```
-- ====================================================
-- SETUP: Create Demo Tables
-- ====================================================

CREATE TABLE Sales (
   Date DATE,
   Order_id INT,
   Item_id INT,
   Customer_id INT,
   Quantity INT,
   Revenue DECIMAL(10,2)
);

CREATE TABLE Items (
   Item_id INT,
   Item_name VARCHAR(50),
   Price DECIMAL(10,2),
   Department VARCHAR(50)
);

CREATE TABLE Customers (
   Customer_id INT,
   First_name VARCHAR(50),
   Last_name VARCHAR(50),
   Country VARCHAR(50),
   City VARCHAR(50)
);

-- ====================================================
-- INSERT SAMPLE DATA
-- ====================================================
```

```sql
INSERT INTO Items VALUES
(1, 'T-shirt', 20.00, 'Clothing'),
(2, 'Jeans', 40.00, 'Clothing'),
(3, 'Laptop', 700.00, 'Electronics'),
(4, 'Mouse', 25.00, 'Electronics');

INSERT INTO Customers VALUES
(101, 'John', 'Doe', 'USA', 'New York'),
(102, 'Alice', 'Smith', 'Germany', 'Berlin'),
(103, 'Raj', 'Kumar', 'India', 'Delhi'),
(104, 'Maria', 'Lopez', 'Spain', 'Madrid');

INSERT INTO Sales VALUES
('2023-01-05', 1001, 1, 101, 2, 40.00),
('2023-01-05', 1002, 2, 102, 1, 40.00),
('2023-01-10', 1003, 3, 103, 1, 700.00),
('2023-01-12', 1004, 4, 104, 3, 75.00),
('2023-03-18', 1005, 1, 101, 1, 20.00),
('2023-03-18', 1006, 2, 101, 2, 80.00);

-- =======================================================
-- SELECT Basics
-- =======================================================
SELECT * FROM Sales LIMIT 5;

-- =======================================================
-- WHERE Filtering
-- =======================================================
SELECT * FROM Sales
WHERE Date = '2023-01-05';
```

```
-- ==================================================
-- AND, OR, NOT Operators
-- ==================================================
SELECT * FROM Customers
WHERE Country = 'Germany' AND City = 'Berlin';

SELECT * FROM Customers
WHERE NOT Country = 'Spain';

SELECT * FROM Customers
WHERE Country = 'Spain' AND First_name LIKE 'M%' OR First_name LIKE 'J%';

-- ==================================================
-- Aggregate Functions
-- ==================================================
SELECT COUNT(*) AS total_sales,
    SUM(Revenue) AS total_revenue,
    AVG(Revenue) AS avg_revenue,
    MIN(Revenue) AS min_rev,
    MAX(Revenue) AS max_rev
FROM Sales;

-- ==================================================
-- GROUP BY
-- ==================================================
SELECT Date, SUM(Revenue) AS total_rev, COUNT(DISTINCT Order_id) AS num_sales
FROM Sales
WHERE Date BETWEEN '2023-01-01' AND '2023-01-31'
GROUP BY Date;
```

```
-- =======================================================
-- ORDER BY
-- =======================================================
SELECT Department, COUNT(Item_id) AS num_items
FROM Items
GROUP BY Department
ORDER BY num_items DESC, Department DESC;

-- =======================================================
-- HAVING (Filter aggregates)
-- =======================================================
SELECT Order_id, SUM(Revenue) AS Rev
FROM Sales
GROUP BY Order_id
HAVING Rev >= 100
ORDER BY Rev DESC;

-- =======================================================
-- SQL Column Functions
-- =======================================================
SELECT SUM(Revenue) / COUNT(DISTINCT Order_id) AS avg_order_val
FROM Sales
WHERE CAST(Date AS CHAR) LIKE '2023-%';

-- =======================================================
-- JOINS
-- =======================================================
SELECT i.Item_id, i.Item_name, SUM(s.Revenue) AS total_rev
FROM Items AS i
LEFT JOIN Sales AS s
ON i.Item_id = s.Item_id
```

```
GROUP BY i.Item_id, i.Item_name;


-- =====================================================
-- Subquery
-- =====================================================
SELECT Order_id, SUM(Revenue) AS Rev
FROM Sales
GROUP BY Order_id
HAVING Rev > (
   SELECT SUM(Revenue)
   FROM Sales
   WHERE Order_id = 1005
);
```