```
SHOW TABLES;
SELECT * FROM categories LIMIT 10;
-- ========================================
-- NORTHWIND SQL ANALYTICS - 15 QUERIES
-- ========================================

-- Query 1: Overall Business Metrics
SELECT
    COUNT(DISTINCT o.orderID) AS total_orders,
    COUNT(DISTINCT o.customerID) AS total_customers,
    COUNT(DISTINCT p.productID) AS total_products,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS total_revenue
FROM orders o
JOIN order_details od ON o.orderID = od.orderID
JOIN products p ON od.productID = p.productID;

-- ========================================

-- Query 2: Revenue by Category
SELECT
    cat.categoryName,
    COUNT(DISTINCT od.orderID) AS order_count,
    SUM(od.quantity) AS units_sold,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue
FROM categories cat
JOIN products p ON cat.categoryID = p.categoryID
JOIN order_details od ON p.productID = od.productID
GROUP BY cat.categoryName
ORDER BY revenue DESC;

-- ========================================
```

```sql
-- Query 3: Top 10 Customers by Revenue
SELECT
    c.customerID,
    c.companyName,
    c.country,
    COUNT(DISTINCT o.orderID) AS total_orders,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS total_revenue
FROM customers c
JOIN orders o ON c.customerID = o.customerID
JOIN order_details od ON o.orderID = od.orderID
GROUP BY c.customerID, c.companyName, c.country
ORDER BY total_revenue DESC
LIMIT 10;

-- ========================================

-- Query 4: Monthly Revenue Trend
SELECT
    YEAR(o.orderDate) AS year,
    MONTH(o.orderDate) AS month,
    COUNT(DISTINCT o.orderID) AS orders,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue
FROM orders o
JOIN order_details od ON o.orderID = od.orderID
GROUP BY YEAR(o.orderDate), MONTH(o.orderDate)
ORDER BY year, month;

-- ========================================

-- Query 5: Product Performance Analysis
```

```
SELECT
    p.productID,
    p.productName,
    cat.categoryName,
    COUNT(DISTINCT od.orderID) AS times_ordered,
    SUM(od.quantity) AS total_units_sold,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue
FROM products p
JOIN categories cat ON p.categoryID = cat.categoryID
JOIN order_details od ON p.productID = od.productID
GROUP BY p.productID, p.productName, cat.categoryName
ORDER BY revenue DESC;

-- ========================================

-- Query 6: Sales by Country
SELECT
    c.country,
    COUNT(DISTINCT c.customerID) AS customer_count,
    COUNT(DISTINCT o.orderID) AS order_count,
    ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue
FROM customers c
JOIN orders o ON c.customerID = o.customerID
JOIN order_details od ON o.orderID = od.orderID
GROUP BY c.country
ORDER BY revenue DESC;

-- ========================================

-- Query 7: Discount Impact Analysis
SELECT
```

```
    CASE
        WHEN discount = 0 THEN 'No Discount'
        WHEN discount <= 0.05 THEN '1-5%'
        WHEN discount <= 0.10 THEN '6-10%'
        WHEN discount <= 0.15 THEN '11-15%'
        ELSE '15%+'
    END AS discount_range,
    COUNT(*) AS line_items,
    SUM(quantity) AS units_sold,
    ROUND(SUM(unitPrice * quantity * (1 - discount)), 2) AS net_revenue
FROM order_details
GROUP BY
    CASE
        WHEN discount = 0 THEN 'No Discount'
        WHEN discount <= 0.05 THEN '1-5%'
        WHEN discount <= 0.10 THEN '6-10%'
        WHEN discount <= 0.15 THEN '11-15%'
        ELSE '15%+'
    END
ORDER BY discount_range;

-- ========================================

-- Query 8: Running Total Revenue by Month (WINDOW FUNCTION!)
WITH monthly_revenue AS (
    SELECT
        DATE_TRUNC('month', o.orderDate) AS month,
        ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS monthly_revenue
    FROM orders o
    JOIN order_details od ON o.orderID = od.orderID
    GROUP BY DATE_TRUNC('month', o.orderDate)
```

```
)
SELECT
    month,
    monthly_revenue,
    ROUND(SUM(monthly_revenue) OVER (ORDER BY month), 2) AS running_total
FROM monthly_revenue
ORDER BY month;

-- ======================================

-- Query 9: Top 3 Products per Category (WINDOW FUNCTION!)
WITH product_revenue AS (
    SELECT
        cat.categoryName,
        p.productName,
        ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue,
        ROW_NUMBER() OVER (
            PARTITION BY cat.categoryName
            ORDER BY SUM(od.unitPrice * od.quantity * (1 - od.discount)) DESC
        ) AS rank
    FROM categories cat
    JOIN products p ON cat.categoryID = p.categoryID
    JOIN order_details od ON p.productID = od.productID
    GROUP BY cat.categoryName, p.productName
)
SELECT
    categoryName,
    productName,
    revenue,
    rank
FROM product_revenue
```

```
WHERE rank <= 3
ORDER BY categoryName, rank;

-- ========================================

-- Query 10: Customer Lifetime Value (CTE!)
WITH customer_revenue AS (
    SELECT
        c.customerID,
        c.companyName,
        c.country,
        COUNT(DISTINCT o.orderID) AS total_orders,
        ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS lifetime_value
    FROM customers c
    JOIN orders o ON c.customerID = o.customerID
    JOIN order_details od ON o.orderID = od.orderID
    GROUP BY c.customerID, c.companyName, c.country
),
total AS (
    SELECT SUM(lifetime_value) AS total_revenue FROM customer_revenue
)
SELECT
    customerID,
    companyName,
    country,
    total_orders,
    lifetime_value,
    ROUND(lifetime_value * 100.0 / (SELECT total_revenue FROM total), 2) AS pct_of_total
FROM customer_revenue
ORDER BY lifetime_value DESC
LIMIT 20;
```

```
-- ======================================

-- Query 11: Year-over-Year Growth (LAG!)
WITH yearly_revenue AS (
    SELECT
        YEAR(o.orderDate) AS year,
        ROUND(SUM(od.unitPrice * od.quantity * (1 - od.discount)), 2) AS revenue
    FROM orders o
    JOIN order_details od ON o.orderID = od.orderID
    GROUP BY YEAR(o.orderDate)
)
SELECT
    year,
    revenue,
    LAG(revenue) OVER (ORDER BY year) AS prev_year_revenue,
    ROUND(revenue - LAG(revenue) OVER (ORDER BY year), 2) AS revenue_change,
    ROUND(
        (revenue - LAG(revenue) OVER (ORDER BY year)) * 100.0 /
        NULLIF(LAG(revenue) OVER (ORDER BY year), 0),
        2
    ) AS growth_pct
FROM yearly_revenue
ORDER BY year;

-- ======================================

-- Query 12: Customer Cohort Retention
WITH first_order AS (
    SELECT
        customerID,
```

```
    DATE_TRUNC('month', MIN(orderDate)) AS cohort_month
  FROM orders
  GROUP BY customerID
)
SELECT
  fo.cohort_month,
  COUNT(DISTINCT fo.customerID) AS cohort_size,
  COUNT(DISTINCT CASE WHEN MONTHS_BETWEEN(o.orderDate, fo.cohort_month) >= 1 THEN
o.customerID END) AS month_1,
  COUNT(DISTINCT CASE WHEN MONTHS_BETWEEN(o.orderDate, fo.cohort_month) >= 3 THEN
o.customerID END) AS month_3
FROM first_order fo
LEFT JOIN orders o ON fo.customerID = o.customerID
GROUP BY fo.cohort_month
ORDER BY fo.cohort_month;

-- ========================================

-- Query 13: Shipping Performance
SELECT
  s.companyName AS shipper,
  COUNT(o.orderID) AS shipments,
  ROUND(AVG(DATEDIFF(o.shippedDate, o.orderDate)), 2) AS avg_days_to_ship,
  ROUND(AVG(o.freight), 2) AS avg_freight_cost,
  SUM(CASE WHEN o.shippedDate > o.requiredDate THEN 1 ELSE 0 END) AS late_shipments
FROM shippers s
JOIN orders o ON s.shipperID = o.shipperID
WHERE o.shippedDate IS NOT NULL
GROUP BY s.companyName
ORDER BY shipments DESC;

-- ========================================
```

```
-- Query 14: Market Basket Analysis
SELECT
    p1.productName AS product_a,
    p2.productName AS product_b,
    COUNT(DISTINCT od1.orderID) AS times_together
FROM order_details od1
JOIN order_details od2
    ON od1.orderID = od2.orderID
    AND od1.productID < od2.productID
JOIN products p1 ON od1.productID = p1.productID
JOIN products p2 ON od2.productID = p2.productID
GROUP BY p1.productName, p2.productName
HAVING COUNT(DISTINCT od1.orderID) >= 5
ORDER BY times_together DESC
LIMIT 20;

-- ========================================

-- Query 15: Created Analytics View for Power BI
CREATE OR REPLACE VIEW analytics AS
SELECT
    o.orderID,
    o.orderDate,
    YEAR(o.orderDate) AS order_year,
    MONTH(o.orderDate) AS order_month,
    o.customerID,
    c.companyName,
    c.country,
    c.city,
    od.productID,
```

```
    p.productName,
    cat.categoryName,
    od.quantity,
    od.unitPrice,
    od.discount,
    ROUND(od.unitPrice * od.quantity, 2) AS line_total_gross,
    ROUND(od.unitPrice * od.quantity * (1 - od.discount), 2) AS line_total_net,
    o.freight,
    DATEDIFF(o.shippedDate, o.orderDate) AS days_to_ship
FROM orders o
JOIN customers c ON o.customerID = c.customerID
JOIN order_details od ON o.orderID = od.orderID
JOIN products p ON od.productID = p.productID
JOIN categories cat ON p.categoryID = cat.categoryID;

-- Query the view
SELECT * FROM analytics ORDER BY orderDate DESC LIMIT 20;

-- ========================================
-- END OF QUERIES
```