UNIVERSITY OF WESTMINSTER

5COSC002W DATABASE SYSTEMS

2021-2022 Tutorial 06 and 07

Querying Relational Databases – Aggregating data in SQL and using subqueries

Model Answer

Case Study

Carefully consider the **Logical ERD** shown below for the ***MegaFirm*** organisation (figure 1).

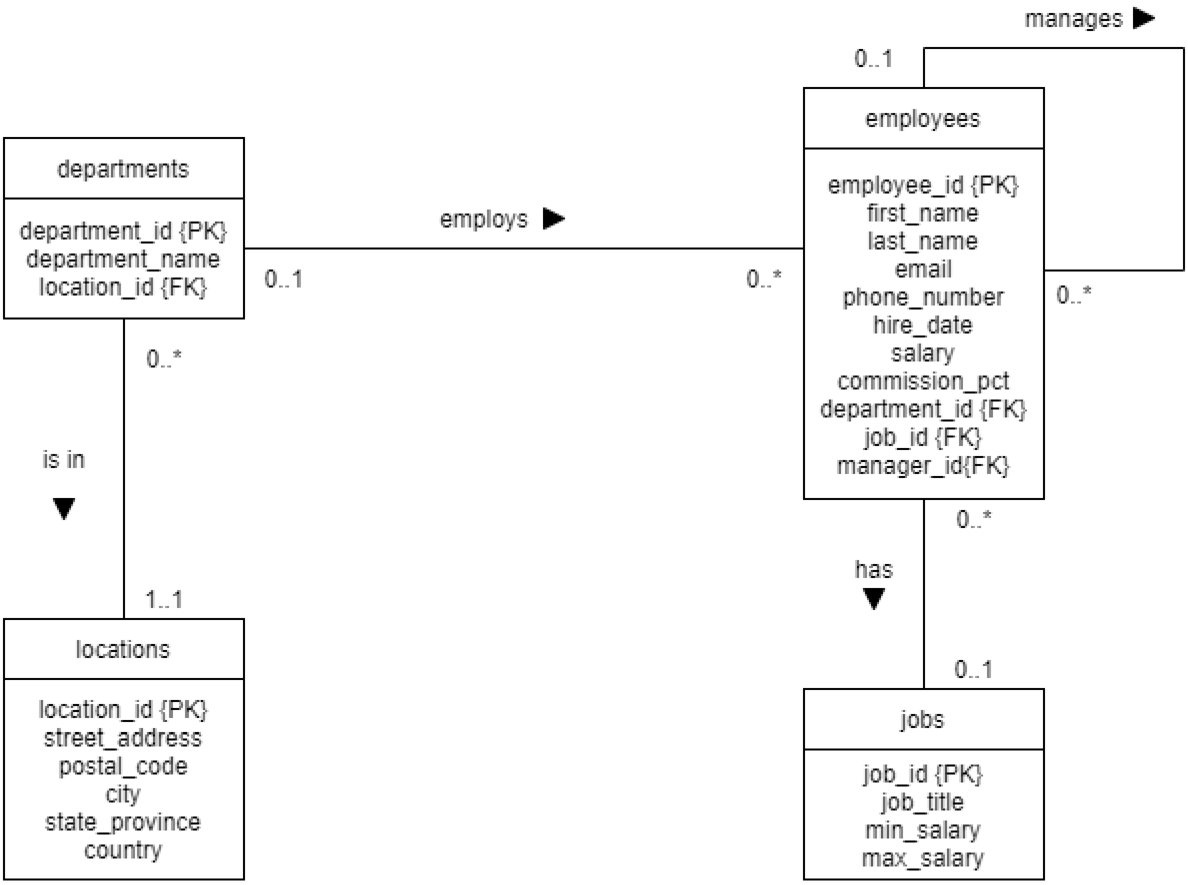

Figure 1: MegaFirm Logical ERD

Task 01: Accessing an IDE and MySQL via PHPMyAdmin

Simple reminder of how to open an IDE and access PHPMyAdmin to interact with MySQL and write your SQL queries.

1.1. Access an IDE.

- Access AppsAnywhere on https://appsanywhere.westminster.ac.uk and locate your preferred IDE (code editor) to write your SQL statements.
- You can choose any from the following list among others: Notepad++, Atom, Brackets, Visual Studio Code, Programmer's Notepad or Emacs.
- For more info, see https://support.ecs.westminster.ac.uk/w/index.php/Title:Text_Editors

1.2. Access the MySQL DBMS (via the PHPMyAdmin admin tool).

- Go to https://support.ecs.westminster.ac.uk/mysql/index.php to authenticate yourself and generate your MySQL database account details (you may need to enter your University login details first to access it).
- Access phpMyAdmin on https://phpmyadmin.ecs.westminster.ac.uk/ and enter the MySQL database account details just generated.
- Access your default database by clicking on your database name on the left hand-side.
- For more info, see the 'Database Systems Module Software Guide' accessible on Blackboard

Task 02: Creating and Populating the MegaFirm Tables in MySQL

Simple reminder of how to create and populate the tables for MegaFirm, as covered in Tutorial 04.
If you already have created and populated the locations, departments, employees and jobs table, please skip this task.

2.1 Access and run the table creation and population SQL script from Blackboard

- If you have not done already, get the "Tutorial 04 – SQL Tables Script" under 'Learning Resources and 'Section 02 – SQL'.
- Open the script in your IDE and copy and paste the code in the SQL area of phpMyAdmin (2nd tab 'SQL') and run it by clicking on "Go".
- Alternatively, import the script (6th tab 'Import') and execute it.

2.2. Check the structure and content of your MegaFirm Database

- You should see your tables as successfully created appearing in the list of tables on the left hand-side.
- Click on the 'Structure' tab at the top to verify the structure of the table.
- Click on the 'Browse' tab at the top to verify the content of the table.

TUTORIAL 06: GROUP FUNCTIONS AND AGGREGATING DATA

Tutorial 06 Question 01

a) Write a query that displays the average salary for all employees with IDs 1022, 1023, 1024 and 1025. Label column as "Average Salary".

```
SELECT AVG(salary) AS "Average Salary"
FROM employees
WHERE employee_id IN (1022, 1023, 1024, 1025);
```

b) Modify the previous query to include these staff who may not have a salary in your average calculation. Treat the absence of salary like a 0 value.

```
SELECT AVG(IFNULL(salary,0)) AS "Average Salary"
FROM employees
WHERE employee_id IN (1022, 1023, 1024, 1025);
```

Tutorial 06 Question 02

Write a query that displays the lowest, highest and average salary for the staff who have a job id that is either 907, 908, or 909 and who were hired in 2017. Label the columns "Minimum", "Maximum" and "Average".

Answer 1:

```
SELECT MIN(salary) AS "Minimum", MAX(salary) AS "Maximum", AVG(salary) AS "Average"
FROM employees
WHERE job_id IN (907,908,909)
AND (hire_date >= '2017-01-01' AND hire_date <= '2017-12-31');
```

Answer 2:

```
SELECT MIN(salary) AS "Minimum", MAX(salary) AS "Maximum", AVG(salary) AS "Average"
FROM employees
WHERE job_id IN (907,908,909)
AND hire_date LIKE '%2017%';
```

Tutorial 06 Question 03

Write a query that displays the manager number and, **FOR EACH manager**, the salary of the lowest-paid employee they manage. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is 47000 or less.

```
SELECT manager_id, MIN(salary)
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary)< 47000;
```

Tutorial 06 Question 04

Write a query that displays department IDs and **FOR EACH department**, the lowest, highest and average salary but only for these departments for which both the maximum and the average salary is greater than 50000. Label the columns “Dept Id”, “Minimum”, “Maximum” and “Average”. Also exclude staff not allocated to work in a department.

```
SELECT department_id, MIN(salary) AS "Minimum", MAX(salary) AS "Maximum",
AVG(salary) AS "Average"
FROM employees
WHERE department_id IS NOT NULL
GROUP BY department_id
HAVING (MAX(salary) >= 50000 AND AVG(salary) >= 50000);
```

Tutorial 06 Question 05

Write a query that displays a list of department IDs and department names and **FOR EACH department** the average salary in that department. Also include the details of those departments who do not have any employees.

```
SELECT d.department_id, d.department_name, AVG(e.salary) AS "Average"
FROM departments d LEFT OUTER JOIN employees e
ON d.department_id = e.department_id
GROUP BY d.department_id;
```

Tutorial 06 Question 06

Write a query that displays a list of job ids and job titles with the number of staff (i.e. a count of the numbers of employees) **FOR EACH job**. Label the columns “Job Id”, “Job Title” and “Staff Count”. Only include the job titles that are not related to management. In the end, only include the jobs that have at least 2 employees for each job and order the output in alphabetic order of job titles.

```
SELECT e.job_id AS "Job Id", j.job_title AS "Job Title",
COUNT(e.job_id) AS "Staff Count"
FROM jobs j JOIN employees e
ON j.job_id = e.job_id
AND j.job_title NOT LIKE '%Manag%'
GROUP BY e.job_id
HAVING COUNT(e.job_id)>=2
ORDER BY j.job_title;
```

TUTORIAL 07: SUBQUERIES

Tutorial 07 Question 01

Write a query that displays the department numbers, last names and hire dates of all the staff who works in the same department as Matos. Exclude Matos. Rank in ascending order of department number. (Hint: check the number of staff called Matos in your dataset).

```
SELECT department_id, last_name, hire_date
FROM employees
WHERE department_id IN
                        (SELECT department_id
                         FROM employees
                         WHERE last_name = 'Matos')
AND last_name <> 'Matos'
ORDER BY department_id;
```

Tutorial 07 Question 02

a) Write a query that displays the last names and salaries of all staff who earn strictly more than the highest salary in department 40. (Hint: think of 2 different ways of writing this query).

Answer 1:

```
SELECT last_name, salary
FROM employees
WHERE salary >
                  (SELECT MAX(salary)
                   FROM employees
                   WHERE department_id = 40);
```

Answer 2:

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

b) Write a query that displays the last names and salaries of all staff who earn strictly more than the lowest salary in department 10. (Hint: think of 2 different ways of writing this query).

Answer 1:

```
SELECT last_name, salary
FROM employees
```

```
WHERE salary >
                  (SELECT MIN(salary)
                   FROM employees
                   WHERE department_id = 10);
```

Answer 2:

```
SELECT last_name, salary
FROM employees
WHERE salary > ANY
                  (SELECT salary
                   FROM employees
                   WHERE department_id = 10);
```

Tutorial 07 Question 03

Write a query that displays the last names, salary, hire dates and department numbers of all employees whose department location ID is 100. (Hint: think of 2 different ways of writing this query, one using a join, the other one using a subquery).

Answer 1:

```
SELECT last_name, salary, hire_date, department_id
FROM employees
WHERE department_id IN
                        (SELECT department_id
                         FROM departments
                         WHERE location_id = 100);
```

Answer 2:

```
SELECT e.last_name, e.salary, e.hire_date, e.department_id
FROM employees e JOIN departments d
ON d.department_id = e.department_id
AND d.location_id = 100;
```

Tutorial 07 Question 04

Write a query that displays the last names and salaries of all staff who are managed by an employee whose first name start with the letter 'J'. (Hint: think of 2 different ways of writing this query, one using a join, the other one using a subquery).

Answer 1:

```
SELECT employee_id, last_name, salary
FROM employees
WHERE manager_id IN
                              (SELECT employee_id
                               FROM employees
                               WHERE first_name LIKE 'J%');
```

Answer 2:

```
SELECT e.employee_id, e.last_name, e.salary
FROM employees m JOIN employees e
ON m.employee_id = e.manager_id
AND m.first_name LIKE 'J%';
```

Tutorial 07 Question 05

Write a query that displays the last names and salaries of all staff who work in the same department as someone who was hired from the 1st of January 2017 onwards, and, at the same time, who earn more than the average salary of those employees not allocated to a department.

```
SELECT employee_id, last_name, salary, hire_date
FROM employees
WHERE department_id IN
                        (SELECT department_id
                         FROM employees
                         WHERE hire_date > '2017-01-01')
AND   salary >          (SELECT AVG(salary)
                         FROM employees
                         WHERE department_id IS NULL);
```

Tutorial 07 Question 06

Write a query that displays a list of department IDs and department names for those departments that do not have any employees with a job code 904.

```
SELECT department_id, department_name
FROM departments
WHERE department_id NOT IN
                              (SELECT department_id
                               FROM employees
                               WHERE job_id = 904
                               AND department_id IS NOT NULL);
```