



SQL PROJECT ON PIZZA SALES

Kanupriya Nailwal



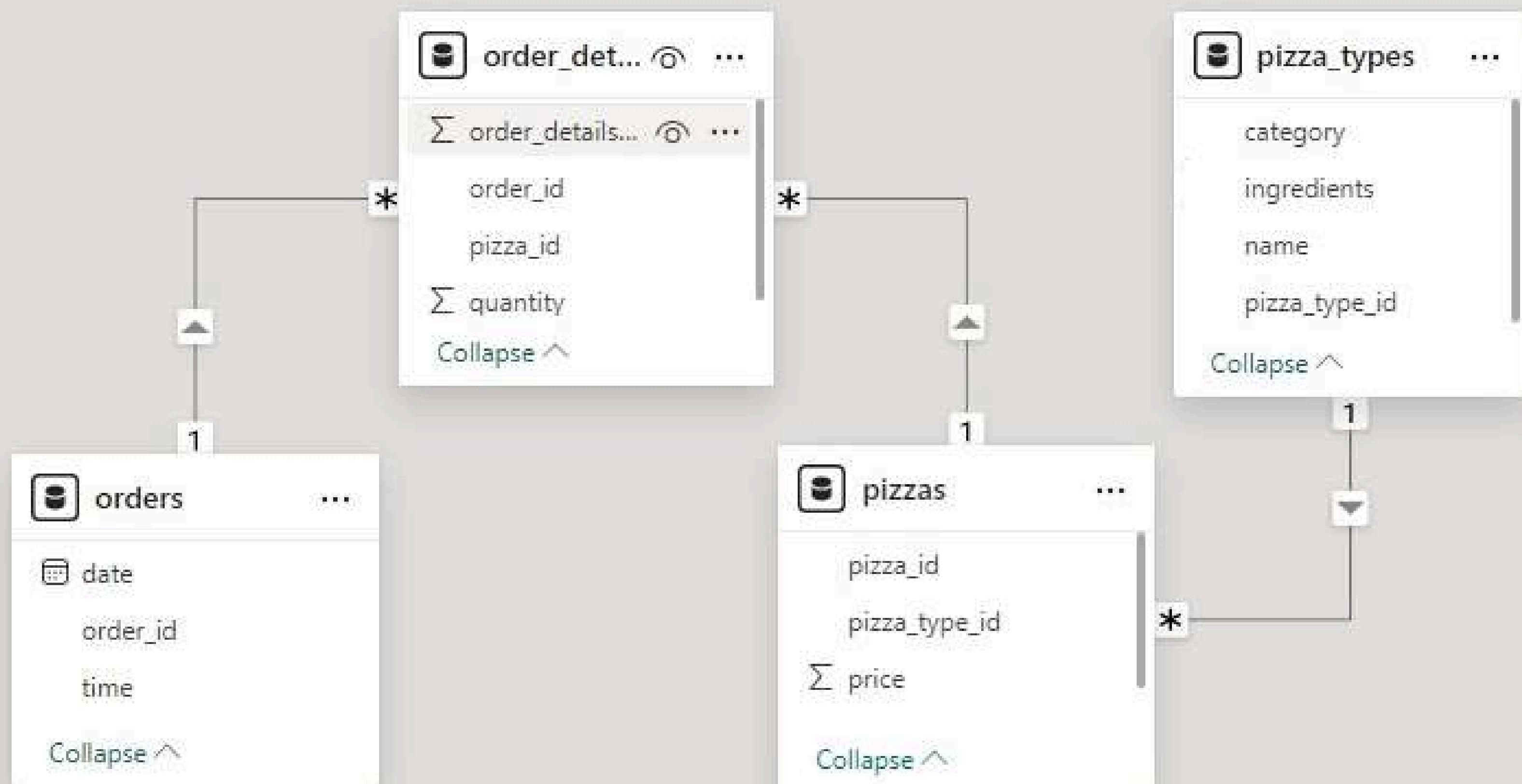


HELLO !

My name is Kanupriya Nailwal. In this project I have utilize SQL query to solve questions that were related to pizza sales.

About The dataset

- This dataset contains **4 tables** in CSV format.
- The **Orders** table contains the date & time that all table orders were placed.
- The **Order Details** table contains the different pizzas served with each order in the Orders table, and their quantities.
- The **Pizzas** table contains the size and price for each distinct pizza in the Order Details, as well as its broader pizza type.
- The **Pizza Types** table contains details on the pizza types in the Pizzas table, including their name as it appears on the menu, the category it falls under, and its list of ingredients.



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



```
Select count(order_id) as Total_order from orders;
```

Result Grid	
	Total_order
▶	21350

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;
```

Result Grid	
	Total_revenue
▶	817860.05

IDENTIFY THE HIGHEST PRICE PIZZA.



```
select 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  
limit 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



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

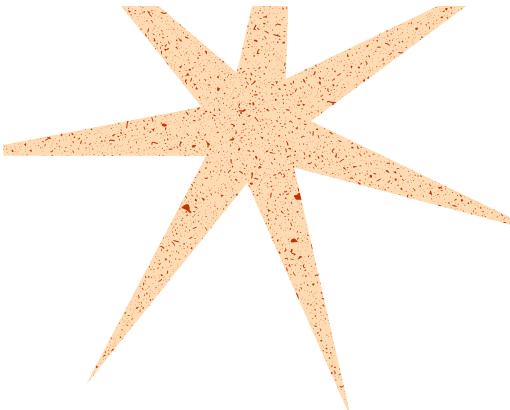
Result Grid | Filter Rows:

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
select pizza_types.name as pizza_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_name
order by Quantity desc
limit 5;
```



Result Grid | Filter Rows:

	pizza_name	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.



```
select pizza_types.category as Pizza_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 category  
order by quantity desc;
```



Result Grid | Filter Rows:

	Pizza_category	Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



DETERMINE THE DISTRIBUTION OF ORDERS BY HOURS OF THE DAY.

```
select hour(order_time) as Hours, count(order_id) as Order_count from orders  
group by hour(order_time)  
order by Order_count desc;
```

	Hours	Order_count
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642
	14	1472
	15	1468
	11	1231
	21	1198
	22	663
	23	28
	10	8
	9	1

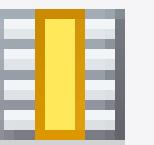
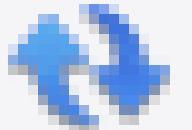
JOINT RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS.

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

Result Grid		
	category	Pizza
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
select round(avg(Quantity),0) as Avg_pizza_ordered_per_day from  
→ (select orders.order_date , sum(order_details.quantity) as Quantity  
from orders join order_details  
on orders.order_id = order_details.order_id  
group by orders.order_date ) as Order_quantity ;
```

	Result Grid		 Filter Rows:
	Avg_pizza_ordered_per_day		
▶	138		

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
select 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 limit 3;
```

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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE OF TOTAL REVENUE.

```
select pizza_types.category as Category,  
round(sum(order_details.quantity*pizzas.price) / (select sum(order_details.quantity*pizzas.price) 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;
```

Result Grid | Filter Rows:

	Category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
Select order_date, sum(revenue) over (order by order_date) as cum_revenue  
from  
(Select orders.order_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.order_date) as Sales
```

	order_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	23990.35000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.30000000003
	2015-01-14	32358.70000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.75000000001
	2015-01-18	40978.60000000006
	2015-01-19	43365.75000000001
	2015-01-20	45763.65000000001

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
select rn, name, revenue from (Select rank() over(partition by pizza_types.category order by sum(order_details.quantity * pizzas.price) desc) as rn, 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 t
where rn <=3;
```

	rn	name	revenue
▶	1	The Thai Chicken Pizza	43434.25
	2	The Barbecue Chicken Pizza	42768
	3	The California Chicken Pizza	41409.5
	1	The Classic Deluxe Pizza	38180.5
	2	The Hawaiian Pizza	32273.25
	3	The Pepperoni Pizza	30161.75
	1	The Spicy Italian Pizza	34831.25
	2	The Italian Supreme Pizza	33476.75
	3	The Sicilian Pizza	30940.5
	1	The Four Cheese Pizza	32265.70000000065
	2	The Mexicana Pizza	26780.75
	3	The Five Cheese Pizza	26066.5

THANK YOU!

< BACK