it's Pizza Time

ORDER NOW

123-456-7890

HELLO

In this project, I, Abhyuday, leveraged my SQL skills to analyze and solve complex queries related to pizza sales. By harnessing the power of SQL, I uncovered valuable insights into customer preferences, sales trends, and revenue growth, demonstrating my ability to extract meaningful data-driven findings from large datasets.

OBJECTIVE

This project analyzes pizza sales data to assess restaurant performance, identify
trends, and inform data-driven decisions, enabling the business to optimize
operations, enhance customer experiences, and plan strategically for future growth.

Retrieve the total no. of orders placed

```sql
select * from orders;
select count(order_id) as total_orders from orders;
```

Result Grid

total_orders
21350

Calculate the total revenue generated from pizza sales

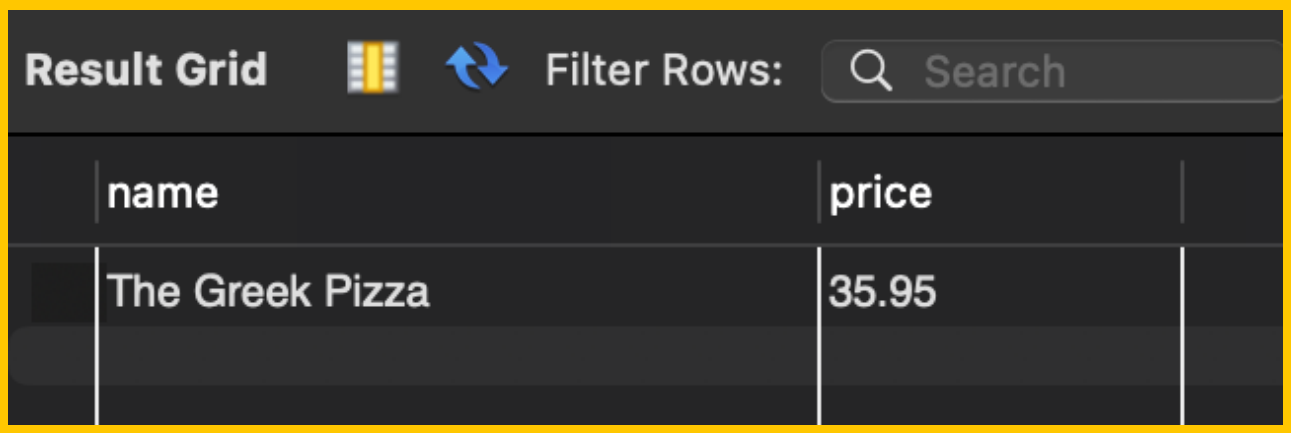
```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid

total_sales
817860.05

identify the highetst priced pizza

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



identify the most comman pizza size ordered

```sql
SELECT
    pizzas.size, COUNT(order_details.order_details_id)
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size;
```

Result Grid Filter Rows: Search

size	count(order_details.order_details...
M	15385
L	18526
S	14137
XL	544
XXL	28

List the top 5 most ordered pizza types along with theie quantity

```sql
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 name order by quantity desc limit 1;
```

name	quantity
The Classic Deluxe Pizza	2453

Join the necessary table to find the total quantity of each pizza category ordered

```sql
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;
```

Result Grid   Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```sql
select * from orders;

select hour(time)as hour,count(order_id) as order_count from orders
group by hour (order_id);

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

	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	
	22	663	
	23	28	
	10	8	
	9	1	

Join relevent tables to find the category-wise distribution of pizzas

```sql
select category ,count(name) from pizza_types
group by category;
```

Result Grid   Filter Rows:

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

Group the orders by date and calculate the average numbers of pizza ordered by day

```sql
SELECT
    ROUND(AVG(quantity), 0)
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 orders_quantity;
```

Result Grid Filter Rows:

round(avg(quantity),0)
138

Determine the top 3 ordered pizza types based on revenue

```sql
select pizza_types.name,sum(order_details.quantity * pizzas.price) as revenue
from pizza_types join pizzas
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 revenue desc limit 3;
```

Result Grid   Filter Rows: Search

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

Calculate the percentage contribution of eaach pizza type to total revenue

```sql
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 pizzas.pizza_id = order_details.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 pizza_types.category ;
```

Result Grid   Filter Ro

category	revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue genetrated over time

```sql
select date,sum(revenue) over (order by date) as cum_revenue
from
(select orders.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.date ) as sales;
```

Result Grid Filter Rows: Search

date	cum_revenue
2015-01-01	2713.8500000000004
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

date	cum_revenue
2015-01-14	32358.700000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.600000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.600000000006
2015-01-24	55013.850000000006
2015-01-25	56631.40000000001
2015-01-26	58515.80000000001

determine the topmost ordered pizza types based on revenue for each pizza category

```sql
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 pizza_types.category, pizza_types.name) as a) as b
where rn <= t3;
```

Result Grid Filter Rows: Search

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

"Thank you, your support is highly valued."