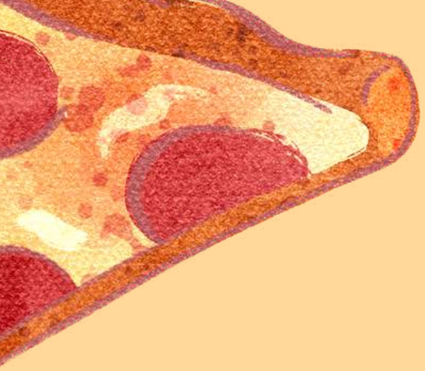


PIZZAHUT!

@AvinashParchake

Hello!

I'm Avinash Parchake, and in this project, I've leveraged SQL queries to address challenges related to pizza sales analytics. Through strategic data analysis, we've delved into the dynamics of pizza consumption, uncovering valuable insights to optimize sales performance and customer satisfaction.

Questions:

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

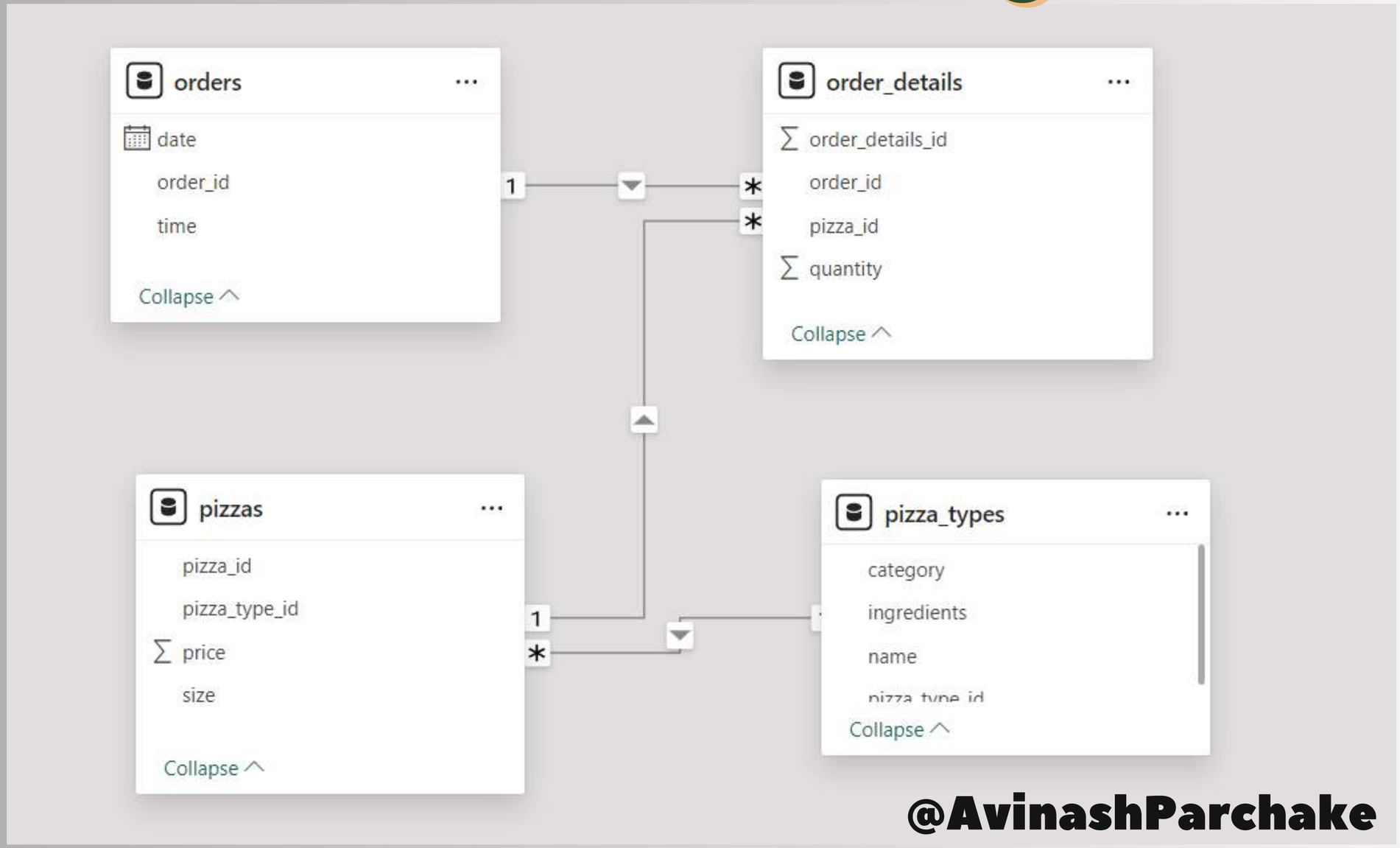
Questions:

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

Questions:

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Data Modeling



@AvinashParchake

Retrieve the total number of orders placed

```sql
SELECT
    COUNT(*) AS total_orders
FROM
    orders;
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Output:

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
select  round(sum(od.quantity * ps.price)) as total_revenue
from order_details od
inner join pizzas ps
on od.pizza_id = ps.pizza_id;
```

Output:

total_revenue
817860

Identify the highest-priced pizza.

```
select pizza_types.name, pizza_types.category,pizzas.price
from pizzas
inner join pizza_types
on pizzas.pizza_type_id =pizza_types.pizza_type_id
order by price desc
limit 1;
```

Output:

Result Grid | Filter Rows: | Export:

name	category	price
The Greek Pizza	Classic	35.95

Identify the most common pizza size ordered.

```
select pizzas.size,count(order_details.order_details_id)as ordered_count from
pizzas
inner join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size;
```

Output:

size	ordered_count
M	15385
L	18526
S	14137
XL	544
XXL	28

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY name
ORDER BY total_quantity DESC
LIMIT 5;
```

Output:

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 category ordered.

```sql
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS total_quantity
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY total_quantity DESC;
```

Output:

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

```sql
select hour(order_time) as hour, count(*) as order_count
from orders
group by hour
order by hour;
```

```sql
SELECT Extract(Hour from order_time) as hour_of_day, count(*) as total_orders
FROM orders
group by hour_of_day
order by hour_of_day;
```

Output:

hour	order_count
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28

Join relevant tables to find the category-wise distribution of pizzas.

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

Output:

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

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

```sql
SELECT
    ROUND(AVG(average_pizzas_ordered)) AS avg_pizza_orders
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS average_pizzas_ordered
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS daily_orders;
```

Output:

	avg_pizza_orders
▶	138

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

```sql
select pizza_types.name as pizza_type , round(sum(order_details.quantity * pizzas.price)) as revenue
from pizzas
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_type
order by revenue desc
limit 3;
```

Output:

pizza_type	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Calculate the percentage contribution of each 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 order_details.pizza_id = pizzas.pizza_id) * 100,2) as revenue
from pizzas
join pizza_types
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.category
order by revenue desc;
```

Output:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```sql
select order_date,
round(sum(revenue) over(order by order_date)) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from
pizzas
join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on order_details.order_id = orders.order_id
group by orders.order_date) as sales;
```

Output:

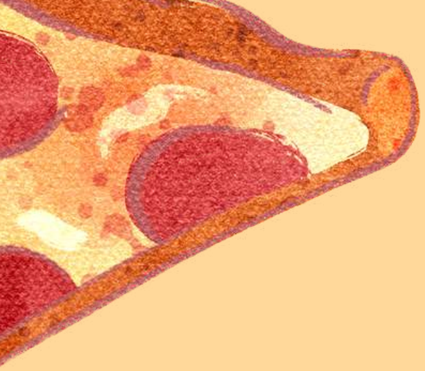
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Determine the top 3 most 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 <=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

Output:



THANK YOU

@AvinashParchake