

# *Delicious* **Pizza**

*Hot &  
fresh*



# Hello Data Geeks!!!

*Michael here, I am a  
professional Data Analyst  
Project Title - SQL analysis  
of Pizza Dataset*

# Questions

## 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:

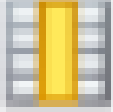

Calculate the percentage contribution of each pizza type to total revenue.

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 number of orders placed.*

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

Result Grid			
	count(order_id)		
▶	21350		

*Calculate the total revenue generated from pizza sales.*

SELECT

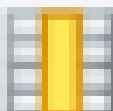


```
ROUND(SUM(pizzas.price * order_details.quantity),  
      2) as total_sale
```

FROM

```
pizzas
```

JOIN

```
order_details ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid			
	total_sale		
	817860.05		

*Identify the highest-priced pizza.*

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
    JOIN
        pizzas ON
pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

*Identify the most common pizza size ordered.*

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

	size	orders
▶	L	18526

*List the 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 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
LIMIT 5;
```

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

*Determine the distribution of orders by hour of the day.*

```
SELECT  
    HOUR(orders.time)  
    AS Hour,  
    COUNT(order_id) AS Order_count  
FROM  
    orders  
GROUP BY hour;
```

	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

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

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

	category	pizzas
▶	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 AVG_orders_per_Day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity)
        AS quantity
    FROM
        orders
    JOIN order_details
    ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```

	AVG_orders_per_Day
▶	138

*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 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

***Calculate the percentage contribution of each pizza type to total revenue.***

```
SELECT pizza_types.category,  
(SUM(order_details.quantity * pizzas.price)/  
( select round(sum(pizzas.price*order_details.quantity),  
2)  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id ))* 100  
as percentage  
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 percentage DESC ;
```

category	percentage
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

*Analyze the cumulative revenue generated over time.*

```
select date , sum(sales_per_day)
over(order by date) as cummulative_revenue from
(select orders.date ,
sum(pizzas.price*order_details.quantity)
as sales_per_day
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.date) as sales;
```

date	cummulative_revenue
2015-01-01	2713.8500000000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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

```
(select category , name , revenue ,  
Rank() over(partition by category order by revenue desc)  
as rn from  
  
(select pizza_types.category , pizza_types.name,  
sum(pizzas.price*order_details.quantity) as revenue  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
join pizza_types  
on pizza_types.pizza_type_id = pizzas.pizza_type_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