1. **Which 5 cities placed most orders (ranked highest to lowest)**

Query:

```
SELECT addresses.city, COUNT(*) AS order_count
FROM orders
JOIN addresses ON orders.address_id = addresses.id
GROUP BY addresses.city
ORDER BY order_count DESC
LIMIT 5;
```

2. **Which 5 states placed most orders (ranked highest to lowest)**

Query:

```
SELECT addresses.state, COUNT(*) AS order_count
FROM orders
JOIN addresses ON orders.address_id = addresses.id
GROUP BY addresses.state
ORDER BY order_count DESC
LIMIT 5;
```

3. **What is the split between cash on delivery and prepaid**

Query:

```
SELECT
  CASE payment_mode
    WHEN 0 THEN 'cash on delivery'
    WHEN 1 THEN 'prepaid'
  END AS payment_mode,
  COUNT(*) AS order_count
FROM orders
GROUP BY payment_mode;
```

4. **Which 10 products were most purchased (ranked highest to lowest)**

Query:

```
SELECT p.name AS product_name, SUM(o.quantity) AS total_quantity_purchased
FROM products p
JOIN order_items o ON p.id = o.product_id
GROUP BY p.name
ORDER BY total_quantity_purchased DESC
LIMIT 10;
```

5. **How much discount have we given in last N number of days**

Query:

```
SELECT SUM(c.discount) AS total_discount
FROM orders o
JOIN coupons c ON o.coupon_id = c.id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL N DAY);
```

6. **What is the revenue in last N number of days (revenue will be on the basis of selling price)**

Query:

```
SELECT SUM(o.quantity * p.selling_price) AS revenue
FROM orders o
JOIN products p ON o.product_id = p.id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL N DAY);
```

7. **If spend on marketing is assumed to be X rupees, how much profit / loss have we made in last N days**

Query:

```
SELECT
  (SUM(o.quantity * p.selling_price) - X) AS profit_or_loss
FROM orders o
JOIN products p ON o.product_id = p.id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL N DAY);
```

8. **What is our repeat rate in last N days**

Query:

```
SELECT
  (COUNT(DISTINCT r.user_id) / COUNT(DISTINCT o.user_id)) * 100 AS repeat_rate
FROM (
  SELECT user_id
  FROM orders
  WHERE order_date >= DATE_SUB(CURDATE(), INTERVAL N DAY)
  GROUP BY user_id
  HAVING COUNT(*) > 1
) r
JOIN orders o ON r.user_id = o.user_id
WHERE o.order_date >= DATE_SUB(CURDATE(), INTERVAL N DAY);
```