



DATA ANALYTICS PROJECT

PIZZA SALES ANALYSIS USING MYSQL

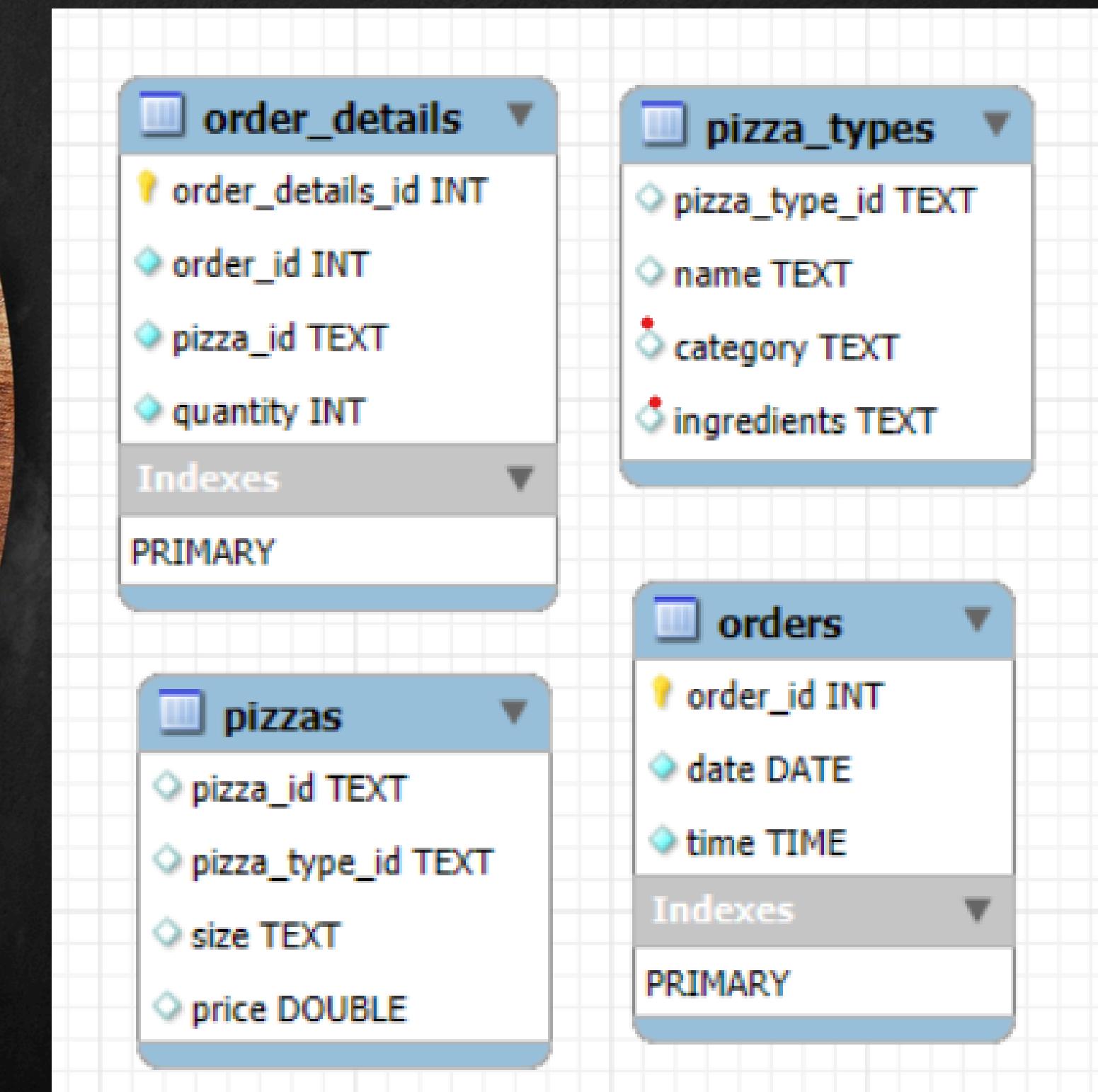


HI, MY NAME IS KRISHNA

I used SQL to analyze pizza sales data and answer key business questions. The analysis focused on identifying trends, such as top-selling pizzas, peak order times, and revenue contributions, providing actionable insights for operational optimization.



ENTITY-RELATIONSHIP DIAGRAM



BUSINESS QUESTIONS

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.
6. Join the necessary tables to find the total quantity of each pizza category ordered.
7. Determine the distribution of orders by hour of the day.
8. Join relevant tables to find the category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. Determine the top 3 most ordered pizza types based on revenue.
11. Calculate the percentage contribution of each pizza type to total revenue.
12. Analyze the cumulative revenue generated over time.
13. 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(pizzas.price * order_details.quantity),  
        2) AS total_revenue  
FROM  
    pizzas  
    .  
    .  
    .  
JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid	
	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 Results

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size, count(order_details.quantity) as ordered
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by ordered desc
limit 1;
```

Result Grid

	size	ordered
▶	L	18526

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

```
select pizza_types.name, count(order_details.order_id) as qty_ordered
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 qty_ordered desc
limit 5;
```

Result Grid |  Filter Rows:

	name	qty_ordered
▶	The Classic Deluxe Pizza	2416
	The Barbecue Chicken Pizza	2372
	The Hawaiian Pizza	2370
	The Pepperoni Pizza	2369
	The Thai Chicken Pizza	2315

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
select pizza_types.category, sum(order_details.quantity) as total_qty_ordered
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 total_qty_ordered desc;
```

Result Grid | Filter Rows:

	category	total_qty_ordered
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
select hour(orders.time), count(orders.order_id)  
from orders group by hour(orders.time)  
order by count(orders.order_id) desc
```

Result Grid |  Filter Rows: _____

	hour(orders.time)	count(orders.order_id)
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	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 avg(total_ordered) as average_orders_per_day from
  (select orders.date, sum(order_details.quantity) as total_ordered
   from orders join order_details on orders.order_id = order_details.order_id
   group by orders.date) as temp;
```

Result Grid |  Filter Row

	average_orders_per_day
▶	138.4749

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
select pizza_types.name, round(sum(pizzas.price * order_details.quantity),0) as price
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 price desc limit 3;
```

Result Grid |  Filter Rows:

	name	price
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

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(pizzas.price*order_details.quantity),2) as total_sales  
from pizzas join order_details on pizzas.pizza_id = order_details.pizza_id ) * 100,2) as revenue_percentage  
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.category;
```

Result Grid |  Filter Rows: 

	category	revenue_percentage
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select date, round(sum(revenue) over(order by date),2) as rev from
(select orders.date, sum(pizzas.price * order_details.quantity) as revenue
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 temp;
```

Result Grid |  Filter Rows: 

	date	rev
▶	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

Default 15 rows

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 R 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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) as B) as A
where R <= 3;
    
```

Result Grid | Filter Rows:

	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

CONCLUSION

Through this analysis of pizza sales data using MySQL, we were able to gain valuable insights into customer preferences, sales trends, and top-performing products. These findings provide actionable data that can help the business make informed decisions on inventory, pricing strategies, and targeted promotions.

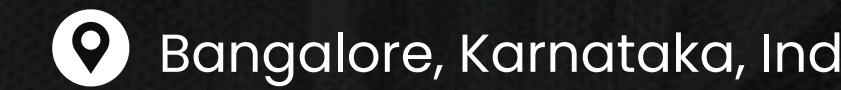




OUR CONTACT



LinkedIn
[krishnamksheerasagar@gmail.com](https://www.linkedin.com/in/krishnamksheerasagar/)



Bangalore, Karnataka, India



PIZZA SALES ANALYSIS

THANK YOU!

