

German International University
Faculty of Informatics and Computer Science
Dr. Iman Awaad
Eng. Ahmed Sherif
Amir Haythem
Mohamed Essam
Yassein Eldamasy

Software Engineering, Winter 2025
Practice Assignment Lab 4

Exercise 4-1

1. Create the Employee table using Postgres database.
The table should have the following columns:

- ID
- firstName
- middleName
- lastName
- country
- salary
- birthDate

Employees should be uniquely identified by their IDs.
No null entries are allowed in the table.

2. Insert seven records into the Employee table.
3. Write sql statement that delete table Employee

Solution:

```
1. CREATE TABLE Employee (
    ID serial PRIMARY KEY,
    firstName TEXT NOT NULL,
    middleName TEXT NOT NULL,
    lastName TEXT NOT NULL,
    country TEXT NOT NULL,
    salary INTEGER NOT NULL,
    birthDate DATE NOT NULL
);

2. INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Youssef', 'Kamal', 'Ahmed', 'Egypt', 10000, '1990-09-15');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
```

```

VALUES ('Amr', 'Talaat', 'Mostafa', 'UAE', 19000, '1992-09-15');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Omar', 'Talaat', 'Mostafa', 'Egypt', 19000, '1994-09-15');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Farida', 'Amr', 'Sameer', 'Qatar', 25000, '1997-09-15');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Donia', 'Ali', 'Mohamed', 'Egypt', 12000, '1999-07-07');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Ali', 'Mohamed', 'Ahmed', 'Qatar', 11500, '1980-07-06');

INSERT INTO Employee (firstName, middleName, lastName, country, salary, birthDate)
VALUES ('Dina', 'Ali', 'Mohamed', 'UAE', 10000, '1990-11-11');

3. DROP TABLE Employee;

```

Exercise 4-2

Using the employee table from Ex 4.1

1. Show all the details about the employees
2. Show the first name and salary of all the employees
3. Show distinct countries for the employees and sort them by country.
4. Show all the details about the employees who have a salary between 10000 and 15000.
5. Show the first name, country and salary of all the employees who have a salary more than 10000 and country is not Egypt.
6. Show all the details about the employees who were born after 11/2/1980 sorted by their country descendingly and within each country sorted by their salary ascendingly.
7. Select all employees with the country ending with an E.
8. Select all employees with the country starting with an E.
9. Select all employees with the country contains 'at'.
10. Select all employees with the country does not contains 'E'.
11. For all the Employees who earn between 10000 and 12000 update their first name to Ali, their last name to Mohammed and their birth date to 1/1/1987.
12. Delete all records of employees who has salary below 20000.

13. Delete all the entries in table Employee.

Solution:

```
1. SELECT * FROM Employee;
2. SELECT firstName, salary FROM Employee;
3. SELECT DISTINCT country FROM Employee ORDER BY country;
4. SELECT * FROM Employee WHERE salary >= 10000 AND salary <= 15000;
5. SELECT * FROM Employee WHERE country <> 'Egypt' AND salary > 10000;
6. SELECT * FROM Employee WHERE birthDate > '1980-02-11' ORDER BY country DESC, salary;
7. SELECT * FROM Employee WHERE country LIKE '%E';
8. SELECT * FROM Employee WHERE country LIKE 'E%';
9. SELECT * FROM Employee WHERE country LIKE '%at%'; â contains
10. SELECT * FROM Employee WHERE country NOT LIKE '%E%'; â does not contain

11. UPDATE Employee
SET firstName = 'Ali', lastName = 'Mohammed', birthDate = '1999-07-07'
WHERE salary BETWEEN 10000 AND 12000;

12. DELETE FROM Employee WHERE salary < 20000;
13. DELETE FROM Employee;
```

Exercise 4-3

Using the employee table from Ex 4.1

1. Show the number of records in the employee table.
2. Show the highest salary of an Egyptian Employee.
3. show the salary average of the Egyptian Employees.
4. show the average salary for each country and sort the results according to country.

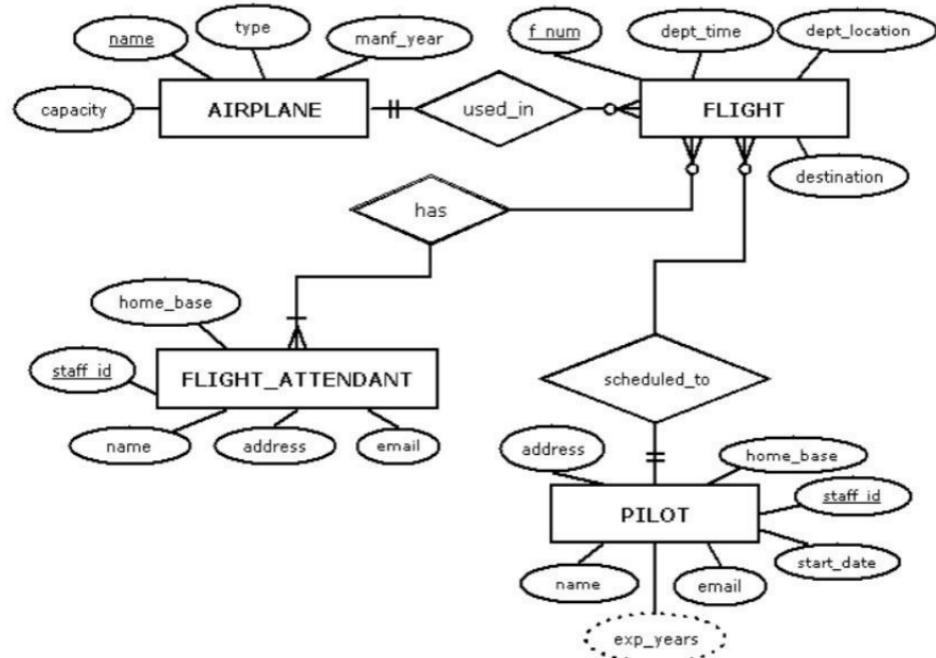
```
1. SELECT COUNT(*) FROM Employee;
2. SELECT MAX(salary) FROM Employee WHERE country = 'Egypt';
3. SELECT AVG(salary) FROM Employee WHERE country = 'Egypt';
4. SELECT country , AVG(salary) FROM Employee GROUP BY country order by country;
```

Exercise 4-4

Masreya Airlines (MA) is an airlines company that manages daily flights between cities around the world. In order to optimize its performance and due to the frequent scheduling problems MA has faced, the manager decided to build a database system that can help improve the company's performance. Although important, the database built should not be big in size since the budget for buying the required software and to hire the specialized staff is low. The system should keep track of airplanes, flights and flight employees. Each airplane has a unique name, type, manufacturing year, and a capacity. Each flight has a unique flight number, a departure time, a destination and a departure location (gate 1, gate 2...). Each scheduled flight has exactly one pilot and at least one flight attendant (steward, stewardess). Pilots can be scheduled to more than one flight. Pilots and flight attendants have respective home bases, to which they return at the end of an assigned flight. For both pilots and flight attendants, other information should be stored such as the staff ids and personal data (name, address, email...). Only in the case of pilots, it is important to keep track of the number of experience years.

1. Draw an Entity-Relationship-Diagram (ERD).

Solution:



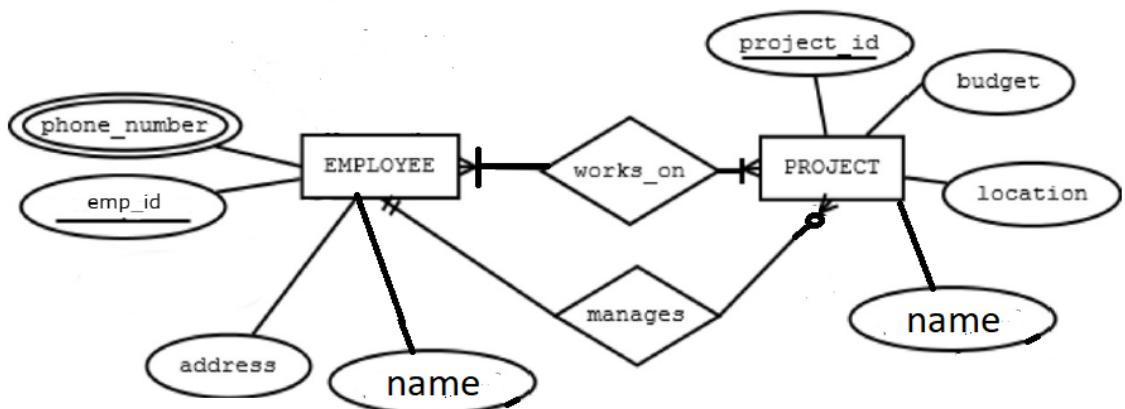
Exercise 4-5

We need to store some information about a company. It contains different employees. Each employee has name. They also have an address and phone number(s). The company has different projects. Each project has a name, location and budget. Employees are involved on at least one project. Each project has a manager.

1. Draw an Entity-Relationship-Diagram (ERD).
2. Create a Database tables from Entity-Relationship-Diagram using sql statement .

Solution:

1.



```

2. create table Employee(
    emp_id serial primary key,
    name text not null,
    address text not null,
    );
    create table project(
    project_id serial primary key,
    name text not null,
    budjet integer not null,
    location text not null,
    emp_id integer not null,
    foreign key(emp_id) references employee(emp_id)
    );
    create table works_on(
    emp_id integer,
    project_id integer,
    foreign key(emp_id) references employee(emp_id),
    foreign key(project_id) references project(project_id),
    primary key(emp_id , project_id)
    );
    create table employee_phone(
    emp_id integer,
    phone_number text,
    foreign key(emp_id) references employee(emp_id),
    primary key(emp_id , phone_number)
    );

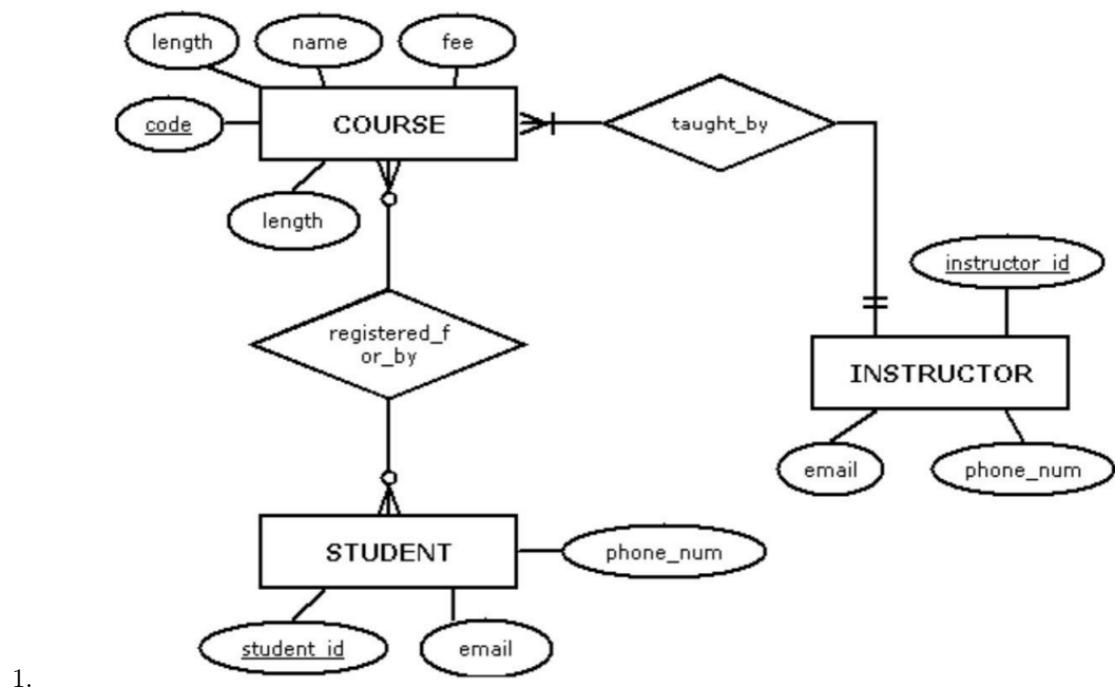
```

Exercise 4-6

ITI Center is a training center that provides programming courses. Many courses are popular, each of which has a unique code, name and fee. Introduction to UNIX and Java Programming are two of the most popular courses. The courses offered vary in length from one week to a month. The center keeps track of each instructor's name, phone number and email. Each course is taught by only one instructor while an instructor may teach several courses. The students can attend several courses over time, and may register for many courses at the same time as long as their timings do not conflict. ITI Center also records contact information about the students and instructors, although some refuse to give their phone numbers.

1. Draw an Entity-Relationship-Diagram (ERD).
2. Create a Database tables from Entity-Relationship-Diagram using sql statement .

Solution:



1.

Solution:

2.

```
CREATE TABLE Course (
    code TEXT PRIMARY KEY,
    name TEXT NOT NULL,
    fee INTEGER NOT NULL,
    length INTEGER NOT NULL,
    instructor_id INTEGER NOT NULL
);

CREATE TABLE Student (
    student_id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    email TEXT NOT NULL,
    phone_num INTEGER NOT NULL
);

CREATE TABLE Instructor (
    instructor_id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    email TEXT NOT NULL,
    phone_num INTEGER NOT NULL
);

CREATE TABLE TaughtBy (
    instructor_id INTEGER NOT NULL,
    code TEXT NOT NULL,
    PRIMARY KEY (instructor_id, code),
    FOREIGN KEY (instructor_id) REFERENCES Instructor(instructor_id),
    FOREIGN KEY (code) REFERENCES Course(code)
);

CREATE TABLE RegisteredForBy (
    student_id INTEGER NOT NULL,
    code TEXT NOT NULL,
    PRIMARY KEY (student_id, code),
    FOREIGN KEY (student_id) REFERENCES Student(student_id),
    FOREIGN KEY (code) REFERENCES Course(code)
);
```