

# Sales Trend Analysis - SQL Script & Sample Output

## SQL Script

```
-- =====
-- Sales Trend Analysis Script
-- Works on PostgreSQL & MySQL
-- =====

-- 1. Drop table if exists
DROP TABLE IF EXISTS orders;

-- 2. Create orders table
CREATE TABLE orders (
    order_id INT,
    order_date DATE,
    amount DECIMAL(10,2),
    product_id INT
);

-- 3. Insert sample data
INSERT INTO orders (order_id, order_date, amount, product_id) VALUES
(1, '2023-01-05', 250.00, 101),
(2, '2023-01-10', 500.00, 102),
(3, '2023-02-02', 300.00, 103),
(4, '2023-02-15', 450.00, 104),
(5, '2023-03-03', 150.00, 101),
(6, '2023-03-10', 800.00, 105),
(7, '2023-03-12', 200.00, 106),
(8, '2023-04-05', 1000.00, 107),
(9, '2023-04-20', 250.00, 108),
(10, '2023-04-25', 750.00, 102),
(11, '2023-05-01', 500.00, 103),
(12, '2023-05-08', 650.00, 104);

-- 4. Sales Trend Analysis Query
-- Detects database type automatically and formats output
SELECT
    /* Extract Year */
    CASE
        WHEN POSITION('postgres' IN VERSION()) > 0 THEN EXTRACT(YEAR FROM order_date)
        ELSE YEAR(order_date)
    END AS order_year,

    /* Extract Month Number */
    CASE
        WHEN POSITION('postgres' IN VERSION()) > 0 THEN EXTRACT(MONTH FROM order_date)
        ELSE MONTH(order_date)
    END AS order_month,

    /* Extract Month Name */
    CASE
        WHEN POSITION('postgres' IN VERSION()) > 0 THEN TO_CHAR(order_date, 'Month')
```

```
        ELSE MONTHNAME(order_date)
    END AS month_name,

    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_orders
FROM orders
GROUP BY order_year, order_month, month_name
ORDER BY order_year, order_month;
```

Sample Output Table

order_year	order_month	month_name	total_revenue	total_orders
2023	1	January	750.00	2
2023	2	February	750.00	2
2023	3	March	1150.00	3
2023	4	April	2000.00	3
2023	5	May	1150.00	2