# **Output**

Create restaurant table

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create Employee table

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Create orders table

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Create table for Customer

A screenshot of a computer

Description automatically generated

# Queries

1. Query : Select all columns and all rows from one table

SELECT \* FROM Restaurant;

A screenshot of a computer

Description automatically generated

1. Query : Select five columns and all rows from one table

SELECT Employee\_ID, Name, Role, Salary, Restaurant\_ID FROM Employee;

A screenshot of a computer

Description automatically generated

1. Query : Select all columns from all rows from one view

CREATE VIEW Review AS

SELECT Employee.Name, Employee.Role, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID, Orders.Order\_Total

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID

-- Query from the view

SELECT \* FROM Review;

A screenshot of a computer

Description automatically generated

1. Query : Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product.

SELECT \* FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID;

A screenshot of a computer

Description automatically generated

1. Select and order data retrieved from one table

SELECT \* FROM Employee

ORDER BY Salary DESC;

A screenshot of a computer

Description automatically generated

1. Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 3 rows.

SELECT Employee.Name, Employee.Role, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID, Orders.order\_Total

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID

LIMIT 3;

A screenshot of a computer

Description automatically generated

1. Query : Select distinct rows using joins on 3 tables

SELECT DISTINCT Employee.Name, Restaurant.Name AS Restaurant\_Name, Orders.Order\_ID

FROM Employee

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

JOIN Orders ON Employee.Employee\_ID = Orders.Employee\_ID;

A screenshot of a computer

Description automatically generated

1. Query: Use GROUP BY and HAVING in a select statement using one or more tables.

SELECT Employee\_ID, COUNT(\*) AS Num\_Orders, SUM(Order\_total) AS Total\_Sales

FROM Orders

GROUP BY Employee\_ID

HAVING COUNT(\*) > 5;

A screenshot of a computer

Description automatically generated

1. Query : Use IN clause to select data from one or more tables

SELECT \* FROM Employee

WHERE salary IN (28000);

A screenshot of a computer

Description automatically generated

1. Query : Select length of one column from one table (use LENGTH function)

SELECT Name, LENGTH(Name) AS Name\_Length FROM Employee;

A screenshot of a computer

Description automatically generated

1. Query: Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed.

SELECT \* FROM Customer;

A screenshot of a computer

Description automatically generated

DELETE FROM Customer WHERE Customer\_ID = 7001;

A screenshot of a computer

Description automatically generated

ROLLBACK;

A screenshot of a message

Description automatically generated

SELECT \* FROM Customer;

A screenshot of a computer

Description automatically generated

1. Query : Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed.

SELECT \* FROM Employee WHERE Employee\_ID = 122;

A screenshot of a computer

Description automatically generated

**ADVANCED QUERIES**  
**1. Aggregate sales by each Restaurant**

SELECT Restaurant.Name, COUNT(Orders.Order\_ID) AS Total\_Orders, SUM(Order\_total) AS Total\_Sales

FROM Orders

JOIN Employee ON Orders.Employee\_ID = Employee.Employee\_ID

JOIN Restaurant ON Employee.Restaurant\_ID = Restaurant.Restaurant\_ID

GROUP BY Restaurant.Name

ORDER BY Total\_Sales DESC;

A screenshot of a computer

Description automatically generated

**2. Employees with most sale**

WITH Restaurant\_Sales AS (

SELECT Restaurant\_ID, SUM(Total\_Amount) AS Total\_Sales

FROM Order

GROUP BY Restaurant\_ID

ORDER BY Total\_Sales DESC

LIMIT 3

)

SELECT Employee.Name, Employee.Role, R.Name AS Restaurant\_Name, SUM(Order.Total\_Amount) AS Employee\_Sales

FROM Order

JOIN Employee ON Order.Employee\_ID = Employee.Employee\_ID

JOIN Restaurant R ON Employee.Restaurant\_ID = R.Restaurant\_ID

JOIN Restaurant\_Sales RS ON R.Restaurant\_ID = RS.Restaurant\_ID

GROUP BY Employee.Employee\_ID, R.Name

ORDER BY Employee\_Sales DESC;