INTRODUCTION TO DATABASE

Q1)Problem Statement: There can be multiple customers, who can place multiple orders on the site. Now a sales person can handle these orders will distribute into multiple sales persons (One order will be assign to one salesperson only). So a sales person can have multiple orders of multiple customers Create Database:

Sol:

Creating the database named SalesDB.

```
mysql> CREATE DATABASE SalesDB;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| SalesDB            |
| aishwarya          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

Q2)Design Schema

Sol:

1. Customer – stores customer details.
2. SalesPerson – stores salesperson details.
3. Orders – stores each order with customer and salesperson references

Q3)Create table:

Sol:
Creating the three above mentioned tables in SalesDB by using the create command.

Sol:

```
mysql> use SalesDB;
Database changed
mysql> CREATE TABLE Customer (
    ->     customer_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     email VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE SalesPerson (
    ->     salesperson_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     region VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql> CREATE TABLE Orders (
    ->     order_id INT PRIMARY KEY,
    ->     customer_id INT,
    ->     salesperson_id INT,
    ->     order_date DATE,
    ->     amount DECIMAL(10,2),
    ->     FOREIGN KEY (customer_id) REFERENCES Customer(customer_id),
    ->     FOREIGN KEY (salesperson_id) REFERENCES SalesPerson(salesperson_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+------------------+
| Tables_in_SalesDB |
+------------------+
| Customer          |
| Orders            |
| SalesPerson       |
+------------------+
3 rows in set (0.00 sec)
```

Q4)Insert sample data

Sol:

```
mysql> INSERT INTO Customer (customer_id, name, email)
    -> VALUES
    ->   (1, 'Aishwarya', 'aishwarya@mail.com'),
    ->   (2, 'Rahul', 'rahul@mail.com'),
    ->   (3, 'Neha', 'neha@mail.com');
Query OK, 3 rows affected (0.03 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> INSERT INTO SalesPerson (salesperson_id, name, region)
    -> VALUES
    ->   (101, 'Raj', 'North'),
    ->   (102, 'Simran', 'East');
Query OK, 2 rows affected (0.04 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Orders (order_id, customer_id, salesperson_id, order_date, amount)
    -> VALUES
    ->   (1001, 1, 101, '2025-06-10', 5000.00),
    ->   (1002, 2, 101, '2025-06-11', 6000.00),
    ->   (1003, 1, 102, '2025-06-12', 7000.00),
    ->   (1004, 3, 101, '2025-06-13', 5500.00);
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

Q5)Find the sales person have multiple orders.

Sol:

```
mysql> SELECT salesperson_id, COUNT(*) AS total_orders
    -> FROM Orders
    -> GROUP BY salesperson_id
    -> HAVING COUNT(*) > 1;
+----------------+--------------+
| salesperson_id | total_orders |
+----------------+--------------+
|            101 |            3 |
+----------------+--------------+
1 row in set (0.00 sec)
```

Q6)Find the all sales person details along with order details

Sol:

Applying inner join here:

```
mysql>
mysql> SELECT s.salesperson_id, s.name AS salesperson_name, s.region,
    ->        o.order_id, o.order_date, o.amount
    -> FROM SalesPerson s
    -> JOIN Orders o ON s.salesperson_id = o.salesperson_id;
+----------------+------------------+--------+----------+------------+---------+
| salesperson_id | salesperson_name | region | order_id | order_date | amount  |
+----------------+------------------+--------+----------+------------+---------+
|            101 | Raj              | North  |     1001 | 2025-06-10 | 5000.00 |
|            101 | Raj              | North  |     1002 | 2025-06-11 | 6000.00 |
|            101 | Raj              | North  |     1004 | 2025-06-13 | 5500.00 |
|            102 | Simran           | East   |     1003 | 2025-06-12 | 7000.00 |
+----------------+------------------+--------+----------+------------+---------+
4 rows in set (0.00 sec)
```

Q7)Create index

Sol:

```
mysql> CREATE INDEX IDX_CUST_ID ON Orders(Customer_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Q8)How to show index on a table

Sol:

```
mysql> show INDEX FROM Orders;
+--------+------------+---------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Table  | Non_unique | Key_name      | Seq_in_index | Column_name   | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+--------+------------+---------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
| Orders |          0 | PRIMARY       |            1 | order_id      | A         |           4 |     NULL |   NULL |      | BTREE      |         |               | YES     | NULL       |
| Orders |          1 | salesperson_id |            1 | salesperson_id | A         |           2 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
| Orders |          1 | IDX_CUST_ID   |            1 | customer_id   | A         |           3 |     NULL |   NULL | YES  | BTREE      |         |               | YES     | NULL       |
+--------+------------+---------------+--------------+---------------+-----------+-------------+----------+--------+------+------------+---------+---------------+---------+------------+
3 rows in set (0.02 sec)
```

Q9)Find the order number, sales person name, along with the customer to whom that order belongs to

Sol:

```
mysql> SELECT o.order_id, s.name AS salesperson_name, c.name AS customer_name
    -> FROM Orders o
    -> JOIN SalesPerson s ON o.salesperson_id = s.salesperson_id
    -> JOIN Customer c ON o.customer_id = c.customer_id;
+----------+-----------------+---------------+
| order_id | salesperson_name | customer_name |
+----------+-----------------+---------------+
|     1001 | Raj             | Aishwarya     |
|     1003 | Simran          | Aishwarya     |
|     1002 | Raj             | Rahul         |
|     1004 | Raj             | Neha          |
+----------+-----------------+---------------+
4 rows in set (0.00 sec)
```