```sql
-- Find missing values in key columns
SELECT * FROM Customers WHERE name IS NULL OR email IS NULL;
SELECT * FROM Products WHERE name IS NULL OR price IS NULL;
SELECT * FROM Sales WHERE total_price IS NULL;

-- Find duplicate customers (same email)
SELECT email, COUNT(*)
FROM Customers
GROUP BY email
HAVING COUNT(*) > 1;


--Total Sales Revenue
SELECT SUM(total_price) AS total_revenue FROM Sales;


--Best-Selling Products
SELECT
P.name AS product_name,
P.category,
SUM(S.quantity) AS total_units_sold,
SUM(S.total_price) AS total_revenue
FROM Sales S
JOIN Products P ON S.product_id = P.product_id
GROUP BY P.name, P.category
ORDER BY total_units_sold DESC LIMIT 10;



--Top Customers by Spending
SELECT
C.name AS customer_name,
C.location,
SUM(S.total_price) AS total_spent
FROM Sales S
JOIN Customers C ON S.customer_id = C.customer_id
GROUP BY C.name, C.location
ORDER BY total_spent DESC
LIMIT 10;


--Customer Purchase Frequency
SELECT
C.name AS customer_name,
COUNT(S.sale_id) AS purchase_count
FROM Sales S
JOIN Customers C ON S.customer_id = C.customer_id
GROUP BY C.name
ORDER BY purchase_count DESC
LIMIT 10;


--Top Sales Employees
SELECT
E.name AS employee_name, E.role,
COUNT(S.sale_id) AS total_sales,
SUM(S.total_price) AS total_revenue_generated
FROM Sales S
JOIN Employees E ON S.employee_id = E.employee_id
GROUP BY E.name, E.role
ORDER BY total_revenue_generated DESC
LIMIT 5;


--Employee Salary vs. Sales Performance
SELECT E.name AS employee_name, E.salary,
COUNT(S.sale_id) AS sales_count,
SUM(S.total_price) AS revenue_generated
FROM Employees E
LEFT JOIN Sales S ON E.employee_id = S.employee_id
GROUP BY E.name, E.salary
ORDER BY revenue_generated DESC;
```

```sql

--Top-Performing Stores
SELECT ST.name AS store_name, ST.location,
SUM(S.total_price) AS total_sales
FROM Sales S
JOIN Stores ST ON S.store_id = ST.store_id
GROUP BY ST.name, ST.location
ORDER BY total_sales DESC
LIMIT 5;


--Store Revenue Contribution
SELECT ST.name AS store_name, ST.revenue,
SUM(S.total_price) AS total_sales,
(SUM(S.total_price) / (SELECT SUM(total_price) FROM Sales)) * 100 AS sales_percentage
FROM Sales S
JOIN Stores ST ON S.store_id = ST.store_id
GROUP BY ST.name, ST.revenue
ORDER BY sales_percentage DESC;


--Low Stock Products
SELECT
name AS product_name, stock_quantity
FROM Products
WHERE stock_quantity < 10
ORDER BY stock_quantity ASC;


--Product Turnover Rate (Most Frequently Sold Items)
SELECT P.name AS product_name,
P.category,
SUM(S.quantity) AS total_sold
FROM Sales S
JOIN Products P ON S.product_id = P.product_id
GROUP BY P.name, P.category
ORDER BY total_sold DESC
LIMIT 10;

```