



LARANA PIZZA

# PIZZA SALES ANALYSIS





# WELCOME TO THE PROJECT

Hi, My name is Niharika Sachan and I made this project based on sql where I utilized some sql queries to solve the questions on pizza sales.



# QUERIES



## Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

## Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

## Advanced:

- 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.

-- Retrieve the total numbers of order placed--

```
SELECT count(order_id) FROM orders;
```

Result Grid	
	count(order_id)
▶	21350

-- Calculate the total revenue generated from pizza sales --

SELECT

ROUND(SUM(order\_details.quantity \* pizzas.price),

3) 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 price 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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



-- Identify the most common pizza size ordered

```
SELECT
    pizzas.size, COUNT(order_details.order_detail_id) AS o_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY o_count DESC
LIMIT 1;
```

Result Grid

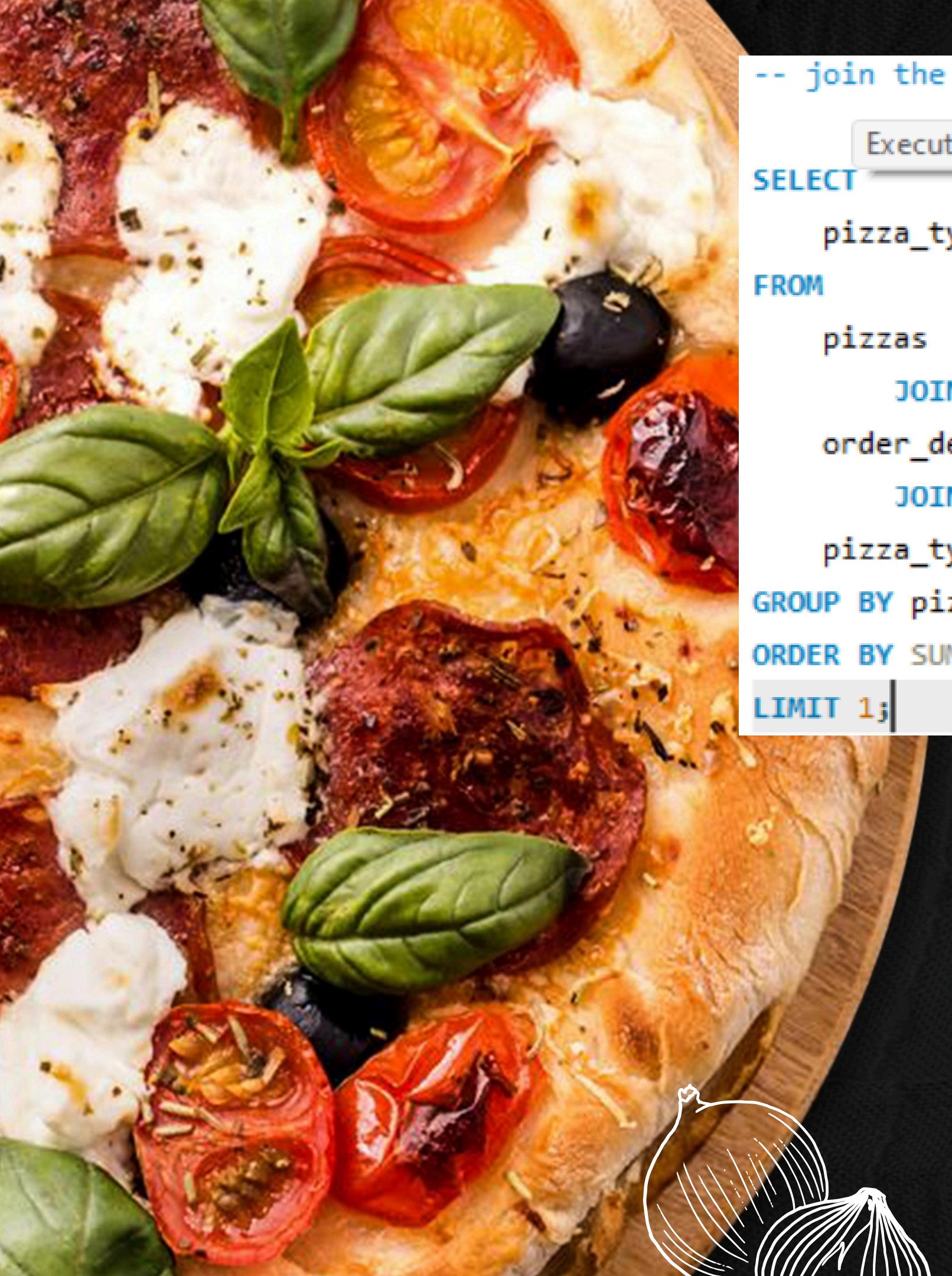
size	o_count
L	18526

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

```
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity) AS s_order_details  
FROM  
    pizzas  
        JOIN  
    pizza_types 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 s_order_details DESC  
LIMIT 5;
```

Result Grid | Filter Rows:

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



-- join the necessary tables to find the total quantity of each pizza category.

Execute the statement under the keyboard cursor

```
SELECT pizza_types.category, SUM(order_details.quantity)
FROM pizzas
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id
JOIN pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY SUM(order_details.quantity) DESC
LIMIT 1;
```

Result Grid |  Filter Rows:

	category	sum(order_details.quantity)
▶	Classic	14888

```
-- Determine the distribution of orders by hour of the day  
  
SELECT hour(order_time) AS hour, count(order_id) AS counting  
FROM  
orders  
group by hour(order_time);
```

	hour	counting
▶	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

-- Joint relevant tables to find the category wise distribution of pizzas

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

Result Grid | Filter Rows

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

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

SELECT avg(quan) FROM
(SELECT orders.order_date, sum(order_details.quantity) AS quan
FROM
orders JOIN order_details
ON
orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS order_quantity;
```

Result Grid	
	avg(quan)
▶	138.4749

```
-- 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 pizza_types.pizza_type_id = pizzas.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;
```

Result Grid | Filter Rows:

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

```
-- 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 pizzas  
JOIN order_details ON pizzas.pizza_id = order_details.pizza_id) * 100,2) AS revenue  
FROM  
pizzas JOIN pizza_types  
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;
```

Result Grid | Filter

	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
order_details.pizza_id = pizzas.pizza_id
JOIN
orders
ON
orders.order_id = order_details.order_id
GROUP BY orders.order_date) AS sales;
```

	order_date	cum_revenue
▶	2015-01-01 00:00:00	2713.8500000000004
	2015-01-02 00:00:00	5445.75
	2015-01-03 00:00:00	8108.15
	2015-01-04 00:00:00	9863.6
	2015-01-05 00:00:00	11929.55
	2015-01-06 00:00:00	14358.5
	2015-01-07 00:00:00	16560.7
	2015-01-08 00:00:00	19399.05
	2015-01-09 00:00:00	21526.4
	2015-01-10 00:00:00	23990.35000000002
	2015-01-11 00:00:00	25862.65
	2015-01-12 00:00:00	27781.7

```
-- Determine the 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.name, pizza_types.category, sum(order_details.quantity*pizzas.price) AS revenue
FROM order_details JOIN pizzas
ON order_details.pizza_id = pizzas.pizza_id
JOIN pizza_types
ON
pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name, pizza_types.category) 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	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5