

1 Review

Review the following terms.

1. Null value

2. Modification

delete
insert
update
case

3. Join types

natural join
join...using
join...on
left, right, full outer join

4. Set membership

in
not in

5. Set comparison

some clause
all clause
exists clause
not exists clause
unique clause

6. View definition

create view

7. Transactions

begin
commit
rollback

8. Constraints

check
Referential integrity

Answer

Review the slides or the textbook (Chapters 3, 4, 5).

2 Advanced SQL Queries

Consider the following relational schema, which obviously does not describe the standard situation at Aalborg University.

We assume that tutors are responsible for one or multiple study groups, students individually (not per group) hand in solutions for exercise sheets and receive individual grades in terms of the number of achieved points per sheet. Some of the tutors are more experienced (senior) than others.

```
student: {[sid: int, firstname: string, lastname: string, semester: int, birthdate: date]}  
tutor: {[tid: int, firstname: string, lastname: string, issenior: boolean]}  
studygroup: {[gid: int, tid → tutor, weekday: string, room: string, starttime: time]}  
exercisesheet: {[eid: int, maxpoints: int]}  
handsin: {[sid → student, eid → exercisesheet, achievedpoints: int]}  
member: {[sid → student, gid → studygroup]}
```

1. List the first and last names of all students who are at least in one studygroup.

Answer

```
SELECT firstname, lastname  
FROM student S  
WHERE S.sid IN (SELECT M.sid  
                FROM member M);
```

2. List the first and last names of the students that obtained higher number of points than at least one of others for exercise sheet 1 (eid = 1).

Answer

```
SELECT firstname, lastname  
FROM student NATURAL JOIN handsin  
WHERE eid = 1 AND  
      achievedPoints > SOME ( SELECT achievedPoints  
                            FROM handsin  
                            WHERE eid = 1);
```

3. Find the first and last names of all students who have study group on Wednesday or Friday.

Answer

```
SELECT s.firstname, s.lastname  
FROM student s  
      JOIN member m ON s.sid = m.sid  
      JOIN studygroup g ON g.gid = m.gid  
WHERE g.weekday = 'Wednesday'  
      OR g.weekday = 'Friday';
```

4. Create a view of student without their birthday, rename it as studentview.

Answer

```
CREATE VIEW studentview AS
SELECT sid, firstname, lastname, semester
FROM student;
```

5. Create an assistant table which contains attributes: (aid int), (sid, int), (coursename varchar(20)), (salary, numeric(8,2)).

- a) The primary key is aid.
- b) The sid references student.
- c) The course name should not be empty.
- d) The salary should not be less than 5000.

Answer

```
CREATE TABLE assistant (
    aid int,
    sid int,
    coursename varchar(20) NOT NULL,
    salary numeric(8,2),
    PRIMARY KEY (aid),
    FOREIGN KEY (sid) REFERENCES student,
    CHECK (salary >= 5000));
```

3 Test your solutions using PostgreSQL