SQL practice questions #2

Write SQL queries to answer the following questions using the COMPANY database.

1. Find the SSN, first and last names of all employees with dependents.

   ```
   SELECT DISTINCT ssn, fname, lname
   FROM EMPLOYEE, DEPENDENT
   WHERE ssn = essn;
   ```

   Employees may have more than one dependent, but nothing in the question indicates that we want duplicates in the answer. SELECT DISTINCT will eliminate duplicates.

2. Find the SSN, first and last names of all employees who have a dependent of the same sex.

   ```
   SELECT DISTINCT ssn, fname, lname
   FROM EMPLOYEE, DEPENDENT
   WHERE ssn = essn AND EMPLOYEE.sex = DEPENDENT.sex;
   ```

3. Find the SSN, first, and last names and the supervisor’s first and last name of all employees who work for department 3.

   ```
   SELECT E.ssn, E.fname, E.lname, S.fname AS SupervisorFirst, S.lname AS SupervisorLast
   FROM EMPLOYEE E, EMPLOYEE S
   WHERE E.dno = 3 and S.ssn = E.super_ssn;
   ```

4. Find the first and last names of supervisors of employees in department 3, the number of employees for each supervisor, and the average salary of the employees for each supervisor.

   ```
   SELECT S.fname, S.lname, count(*), format(avg(E.salary), 2)
   FROM EMPLOYEE E, EMPLOYEE S
   WHERE E.dno = 3 and S.ssn = E.super_ssn
   GROUP BY S.fname, S.lname;
   ```

   Note the format function isn’t in the slides, but it’s handy for the aggregate functions. It simply rounds the input to the given number of decimals places. You won’t need to know this for the class, just remember it exists for your future use. Try running the answer both with and without the format.

5. For each supervisor with more than 30 employees, list the supervisors first and last names, the count of the number of employees, and the total salary of those employees.

```
SELECT S.fname, S.lname, count(*), format(sum(E.salary), 2)
FROM EMPLOYEE E, EMPLOYEE S
WHERE S.ssn = E.super_ssn
GROUP BY S.fname, S.lname
HAVING count(*) > 30;
```

6. List the first and last name, salary, and department of every employee who does not work for department 3 but makes more than the average salary of department 3 employees, sorted by department.

```
SELECT Lname, Fname, Salary, Dno
FROM   EMPLOYEE
WHERE  Dno <> 3 AND Salary > ALL (SELECT avg(salary)
                                  FROM   EMPLOYEE
                                  Dno = 3)
ORDER BY Dno;
```

7. List the department name and department number of all departments which have a project in the same location as one of the locations for that department. Also list that shared location.

   The key to this one is making sure that when the locations are the same it’s also the same department. Also note that we need to specify which table we’re getting the dnumber column from, even though the natural join means the value in either table with that column is the same. (Note that this query correctly returns the empty set).

```
SELECT dname, DEPARTMENT .dnumber, dlocation
FROM DEPARTMENT
JOIN PROJECT ON dnum = dnumber
NATURAL JOIN DEPT_LOCATIONS
WHERE plocation = dlocation;
```

8. Find the SSN, first and last names of all employees who are managers who do not have a dependent with a relationship of ‘spouse’

   ```
   SELECT ssn, fname, lname
   FROM
   EMPLOYEE, DEPARTMENT
   WHERE ssn = mgrssn
   EXCEPT  (SELECT ssn, fname, lname
   FROM EMPLOYEE, DEPENDENT
   WHERE ssn = essn AND relationship = 'spouse');
   ```

   In the company database, this gives 19 rows in the result.

   If instead you try to avoid EXCEPT and join EMPLOYEE, DEPARTMENT, and DEPENDENT using a WHERE clause that includes “relationship <> 'spouse'”, you will only get 8 rows returned (or 7 if you use DISTINCT), because employees with no dependents are excluded from the join. Additionally, 2 of those 7 rows are managers who have a spouse, but end up included because they have other dependents as well.