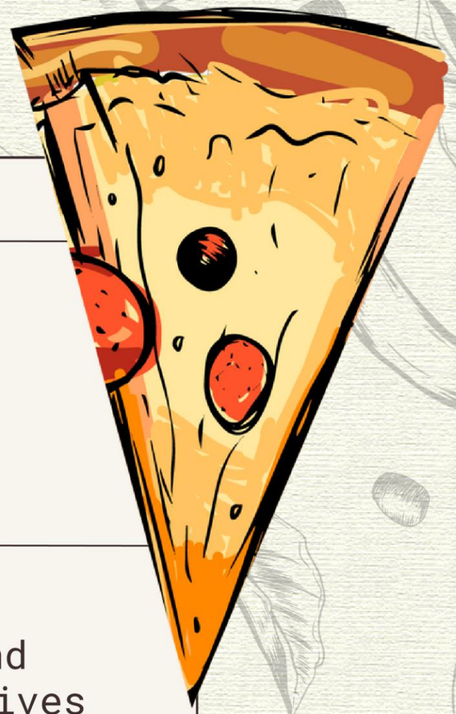
Hello
This is SQL Project on Pizza Sales

By Kuldeep Vighane

Here the presentation begins

HELLO I’m Kuldeep

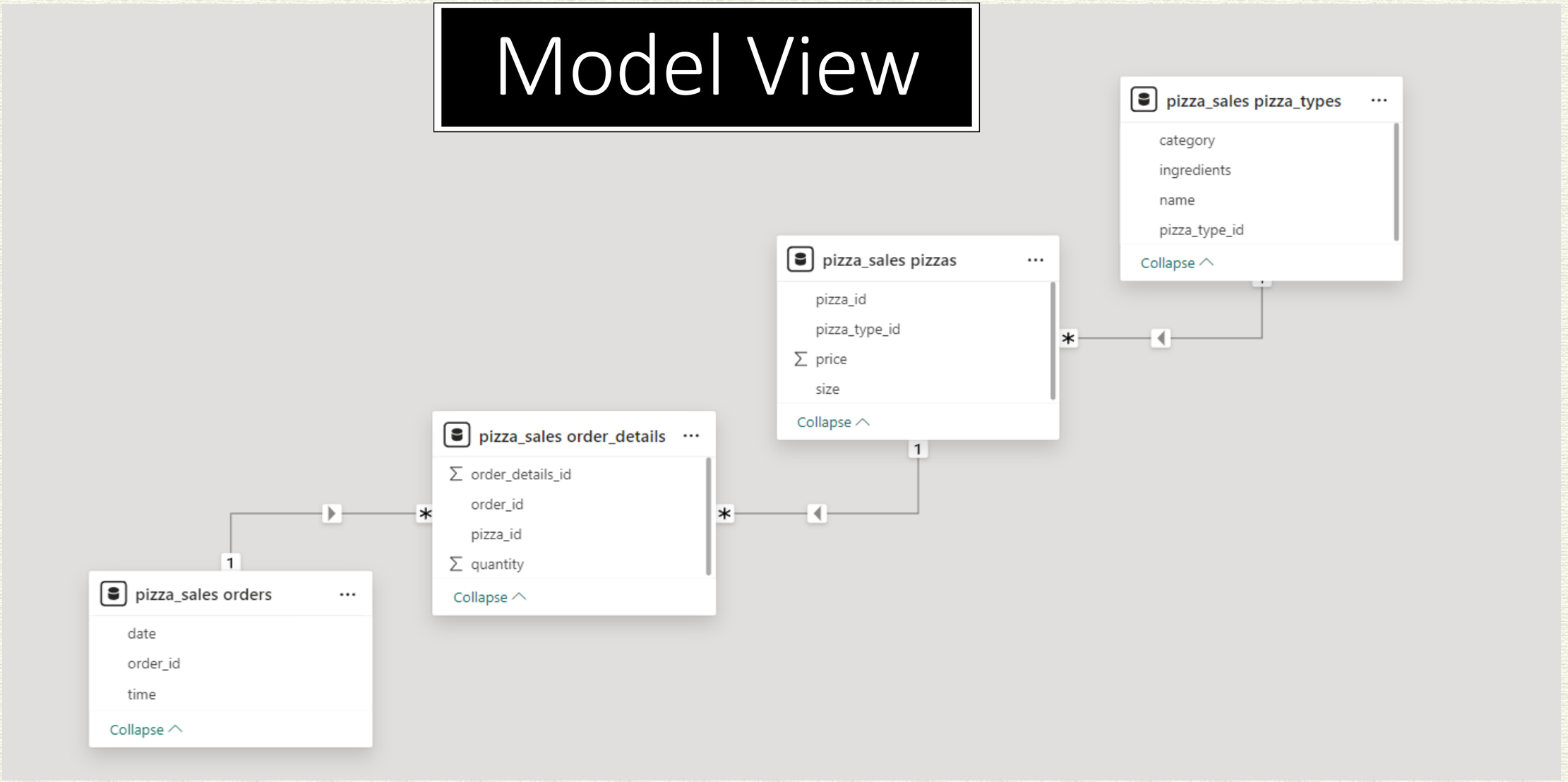
I'm excited to share my latest project, a deep dive into pizza sales analysis using SQL. As a student passionate about data analytics, this project highlights my SQL skills and ability to derive meaningful insights from complex datasets. Join me as I showcase my proficiency in querying, data manipulation, and visualization, offering valuable perspectives for businesses in the food industry. Welcome to the unveiling of pizza sales insights: a testament to my commitment to mastering the art of data-driven decision-making.



OBJECTIVES

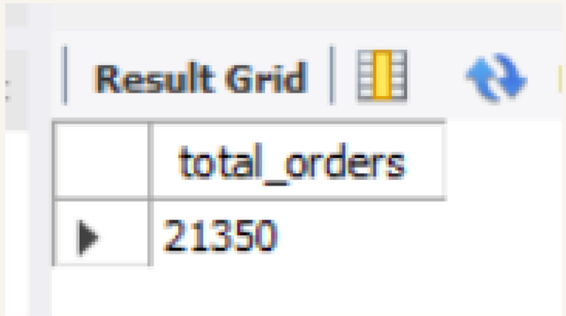
This project aims to showcase proficiency in SQL querying and analytical skills by delving into pizza sales data. Key objectives include retrieving total orders and calculating revenue to understand customer demand and financial performance. Additionally, identifying pricing trends through the highest-priced pizza and analyzing size preferences will inform inventory management and menu decisions. Further, undertaking deeper analysis, such as order distribution and revenue trends, will uncover nuanced insights, demonstrating a comprehensive understanding of data analytics techniques.

Model View


pizza_sales pizza_types
category
ingredients
name
pizza_type_id
Collapse
pizza_sales pizzas
pizza_id
pizza_type_id
price
size
Collapse
pizza_sales order_details
order_details_id
order_id
pizza_id
quantity
Collapse
pizza_sales orders
date
order_id
time
Collapse
1
*
1
*
*

Retrieve the total number of orders placed.

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



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 order_details.pizza_id = pizzas.pizza_id;
```

Result Grid

	total_revenue
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.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.

Result Grid | Filter Rc

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

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

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS quantity_order
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_order DESC
LIMIT 5;
```

name	quantity_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.

```sql
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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(orders.time), SUM(orders.order_id) AS quantity
FROM
    orders
GROUP BY HOUR(orders.time)
ORDER BY HOUR(orders.time);
```

hour(orders.time)	quantity
9	19176
10	73999
11	13336362
12	26929470
13	26615205
14	14867592
15	15634879
16	20551671
17	24312547
18	25808745
19	21634044
20	17668990
21	12868673
22	7269872
23	330700

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

Analyze the cumulative revenue generated over time.

```
SELECT date , sum(revenue) OVER (order by date)
FROM
    (SELECT orders.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 orders.order_id = order_details.order_id
    GROUP BY orders.date) AS sales;
```

	date	sum(reve date)
▶	2015-01-01	2713.850
	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

Calculate the percentage contribution of each pizza type to total revenue.

```sql
SELECT pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                    ROUND(SUM(pizzas.price * order_details.quantity),
                FROM
                    pizzas
                        JOIN
                    order_details ON pizzas.pizza_id = order_details.pizza_id) * 100,
            2) AS revenue_in_per
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;
```

category	revenue_in_per
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

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

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

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

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

```sql
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 pizzas.pizza_id = order_details.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

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_qnt_order_per_dayy
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS avg_order;
```

avg_qnt_order_per_dayy
138

THANKS!

Do you have any questions?
nishupvighane@gmail.com
LinkedIN:
https://www.linkedin.com/in/kuldeep-vighane-11bb47309/

made with <3 by Kuldeep