**SQL QUERIES ASSIGNMENT**

**TITLE :** SQL QUERIES TASK

**AUTHOR :** MADHESH N

**DESIGNATION :** TRAINEE | IMPACT TRAINING

**CREATED ON :** 22/11/2024

**MODIFIED ON :** 23/11/2024

**REVIEWED BY :**

**REVIEWED ON :**

**BASIC QUERIES**

1. **Write a query to display all rows and columns from the employees table.**

* Query for display all rows and columns from the employees table

**SELECT \* FROM employees;**

1. **Retrieve only the name and salary of all employees from the employees table.**

* Query for Retrieve only the name and salary of all employees from the employees table.

**SELECT name, salary FROM employees;**

**3 .Write a query to find all employees whose salary is greater than 50,000.**

* Query for find all employees whose salary is greater than 50,000.

**SELECT \* FROM employees WHERE salary > 50000;**

1. **List all employees who joined the company in the year 2020.**

* Query for List all employees who joined the company in the year 2020.

**SELECT \* FROM employees WHERE join\_year = 2020;**

1. **Retrieve the details of employees whose names start with the letter 'A'.**

* Query for Retrieve the details of employees whose names start with the letter 'A'.

**SELECT \* FROM employees WHERE POSITION('A' IN name) = 1;**

**Aggregate Functions**

1. **Write a query to calculate the average salary of all employees.**

* Query for Write a query to calculate the average salary of all employees.

**SELECT AVG( salary ) FROM employees;**

1. **Find the total number of employees in the company.**

* Query for Find the total number of employees in the company.

**SELECT COUNT(\*) FROM employees;**

1. **Write a query to find the highest salary in the employees table.**

* Query for Find the total number of employees in the company.

**SELECT MAX(salary) FROM employees;**

1. **Calculate the total salary paid by the company for all employees.**

* Query for calculate the total salary paid by the company for all employees.

**SELECT SUM(salary) FROM employees;**

1. **Find the count of employees in each department.**

* Query to Find the count of employees in each department.

**SELECT department, COUNT(\*) employee\_count FROM employees**

**GROUP BY department;**

**Joins**

1. **Write a query to retrieve employee names along with their department names (using employees and departments tables).\**

* Query to retrieve employee names along with their department names

**SELECT employee.name AS employee\_name, department.department\_name FROM employees employee JOIN departments department ON employee.department\_name = department.department\_name;**

1. **List all employees who have a manager**

* Query to retrieve employees who have a manager

**SELECT employee.name AS employee\_name FROM employees employee JOIN employees manager ON employee.manager\_id = manager.id;**

1. **Find the names of employees who are working on multiple projects**

* Query to Find the names of employees who are working on multiple projects

**SELECT project.project\_name, employee.employee\_name FROM projects project**

**JOIN employee\_projects employee\_project ON project.project\_id = employee.project\_id**

**JOIN employees employee ON employee\_project.employee\_id = employee.employee\_id;**

1. **Write a query to display all projects and the employees assigned to them.**

* Query to display all projects and the employees assigned to them.

**SELECT project.project\_name, employee.employee\_name FROM projects project**

**JOIN employee\_projects employee\_project ON project.project\_id = employee\_project.project\_id**

**JOIN employees employee ON employee\_project.employee\_id = employee.employee\_id;**

1. **Retrieve the names of employees who do not belong to any department.**

* Query to Retrieve the names of employees who do not belong to any department.

**SELECT name FROM employees WHERE department\_name IS NULL;**

**SUBQUERIES**

1. **Write a query to find the employees with the second-highest salary.**

* **SELECT name FROM employees WHERE salary = (SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees));**

1. **Retrieve the names of employees whose salary is above the department average salary.**

* **SELECT employee.name FROM employees employee JOIN (SELECT department\_name, AVG(salary) AS avg\_salary FROM employees GROUP BY department\_name**

**) department ON employee.department\_name = department.department\_name**

**WHERE employee.salary > department.avg\_salary;**

1. **Find employees who earn more than the average salary of the entire company.**

* **SELECT name FROM employees WHERE salary > (SELECT AVG(salary) FROM employees);**

**4. Write a query to find the department with the highest number of employees.**

* **SELECT department\_name FROM employees GROUP BY department\_name**

**ORDER BY COUNT(\*) DESC LIMIT 1;**

**5. List all employees who work in a department located in 'New York'.**

**SELECT employee.name FROM employees employee JOIN departments department ON employee.department\_name = department.department\_name WHERE department.location = 'New York';**

**Set Operators**

1. **Write a query to find employees who work in either the 'HR' or 'Finance' department.**

* **SELECT name FROM employees WHERE department\_name = 'HR' UNION**

**SELECT name FROM employees WHERE department\_name = 'Finance';**

1. **Retrieve the names of employees who are working on both Project A and Project B.**

* **SELECT employee.name FROM employees employee JOIN employee\_projects employee\_project ON employee.employee\_id = employee\_project.employee\_id WHERE employee\_project.project\_name = 'Project A' INTERSECT SELECT employee.name FROM employees employee JOIN employee\_projects employee\_project ON employee.employee\_id = employee\_project.employee\_id WHERE employee\_project.project\_name = 'Project B';**

**3. Find employees who are not assigned to any project.**

* **SELECT employee.name FROM employees employee EXCEPT SELECT employee.name FROM employees employee JOIN employee\_projects employee\_project ON employee.employee\_id = employee\_project.employee\_id;**

1. **Write a query to get all unique job titles across all departments.**

* **SELECT job\_title FROM employees UNION SELECT job\_title FROM departments;**

1. **Combine two tables (employees and former\_employees) and remove duplicates.**

* **SELECT name, department\_name, job\_title FROM employees UNION SELECT name, department\_name, job\_title FROM former\_employees;**

**DML and DDL**

1. **Write a query to add a new employee to the employees table.**

* **INSERT INTO employees (employee\_id, name, department\_name, salary, join\_date)**

**VALUES (ACE12434, 'Madhesh', 'IAS', 30000, '2024-11-15');**

1. **Update the salary of all employees in the 'IT' department by 10%.**

* **UPDATE employees SET salary = salary \* 1.10 WHERE department\_name = 'IT';**

1. **Delete all employees who have not worked for more than 5 years.**

* **DELETE FROM employees WHERE YEAR(CURDATE()) – YEAR(join\_date) > 5;**

**4. Create a new table departments\_backup with the same structure as the departments table.**

* **CREATE TABLE departments\_backup ( department\_id INT, department\_name VARCHAR(100), location VARCHAR(100));**

**5.Drop the temporary\_data table from the database.**

* **DROP TABLE temporary\_data;**

**Constraints**

1. **Add a primary key to the employees table.**

* **ALTER TABLE employees ADD CONSTRAINT primaryket\_employee\_id PRIMARY KEY (employee\_id);**

1. **Write a query to create a foreign key between employees and departments tables.**

* **ALTER TABLE employees ADD CONSTRAINT fk\_department\_id FOREIGN KEY (department\_id) REFERENCES departments(department\_id);**

1. **Add a unique constraint to the email column in the employees table.**

* **ALTER TABLE employees ADD CONSTRAINT unique\_email UNIQUE (email);**

1. **Write a query to check all constraints applied on the employees table.**

* **SELECT CONSTRAINT\_NAME, CONSTRAINT\_TYPE, TABLE\_NAME FROM information\_schema.TABLE\_CONSTRAINTS WHERE TABLE\_NAME = 'employees';**

1. **Remove the NOT NULL constraint from the phone\_number column in the employees table.**

* **ALTER TABLE employees MODIFY phone\_number VARCHAR(10) NULL;**