SQL Queries and Answers for Exam Preparation

Find names of all instructors:

SELECT name FROM instructor;

Find names of all departments (distinct):

SELECT DISTINCT dept_name FROM instructor;

Display all attributes of instructor:

SELECT * FROM instructor;

Find instructor names and their monthly salaries:

SELECT ID, name, dept_name, salary/12 FROM instructor;

Find instructors from Comp. Sci. department with salary > 80000:

SELECT name FROM instructor WHERE dept_name = 'Comp. Sci.' AND salary > 80000;

Find Cartesian product of instructor and teaches:

SELECT * FROM instructor, teaches;

Find instructors who have taught courses (names & course_id):

SELECT name, course_id FROM instructor, teaches WHERE instructor.ID = teaches.ID;

Find course details of Comp. Sci. department:

SELECT section.course_id, semester, year, title FROM section, course WHERE section.course_id = course.course_id AND dept_name = 'Comp. Sci.';

Using natural join:

SELECT name, course_id FROM instructor NATURAL JOIN teaches;

Find instructor names containing 'dar':

SELECT name FROM instructor WHERE name LIKE '%dar%';

List instructor names in alphabetical order:

SELECT DISTINCT name FROM instructor ORDER BY name;

List instructor names in descending order:

SELECT name FROM instructor ORDER BY name DESC;

Courses offered in Fall 2009 or Spring 2010:

```
(SELECT course_id FROM section WHERE sem = 'Fall' AND year = 2009)
UNION
(SELECT course_id FROM section WHERE sem = 'Spring' AND year = 2010);
```

Courses in both Fall 2009 and Spring 2010:

```
(SELECT course_id FROM section WHERE sem = 'Fall' AND year = 2009)
INTERSECT
(SELECT course_id FROM section WHERE sem = 'Spring' AND year = 2010);
```

Courses in Fall 2009 but not in Spring 2010:

```
(SELECT course_id FROM section WHERE sem = 'Fall' AND year = 2009)
EXCEPT
(SELECT course_id FROM section WHERE sem = 'Spring' AND year = 2010);
```

Average salary of CS instructors:

```
SELECT AVG(salary) FROM instructor WHERE dept_name = 'Comp. Sci.';
```

Count distinct instructors in Spring 2010:

```
SELECT COUNT(DISTINCT ID) FROM teaches WHERE semester = 'Spring' AND year = 2010;
```

Count total tuples in course table:

```
SELECT COUNT(*) FROM course;
```

Average salary per department:

```
SELECT dept_name, AVG(salary) FROM instructor GROUP BY dept_name;
```

Departments with avg salary > 42000:

```
SELECT dept_name, AVG(salary) FROM instructor GROUP BY dept_name HAVING AVG(salary) > 42000;
```

Instructors earning more than some in Biology:

```
SELECT name FROM instructor WHERE salary > SOME (SELECT salary FROM instructor WHERE dept_name = 'Biology');
```

Instructors earning more than all in Biology:

```
SELECT name FROM instructor WHERE salary > ALL (SELECT salary FROM instructor WHERE dept_name = 'Biology');
```

Departments with highest avg salary:

SELECT dept_name FROM instructor GROUP BY dept_name HAVING AVG(salary) >= ALL (SELECT AVG(salary) FROM instructor GROUP BY dept_name);

Courses taught by instructor 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);

Delete all instructors:

DELETE FROM instructor;

Delete Finance dept instructors:

DELETE FROM instructor WHERE dept_name = 'Finance';

Delete instructors with salary between 13000 and 15000:

DELETE FROM instructor WHERE salary BETWEEN 13000 AND 15000;

Insert new course:

INSERT INTO course VALUES ('CS-437', 'Database Systems', 'Comp. Sci.', 4);

Insert students from Music dept into instructor:

INSERT INTO instructor SELECT ID, name, dept_name, 18000 FROM student WHERE dept_name = 'Music' AND tot_cred > 144;

Update all instructors' salary by 5%:

UPDATE instructor SET salary = salary * 1.05;

Conditional update using CASE:

UPDATE instructor SET salary = CASE WHEN salary <= 100000 THEN salary * 1.05 ELSE salary * 1.03 END;

Update student credits with scalar subquery:

UPDATE student S SET tot_cred = (SELECT SUM(credits) FROM takes NATURAL JOIN course WHERE S.ID = takes.ID AND grade <> 'F' AND grade IS NOT NULL);