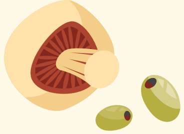
PIZZA SALES PROJECT

PIZZA SALES ANALYSIS USING SQL

TABLE OF CONTENTS



01 About Project

We have Solved and Analysed 13 Pizza sales Questions using Sql Queries.

02 Objective

You will see objective of this project.

03 Tools Used and Methodology

Which tools and Methodology have I used in this project.

04 Data Model Overview

You will see relation between pizza tables.

05 Excel Table Overview

You will see Excel tables Overview.

06 Table of Questions

There are 13 Sql Questons of pizza sales.

07 Sql Queries and Output

You will see 13 Sql Queries and result of these Sql Queries.

08 Insight

What insight have we got in this Project.

09 Conclusion

Conclusion about this Project.

01

My Name is Aryan Raj.

In this Project, I have utilized the SQL Queries to solve 13 Questions that were related to Pizza sales.

Analyse the Pizza Sales using SQL Queries

02

Objective

The objective of this Project is to analyze pizza sales data to identify trends and provide actionable insights that can help to increase sales and aim to uncover key metrics and patterns within the sales data by leveraging SQL queries in MySQL.

Tools Used

- Database Management System : MySQL.
- Query Tool : MySQL Workbench

Methodology

Database Setup :

- Created a database named "pizzahut" and Imported tables (orders ,pizzas , order_details , pizza_types) into the database.

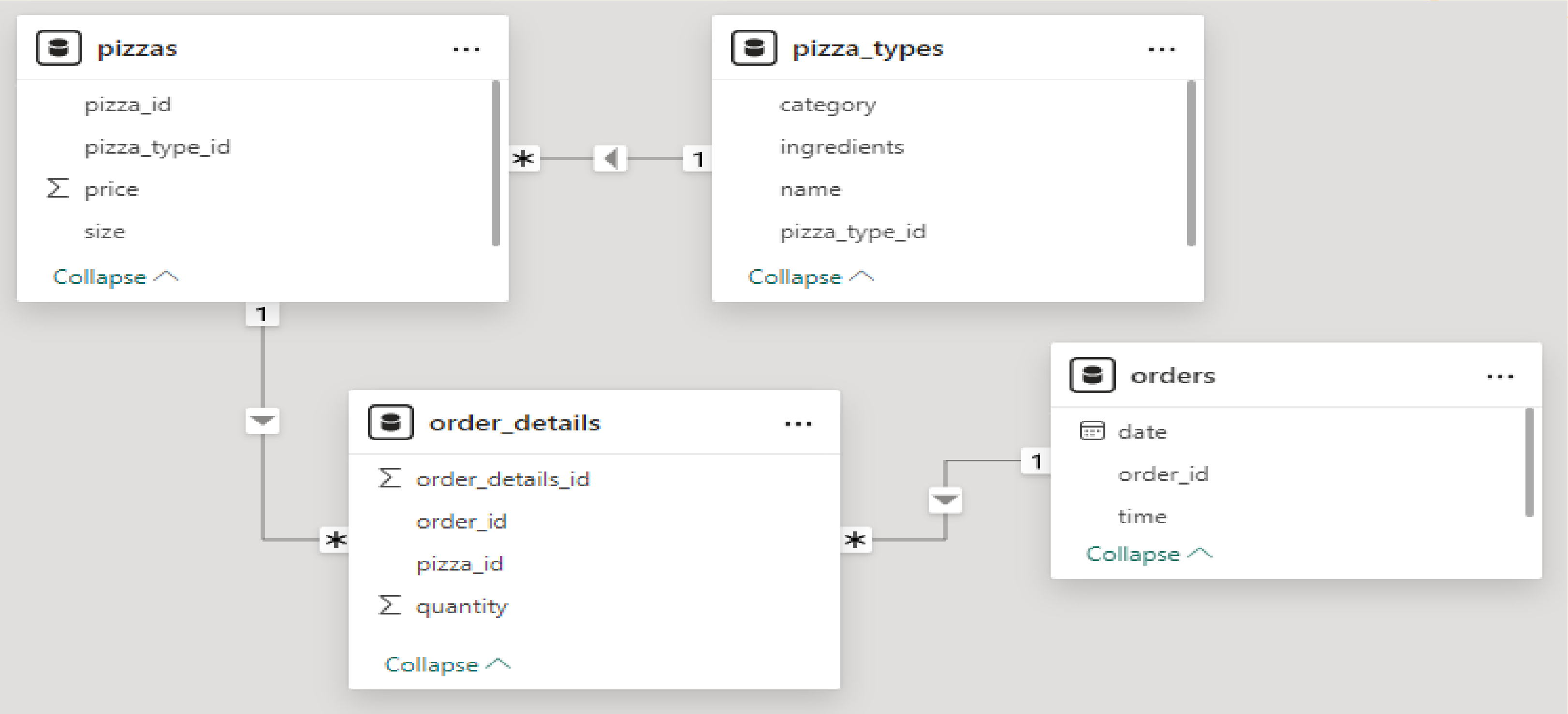
Data Exploration and Cleaning :

- Explored the structure and contents of each table to understand the relationships and data points.
- Ensured data integrity by checking for null values, duplicates, and any inconsistencies.

Querying and Analysis :

- SQL queries were written to extract meaningful insights from the data.
- These queries addressed various aspects of pizza sales, including total orders, revenue generation, popular pizza types, and order distribution.

04 Data Model Overview



pizzas

- pizza_id
- pizza_type_id
- Σ price
- size

Collapse

pizza_types

- category
- ingredients
- name
- pizza_type_id

Collapse

order_details

- Σ order_details_id
- order_id
- pizza_id
- Σ quantity

Collapse

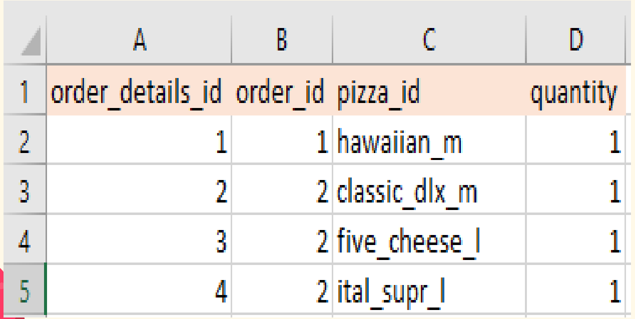
orders

- date
- order_id
- time

Collapse

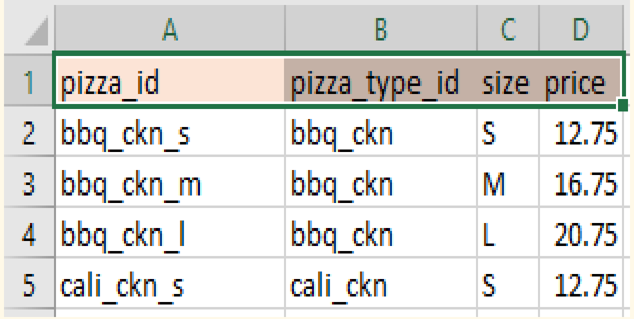
05 Excel Table Overview

Order Detail table



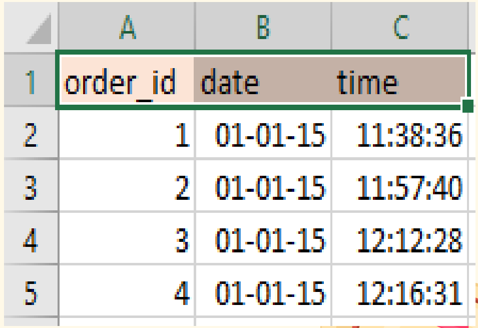
	A	B	C	D
1	order_details_id	order_id	pizza_id	quantity
2	1	1	hawaiian_m	1
3	2	2	classic_dlx_m	1
4	3	2	five_cheese_l	1
5	4	2	ital_supr_l	1

Pizzas table



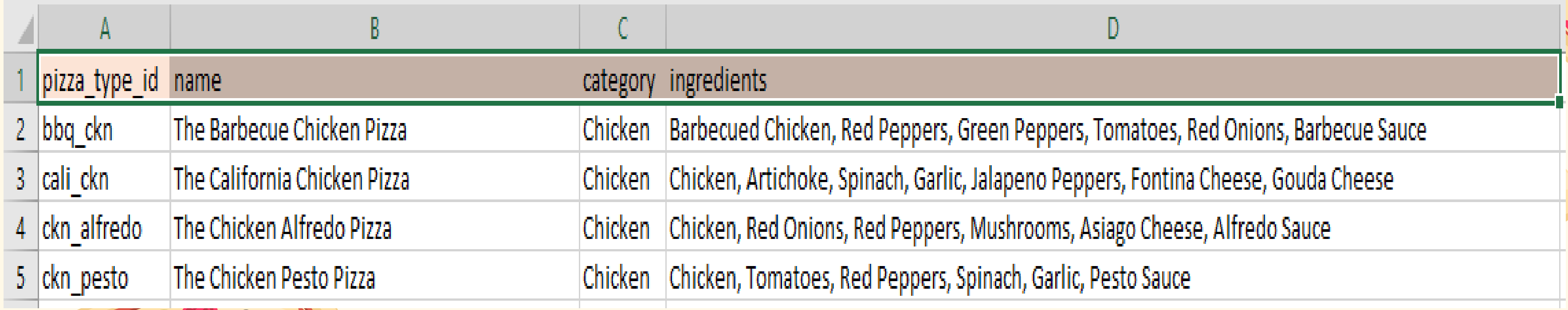
	A	B	C	D
1	pizza_id	pizza_type_id	size	price
2	bbq_ckn_s	bbq_ckn	S	12.75
3	bbq_ckn_m	bbq_ckn	M	16.75
4	bbq_ckn_l	bbq_ckn	L	20.75
5	cali_ckn_s	cali_ckn	S	12.75

Order table



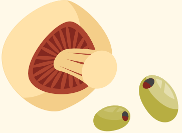
	A	B	C
1	order_id	date	time
2	1	01-01-15	11:38:36
3	2	01-01-15	11:57:40
4	3	01-01-15	12:12:28
5	4	01-01-15	12:16:31

Pizza Type table



	A	B	C	D
1	pizza_type_id	name	category	ingredients
2	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppers, Tomatoes, Red Onions, Barbecue Sauce
3	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno Peppers, Fontina Cheese, Gouda Cheese
4	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms, Asiago Cheese, Alfredo Sauce
5	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garlic, Pesto Sauce

06 TABLE OF QUESTIONS



01 Retrieve the total number of orders placed.

02 Calculate the total revenue generated from pizza sales.

03 Identify the highest-priced pizza.

04 Identify the most common pizza size ordered..

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

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

07 Determine the distribution of orders by hour of the day.

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

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

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

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

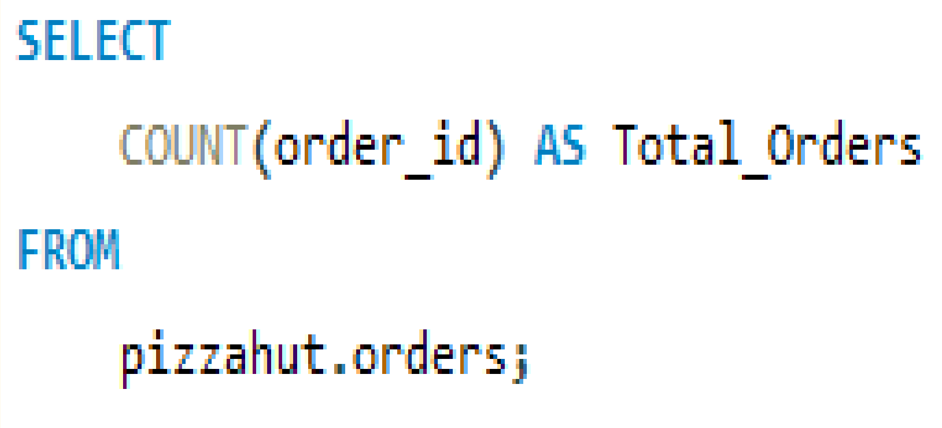
12 Analyze the cumulative revenue generated over time.

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

07

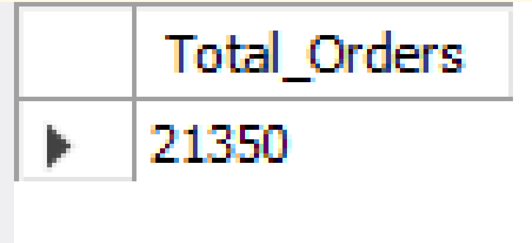
01 Retrieve the total number of orders placed.

Query



```
SELECT
    COUNT(order_id) AS Total_Orders
FROM
    pizzahut.orders;
```

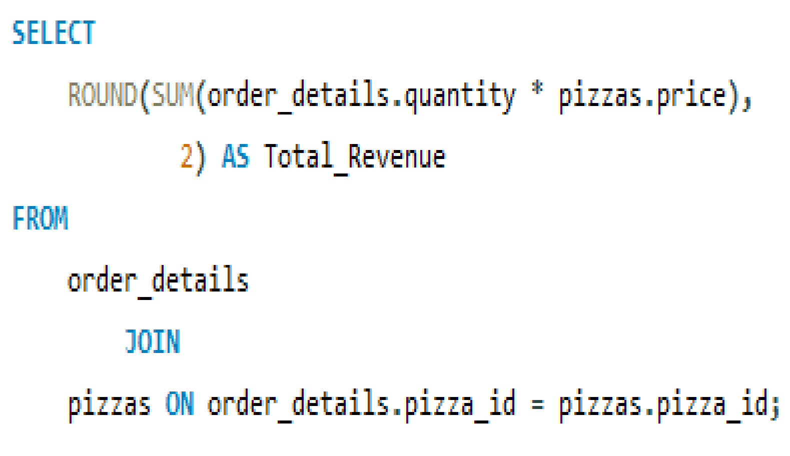
OUTPUT



	Total_Orders
▶	21350

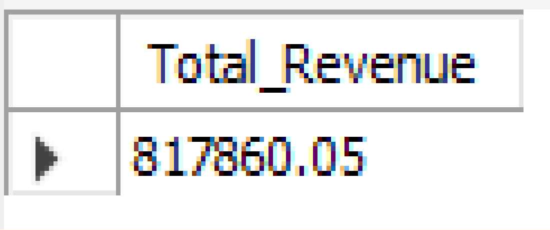
02 Calculate the total revenue generated from pizza sales.

Query



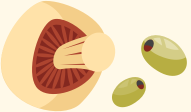
```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Total_Revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

OUTPUT

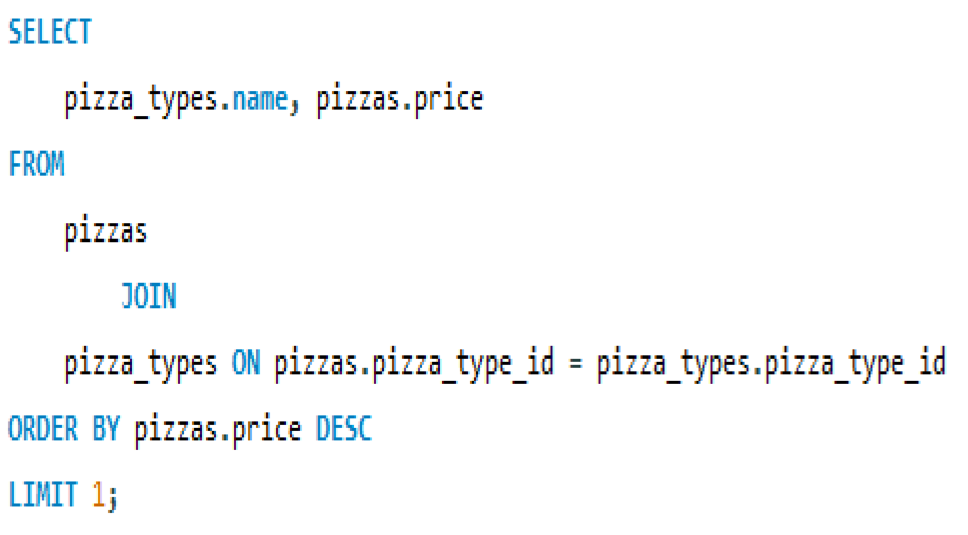


```
   | Total_Revenue
 ▶ | 817860.05
```

03 Identify the highest-priced pizza

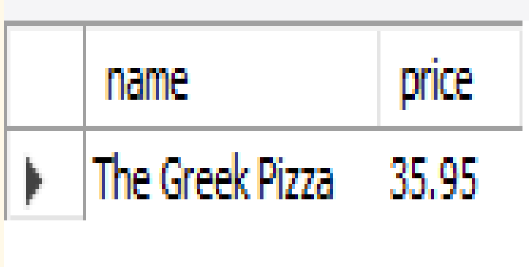


Query

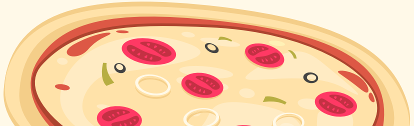


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

OUTPUT



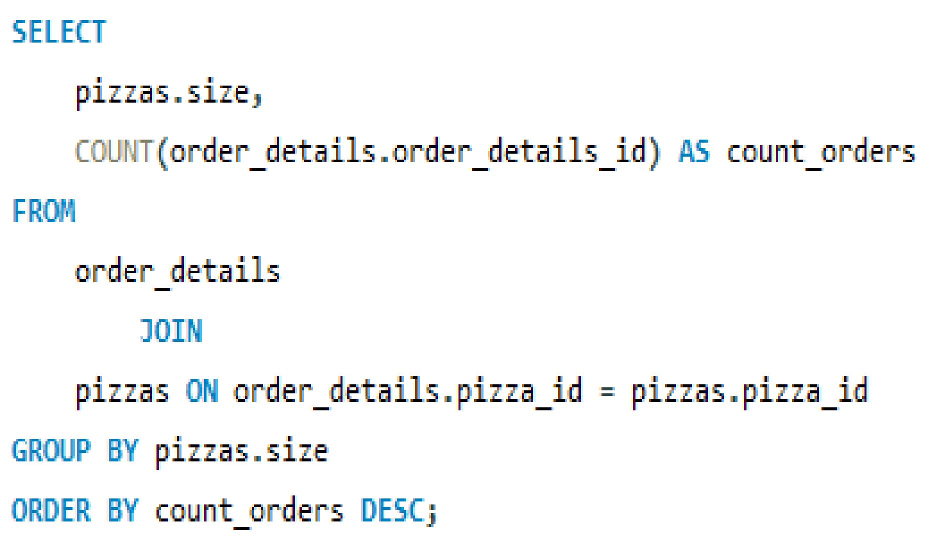
	name	price
▶	The Greek Pizza	35.95



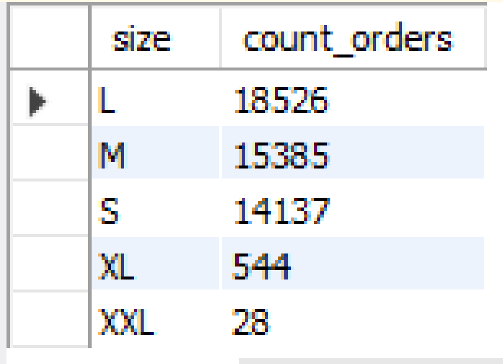
04 Identify the most common pizza size ordered.

Query

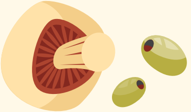
OUTPUT



```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS count_orders
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY count_orders DESC;
```

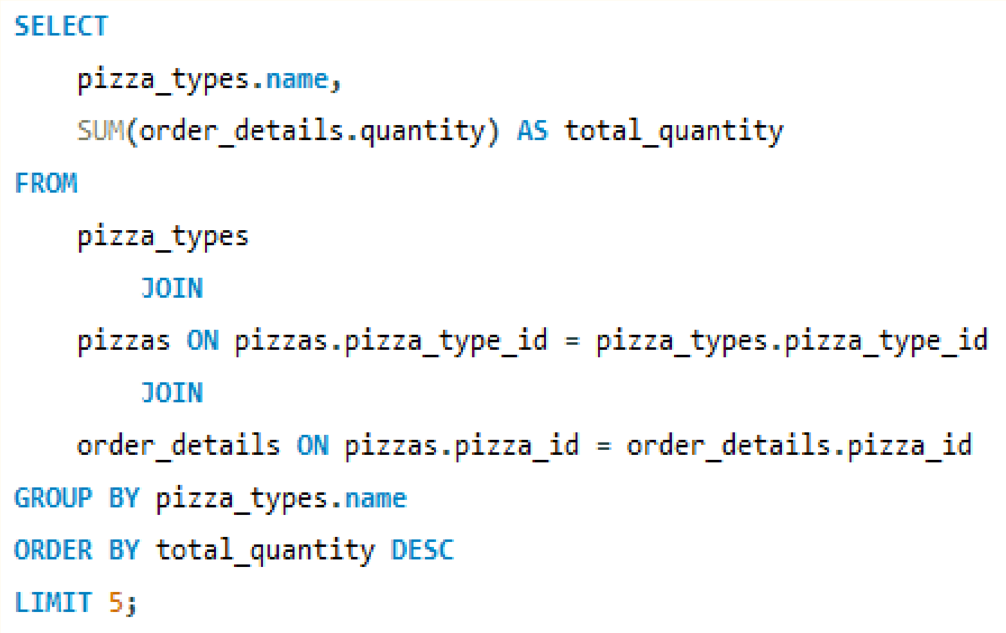


	size	count_orders
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



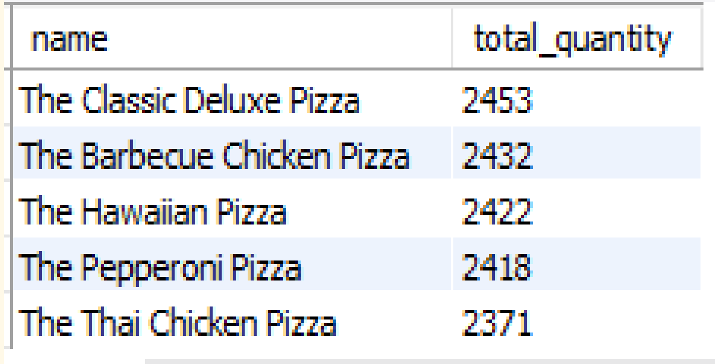
05 List the top 5 most ordered pizza types along with their quantities.

Query

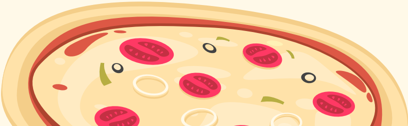


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5;
```

OUTPUT

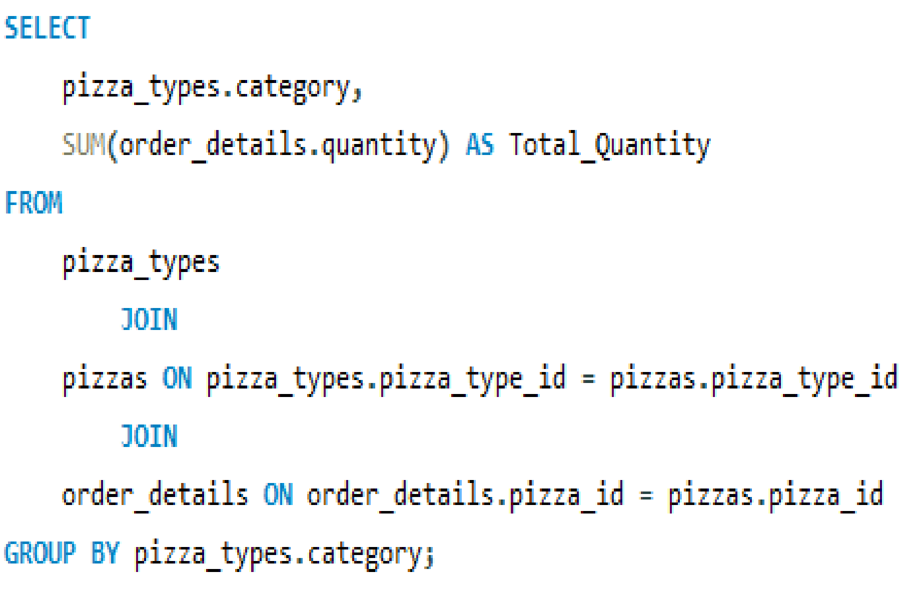


name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



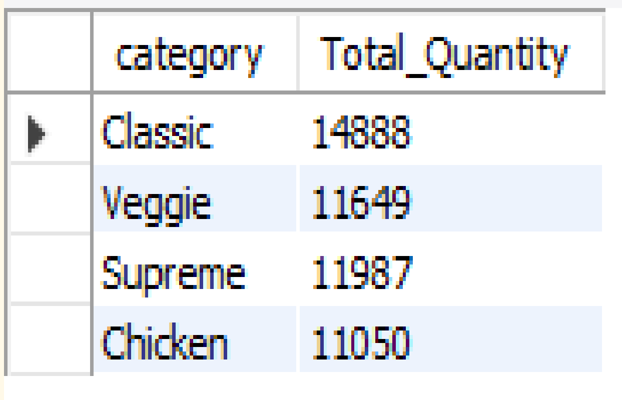
06 Join the necessary tables to find the total quantity of each pizza category ordered.

Query



```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS Total_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;
```

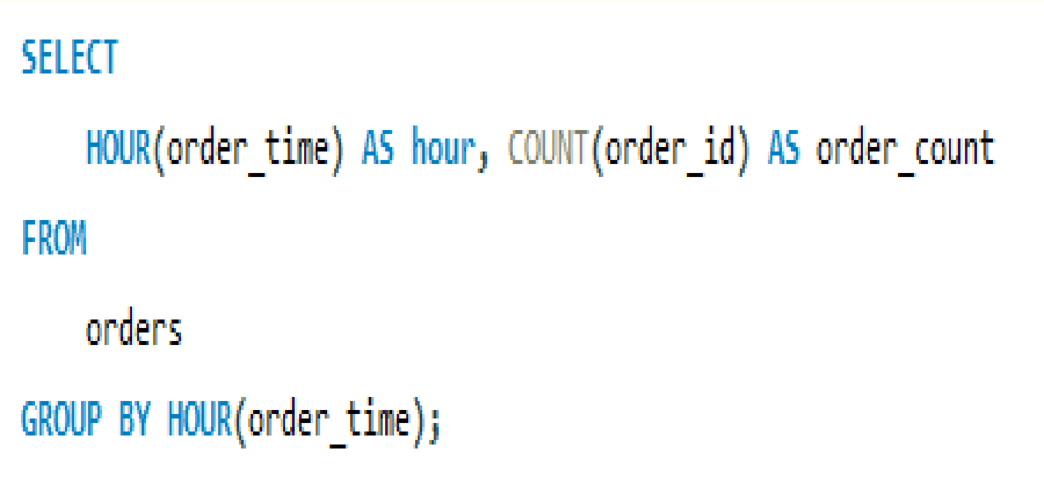
OUTPUT



	category	Total_Quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

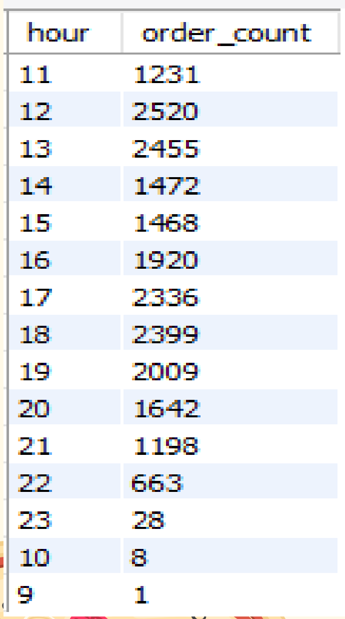
07 Determine the distribution of orders by hour of the day.

Query

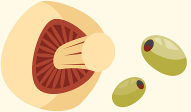


```
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

OUTPUT

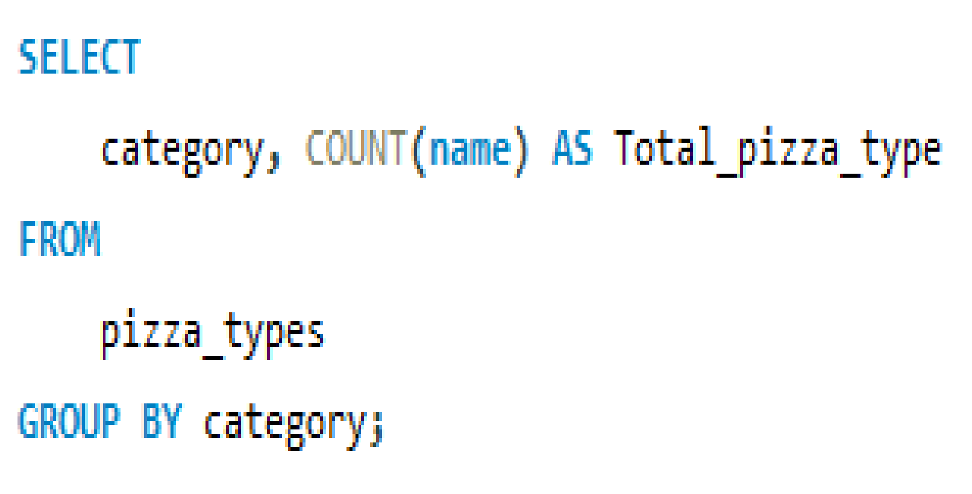


hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1



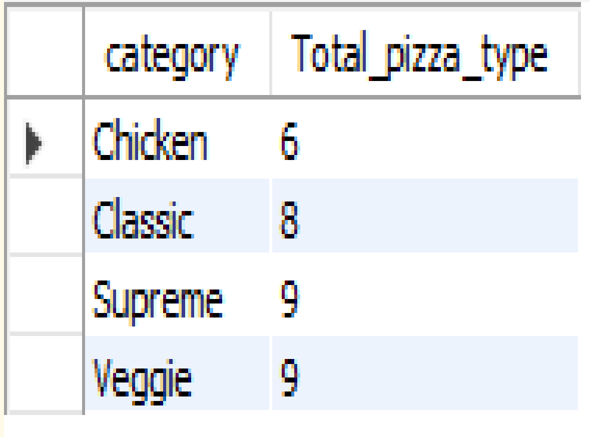
08 Join relevant tables to find the category-wise distribution of pizzas.

Query

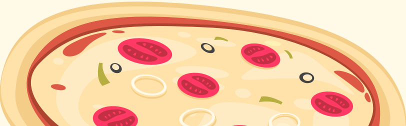


```
SELECT
    category, COUNT(name) AS Total_pizza_type
FROM
    pizza_types
GROUP BY category;
```

OUTPUT

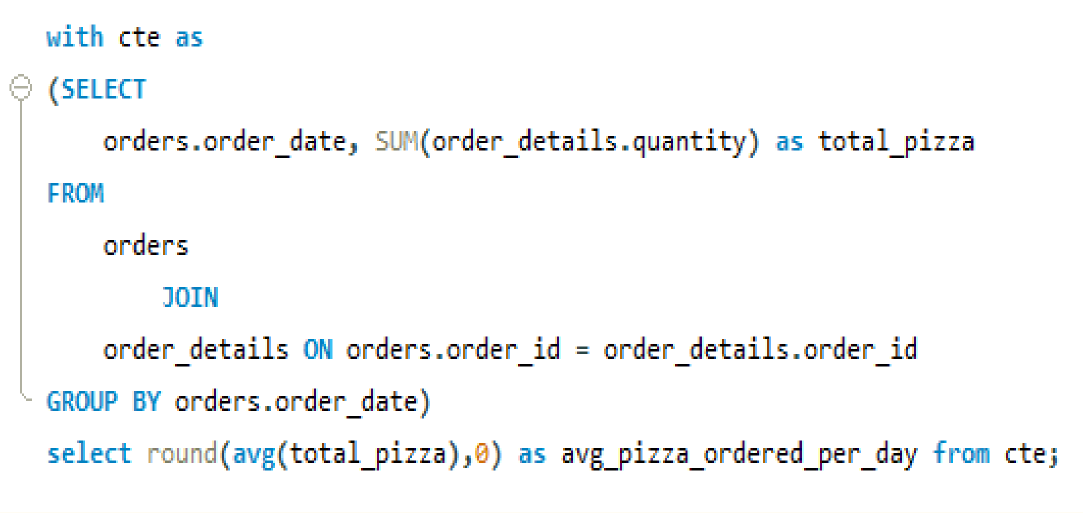


category	Total_pizza_type
Chicken	6
Classic	8
Supreme	9
Veggie	9



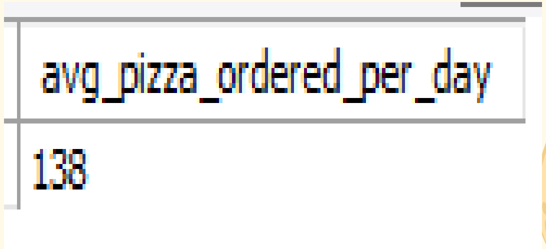
09 Group the orders by date and calculate the average number of pizzas ordered per day.

Query



```
with cte as
(SELECT
    orders.order_date, SUM(order_details.quantity) as total_pizza
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY orders.order_date)
select round(avg(total_pizza),0) as avg_pizza_ordered_per_day from cte;
```

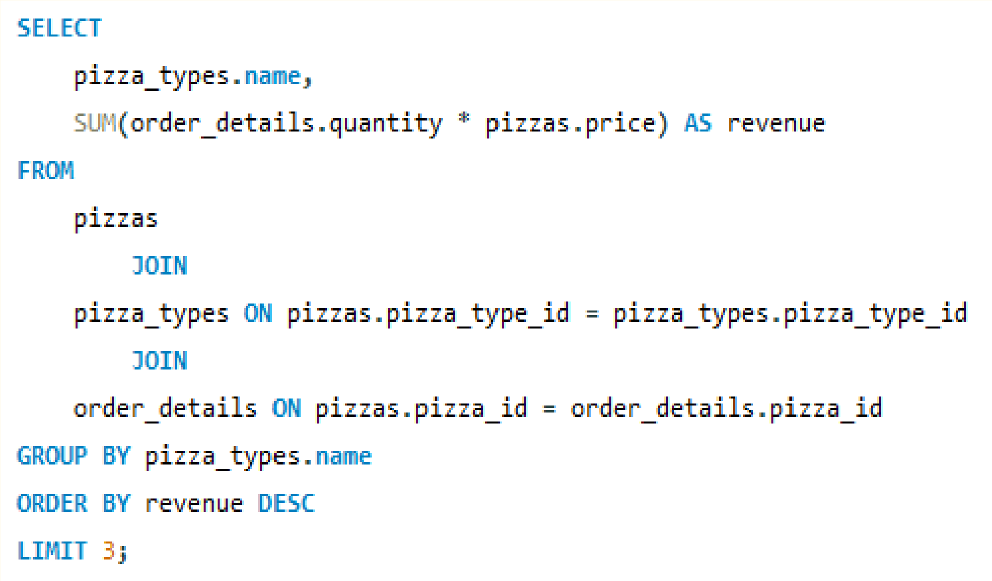
OUTPUT



```
avg_pizza_ordered_per_day
138
```

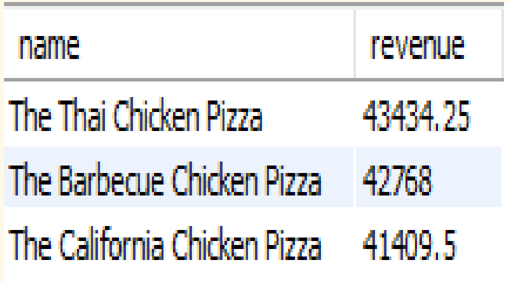
10 Determine the top 3 most ordered pizza types based on revenue.

Query

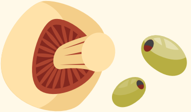


```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

OUTPUT



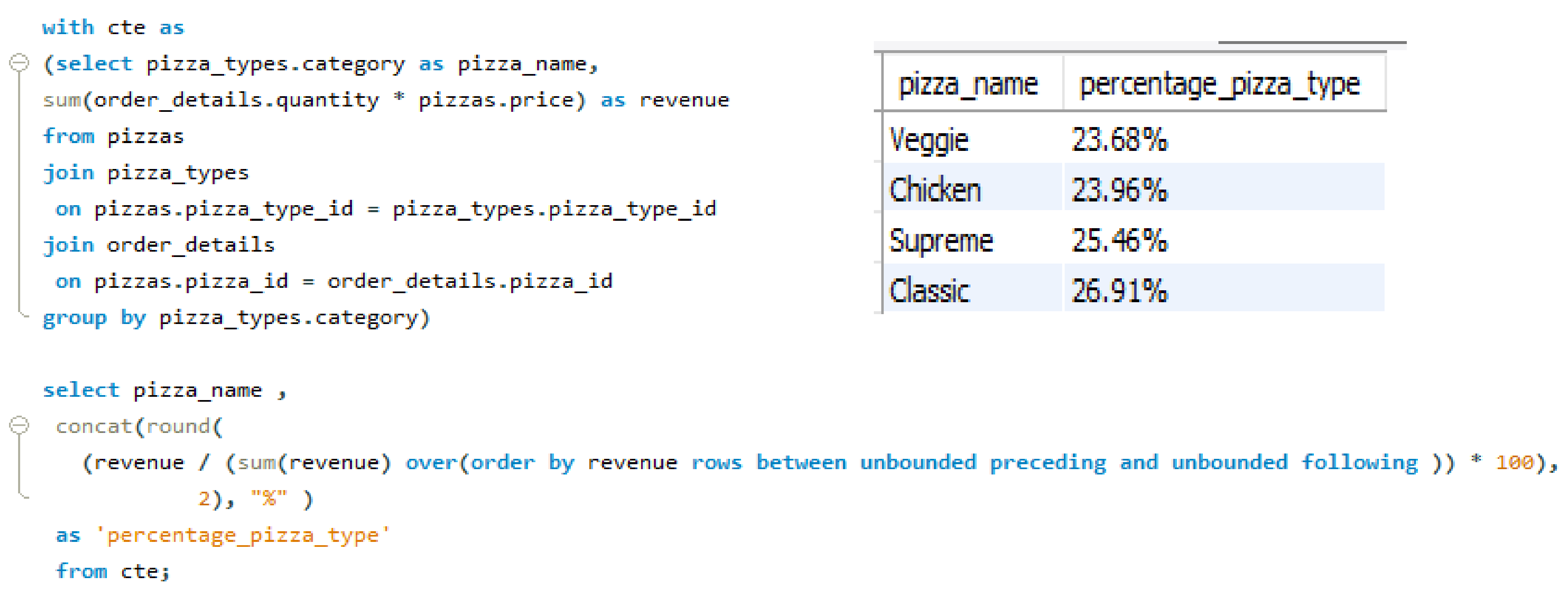
name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



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

Query

OUTPUT



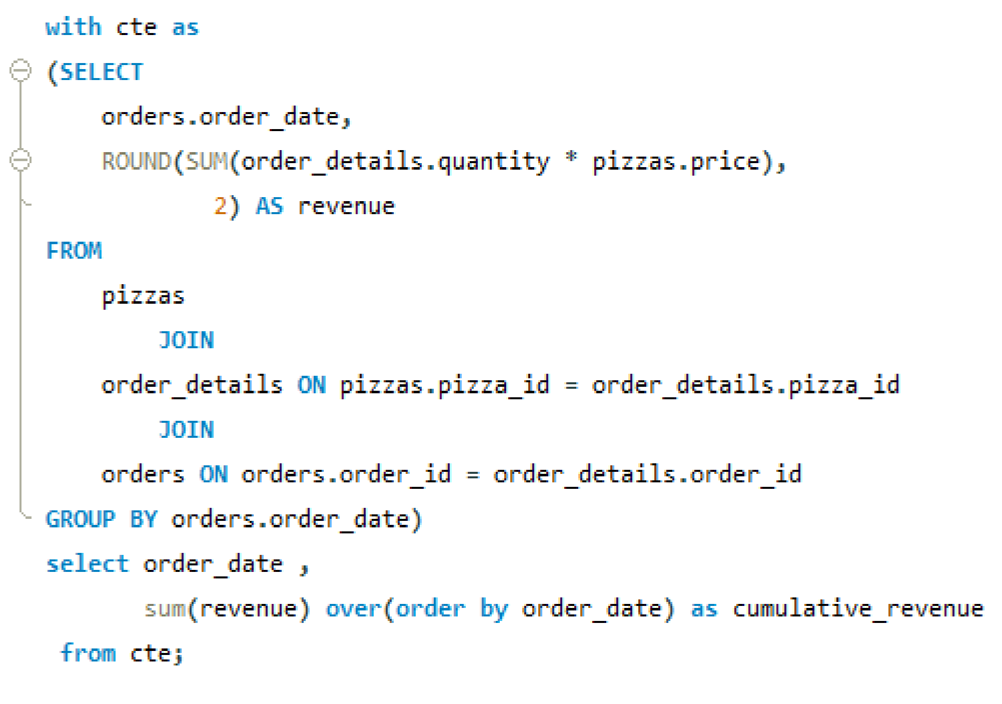
```
with cte as
(select pizza_types.category as pizza_name,
sum(order_details.quantity * pizzas.price) as revenue
from pizzas
join pizza_types
 on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
 on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category)

select pizza_name ,
 concat(round(
   (revenue / (sum(revenue) over(order by revenue rows between unbounded preceding and unbounded following )) * 100),
            2), "%" )
 as 'percentage_pizza_type'
 from cte;
```

pizza_name	percentage_pizza_type
Veggie	23.68%
Chicken	23.96%
Supreme	25.46%
Classic	26.91%

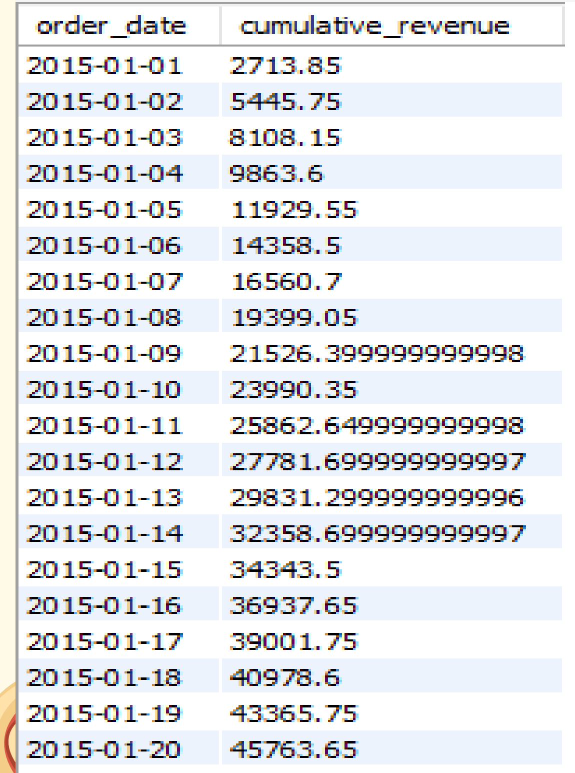
12 Analyze the cumulative revenue generated over time.

Query

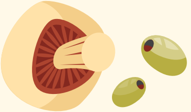


```
with cte as
(SELECT
    orders.order_date,
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS revenue
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY orders.order_date)
select order_date ,
       sum(revenue) over(order by order_date) as cumulative_revenue
 from cte;
```

OUTPUT



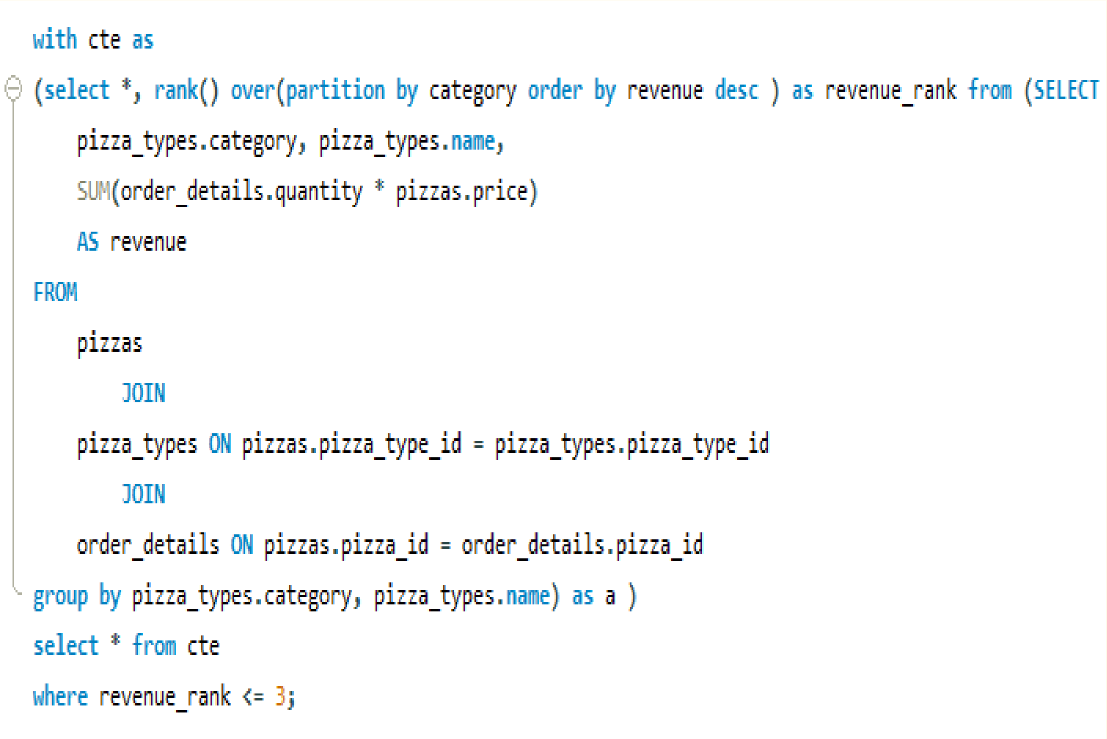
order_date	cumulative_revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65



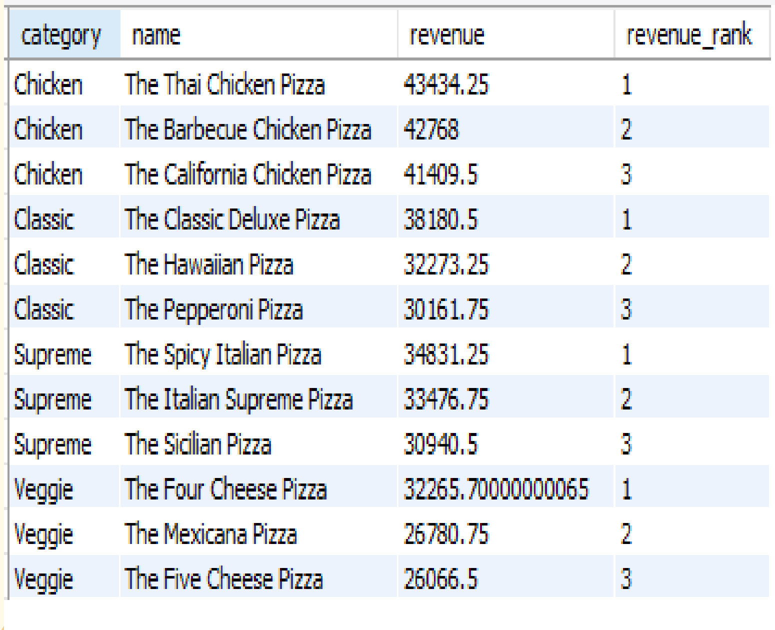
13 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Query

OUTPUT

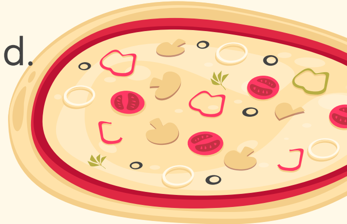


```
with cte as
(select *, rank() over(partition by category order by revenue desc ) as revenue_rank from (SELECT
    pizza_types.category, pizza_types.name,
    SUM(order_details.quantity * pizzas.price)
    AS revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) as a )
select * from cte
where revenue_rank <= 3;
```

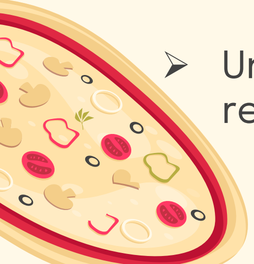


category	name	revenue	revenue_rank
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3

08 Insights



- The analysis revealed that a **Large(L) size pizza** is most commonly ordered.
- The **Thai Chicken Pizza($43434.25)** , **Barbecue Chiken Pizza($42768)** and **California Chiken pizza($41409.5)** generate the highest revenue.
- The most ordered pizza types based on quantities are the **classic delux pizza(2453)** , the **barbecue chicken pizza(2432)** and the **Hawaiian pizza(2422).**
- The highest-priced pizza is the **Greek Pizza ($35.95)** is contributing significantly to the revenue.
- The average number of pizzas ordered per day is **138** .
- Cumulative revenue trends provide a long-term view of performance.
- Understanding the percentage contribution of each pizza type to total revenue helps in identifying customer preferences.



09 Conclusion

- The analysis of pizza sales data reveals valuable insights for **optimizing business operations** and driving **sales growth**.

- By leveraging MySQL queries we've identified total orders and revenue, as well as **customer preferences and temporal patterns** in ordering behavior.

- These findings inform actionable recommendations for menu optimization, **pricing strategies**, and resource allocation.

- Moving forward, continuous monitoring and adaptation based on data-driven insights will be crucial for sustaining competitive advantage and **achieving long-term success**.

THANKS!

Please Like and Follow me

- GitHub : https://github.com/aryan98875
- LinkedIn : aryan98875
- Gmail : aryan98875@gmail.com