**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);**

mysql> select \* from reaction;

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

| 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. |

| 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 |

| 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 |

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

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

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.**

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.**

mysql> select \* from reaction where mood\_level between 4 and 8;

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

| 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. |

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

| 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 |

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

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

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

mysql> select reaction\_type from reaction where reaction\_type = 'like' and mood\_level>7;

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

| reaction\_type |

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

| like |

| like |

| like |

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

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

mysql> select reaction\_type from reaction where reaction\_type='love' or 'angry';

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

| reaction\_type |

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

| love |

| love |

| love |

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

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

mysql> select reaction\_type from reaction where not reaction\_type='sad';

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

| reaction\_type |

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

| like |

| love |

| angry |

| wow |

| love |

| like |

| like |

| angry |

| like |

| wow |

| love |

| like |

| angry |

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

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

mysql> select reaction\_type from reaction where reaction\_type in ('like','wow','love');

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

| reaction\_type |

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

| like |

| love |

| wow |

| love |

| like |

| like |

| like |

| wow |

| love |

| like |

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

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

mysql> select comment from reaction where comment is null;

+---------+

| comment |

+---------+

| NULL |

| NULL |

| NULL |

| NULL |

| NULL |

| NULL |

| NULL |

| NULL |

+---------+

**IS NOT NULL**  
**Wr** **mysql> select comment from reaction where comment is not null;**

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

| comment |

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

| Nice post! |

| Awesome! |

| Interesting point. |

| Well said. |

| Completely agree! |

| Not okay with this. |

| Fantastic work! |

+---------------------+**.**

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

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.**

mysql> select lower(reaction\_type) from reaction;

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

| lower(reaction\_type) |

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

| like |

| love |

| angry |

| wow |

| sad |

| love |

| like |

| like |

| angry |

| sad |

| like |

| wow |

| love |

| like |

| angry |

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

**LENGTH()**  
**W** **mysql> select length(user\_name),length(user\_name) as len from reaction where length(user\_name)>6;**

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

| length(user\_name) | len |

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

| 7 | 7 |

| 9 | 9 |

+-------------------+------+rite a query to find all users whose usernames are longer than 6 characters.

**NOW()ite a query to find all reactions made today (based on the created\_at column).**

Wr mysql> select now() from reaction;

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

| now() |

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

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

| 2025-08-07 11:41:14 |

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

**DATE()**  
**Write a query to extract and show only the date part from the created\_at column.**

mysql> select curdate() from reaction;

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

| curdate() |

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

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

| 2025-08-07 |

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

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

mysql> select user\_name from reaction where user\_name like '%an%';

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

| user\_name |

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

| Diana |

| Ethan |

| Hannah |

| Ian |

| Jane |

| Anita |

| Brian |

| Daniel |

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

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

mysql> select round(mood\_level,5) from reaction;

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

| round(mood\_level,5) |

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

| 8 |

| 9 |

| 3 |

| 7 |

| 2 |

| 6 |

| 5 |

| 9 |

| 4 |

| 1 |

| 8 |

| 7 |

| 10 |

| 6 |

| 2 |

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

**SUBSTR + UPPER()**  
**Write a query to show the first 2 letters of each user's name in uppercase.**

mysql> select substr(upper(user\_name),1,2) from reaction;

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

| substr(upper(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 10, 20, or 30.**

mysql> select post\_id from reaction where post\_id not in ('10','20','30');

+---------+

| post\_id |

+---------+

| 101 |

| 102 |

| 101 |

| 103 |

| 104 |

| 102 |

| 105 |

| 106 |

| 107 |

| 108 |

| 101 |

| 109 |

| 110 |

| 111 |

| 112 |

+---------+

**COUNT and IS NULL**  
**Write a query to count how many reactions have no comment.**

mysql> select count(\*) as no\_comment\_count from reaction where comment is null;

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

| no\_comment\_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');

mysql> select \* from orders;

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

| 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 |

| 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 |

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

**LIKE**

1. **Write a query to find all customers whose names end with 'a'. select customer name from orders where customer name like ‘%a’;**

mysql> select \* from orders where customer\_name like '%a';

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

| 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 |

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

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

1. **Write a query to find orders where the product name contains the word 'Phone'. Select product name from orders where product name like ’phone’;**

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 |

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

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

**Select product name from orders where product name like ‘\_\_\_\_\_’;**

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)**

1. **Write a query to show customer name as Buyer and price as Unit\_Price.**

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 |

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

1. **Write a query to show order ID and total amount (quantity × price) as Total\_Cost.**

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**

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

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 |

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

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

mysql> select \* from orders where 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**

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

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 |

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

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

mysql> select\* from orders where customer\_name like 'alice' and status like '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**

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

mysql> select \* from orders where status like 'pending' or 'processing';

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

| 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 |

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

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

mysql> select \* from orders where customer\_name like 'alice' or '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 |

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

**IN**

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

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 |

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

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

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 |

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

**IS NULL / IS NOT NULL**

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

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 |

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

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

mysql> select \* from orders where shipping\_address != '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**

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

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 |

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

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

mysql> select upper(customer\_name) from orders;

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

| upper(customer\_name) |

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

| ALICE |

| BOB |

| CHARLIE |

| DIANA |

| ETHAN |

| FIONA |

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

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

mysql> select lower(customer\_name) from orders;

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

| lower(customer\_name) |

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

| alice |

| bob |

| charlie |

| diana |

| ethan |

| fiona |

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

6 rows in set (0.02 sec)

1. **Write a query to return the first 3 letters of each product name with an alias Short\_Name.**

mysql> select substring(product\_name,1,3) as Short\_Name from orders;

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

| Short\_Name |

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

| Lap |

| Pho |

| Tab |

| Mon |

| Key |

| Mou |

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