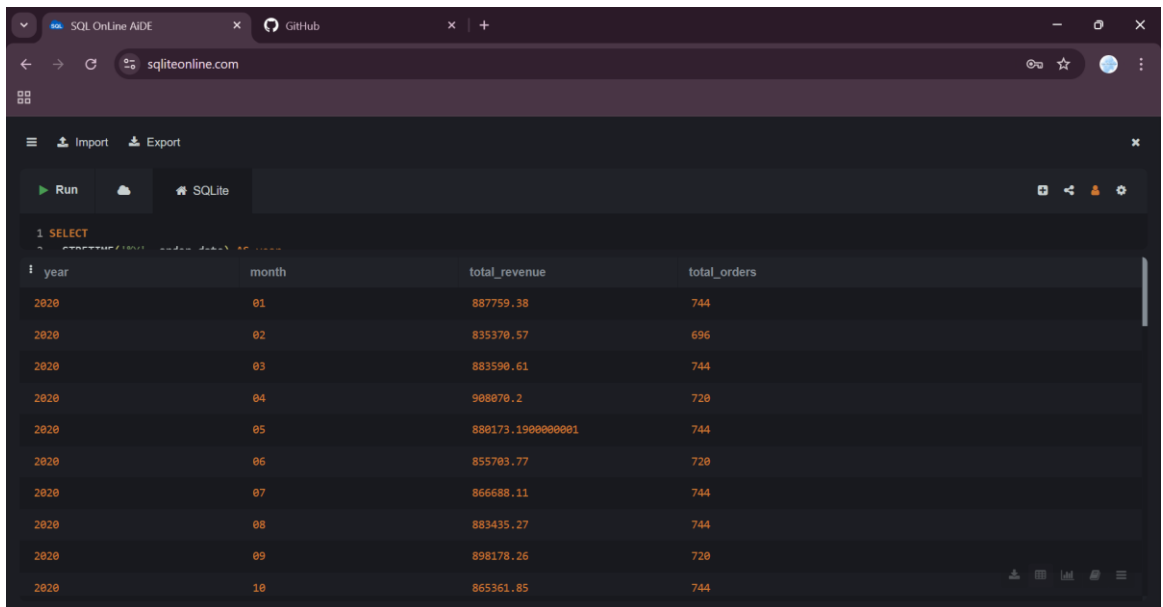


SQL SCRIPT+RESULT TABLE

MONTHLY REVENUE AND ORDER VOLUME:

```
SELECT  
  
    STRFTIME('%Y', order_date) AS year,  
  
    STRFTIME('%m', order_date) AS month,  
  
    SUM(amount) AS total_revenue,  
  
    COUNT(DISTINCT order_id) AS total_orders  
  
FROM  
  
    online_sales_data  
  
GROUP BY  
  
    year, month  
  
ORDER BY  
  
    year, month;
```

RESULT TABLE:



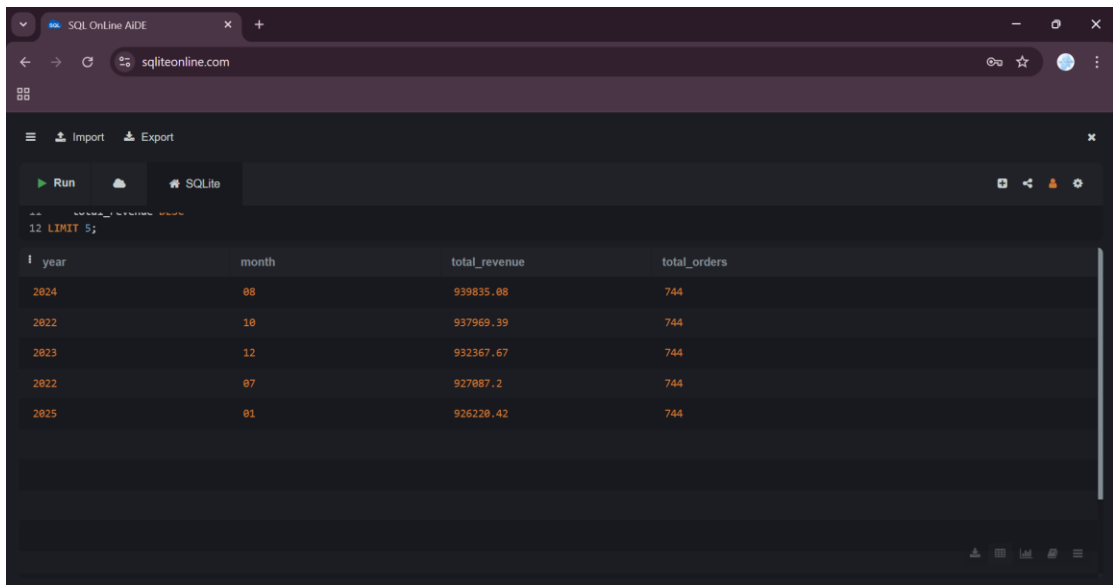
The screenshot shows a web browser window with the URL `sqliteonline.com`. The interface includes a menu, 'Import', 'Export', 'Run', and 'SQLite' buttons. The SQL query from the previous block is entered in the editor. Below the editor, the results are displayed in a table with 4 columns: year, month, total_revenue, and total_orders. The table contains 10 rows of data for the year 2020, one for each month.

year	month	total_revenue	total_orders
2020	01	887759.38	744
2020	02	835370.57	696
2020	03	883590.61	744
2020	04	908070.2	720
2020	05	880173.1900000001	744
2020	06	855703.77	720
2020	07	866688.11	744
2020	08	883435.27	744
2020	09	898178.26	720
2020	10	865361.85	744

TOP 5 REVENUE MONTHS:

```
SELECT  
  
    STRFTIME('%Y', order_date) AS year,  
  
    STRFTIME('%m', order_date) AS month,  
  
    SUM(amount) AS total_revenue,  
  
    COUNT(DISTINCT order_id) AS total_orders  
  
FROM  
  
    online_sales_data  
  
GROUP BY  
  
    year, month  
  
ORDER BY  
  
    total_revenue DESC  
  
LIMIT 5;
```

RESULT TABLE:



The screenshot shows a web browser window with the URL `sqliteonline.com`. The interface includes a 'Run' button and a 'SQLite' tab. The SQL query is entered in the editor, and the results are displayed in a table below. The table has four columns: 'year', 'month', 'total_revenue', and 'total_orders'. The results are sorted by 'total_revenue' in descending order, showing the top 5 months.

year	month	total_revenue	total_orders
2024	08	939835.08	744
2022	10	937969.39	744
2023	12	932367.67	744
2022	07	927087.2	744
2025	01	926220.42	744