



PIZZA SALES ANALYSIS

BUSINESS INSIGHTS USING SQL



VAIBHAV GOYAL

Table of Contents

1. Project Title & Cover Page
2. Project Overview
3. Dataset Description
4. Tools & Technologies
5. Data Cleaning & Preparation
6. Step-by-Step Analysis
 - 6.1 Total Orders Placed
 - 6.2 Total Revenue Generated
 - 6.3 Highest-Priced Pizza
 - 6.4 Most Common Pizza Size Ordered
 - 6.5 Top 5 Most Ordered Pizzas
 - 6.6 Category-Wise Quantity Ordered
 - 6.7 Order Distribution by Hour
 - 6.8 Category-Wise Pizza Distribution
 - 6.9 Average Pizzas Ordered per Day
 - 6.10 Top 3 Pizzas by Revenue
7. Key Business Insights
8. Conclusion
9. Screenshots

Detailed Project Summary – Pizza Sales Analysis (SQL)

This project focuses on analyzing Pizza Hut sales data using SQL to uncover valuable business insights. The dataset, consisting of approximately 50,000 records spread across four files (pizza_types.csv, pizzas.csv, orders.sql, and order_details.sql), provided details about pizza categories, sizes, prices, customer orders, and sales transactions.

The analysis was structured to answer 10 business questions aimed at evaluating revenue generation, customer behavior, product performance, and operational trends.

Step-by-Step Analysis

1 Total Orders Placed

The total number of unique orders placed was 21,350.

2 Total Revenue Generated

Revenue was computed using the formula $\text{quantity} \times \text{price}$. The total revenue generated from pizza sales was \$817,860.

3 Highest-Priced Pizza

The Greek Pizza was the highest-priced item, at \$35.95.

4 Most Common Pizza Size Ordered

Among the five available sizes, Large (L) pizzas were most popular with 18,526 orders, followed by Medium (15,385) and Small (14,137).

5 Top 5 Most Ordered Pizzas

Ranked by order quantity, the top 5 pizzas highlighted consistent customer preferences and demand concentration among certain SKUs.

6 Category-Wise Quantity Ordered

By grouping pizzas into categories (Classic, Supreme, Chicken, Veggie), Classic pizzas emerged as the most popular with 14,888 orders.

7 Order Distribution by Hour

Peak demand occurred between 12 PM – 1 PM, where orders exceeded 2,500 per hour.

8 Category-Wise Pizza Distribution

Both Classic and Chicken pizzas were leaders in overall distribution, reflecting strong customer preference.

9 Average Pizzas Ordered per Day






On average, 138 pizzas were sold per day, an important metric for inventory planning and forecasting.

Top 3 Pizzas by Revenue

Ranked by revenue contribution, the top 3 pizzas were:

- Thai Chicken Pizza → \$43,434.25
- Barbecue Chicken Pizza → \$42,768.00
- California Chicken Pizza → \$41,409.50

Key Business Insights

-  Revenue Drivers: Chicken-based pizzas consistently dominated both sales and revenue.
- Peak Hours: Lunchtime (12–1 PM) generated the highest sales volumes.
-  Size Preference: Large pizzas (L) were overwhelmingly popular.
-  Premium Pricing: The Greek Pizza was the most expensive but not a top seller.
-  Top Performer: The Thai Chicken Pizza was the single largest revenue contributor.
-  Category Trends: Classic and Chicken pizzas drove the majority of customer demand.

Conclusion

This SQL-based analysis of Pizza Hut's sales data provided actionable insights into product performance, customer behavior, and sales patterns.

Key outcomes include:

- **Menu Engineering:** Focus on high-revenue pizzas and optimize underperforming ones.
- **Operational Planning:** Align staffing with peak lunchtime demand.
- **Promotions & Marketing:** Bundle deals and targeted offers for popular sizes and categories.

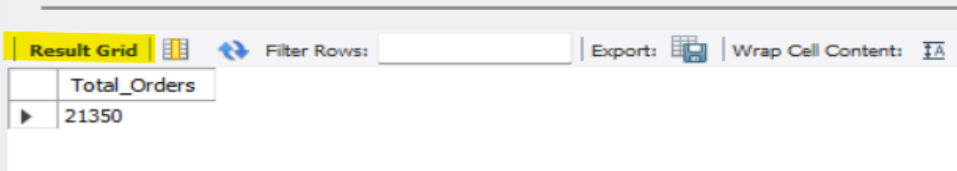
Through structured SQL queries, this project transformed raw data into business intelligence that can directly guide decision-making, menu optimization, and growth strategies.

Screenshots for each queries have been pasted below for reference:

Screenshots

- **Total Orders**

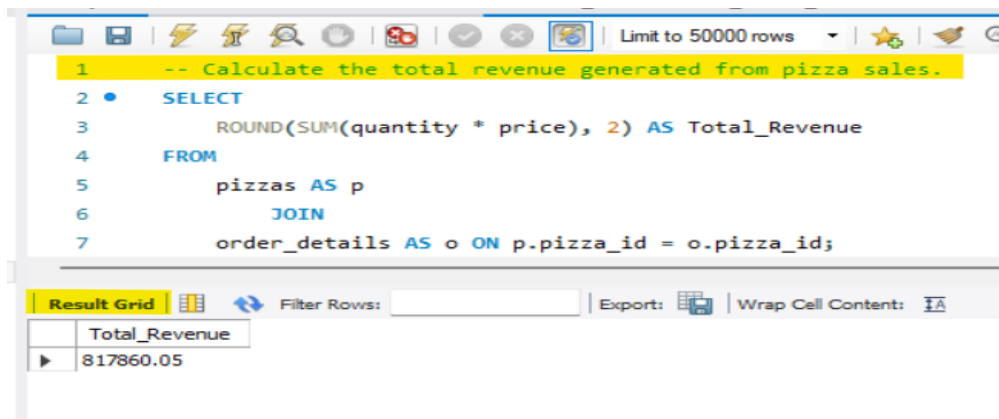
```
3  -- Retrieve the total number of orders placed.
4  •  select count(order_id) as Total_Orders
5     from orders;
```



The screenshot shows the SQL query results in a 'Result Grid' format. The query is: `select count(order_id) as Total_Orders from orders;`. The result is a single row with the column name 'Total_Orders' and the value '21350'.

Total_Orders
21350

Total Revenue

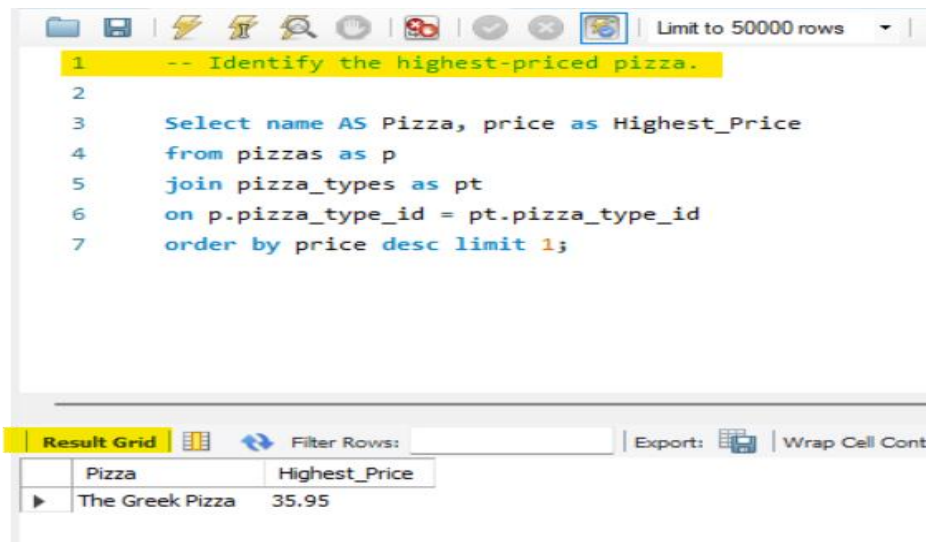


The screenshot shows the SQL query editor with the following query: `SELECT ROUND(SUM(quantity * price), 2) AS Total_Revenue FROM pizzas AS p JOIN order_details AS o ON p.pizza_id = o.pizza_id;`. The query is highlighted in yellow. Below the query, the 'Result Grid' shows the result: a single row with the column name 'Total_Revenue' and the value '817860.05'.

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

Total_Revenue
817860.05

Highest Priced Pizza



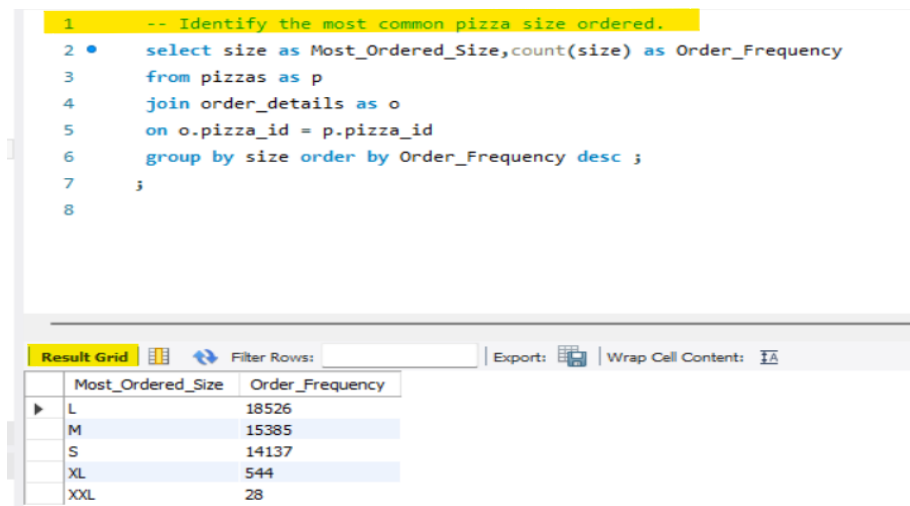
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The SQL editor contains the following query:

```
1  -- Identify the highest-priced pizza.
2
3  Select name AS Pizza, price as Highest_Price
4  from pizzas as p
5  join pizza_types as pt
6  on p.pizza_type_id = pt.pizza_type_id
7  order by price desc limit 1;
```

Below the editor is the 'Result Grid' tab, which displays the query results in a table:

Pizza	Highest_Price
The Greek Pizza	35.95

Most Ordered Pizza Size



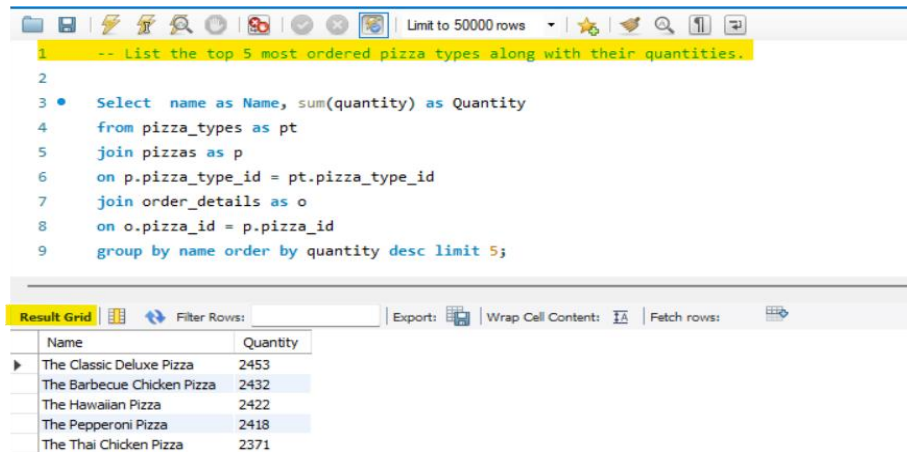
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The SQL editor contains the following query:

```
1  -- Identify the most common pizza size ordered.
2  • select size as Most_Ordered_Size, count(size) as Order_Frequency
3  from pizzas as p
4  join order_details as o
5  on o.pizza_id = p.pizza_id
6  group by size order by Order_Frequency desc ;
7  ;
8
```

Below the editor is the 'Result Grid' tab, which displays the query results in a table:

Most_Ordered_Size	Order_Frequency
L	18526
M	15385
S	14137
XL	544
XXL	28

Top 5 Most Ordered Pizzas

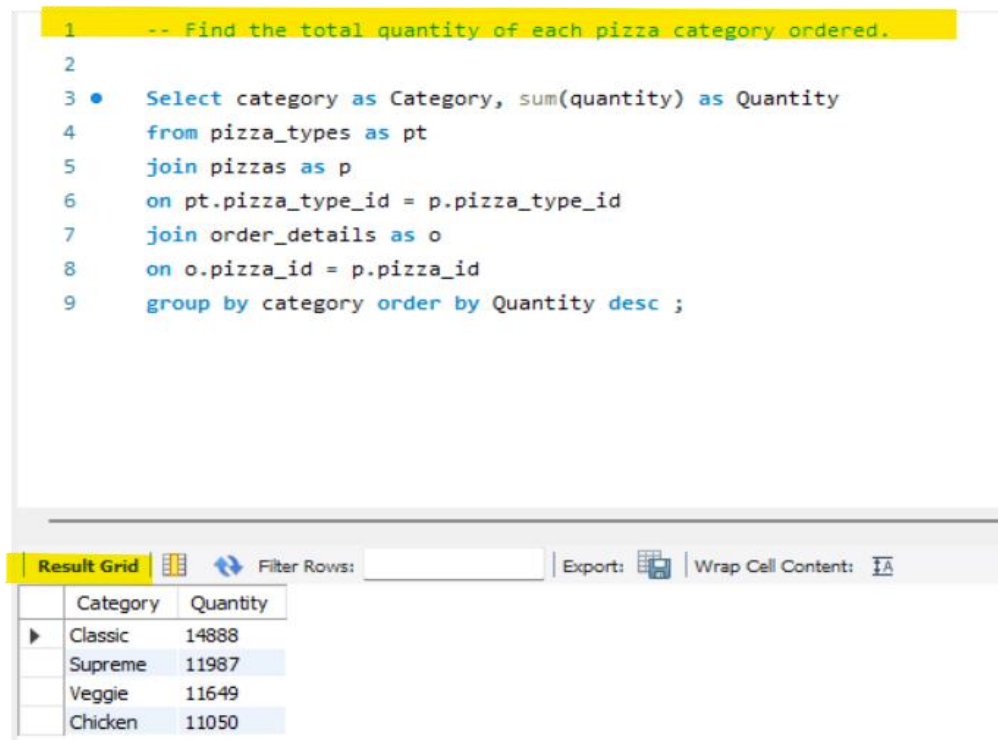


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 50000 rows' dropdown. The SQL editor contains a query to find the top 5 most ordered pizzas. Below the editor is the 'Result Grid' tab, which displays a table with two columns: 'Name' and 'Quantity'. The results are sorted in descending order of quantity.

```
1 -- List the top 5 most ordered pizza types along with their quantities.
2
3 • Select name as Name, sum(quantity) as Quantity
4   from pizza_types as pt
5   join pizzas as p
6   on p.pizza_type_id = pt.pizza_type_id
7   join order_details as o
8   on o.pizza_id = p.pizza_id
9   group by name order by quantity desc limit 5;
```

Name	Quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Most Ordered Category

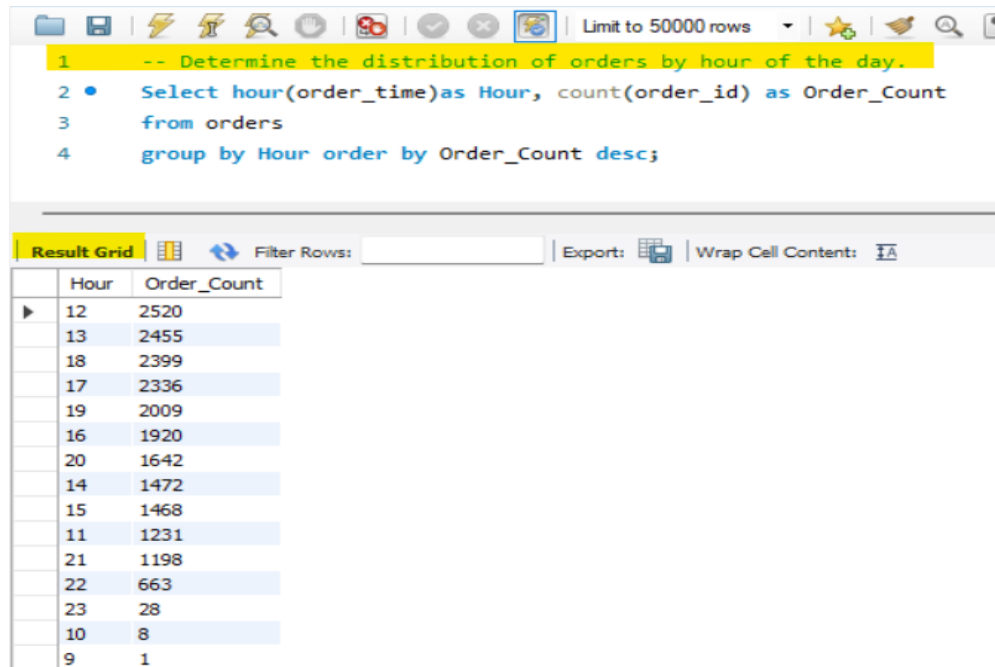


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 50000 rows' dropdown. The SQL editor contains a query to find the total quantity of each pizza category ordered. Below the editor is the 'Result Grid' tab, which displays a table with two columns: 'Category' and 'Quantity'. The results are sorted in descending order of quantity.

```
1 -- Find the total quantity of each pizza category ordered.
2
3 • Select category as Category, sum(quantity) as Quantity
4   from pizza_types as pt
5   join pizzas as p
6   on pt.pizza_type_id = p.pizza_type_id
7   join order_details as o
8   on o.pizza_id = p.pizza_id
9   group by category order by Quantity desc ;
```

Category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Hours of Maximum Sales

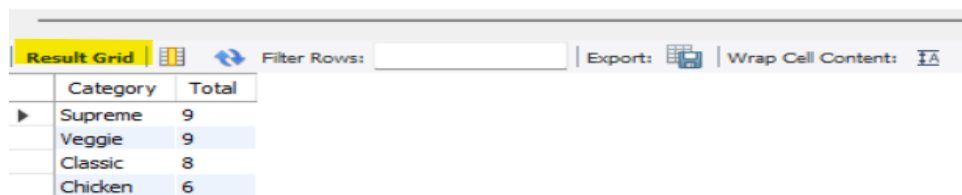


```
1 -- Determine the distribution of orders by hour of the day.
2 • Select hour(order_time) as Hour, count(order_id) as Order_Count
3 from orders
4 group by Hour order by Order_Count desc;
```

	Hour	Order_Count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

Category Distribution

```
1 -- Find the category-wise distribution of pizzas.
2
3 • Select Category, count(Category) as Total
4 from pizza_types
5 group by category order by Total desc;
```



	Category	Total
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

Average Pizza Sold Per Day

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3  • Select round(avg(total),0) as Avergae_PerDay from
4  (Select order_date, sum(quantity) as Total
5   from orders as o
6   join order_details as od
7   on o.order_id = od.order_id
8   group by o.order_date) as Order_Quantity;
9
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Avergae_PerDay				
▶	138			

Top 3 Most Selling Pizzas

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2
3  • Select Name as Name, sum(quantity*price) as Revenue
4    from pizza_types as pt
5    join pizzas as p
6    on p.pizza_type_id = pt.pizza_type_id
7    join order_details as od
8    on od.pizza_id = p.pizza_id
9    group by Name order by Revenue desc limit 3;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
Name		Revenue			
▶	The Thai Chicken Pizza	43434.25			
	The Barbecue Chicken Pizza	42768			
	The California Chicken Pizza	41409.5			