

Delicious Pizza for Everyone!

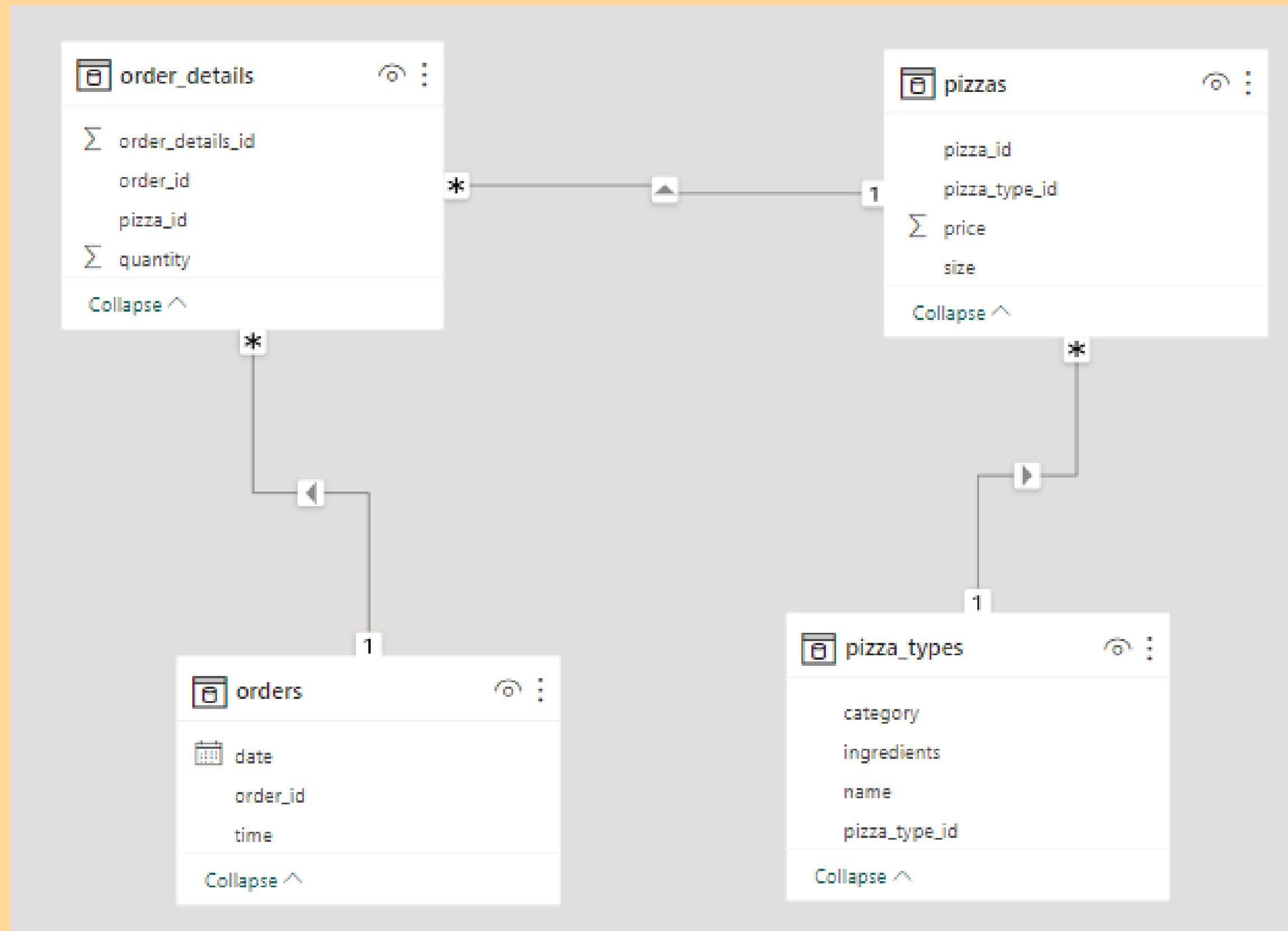
PIZZA SALES PROJECT

@suraj rajak

objective

The Pizza Sales Project using **SQL** aimed to analyze sales data from a fictional pizza restaurant chain to gain insights into customer preferences, sales trends, and performance of different menu items. Leveraging **SQL** queries, the project extracted and manipulated data from a relational database containing information such as orders, customers, products, and transactions.

Data Model



Retrieve the total number of orders placed.

```
SELECT count(order_ID) FROM orders;
```

Result Grid	
	count(order_ID)
▶	21350

Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(orders_details.quantity * pizzas.price),2) AS total_revenue  
FROM  
    pizzas  
    JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
```

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

```
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY price DESC  
LIMIT 1
```

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
SELECT  
    pizzas.size,  
    COUNT(orders_details.order_details_id) AS most_common  
FROM  
    pizzas  
        JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY most_common DESC
```

	size	most_common
▶	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, SUM(orders_details.quantity) AS most_order  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY most_order DESC  
LIMIT 5
```

	name	most_order
▶	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,  
    SUM(orders_details.quantity) AS total_quantity  
FROM  
    orders_details  
        JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
        JOIN  
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
GROUP BY pizza_types.category
```

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT  
    HOUR(order_time), COUNT(order_ID)  
FROM  
    orders  
GROUP BY HOUR(order_time)
```

HOUR(order_time)	COUNT(order_ID)
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

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

```
SELECT  
    category, COUNT(pizza_type_id)  
FROM  
    pizza_types  
GROUP BY category
```

	category	COUNT(pizza_type_id)
▶	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)  
FROM  
    (SELECT  
        orders.order_date, SUM(orders_details.quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_ID = orders_details.order_id  
    GROUP BY orders.order_date) AS order_quantity
```

	avg_number
▶	138



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



```
SELECT  
    pizza_types.name,  
    SUM(orders_details.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON orders_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 to total revenue!

```
SELECT  
    pizza_types.category,  
    round((SUM(orders_details.quantity * pizzas.price)/(SELECT  
        ROUND(SUM(orders_details.quantity * pizzas.price),2)  
FROM  
    pizzas  
        JOIN  
    orders_details ON pizzas.pizza_id = orders_details.pizza_id))*100,2) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category  
order BY revenue DESC
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time.

```
select order_date,  
round(sum(revenue) over(order by order_date),2) as cum_revenue  
from  
(SELECT  
    orders.order_date,  
    SUM(orders_details.quantity * pizzas.price) AS revenue  
FROM  
    orders  
    JOIN  
    orders_details ON orders.order_ID = orders_details.order_id  
    JOIN  
    pizzas ON pizzas.pizza_id = orders_details.pizza_id  
GROUP BY orders.order_date) as sales
```

order_date	cum_revenue
2015-01-01	2713.85
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	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65

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

```
select category, name, revenue,  
row_number() over (partition by category order by revenue desc) as rn  
from  
(select category, name, revenue,  
row_number() over (partition by category order by revenue desc) as rn  
from  
(SELECT  
    pizza_types.category, pizza_types.name,  
    SUM(orders_details.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
        JOIN  
    orders_details ON orders_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.category, pizza_types.name) as a ) as b  
where rn<=3
```

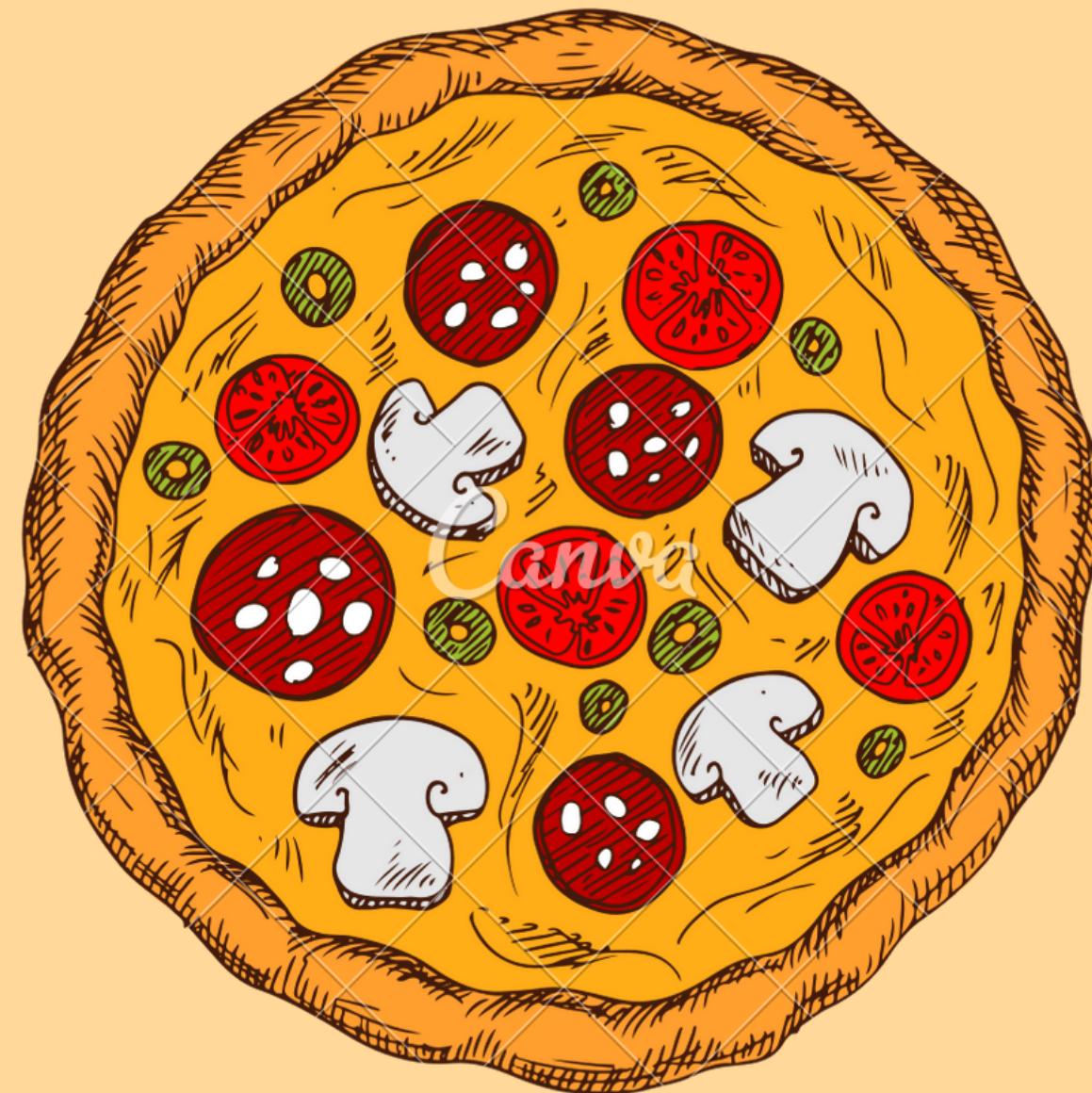
category	name	revenue	rn
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian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

Call us for Order



Phone Number:
+91-8617768219

Linkedin:
www.linkedin.com/in/suraj-rajak



Pizza Boxcar Present

**THANK
YOU**

@suraj rajak