



Indian Institute Of Information Technology , Lucknow

# Pizza sales Analysis using SQL

Presenter: Shubham Singh (LCI2021036 , IIIT Lucknow )



# INTRODUCTION

In this project, comprehensive sales analysis of pizza has been conducted using Structured Query Language (SQL). The aim is to uncover valuable insights into our sales performance, customer preferences, and revenue trends. This analysis is pivotal for understanding our business dynamics and making data-driven decisions.

## Methodology:

Utilizing SQL queries, various analyses have been performed, and the results are presented in a tabular format. These analyses provide a detailed overview of the sales data, highlighting key metrics and trends.

# EXECUTIVE SUMMARY

- Total Orders: Determine the total number of orders placed.
- Revenue: Calculate the total revenue generated from pizza sales.
- Pricing: Identify the highest-priced pizza.
- Customer Preferences:
  - Most common pizza size ordered.
  - Top 5 most ordered pizza types along with their quantities.
- Category Insights:
  - Join necessary tables to find the total quantity of each pizza category ordered.
- Temporal Analysis:
  - Determine the distribution of orders by hour of the day.
  - Group orders by date and calculate the average number of pizzas ordered per day.
- Revenue Analysis:
  - Determine the top 3 most ordered pizza types based on revenue.
  - Calculate the percentage contribution of each pizza type to total revenue.
  - Analyze the cumulative revenue generated over time.
  - Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# Database name : Pizza\_Sales\_analysis

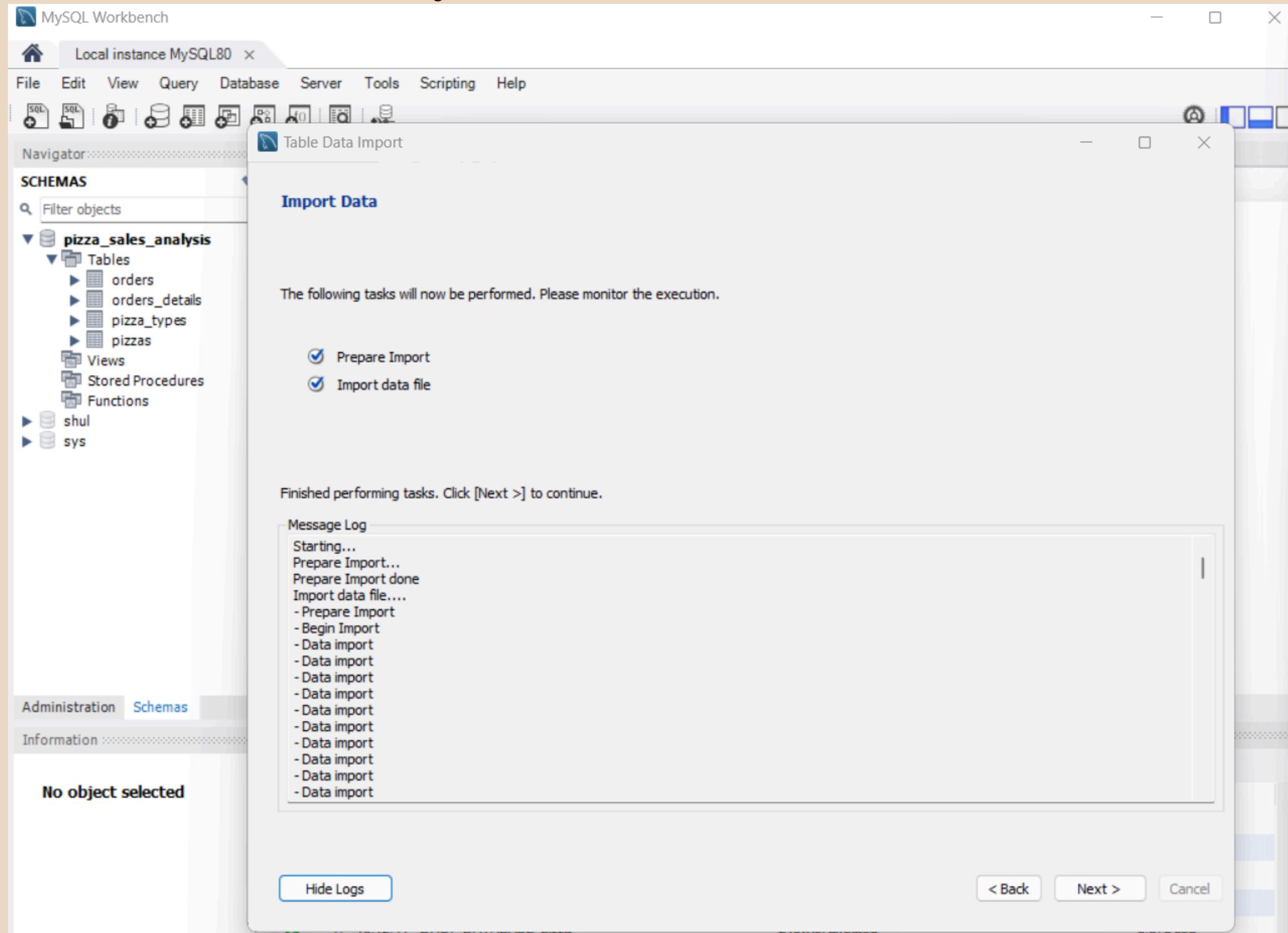
```
1 -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2 • create database Pizza_Sales_analysis;
```

## Tables Used in the database:

- pizzas
- pizza\_types
- orders
- order\_types

# Data insertion in tables:

- Data were imported from: [Link](#)



# Description of pizzas table (dataset):

- Number of records = 96.

Information

**Table:** pizzas

**Columns:**

pizza_id	text
pizza_type_id	text
size	text
price	double

```
1 -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2 -- pizzas table  
3 • SELECT * FROM pizza_sales_analysis.pizzas;
```

	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
	cfn_alfredo_s	cfn_alfredo	S	12.75
	cfn_alfredo_m	cfn_alfredo	M	16.75
	cfn_alfredo_l	cfn_alfredo	L	20.75
	cfn_pesto_s	cfn_pesto	S	12.75
	cfn_pesto_m	cfn_pesto	M	16.75
	cfn_pesto_l	cfn_pesto	L	20.75
	southw_ckn_s	southw_ckn	S	12.75
	southw_ckn_m	southw_ckn	M	16.75
	southw_ckn_l	southw_ckn	L	20.75
	thai_ckn_s	thai_ckn	S	12.75
	thai_ckn_m	thai_ckn	M	16.75
	thai_ckn_l	thai_ckn	L	20.75
	big_meat_s	big_meat	S	12
	big_meat_m	big_meat	M	16
	big_meat_l	big_meat	L	20.5
	dassic_dlx_s	dassic_dlx	S	12
	dassic_dlx_m	dassic_dlx	M	16
	dassic_dlx_l	dassic_dlx	L	20.5
	hawaiian_s	hawaiian	S	10.5
	hawaiian_m	hawaiian	M	13.25

# Description of pizza\_types table (dataset):

- Number of records = 32.

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :  
2  
3 • SELECT * FROM pizza_sales_analysis.pizza_types;
```

	pizza_type_id	name	category	ingredients
▶	bbq_ckn	The Barbecue Chicken Pizza	Chicken	Barbecued Chicken, Red Peppers, Green Pepp...
	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...
	big_meat	The Big Meat Pizza	Classic	Bacon, Pepperoni, Italian Sausage, Chorizo Sau...
	classic_dlx	The Classic Deluxe Pizza	Classic	Pepperoni, Mushrooms, Red Onions, Red Peppe...
	hawaiian	The Hawaiian Pizza	Classic	Sliced Ham, Pineapple, Mozzarella Cheese
	ital_cpdlo	The Italian Capocollo Pizza	Classic	Capocollo, Red Peppers, Tomatoes, Goat Chee...
	napolitana	The Napolitana Pizza	Classic	Tomatoes, Anchovies, Green Olives, Red Onion...
	pep_msh_pep	The Pepperoni, Mushroom, ...	Classic	Pepperoni, Mushrooms, Green Peppers
	pepperoni	The Pepperoni Pizza	Classic	Mozzarella Cheese, Pepperoni
	the_greek	The Greek Pizza	Classic	Kalamata Olives, Feta Cheese, Tomatoes, Garli...
	brie_carre	The Brie Carre Pizza	Supreme	Brie Carre Cheese, Prosciutto, Caramelized Oni...
	calabrese	The Calabrese Pizza	Supreme	'Nduja Salami, Pancetta, Tomatoes, Red Onions...
	ital_supr	The Italian Supreme Pizza	Supreme	Calabrese Salami, Capocollo, Tomatoes, Red O...
	peppr_salami	The Pepper Salami Pizza	Supreme	Genoa Salami, Capocollo, Pepperoni, Tomatoes,...
	prsc_argla	The Prosciutto and Arugula ...	Supreme	Prosciutto di San Daniele, Arugula, Mozzarella C...

**Table: pizza\_types**

## Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

# Description of orders table (dataset):

- Number of records = 21350.

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :  
2 • CREATE TABLE orders (  
3     order_id INT NOT NULL,  
4     order_date DATE NOT NULL,  
5     order_time TIME NOT NULL,  
6     PRIMARY KEY (order_id)  
7 );  
8 • SELECT * FROM pizza_sales_analysis.orders;
```

## Table: orders

### Columns:

<b>order_id</b>	int PK
<b>order_date</b>	date
<b>order_time</b>	time

	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
	6	2015-01-01	12:29:36
	7	2015-01-01	12:50:37
	8	2015-01-01	12:51:37
	9	2015-01-01	12:52:01
	10	2015-01-01	13:00:15
	11	2015-01-01	13:02:59
	12	2015-01-01	13:04:41
	13	2015-01-01	13:11:55
	14	2015-01-01	13:14:19

# Description of order\_details table (dataset):

- Number of records = 48620.

```
1 -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2 • CREATE TABLE order_details (  
3     order_details_id INT NOT NULL,  
4     order_id INT NOT NULL,  
5     pizza_id TEXT NOT NULL,  
6     quantity INT NOT NULL,  
7     PRIMARY KEY (order_details_id)  
8 );  
9 • SELECT * FROM pizza_sales_analysis.order_details;
```

Table: order\_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

	order_details_id	order_id	pizza_id	quantity
1	1	1	hawaiian_m	1
2	2	2	classic_dlx_m	1
3	2	2	five_cheese_l	1
4	2	2	ital_supr_l	1
5	2	2	mexicana_m	1
6	2	2	thai_dkn_l	1
7	3	3	ital_supr_m	1
8	3	3	prsc_argla_l	1
9	4	4	ital_supr_m	1
10	5	5	ital_supr_m	1
11	6	6	bbq_dkn_s	1

# Analysis

- 01** Retrieve the total number of orders placed.
- 02** Finding the total revenue generated from pizza sales.
- 03** Identifying the highest-priced pizza.
- 04** The most common pizza size ordered.
- 05** Top 5 most ordered pizza types along with their quantities.
- 06** Total quantity of each pizza category ordered
- 07** The distribution of orders by hour of the day.
- 08** The category-wise distribution of pizzas.

# Analysis

- 09** Group the orders by date and give the average number of pizzas ordered per day.
- 10** The top 3 most ordered pizza types based on revenue.
- 11** Percentage contribution of each pizza type to total revenue.
- 12** Analyse the cumulative revenue generated over time.
- 13** The top 3 most ordered pizza types based on revenue for each pizza category.

# The total number of orders placed

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :  
2 -- 1. Retrieving the total number of orders placed.  
3  
4 • SELECT  
5     COUNT(order_id) AS total_orders  
6 FROM  
7     orders;  
8
```

The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window containing the SQL query shown above. Below the code editor is a toolbar with several buttons: 'Result Grid' (selected), 'Filter Rows:', 'Export:', and 'Wrap Cell Content:'. The main area displays the results of the query in a table:

	total_orders
▶	21350

# Total revenue generated from pizza sales

```
1      -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :
2      -- 2. Calculate the total revenue generated from pizza sales.
3
4 •   SELECT
5     ROUND(SUM(order_details.quantity * pizzas.price),
6           2) AS total_sales
7
8   FROM
9     order_details
10    JOIN
11      pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

---

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

	total_sales
▶	817860.05

# Identifying the highest-priced pizza

```
1      -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2      -- 3. Identify the highest-priced pizza.  
3  
4 •  SELECT  
5      pizza_types.name, pizzas.price  
6  FROM  
7      pizza_types  
8      JOIN  
9      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
10     ORDER BY pizzas.price DESC  
11     LIMIT 1;
```

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

	name	price
▶	The Greek Pizza	35.95

# The most common pizza size ordered

```
1  -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :
2  -- 4. Identify the most common pizza size ordered.
3
4 • SELECT
5      pizzas.size,
6      COUNT(order_details.order_details_id) AS order_count
7  FROM pizzas
8  JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC;
```

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

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

# Top 5 most ordered pizza types along with their quantities

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :
2 -- 5. List the top 5 most ordered pizza types along with their quantities.
3
4 • SELECT
5     pizza_types.name, SUM(order_details.quantity) AS quantity FROM pizza_types
6     JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
7     JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
8     GROUP BY pizza_types.name ORDER BY quantity DESC LIMIT 5;
```

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

	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

# Total quantity of each pizza category ordered

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :
2 -- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
3
4 • SELECT
5     pizza_types.category,
6     SUM(order_details.quantity) AS quantity FROM pizza_types
7     JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8     JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
9     GROUP BY pizza_types.category ORDER BY quantity DESC;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Category	quantity		
▶	Classic	14888		
	Supreme	11987		
	Veggie	11649		
	Chicken	11050		

# The distribution of orders by hour of the day

```
1 -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2 -- 7. Determine the distribution of orders by hour of the day.  
3  
4 • SELECT HOUR(order_time) as hour, COUNT(order_id)  
5   AS order_count  
6   FROM  
7   orders GROUP BY HOUR(order_time);
```

	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

# The category-wise distribution of pizzas

```
1    -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :  
2  
3    -- 8. Join relevant tables to find the category-wise distribution of pizzas.  
4  
5 • SELECT category, COUNT(*)  
6     FROM pizza_types  
7     GROUP BY category;
```

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

	category	COUNT(*)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# Group the orders by date and give the average number of pizzas ordered per day

```
1 -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2 -- 9. Group the orders by date and calculate the average number of pizzas ordered per day.  
3 -- Sol: the solution is solved by using subquery method  
4  
5 • SELECT  
6     round(AVG(quantity),0) average FROM  
7     (SELECT orders.order_date, SUM(order_details.quantity) AS quantity  
8      FROM orders  
9     JOIN order_details ON orders.order_id = order_details.order_id  
10    GROUP BY orders.order_date) AS temptable;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
average			
138			

# The top 3 most ordered pizza types based on revenue

```
1 -- Shubham's (LCI2021036) SQL based pizza_sales_analysis project :
2 -- 10. Determine the top 3 most ordered pizza types based on revenue.
3
4 • SELECT pizza_types.name,
5      SUM(order_details.quantity * pizzas.price) AS revenue
6  FROM pizza_types
7  JOIN pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
8  JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
9  GROUP BY pizza_types.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

# Percentage contribution of each pizza type to total revenue

```
1      -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :
2      -- 11. Give the percentage contribution of each pizza type to total revenue.
3 •  SELECT
4      pizza_types.category,
5      ROUND(SUM(order_details.quantity * pizzas.price / (SELECT
6          ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales FROM order_details JOIN
7          pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100), 2) AS revenue FROM pizza_types
8      JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9      JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
10     GROUP BY pizza_types.category ORDER BY revenue DESC;
```

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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# The cumulative revenue generated over time

```
1      -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :  
2      -- 12. Analyze the cumulative revenue generated over time.  
3 •  select order_date,  
4      sum(revenue) over(order by order_date) as cum_revenue  
5      from  
6      ( select orders.order_date,  
7          sum(order_details.quantity * pizzas.price ) as revenue  
8          from order_details join pizzas  
9          on order_details.pizza_id = pizzas.pizza_id  
10         join orders  
11         on orders.order_id = order_details.order_id  
12     group by orders.order_date ) as sales;
```

	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	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.75000000001
	2015-01-18	40978.60000000006
	2015-01-19	43365.75000000001
	2015-01-20	45763.65000000001
	2015-01-21	47804.20000000001



# The top 3 most ordered pizza types based on revenue for each pizza category

```
1  -- Shubham's (LCI2021036) SQL based pizza sales_analysis project :
2  -- 13. Determine the top 3 most ordered pizza types based
3  -- on revenue for each pizza category.
4
5 • select name,revenue from
6 ⊖ (select category, name, revenue,
7      rank() over (partition by category order by revenue desc) as ranki
8      from
9      (select pizza_types.category, pizza_types.name ,
10         sum((order_details.quantity) * pizzas.price ) as revenue
11        from pizza_types join pizzas
12          on pizza_types.pizza_type_id = pizzas.pizza_type_id
13        join order_details
14        group by pizza_types.category,pizza_types.name ) as alpha ) as beta
15      where ranki <=3 ;
```

	name	revenue
►	The Thai Chicken Pizza	2491093.5
	The Southwest Chicken Pizza	2491093.5
	The Chicken Pesto Pizza	2491093.5
	The Chicken Alfredo Pizza	2491093.5
	The California Chicken Pizza	2491093.5
	The Barbecue Chicken Pizza	2491093.5
	The Greek Pizza	5450661.3000048855
	The Napolitana Pizza	2404339
	The Italian Capocollo Pizza	2404339
	The Classic Deluxe Pizza	2404339
	The Big Meat Pizza	2404339
	The Spinach Supreme Pizza	2466306.5
	The Spicy Italian Pizza	2466306.5
	The Soppressata Pizza	2466306.5
	The Prosciutto and Arugula ...	2466306.5
	The Pepper Salami Pizza	2466306.5
	The Italian Supreme Pizza	2466306.5
	The Italian Vegetables Pizza	2503487
	The Spinach Pesto Pizza	2466306.5
	The Vegetables + Vegetable...	2391945.5
	The Spinach and Feta Pizza	2391945.5
	The Mexicana Pizza	2391945.5
	The Mediterranean Pizza	2391945.5
	The Green Garden Pizza	2391945.5

# Conclusion

This project not only highlights the power of SQL in data analysis but also offers actionable insights to optimize our pizza sales strategy.

# THANK YOU