SCHOOL OF
COMPUTER AND SECURITY SCIENCE

CSG1207/CSI5135: Systems and Database Design
Lab 05

Introduction

This lab allows you to practise the SQL covered in this week's lecture. Do your best to answer the following questions and write the specified queries. You are encouraged to experiment with SQL – it is a very flexible language, so if you can think of something that would be useful to achieve in a query, it can probably be done. This lab uses the "company" database, which you can create by running the script file Module 5 of the unit materials. It is assumed that you are working in SQL Server Management Studio, also covered in Module 5.

If you are having trouble writing an SQL query, read any error messages and try to fix the error. Search for examples in the unit materials, and ask your tutor for assistance if needed. Contact your tutor if you spot something which appears to be incorrect in any of the labs.

Lab Tasks

Q1. There are two errors in the following query. Identify the errors, and fix the query so that it retrieves the first_name, last_name and gender of all employees.

```
SELECT first_name, lastname gender
FROM employee;
```

Q2. Write a query which selects all columns from the "department" table.

Q3. Write a query which selects the following from the "employee" table:

- Employee ID numbers
- Last names of employees
- Employee salaries
- Employee salaries multiplied by 12
- 75% of employee salaries (this can easily be done by multiplying salary by 0.75)

Q4. Modify the query you wrote in Q3 to give the last two columns appropriate aliases. Name the salary multiplied by 12 "annual_salary" and 75% of the salary "penalty_salary"

Q5. Write a query which uses the concatenation operator ("+") to select the following:

- Full names of all employees, e.g. "Steven King"
- Full names of all employees with last name first, e.g. "King, Steven"

Q6. Write a query which selects *unique* job IDs from the "employee" table. i.e. even though there are multiple employees who have a job ID of "IT_PROG", it should only appear once in the results of the query.

Q7. The following query is meant to retrieve the full names of employees and their emails, e.g. "Steven King - SKING", in a column named "name_and_email" but contains some errors. Identify the errors, and fix the query so that it works as intended.

```
SELECT first_name + " " + last_name + " - " email
       AS "name_and_email'
FROM employee;
```

Q8. Write a query which selects the following from the "employee" table:

- Full name of employees, with job ID in parentheses at the end of the name, e.g. "Steven King (AD_PRES)"
- Emails of employees, with "@COMPANY.COM" concatenated to the end, e.g. "SKING@COMPANY.COM"

Give both columns appropriate aliases, such as "name_and_job" for the first column.

Q9. Write a query which selects the following from the "job" table:

- Job ID followed by a hyphen and then job title, e.g. "IT_PROG - Programmer"
- Maximum salary minus minimum salary, resulting in the difference between the two, e.g. the IT_PROG job has a difference of 6000 between the min and max

Give both columns appropriate aliases, such as "job_id_and_title" for the first column.

Challenge Query!

The following query is more difficult than the previous queries, and may involve SQL syntax which has not yet been covered in the unit.

Q10. Write a query which uses concatenation to return employee names and email addresses in the form of a HTML email hyperlink. An example of the format is:

```
<a href="mailto:sking@company.com">Steven KING</a>
```

The parts of the example that are drawn from columns in the database are in bold. You need to concatenate the rest of the example around the data.

You can change the capitalisation of something by using UPPER() and LOWER(), e.g.

```
SELECT UPPER(first_name), LOWER(first_name)
FROM employee;
```

UPPER() and LOWER() are row-level functions, covered in Module 10.