**Answer Script**

| Question No. 01 |
| --- |
| Explain the difference between Primary Key and Foreign Key in SQL. |
| Answer No. 01 |
| **Primary Key**   * Primary key is like a unique key for each of the data we store in a database. We have to store the unique value of a column and we name that uniqueness as primary key. The purpose of the primary key is to uniquely identify each row of the table.   **Foreign Key**   * Foreign key is the reference of the primary key that is used in another table. The purpose of foreign key is to establish relation between two tables. |

| Question No. 02 |
| --- |
| 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 No. 02 |
| Self join is a type of table join where a table is joined with itself. The purpose maybe when we compare two different row in a same table and do something with it. Now showing the first names and manager ids of those who share the same manager id:  SELECT emp.first\_name AS Employee\_name, mgr.manager\_id AS Manager\_id  FROM employees AS emp  JOIN employees AS mgr  ON emp.manager\_id = mgr.employee\_id; |

| Question No. 03 |
| --- |
| 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   Create a table Projects with the following fields:   1. Project ID 2. Project Name 3. Start Date 4. End Date 5. Budget   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 No.3 |
| create database Mid\_term;  use Mid\_term;  CREATE TABLE Employees (  Employee\_Id INT PRIMARY KEY,  First\_Name VARCHAR(50) NOT NULL,  Last\_Name VARCHAR(50),  Date\_of\_Birth DATE,  Department\_Id INT,  Salary INT  );  CREATE TABLE Projects (  Project\_ID INT PRIMARY KEY,  Project\_Name VARCHAR(100) NOT NULL,  Start\_Date DATE,  End\_Date DATE,  Budget INT  );  CREATE TABLE Employee\_Projects (  Employee\_Id INT,  Project\_ID INT,  FOREIGN KEY (Employee\_Id) REFERENCES Employees(Employee\_Id),  FOREIGN KEY (Project\_ID) REFERENCES Projects(Project\_ID),  PRIMARY KEY (Employee\_Id, Project\_ID)  ); |

| Question No. 04 |
| --- |
| Using the dummydb, write an SQL query to get the third-highest salary in the employees table. |
| Answer No. 04 |
| select distinct salary  from employees  order by salary desc  limit 1  offset 2; |

| Question No. 5 |
| --- |
| Write a query to show the department names and the number of employees in each department. |
| Answer No. 05 |
| SELECT d.Department\_Name, COUNT(e.Employee\_Id) AS Number\_of\_Employees  FROM Departments AS d  JOIN Employees AS e  ON d.Department\_Id = e.Department\_Id  GROUP BY d.Department\_Name; |

| Question No. 06 |
| --- |
| Illustrate INNER JOIN, LEFT JOIN, RIGHT JOIN, and CROSS JOIN with examples using the employees and departments tables. |
| Answer No. 06 |
| -- Inner join, right, left, cross join  select employees.first\_name, departments.department\_name  from employees  inner join departments using(department\_id); -- it is normal join  select employees.first\_name, departments.department\_name  from employees  left join departments using(department\_id);    select employees.first\_name, departments.department\_name  from employees  right join departments using(department\_id);  select employees.first\_name, departments.department\_name  from employees  cross join departments using(department\_id); |

| Question No. 07 |
| --- |
| 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 No. 07 |
| CTE is common table expression where a result of a query is saved temporarily using a different name than the table.  USE dummydb;  WITH AvgSalary AS (  SELECT AVG(salary) AS avg\_sal  FROM employees  )  SELECT first\_name, last\_name, salary  FROM employees  WHERE salary > (SELECT avg\_sal FROM AvgSalary); |

| Question No. 08 |
| --- |
| Write a query to display the names of employees who earn a salary less than the employee "Steven King". |
| Answer No. 08 |
| SELECT first\_name, last\_name, salary  FROM employees  WHERE salary < (  SELECT salary  FROM employees  WHERE first\_name = 'Steven' AND last\_name = 'King'  ); |

| Question No. 09 |
| --- |
| Write a query to find the department names and the names of the managers for each department. |
| Answer No. 9 |
| SELECT d.department\_name, e.first\_name AS manager\_name  FROM departments AS d  JOIN employees AS e  ON d.manager\_id = e.employee\_id; |

| Question No. 10 |
| --- |
| Write a query to display the **names of all cities** where **departments are located**. |
| Answer No. 10 |
| SELECT DISTINCT l.city  FROM departments AS d  JOIN locations AS l  ON d.location\_id = l.location\_id; |