PIZZA SALES

SQL PROJECT

Dive In >>>

This project unveiled many valuable insights from the Pizza Sales data which can be used for making business decisions.

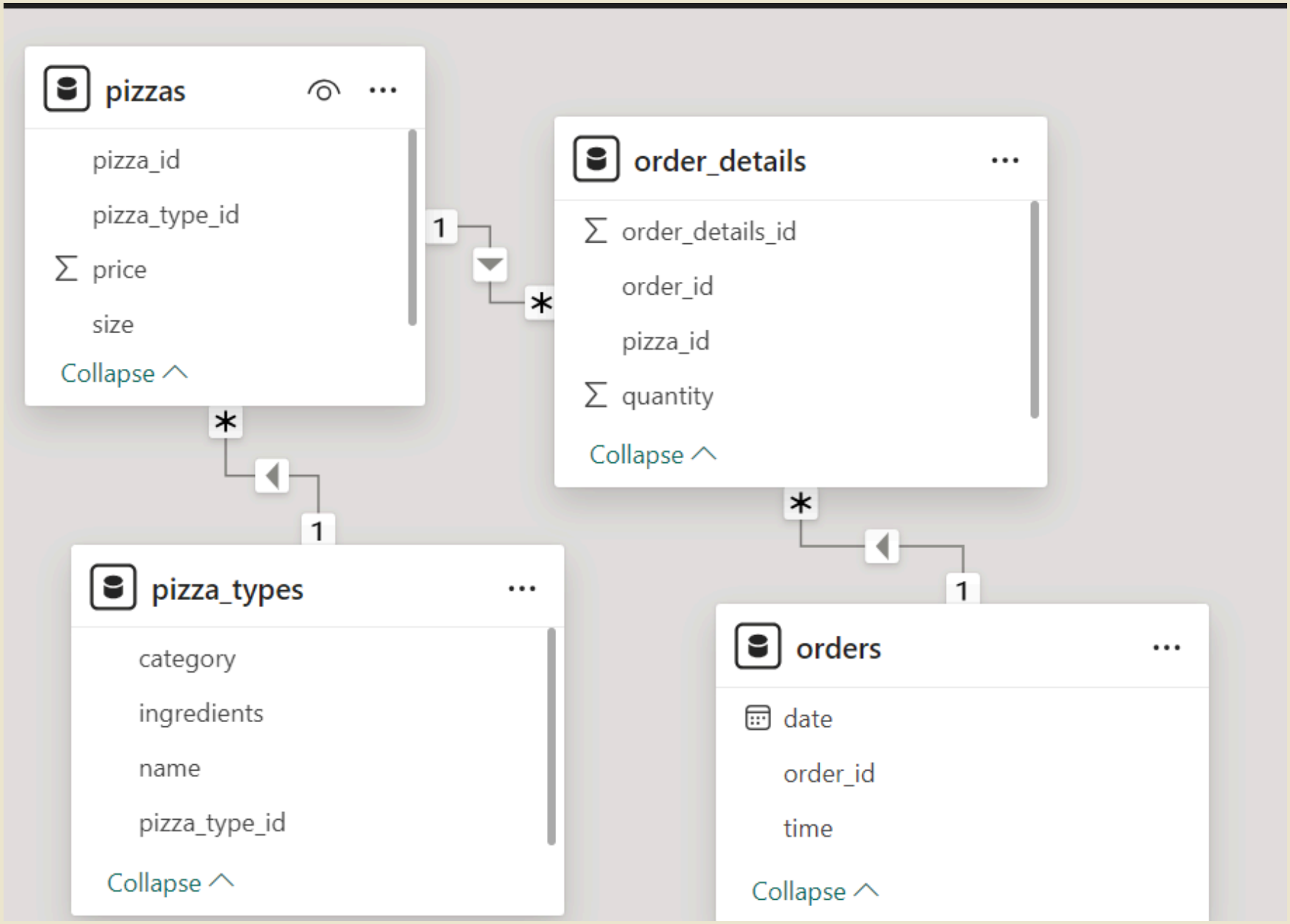
Key Indicators

- Sales Trends: peak sales periods and seasonal trends.
- Customer Preferences: popular pizza types and customization preferences.
- Operational Efficiency: Assessed delivery times and optimized supply chain processes.

Pizza Sales Tables

1. PIZZAS
2. PIZZA_TYPES
3. ORDERS
4. ORDER_DETAILS

Pizza Sales Data Model



Total number of orders placed

```sql
SELECT
    COUNT(order_id) AS totalorders
FROM
    order_details;
```

Result Grid

	Total_Orders
▶	48620

Total revenue generated from pizza sales

```sql
select round(sum(order_details.quantity * pizzas.price),2) as Totalrevenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	Totalrevenue
▶	817860.05

Top 5 pizza orders with quantity

```sql
select pizza_types.name as Pizza_Name , sum(order_details.quantity) as Pizza_Qnty
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 Pizza_Qnty desc
limit 5;
```

Result Grid | Filter Rows:

Pizza_Name	Pizza_Qnty
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Highest-priced pizza

```sql
select pizza_types.name as Pizza_Name, pizzas.price as Price
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by price desc
limit 1;
```

Result Grid | Filter Ro

	Pizza_Name	Price
▶	The Greek Pizza	35.95

Most Common Pizza-size Ordered

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

Result Grid | Fi

	size	Pizza_Qnty
▶	L	18956

Total quantity of each pizza category ordered

```sql
select pizza_types.category, sum(order_details.quantity) as Quantity
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 category;
```

Result Grid | Filte

category	Quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Distribution of orders by hour of the day

```sql
select Hour(orders.time) as hour_day, count(order_id) as total_number
from orders
group by hour_day
order by hour_day;
```

Result Grid | Filter Row

	hour_day	total_number
▶	9	1
	10	8
	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

Category-wise distribution of pizzas

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

Result Grid

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Average number of pizzas ordered per day

```
select avg(total_orders) as Average_num from
(select orders.date, sum(order_details.quantity) as total_orders from orders
join order_details on order_details.order_id= orders.order_id
group by orders.date) as order_quantity;
```

Result Grid

	Average_num
▶	138.4749

Top 3 most ordered pizza types based on revenue

```sql
select pizza_types.name, round(sum(order_details.quantity * pizzas.price),2) as Totalrevenue
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.name
order by Totalrevenue desc
limit 3;
```

Result Grid | Filter Rows:

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

The percentage contribution of each pizza type to total revenue

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

Result Grid | Filter Rows

category	Totalrevenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

The cumulative revenue generated over time

```
(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 order_details.order_id = orders.order_id
group by orders.date) as sales;
```

Top 3 most ordered pizza types based on revenue for each pizza category

```
select name, Totalrevenue from
(select category, name, Totalrevenue,
Rank() over(partition by category order by Totalrevenue desc) as Ranknr from
(select pizza_types.category, pizza_types.name,
sum(order_details.quantity * pizzas.price) as Totalrevenue
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 Ranknr <=3;
```

Key Insights

- Identified the highest-priced and most ordered pizza types.
- Analyzed revenue distribution and operational performance.
- Evaluated customer preferences and ordering patterns.