Pizza Sales Data Analysis with SQL

Business Insights from Real-World Dataset

ABOUT ME

Hello , I'm **Aditya Kapse** — an aspiring Data Analyst with a solid academic foundation in tools like Excel, SQL, Power BI, Tableau, and Python. I’m driven by a passion for transforming raw data into actionable insights and solving real-world business challenges through analytical thinking and creative visualization.

TABLE OF CONTENTS

01 DATASET OVERVIEW

Description of the Pizza Sales dataset used for the project

02 BUSINESS QUESTIONS

Core business problems analyzed to uncover sales, revenue, and customer trends.

03 SQL QUERIES WITH OUTPUTS

Complete list of SQL queries with results and derived business insights.

04 KEY FINDINGS (INSIGHTS)

Actionable insights on revenue, top pizzas, order patterns, and customer preferences.

01

DATASET OVERVIEW

HERE’s A BRIEF INTRODUCTION TO THE PIZZA SALES DATASET

DOWNLOAD FILE

TABLE STRUCTURES

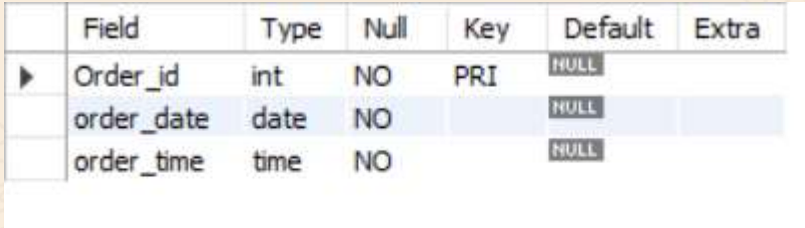
1.orders :

Fields :

- **order_id** : Unique order identifier
- **order_date** : Date when the order was placed
- **time** : Time when the order was placed

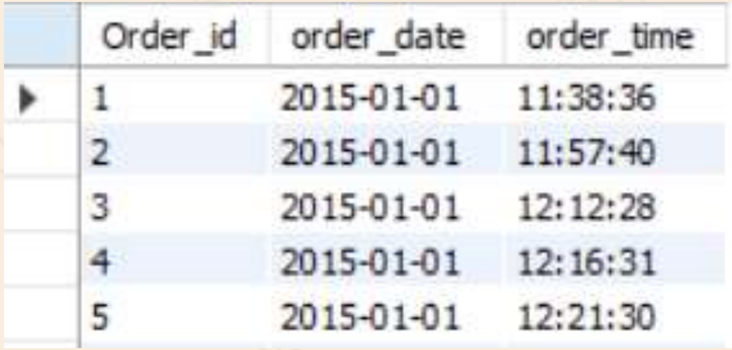
Purpose: Stores information about each customer order (when it happened).

TABLE STRUCTURE :



	Field	Type	Null	Key	Default	Extra
▶	Order_id	int	NO	PRI	NULL	
	order_date	date	NO		NULL	
	order_time	time	NO		NULL	

SAMPLE DATA :



	Order_id	order_date	order_time
▶	1	2015-01-01	11:38:36
	2	2015-01-01	11:57:40
	3	2015-01-01	12:12:28
	4	2015-01-01	12:16:31
	5	2015-01-01	12:21:30

DOWNLOAD FILE

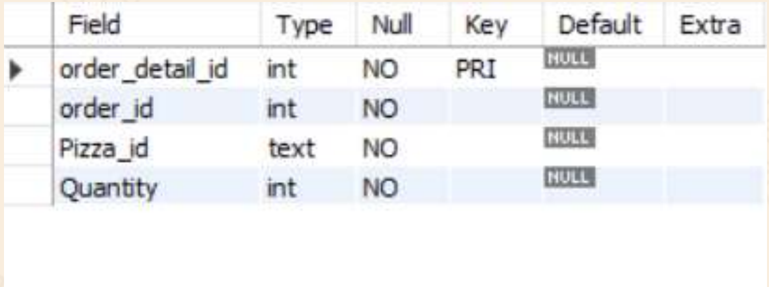
2.order_details :

Fields :

- **order_detail_id** : Unique ID for each row (line item)
- **order_id** : Links to Orders table
- **pizza_id** : Links to Pizza Table
- **quantity** : Number of pizzas ordered in that order

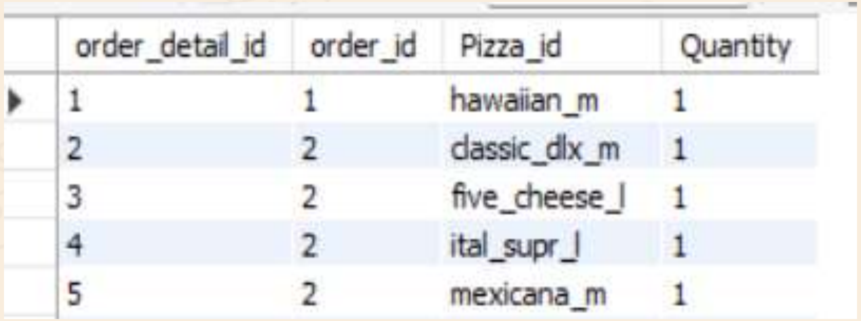
Purpose: Line-level details of each order (which pizzas and how many).

TABLE STRUCTURE :



Field	Type	Null	Key	Default	Extra
order_detail_id	int	NO	PRI	NULL	
order_id	int	NO		NULL	
Pizza_id	text	NO		NULL	
Quantity	int	NO		NULL	

SAMPLE DATA :



order_detail_id	order_id	Pizza_id	Quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1

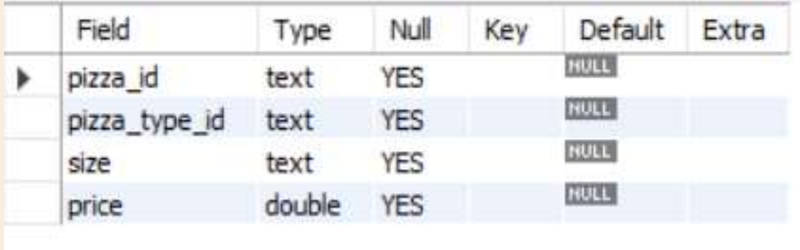
3.Pizzas:

Fields :

- **pizza_id :**Unique pizza identifier
- **pizza_type_id :** Links to pizza_types table
- **Size :** Pizza size (S, M, L, XL, XXL)
- **Price :** Price of that pizza

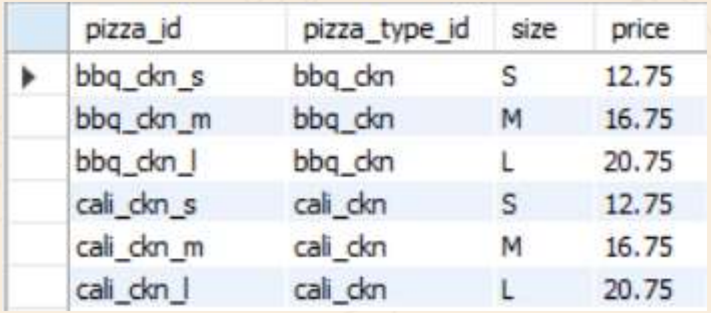
Purpose: Defines each pizza available (variation by size and price).

TABLE STRUCTURE :



Field	Type	Null	Key	Default	Extra
pizza_id	text	YES		NULL	
pizza_type_id	text	YES		NULL	
size	text	YES		NULL	
price	double	YES		NULL	

SAMPLE DATA :



pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75
bbq_ckn_l	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75

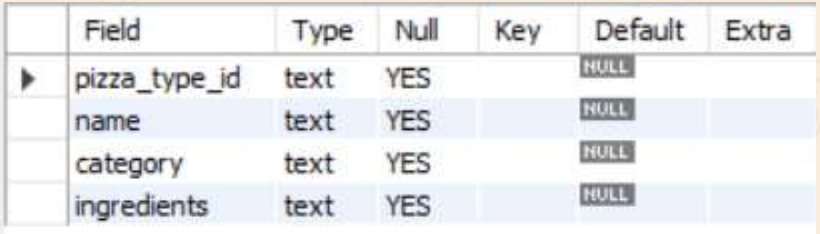
3.Pizza_types:

Fields :

- **pizza_type_id** : Unique identifier for each pizza recipe/type
- **name** : Name of the pizza (e.g., Pepperoni Pizza)
- **category** : Category (Classic, Supreme, Chicken, Veggie)
- **ingredients** : List of ingredients in the pizza

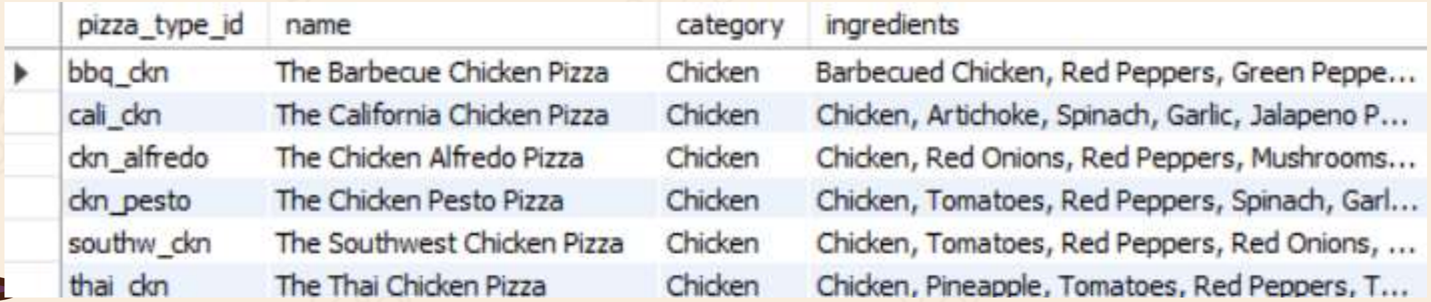
Purpose: Defines each pizza available (variation by size and price).

TABLE STRUCTURE :



	Field	Type	Null	Key	Default	Extra
▶	pizza_type_id	text	YES		NULL	
	name	text	YES		NULL	
	category	text	YES		NULL	
	ingredients	text	YES		NULL	

SAMPLE DATA :



	pizza_type_id	name	category	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Peppe...
	cali_ckn	The California Chicken Pizza	Chicken	Chicken, Artichoke, Spinach, Garlic, Jalapeno P...
	ckn_alfredo	The Chicken Alfredo Pizza	Chicken	Chicken, Red Onions, Red Peppers, Mushrooms...
	ckn_pesto	The Chicken Pesto Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Spinach, Garl...
	southw_ckn	The Southwest Chicken Pizza	Chicken	Chicken, Tomatoes, Red Peppers, Red Onions, ...
	thai_ckn	The Thai Chicken Pizza	Chicken	Chicken, Pineapple, Tomatoes, Red Peppers, T...

02

BUSINESS QUESTIONS

Here Are the Business Questions solved in this project

BASIC QUESTIONS

- ◆ Total number of orders placed
- ◆ Total revenue generated
- ◆ Highest-priced pizza
- ◆ Most common pizza size ordered
- ◆ Top 5 most ordered pizzas (by quantity)
- ◆ Find Total Number of Pizzas Sold
- ◆ Find out which day of week makes the highest revenue

INTERMEDIATE QUESTIONS

◆ Total quantity of each pizza category ordered
◆ Orders distribution by hour of the day
◆ Category-wise distribution of pizzas
◆ Average number of pizzas ordered per day
◆ Top 3 pizzas by revenue
◆ Average Order value
◆ Find Out the Slow movers → pizzas with least sales

ADVANCED QUESTIONS

- ◆ Percent (%) contribution of each pizza type to total revenue

- ◆ Cumulative revenue over time

- ◆ Top 3 pizzas by revenue in each category

03

SQL QUERIES WITH OUTPUTS

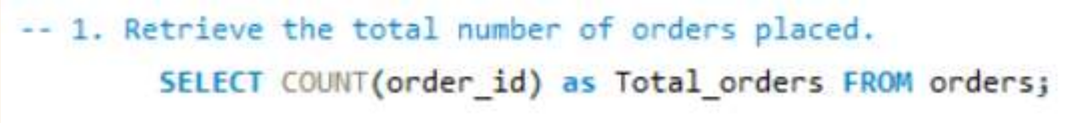
Here Are the Queries used along with the outputs

01

BASIC QUESTIONS

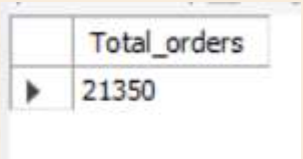
QUESTION: Total number of orders placed

QUERY :



```sql
-- 1. Retrieve the total number of orders placed.
        SELECT COUNT(order_id) as Total_orders FROM orders;
```

OUTPUT :



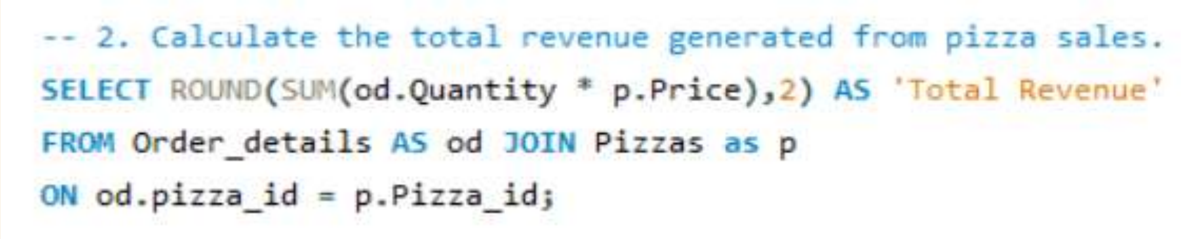
	Total_orders
▶	21350

CONCLUSION : Total 21k + orders are placed from the pizza store

02 BASIC QUESTIONS

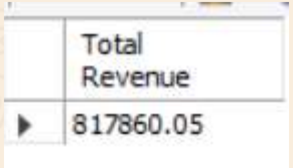
QUESTION: Total Revenue Generated

QUERY :



```sql
-- 2. Calculate the total revenue generated from pizza sales.
SELECT ROUND(SUM(od.Quantity * p.Price),2) AS 'Total Revenue'
FROM Order_details AS od JOIN Pizzas as p
ON od.pizza_id = p.Pizza_id;
```

OUTPUT :



	Total Revenue
▶	817860.05

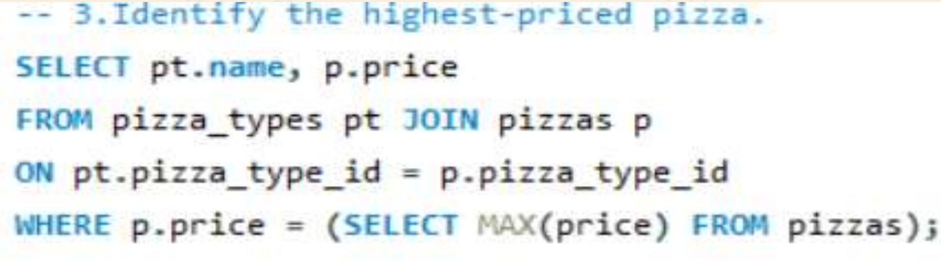
CONCLUSION : Around $817K+ revenue is generated

03

BASIC QUESTIONS

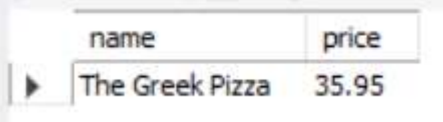
QUESTION: Find the Highest-priced pizza

QUERY :



```
-- 3.Identify the highest-priced pizza.
SELECT pt.name, p.price
FROM pizza_types pt JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id
WHERE p.price = (SELECT MAX(price) FROM pizzas);
```

OUTPUT :



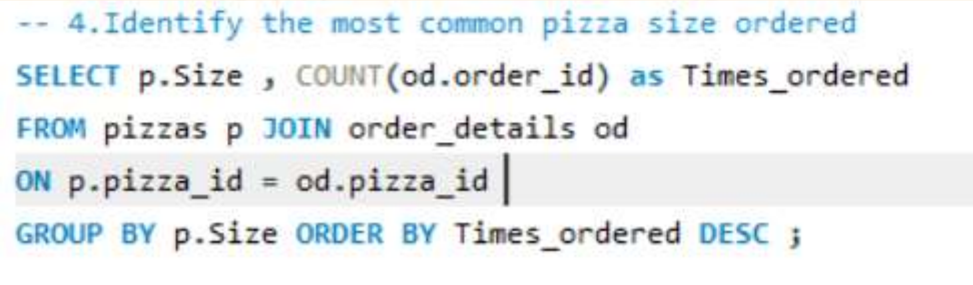
name	price
The Greek Pizza	35.95

CONCLUSION : ‘The Greek Pizza’ is the Highest Priced Pizza which costs around $36

04 BASIC QUESTIONS

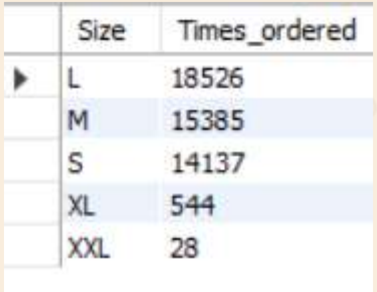
QUESTION: Find Most common pizza size ordered

QUERY :



```sql
-- 4.Identify the most common pizza size ordered
SELECT p.Size , COUNT(od.order_id) as Times_ordered
FROM pizzas p JOIN order_details od
ON p.pizza_id = od.pizza_id
GROUP BY p.Size ORDER BY Times_ordered DESC ;
```

OUTPUT :



Size	Times_ordered
L	18526
M	15385
S	14137
XL	544
XXL	28

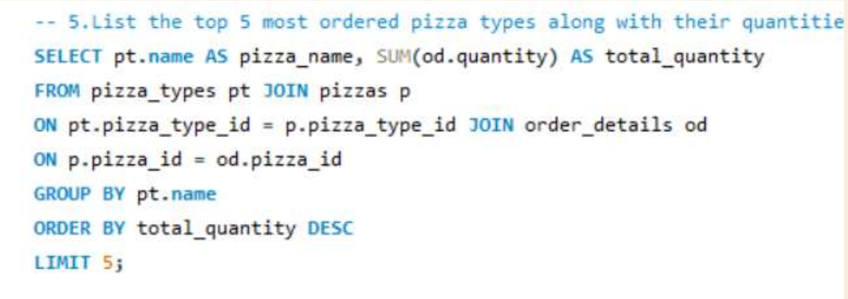
CONCLUSION : ‘L’ is the Highest Ordered Pizza Size which is ordered around 18K + times followed by ‘M’ and ‘S’

05

BASIC QUESTIONS

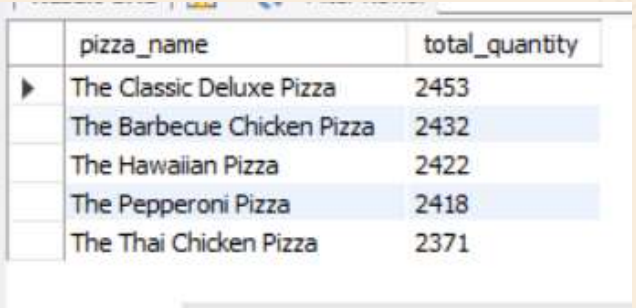
QUESTION: Find Top 5 most ordered pizzas (by quantity)

QUERY :



```
-- 5.List the top 5 most ordered pizza types along with their quantitie
SELECT pt.name AS pizza_name, SUM(od.quantity) AS total_quantity
FROM pizza_types pt JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id JOIN order_details od
ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

OUTPUT :



pizza_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

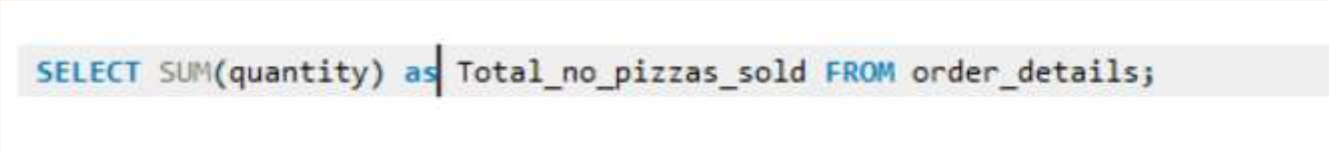
CONCLUSION : Given Table Shows top 5 most ordered pizzas with total
Quantities ordered

06

BASIC QUESTIONS

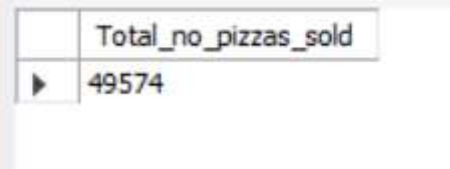
QUESTION: Find Total Number of Pizzas Sold

QUERY :



```
SELECT SUM(quantity) as Total_no_pizzas_sold FROM order_details;
```

OUTPUT :



	Total_no_pizzas_sold
▶	49574

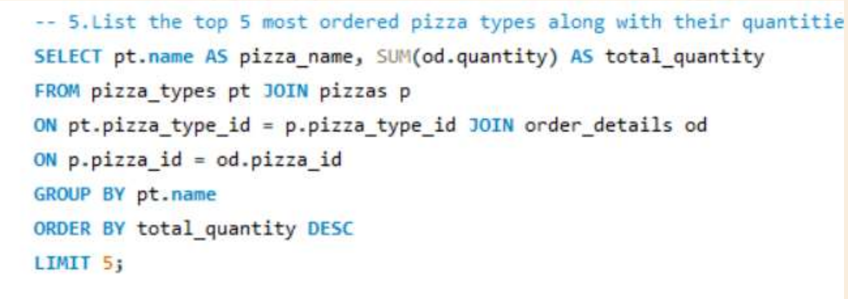
CONCLUSION : Almost 50K pizzas have been sold

05

BASIC QUESTIONS

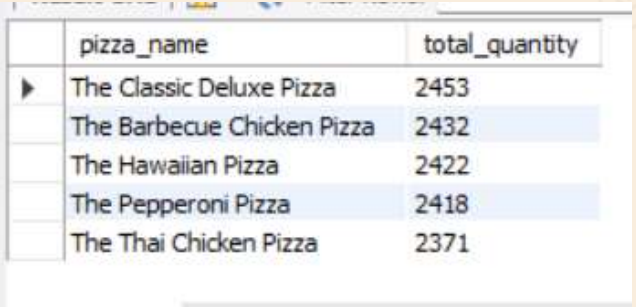
QUESTION: Find Top 5 most ordered pizzas (by quantity)

QUERY :



```
-- 5.List the top 5 most ordered pizza types along with their quantitie
SELECT pt.name AS pizza_name, SUM(od.quantity) AS total_quantity
FROM pizza_types pt JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id JOIN order_details od
ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY total_quantity DESC
LIMIT 5;
```

OUTPUT :



pizza_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

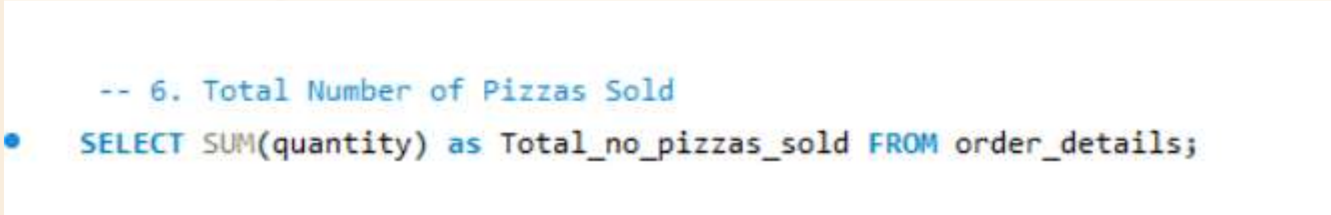
CONCLUSION : Given Table Shows top 5 most ordered pizzas with total Quantities ordered

06

BASIC QUESTIONS

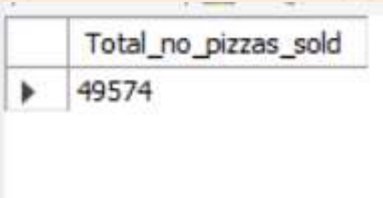
QUESTION: Find Total Number of Pizzas Sold

QUERY
:



```
-- 6. Total Number of Pizzas Sold
SELECT SUM(quantity) as Total_no_pizzas_sold FROM order_details;
```

OUTPU
T :



```
  Total_no_pizzas_sold
▶ 49574
```

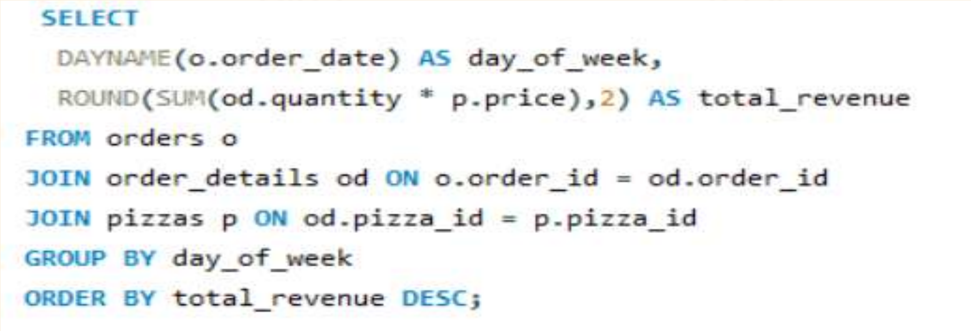
CONCLUSION : Around 50K Pizzas have been sold

07

BASIC QUESTIONS

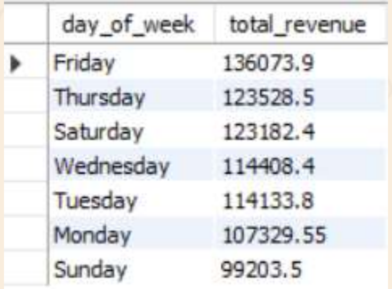
QUESTION: Find out which day of week makes the highest revenue

QUERY :



```
 SELECT
  DAYNAME(o.order_date) AS day_of_week,
  ROUND(SUM(od.quantity * p.price),2) AS total_revenue
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN pizzas p ON od.pizza_id = p.pizza_id
GROUP BY day_of_week
ORDER BY total_revenue DESC;
```

OUTPUT :



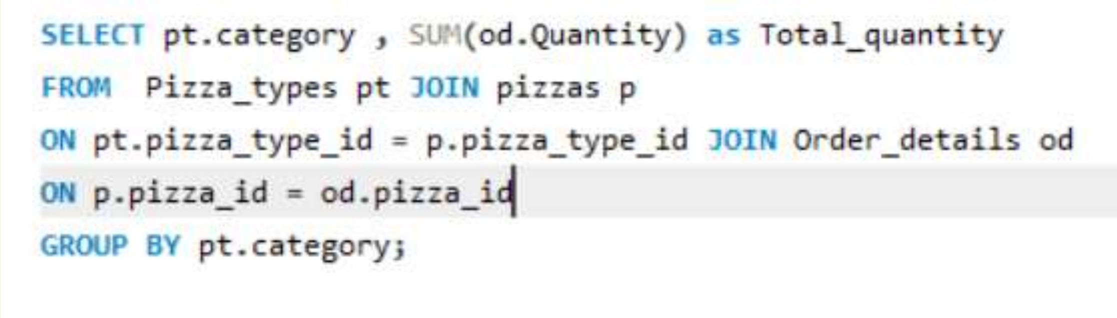
day_of_week	total_revenue
Friday	136073.9
Thursday	123528.5
Saturday	123182.4
Wednesday	114408.4
Tuesday	114133.8
Monday	107329.55
Sunday	99203.5

CONCLUSION : Around 50K Pizzas have been sold

01 INTERMEDIATE QUESTIONS

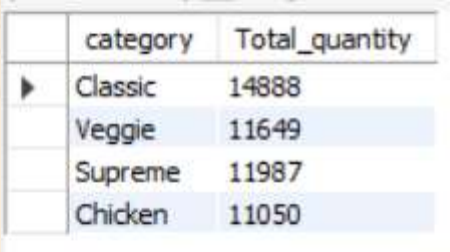
QUESTION: find the total quantity of each pizza category ordered

QUERY :



```
SELECT pt.category , SUM(od.Quantity) as Total_quantity
FROM  Pizza_types pt JOIN pizzas p
ON pt.pizza_type_id = p.pizza_type_id JOIN Order_details od
ON p.pizza_id = od.pizza_id
GROUP BY pt.category;
```

OUTPU T :



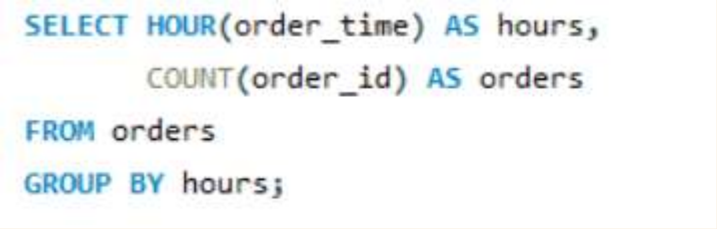
	category	Total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

CONCLUSION : Classic Pizzas are Most demanded followed by Veggie and Supreme

02 INTERMEDIATE QUESTIONS

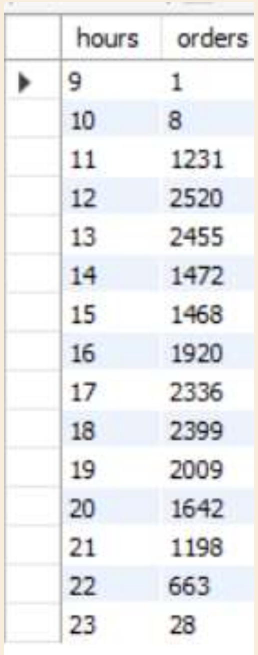
QUESTION: Determine the distribution of orders by hour of the day

QUERY :



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

OUTPUT :



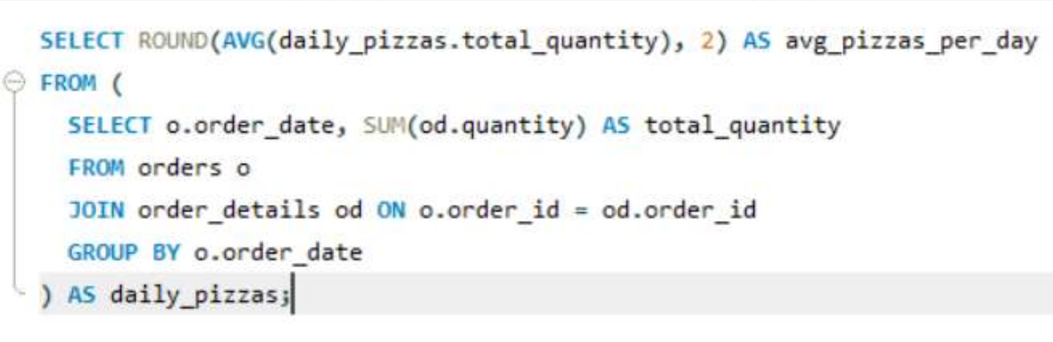
hours	orders
9	1
10	8
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

CONCLUSION : There’s maximum engagement during midday to early evening, with a clear peak around lunchtime

03 INTERMEDIATE QUESTIONS

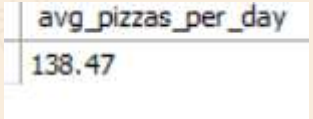
QUESTION: calculate the average number of pizzas ordered per day

QUERY :



```
SELECT ROUND(AVG(daily_pizzas.total_quantity), 2) AS avg_pizzas_per_day
FROM (
  SELECT o.order_date, SUM(od.quantity) AS total_quantity
  FROM orders o
  JOIN order_details od ON o.order_id = od.order_id
  GROUP BY o.order_date
) AS daily_pizzas;
```

OUTPU T :



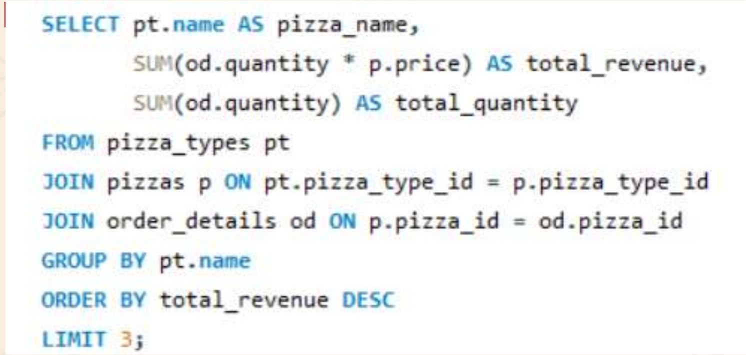
```
avg_pizzas_per_day
138.47
```

CONCLUSION : The Business consistently sell around 138 pizzas a day

04 INTERMEDIATE QUESTIONS

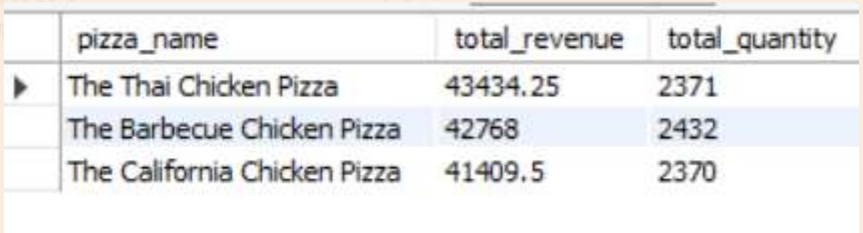
QUESTION: Determine the top 3 most ordered pizza types based on reve

QUERY :



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

OUTPU
T :



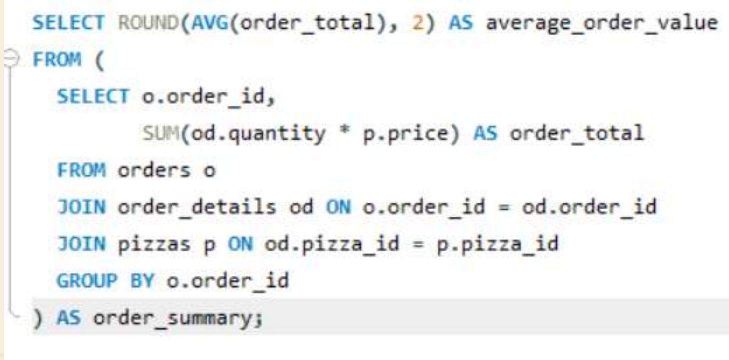
pizza_name	total_revenue	total_quantity
The Thai Chicken Pizza	43434.25	2371
The Barbecue Chicken Pizza	42768	2432
The California Chicken Pizza	41409.5	2370

CONCLUSION : The Thai Chicken Pizza earns the highest revenue, while Barbecue Chicken Pizza leads in quantity sold—both are top performers.

05 INTERMEDIATE QUESTIONS

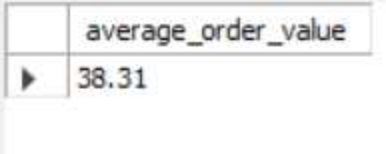
QUESTION: Determine the Average Order value

QUERY
:



```
SELECT ROUND(AVG(order_total), 2) AS average_order_value
FROM (
  SELECT o.order_id,
         SUM(od.quantity * p.price) AS order_total
  FROM orders o
  JOIN order_details od ON o.order_id = od.order_id
  JOIN pizzas p ON od.pizza_id = p.pizza_id
  GROUP BY o.order_id
) AS order_summary;
```

OUTPU
T :



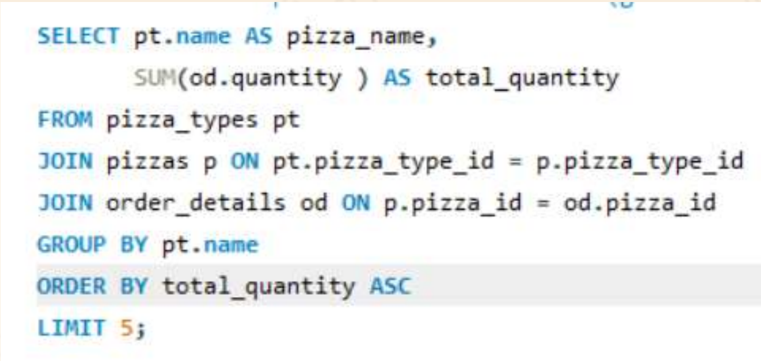
	average_order_value
▶	38.31

CONCLUSION : Each customer typically spends around $38 per transaction.

06 INTERMEDIATE QUESTIONS

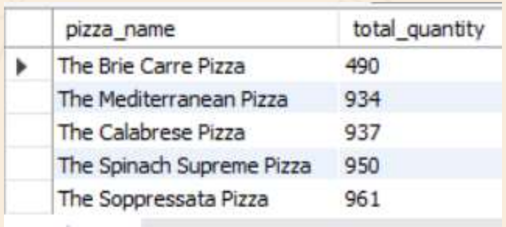
QUESTION: Determine the Slow movers → pizzas with least sales

QUERY :



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

OUTPU T :



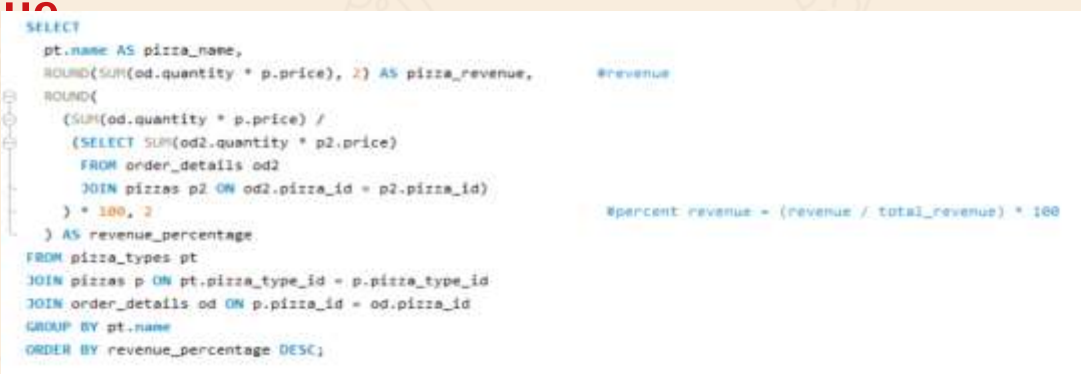
pizza_name	total_quantity
The Brie Carre Pizza	490
The Mediterranean Pizza	934
The Calabrese Pizza	937
The Spinach Supreme Pizza	950
The Soppressata Pizza	961

CONCLUSION : The given list shows worst performing pizzas .

01 ADVANCED QUESTIONS

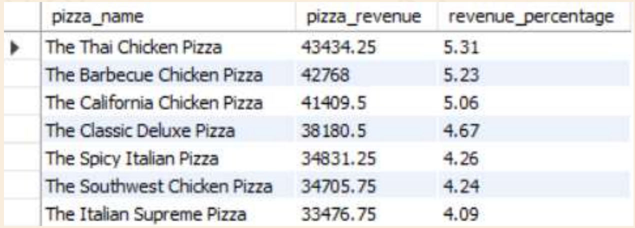
QUESTION: Calculate the percentage contribution of each pizza type to total revenue

QUERY :



```
SELECT
  pt.name AS pizza_name,
  ROUND(SUM(od.quantity * p.price), 2) AS pizza_revenue,        #revenue
  ROUND(
    (SUM(od.quantity * p.price) /
     (SELECT SUM(od2.quantity * p2.price)
      FROM order_details od2
      JOIN pizzas p2 ON od2.pizza_id = p2.pizza_id)
    ) * 100, 2                                                   #percent revenue = (revenue / total_revenue) * 100
  ) AS revenue_percentage
FROM pizza_types pt
JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
JOIN order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY revenue_percentage DESC;
```

OUTPUT :



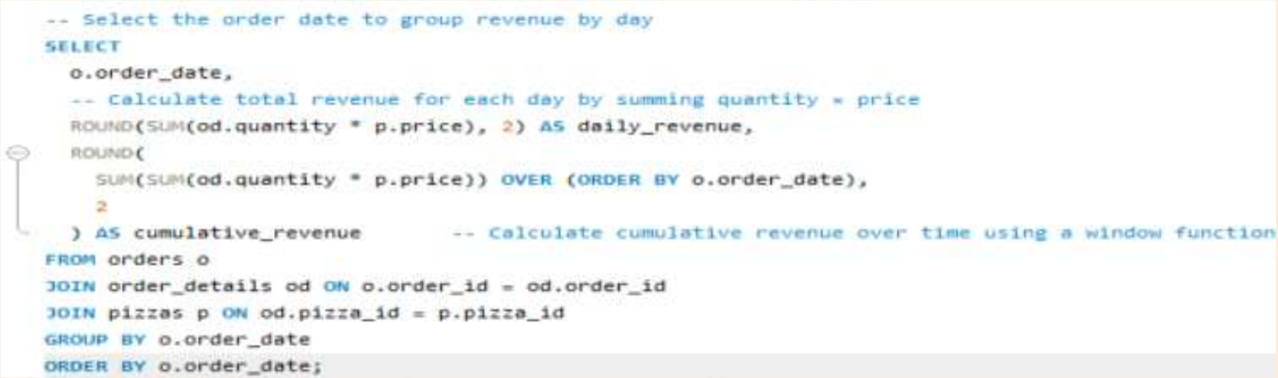
pizza_name	pizza_revenue	revenue_percentage
The Thai Chicken Pizza	43434.25	5.31
The Barbecue Chicken Pizza	42768	5.23
The California Chicken Pizza	41409.5	5.06
The Classic Deluxe Pizza	38180.5	4.67
The Spicy Italian Pizza	34831.25	4.26
The Southwest Chicken Pizza	34705.75	4.24
The Italian Supreme Pizza	33476.75	4.09

CONCLUSION : The top 4-5 pizzas make around 25 % of the Revenue

02 ADVANCED QUESTIONS

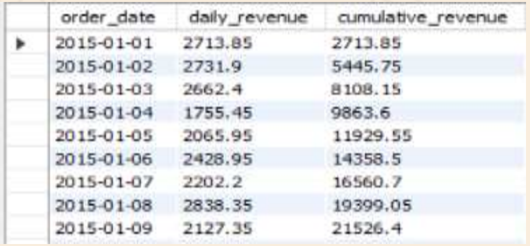
QUESTION: Analyze the cumulative revenue generated over time..

QUERY :



```
-- Select the order date to group revenue by day
SELECT
  o.order_date,
  -- Calculate total revenue for each day by summing quantity * price
  ROUND(SUM(od.quantity * p.price), 2) AS daily_revenue,
  ROUND(
    SUM(SUM(od.quantity * p.price)) OVER (ORDER BY o.order_date),
    2
  ) AS cumulative_revenue        -- Calculate cumulative revenue over time using a window function
FROM orders o
JOIN order_details od ON o.order_id = od.order_id
JOIN pizzas p ON od.pizza_id = p.pizza_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

OUTPUT :



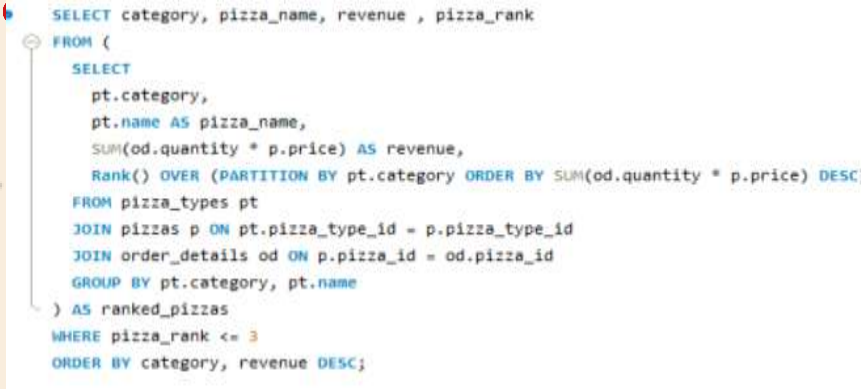
	order_date	daily_revenue	cumulative_revenue
▶	2015-01-01	2713.85	2713.85
	2015-01-02	2731.9	5445.75
	2015-01-03	2662.4	8108.15
	2015-01-04	1755.45	9863.6
	2015-01-05	2065.95	11929.55
	2015-01-06	2428.95	14358.5
	2015-01-07	2202.2	16560.7
	2015-01-08	2838.35	19399.05
	2015-01-09	2127.35	21526.4

CONCLUSION : The Given table shows cumulative revenue which helps to track total earnings over time

03 ADVANCED QUESTIONS

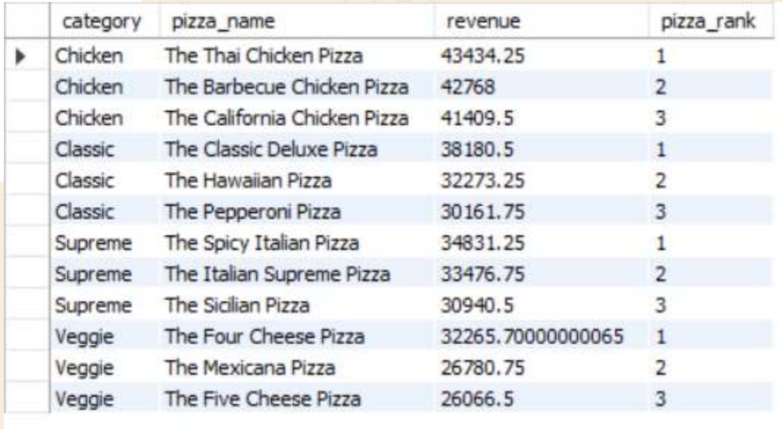
QUESTION: Determine the top 3 most ordered pizza types based on revenue for each pizza categ

QUERY :



```
SELECT category, pizza_name, revenue , pizza_rank
FROM (
  SELECT
    pt.category,
    pt.name AS pizza_name,
    SUM(od.quantity * p.price) AS revenue,
    Rank() OVER (PARTITION BY pt.category ORDER BY SUM(od.quantity * p.price) DESC) AS pizza_rank
  FROM pizza_types pt
  JOIN pizzas p ON pt.pizza_type_id = p.pizza_type_id
  JOIN order_details od ON p.pizza_id = od.pizza_id
  GROUP BY pt.category, pt.name
) AS ranked_pizzas
WHERE pizza_rank <= 3
ORDER BY category, revenue DESC;
```

OUTPU T :



category	pizza_name	revenue	pizza_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

CONCLUSION : The given table shows top 3 most orders Pizza types based on Revenue for each pizza category

04

KEY INSIGHTS

HERE ARE SOME KEY INSIGHTS FROM THE ANALYSIS WE HAVE PERFORMED

REVENUE AND SALES VOLUME

- 💰 Total Revenue: Over $817K generated from pizza sales.

- 🍕 Total Pizzas Sold: Nearly 50,000 pizzas sold.

- 📊 Average Daily Sales: ~138 pizzas/day, with an average order value of $38.31.

CUSTOMER BEHAVIOR & TIMING

- 🕒 Peak Hours: Highest order volume between 11 AM and 1 PM—ideal for staffing and promotional targeting.
- 📅 Top Day: Fridays drive the highest revenue, making them prime for special offers or combo deals.

PRODUCT PERFORMANCE

🏆 Top-Selling Pizzas:

❖ - The Thai Chicken Pizza leads in revenue.
❖ - Barbecue Chicken Pizza leads in quantity sold.
❖ - Classic Deluxe Pizza is the most ordered overall.

Slow Movers Pizzas:

❖ Pizzas like Brie Carre and Five Cheese have low sales—candidates for revamp or removal.

CATEGORY INSIGHTS

- ❖ □ Most Popular Category: Classic pizzas dominate in quantity sold.

- ❖ - 🎁 Size Preference: Large (L) pizzas are most ordered—important for inventory and pricing strategy.

REVENUE DISTRIBUTION

- 🎯 Top 5 Pizzas contribute ~25% of total revenue—focus marketing and bundling efforts here.

- 📈 Cumulative Revenue Tracking helps monitor growth and forecast future performance.

THANKS!

Aditya Rajendra Kapse

- Aspiring Data Analyst

Project Link : Pizza Sales Analysis WIth MySQL