Basic format

Creating the table

```
CREATE TABLE reaction (
   id INT AUTO_INCREMENT PRIMARY KEY,
   user_name VARCHAR(100),
   reaction_type VARCHAR(20),
   post_id INT,
   created_at DATETIME,
   location VARCHAR(100),
   mood_level INT,
   comment TEXT
);
```

Inserting the data

```
INSERT INTO reaction (user_name, reaction_type, post_id, created_at, location, mood_level, comment) VALUES
('Alice', 'like', 101, '2025-08-07 10:15:00', 'New York', 8, 'Nice post!'),
('Bob', 'love', 102, '2025-08-06 14:20:00', 'Los Angeles', 9, 'Awesome!'),
('Charlie', 'angry', 101, '2025-08-05 09:10:00', 'Chicago', 3, NULL),
('Diana', 'wow', 103, '2025-08-07 08:30:00', 'Miami', 7, 'Interesting point.'),
('Ethan', 'sad', 104, '2025-08-04 16:45:00', 'Dallas', 2, NULL),
('Fiona', 'love', 102, '2025-08-07 12:00:00', 'Boston', 6, 'Well said.'),
('George', 'like', 105, '2025-08-03 11:25:00', 'Seattle', 5, NULL),
('Hannah', 'like', 106, '2025-08-07 15:50:00', 'Denver', 9, 'Completely agree!'),
('Ian', 'angry', 107, '2025-08-06 13:15:00', 'Phoenix', 4, 'Not okay with this.'),
('Jane', 'sad', 108, '2025-08-02 18:40:00', 'Atlanta', 1, NULL),
('Sam', 'like', 101, '2025-08-07 10:50:00', 'New York', 8, 'NULL'),
('Anita', 'wow', 109, '2025-08-01 20:10:00', 'Houston', 7, NULL),
('Brian', 'love', 110, '2025-08-07 09:05:00', 'San Francisco', 10, 'Fantastic work!'),
('Catherine', 'like', 111, '2025-08-07 08:55:00', 'New York', 6, NULL),
```

('Daniel', 'angry', 112, '2025-08-05 17:35:00', 'Chicago', 2, NULL);

LIKE
Write a query to find all reactions made by users whose names start with the letter 'A'.

ANS:

mysql> select * from reaction where user_name like 'A%';

```
+----+-----------+---------------+---------+---------------------+----------+------------+------------+
| id | user_name | reaction_type | post_id | created_at          | location | mood_level | comment    |
+----+-----------+---------------+---------+---------------------+----------+------------+------------+
|  1 | Alice     | like          |     101 | 2025-08-07 10:15:00 | New York |          8 | Nice post! |
| 12 | Anita     | wow           |     109 | 2025-08-01 20:10:00 | Houston  |          7 | NULL       |
+----+-----------+---------------+---------+---------------------+----------+------------+------------+
```

AS (Alias)
Write a query to display user_name as Reactor and reaction_type as Type.

ANS: mysql> select user_name as reactor,reaction_type as type from reaction;

```
+-----------+-------+
| reactor   | type  |
+-----------+-------+
| Alice     | like  |
| Bob       | love  |
| Charlie   | angry |
| Diana     | wow   |
| Ethan     | sad   |
| Fiona     | love  |
| George    | like  |
| Hannah    | like  |
| Ian       | angry |
| Jane      | sad   |
| Sam       | like  |
| Anita     | wow   |
| Brian     | love  |
```

| Catherine | like |

| Daniel | angry |

+-----------+-------+

BETWEEN

Write a query to find all reactions where the mood level is between 4 and 8.

ANS: mysql> select reaction_type,mood_level from reaction where mood_level between 4 and 8;

+---------------+------------+

| reaction_type | mood_level |

+---------------+------------+

| like | 8 |

| wow | 7 |

| love | 6 |

| like | 5 |

| angry | 4 |

| like | 8 |

| wow | 7 |

| like | 6 |

+---------------+------------+

AND

Write a query to get all reactions where the type is 'like' and the mood level is above 7.

ANS: mysql> select * from reaction where reaction_type='like' and mood_level>7;

+----+-----------+---------------+---------+---------------------+----------+------------+-------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+-----------+---------------+---------+---------------------+----------+------------+-------------------+

| 1 | Alice | like | 101 | 2025-08-07 10:15:00 | New York | 8 | Nice post! |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 11 | Sam | like | 101 | 2025-08-07 10:50:00 | New York | 8 | NULL |

+----+-----------+---------------+---------+---------------------+----------+------------+-------------------+

OR
Write a query to list all reactions where the type is either 'love' or 'angry'.

ANS: mysql> select * from reaction where reaction_type='like' or reaction_type='angry';

```
+----+-----------+---------------+---------+---------------------+----------+------------+---------------------+
| id | user_name | reaction_type | post_id | created_at          | location | mood_level | comment             |
+----+-----------+---------------+---------+---------------------+----------+------------+---------------------+
|  1 | Alice     | like          |     101 | 2025-08-07 10:15:00 | New York |          8 | Nice post!          |
|  3 | Charlie   | angry         |     101 | 2025-08-05 09:10:00 | Chicago  |          3 | NULL                |
|  7 | George    | like          |     105 | 2025-08-03 11:25:00 | Seattle  |          5 | NULL                |
|  8 | Hannah    | like          |     106 | 2025-08-07 15:50:00 | Denver   |          9 | Completely agree!   |
|  9 | Ian       | angry         |     107 | 2025-08-06 13:15:00 | Phoenix  |          4 | Not okay with this. |
| 11 | Sam       | like          |     101 | 2025-08-07 10:50:00 | New York |          8 | NULL                |
| 14 | Catherine | like          |     111 | 2025-08-07 08:55:00 | New York |          6 | NULL                |
| 15 | Daniel    | angry         |     112 | 2025-08-05 17:35:00 | Chicago  |          2 | NULL                |
+----+-----------+---------------+---------+---------------------+----------+------------+---------------------+
```

NOT
Write a query to show all reactions that are not of type 'sad'.

ANS: mysql> select * from reaction where not reaction_type= 'sad';

```
+----+-----------+---------------+---------+---------------------+-------------+------------+---------------------+
| id | user_name | reaction_type | post_id | created_at          | location    | mood_level | comment             |
+----+-----------+---------------+---------+---------------------+-------------+------------+---------------------+
|  1 | Alice     | like          |     101 | 2025-08-07 10:15:00 | New York    |          8 | Nice post!          |
|  2 | Bob       | love          |     102 | 2025-08-06 14:20:00 | Los Angeles |          9 | Awesome!            |
|  3 | Charlie   | angry         |     101 | 2025-08-05 09:10:00 | Chicago     |          3 | NULL                |
|  4 | Diana     | wow           |     103 | 2025-08-07 08:30:00 | Miami       |          7 | Interesting point.  |
|  6 | Fiona     | love          |     102 | 2025-08-07 12:00:00 | Boston      |          6 | Well said.          |
|  7 | George    | like          |     105 | 2025-08-03 11:25:00 | Seattle     |          5 | NULL                |
```

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 9 | Ian | angry | 107 | 2025-08-06 13:15:00 | Phoenix | 4 | Not okay with this. |

| 11 | Sam | like | 101 | 2025-08-07 10:50:00 | New York | 8 | NULL |

| 12 | Anita | wow | 109 | 2025-08-01 20:10:00 | Houston | 7 | NULL |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

| 14 | Catherine | like | 111 | 2025-08-07 08:55:00 | New York | 6 | NULL |

| 15 | Daniel | angry | 112 | 2025-08-05 17:35:00 | Chicago | 2 | NULL |

+----+----------+-------------+---------+---------------------+---------------+------------+--------------------+

IN

Write a query to find all reactions where the reaction type is 'like', 'wow', or 'love'.

ANS: mysql> select * from reaction where reaction_type in('like','wow','love');

+----+----------+-------------+---------+---------------------+---------------+------------+--------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+-------------+---------+---------------------+---------------+------------+--------------------+

| 1 | Alice | like | 101 | 2025-08-07 10:15:00 | New York | 8 | Nice post! |

| 2 | Bob | love | 102 | 2025-08-06 14:20:00 | Los Angeles | 9 | Awesome! |

| 4 | Diana | wow | 103 | 2025-08-07 08:30:00 | Miami | 7 | Interesting point. |

| 6 | Fiona | love | 102 | 2025-08-07 12:00:00 | Boston | 6 | Well said. |

| 7 | George | like | 105 | 2025-08-03 11:25:00 | Seattle | 5 | NULL |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 11 | Sam | like | 101 | 2025-08-07 10:50:00 | New York | 8 | NULL |

| 12 | Anita | wow | 109 | 2025-08-01 20:10:00 | Houston | 7 | NULL |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

| 14 | Catherine | like | 111 | 2025-08-07 08:55:00 | New York | 6 | NULL |

+----+----------+-------------+--------+---------------------+--------------+------------+--------------------+

IS NULL

Write a query to display all reactions where the comment is missing (i.e., NULL).

ANS: mysql> select * from reaction where comment is null;

+----+----------+--------------+---------+---------------------+----------+------------+---------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+--------------+---------+---------------------+----------+------------+---------+

| 3 | Charlie | angry | 101 | 2025-08-05 09:10:00 | Chicago | 3 | NULL |

| 5 | Ethan | sad | 104 | 2025-08-04 16:45:00 | Dallas | 2 | NULL |

| 7 | George | like | 105 | 2025-08-03 11:25:00 | Seattle | 5 | NULL |

| 10 | Jane | sad | 108 | 2025-08-02 18:40:00 | Atlanta | 1 | NULL |

| 11 | Sam | like | 101 | 2025-08-07 10:50:00 | New York | 8 | NULL |

| 12 | Anita | wow | 109 | 2025-08-01 20:10:00 | Houston | 7 | NULL |

| 14 | Catherine | like | 111 | 2025-08-07 08:55:00 | New York | 6 | NULL |

| 15 | Daniel | angry | 112 | 2025-08-05 17:35:00 | Chicago | 2 | NULL |

+----+----------+--------------+---------+---------------------+----------+------------+---------+

IS NOT NULL

Write a query to find all reactions that include a comment.

AND: mysql> select * from reaction where comment is not null;

+----+----------+-------------+--------+---------------------+--------------+------------+--------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+-------------+--------+---------------------+--------------+------------+--------------------+

| 1 | Alice | like | 101 | 2025-08-07 10:15:00 | New York | 8 | Nice post! |

| 2 | Bob | love | 102 | 2025-08-06 14:20:00 | Los Angeles | 9 | Awesome! |

| 4 | Diana | wow | 103 | 2025-08-07 08:30:00 | Miami | 7 | Interesting point. |

| 6 | Fiona | love | 102 | 2025-08-07 12:00:00 | Boston | 6 | Well said. |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 9 | Ian | angry | 107 | 2025-08-06 13:15:00 | Phoenix | 4 | Not okay with this. |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

+----+----------+-------------+--------+---------------------+---------------+------------+---------------------+

UPPER()
Write a query to show all usernames in uppercase.

ANS: mysql> select upper(user_name) from reaction;

+------------------+

| upper(user_name) |

+------------------+

| ALICE |

| BOB |

| CHARLIE |

| DIANA |

| ETHAN |

| FIONA |

| GEORGE |

| HANNAH |

| IAN |

| JANE |

| SAM |

| ANITA |

| BRIAN |

| CATHERINE |

| DANIEL |

+------------------+

LOWER()
Write a query to show all reaction types in lowercase.

ANS: mysql> select lower(user_name) from reaction;

+------------------+

| lower(user_name) |

+------------------+

| alice |

| bob |

| charlie |

| diana |

| ethan |

| fiona |

| george |

| hannah |

| ian |

| jane |

| sam |

| anita |

| brian |

| catherine |

| daniel |

+------------------+

LENGTH()

Write a query to find all users whose usernames are longer than 6 characters.

ANS: mysql> select user_name, length(user_name) as len from reaction where length(user_name)>6;

+-----------+------+

| user_name | len |

+-----------+------+

| Charlie | 7 |

| Catherine | 9 |

+-----------+------+

NOW()

Write a query to find all reactions made today (based on the created_at column).

ANS:mysql> select * from reaction where date_format(created_at,'%Y-%m-%d')=curdate();

+----+----------+-------------+---------+---------------------+--------------+------------+--------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+-------------+---------+---------------------+--------------+------------+--------------------+

| 1 | Alice | like | 101 | 2025-08-07 10:15:00 | New York | 8 | Nice post! |

| 4 | Diana | wow | 103 | 2025-08-07 08:30:00 | Miami | 7 | Interesting point. |

| 6 | Fiona | love | 102 | 2025-08-07 12:00:00 | Boston | 6 | Well said. |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 11 | Sam | like | 101 | 2025-08-07 10:50:00 | New York | 8 | NULL |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

| 14 | Catherine | like | 111 | 2025-08-07 08:55:00 | New York | 6 | NULL |

+----+----------+-------------+---------+---------------------+--------------+------------+--------------------

+DATE()

Write a query to extract and show only the date part from the created_at column.

ANS:mysql> select date(created_at) from reaction;

+------------------+

| date(created_at) |

+------------------+

| 2025-08-07 |

| 2025-08-06 |

| 2025-08-05 |

| 2025-08-07 |

| 2025-08-04 |

| 2025-08-07 |

| 2025-08-03 |

| 2025-08-07 |

| 2025-08-06 |

| 2025-08-02 |

| 2025-08-07 |

| 2025-08-01 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-05 |

+------------------+

LIKE with %
Write a query to find all users whose name contains the substring 'an'.

ANS: mysql> select * from reaction where user_name like'%an%';

+----+----------+-------------+---------+------------------+--------------+------------+---------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+-------------+---------+------------------+--------------+------------+---------------------+

| 4 | Diana | wow | 103 | 2025-08-07 08:30:00 | Miami | 7 | Interesting point. |

| 5 | Ethan | sad | 104 | 2025-08-04 16:45:00 | Dallas | 2 | NULL |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 9 | Ian | angry | 107 | 2025-08-06 13:15:00 | Phoenix | 4 | Not okay with this. |

| 10 | Jane | sad | 108 | 2025-08-02 18:40:00 | Atlanta | 1 | NULL |

| 12 | Anita | wow | 109 | 2025-08-01 20:10:00 | Houston | 7 | NULL |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

| 15 | Daniel | angry | 112 | 2025-08-05 17:35:00 | Chicago | 2 | NULL |

+----+----------+-------------+---------+------------------+--------------+------------+---------------------+

ROUND()
Write a query to display each user's mood level rounded to the nearest 5.

ANS:mysql> select user_name, round(mood_level/5.0)*5 from reaction;

+-----------+------------------------+

| user_name | round(mood_level/5.0)*5 |

+-----------+-------------------------+

| Alice | 10 |

| Bob | 10 |

| Charlie | 5 |

| Diana | 5 |

| Ethan | 0 |

| Fiona | 5 |

| George | 5 |

| Hannah | 10 |

| Ian | 5 |

| Jane | 0 |

| Sam | 10 |

| Anita | 5 |

| Brian | 10 |

| Catherine | 5 |

| Daniel | 0 |

+-----------+-------------------------+

SUBSTR + UPPER()

Write a query to show the first 2 letters of each user's name in uppercase.

ANS: mysql> select upper(substring(user_name,1,2))from reaction ;

+---------------------------------+

| upper(substring(user_name,1,2)) |

+---------------------------------+

| AL |

| BO |

| CH |

| DI |

| ET |

| FI |

| GE |

| HA |

| IA |

| JA |

| SA |

| AN |

| BR |

| CA |

| DA |

+---------------------------------+

NOT IN (with values)
Write a query to find all reactions not made on posts with IDs 110, 115, or 120.

ANS:

mysql> select * from reaction where post_id not in(101,115,120);

+----+----------+-------------+---------+---------------------+---------------+------------+---------------------+

| id | user_name | reaction_type | post_id | created_at | location | mood_level | comment |

+----+----------+-------------+---------+---------------------+---------------+------------+---------------------+

| 2 | Bob | love | 102 | 2025-08-06 14:20:00 | Los Angeles | 9 | Awesome! |

| 4 | Diana | wow | 103 | 2025-08-07 08:30:00 | Miami | 7 | Interesting point. |

| 5 | Ethan | sad | 104 | 2025-08-04 16:45:00 | Dallas | 2 | NULL |

| 6 | Fiona | love | 102 | 2025-08-07 12:00:00 | Boston | 6 | Well said. |

| 7 | George | like | 105 | 2025-08-03 11:25:00 | Seattle | 5 | NULL |

| 8 | Hannah | like | 106 | 2025-08-07 15:50:00 | Denver | 9 | Completely agree! |

| 9 | Ian | angry | 107 | 2025-08-06 13:15:00 | Phoenix | 4 | Not okay with this. |

| 10 | Jane | sad | 108 | 2025-08-02 18:40:00 | Atlanta | 1 | NULL |

| 12 | Anita | wow | 109 | 2025-08-01 20:10:00 | Houston | 7 | NULL |

| 13 | Brian | love | 110 | 2025-08-07 09:05:00 | San Francisco | 10 | Fantastic work! |

| 14 | Catherine | like | 111 | 2025-08-07 08:55:00 | New York | 6 | NULL |

| 15 | Daniel | angry | 112 | 2025-08-05 17:35:00 | Chicago | 2 | NULL |

+----+----------+----------+--------+---------------------+--------------+------------+---------------------+

COUNT and IS NULL

Write a query to count how many reactions have no comment.

ANS: mysql> select count(*) from reaction where comment is null;

+----------+

| count(*) |

+----------+

| 8 |

+----------+

Create the order table

```
CREATE TABLE orders (
   order_id INT AUTO_INCREMENT PRIMARY KEY,
   customer_name VARCHAR(100),
   product_name VARCHAR(100),
   order_date DATETIME,
   quantity INT,
   price DECIMAL(10,2),
   status VARCHAR(20),
   shipping_address TEXT
);
```

Insert the data

```
INSERT INTO orders (customer_name, product_name, order_date, quantity, price, status, shipping_address) VALUES
('Alice', 'Laptop', '2025-08-07 09:00:00', 1, 1200.00, 'shipped', '123 Main St, New York'),
('Bob', 'Phone', '2025-08-06 14:10:00', 2, 650.50, 'pending', '456 Elm St, LA'),
('Charlie', 'Tablet', '2025-08-05 11:25:00', 1, 300.00, 'cancelled', NULL),
```

('Diana', 'Monitor', '2025-08-07 15:40:00', 3, 199.99, 'shipped', '789 Pine St, Chicago'),

('Ethan', 'Keyboard', '2025-08-03 08:20:00', 5, 49.99, 'processing', NULL),

('Fiona', 'Mouse', '2025-08-04 10:30:00', 4, 25.00, 'shipped', '321 Oak St, Houston');

LIKE

1. **Write a query to find all customers whose names end with 'a'.**

   ANS: mysql> select customer_name from orders where customer_name like '%a';

   +---------------+

   | customer_name |

   +---------------+

   | Diana |

   | Fiona |

   +---------------+

2. **Write a query to find orders where the product name contains the word 'Phone'.**

   ANS:

   mysql> select * from orders where product_name like 'phone';

   +----------+---------------+--------------+------------------+---------+--------+---------+------------------+

   | order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

   +----------+---------------+--------------+------------------+---------+--------+---------+------------------+

   | 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

   +----------+---------------+--------------+------------------+---------+--------+---------+------------------+

3. **Write a query to find orders where the product name is exactly 5 characters long.**

ANS: mysql> select * from orders where length(product_name)=5;

+----------+---------------+--------------+------------------+----------+--------+---------+---------------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+----------+---------------+--------------+------------------+----------+--------+---------+---------------------+

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 6 | Fiona | Mouse | 2025-08-04 10:30:00 | 4 | 25.00 | shipped | 321 Oak St, Houston |

+---------+--------------+------------+------------------+----------+--------+---------+--------------------+

AS(Alias)

4. **Write a query to show customer name as Buyer and price as Unit_Price.**

   ANS: mysql> select customer_name as Buyer,price as Unit_Price from orders;

   +---------+------------+
   | Buyer | Unit_Price |
   +---------+------------+
Alice	1200.00
Bob	650.50
Charlie	300.00
Diana	199.99
Ethan	49.99
Fiona	25.00
   +---------+------------+

5. **Write a query to show order ID and total amount (quantity × price) as Total_Cost.**

   ANS: mysql> select order_id,(quantity*price) as Total_Cost from orders;

   +----------+------------+

   | order_id | Total_Cost |

   +----------+------------+

   | 1 | 1200.00 |

   | 2 | 1301.00 |

   | 3 | 300.00 |

   | 4 | 599.97 |

   | 5 | 249.95 |

   | 6 | 100.00 |

   +----------+------------+

NOT

6. **Write a query to find all orders not placed by 'Bob'.**

   ANS: mysql> select * from orders where not customer_name='bob';

   +----------+--------------+------------+------------------+---------+--------+-----------+-----------------
   ----+

   | order_id | customer_name | product_name | order_date | quantity | price | status
   | shipping_address |

   +----------+--------------+------------+------------------+---------+--------+-----------+-----------------
   ----+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL |

| 4 | Diana | Monitor | 2025-08-07 15:40:00 | 3 | 199.99 | shipped | 789 Pine St, Chicago |

| 5 | Ethan | Keyboard | 2025-08-03 08:20:00 | 5 | 49.99 | processing | NULL |

| 6 | Fiona | Mouse | 2025-08-04 10:30:00 | 4 | 25.00 | shipped | 321 Oak St, Houston |

+---------+-------------+------------+-----------------+---------+--------+-----------+-----------------+

7. **Write a query to find orders where status is not 'shipped'.**

ANS:

mysql> select * from orders where not status='shipped';

+---------+-------------+------------+-----------------+---------+-------+-----------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+-----------------+---------+-------+-----------+-----------------+

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL |

| 5 | Ethan | Keyboard | 2025-08-03 08:20:00 | 5 | 49.99 | processing | NULL |

+---------+-------------+------------+-----------------+---------+-------+-----------+-----------------+

AND

8. **Write a query to find orders with quantity greater than 2 and price less than 500.**

ANS: mysql> select * from orders where quantity >2 and price<500;

+---------+-------------+------------+-----------------+---------+-------+-----------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+-----------------+---------+-------+-----------+-----------------+

| 4 | Diana | Monitor | 2025-08-07 15:40:00 | 3 | 199.99 | shipped | 789 Pine St, Chicago |

| 5 | Ethan | Keyboard | 2025-08-03 08:20:00 | 5 | 49.99 | processing | NULL |

| 6 | Fiona | Mouse | 2025-08-04 10:30:00 | 4 | 25.00 | shipped | 321 Oak St, Houston |

+---------+-------------+------------+------------------+---------+--------+------------+-----------------+

9. **Write a query to find orders placed by 'Alice' and status is 'shipped'.**

ANS: mysql> select * from orders where customer_name='Alice' and status='shipped';

+---------+-------------+------------+------------------+---------+--------+--------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+------------------+---------+--------+--------+-----------------+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

+---------+-------------+------------+------------------+---------+--------+--------+-----------------+

OR

10. **Write a query to find orders where status is 'pending' or 'processing'.**

ANS: mysql> select * from orders where status='processing' or status='pending';

+---------+-------------+------------+------------------+---------+-------+------------+---------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+------------------+---------+-------+------------+---------------+

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 5 | Ethan | Keyboard | 2025-08-03 08:20:00 | 5 | 49.99 | processing | NULL |

+---------+-------------+------------+------------------+---------+-------+------------+---------------+

11. **Write a query to find orders made by 'Alice' or 'Diana'.**

ANS: mysql> select * from orders where customer_name='Alice' or customer_name='Diana';

+---------+-------------+------------+------------------+---------+--------+--------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+-----------------+---------+--------+--------+------------------+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

| 4 | Diana | Monitor | 2025-08-07 15:40:00 | 3 | 199.99 | shipped | 789 Pine St, Chicago |

+---------+-------------+------------+-----------------+---------+--------+--------+------------------+

IN

12. **Write a query to find orders for products in the categories: 'Laptop', 'Phone', or 'Tablet'.**

ANS: mysql> select * from orders where product_name in('Laptop', 'Phone', 'Tablet');

+---------+-------------+------------+-----------------+---------+--------+----------+------------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+-----------------+---------+--------+----------+------------------+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL |

+---------+-------------+------------+-----------------+---------+--------+----------+------------------+

13. **Write a query to find orders placed by customers in a given list: 'Alice', 'Bob', 'Charlie'.**

ANS: mysql> select * from orders where customer_name in('Alice', 'Bob', 'Charlie');

+---------+-------------+------------+-----------------+---------+--------+----------+------------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+------------+-----------------+---------+--------+----------+------------------+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL |

+---------+-------------+-----------+-----------------+---------+--------+-----------+-----------------+

IS NULL / IS NOT NULL

14. **Write a query to find orders where the shipping address is missing.**

ANS: mysql> select * from orders where shipping_address is null;

+---------+-------------+-----------+-----------------+---------+--------+-----------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+-----------+-----------------+---------+--------+-----------+-----------------+

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL |

| 5 | Ethan | Keyboard | 2025-08-03 08:20:00 | 5 | 49.99 | processing | NULL |

+---------+-------------+-----------+-----------------+---------+--------+-----------+-----------------+

15. **Write a query to find orders where the shipping address is provided.**

ANS:

mysql> select * from orders where shipping_address is not null;

+---------+-------------+-----------+-----------------+---------+--------+--------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status | shipping_address |

+---------+-------------+-----------+-----------------+---------+--------+--------+-----------------+

| 1 | Alice | Laptop | 2025-08-07 09:00:00 | 1 | 1200.00 | shipped | 123 Main St, New York |

| 2 | Bob | Phone | 2025-08-06 14:10:00 | 2 | 650.50 | pending | 456 Elm St, LA |

| 4 | Diana | Monitor | 2025-08-07 15:40:00 | 3 | 199.99 | shipped | 789 Pine St, Chicago |

| 6 | Fiona | Mouse | 2025-08-04 10:30:00 | 4 | 25.00 | shipped | 321 Oak St, Houston |

+----------+---------------+------------+------------------+---------+--------+--------+------------------
-+

FUNCTION-BASED

16. **Write a query to find customers whose name is longer than 5 characters.**

ANS: mysql> select * from orders where length(customer_name)>5;

+----------+---------------+------------+------------------+---------+--------+----------+-----------------+

| order_id | customer_name | product_name | order_date | quantity | price | status
| shipping_address |

+----------+---------------+------------+------------------+---------+--------+----------+-----------------+

| 3 | Charlie | Tablet | 2025-08-05 11:25:00 | 1 | 300.00 | cancelled | NULL
|

+----------+---------------+------------+------------------+---------+--------+----------+-----------------+

17. **Write a query to show all customer names in uppercase.**

ANS:

mysql> select upper(customer_name) from orders;

+----------------------+

| upper(customer_name) |

+----------------------+

| ALICE |

| BOB |

| CHARLIE |

| DIANA |

| ETHAN |

| FIONA |

+----------------------+

18. **Write a query to find orders where the product name is all lowercase.**

ANS: mysql> select lower(product_name) from orders;

+---------------------+

| lower(product_name) |

+---------------------+

| laptop |

| phone |

| tablet |

| monitor |

| keyboard |

| mouse |

+----------+

19. **Write a query to return the first 3 letters of each product name with an alias Short_Name.**

ANS: mysql> select left(product_name,3) as short_name from orders;

+------------+

| short_name |

+------------+

| Lap |

| Pho |

| Tab |

| Mon |

| Key |

| Mou |

+------------+