

PIZZA SALES PROJECT

using “my sql”

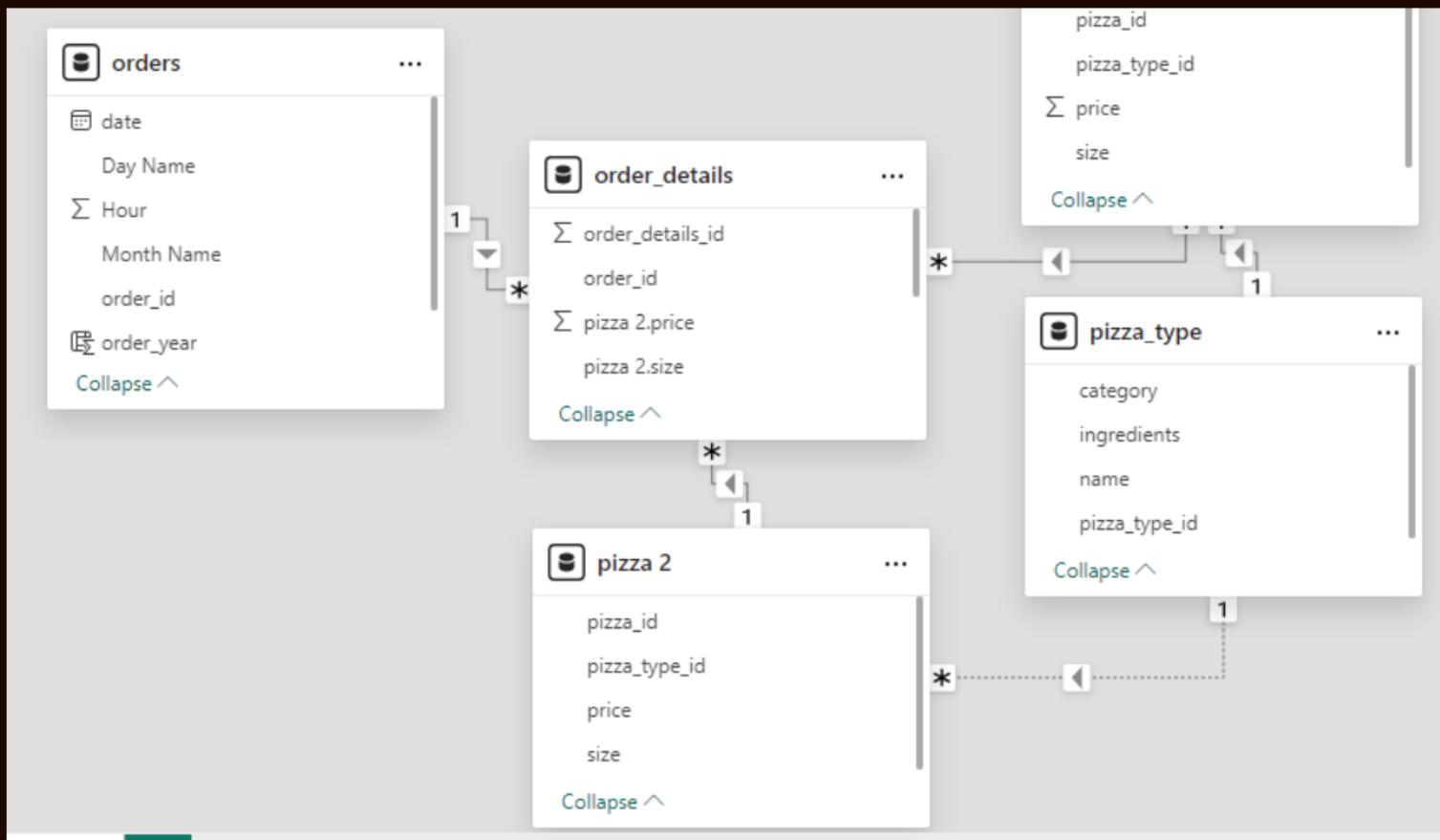




ABOUT ME

i am neha gupta, i am data analyst and here is my data analyst project using mysql.

DATA MODEL VIEW



1 Retrieve the total number of orders placed

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

Result Grid	
	total_orders
▶	21350



2 calculate the total revenue generated from pizza sales.

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

Result Grid	
	total_sales
▶	817860.05



3 identify the highest priced pizza

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

Result Grid		
	name	price
▶	The Greek Pizza	35.95



4 identify the most common pizza size ordered.

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

Result Grid		
	size	total
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



5 list the top 5 most ordered pizza types along with their quantity.

```
select t.name, sum(o.quantity) as total_quantity
from order_details as o
join pizzas as p
on p.pizza_id = o.pizza_id
join pizza_types as t
on p.pizza_type_id = t.pizza_type_id
group by t.name
order by total_quantity desc
limit 5;
```

	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

6 join the necessary tables to find the total quantity of each pizza catagory ordered.

```
select t.category, sum(o.quantity) as total_quantity
from order_details as o
join pizzas as p
on p.pizza_id = o.pizza_id
join pizza_types as t
on p.pizza_type_id = t.pizza_type_id
group by t.category
order by total_quantity desc;
```

Result Grid		Filter Rows:
	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

7 determine the distribution of orders by hour of the day

```
select hour(time), count(order_id)  
from orders  
group by hour(time);
```



	hour(time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

8 join relevant tables to find the category wise distribution of pizza

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

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

9 group the orders by date and calculate the average number of pizzas ordered per day

```
select round(avg(quantity),0) from
(select o.date, sum(d.quantity) as quantity
from orders as o
join order_details as d
on o.order_id = d.order_id
group by date
order by sum(d.quantity) desc) as order_quantity;
```

	round(avg(quantity),0)
▶	138



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

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

Result Grid | Filter Rows:

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

11 analyze thr cumulative revenue generated over time

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

Result Grid		Filter Rows:
	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	23990.350000000002

12 determine the top 3 most ordered pizza type based on revenue for each pizza catagory.

```
select name, revenue, category from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select t.name, t.category, sum(d.quantity * p.price) as revenue
from order_details as d
join pizzas as p
on d.pizza_id = p.pizza_id
join pizza_types as t
on p.pizza_type_id = t.pizza_type_id
group by t.name, t.category) as a) as b
where rn<= 3;
```



	name	revenue	category
▶	The Thai Chicken Pizza	43434.25	Chicken
	The Barbecue Chicken Pizza	42768	Chicken
	The California Chicken Pizza	41409.5	Chicken
	The Classic Deluxe Pizza	38180.5	Classic
	The Hawaiian Pizza	32273.25	Classic
	The Pepperoni Pizza	30161.75	Classic
	The Spicy Italian Pizza	34831.25	Supreme
	The Italian Supreme Pizza	33476.75	Supreme
	The Sicilian Pizza	30940.5	Supreme
	The Four Cheese Pizza	32265.70000000065	Veggie



CHECK MY
PORTFOLIO

<https://github.com/56801>

THANK YOU
FOR
WATCHING

