SQL QUERIES
Online Food Delivery System
(505,515,522,523)

Hard-coded values for tables

```
mysql> desc customer;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| c_no   | int         | NO   | PRI | NULL    |       |
| c_name | varchar(50) | NO   |     | NULL    |       |
| c_mob  | varchar(10) | NO   | UNI | NULL    |       |
| c_dob  | date        | YES  |     | NULL    |       |
| c_city | varchar(50) | YES  |     | Pune    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

```
mysql> select * from customer ;
+------+-----------------+------------+------------+----------+
| c_no | c_name          | c_mob      | c_dob      | c_city   |
+------+-----------------+------------+------------+----------+
|    1 | Aditi Kulkarni  | 9876543210 | 1995-04-15 | Pune     |
|    2 | Sneha Patil     | 9123456789 | 1998-08-10 | Mumbai   |
|    3 | Priya Deshmukh  | 9876501234 | 1993-02-25 | Nagpur   |
|    4 | Rohan Jadhav    | 9123487654 | 1990-12-05 | Kolhapur |
|    5 | Shreya Joshi    | 1234567889 | 1998-04-02 | Pune     |
|    6 | Arjun Kale      | 9123001122 | 1992-09-15 | Pune     |
|    7 | Ritika Gokhale  | 9876542233 | 1996-07-21 | Mumbai   |
|    8 | Ankit Joshi     | 9123445566 | 1994-01-19 | Mumbai   |
|    9 | Sakshi Khedekar | 9876503344 | 1997-11-30 | Pune     |
|   10 | Vikram Patil    | 9123004455 | 1989-03-12 | Thane    |
|  101 | Shekhar         | 1234567890 | 1999-05-04 | Mumbai   |
+------+-----------------+------------+------------+----------+
11 rows in set (0.00 sec)
```

```
mysql> select * from primecustomer;
+------+------------+------------+-------------+
| c_no | dom_start  | dom_end    | amount_paid |
+------+------------+------------+-------------+
|    1 | 2023-01-01 | 2023-12-31 |        6000 |
|    2 | 2023-02-01 | 2023-12-31 |        4500 |
|    3 | 2023-03-01 | 2023-12-31 |        5000 |
|    4 | 2023-04-01 | 2023-12-31 |        5500 |
|    5 | 2024-05-05 | 2024-06-05 |        1000 |
|    6 | 2023-06-01 | 2023-12-31 |        4800 |
|    7 | 2023-07-01 | 2023-12-31 |        5200 |
|    8 | 2023-08-01 | 2023-12-31 |        4700 |
|    9 | 2023-09-01 | 2023-12-31 |        5300 |
|   10 | 2023-10-01 | 2023-12-31 |        4900 |
+------+------------+------------+-------------+
10 rows in set (0.00 sec)
```

```
mysql> desc primecustomer;
+-------------+--------+------+-----+---------+-------+
| Field       | Type   | Null | Key | Default | Extra |
+-------------+--------+------+-----+---------+-------+
| c_no        | int    | NO   | PRI | NULL    |       |
| dom_start   | date   | YES  |     | NULL    |       |
| dom_end     | date   | YES  |     | NULL    |       |
| amount_paid | double | NO   |     | NULL    |       |
+-------------+--------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
mysql> desc regularcustomer;
+--------+------+------+-----+---------+-------+
| Field  | Type | Null | Key | Default | Extra |
+--------+------+------+-----+---------+-------+
| c_no   | int  | NO   | PRI | NULL    |       |
| points | int  | YES  |     | NULL    |       |
+--------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

```
mysql> select * from regularcustomer;
+------+--------+
| c_no | points |
+------+--------+
|    1 |    120 |
|    2 |    140 |
|    3 |    180 |
|    4 |    200 |
|    6 |    150 |
|    7 |    175 |
|    8 |    130 |
|    9 |    190 |
|   10 |    160 |
+------+--------+
9 rows in set (0.00 sec)
```

```
mysql> desc menu;
+-------------+-----------------------+------+-----+---------+----------------+
| Field       | Type                  | Null | Key | Default | Extra          |
+-------------+-----------------------+------+-----+---------+----------------+
| m_id        | int                   | NO   | PRI | NULL    | auto_increment |
| m_name      | varchar(50)           | YES  | UNI | NULL    |                |
| m_price     | double                | NO   |     | NULL    |                |
| m_type      | enum('Veg','Non-Veg') | YES  |     | NULL    |                |
| m_available | tinyint(1)            | YES  |     | NULL    |                |
+-------------+-----------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

```
mysql> select * from menu;
+------+------------------+---------+---------+-------------+
| m_id | m_name           | m_price | m_type  | m_available |
+------+------------------+---------+---------+-------------+
|    3 | Paratha          |     120 | Veg     |           0 |
|    4 | Misal Pao        |      90 | Veg     |           1 |
|    5 | Noodles          |     150 | Veg     |           1 |
|    6 | Manchurian       |     180 | Veg     |           1 |
|    7 | Fries            |      70 | Veg     |           1 |
|    8 | Omlette          |      85 | Non-Veg |           0 |
|    9 | Veg Burger       |     100 | Veg     |           0 |
|   10 | Chicken Burger   |     150 | Non-Veg |           1 |
|   11 | Lobster Curry    |     350 | Non-Veg |           1 |
|   12 | Mutton Biryani   |     250 | Non-Veg |           1 |
|   13 | Paneer Tikka     |     220 | Veg     |           0 |
|   14 | Lamb Chop        |     300 | Non-Veg |           1 |
|   15 | Veg Pulao        |     220 | Veg     |           1 |
|   16 | Butter Chicken   |     280 | Non-Veg |           0 |
|   17 | Prawn Masala     |     320 | Non-Veg |           1 |
|   18 | Veg Lasagna      |     260 | Veg     |           1 |
|   19 | Chicken Shawarma |     240 | Non-Veg |           1 |
|   20 | Steak            |     400 | Non-Veg |           0 |
|   21 | Sizzling Brownie |     250 | Veg     |           1 |
|   22 | Malai Rabdi      |     230 | Veg     |           1 |
|  201 | Methi Paratha    |     150 | Veg     |           0 |
|  301 | masala puri      |      35 | Veg     |           0 |
+------+------------------+---------+---------+-------------+
22 rows in set (0.00 sec)
```

```
mysql> desc orderitems;
+----------+--------+------+-----+---------+-------+
| Field    | Type   | Null | Key | Default | Extra |
+----------+--------+------+-----+---------+-------+
| o_id     | int    | NO   | PRI | NULL    |       |
| m_id     | int    | YES  | MUL | NULL    |       |
| quantity | int    | YES  |     | NULL    |       |
| o_price  | double | YES  |     | NULL    |       |
| c_no     | int    | YES  | MUL | NULL    |       |
+----------+--------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> select * from orderitems;
+------+------+----------+---------+------+
| o_id | m_id | quantity | o_price | c_no |
+------+------+----------+---------+------+
|    1 |    3 |        2 |     240 |    1 |
|    2 |    4 |        3 |     270 |    5 |
|    3 |    5 |        1 |     150 |    2 |
|    4 |    6 |        4 |     720 |    7 |
|    5 |    7 |        1 |      70 |    1 |
|    6 |   10 |        2 |     300 |    3 |
|    7 |   11 |        3 |    1050 |    8 |
|    8 |   12 |        1 |     250 |    4 |
|    9 |   14 |        2 |     600 |    5 |
|   10 |   17 |        3 |     960 |    2 |
|   11 |   21 |        1 |     250 |   10 |
+------+------+----------+---------+------+
11 rows in set (0.00 sec)
```

```
mysql> desc payment;
+----------+--------------------------+------+-----+---------+-------+
| Field    | Type                     | Null | Key | Default | Extra |
+----------+--------------------------+------+-----+---------+-------+
| p_id     | int                      | NO   | PRI | NULL    |       |
| o_id     | int                      | YES  | MUL | NULL    |       |
| p_date   | date                     | YES  |     | NULL    |       |
| p_method | enum('Cash','UPI','Card') | YES  |     | NULL    |       |
| p_status | enum('Paid','Not paid')  | YES  |     | NULL    |       |
+----------+--------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> select * from payment;
+------+------+------------+----------+----------+
| p_id | o_id | p_date     | p_method | p_status |
+------+------+------------+----------+----------+
|    1 |    1 | 2023-11-12 | Cash     | Paid     |
|    2 |    2 | 2023-11-12 | UPI      | Paid     |
|    3 |    3 | 2023-11-12 | Card     | Paid     |
|    4 |    4 | 2023-11-12 | Cash     | Paid     |
|    5 |    5 | 2023-11-12 | Card     | Paid     |
|    6 |    6 | 2023-11-12 | UPI      | Paid     |
|    7 |    7 | 2023-11-12 | Cash     | Paid     |
|    8 |    8 | 2023-11-12 | Card     | Paid     |
|    9 |    9 | 2023-11-12 | UPI      | Paid     |
|   10 |   10 | 2023-11-12 | Cash     | Paid     |
|  607 |    2 | 2024-09-09 | Cash     | Paid     |
+------+------+------------+----------+----------+
11 rows in set (0.00 sec)
```

TRIGGER (to calculate bill amount) :-

```
 DELIMITER //
mysql> CREATE Trigger calc_price
    -> BEFORE INSERT on OrderItems
    -> for each ROW
    -> BEGIN
    -> DECLARE item_price DOUBLE;
    -> DECLARE item_quantity INT;
    ->
    -> SELECT m_price into item_price from Menu WHERE m_id =
NEW.m_id ;
    ->
    ->
    -> SET NEW.o_price = item_price * NEW.quantity ;
    -> END //
Query OK, 0 rows affected (0.04 sec)
mysql> DELIMITER ;
```

Question 1 : Show Customers who have not

completed payment.

PROCEDURE

```
DELIMITER //
CREATE PROCEDURE customer_notpaid()
BEGIN
    SELECT DISTINCT c.c_name
    FROM Customer c
    LEFT JOIN OrderItems oi ON c.c_no = oi.c_no
    LEFT JOIN Payment p ON oi.o_id = p.o_id
    WHERE p.p_status = 'Not paid' OR p.o_id IS NULL;
END //
DELIMITER ;
mysql> CALL customer_notpaid();
+----------------+
| c_name |
+----------------+
| Rohan Jadhav |
| Ritika Gokhale |
+----------------+
2 rows in set (0.00 sec)
```

Question 2: Show Customers who have ordered Non-veg food.

(using Subqeury)

```
SELECT c_name
FROM Customer
WHERE c_no IN (
    SELECT c_no
    FROM OrderItems
    WHERE m_id IN (
        SELECT m_id
        FROM Menu
        WHERE m_type = 'Non-Veg'
    )
);
+----------------+
```

```
| c_name |
+----------------+
| Sneha Patil |
| Priya Deshmukh |
| Rohan Jadhav |
| Mahesh Shinde |
| Ankit Joshi |
+----------------+
5 rows in set (0.00 sec)
```

Question 3: Calculate Customers' age and their names in Ascending order of their age.

FUNCTION

```
create function calc_age(c_dob date)
    -> returns int
    -> deterministic
    -> begin
    -> declare age int;
    -> set age=0;
    -> SET age = TIMESTAMPDIFF(YEAR, c_dob, CURDATE());
    -> return age;
    -> end //
Query OK, 0 rows affected (0.09 sec)
```

Calling function, and using ORDER BY clause

```
 select c_name, Calc_age(c_dob) as age from customer order by
age;
+------+
| age |
+------+
| 26 |
| 29 |
| 31 |
| 33 |
| 36 |
```

```
+------+
5 rows in set (0.01 sec)
```

Question 4: Display Menu Items with price between 100 and 200.

Using BETWEEN clause

```
select * from menu where _price between 100 and 200);
+------+-----------+---------+--------+-------------+
| m_id | m_name | m_price | m_type | m_available |
+------+-----------+---------+--------+-------------+
| 1 | Paratha | 120 | Veg | 1 | | 2 | Misal Pao | 90
| Veg | 1 | | 3 | Noodles | 150 | Veg | 1 | | 4 |
Manchurian     |     180     |     Veg     |     1     |
+------+-----------+---------+--------+-------------+
4 rows in set (0.01 sec)
```

Question 5: Find Customers who are born in 1998.

Using LIKE clause

```
select * from customer where c_dob like '1998-__-__';
+------+-------------+-----------+-----------+--------+
| c_no | c_name | c_mob | c_dob | c_city |
+------+-------------+-----------+-----------+--------+
| 2 | Sneha Patil | 9123456789 | 1998-08-10 | Mumbai |
+------+-------------+-----------+-----------+--------+
1 row in set (0.00 sec);
```

Question 6: Show PrimeCustomer IDs along with how many days membership they have.

Query on Date

```
mysql> SELECT c_no, DATEDIFF(dom_end, dom_start) AS days_diff
from PrimeCustomer;
+------+-----------+
| c_no | days_diff |
+------+-----------+
| 1 | 364 |
| 2 | 333 |
| 3 | 305 |
| 4 | 274 |
| 5 | 244 |
| 6 | 213 |
| 7 | 183 |
| 8 | 152 |
| 9 | 121 |
| 10 | 91 |
+------+-----------+
10 rows in set (0.00 sec)
```

Question 7: Display number of Veg and Non-veg orders

Using GROUP BY clause

```
SELECT m_type, COUNT(DISTINCT o_id) AS num_orders
FROM menu
JOIN orderitems ON menu.m_id = orderitems.m_id
GROUP BY m_type;
+---------+------------+
| m_type | num_orders |
+---------+------------+
| Veg | 6 |
| Non-Veg | 5 |
+---------+------------+
2 rows in set (0.00 sec)
```

Question 8: Display RegularCustomers with less than 150.

Using INNER JOIN

```
SELECT c_name, c_no
FROM customer
INNER JOIN regularcustomer ON customer.c_no =
regularcustomer.c_no
WHERE points < 150;
+------+----------------+-----------+-----------+--------+----
--+--------+
| c_no | c_name | c_mob | c_dob | c_city | c_no | points |
+------+----------------+-----------+-----------+--------+----
--+--------+
| 1 | Aditi Kulkarni | 9876543210 | 1995-04-15 | Pune | 1 |
120 |
| 2 | Sneha Patil | 9123456789 | 1998-08-10 | Mumbai | 2 |
140 |
+------+----------------+-----------+-----------+--------+----
--+--------+
2 rows in set (0.00 sec)
```

Question 9: Display Customers who have done payment through UPI.

Using OUTER JOIN

```
SELECT c.c_name, p.p_status, p.p_method
FROM Customer c
LEFT JOIN RegularCustomer rc ON c.c_no = rc.c_no
LEFT JOIN PrimeCustomer pc ON c.c_no = pc.c_no
LEFT JOIN Payment p ON p.o_id IN (SELECT oi.o_id FROM OrderItems
oi WHERE oi.c_no = c.c_no)
WHERE p.p_method = 'UPI';
+---------------+----------+----------+
| c_name | p_status | p_method |
+---------------+----------+----------+
| Sneha Patil | Paid | UPI |
| Mahesh Shinde | Not paid | UPI |
+---------------+----------+----------+
2 rows in set (0.00 sec)
```

Question 10: Display a View of Orders and Prices.

Create and display a view of OrderItems table

```
mysql> CREATE View v1 AS SELECT o_id, o_price from OrderItems;
Query OK, 0 rows affected (0.01 sec)

mysql> SET Profiling=1;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> SELECT * from OrderItems WHERE o_price<100;
+------+------+----------+---------+------+
| o_id | m_id | quantity | o_price | c_no |
+------+------+----------+---------+------+
| 5 | 7 | 1 | 70 | 1 |
+------+------+----------+---------+------+
1 row in set (0.01 sec)

mysql> SHOW profiles;
+----------+------------+--------------------------------------
-----+
| Query_ID | Duration | Query
|
+----------+------------+--------------------------------------
-----+
| 1 | 0.00480400 | SELECT * from OrderItems WHERE
o_price<100 |
+----------+------------+--------------------------------------
-----+
1 row in set, 1 warning (0.00 sec)

mysql> SELECT * from v1;
+------+---------+
| o_id | o_price |
+------+---------+
| 1 | 240 |
| 2 | 270 |
```

```
| 3 | 150 |
| 4 | 720 |
| 5 | 70 |
| 6 | 300 |
| 7 | 1050 |
| 8 | 250 |
| 9 | 600 |
| 10 | 960 |
| 11 | 250 |
+------+---------+
11 rows in set (0.01 sec)
```

Question 11 : Find number of Customers living in Pune.

Using aggregate function

```
mysql> SELECT COUNT(*) AS num_customers FROM Customer WHERE
c_city = 'Pune';
+---------------+
| num_customers |
+---------------+
| 3 |
+---------------+
1 row in set (0.01 sec)
```

Question 12: Show Customers whose membership started before 1st June 2023.

Using Join

```
SELECT c.c_no, c.c_name
FROM Customer c
JOIN PrimeCustomer pc ON c.c_no = pc.c_no
WHERE pc.dom_start < '2023-06-01';
+------+----------------+
| c_no | c_name |
+------+----------------+
```

```
| 1 | Aditi Kulkarni |
| 2 | Sneha Patil |
| 3 | Priya Deshmukh |
| 4 | Rohan Jadhav |
| 5 | Mahesh Shinde |
+------+----------------+
5 rows in set (0.00 sec)
```

Question 13: Group Customers by mode of payment.

Using GROUP BY clause

```
  mysql> SELECT p.p_method AS payment_method, COUNT(*) AS
customer_count
    -> FROM Payment p
    -> WHERE p.p_status = 'Paid'
    -> GROUP BY p.p_method;
+----------------+----------------+
| payment_method | customer_count |
+----------------+----------------+
| Cash | 3 |
| UPI | 3 |
| Card | 1 |
+----------------+----------------+
3 rows in set (0.00 sec)
```

Question 14: Find items in Menu having Chicken

Using LIKE clause

```
mysql> SELECT m_name, m_price, m_type
    -> FROM Menu
    -> WHERE m_name LIKE '%chicken%';
+-----------------+---------+---------+
| m_name | m_price | m_type |
+-----------------+---------+---------+
```

```
| Chicken Burger | 150 | Non-Veg |
| Butter Chicken | 280 | Non-Veg |
| Chicken Shawarma | 240 | Non-Veg |
+------------------+---------+---------+
3 rows in set (0.00 sec)
```

Question 15: Show number of available and not available items from Menu.

```
mysql> SELECT m_available AS availability_status, COUNT(*) AS
item_count
    -> FROM Menu
    -> GROUP BY m_available;
+---------------------+------------+
| availability_status | item_count |
+---------------------+------------+
| 1 | 15 |
| 0 | 6 |
+---------------------+------------+
2 rows in set (0.00 sec)
```

Question 16: Display Customers who have placed multiple orders.

```
Using JOIN & Group By
SELECT c.c_no, c.c_name, COUNT(oi.o_id) AS items_ordered
FROM Customer c
JOIN OrderItems oi ON c.c_no = oi.c_no
GROUP BY c.c_no, c.c_name
HAVING COUNT(oi.o_id) >
+------+---------------+---------------+
| c_no | c_name | items_ordered |
+------+---------------+---------------+
| 1 | Aditi Kulkarni | 2 |
| 2 | Sneha Patil | 2 |
| 5 | Mahesh Shinde | 2 |
+------+---------------+---------------+
```

```
3 rows in set (0.00 sec)
```

Question 17: Show total Bill Amount of each Customer.

```
Using JOIN & aggregate function

SELECT c.c_no, c.c_name, SUM(oi.o_price) AS bill_amount

FROM Customer c

JOIN OrderItems oi ON c.c_no = oi.c_no
GROUP BY c.c_no, c.c_name;
+------+----------------+-------------+
| c_no | c_name | bill_amount |
+------+----------------+-------------+
| 1 | Aditi Kulkarni | 310 |
| 5 | Mahesh Shinde | 870 |
| 2 | Sneha Patil | 1110 |
| 7 | Ritika Gokhale | 720 |
| 3 | Priya Deshmukh | 300 |
| 8 | Ankit Joshi | 1050 |
| 4 | Rohan Jadhav | 250 |
| 10 | Vikram Patil | 250 |
+------+----------------+-------------+
8 rows in set (0.00 sec)
```

Question 18: Show Customers ordered by their Points

```
Using INNER JOIN

SELECT c.c_name, rc.points

FROM Customer c
INNER JOIN RegularCustomer rc ON c.c_no = rc.c_no
ORDER BY rc.points;mysql> SELECT c.c_name, rc.points
-> FROM Customer c
    -> INNER JOIN RegularCustomer rc ON c.c_no = rc.c_no
    -> ORDER BY rc.points;
```

```
+-----------------+--------+
| c_name | points |
+-----------------+--------+
| Aditi Kulkarni | 120 |
| Ankit Joshi | 130 |
| Sneha Patil | 140 |
| Arjun Kale | 150 |
| Mahesh Shinde | 160 |
| Vikram Patil | 160 |
| Ritika Gokhale | 175 |
| Priya Deshmukh | 180 |
| Sakshi Khedekar | 190 |
| Rohan Jadhav | 200 |
+-----------------+--------+
10 rows in set (0.00 sec)
```

Question 19 : Show Customers who have order Sizzling Brownie

Using JOIN and Subquery

```
mysql> SELECT c_name
    -> FROM Customer
    -> WHERE c_no IN (
    -> SELECT o.c_no
    -> FROM OrderItems o
    -> JOIN Menu m ON o.m_id = m.m_id
    -> WHERE m.m_name = 'Sizzling Brownie'
    -> );
+--------------+
| c_name |
+--------------+
| Vikram Patil |
+--------------+
1 row in set (0.00 sec)
```

Question 20 :Display Menu Items in order of increasing prices.

Using ORDER BY clause

```
mysql> SELECT m_name, m_price from Menu ORDER BY m_price;
+--------------------+---------+
| m_name | m_price |
+--------------------+---------+
| Fries | 70 |
| Omlette | 85 |
| Misal Pao | 90 |
| Veg Burger | 100 |
| Paratha | 120 |
| Noodles | 150 |
| Chicken Burger | 150 |
| Manchurian | 180 |
| Nutella Cheesecake | 200 |
| Paneer Tikka | 220 |
| Veg Pulao | 220 |
| Malai Rabdi | 230 |
| Chicken Shawarma | 240 |
| Mutton Biryani | 250 |
| Sizzling Brownie | 250 |
| Veg Lasagna | 260 |
| Butter Chicken | 280 |
| Lamb Chop | 300 |
| Prawn Masala | 320 |
| Lobster Curry | 350 |
| Steak | 400 |
+--------------------+---------+
21 rows in set (0.00 sec)
```