Sales Trend Analysis Using Aggregations (SQLite)

Introduction

This guide explains how to perform Sales Trend Analysis using SQL in SQLite. It's aimed at beginners who want to understand how to group data by time (months/years) and extract meaningful insights like total sales and order volumes from a table.

Step 1: Create the Table

We create a table called 'online_sales' with four columns:
- order_id: Unique order number
- order_date: Date when the order was placed
- amount: Sale amount
- product_id: ID of the product sold

```
CREATE TABLE online_sales (
    order_id INTEGER,
    order_date TEXT,
    amount REAL,
    product_id TEXT
);
```

Step 2: Insert Sample Data

We insert 10 rows of sample data to simulate online sales records from January to May 2023.

```
INSERT INTO online_sales (order_id, order_date, amount, product_id) VALUES
(1, '2023-01-15', 200.00, 'P001'),
(2, '2023-01-20', 200.00, 'P002'),
(3, '2023-02-10', 250.00, 'P001'),
(4, '2023-02-22', 230.00, 'P003'),
(5, '2023-03-05', 180.00, 'P004'),
(6, '2023-03-18', 200.00, 'P002'),
(7, '2023-03-30', 210.00, 'P001'),
(8, '2023-04-12', 250.00, 'P003'),
(9, '2023-04-25', 250.00, 'P004'),
(10, '2023-05-02', 400.00, 'P005');
```

Step 3: Analyze Monthly Trends with SQL

We use SQLite's strftime() function to extract year and month from the order_date. Then we group the data by these values and calculate:
- total_revenue using SUM(amount)
- total_order_volume using COUNT(DISTINCT order_id)
We also use WHERE to limit analysis to 2023.

```
SELECT
    strftime('%Y', order_date) AS order_year,
    strftime('%m', order_date) AS order_month,
```

```
    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_order_volume
FROM
    online_sales
WHERE
    order_date BETWEEN '2023-01-01' AND '2023-12-31'
GROUP BY
    order_year, order_month
ORDER BY
    order_year, order_month;
```

Step 4: Sample Output Explanation

This is how the result of the query looks. Each row shows the total revenue and number of orders for a month.

order_year	order_month	total_revenue	total_order_volume
2023	01	400.00	2
2023	02	480.00	2
2023	03	590.00	3
2023	04	500.00	2
2023	05	400.00	1

Conclusion

By following these steps, beginners can learn how to:
- Use SQL aggregation functions
- Group data by month and year
- Filter and sort time-series sales data

This kind of analysis is extremely useful in understanding business trends over time.