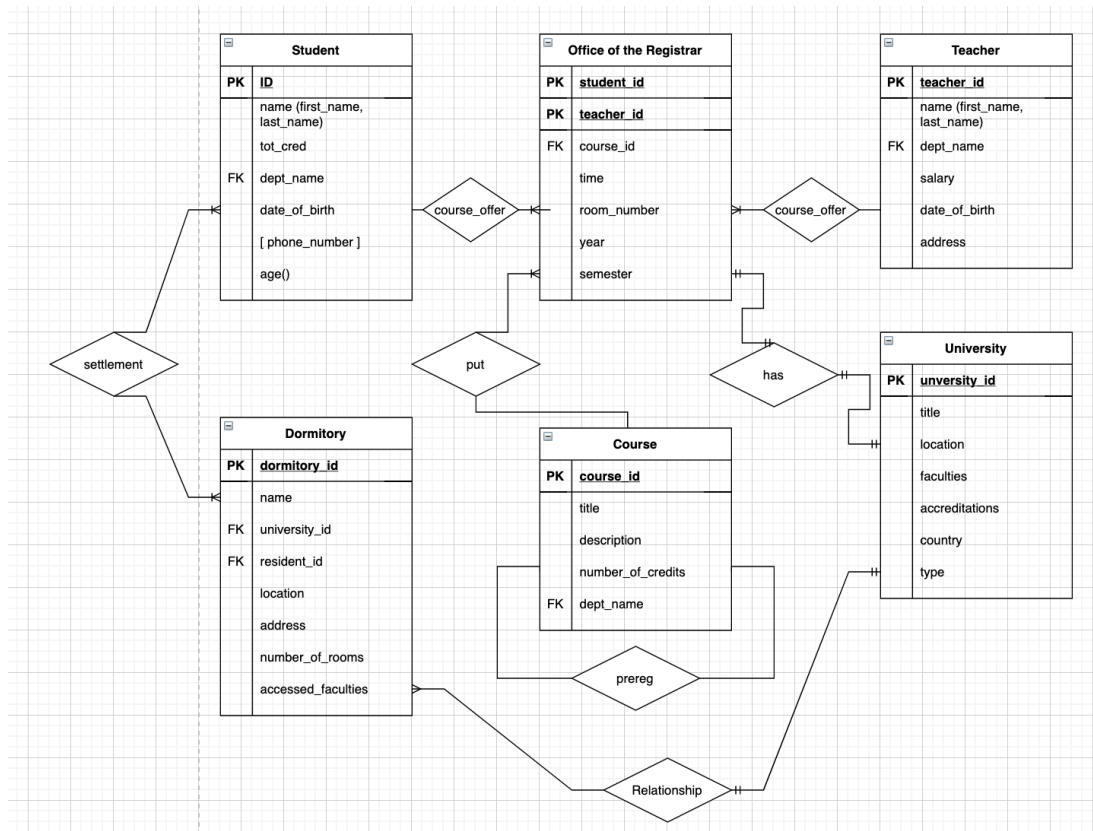


Figure 7 – ER model for Task 2

Task 5.

In the Figure 8 we can see ER data model for IT department.

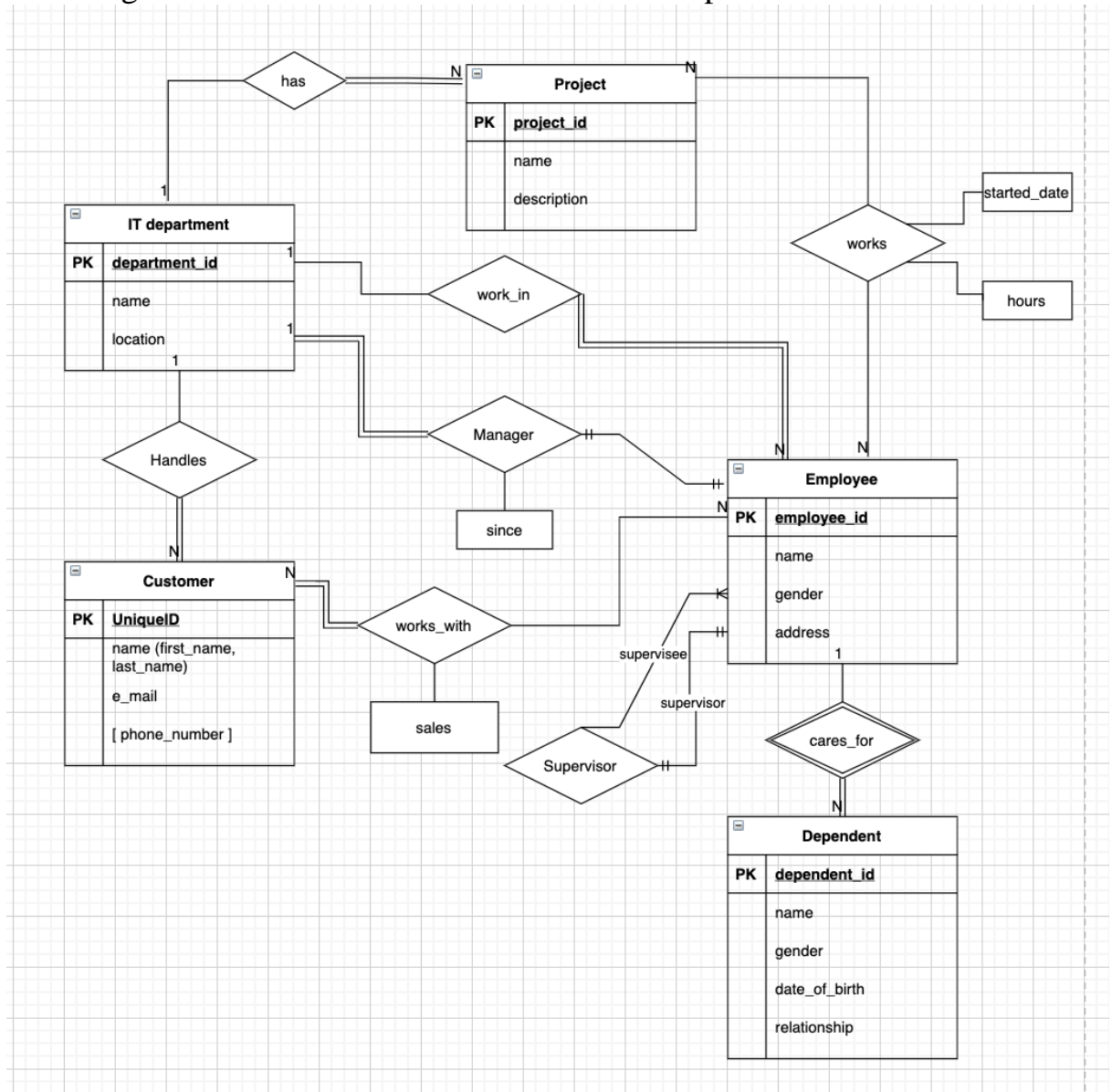


Figure 8 – ER data model for IT department

That's all of laboratory work 4.