

## SQL Assignmen-4

//Table: employees

```
CREATE TABLE employees (  
    employee_id INT PRIMARY KEY,  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    department_id INT,  
    salary DECIMAL(10, 2),  
    hire_date DATE, manager_id INT );
```

//Insert sample data for the employees table

```
INSERT INTO employees VALUES  
(1, 'John', 'Doe', 1, 60000.00, '2023-01-01', 3),  
(2, 'Jane', 'Smith', 2, 75000.00, '2023-02-15', 4),  
(3, 'Vim', 'Doe', 3, 50000.00, '2024-03-30', 3),  
(4, 'Kim', 'Smith', 4, 55000.00, '2022-12-15', 2),  
(5, 'Tim', 'Kook', 3, 30000.00, '2023-01-01', 1),  
(6, 'Red', 'Smith', 2, 25000.00, '2024-02-15', 4),  
(7, 'horse', 'rider', 4, 100000.00, '2023-01-01', null);
```

//Table: departments

```
CREATE TABLE departments (  
    department_id INT PRIMARY KEY,  
    department_name VARCHAR(50) );
```

//insert sample data for the departments table

```
INSERT INTO departments VALUES (1, 'Sales');  
INSERT INTO departments VALUES (2, 'IT');  
INSERT INTO departments VALUES (3, 'MARKETING');  
INSERT INTO departments VALUES (4, 'SERVICE');
```

## SQL Questions:

### 1. Select Basics:

- Write an SQL query to retrieve all columns from the "employees" table.

Ans: `select * from employees`

- Write an SQL query to select only the "first\_name" and "last\_name" columns from the "employees" table.

Ans: `select first_name, last_name from employees`

### 2. Filtering Data:

- Retrieve all employees from the "employees" table who work in the "Sales" department.

Ans: `SELECT * FROM employees`

`WHERE department_id = (`

`SELECT department_id FROM departments WHERE department_name = 'Sales');`

- Retrieve employees from the "employees" table who have a salary greater than \$50,000.

Ans: `select * from employees WHERE salary>50000;`

### 3. Sorting and Ordering:

- Write an SQL query to display the "employees" table sorted by the "hire\_date" in descending order.

Ans: `SELECT * from employees ORDER BY hire_date DESC`

- Retrieve the top 5 highest-paid employees from the "employees" table.

Ans: `SELECT * from employees ORDER by salary DESC limit 5`

### 4. Aggregation:

- Calculate the average salary of all employees in the "employees" table.

Ans: `SELECT AVG(salary) from employees;`

- Count the number of employees in each department.

Ans: `select COUNT( DISTINCT department_id) from employees`

## 5. Joins:

- Write an SQL query to retrieve all employees and their corresponding department names from the "employees" and "departments" tables.

Ans:    select employees.\*, departments.department\_name from employees  
          JOIN departments on employees.department\_id = departments.department\_id

- Retrieve a list of employees and their managers from the "employees" table.

Ans:    SELECT e.employee\_id ,  
          concat (e.first\_name,' ',e.last\_name) as employee\_name,  
          e.manager\_id,  
          e1.first\_name || ' ' || e1.last\_name as manager\_name  
          from employees as e

join employees as e1 on e.manager\_id = e1.employee\_id

## 6. Group By:

- Display the total salary expense for each department from the "employees" table.

Ans:    select department\_id, sum(salary) as average\_salary  
          from employees GROUP BY department\_id

- Find the department with the highest average salary.

Ans:    select department\_id, avg(salary) as average\_salary  
          from employees  
          GROUP BY department\_id  
          LIMIT 1

## 7. Subqueries:

- Retrieve all employees from the "employees" table who have a salary greater than the average salary.

Ans:    select \* from employees  
          where salary > (select avg(salary) from employees );

- Find the employees who do not have a manager in the "employees" table.

Ans:    SELECT employee\_id,  
          first\_name || ' ' || last\_name as Full\_name,  
          department\_id,  
          salary,  
          manager\_id  
          FROM employees  
          where manager\_id is NULL

## 8. Update and Delete:

- Update the salary of all employees in the "IT" department to increase by 10%.

Ans:    update employees set salary = salary + salary/10  
          where department\_id = ( select employee\_id from employees  
                                  join departments on employees.department\_id = departments.department\_id  
                                  where department\_name='IT')

- Delete all employees from the "employees" table who have a salary less than \$40,000.

Ans:    DELETE from employees where salary < 40000;

## 9. Constraints:

- Create a new table named "projects" with columns for project ID, project name, and project manager ID.

Ans: Table: projects

```
CREATE TABLE projects ( project_id INT PRIMARY KEY,  
                          project_name VARCHAR(50),  
                          project_manager_id INT );
```

Insert sample data for the projects table

```
INSERT INTO projects VALUES (101, 'Project A', 3);
```

```
INSERT INTO projects VALUES (102, 'Project B', 4);
```

```
INSERT INTO projects VALUES (103, 'Project C', 1);
```

```
INSERT INTO projects VALUES (104, 'Project D', 2);
```

- Add a unique constraint to the "email" column in the "employees" table

Ans: ALTER TABLE employees add COLUMN employee\_email varchar(20) UNIQUE;