

SQL Project Pizza Sales Analysis



Project Summary

1. Objective: To analyze pizza sales data to:

- to provide actionable insights on sales performance,
- customer preferences,
- revenue generation.

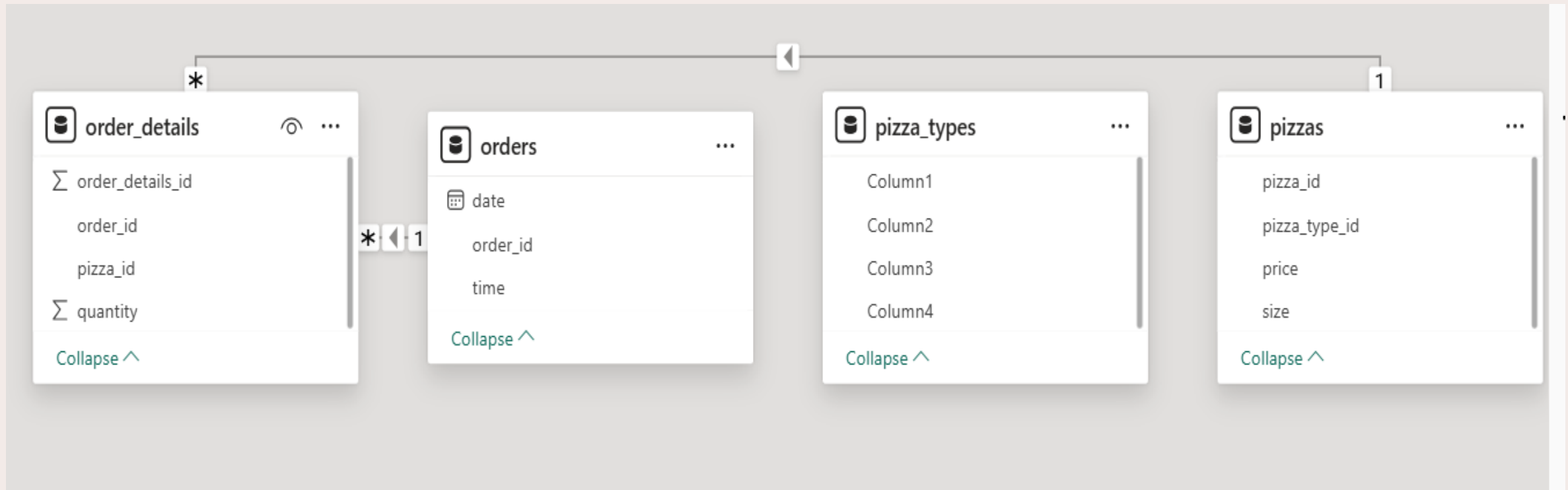
2. About the dataset: Analyzing sales records from a pizza restaurant, spanning one year. Dataset has 48,620 sales transaction to analyze.

3. Features of the project:

- Sub Query
- Group by
- Cumulative Sum
- Rank

4. Tools Used: MySQL for data querying and manipulation.

Data Schema



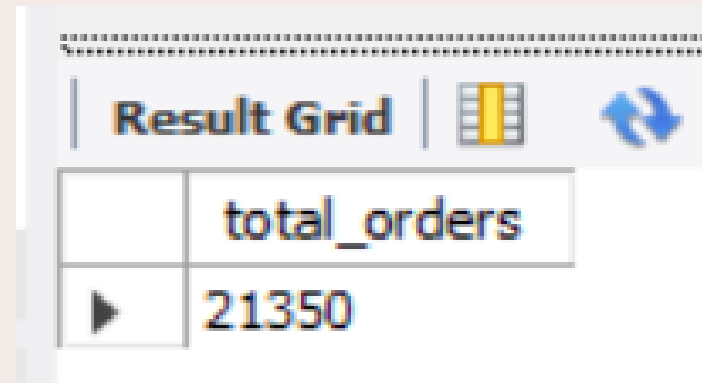
“

Basic Questions

”

Retrieve the total number of orders placed.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```



The screenshot shows a database interface with a 'Result Grid' header. Below the header, there is a table with two rows. The first row has a column header 'total_orders'. The second row has a value '21350' in the same column. There is a small icon of a document with a yellow highlight and a blue double-headed arrow icon to the right of the header.

	total_orders
▶	21350

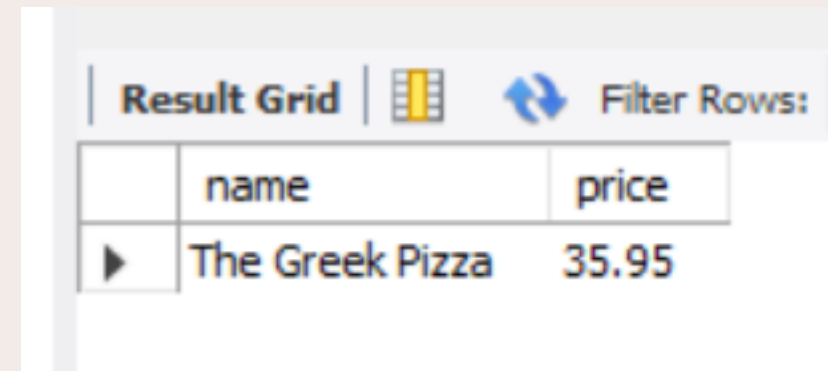
Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
          2) AS total_sales
FROM
    order_details
    JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	total_sales
▶	817860.05

Identify the highest-priced pizza.

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```





The screenshot shows a 'Result Grid' interface with a table containing one row of data. The table has two columns: 'name' and 'price'. The row shows 'The Greek Pizza' with a price of 35.95. Above the table, there are icons for a grid, a refresh button, and a 'Filter Rows:' label.

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
SELECT pizzas.size , count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

Result Grid					Filter
	size	order_count			
▶	L	18526			
	M	15385			
	S	14137			
	XL	544			
	XXL	28			

List the top 5 most ordered pizza types along with their quantities.

```
SELECT pizza_types.name,  
sum(order_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.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

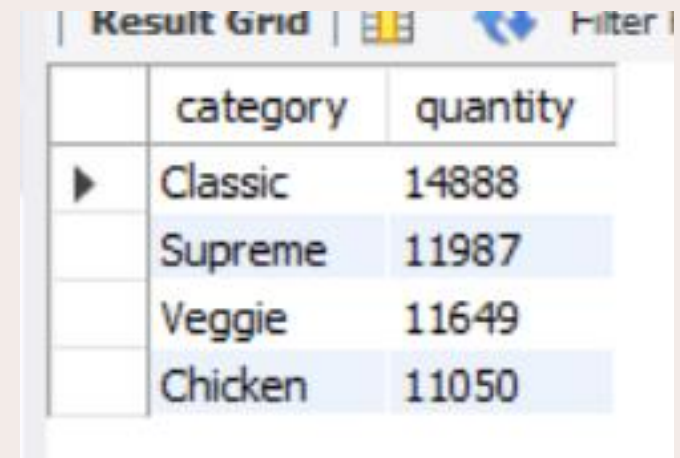
“

Intermediate Questions

”

Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT pizza_types.category,  
sum(order_details.quantity) as quantity  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by quantity desc limit 5;
```

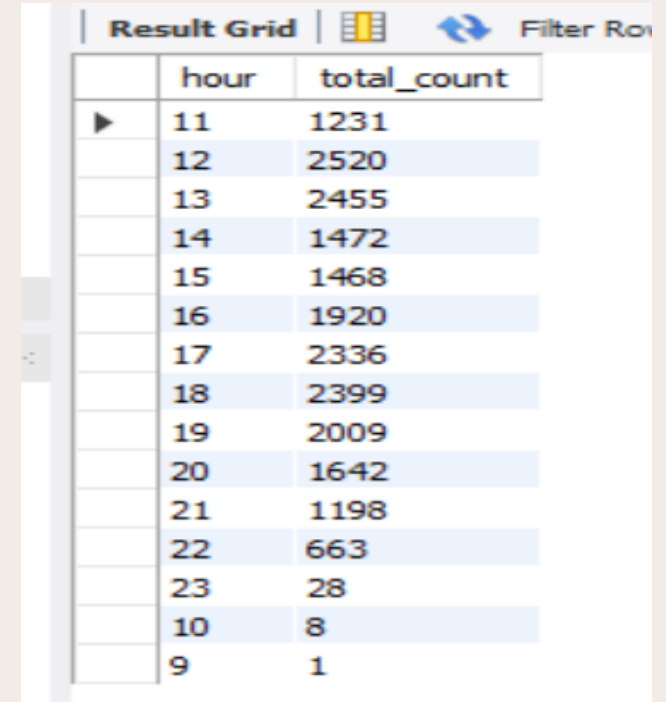


The screenshot shows a 'Result Grid' window with a table containing the top 5 pizza categories by quantity. The table has two columns: 'category' and 'quantity'. The rows are: Classic (14888), Supreme (11987), Veggie (11649), and Chicken (11050). The first row is highlighted with a play button icon in the first column.

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS total_count
FROM
    orders
GROUP BY HOUR(order_time);
```




The screenshot shows a 'Result Grid' with a table containing two columns: 'hour' and 'total_count'. The data is sorted by hour in descending order, from 11 down to 9. The table has a light blue header and alternating light blue and white rows. A 'Filter Rows' button is visible in the top right corner of the grid.

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

Join relevant tables to find the category-wise distribution of pizzas.

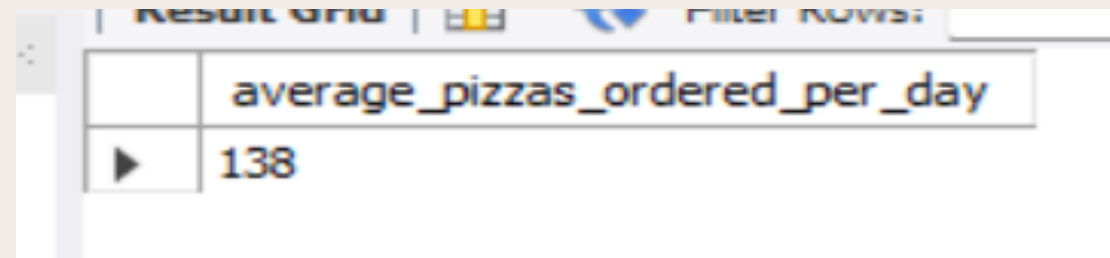
```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid |   Filter Rows:

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT round(avg(quantity),0) as average_pizzas_ordered_per_day from
(SELECT orders.order_date, sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```



The screenshot shows a database result grid with the following data:

	average_pizzas_ordered_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue.

```
SELECT pizza_types.name,  
sum(order_details.quantity * pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.name order by revenue desc limit 3;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

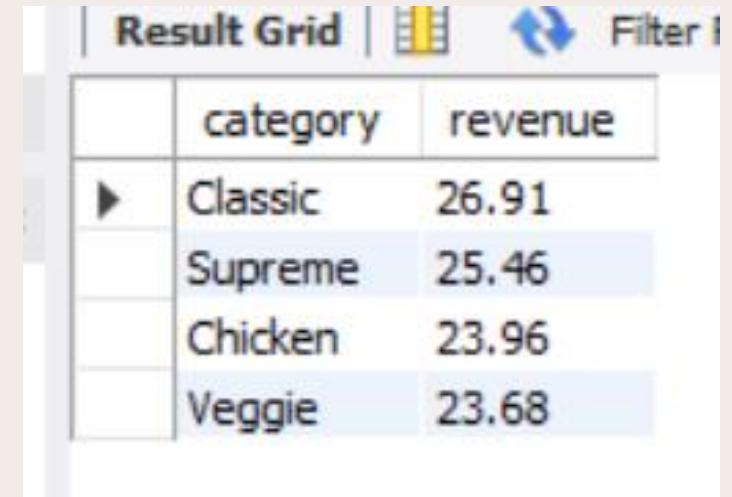
“

Advanced Questions

”

Calculate the percentage contribution of each pizza type to total revenue.

```
SELECT pizza_types.category,  
round(sum(order_details.quantity * pizzas.price)/ (SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_sales  
FROM  
    order_details  
    JOIN  
        pizzas ON pizzas.pizza_id = order_details.pizza_id)* 100, 2)as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```



The screenshot shows a 'Result Grid' window with a table of pizza categories and their revenue. The table has two columns: 'category' and 'revenue'. The data is sorted in descending order of revenue. The categories listed are Classic, Supreme, Chicken, and Veggie. The revenue values are 26.91, 25.46, 23.96, and 23.68 respectively. The window also includes a 'Filter' button and a 'Result Grid' title bar.

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Analyze the cumulative revenue generated over time.

```
SELECT order_date,  
sum(revenue) over (order by order_date) as cum_revenue  
from  
(SELECT orders.order_date,  
SUM(order_details.quantity * pizzas.price) AS revenue  
FROM order_details JOIN pizzas  
ON pizzas.pizza_id = order_details.pizza_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales;
```

Result Grid			Filter Rows:
	order_date	cum_revenue	
▶	2015-01-01	2713.8500000000004	
	2015-01-02	5445.75	
	2015-01-03	8108.15	
	2015-01-04	9863.6	
	2015-01-05	11929.55	
	2015-01-06	14358.5	
	2015-01-07	16560.7	
	2015-01-08	19399.05	
	2015-01-09	21526.4	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	
	2015-01-13	29831.300000000003	
	2015-01-14	32358.700000000004	
	2015-01-15	34343.50000000001	
	2015-01-16	36937.65000000001	
	2015-01-17	39001.75000000001	

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
SELECT category, name, revenue from
(SELECT category, name, revenue,
rank() over (partition by category order by revenue desc) as rn
from
(SELECT pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3;
```

Result Grid				Filter Rows:	Export:
	category	name	revenue		
▶	Chicken	The Thai Chicken Pizza	43434.25		
	Chicken	The Barbecue Chicken Pizza	42768		
	Chicken	The California Chicken Pizza	41409.5		
	Classic	The Classic Deluxe Pizza	38180.5		
	Classic	The Hawaiian Pizza	32273.25		
	Classic	The Pepperoni Pizza	30161.75		
	Supreme	The Spicy Italian Pizza	34831.25		
	Supreme	The Italian Supreme Pizza	33476.75		
	Supreme	The Sicilian Pizza	30940.5		
	Veggie	The Four Cheese Pizza	32265.70000000065		
	Veggie	The Mexicana Pizza	26780.75		
	Veggie	The Five Cheese Pizza	26066.5		

Thank you

Muskan Kataria

<https://github.com/muskankataria-source/SQL-Project-Pizza-Sales-Analysis>

<https://www.linkedin.com/in/muskan-kataria-a38443342/>

