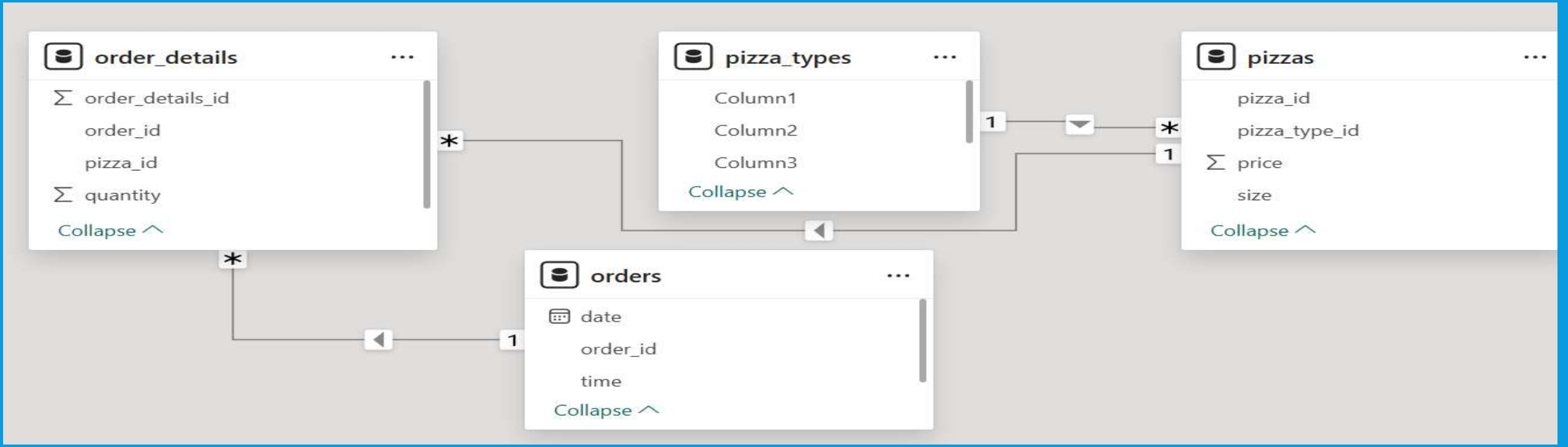
PIZZA SALES

SQL PROJECT

PIZZA SALES ANALYSIS USING SQL SERVER

This project analyzes pizza sales data using Microsoft SQL Server Management Studio (SSMS). The dataset consists of four tables: **Orders, Order_Details, Pizzas,** and **Pizza_Types,** and aims to answer key business questions across basic, intermediate, and advanced levels of complexity **SQL Server.**



KEY OBJECTIVES-

Basic Analysis:

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

Intermediate Analysis:

1. Join necessary tables to find the total quantity of each pizza category ordered.
2. Analyze the distribution of orders by hour of the day.
3. Determine the category-wise distribution of pizzas.
4. Group orders by date and calculate the average number of pizzas ordered per day.
5. Identify the top 3 most ordered pizza types based on revenue.

Advanced Analysis:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze cumulative revenue trends over time.
3. Determine the top 3 most ordered pizza types by revenue within each pizza category.

```sql
--Retrieve the total number of orders placed.

select COUNT(order_id) From orders as total_orders;
```

100 %

Results | Messages

	(No column name)
1	21350

```sql
--Calculate the total revenue generated from pizza sales.

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

Results | Messages

	total_revenue
1	817860.05

```sql
--Identify the highest-priced pizza.

select top 1 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 ;
```

100 %

Results | Messages

	name	price
1	The Greek Pizza	35.95

```sql
--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 1 ;
```

100 %

Results | Messages

	size	order_count
1	L	18526
2	M	15385
3	S	14137
4	XL	544
5	XXL	28

```sql
--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 order_details.pizza_id=pizzas.pizza_id
group by pizza_types.name
order by quantity desc
;
```

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

```sql
--Join the necessary tables to find the total quantity of each pizza category ordered.

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;
```

Results | Messages

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

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

SELECT DATEPART(HOUR, time) AS OrderHour, COUNT(order_id) as order_count
from orders
GROUP BY DATEPART(HOUR, time)
order by OrderHour desc;
```

Results | Messages

	OrderHour	order_count
1	23	28
2	22	663
3	21	1198
4	20	1642
5	19	2009
6	18	2399
7	17	2336

```sql
--Join relevant tables to find the category-wise distribution of pizza

SELECT category, COUNT(name) as count
from pizza_types
group by category
```

Results | Messages

	category	count
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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

select avg (TotalQuantity) as average_quantity
from (SELECT orders.date, SUM(order_details.quantity) AS TotalQuantity
FROM orders
JOIN order_details
ON orders.order_id = order_details.order_id
GROUP BY orders.date ) as order_quantity;
```

Results | Messages

	average_quantity
1	138

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

select top 3 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.name
order by revenue desc
```

Results | Messages

	name	revenue
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

```sql
--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_revenue
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 category
order by revenue desc;
```

Results | Messages

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

```sql
SELECT
    sales.date,
    SUM(sales.revenue) OVER (ORDER BY sales.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;
```

Results | Messages

	date	cum_revenue
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4

Query executed successfully.

```sql
--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 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<=3;
```

Results | Messages

	category	name	revenue
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768
3	Chicken	The California Chicken Pizza	41409.5
4	Classic	The Classic Deluxe Pizza	38180.5
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.5

Query executed successfully.