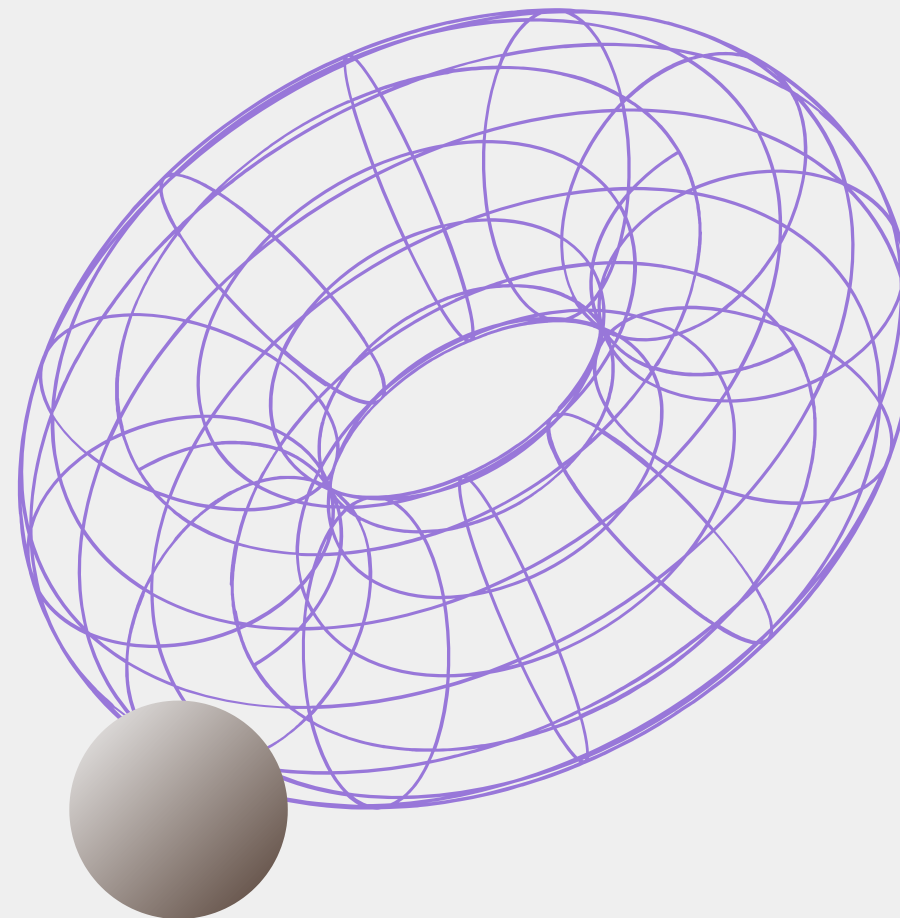


**PIZZA SALES ANALYSIS**

# Pizza Sales Data Analysis using SQL

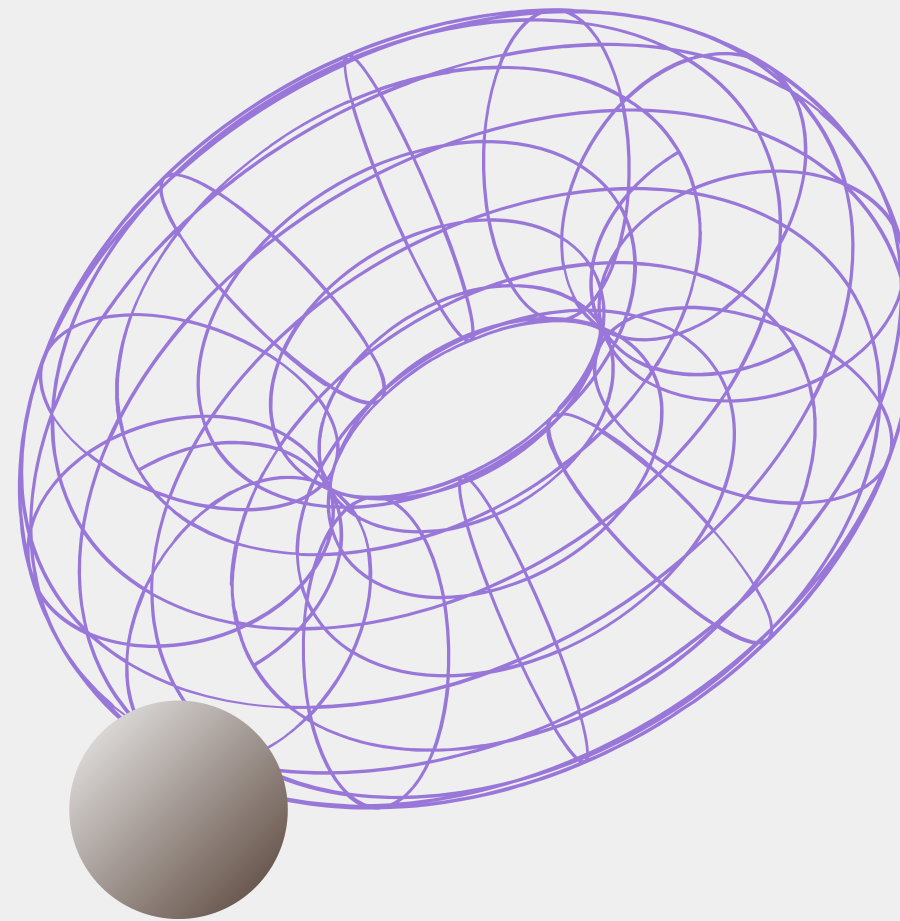
From Basic to Advanced SQL Queries, Presented by Ansh Raj

# Analyzing Pizza Sales Data with SQL



This project focuses on using SQL to analyze pizza sales data effectively. By leveraging tools like MySQL and CSV files, we can gain valuable insights into sales trends, customer preferences, and revenue generation.

# Setting Up the Database Structure



Creating a well-structured database is essential for efficient data analysis. The **pizzahut** database comprises multiple tables: orders, order\_details, pizzas, and pizza\_types, all imported from CSV files to ensure seamless integration and accessibility of data.

# Total number of orders placed.

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

	total_orders
▶	21350

# 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
```

	total_sales
▶	817860.05

# 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;
```

	name	price
▶	The Greek Pizza	35.95

# 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;
```

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

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



# 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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# Distribution of orders by hour of the day

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

	hour	order_count
▶	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

# Category-wise distribution of pizzas

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

	category	count_name
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

# Average number of pizzas ordered per day

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_orderd_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;
```

	<code>round(avg(quantity),0)</code>
▶	138

# 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;
```

	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

```
SELECT
  pizza_types.category,
  ROUND(
    (SUM(order_details.quantity * pizzas.price) /
     (SELECT ROUND(SUM(order_details.quantity * pizzas.price), 2)
      FROM order_details
      JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100, 2
  ) 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
ORDER BY
  revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# 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;
```

	order_date	cum_revenue
	2015-12-07	769964.25
	2015-12-08	771820.5
	2015-12-09	774392.05
	2015-12-10	776377.65
	2015-12-11	779011.65
	2015-12-12	780971.8
	2015-12-13	783216.9500000001
	2015-12-14	785389.55
	2015-12-15	787777
	2015-12-16	790011.8
	2015-12-17	791892.55
	2015-12-18	794778.8500000001
	2015-12-19	797083.05
	2015-12-20	799187.9500000001
	2015-12-21	801288.65
	2015-12-22	803171.6
	2015-12-23	805415.9
	2015-12-24	807553.75
	2015-12-26	809196.8
	2015-12-27	810615.8
	2015-12-28	812253
	2015-12-29	813606.25
	2015-12-30	814944.05
	2015-12-31	817860.05

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

```
select 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;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.700000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5





# Thank You for Your Attention

Connect with me on GitHub and LinkedIn

I appreciate your time and interest in my presentation.  
For further inquiries or collaboration, feel free to reach  
out through GitHub, LinkedIn, or email. Let's connect and  
explore opportunities together!