

CSG1207/CSI5135: Systems and Database Design

Lab 10

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.** Write a query which uses a subquery to select the last name and hire date of any employees who were hired after the employee with an ID of 17. Order the results by hire date, with the most recently hired employees at the top.
- Q2.** Write a query which uses subqueries to select the last name, salary and department name of any employees who earn less than the employee with an ID of 18 or are in the same department as the employee with an ID of 5.
- Q3.** Write a query which uses a subquery to select the employee ID, last name and department ID of any employees who work in the same department as employee 19, employee 18 or employee 2.
- Q4.** Write a query which uses a subquery to select the last name and salary of any employees who earn more than all of the employees in department 40.
- Q5.** Create the “emp_summary” table from the challenge task of Lab 8 as a view. The view should be named “emp_summary_view”, and it should contain the employee ID (named as “emp_id”), full name, lower case email address with “@company.com” at the end and phone number of all employees.

Q6. Alter the view you created in the previous task so that it also includes the job title and department name of employees. Make sure the view includes employees who are not in a department.

Q7. *Without running the following query, try to determine what it would output. Note that it does not interact with a database; it simply manipulates the string in the query.*

```
SELECT CHARINDEX('e', SUBSTRING(RIGHT(REVERSE('Hello World'), 5), 2, 3));
```

Q8. Write a query which displays the following employee details:

- Their initials, e.g. "SK" for "Steven King"
- The gender column, with NULLs replaced by a "?"
- The email column, in lower case with "@company.com" appended to the end
- The number of years since they were hired, using the hire_date column
- The number of characters in their job ID, e.g. 5 for "AD_VP"
- Their salary multiplied by the value of pi and rounded to two decimal places

Give all columns appropriate aliases.

Q9. Use row level functions to select the last name and commission value of employees. The commission should be multiplied by 100, have no decimal places, and should have a "%" at the end, e.g. a value of "0.20" becomes "20%". Limit your results to only show employees who have a commission.

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 the "job_history" table to produce the following results:

```
job_history
-----
Neena Kochhar was a Public Accountant for 4 years
Neena Kochhar was a Accounting Manager for 4 years
Lex De Haan was a Programmer for 5 years
Jonathon Taylor was a Sales Representative for 0 years
Jonathon Taylor was a Sales Manager for 0 years
Jennifer Whalen was a Administration Assistant for 6 years
Jennifer Whalen was a Public Accountant for 4 years
Michael Hartstein was a Marketing Representative for 3 years

(8 row(s) affected)
```

This will require joins with the employee and job tables for the names and job titles, a date-based row level function, converting data types, and quite a bit of concatenation.

For an extra challenge, make the query show "for less than a year" instead of "0 years" wherever appropriate. This involves using [CASE](#), which we don't cover in this unit.