

# PIZZA SALES ANALYSIS



# Tables we have

- **orders**
- **order\_details**
- **pizzas**
- **pizza\_types**

# Business Question to Answer

- 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.





- Join the necessary tables to find 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 average number of pizzas ordered per day.
- 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.

# Retrieve the total number of orders placed

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

Result Grid	
	total_orders
▶	21350





# Calculate the total revenue generated from pizza sales

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

	total_revenue
▶	817860.05



# Identify the 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;
```



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_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

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS order_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 order_quantity DESC
LIMIT 5;
```

name	order_quantity
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 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



# Determine the distribution of orders by hour of the day

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

	order_time	total_orders
▶	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



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

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

Result Grid | Filter Row

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



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

```
SELECT  
    ROUND(AVG(quantity), 0) AS average_pizza_ordered_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;
```



A screenshot of a database query results window. At the top, there are tabs for "Result Grid" and "Filter Rows:". Below the tabs, the column header "average\_pizza\_ordered\_per\_day" is shown in blue. A single data row is present, containing the value "138".

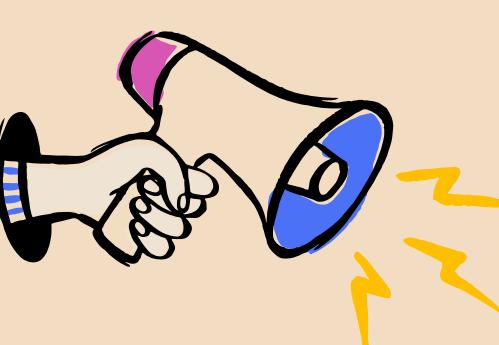
average_pizza_ordered_per_day
138



# Determine the top 3 most ordered pizza types based on revenue

```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) 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.name
ORDER BY price DESC
LIMIT 3;
```

	name	price
▶	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(pizzas.price * order_details.quantity) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_revenue
    )
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) AS perc_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
```

Result Grid |   Filter Row

	category	perc_revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96



# Analyze the cumulative revenue generated over time

```
select order_date,  
round(sum(revenue) over (order by order_date),2) as cum_revenue  
from  
(select orders.order_date, sum(pizzas.price * order_details.quantity) as revenue  
from  
orders  
join  
order_details on  
orders.order_id = order_details.order_id  
join  
pizzas  
on  
pizzas.pizza_id = order_details.pizza_id  
group by orders.order_date) as sales;
```

order_date	cum_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	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40079.6



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

```
select category, 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, round(sum(pizzas.price * order_details.quantity),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, pizza_types.name) as sub_table)
as sub_sub_table
where rn<=3;
```

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5



# Thank You!

-BY RISHABH JAIN