```sql
SELECT name FROM sys.databases;

CREATE DATABASE LaDeDB;

SELECT TOP 10 * FROM orders;
SELECT TOP 10 * FROM customers;

SELECT *,
       CASE
           WHEN order_delivered_customer_date > order_estimated_delivery_date THEN 1
           ELSE 0
       END AS is_late
FROM orders;



SELECT
    region,
    COUNT(*) AS total_orders,
    SUM(CASE WHEN is_failed = 1 THEN 1 ELSE 0 END) AS failed_orders,
    ROUND(100.0 * SUM(CASE WHEN is_failed = 1 THEN 1 ELSE 0 END) / COUNT(*), 2) AS failed_rate
FROM orders
GROUP BY region
ORDER BY failed_rate_percent DESC;


SELECT *
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';

SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_TYPE = 'BASE TABLE';

-- Average delivery time per region (proxy for possible delay/idle vehicle issues)
SELECT
    c.geolocation_state AS region,
    COUNT(*) AS deliveries,
    AVG(DATEDIFF(HOUR, o.order_purchase_timestamp, o.order_delivered_customer_date)) AS avg_de
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
WHERE o.order_delivered_customer_date IS NOT NULL
GROUP BY c.geolocation_state
ORDER BY avg_delivery_hours DESC;


-- Top 10 sellers (couriers) with most completed deliveries
SELECT
    oi.seller_id,
    COUNT(DISTINCT oi.order_id) AS total_orders
FROM order_items oi
JOIN orders o ON oi.order_id = o.order_id
WHERE o.order_delivered_customer_date IS NOT NULL
```

```
GROUP BY oi.seller_id
ORDER BY total_orders DESC
OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

-- Order volume by day of week
SELECT
    DATENAME(WEEKDAY, order_purchase_timestamp) AS day_of_week,
    COUNT(order_id) AS total_orders,
    DATEPART(WEEKDAY, order_purchase_timestamp) AS weekday_number
FROM orders
GROUP BY
    DATENAME(WEEKDAY, order_purchase_timestamp),
    DATEPART(WEEKDAY, order_purchase_timestamp)
ORDER BY
    weekday_number;


 SELECT
    c.geolocation_state + '-' + c.geolocation_city AS route_cluster,
    COUNT(*) AS total_orders,
    ROUND(AVG(DATEDIFF(HOUR, o.order_purchase_timestamp, o.order_delivered_customer_date)), 2)
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id
WHERE o.order_delivered_customer_date IS NOT NULL
GROUP BY c.geolocation_state + '-' + c.geolocation_city
ORDER BY avg_delivery_hours DESC;
```