SCHOOL OF
COMPUTER AND SECURITY SCIENCE

AUSTRALIA ECU EDITH COWAN UNIVERSITY

CSG1207/CSI5135: Systems and Database Design
Lab 10 - 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 last_name, hire_date
FROM employee
WHERE hire_date > ( SELECT hire_date
                    FROM employee
                    WHERE employee_id = 17 )
ORDER BY hire_date DESC;
```

Q2.

```
SELECT last_name, salary, department_name
FROM employee AS e INNER JOIN department AS d
     ON e.department_id = d.department_id
WHERE salary < (SELECT salary
                FROM employee
                WHERE employee_id = 18)
      OR e.department_id = (SELECT department_id
                            FROM employee
                            WHERE employee_id = 5);
```

Q3.

```
SELECT employee_id, last_name, department_id
FROM employee
WHERE department_id IN ( SELECT department_id
                         FROM employee
                         WHERE employee_id IN(19, 18, 2) );
```

Q4.

```
SELECT last_name, salary
FROM employee
WHERE salary > ALL (SELECT salary
                    FROM employee
                    WHERE department_id = 40);
```

(Note: An alternate solution would be to select `MAX(salary)` in the subquery, and omit the `ALL` from the outer query)

Q5.

```
CREATE VIEW emp_summary_view
AS SELECT employee_id AS 'emp_id',
          first_name + ' ' + last_name AS 'full_name',
          LOWER(email) + '@company.com' AS 'email',
          phone_number AS 'phone'
   FROM employee;
```

Q6.

```
ALTER VIEW emp_summary_view
AS SELECT employee_id AS 'emp_id',
          first_name + ' ' + last_name AS 'full_name',
          LOWER(email) + '@company.com' AS 'email',
          phone_number AS 'phone',
          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. 3

Q8.

```
SELECT LEFT(first_name, 1) + LEFT(last_name, 1) AS 'initials',
       ISNULL(gender, '?') AS 'gender',
       LOWER(email) + '@company.com' AS 'email',
       DATEDIFF(YEAR, hire_date, GETDATE()) AS 'years_worked',
       LEN(job_id) AS 'job_length',
       ROUND(salary * PI(), 2) AS 'salary_pi'
FROM employee;
```

Q9. Note: There are several ways to do this, but I find that using STR and LTRIM is the most reliable. Using STR allows the decimal places to be avoided easily, and LTRIM allows the output of STR to have no whitespace without needing to specify a length.

```
SELECT last_name, LTRIM(STR(commission_pct * 100)) + '%' AS 'comm'
FROM employee
WHERE commission_pct IS NOT NULL;
```

Challenge Query!

Q10.

```
SELECT first_name + ' ' + last_name + ' was a ' + job_title + ' for
       ' + CAST(DATEDIFF(YEAR, start_date, end_date) AS VARCHAR) +
       ' years' AS 'job_history'
FROM employee AS e INNER JOIN job_history AS jh
     ON e.employee_id = jh.employee_id
     INNER JOIN job AS j
     ON jh.job_id = j.job_id;
```

Using CASE to meet the extra challenge requirement, the query becomes…

```
SELECT first_name + ' ' + last_name + ' was a ' + job_title + ' for ' +
       CASE DATEDIFF(YEAR, start_date, end_date)
           WHEN 0 THEN 'less than a year'
           ELSE CAST(DATEDIFF(YEAR, start_date, end_date) AS VARCHAR) + ' years'
       END AS 'job_history'
FROM employee AS e INNER JOIN job_history AS jh
     ON e.employee_id = jh.employee_id
     INNER JOIN job AS j
     ON jh.job_id = j.job_id;
```