



SPECIAL PIZZA SALE



Contact Us:

- +123-456-7890
- www.reallygreatsite.com



ORDER NOW >>>>

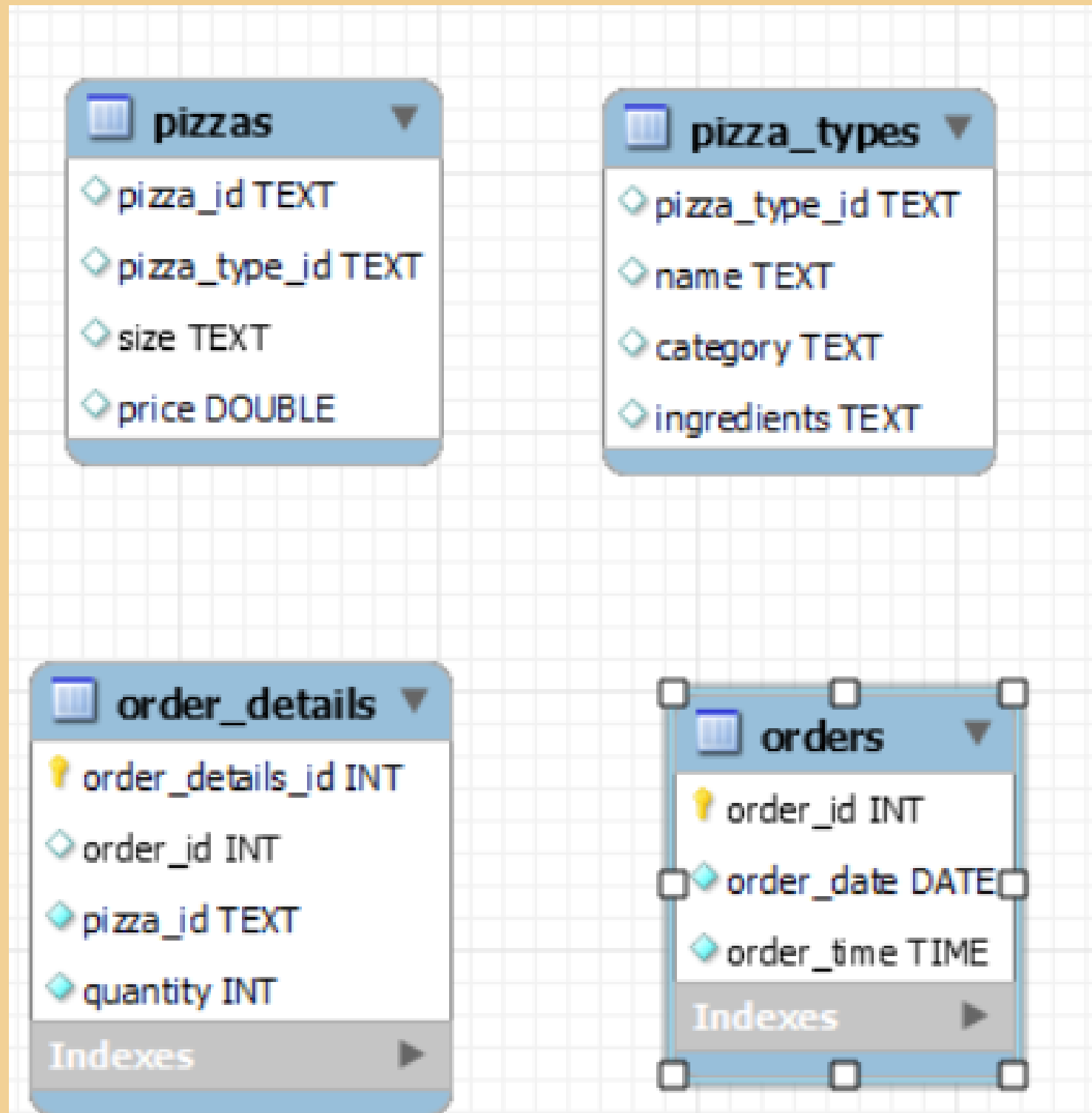


Hello,

my name is Sawai Singh I have worked on numerous projects where I analyzed sales data, identified trends, and provided actionable recommendations to improve performance. In this project, I will be using SQL queries to delve into the intricacies of pizza sales.



Entity-Relationship (ER) Diagram



Objectives

BASICS :

1. Calculate the total number of orders placed.
2. Calculate the total revenue generated from the pizza sales.
3. Identify the highest prize pizza.
4. Identify the most common pizza size ordered
5. List the top 5 most ordered pizza size-types along with there quantities

INTERMEDIATE

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

Advanced

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

Calculate 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 the pizza sales.

SELECT

 ROUND(SUM(order_details.quantity * pizzas.price),
2) AS total_sales

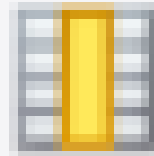
FROM

order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id

Result Grid



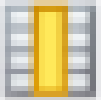

total_sales



817860.05



Identify the highest prize 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
    order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid					Filter Rows:
	size	order_count			
▶	L	18526			
	M	15385			
	S	14137			
	XL	544			
	XXL	28			

List the top 5 most ordered pizza size-types along with there quantities

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

Result Grid			  Filter Rows: <input type="text"/>	
	name	Total_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 ordered.

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_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 total_quantity DESC;
```

Result Grid			Filter Rows:	
	category	total_quantity		
▶	Classic	14888		
	Supreme	11987		
	Veggie	11649		
	Chicken	11050		

Determine the distribution of orders by hour of the day.



```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY 1;
```

HOUR(order_time)	COUNT(order_id)
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

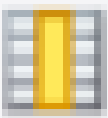


Join relevant tables to find the category-wise distribution of prizes

```
SELECT category, count(name)
FROM pizza_types
GROUP BY 1;
```

Result Grid   Filter Rows:		
	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Group the order by data and calculate the average number of pizzas ordered per day

```
SELECT ROUND(AVG(quantity),0) AS avg_pozza_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 1) AS order_quantity;
```

Result Grid				Filter Rows:
	avg_pozza_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 1
ORDER BY 2 DESC
LIMIT 3;
```

Result Grid   Filter Rows: <input type="text"/>		
	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
        order_details
        JOIN
            pizzas ON order_details.pizza_id = pizzas.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 1;
```

Result Grid			Filter R
	category	revenue	
▶	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

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 1) AS sales;
```

order_date	cum_revenue
2015-01-01	2713.8500000000000
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	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

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.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 1,2) 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

CONCLUSION

Completing the project on "Pizza Sales" has been an eye-opening journey that significantly enhanced my skills and knowledge. Throughout the process, I delved deeply into the intricacies of SQL Queries, mastering complex commands and optimizing database performance. This hands-on experience was not only intellectually stimulating but also remarkably rewarding. It allowed me to tackle real-world data problems, thereby sharpening my problem-solving abilities. I encountered various challenges that required creative solutions, which in turn expanded my understanding of data management and analytics. Overall, this project has been a wonderful learning experience, equipping me with valuable skills that I am excited to apply in future endeavors.