

TASK 6: SALES TREND ANALYSIS USING AGGREGATIONS

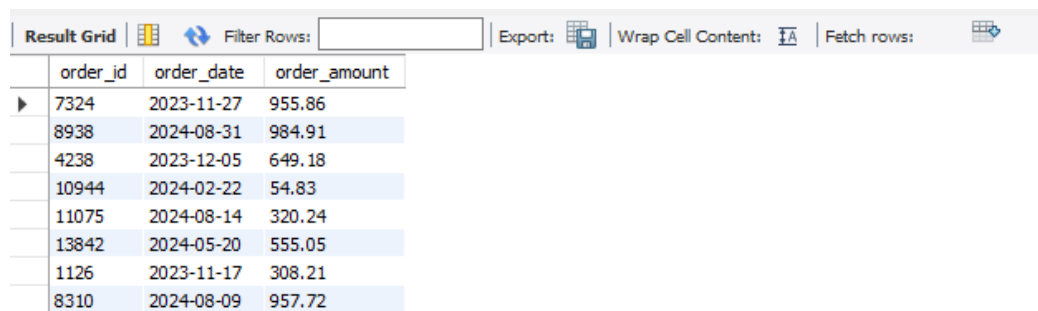
Goal:

- Use `EXTRACT(MONTH FROM order_date)` for month.
- `GROUP BY` year/month.
- Use `SUM()` for revenue.
- `COUNT(DISTINCT order_id)` for volume.
- Use `ORDER BY` for sorting.
- Limit results for specific time periods.

Query:

```
create database sales;  
create table orders (order_id int, order_date date, order_amount double(10,2));  
select * from orders;
```

Output Screenshot:



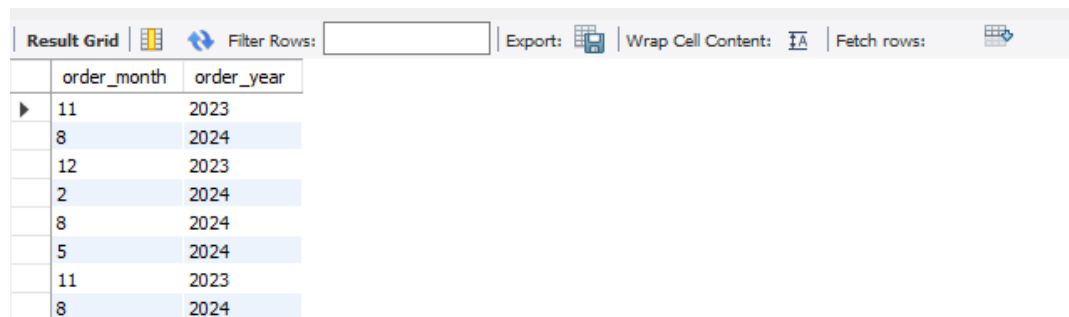
The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a table with three columns: 'order_id', 'order_date', and 'order_amount'. The table contains 8 rows of data.

order_id	order_date	order_amount
7324	2023-11-27	955.86
8938	2024-08-31	984.91
4238	2023-12-05	649.18
10944	2024-02-22	54.83
11075	2024-08-14	320.24
13842	2024-05-20	555.05
1126	2023-11-17	308.21
8310	2024-08-09	957.72

Query:

```
select month(order_date) as order_month, year(order_date) as order_year from orders;
```

Output Screenshot:



The screenshot shows a database interface with a toolbar at the top containing options like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a table with two columns: 'order_month' and 'order_year'. The table contains 8 rows of data.

order_month	order_year
11	2023
8	2024
12	2023
2	2024
8	2024
5	2024
11	2023
8	2024

Query:

```
select month(order_date) as order_month, year(order_date) as order_year,
sum(order_amount) as total_revenue, count(distinct order_id) as volume from orders where
extract(year from order_date) = 2024 group by order_year, order_month order by order_year,
order_month;
```

Output Screenshot:

	order_month	order_year	total_revenue	volume
▶	1	2024	3667447.62	1284
	2	2024	3300225.42	1173
	3	2024	3756172.48	1306
	4	2024	3493861.41	1213
	5	2024	3590164.64	1267
	6	2024	3355548.13	1190
	7	2024	3760470.93	1320
	8	2024	3737818.32	1308

Query:

```
select month(order_date) as order_month, year(order_date) as order_year,
sum(order_amount) as total_revenue, count(distinct order_id) as volume from orders group by
order_year, order_month order by total_revenue desc limit 3;
```

Output Screenshot:

	order_month	order_year	total_revenue	volume
▶	1	2024	3667447.62	1284
	2	2024	3300225.42	1173
	3	2024	3756172.48	1306
	4	2024	3493861.41	1213
	5	2024	3590164.64	1267
	6	2024	3355548.13	1190
	7	2024	3760470.93	1320
	8	2024	3737818.32	1308