Infosys internship 6.0

SQL Task

1. Introduction to SQL

SQL(StructuredQuery Language) is the standard language for managing and manipulating relational databases. It is used for creating, reading, updating, and deleting data efficiently.

2. Database and Table Management

```
1 CREATE DATABASE company;
2 USE company;
3 CREATE TABLE employees (id INT PRIMARY KEY, name VARCHAR(50), age INT, salary
  DECIMAL(10,2), department_id INT);
4 SHOW TABLES;
5 DESCRIBE employees;
6 ALTER TABLE employees ADD COLUMN hire_date DATE;
7 ALTER TABLE employees DROP COLUMN age;
8 DROP DATABASE company;
```

3. CRUD Operations

```
1 INSERTINTOemployees(id, name, salary, department_id) VALUES (1, 'John Doe', 55000.00, 2);
2 UPDATE employees SET salary = salary * 1.10 WHERE department_id = 2;
3 DELETE FROM employees WHERE id = 3;
4 SELECT * FROM employees;
```

4. Filtering and Sorting

```
1 SELECT * FROMemployees WHERE salary BETWEEN 40000 AND 80000;
2 SELECT name, salary FROM employees ORDER BY salary DESC LIMIT 5;
3 SELECT * FROM employees WHERE name LIKE 'A%';
4 SELECT DISTINCT department_id FROM employees;
```

5. Aggregate Functions and Grouping

```
1 SELECT COUNT(*)AS total_employees FROMemployees;
2 SELECT department_id, AVG(salary) AS avg_salary FROM employees GROUP BY department_id;
3 SELECT department_id, MAX(salary), MIN(salary) FROM employees GROUP BY department_id;
4 SELECT department_id, SUM(salary) FROM employees GROUP BY department_id HAVING SUM(salary) > 100000;
```

6. SQL Joins

```
1 SELECTe.name, d.department_name FROM employees e INNER JOIN departments d ON e.department_id = d.id;
2 SELECT e.name, d.department_name FROM employees e LEFT JOIN departments d ON e.department_id = d.id;
3 SELECT e.name, d.department_name FROM employees e RIGHT JOIN departments d ON e.department_id = d.id;
  SELECT * FROM employees e FULL OUTER JOIN departments d ON e.department_id = d.id;
4
```

7. Subqueries and Views

```
1 SELECT name FROMemployeesWHERE salary > (SELECT AVG(salary) FROM employees);
2 CREATE VIEW top_earners AS SELECT name, salary FROM employees WHERE salary > 80000;
3 SELECT * FROM top_earners;
4 DROP VIEW top_earners;
```

8. Indexes and Constraints

```
1 CREATEINDEX idx_salary ONemployees(salary);
2 DROP INDEX idx_salary ON employees;
3 ALTER TABLE employees ADD CONSTRAINT fk_dept FOREIGN KEY (department_id) REFERENCES departments(id);
4 ALTER TABLE employees DROP CONSTRAINT fk_dept;
```

9. Miscellaneous Useful Commands

```
1 SHOW DATABASES;
2 SHOW COLUMNS FROM employees;
3 EXPLAIN SELECT * FROM employees;
4 DESCRIBE departments;
5 SELECT NOW();
6 SELECT DATABASE();
```