

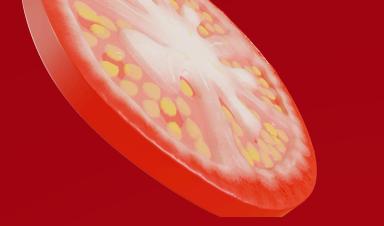
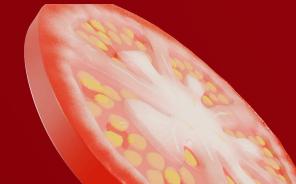
PIZZA_SALES



Objectives

The objective of this project is to analyze pizza sales data to extract meaningful insights that help understand sales trends, revenue generation, and customer preferences. The analysis will provide key performance indicators (KPIs) such as:

- total revenue
- average order value
- best-selling pizza types
- peak order times.



--Retrieve the total number of orders placed.

```
select count(order_id) as orders_placed  
from orders;
```

A screenshot of a SQL query results window. The title bar shows "110 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has one column labeled "orders_placed". A single row contains the value "21350", which is highlighted with a dashed border.

orders_placed
21350

--Calculate the total revenue generated from pizza sales.

```
select round(sum(order_details.quantity * pizzas.price), 0) as total_revenue  
from order_details  
join pizzas  
on order_details.pizza_id = pizzas.pizza_id
```

A screenshot of a SQL query results window. The title bar shows "110 %". Below it are two tabs: "Results" (selected) and "Messages". The results table has one column labeled "total_revenue". A single row contains the value "817860", which is highlighted with a dashed border.

total_revenue
817860

--Identify the highest-priced pizza.

```
-select top 1 pizza_types.name, pizzas.size,  
round(max(pizzas.price), 0) as highest_priced_pizza  
from pizza_types  
join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
group by pizza_types.name, pizzas.size  
order by highest_priced_pizza desc
```

	name	size	highest_priced_pizza
1	The Greek Pizza	XXL	36

--Identify the most common pizza size ordered.

```
-select pizzas.size,  
sum(order_details.quantity) as order_size  
from pizzas  
join order_details  
on pizzas.pizza_id = order_details.pizza_id  
group by pizzas.size  
order by order_size desc;
```

	size	order_size
1	L	18956
2	M	15635
3	S	14403
4	XL	552
5	XXL	28

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

```
select top 5 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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.name
order by quantity desc;
```

110 %

Results Messages

	name	quantity
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	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 orders_quantity
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
order by orders_quantity desc;
```

110 %

Results Messages

	category	orders_quantity
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

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

```
select DATEPART(hour, orders.time) as hours,  
count(order_id) as order_by_hour  
from orders  
group by DATEPART(hour, orders.time)  
order by order_by_hour desc;
```

110 %

Results Messages

	hours	order_by_hour
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009
6	16	1920
7	20	1642
8	14	1472

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

```
select avg(total_pizza_per_day) as average_pizza_per_day  
from  
(select orders.date,  
sum(order_details.quantity) as total_pizza_per_day  
from orders  
join order_details  
on orders.order_id = order_details.order_id  
group by orders.date) as a
```

110 %

Results Messages

	average_pizza_per_day
1	138

--Determine the top 3 most ordered pizza types based on revenue.

```
select top 3 pizza_types.name,
round(sum(order_details.quantity * pizzas.price), 0) 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.name
order by revenue desc
```

110 %

Results Messages

	name	revenue
1	The Thai Chicken Pizza	43434
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41410

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

```
select pizza_types.name,
round(sum(order_details.quantity * pizzas.price) * 100 /
(select sum(order_details.quantity * pizzas.price))
from order_details
join pizzas
on order_details.pizza_id = pizzas.pizza_id), 2) as percentage_contribution_by_total_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.name
order by percentage_contribution_by_total_revenue desc;
```

110 %

Results Messages

	name	percentage_contribution_by_total_revenue
1	The Thai Chicken Pizza	5.31
2	The Barbecue Chicken Pizza	5.23
3	The California Chicken Pizza	5.06
4	The Classic Deluxe Pizza	4.67
5	The Spicy Italian Pizza	4.26
6	The Southwest Chicken Pizza	4.24
7	The Italian Supreme Pizza	4.09
8	The Four Cheese Pizza	3.95
9	The Hawaiian Pizza	3.85

```
--Calculate the percentage contribution of each pizza category to total revenue.
```

```
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price) * 100 /
(select sum(order_details.quantity * pizzas.price)
from order_details
join pizzas
on order_details.pizza_id = pizzas.pizza_id), 2) as revenue_percent
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
order by revenue_percent desc;
```

```
--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) 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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn<=3;
```

The screenshot shows a SQL query results window with a title bar "110 %". Below it are two tabs: "Results" (selected) and "Messages". The "Results" tab displays a table with three columns: "category", "revenue_percent", and a row number "1". The data rows are:

	category	revenue_percent
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

The screenshot shows a SQL query results window with a title bar "110 %". Below it are two tabs: "Results" (selected) and "Messages". The "Results" tab displays a table with four columns: "category", "name", and "revenue". The data rows are:

	category	name	revenue
1	Chicken	The Chicken Pesto Pizza	16701.75
2	Chicken	The Chicken Alfredo Pizza	16900.25
3	Chicken	The Southwest Chicken Pizza	34705.75
4	Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
5	Classic	The Big Meat Pizza	22968
6	Classic	The Napolitana Pizza	24087
7	Supreme	The Brie Carré Pizza	11588.4998130798
8	Supreme	The Spinach Supreme Pizza	15277.75
9	Supreme	The Calabrese Pizza	15934.25



Database Schema:

The database consists of the following key tables:

1. Orders – Contains order details including order ID, date, and time.
2. Order_Details – Stores information about each order, including pizza ID and quantity ordered.
3. Pizzas – Contains details about different pizzas, including pizza ID, price, and type ID.
4. Pizza_Types – Holds pizza category information such as name and description.





Schema Relationships:

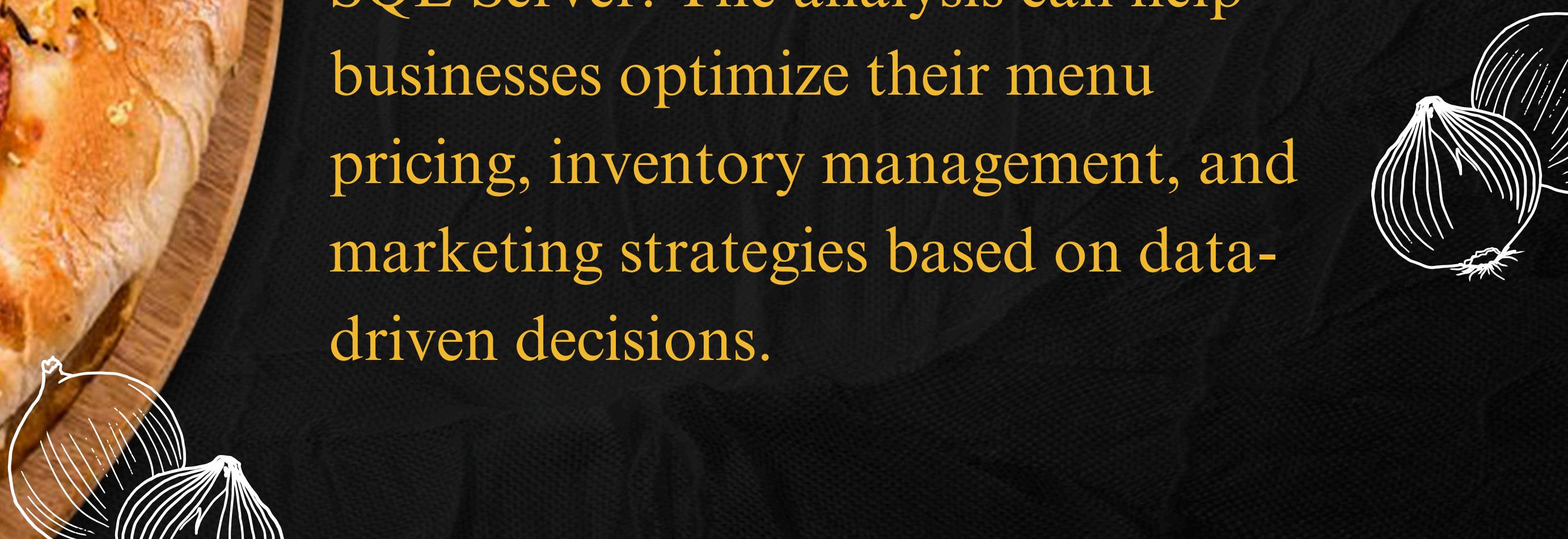
- Order_Details references Orders via Order_ID.
- Order_Details references Pizzas via Pizza_ID.
- Pizzas references Pizza_Types via Pizza_Type_ID.





Conclusion:

This project provides valuable insights into sales performance, customer preferences, and revenue trends using SQL Server. The analysis can help businesses optimize their menu pricing, inventory management, and marketing strategies based on data-driven decisions.



THANK YOU!

