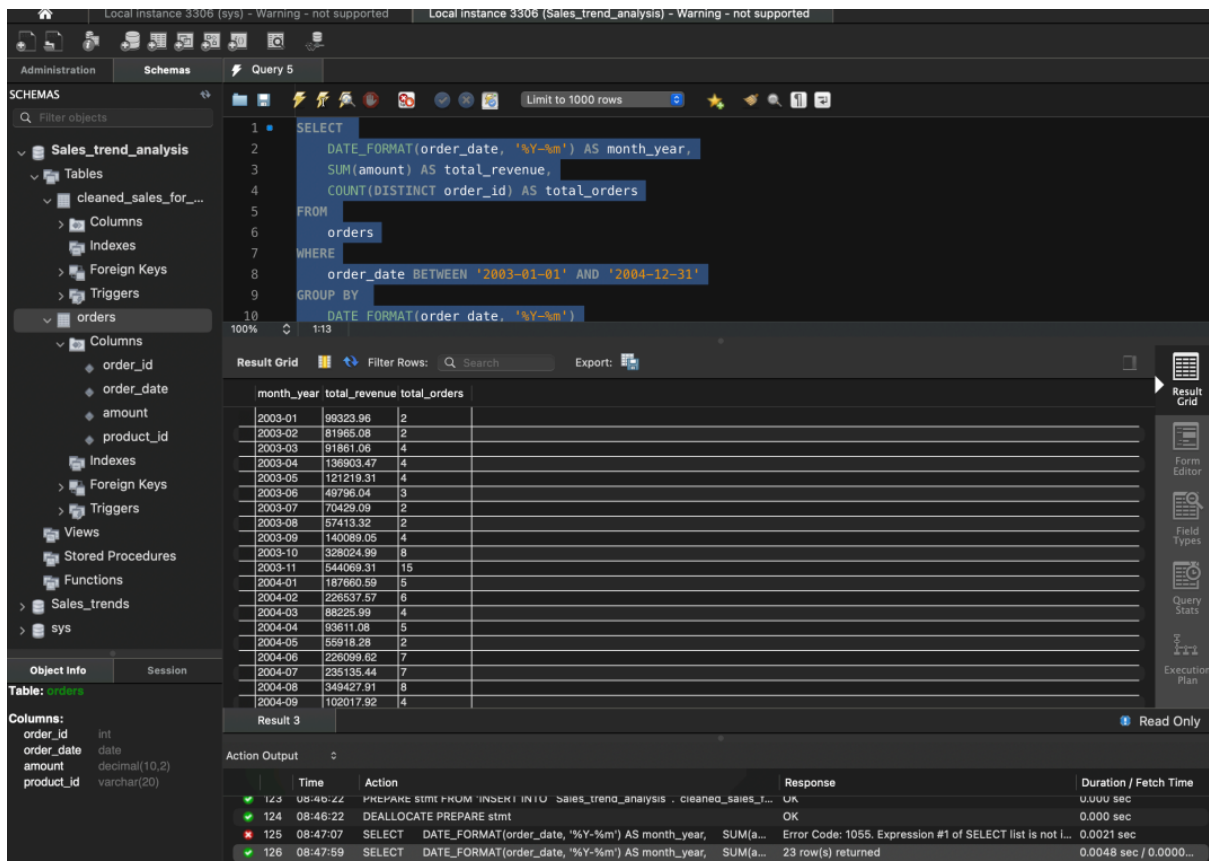


SQL Script:-

```
SELECT
    DATE_FORMAT(order_date, '%Y-%m') AS month_year,
    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_orders
FROM
    orders
WHERE
    order_date BETWEEN '2003-01-01' AND '2004-12-31'
GROUP BY
    DATE_FORMAT(order_date, '%Y-%m')
ORDER BY
    DATE_FORMAT(order_date, '%Y-%m');
```

Output:-



The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'Sales_trend_analysis' database with the 'orders' table selected. The center pane shows the SQL query being executed. The right pane shows the 'Result Grid' with the following data:

month_year	total_revenue	total_orders
2003-01	99323.96	2
2003-02	81965.08	2
2003-03	91861.06	4
2003-04	136903.47	4
2003-05	121219.31	4
2003-06	49796.04	3
2003-07	70429.09	2
2003-08	57413.32	2
2003-09	140089.05	4
2003-10	328024.99	8
2003-11	544069.31	15
2004-01	187660.59	5
2004-02	226537.57	6
2004-03	88225.99	4
2004-04	93611.08	5
2004-05	55918.28	2
2004-06	226099.62	7
2004-07	235135.44	7
2004-08	349427.31	8
2004-09	102017.92	4

The bottom pane shows the 'Action Output' with the following details:

Time	Action	Response	Duration / Fetch Time
123 08:46:22	PREPARE stmt FROM 'INSERT INTO Sales_trend_analysis . cleaned_sales_t...	OK	0.000 sec
124 08:46:22	DEALLOCATE PREPARE stmt	OK	0.000 sec
125 08:47:07	SELECT DATE_FORMAT(order_date, '%Y-%m') AS month_year, SUM(a...	Error Code: 1055. Expression #1 of SELECT list is not l...	0.0021 sec
126 08:47:59	SELECT DATE_FORMAT(order_date, '%Y-%m') AS month_year, SUM(a...	23 row(s) returned	0.0048 sec / 0.0000...

Data Cleaning Steps (Python)

```
import pandas as pd
df = pd.read_csv('Superstore_sales_dataset.csv')
df.columns = df.columns.str.strip().str.lower().str.replace(' ', '_')

df['order_date'] = pd.to_datetime(df['order_date'], errors='coerce', dayfirst=False)
df['ship_date'] = pd.to_datetime(df['ship_date'], errors='coerce', dayfirst=False)

df['postal_code'].fillna(method='ffill', inplace=True) # forward fill
```

```
df.drop_duplicates(inplace=True)
df.reset_index(drop=True, inplace=True)
df.to_csv('superstore_sales_cleaned.csv', index=False)
print(" Data cleaned and saved as 'superstore_sales_cleaned.csv")
```