

PIZZA SALES ANALYSIS

Using SQL



PIZZA



PIZZA SALES ANALYSIS USING SQL

**Data Insights from Pizza
Sales Dataset**

Presented by: Anjali

Date: 30/06/2025

OBJECTIVE

To explore and analyze pizza sales data using SQL queries to uncover business insights.



Tools Used:

- SQL (MySQL)
- Pizza Sales Dataset

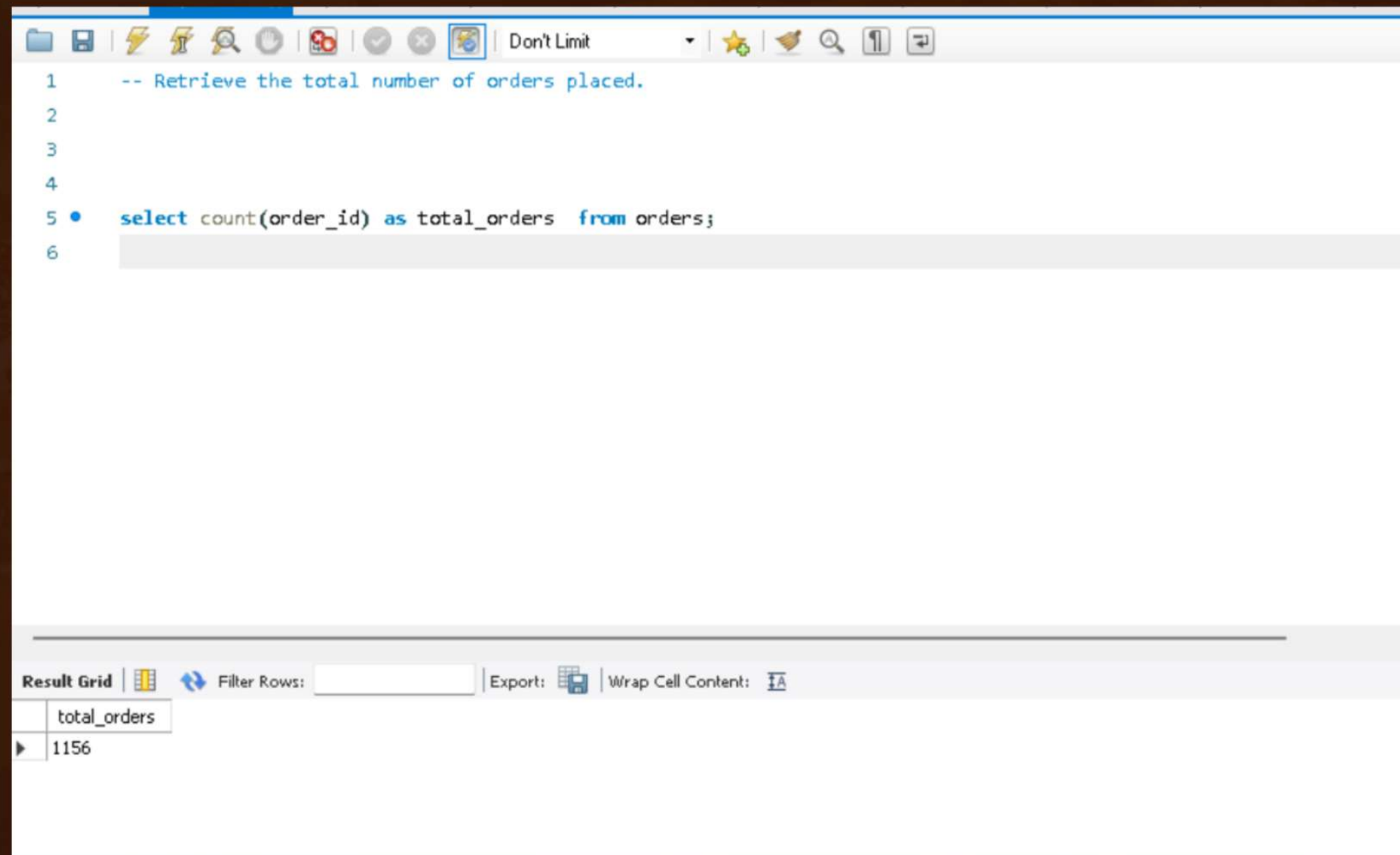


**ORDER
NOW**

DATABASE SCHEMA

- orders table (order_id, order_time, date)
- order_details table (order_id, pizza_id, quantity)
- pizzas table (pizza_id, name, size, price)
- pizza_types table (pizza_type_id, category, name)

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings. The main text area contains a SQL query with line numbers 1 through 6. The query is a single-line statement to count the total number of orders. The bottom toolbar includes options for the result grid, filtering, exporting, and wrapping cell content. The result grid at the bottom displays the output of the query.

```
1  -- Retrieve the total number of orders placed.  
2  
3  
4  
5  • select count(order_id) as total_orders from orders;  
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	total_orders
▶	1156

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4  ROUND(SUM(order_details.quantity * pizzas.price),
5        2) AS total_revenue
6  FROM
7  order_details
8  JOIN
9  pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid   Filter Rows: Export:  Wrap Cell Content: 

	total_revenue
▶	16590.45

ANALYZE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
1  -- analyze the distribution of orders by hour of the day.
2
3
4  • SELECT
5      HOUR(order_time) AS hour, COUNT(order_id) AS order_count
6  FROM
7      orders
8  GROUP BY HOUR(order_time);
```

Result Grid

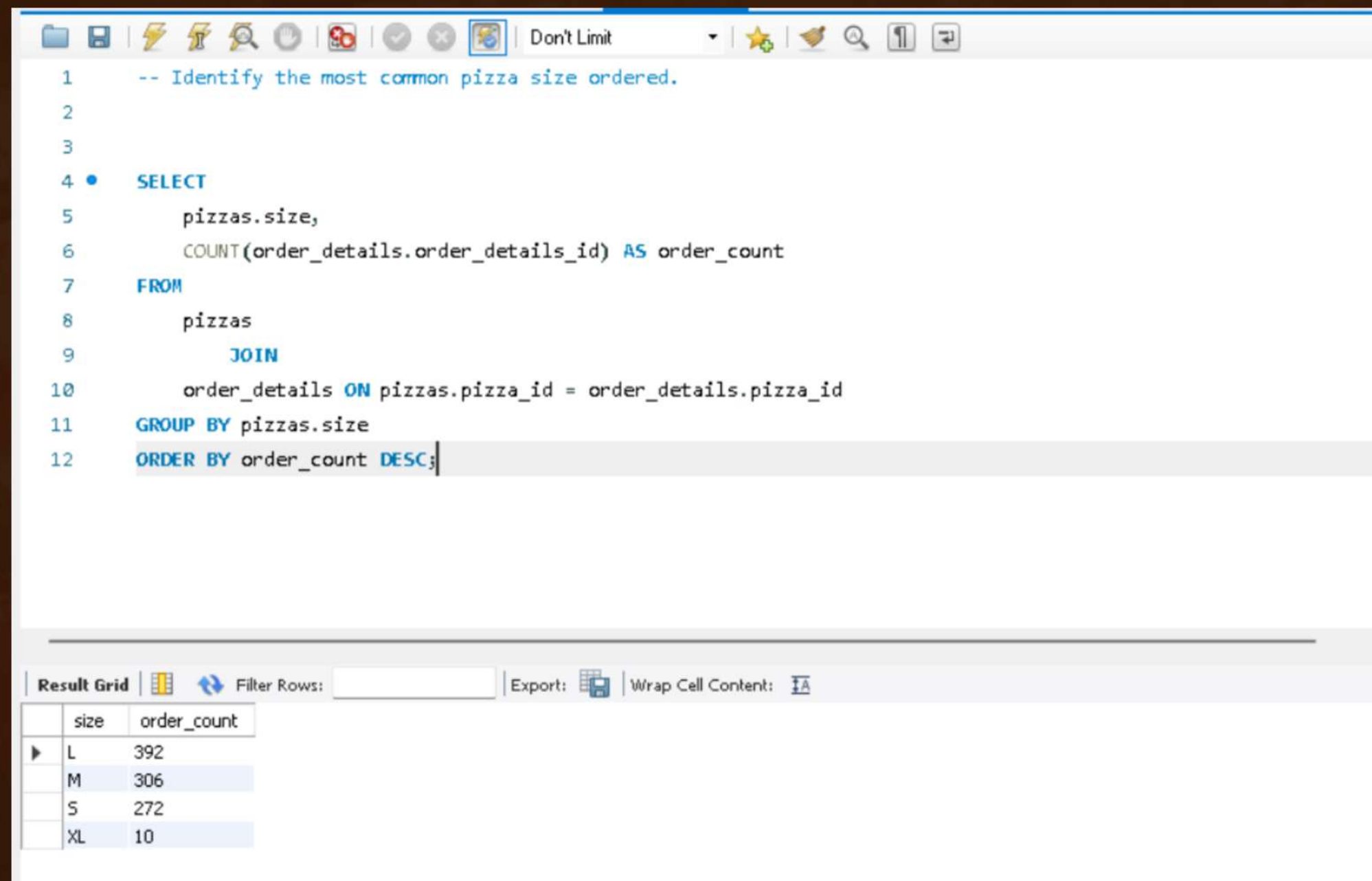
	hour	order_count
14	101	
15	85	
16	99	
17	130	
18	127	
19	111	
20	80	
21	57	
22	34	

Result 1 x

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3  • SELECT
4      pizza_types.category,
5      SUM(order_details.quantity) AS quantity
6  FROM
7      pizza_types
8      JOIN
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10     JOIN
11     order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY quantity DESC;
```


IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



The screenshot shows a SQL IDE window with a query editor and a result grid. The query is designed to identify the most common pizza size ordered by counting the number of orders for each size and sorting them in descending order.

```
1  -- Identify the most common pizza size ordered.  
2  
3  
4  • SELECT  
5      pizzas.size,  
6      COUNT(order_details.order_details_id) AS order_count  
7  FROM  
8      pizzas  
9      JOIN  
10     order_details ON pizzas.pizza_id = order_details.pizza_id  
11 GROUP BY pizzas.size  
12 ORDER BY order_count DESC;
```

The result grid displays the following data:

	size	order_count
▶	L	392
	M	306
	S	272
	XL	10



THANK YOU