

Mentoring Session Week 3

Pl. Ensure that "hr" database is created or downloaded from MYSQL sample databases before getting started with this exercise.

Once MySQL workbench is launched, spend a couple of minutes in familiarising participants with MYSQL Workbench options

Before taking this session, please ensure that participants are well aware and familiar with basic MYSQL commands as in this session focus in on Joins, Subquery, and Window Functions.

Section A: Joins, Sub-Queries & Window Functions

1. Execute the following basic commands to get started with the session

```
show databases;
USE hr;
show tables;
```

2. Write a query in SQL to display those employees who contain a letter z to their first name and also display their last name, department, city, and state province. (3 rows)

```
SELECT e.first_name
    ,e.last_name
    ,e.department_id
    ,d.department_name
    ,l.city
    ,l.state_province
FROM employees e
INNER JOIN departments d ON e.department_id = d.department_id
INNER JOIN locations l ON d.location_id = l.location_id
WHERE e.first_name LIKE "%z%";
```



3. Write a query in SQL to display the job title, department id, full name (first and last name) of employee, starting date and end date for all the jobs which started on or after 1st January, 1993 and ending with on or before 31 August, 2000. (8 rows)

4. Display the employee number, name (first name and last name) and job title for all employees whose salary is smaller than the minimum salary of those employees whose job title is Programmer using subquery. (44 rows)



5. Write a query in SQL to display the country name, city, and number of those departments where at least 2 employees are working. (5 rows)

6. Write a query to fetch the employee ID, First Name, Last Name, Salary and Department ID of those employees who draw a salary more than the average salary of their respective department. (38 rows)



7. Write a query in SQL to display the first and last name, salary, and department ID for those employees who earn less than the average salary, and also work at the department where the employee Laura is working as a first name holder. (41 rows)

8. Using HR Schema, write a Query to find the maximum salary of the most recent job that every employee holds.

```
SELECT e.employee id
  ,e.first name
  ,e.last name
 ,j.job_title
 ,MAX(e.salary) AS max salary
FROM employees e
JOIN job history jh ON e.employee id = jh.employee id
JOIN jobs j ON jh.job id = j.job \overline{id}
WHERE (
         e.employee id
          ,jh.start date
         ) IN (
         SELECT employee id
                 ,MAX(start date)
         FROM job history
         GROUP BY employee id
GROUP BY e.employee id
 ,e.first name
 ,e.last name
 ,j.job title;
```

9. Using HR Schema, write a Query to List the old designation and new designation of all the employees in the company where old designation is not null. (10 rows)



```
SELECT DISTINCT e.employee_id
    ,e.first_name
    ,e.last_name
    ,e.job_id AS current_job
    ,j.job_id AS old_job
    ,jo.job_title AS CURRENT
FROM employees AS e
INNER JOIN job_history AS j ON e.employee_id = j.employee_id
INNER JOIN jobs AS jo ON jo.job_id = e.job_id;
```

10. Retrieve the employee details along with the highest salary of their department and the difference between their salary and the highest salary:

11. Write an SQL query to retrieve the employee details, including their ID, first name, last name, and the average salary within a range that includes the current employee's salary and the salaries of the preceding and succeeding employees based on their hire dates.



12. Find the average salary of employees in each department, along with the highest-paid employee details within that department. Additionally, you want to display the department name, manager name, and location details for each department.

```
SELECT d.department name
                ,e1.first_name || ' ' || e1.last_name AS manager_name
               ,1.street_address || ', ' || 1.city || ', ' ||
1.state province || ', ' || 1.country id AS location details
                ,e2.first name || ' ' || e2.last name AS
highest paid employee
                ,e2.salary AS highest salary
                ,AVG(e1.salary) OVER (PARTITION BY d.department id) AS
avg salary
      FROM departments d
      JOIN employees e1 ON d.manager id = e1.employee id
      JOIN locations 1 ON d.location id = 1.location id
      JOIN (
       SELECT employee id
               ,first name
                ,last name
                ,salary
                ,department id
                , RANK () OVER (
                       PARTITION BY department id ORDER BY salary DESC
                       ) AS rank1
       FROM employees
       ) e2 ON e1.department id = e2.department id
      AND e2.rank1 = 1
      ORDER BY d.department id;
```



Section B: General Queries without using any dataset

• Write a Query to display the word 'Great Learning' by removing the vowels.

```
SELECT regexp replace("Great Learning", "[a,e,i,o,u]", "");
```

• Write a Query to remove all the leading and trailing exclamatory marks from the string '!!!!!Great Learning!!!!!!.

```
SELECT replace('!!!!!Great Learning!!!!!!', "!", "");
```

• Write a Query to divide the number 100 by 3 and print the remainder after division.

```
SELECT 100 mod 3;
```

• Display 'Great Learnings' in capital letter

```
SELECT upper('Great Learnings');
```

• Display the difference between '2020-01-21' and '2020-01-21'

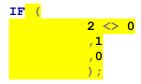
```
SELECT datediff('2020-01-21', '2020-01-01') AS
Difference_Between_Dates;
```

• Display the age if the date of birth is '1999-09-08'

```
SELECT datediff(curdate(), '1999-09-08') / 365 AS Age;
```



• Display '1' if 2<>0 condition is true otherwise display '0' **SELECT**



• Display the square of 9

```
SELECT power(9, 2);
```