You should work on the following assignments in fixed teams of two. Please note that *every* team member must be able to explain *all* solutions of the team of two. Please submit only one solution in our moodle room for each team of two. The submission must be a PDF file with the name and matriculation number. Solutions must be in digital format with intermediate steps and detailed explanations. You can use any drawing tool of your choice to create the diagrams. If you have questions or need any support, help each other, and use the forum in our moodle room.

It is recommended that you also familiarize yourself with the assignments that do not have to be submitted and will be worked on during the lab to allow a more effective participation.

Deadline to upload your solution for (ALL) assignments 1, 2, 3, 4, and 5:
Friday, 11:59 pm bevor the laboratory.

Assignment 1: SQL-statements for the Student Information System

Consider the following relational schema for the Student Information System (analogous to the Assignments from Lab 1):

STUDENT(studentID, fistName, lastName, dob, programID(FK))
PROGRAM(programID, name, requiredCPs)
COURSE(courseID, name, description, creditPoints, programID(FK))
ATTEMPTS(studentID(FK),courseID(FK), year, term, grade)
PREREQUISITE(advancedCourseID(FK), prerequisiteCourseID (FK))

1. Write SQL-statements that create the corresponding tables. Come up with reasonable constraints and datatypes for the fields of the tables.
2. Write SQL-queries that insert example data into your created tables. Make sure that each table contains at least 2 rows of data. Here are some sample data.

Table **STUDENT**

studentID	firstName	lastName	dob	programID
123456	John	Wayne	11.05.1998	1
234567	Anna	Meyer	13.02.1999	1
…	…	…	…	…

Table **PROGRAM**

programID	Name	requiredCPs
1	Information Engineering	120
2	Renewable Energies	110
…	…	…

Table **COURSE**

courseID	Name	Description	creditPoints	programID
4	MA1	Mathematics 1	8	1
9	MA2	Mathematics 2	8	1
13	SS1	Signals and Systems 1	6	1
15	DB	Databases	6	1
…	…	…	…	…

Table **PREREQUISITE**

advancedCourse	prerequisiteCourse
9	4
13	9
13	4
…	…

Table **ATTEMPTS**

studentID	courseID	Year	Term	grade
123456	4	2021	1	7
123456	9	2021	2	9
123456	13	2022	1	3
123456	13	2022	2	6
…	…	…	…	…

3. Write a SQL-query for the created database that returns all students (first name + last name) that study the program “Information Engineering”.
4. Write a SQL-query that returns the name of all courses that have prerequisite courses.
5. Write a SQL-query that returns the sum of all credit points successfully achieved by student “John Wayne”. Keep in mind that the credit points only count when the student has an attempt with a grade of 5 or more points.
6. A student needs to be removed from the database. Write SQL-statements to remove the student with the name “John Wayne” from the database.

Assignment 2: SQL-statements for a Shipping company

A shipping company wants to use a SQL-database to keep track of its ships and employed sailors based on the following relation schema:

HARBOR (harborID, location, establishedIn)
SAILOR (sailorID, lastName, dob, trainedAt(FK -> harborID))
SHIP (shipID, name, grossWeight, launchDate, baseHarbor(FK -> harborID))
HIRE (sailor(FK -> sailorID), ship(FK -> shipID), startOfService, annualSalary)

You can use the provided SQL-script for creating the tables and inserting some data in the tables.

1. Create a SQL-query that returns the dob (date of birth) of sailors in descending order that were hired on August 3rd, 2012.
2. Create a SQL-query that returns all information of sailors that were hired between July 3rd, 2011, and September 3rd, 2012, and whose last name starts with a 'J'.
3. Create a SQL-query that returns for each ship the sum of the annual salary of every sailor who is hired for that ship.
4. Create a SQL-query that returns the location of all harbors that are not base harbor to any ship in the database.
5. Create a SQL-query that returns the shipId, ship name and the number of sailors who are hired on the ship and earn maximum 42.000$.
6. Describe in your own words the result of the following query:

```
SELECT DISTINCT h1.location
FROM SHIP s1, SHIP s2, HARBOR h1, HARBOR h2
WHERE s1.baseHarbor = h1.harborID
       AND s2.baseHarbor = h2.harborID
       AND s1.launchDate = s2.launchDate
       AND h1.location = h2.location
       AND h1.harborID != h2.harborID;
```

Assignment 3: SQL-statements for the COMPANY example from Elmasri also used in the lecture

Let's have a look on the COPMPANY example from the book „Fundamentals of Database Systems" from Elmasri which is also used in the lecture. Given is the database schema in Figure 1 and the database state in Figure 2.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 1: Schema diagram for the COMPANY relational database schema (see Elmasri, page 71)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Figure 2: A possible database state for the COMPANY relational database schema (see Elmasri, page 72)

Write SQL statement for the following tasks:

1. Retrieve the names of all employees in department 5 who work more than 10 hours per week on a project.
2. List the names of all employees who have a dependent with the same first name as themselves.
3. Find the names of all employees who are directly supervised by ‘Franklin Wong’.
4. Suppose that the EMPLOYEE table’s constraint EMPSUPERFK as specified in Figure 3 on the next page is changed to read as follows:

```
CONSTRAINT EMPSUPERFK
FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
ON DELETE CASCADE ON UPDATE CASCADE;
```

   Answer the following questions:

   - What happens when the following command is run on the database state shown in Figure 2?

```
DELETE FROM EMPLOYEE WHERE Lname = ‘Borg’ ;
```

   - Is it better to **CASCADE** or **SET NULL** in case of EMPSUPERFK constraint **ON DELETE**?

```
CREATE TABLE EMPLOYEE
    (  . . . ,
       Dno        INT            NOT NULL        DEFAULT 1,
    CONSTRAINT EMPPK
       PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
       FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                    ON DELETE SET NULL          ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
       FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                    ON DELETE SET DEFAULT       ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
    (  . . . ,
       Mgr_ssn    CHAR(9)        NOT NULL        DEFAULT '888665555',
       . . . ,
    CONSTRAINT DEPTPK
       PRIMARY KEY(Dnumber),
    CONSTRAINT DEPTSK
       UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
       FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                    ON DELETE SET DEFAULT   ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
    (  . . . ,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                    ON DELETE CASCADE           ON UPDATE CASCADE);
```

Figure 3: Database schema 2 (see Elmasri, page 95)

5. For each project, list the project name and the total hours per week (by all employees) spent on that project.
6. Retrieve the average salary of all female employees.
7. Write SQL statements to create a table EMPLOYEE_BACKUP to back up the EMPLOYEE table shown.
8. For each department, whose average employee salary is more than $30,000, retrieve the department name and the number of employees working for that department.

Assignment 4: Relational Algebra vs. SQL query for a Cinema Database

The following excerpt from a database schema models a database about a cinema. The following assignments are to be answered in the form of relational algebra AND additionally in the form of an SQL query.

Movies (FilmID (PK), Title, Director, Release Year, Genre)
Actors (ActorID (PK), First Name, Last Name, Birthdate)
Screenings (ScreeningID (PK), FilmID (FK), Cinema Hall, Date, Time)
Reservations (ReservationID (PK), ScreeningID (FK), Seat, Customer Name, Booking Date)

1. Do a projection to see a list of all movie titles and their directors.
2. Display a list of actors (first name, last name) and their roles in a specific movie (e.g., "FilmXYZ").
3. Find all movies that will be shown in the screenings (ScreeningID) for the movie theater "HallA" on 2024-01-30 at 19:00.
4. Create a table with information about all reservations made by customers with the last name "Schmidt", including the movie title and seat number.

Assignment 5: Relational Algebra vs. SQL query for a Weather Database

The following excerpt from a database schema models a database about a weather station. The following assignments are to be answered in the form of relational algebra AND additionally in the form of an SQL query.

CITY (CityID, Name, Country)
WEATHER_DATA (DataID, CityID (FK), Date, Temperature, Humidity, Precipitation, WindSpeed)

1. Which cities have an average daily temperature above 25°C in August 2023?
2. Which cities experienced no precipitation on any day in July 2023?
3. On which particular day did the cities have the highest wind speed?
4. Which cities recorded the highest temperature?
5. Which cities had the lowest humidity in May 2023?