

## CSG1207/CSI5135: Systems and Database Design

### Lab 09 - Solutions

#### Standard Disclaimer

*Many questions you encounter in this and other labs have more than one solution which is valid and correct. There are often numerous ways to achieve the same results in an SQL query.*

*The solutions provided here may NOT be the only correct answers to the questions. If you have arrived at solution to a lab task that differs substantially from what is provided here and would like feedback on your solution, please contact your tutor.*

#### Lab Tasks

- Q1.**
- ```
SELECT first_name + ' ' + last_name AS 'full_name',
       salary, department_name
FROM employee AS e INNER JOIN department AS d
  ON e.department_id = d.department_id
WHERE salary > 10000;
```
- Q2.**
- ```
SELECT DISTINCT job_title
FROM employee AS e INNER JOIN job AS j
  ON e.job_id = j.job_id
WHERE e.department_id = 50;
```
- Q3.**
- ```
SELECT last_name, commission_pct, department_name, city
FROM employee AS e LEFT OUTER JOIN department AS d
  ON e.department_id = d.department_id
  LEFT OUTER JOIN location AS l
    ON d.location_id = l.location_id
WHERE commission_pct IS NOT NULL
ORDER BY commission_pct DESC;
```

Note: If INNER JOINs are used, the results will omit one of the employees, who has a commission but does not work in a department.

- Q4.** A LEFT OUTER JOIN returns matched rows and unmatched rows due to *NULL* values, in the *left* table. A RIGHT OUTER JOIN returns matched rows, and unmatched rows due to *no matches*, in the *right* table. A FULL OUTER JOIN returns matched rows, and unmatched rows due to *NULLs* in *left* table *and no matches* in *right* table. The word “OUTER” is optional since it only makes sense to specify LEFT, RIGHT or FULL for an outer join. (Copied from slide 20 of the Module 9’s lecture)

- Q5.**
- ```
SELECT country_name, region_name
```

```
FROM country AS c RIGHT OUTER JOIN region AS r
ON r.region_id = c.region_id;
```

**Q6.** First join condition should involve e.job\_id, not e.employee\_id. "FROM employee AS e" should not be repeated for second join. Should be using a LEFT OUTER JOIN not a RIGHT OUTER JOIN. Fixed query:

```
SELECT last_name, job_title, department_name
FROM employee AS e INNER JOIN job AS j
ON e.job_id = j.job_id
LEFT OUTER JOIN department AS d
ON e.department_id = d.department_id;
```

**Q7.**

```
SELECT last_name, salary, job_title, min_salary
FROM employee AS e INNER JOIN job AS j
ON e.job_id = j.job_id
WHERE e.salary < j.min_salary;
```

(Note: You can also achieve this using multiple join conditions – simply replace the WHERE with AND, and you are specifying a second condition for the join)

**Q8.**

```
SELECT e.first_name + ' ' + e.last_name AS 'employee_full_name',
       m.first_name + ' ' + m.last_name AS 'manager_full_name'
FROM employee AS e LEFT OUTER JOIN employee AS m
ON e.manager_id = m.employee_id
ORDER BY employee_full_name;
```

**Q9.**

```
SELECT AVG(min_salary) AS 'average_min_salary',
       AVG(max_salary) AS 'average_max_salary'
FROM job;
```

**Q10.**

```
SELECT job_id, MIN(salary) AS 'lowest_salary',
       COUNT(*) AS 'employees'
FROM employee
GROUP BY job_id;
```

**Q11.**

```
SELECT department_name, COUNT(*) AS 'employees',
       AVG(salary) AS 'average_salary',
       MIN(salary) AS 'minimum_salary',
       MAX(salary) AS 'maximum_salary'
FROM employee AS e INNER JOIN department AS d
ON e.department_id = d.department_id
GROUP BY department_name
ORDER BY department_name;
```

**Q12.**

```
SELECT department_name, gender, COUNT(*) AS 'employees',
       AVG(salary) AS 'average_salary',
       MIN(salary) AS 'minimum_salary',
       MAX(salary) AS 'maximum_salary'
FROM employee AS e INNER JOIN department AS d
ON e.department_id = d.department_id
GROUP BY department_name, gender
ORDER BY department_name, gender;
```

**Q13.**       SELECT last\_name, department\_name, city, country\_name, region\_name  
FROM employee AS e FULL OUTER JOIN department AS d  
          ON e.department\_id = d.department\_id  
          FULL OUTER JOIN location AS l  
          ON d.location\_id = l.location\_id  
          FULL OUTER JOIN country AS c  
          ON l.country\_id = c.country\_id  
          FULL OUTER JOIN region AS r  
          ON c.region\_id = r.region\_id;

**Q14.**       SELECT department\_name  
FROM employee AS e INNER JOIN department AS d  
          ON e.department\_id = d.department\_id  
GROUP BY department\_name  
HAVING AVG(salary) < 7500;

**Q15.**       SELECT e.last\_name AS 'employee', m.last\_name AS 'manager',  
          e.salary - m.salary AS 'difference'  
FROM employee AS e LEFT OUTER JOIN employee AS m  
          ON e.manager\_id = m.employee\_id  
WHERE e.salary > m.salary;