SQL PRACTICE SCRIPT - SalesPerson and Sales Example (PostgreSQL)

```
-- ============================================================
-- SQL PRACTICE SCRIPT - SalesPerson and Sales Example (PostgreSQL)
-- ============================================================

-- PART 1: CREATE DATABASE AND TABLES

-- (Run separately)
-- CREATE DATABASE sales_demo;
-- \c sales_demo

DROP TABLE IF EXISTS Sales;
DROP TABLE IF EXISTS SalesPerson;

CREATE TABLE SalesPerson (
    salesperson_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    region VARCHAR(50),
    hire_date DATE,
    salary NUMERIC(10,2)
);

CREATE TABLE Sales (
    sale_id SERIAL PRIMARY KEY,
    salesperson_id INT REFERENCES SalesPerson(salesperson_id),
    sale_date DATE,
    amount NUMERIC(10,2)
);

-- Insert sample data
INSERT INTO SalesPerson (name, region, hire_date, salary) VALUES
('Alice Johnson', 'North', '2020-02-15', 4500.00),
('Bob Smith', 'South', '2019-06-20', 5000.00),
('Charlie Brown', 'East', '2021-01-10', 4000.00),
('Diana Miller', 'West', '2022-09-01', 4200.00);

INSERT INTO Sales (salesperson_id, sale_date, amount) VALUES
(1, '2024-01-10', 1200.00),
(1, '2024-02-12', 1800.00),
(2, '2024-02-15', 2200.00),
(2, '2024-03-01', 2500.00),
(3, '2024-03-05', 3000.00),
(4, '2024-04-10', 2750.00);

-- Simple SELECT, ORDER, and Pagination
SELECT * FROM Sales ORDER BY sale_date DESC FETCH FIRST 3 ROWS ONLY;


-- ============================================================
-- PART 1.1: TEXT & NUMERIC FUNCTIONS
-- ============================================================

SELECT name, POSITION('a' IN name) AS first_a, SUBSTRING(name,1,5) AS first5,
       LENGTH(name) AS len, UPPER(region), LOWER(region), INITCAP(region)
FROM SalesPerson;

SELECT TO_CHAR(salary, 'FM$9999.00') AS formatted_salary FROM SalesPerson;
SELECT ROUND(amount/3,2) AS avg_per_item, TRUNC(amount, -2) AS rounded, MOD(amount, 7) AS mod_7 FROM Sal


-- ============================================================
-- PART 2: DATE FUNCTIONS
-- ============================================================

SELECT sale_date, DATE_PART('month', sale_date) AS month, DATE_TRUNC('month', sale_date) AS month_start,
-- ============================================================
```

```sql
-- PART 3: AGGREGATIONS
-- ============================================================

SELECT salesperson_id, SUM(amount) AS total_sales, AVG(amount) AS avg_sale, MIN(amount), MAX(amount)
FROM Sales
GROUP BY salesperson_id
HAVING SUM(amount) > 2000;


-- ============================================================
-- PART 4: JOINS
-- ============================================================

-- INNER JOIN
SELECT sp.name, sp.region, s.sale_date, s.amount
FROM SalesPerson sp
JOIN Sales s ON sp.salesperson_id = s.salesperson_id;

-- NON-EQUI JOIN Example: Compare salesperson salary vs sale amount
SELECT sp.name, s.amount, sp.salary
FROM SalesPerson sp
JOIN Sales s ON s.amount BETWEEN sp.salary * 0.2 AND sp.salary * 0.6;

-- SELF JOIN Example: Compare two salespeople in same region
SELECT a.name AS person_a, b.name AS person_b, a.region
FROM SalesPerson a
JOIN SalesPerson b ON a.region = b.region AND a.salesperson_id <> b.salesperson_id;


-- ============================================================
-- PART 4.1: OUTER JOINS
-- ============================================================

-- LEFT JOIN
SELECT sp.name, s.sale_date, s.amount
FROM SalesPerson sp
LEFT JOIN Sales s ON sp.salesperson_id = s.salesperson_id;

-- RIGHT JOIN
SELECT sp.name, s.sale_date, s.amount
FROM SalesPerson sp
RIGHT JOIN Sales s ON sp.salesperson_id = s.salesperson_id;

-- FULL JOIN
SELECT sp.name, s.sale_date, s.amount
FROM SalesPerson sp
FULL JOIN Sales s ON sp.salesperson_id = s.salesperson_id;
```