EXAMPLE

Database Design. Introduction to SQL.
Final Exam.

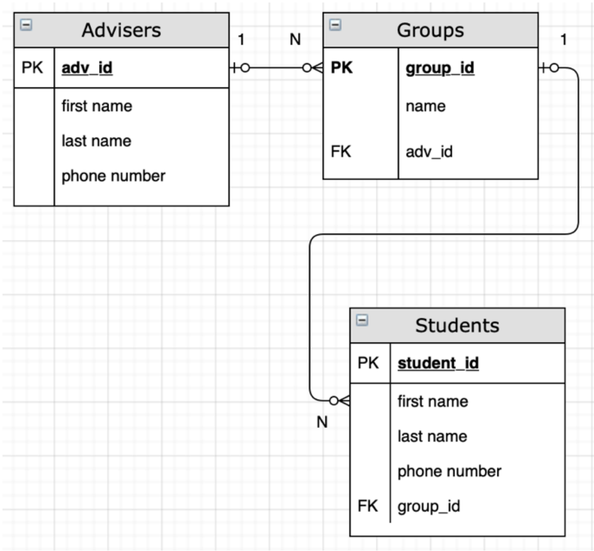
Upload: *.doc or *.pdf with answers for all questions (ER-diagram, SQL code)
Instruction: During the exam, students can use DBMS PostgreSQL and software to draw the ER-diagram (gliffy.com, creately.com, draw.io, lucidchart.com).

Task:
The database contains information about students, advisers and student groups. Every student can be enrolled only in one group, while one group contains about 20 students. Each adviser can lead several groups, but each group has only one adviser.
The database stores at least the following information: about students - student ID, name, surname, phone number, about advisers - name and surname, phone number. The group table stores information about the name of the group.
Moreover, first and last names must be filled in for all students and advisers, for this, a constraint should be provided in the database.

1. Construct the ER-diagram (with Crow's foot notation) of the database using its description. The ER-diagram should be in the 3NF and include all entities with all attributes (including Primary keys for all tables and Foreign keys), relationships with noted types.
2. Write the SQL code to create this database and fill in information in all tables (min 2 rows in each of them).
3. Write the SQL code to implement the following queries:
    a. Show student id of all students whose last names begin with A.
    b. Show the group names and the number of students corresponding to these groups. The number of students should be the result of the function, not the value stored in the table. Only groups where the number of students is greater than or equal to 20 should be shown. Sort the list by group name alphabetically.

ANSWER

1.



2.

```
CREATE TABLE Advisers(
        adv_id int PRIMARY KEY,
        first_name varchar(20) NOT NULL,
        last_name varchar(20) NOT NULL,
        phone_number bigint
);

CREATE TABLE Groups(
        group_id int PRIMARY KEY,
        name varchar(20),
        adv_id int REFERENCES Advisers(adv_id)
);

CREATE TABLE Students(
        student_id int PRIMARY KEY,
        first_name varchar(20) NOT NULL,
        last_name varchar(20) NOT NULL,
        phone_number bigint,
        group_id int REFERENCES Groups(group_id)
);
```

```
INSERT INTO Advisers VALUES (1,'Adv_First1','Adv_Last1',12345678),
(2,'Adv_First2','Adv_Last2',2345754);
INSERT INTO Groups VALUES (1,'CSSE-1',1), (2,'CSSE-2',2);
INSERT INTO Students VALUES (1,'Stud_First1','Stud_Last1',22345678, 1),
(2,'Stud_First2','Stud_Last2',3345754, 2);
```

3.

```
--a
SELECT student_id
FROM Students
WHERE last_name LIKE 'A%';

--b
SELECT g.name, count(*)
FROM Groups g, Students s
WHERE g.group_id = s.group_id
GROUP BY g.name
HAVING count(*)>=20
ORDER BY g.name ASC;
```