**Mid Term Exam**

**Total Marks:100 Date: 11-7-2024**

| **Q.No** | **QUESTIONS** | **Marks** |
| --- | --- | --- |
| 1 | Explain the difference between Primary Key and Foreign Key in SQL.  Answer:  Primary Key is a unique column of a table for each record while Foreign Key is a column which is made relation with another table's Primary Key. | 5 |
| 2 | What is a Self Join? Provide an example using the employees table to list employees who share the same **manager ID**, showing only their **first name**s and **manager ID**s.  Answer:  Self Join: A single table if act as two different tables and make Join or a relation then it’s called Self Join.  SELECT e1.first\_name, e2.first\_name, e1.manager\_id  FROM employees AS e1  JOIN employees AS e2  ON e1.manager\_id = e2.manager\_id  AND e1.employee\_id <> e2.employee\_id; | 5 |
| 3 | Create a table of Employees which has the following fields   * 1. Employee\_Id   2. First Name   3. Last Name   4. Date of Birth   5. Department Id   6. Salary   Answer:  CREATE TABLE Employees (  Employee\_Id INT PRIMARY KEY,  First\_Name VARCHAR(50),  Last\_Name VARCHAR(50),  Date\_of\_Birth DATE,  Department\_Id INT,  Salary DECIMAL(10, 2)  );  Create a table Projects with the following fields:   1. Project ID 2. Project Name 3. Start Date 4. End Date 5. Budget   Answer:  CREATE TABLE Projects (  Project\_Id INT PRIMARY KEY,  Project\_Name VARCHAR(50),  Start\_Date DATE,  End\_Date DATE,  Budget DECIMAL(12, 2)  );  Create a table Employee\_Projects with the following fields:   1. Employee ID 2. Project ID   Ensure that each employee can work on multiple projects and a project can have multiple employees.  Answer:  CREATE TABLE Employee\_Projects (  Employee\_Id INT,  Project\_Id INT,  PRIMARY KEY (Employee\_Id, Project\_Id),  FOREIGN KEY (Employee\_Id) REFERENCES Employees(Employee\_Id),  FOREIGN KEY (Project\_Id) REFERENCES Projects(Project\_Id)  ); | 20 |
| **Use dummydb in MySQL to answer the following questions:** [**Link**](https://drive.google.com/file/d/1Lkklsw33VSjlRKQ3bxmwIUqGXtfQgJ-r/view?usp=drive_link) | | |
| 4 | Using the dummydb, write an SQL query to get the third-highest salary in the employees table.  Answer:  SELECT \*  FROM EMPLOYEES  WHERE SALARY = (SELECT MAX(SALARY)  FROM EMPLOYEES  WHERE SALARY < (SELECT MAX(SALARY)  FROM EMPLOYEES  WHERE SALARY < (SELECT MAX(SALARY)  FROM EMPLOYEES  )  )  ); | 8 |
| 5 | Write a query to show the department names and the number of employees in each department.  Answer:  SELECT DEPARTMENTS.DEPARTMENT\_NAME, COUNT(EMPLOYEES.EMPLOYEE\_ID) AS CNT  FROM DEPARTMENTS  JOIN EMPLOYEES ON DEPARTMENTS.DEPARTMENT\_ID = EMPLOYEES.DEPARTMENT\_ID  GROUP BY DEPARTMENTS.DEPARTMENT\_NAME; | 10 |
| 6 | Illustrate INNER JOIN, LEFT JOIN, RIGHT JOIN, and CROSS JOIN with examples using the employees and departments tables.  Answer:  SELECT \* FROM EMPLOYEES;  SELECT \* FROM DEPARTMENTS;  -- INNER JOIN  SELECT EMPLOYEES.FIRST\_NAME, DEPARTMENTS.DEPARTMENT\_NAME  FROM EMPLOYEES  INNER JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID;  -- LEFT JOIN  SELECT DEPARTMENTS.DEPARTMENT\_NAME  FROM DEPARTMENTS  LEFT JOIN EMPLOYEES  ON DEPARTMENTS.DEPARTMENT\_ID = EMPLOYEES.DEPARTMENT\_ID  WHERE EMPLOYEES.DEPARTMENT\_ID IS NULL;  -- RIGHT JOIN  SELECT EMPLOYEES.FIRST\_NAME, DEPARTMENTS.DEPARTMENT\_NAME  FROM EMPLOYEES  RIGHT JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID;  -- CROSS JOIN  SELECT EMPLOYEES.FIRST\_NAME, DEPARTMENTS.DEPARTMENT\_NAME  FROM EMPLOYEES  CROSS JOIN DEPARTMENTS ON EMPLOYEES.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID; | 14 |
| 7 | What is a Common Table Expression (CTE)? Write an example query using CTE to show the names of employees whose salary is higher than the average salary.  Answer:  A Common Table Expression (CTE) is the result of query which is saved in a temporary table view for using later.  WITH AVG\_SALARY AS (  SELECT AVG(SALARY) AS Sal  FROM EMPLOYEES  )  SELECT EMPLOYEES.FIRST\_NAME, EMPLOYEES.SALARY  FROM EMPLOYEES  JOIN AVG\_SALARY ON EMPLOYEES.SALARY > AVG\_SALARY.SAL; | 8 |
| 8 | Write a query to display the names of employees who earn a salary less than the employee "Steven King".  Answer:  SELECT \* FROM EMPLOYEES;  SELECT FIRST\_NAME, LAST\_NAME  FROM EMPLOYEES  WHERE SALARY < (  SELECT SALARY  FROM EMPLOYEES  WHERE FIRST\_NAME = 'Steven' AND LAST\_NAME = 'King'  ); | 10 |
| 9 | Write a query to find the department names and the names of the managers for each department.  Answer:  SELECT DEPARTMENTS.DEPARTMENT\_NAME, EMPLOYEES.FIRST\_NAME AS MANAGER  FROM DEPARTMENTS  JOIN EMPLOYEES ON DEPARTMENTS.MANAGER\_ID = EMPLOYEES.EMPLOYEE\_ID; | 10 |
| 10 | Write a query to display the **names of all cities** where **departments are located**.  Answer:  SELECT \* FROM DEPARTMENTS;  SELECT \* FROM LOCATIONS;  SELECT DISTINCT LOCATIONS.CITY  FROM DEPARTMENTS  JOIN LOCATIONS ON DEPARTMENTS.LOCATION\_ID = LOCATIONS.LOCATION\_ID; | 10 |