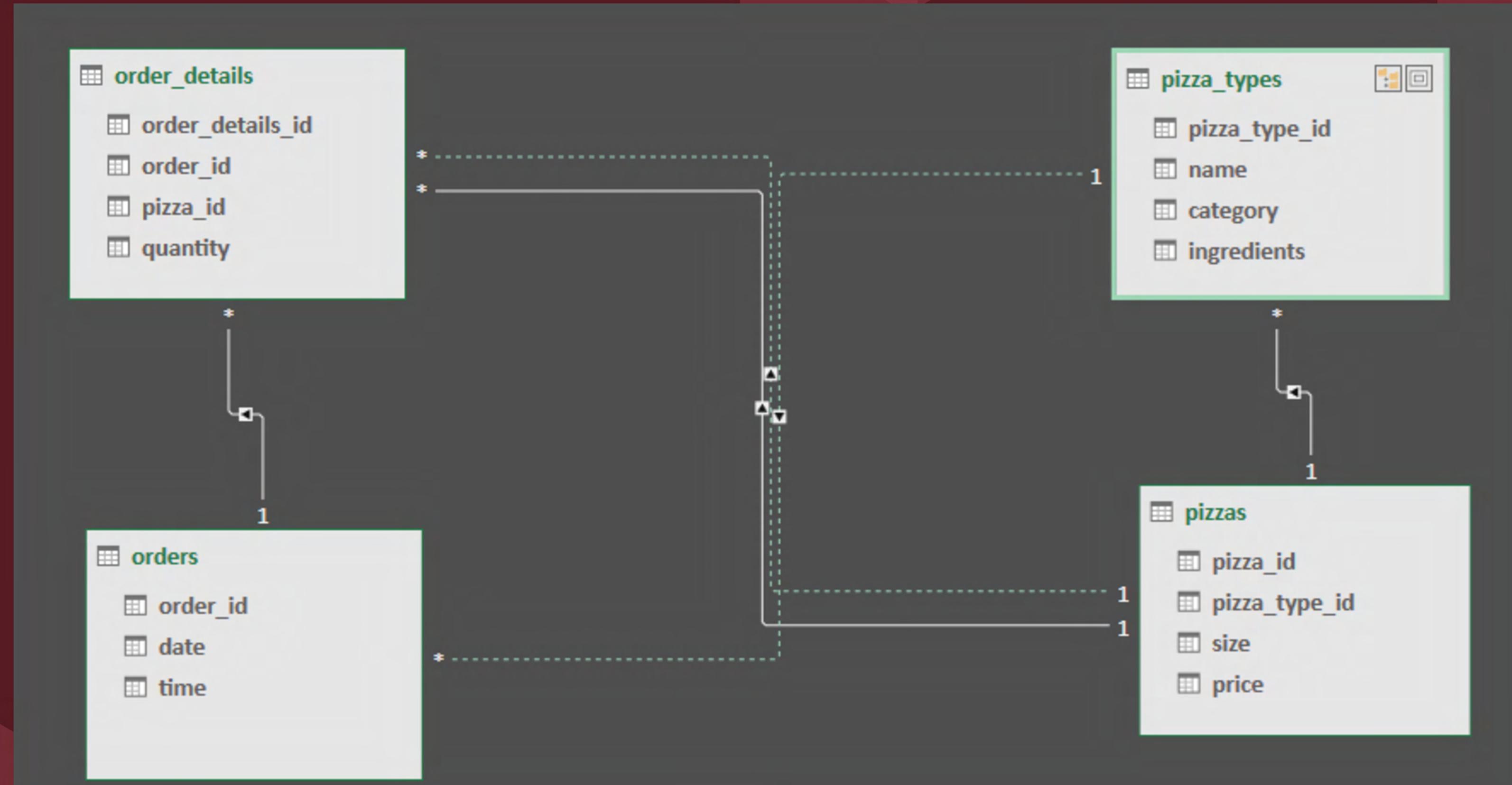


PIZZA SALES REPORT

09, Feb 2025

INTRODUCTION

Hi, This Is Vaibhav Satras,
I have get small Introduction about my Projct
This Is Pizaa Sales Report,
In That Project I have Utilize some SQL
Queries to analyse and evaluate the Sales
Report.



SUMMARY

This summary provides a concise snapshot of to Perform few SQL queries to represents and analyse the sales report, and it clearly visualize thorugh this queries

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
    TRUNCATE(SUM(order_details.quantity * pizzas.price),
              2) AS 'Total Revenue'
  FROM
    order_details
    CROSS JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

| Result Grid | |
|-------------|---------------|
| | Total Revenue |
| ▶ | 817860.04 |

Identify the highest-priced pizza.

```
SELECT
    pizza_types.name, SUM(pizzas.price) AS 'Price'
FROM
    pizzas
        CROSS JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY Price DESC
LIMIT 1;
```

.....

Result Grid | Filter Rows

| | name | Price |
|---|-----------------|--------|
| ▶ | The Greek Pizza | 109.95 |

Identify the most common pizza size ordered.

```
SELECT
    pizzas.size, COUNT(orders.order_id) as 'Pizza Ordered Count'
FROM
    pizzas
        CROSS JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        CROSS JOIN
    orders ON orders.order_id = order_details.order_id
GROUP BY pizzas.size;
```

| | size | Pizza Ordered Count |
|---|------|---------------------|
| ▶ | M | 15385 |
| | L | 18526 |
| | S | 14137 |
| | XL | 544 |
| | XXL | 28 |

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

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

| | 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,
    SUM(order_details.quantity) AS 'Total Quantity'
FROM
    pizza_types
        CROSS JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        CROSS JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_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(orders.time) AS 'Hours', COUNT(orders.order_id) 'Order Count'  
FROM  
    orders  
GROUP BY Hours limit 5;
```

| | Hours | Order Count |
|---|-------|-------------|
| ▶ | 11 | 1231 |
| | 12 | 2520 |
| | 13 | 2455 |
| | 14 | 1472 |
| | 15 | 1468 |

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

```
SELECT
    pizza_types.category,
    COUNT(pizzas.pizza_id) AS 'Count Of Pizzas'
FROM
    pizza_types
        CROSS JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category;
```

| | category | Count Of Pizzas |
|---|----------|-----------------|
| ▶ | Chicken | 18 |
| | Classic | 26 |
| | Supreme | 25 |
| | Veggie | 27 |

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

```
SELECT
    orders.date,
    COUNT(order_details.order_id) AS 'Noumber Of orders',
    TRUNCATE(AVG(order_details.quantity), 2) AS 'Average Of Pizzas'
FROM
    orders
        CROSS JOIN
    order_details ON orders.order_id = order_details.order_id
        CROSS JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.date;
```

| | date | Noumber Of orders | Average Of Pizzas |
|---|------------|-------------------|-------------------|
| ▶ | 2015-01-01 | 161 | 1.00 |
| | 2015-01-02 | 160 | 1.03 |
| | 2015-01-03 | 154 | 1.02 |
| | 2015-01-04 | 106 | 1.00 |
| | 2015-01-05 | 121 | 1.03 |
| | 2015-01-06 | 144 | 1.02 |
| | 2015-01-07 | 133 | 1.03 |
| | 2015-01-08 | 171 | 1.01 |
| | 2015-01-09 | 123 | 1.03 |

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

```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS 'revenue'
FROM
    pizzas
        CROSS JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
        CROSS JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

| date | Noumber Of orders | Average Of Pizzas |
|------------|-------------------|-------------------|
| 2015-01-01 | 161 | 1.00 |
| 2015-01-02 | 160 | 1.03 |
| 2015-01-03 | 154 | 1.02 |
| 2015-01-04 | 106 | 1.00 |
| 2015-01-05 | 121 | 1.03 |
| 2015-01-06 | 144 | 1.02 |
| 2015-01-07 | 133 | 1.03 |
| 2015-01-08 | 171 | 1.01 |

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

```
SELECT
    pizza_types.name,
    TRUNCATE((SUM(order_details.quantity * pizzas.price) / COUNT(order_details.quantity)),
    2) AS 'Total_Revenue'
FROM
    pizzas
        CROSS JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        CROSS JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.name;
```

| name | Total_Revenue |
|----------------------------------|---------------|
| The Hawaiian Pizza | 13.61 |
| The Classic Deluxe Pizza | 15.8 |
| The Five Cheese Pizza | 19.18 |
| The Italian Supreme Pizza | 18.1 |
| The Mexicana Pizza | 18.39 |
| The Thai Chicken Pizza | 18.76 |
| The Prosciutto and Arugula Pizza | 16.94 |
| The Barbecue Chicken Pizza | 18.03 |

Analyze the cumulative revenue generated over time.

```
select date, truncate(sum(Revenue) over(order by date),2) as 'Cumi_revne'  
from (select orders.date,(truncate(sum(pizzas.price * order_details.quantity),2)) as 'Revenue' from pizzas  
cross join order_details on pizzas.pizza_id = order_details.pizza_id  
cross join orders on orders.order_id = order_details.order_id group by orders.date) as sales;
```

| date | Cumi_revne |
|------------|------------|
| 2015-01-01 | 2713.85 |
| 2015-01-02 | 5445.74 |
| 2015-01-03 | 8108.14 |
| 2015-01-04 | 9863.59 |
| 2015-01-05 | 11929.53 |
| 2015-01-06 | 14358.47 |
| 2015-01-07 | 16560.67 |
| 2015-01-08 | 19399.01 |

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

```
select category, 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
cross join pizzas on pizzas.pizza_type_id = pizza_types.pizza_type_id
cross join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as b where RN <4 ;
```

| category | name | revenue |
|----------|------------------------------|----------|
| Chicken | The Thai Chicken Pizza | 43434.25 |
| Chicken | The Barbecue Chicken Pizza | 42768 |
| Chicken | The California Chicken Pizza | 41409.5 |
| Classic | The Classic Deluxe Pizza | 38180.5 |
| Classic | The Hawaiian Pizza | 32273.25 |
| Classic | The Pepperoni Pizza | 30161.75 |
| Supreme | The Spicy Italian Pizza | 34831.25 |
| Supreme | The Italian Supreme Pizza | 33476.75 |

THANK YOU

09 Feb 2025