```sql
-- Create the table
CREATE TABLE sales (
  sale_id INT PRIMARY KEY,
  sale_date DATE,
  product_id INT,
  customer_id INT,
  quantity INT,
  unit_price DECIMAL(10,2),
  total_amount DECIMAL(10,2)
);

-- Insert sample data
INSERT INTO sales (sale_id, sale_date, product_id, customer_id, quantity, unit_price, total_amount)
VALUES
(1, '2024-01-05', 101, 1, 2, 10.00, 20.00),
(2, '2024-01-15', 102, 2, 1, 15.00, 15.00),
(3, '2024-02-03', 101, 3, 3, 10.00, 30.00),
(4, '2024-02-20', 103, 4, 5, 8.00, 40.00),
(5, '2024-03-12', 101, 1, 1, 10.00, 10.00),
(6, '2024-03-22', 104, 2, 2, 12.00, 24.00),
(7, '2024-04-02', 102, 3, 4, 15.00, 60.00),
(8, '2024-04-18', 103, 4, 2, 8.00, 16.00);

-- Query to retrieve data
SELECT * FROM sales;
```

STDIN

Input for the program (Optional)

Output:

sale_id	sale_date	product_id	customer_id	quantity	unit_price
1	2024-01-05	101	1	2	10.00
2	2024-01-15	102	2	1	15.00
3	2024-02-03	101	3	3	10.00
4	2024-02-20	103	4	5	8.00
5	2024-03-12	101	1	1	10.00
6	2024-03-22	104	2	2	12.00
7	2024-04-02	102	3	4	15.00
8	2024-04-18	103	4	2	8.00

queries.sql

43grjwerv

AI NEW SQLSERVER RUN

```sql

-- Insert sample data
INSERT INTO sales (sale_id, sale_date, product_id,
customer_id, quantity, unit_price, total_amount)
VALUES
(1, '2024-01-05', 101, 1, 2, 10.00, 20.00),
(2, '2024-01-15', 102, 2, 1, 15.00, 15.00),
(3, '2024-02-03', 101, 3, 3, 10.00, 30.00),
(4, '2024-02-20', 103, 4, 5, 8.00, 40.00),
(5, '2024-03-12', 101, 1, 1, 10.00, 10.00),
(6, '2024-03-22', 104, 2, 2, 12.00, 24.00),
(7, '2024-04-02', 102, 3, 4, 15.00, 60.00),
(8, '2024-04-18', 103, 4, 2, 8.00, 16.00);

-- Query to retrieve data
SELECT * FROM sales;
SELECT
  FORMAT(sale_date, 'yyyy-MM') AS sale_month,
  SUM(total_amount) AS total_sales
FROM sales
GROUP BY FORMAT(sale_date, 'yyyy-MM')
ORDER BY sale_month;
WITH monthly_data AS (
  SELECT
    FORMAT(sale_date, 'yyyy-MM') AS sale_month,
    SUM(total_amount) AS total_sales
  FROM sales
  GROUP BY FORMAT(sale_date, 'yyyy-MM')
)
SELECT
  sale_month,
  total_sales,
  LAG(total_sales) OVER (ORDER BY sale_month) AS prev_month_sales,
  ROUND(
    (total_sales - LAG(total_sales) OVER (ORDER BY sale_month)) * 100.0 /
    NULLIF(LAG(total_sales) OVER (ORDER BY sale_month), 0), 2
  ) AS percent_change
FROM monthly_data;

```

STDIN

Input for the program (Optional)

Output:

```
sale_id     sale_date        product_id  customer_id quantity    unit_price   total_amount
----------- ---------------- ----------- ----------- ----------- ------------ ------------
          1       2024-01-05         101           1           2        10.00        20.00
          2       2024-01-15         102           2           1        15.00        15.00
          3       2024-02-03         101           3           3        10.00        30.00
          4       2024-02-20         103           4           5         8.00        40.00
          5       2024-03-12         101           1           1        10.00        10.00
          6       2024-03-22         104           2           2        12.00        24.00
          7       2024-04-02         102           3           4        15.00        60.00
          8       2024-04-18         103           4           2         8.00        16.00
sale_month
----------------------------------------------------------------------------------------------------
2024-01
2024-02
2024-03
2024-04
sale_month
----------------------------------------------------------------------------------------------------
2024-01
2024-02
2024-03
2024-04
```