```
-- create a table
create table instructor (
    ID char(5),
    name varchar(20) not null,
    dept_name varchar(20),
    salary numeric(8, 2),
    primary key (ID),
    foreign key (dept_name) references department
);


create table student (
    ID varchar(5), name varchar(20) not null, dept_name varchar(20), tot_cred numeric(3, 0),
    primary key (ID),
    foreign key (dept_name) references department
);

create table takes (
    ID varchar(5),
    course_id varchar(8),
    sec_id varchar(8),
    semester varchar(6),
    year numeric(4,0),
    grade varchar(2),
    primary key(ID, course_id, sec_id, semester, year),
    foreign key (ID) references student,
    foreign key (course_id, sec_id, semester, year) references section
);

create table course (
    course_id varchar(8),
    title varchar(50),
    dept_name varchar(20),
    credits numeric(2,0),
    primary key(course_id),
    foreign key (dept_name) references department
);

-- Insert
insert into instructor values ('10211', 'Smith', 'Biology', 66000);

-- Delete
/* Remove all tuples from the student relation */
delete from student

-- Select
/* find the names of all instructors */
select name from instructor

/* To find all instructors in Comp.Sci. dept */
select name from instructor
where dept_name = 'Comp.Sci.';

/* To find all instructors in Comp.Sci. dept with salary > 70000 */
select name from instructor
where dept_name = 'Comp.Sci.' and salary > 70000;

-- From
/* Find the Cartesian product instructor X teaches */
select * from instructor, teaches;

/* Find the names of all instructors who have taught some course and the course_id */
select name, course_id from instructor, teaches
where instructor.ID = teaches.ID;
```

```
/* Find the names of all instructors in the Art depatment who have taught some course and
the course_id */
select name, course_id from instructor, teaches
where instructor.ID = teaches.ID and instructor.dept_name = 'Art';

/* Find the names of all instructors who have higher salary than some instructor in
'Comp.Sci' */
select distinct T.name from instructor as T, instructor as S
where T.salary > S.salary and S.dept_name = 'Comp.Sci';

-- String Operations (%) and (_)
/* Find the names of all instructors whose name includes the substring "dar" */
select name from instructor
where name like '%dar%';

-- order by
/* List in alphabetic order the names of all instructors */
select distinct name from instructor
order by name;

-- between
/* Find the names of all instructors with salary between $90,000 and $100,000 */
select name from instructor
where salary between 90000 and 100000;

-- Set Operations
/* Find courses that ran in Fall 2017 or in Spring 2018 */
(select course_id from section where sem = 'Fall' and year = 2017)
union
(select course_id from section where sem = 'Spring' and year = 2018);

/* Find courses that ran in Fall 2017 and in Spring 2018 */
(select course_id from section where sem = 'Fall' and year = 2017)
intersect
(select course_id from section where sem = 'Spring' and year = 2018);

/* Find course that ran in Fall 2017 but not in Spring 2018 */
(select course_id from section where sem = 'Fall' and year = 2017)
except
(select course_id from section where sem = 'Spring' and year = 2018);

-- null values
/* Find all instructors whose salary is null */
select name from instructor
where salary is null;

-- Aggregate Functions
/* Find the average salary of instructors in the Computer Science department */
select avg(salary) from instructor
where dept_name = 'Comp.Sci.';

/* Find the total number of instructors who teach a course in the Spring 2018 semester */
select count(distinct ID) from teaches
where semester = 'Spring' and year = 2018;

/* Find the number of tuples in the course relation */
select count(*) from course;

/* Find the average salary of instructors in each department */
select dept_name, avg(salary) as avg_salary from instructor
group by dept_name;


/* Find the names and average salaries of all departments whose average salary is greater
```

```
than 42000 */
select dept_name, avg(salary) as avg_salary from instructor
group by dept_name
having avg(salary) > 42000;


-- subqueries
/* Find courses offered in Fall 2017 and in spring 2018 */
select distinct course_id from section
where semester = 'Fall' and year = 2017 and
course_id in (select course_id from section
    where semester = 'Spring' and year = 2018);

/* Find courses offered in Fall 2017 but not in Spring 2018 */
select distinct course_id from section
where semester = 'Fall' and year = 2017 and
course_id not in (select course_id from section
    where semester = 'Spring' and year = 2018);

/* Name all instructors whose name is neither "Mozart" nor "Einstein" */
select distinct name from instructor
where name not in ('Mozart', 'Einstein');

/* Find the total number of (distinct) students who have taken course sections taught by the
instructor with ID 10101 */
select count (distinct ID) from takes
where (course_id, sec_id, semester, year) in
(select course_id, sec_id, semester, year from teaches
    where teaches.ID = 10101);


-- Some
/* Find names of instructors with salary greater than that of some instructor in the Biology
department */
select distinct T.name from instructor as T, instructor as S
where T.salary > S.salary and S.dept name = 'Biology';

/* Same query using > some clause */
select name from instructor
where salary > some (select salary from instructor
where dept_name = 'Biology');

-- All clause
/* Find the names of all instructors whose salary is greater than the salary of all
instructors in the Biology department */
select name from instructor
where salary > all (select salary from instructor
where dept_name = 'Biology');

-- Exists
/* Find all courses taught in both the Fall 2017 semester and in the Spring 2018 semester */
select course_id from section as S
where semester = 'Fall' and year = 2017 and
exists (select * from section as T
where semester = 'Spring' and year = 2018
and S.course_id = T.course_id);

-- unqiue
/* Find all courses that were offered at most once in 2017 */
select T.course_id from course as T
where unique (select R.course_id from section as R
where T.course_id = R.course_id and R.year = 2017);

-- subqueries in the from clause
/* Find the average instructors salaries of those departments where the average salary is
```

```
greater than $42,000 */
select dept_name, avg_salary
from (select dept_name, avg(salary) as avg_salary
from instructor
group by dept_name)
where avg_salary > 42000;

/* Another way to write the above query */
select dept_name, avg_salary
from (select dept_name, avg(salary)
from instructor
group by dept_name)
as dept_avg(dept_name, avg_salary)
where avg_salary > 42000;

-- with
/* Find all departments with the maximum budget */
with max_budget(value) as
(select max(budget) from department)
select department.name
from department, max_budget
where department.budget = max_budget.value;

-- Scalar Subquery (is used where a single value is expected)
/* List all departments along with the number of instructors in each department */
select dept_name,
(select count(*) from instructor
where department.dept_name = instructor.dept_name) as num_instructors from department;

-- delete
/* delete all instructors */
delete from instructors

/* delete all instructors from Finance department */
delete from instructor
where dept_name = 'Finance';

/* Delete all tuples in the instructor relation for those instructors associated with a
department located in the Watson Building */
delete from instructor
where dept_name in (select dept_name from department
where building = 'Watson');


/* Delete all instructors whose salary is less than the average salary of instructors */
delete from instructor
where salary < (select avg(salary) from instructor);


-- insertion
/* Add a new tuple to course */
insert into course
values ('CS-427', 'Database Systems', 'Comp.Sci.', 4);

/* or equivalently */
insert into course (course_id, title, dept_name, credits)
values ('CS-437', 'Database Systems', 'Comp.Sci.' 4);

/* Add a new tuple to student with tot_creds set to null */
insert into student
values ('3003', 'Green', 'Finance', null);

/* Make each student in the Music department who has earned more than 144 credit hours an
instructor in the Music department
with a salary of $18,000 */
```

```
insert into instructor
select ID, name, dept_name, 18000 from student
where dept_name = 'Music' and total_cred > 144;


-- Updates
/* Give a 5% salary raise to all instructors */
update instructor
set salary = salary * 1.05;

/* Give a 5% salary raise to those instructors who earn less than 70000 */
update instructor
set salary = salary * 1.05
where salary < 70000;

/* Give a 5% salary raise to instructors whose salary is less than average */
update instructor
set salary = salary * 1.05
where salary < (select avg(salary) from instructor);


/* Increase salaries of instructors whose salary is over $100,000 by 3% and all others by 5%
*/
update instructor
set salary = salary * 1.03
where salary > 100000
update instructor
set salary = salary * 1.05
where salary <= 100000;

/* Same query as before but with case statement */
update instructor
set salary = case
when salary <= 100000 then salary * 1.05
else salary * 1.03
end
```