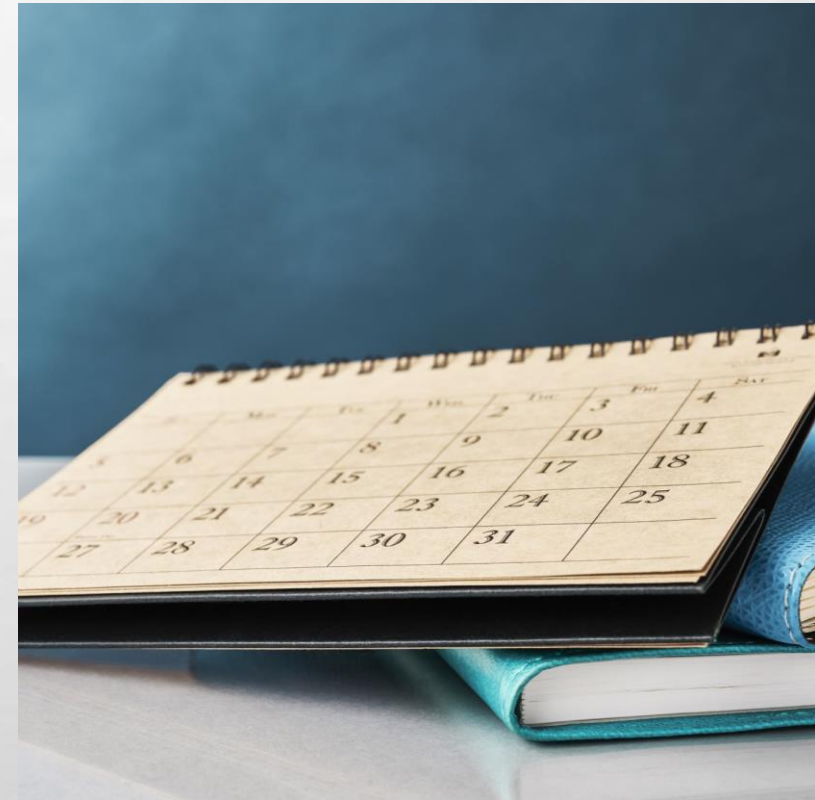


PIZZA SALES SQL PROJECT FOR DATA ANALYSIS



PROJECT DESCRIPTION

- **OBJECTIVE:** UTILIZE SQL QUERIES TO ANALYZE PIZZA SALES DATA.
- **GOAL:** DETERMINE PIZZA SALES AND PERFORM SALES ANALYSIS.
- **KEY TASKS:**
 - EXTRACT RELEVANT DATA FROM THE DATABASE.
 - CALCULATE TOTAL SALES, POPULAR PIZZA TYPES, AND TRENDS.
 - CREATE VISUALIZATIONS TO COMMUNICATE INSIGHTS.



SCHEMA

SQL QUESTION OF PIZZA SALES

----- Retrieve the total number of orders placed

```
SELECT  
    count(order_id) AS total_number_orders_placed  
FROM  
    orders;
```

result Grid		Filter Rows:
	total_number_orders_placed	
	21350	

SQL QUESTION OF PIZZA SALES

----- Calculate the total revenue generated from pizza sales -----

```
SELECT  
round(SUM(order_details.quantity * pizzas.price),2) as pizza_price  
FROM order_details JOIN pizzas  
ON order_details.pizza_id = pizzas.pizza_id  
ORDER BY pizza_price DESC;
```

Result Grid | 

	pizza_price
▶	817860.05

SQL QUESTION OF PIZZA SALES

----- 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 pizzas.price DESC
LIMIT 1;
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	

SQL QUESTION OF PIZZA SALES

----- Identify the most common pizza size ordered-----

```
SELECT
    quantity, COUNT(order_details_id) AS most_common_pizza
FROM
    order_details
GROUP BY quantity;
```

Result Grid   Filter Rows:		
	quantity	most_common_pizza
▶	1	47693
	2	903
	3	21
	4	3

SQL QUESTION OF PIZZA SALES

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_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.name
ORDER BY total_quantity DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	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	

SQL QUESTION OF PIZZA SALES

----- Join the necessary tables to find the total quantity of each pizza category ordered-----

```
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;
```

Result Grid			Filter
	category	quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

SQL QUESTION OF PIZZA SALES

----- Determine the distribution of orders by hour of the day-----

```
SELECT
```

```
    HOUR(order_time), COUNT(order_id) AS total_distribution
```

```
FROM
```

```
    orders
```

```
GROUP BY HOUR(order_time);
```

Result Grid			Filter Rows:
	hour(order_time)	total_distribution	
▶	11	1231	
	12	2520	
	13	2455	
	14	1472	
	15	1468	
	16	1920	
	17	2336	
	18	2399	
	19	2009	
	--	--	

SQL QUESTION OF PIZZA SALES

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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

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

SQL QUESTION OF PIZZA SALES

----- 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;
```

	avg_pizza_ordered_per_day
▶	138

SQL QUESTION OF PIZZA SALES

----- 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 pizzas.pizza_type_id = pizza_types.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;
```

Result Grid			Filter Rows:
	name	revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	

SQL QUESTION OF PIZZA SALES

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

```
select pizza_types.category,  
(sum(order_details.quantity * pizzas.price)/ (select round(SUM(order_details.quantity * pizzas.price),2) as total_sales  
FROM order_details JOIN pizzas  
ON order_details.pizza_id = pizzas.pizza_id) ) *100 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  
ORDER BY revenue desc;
```

Result Grid			Filter Rows:
	category	revenue	
▶	Classic	26.90596025566967	
	Supreme	25.45631126009862	
	Chicken	23.955137556847287	
	Veggie	23.682590927384577	

SQL QUESTION OF PIZZA SALES

----- 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	23990.350000000002
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.300000000003

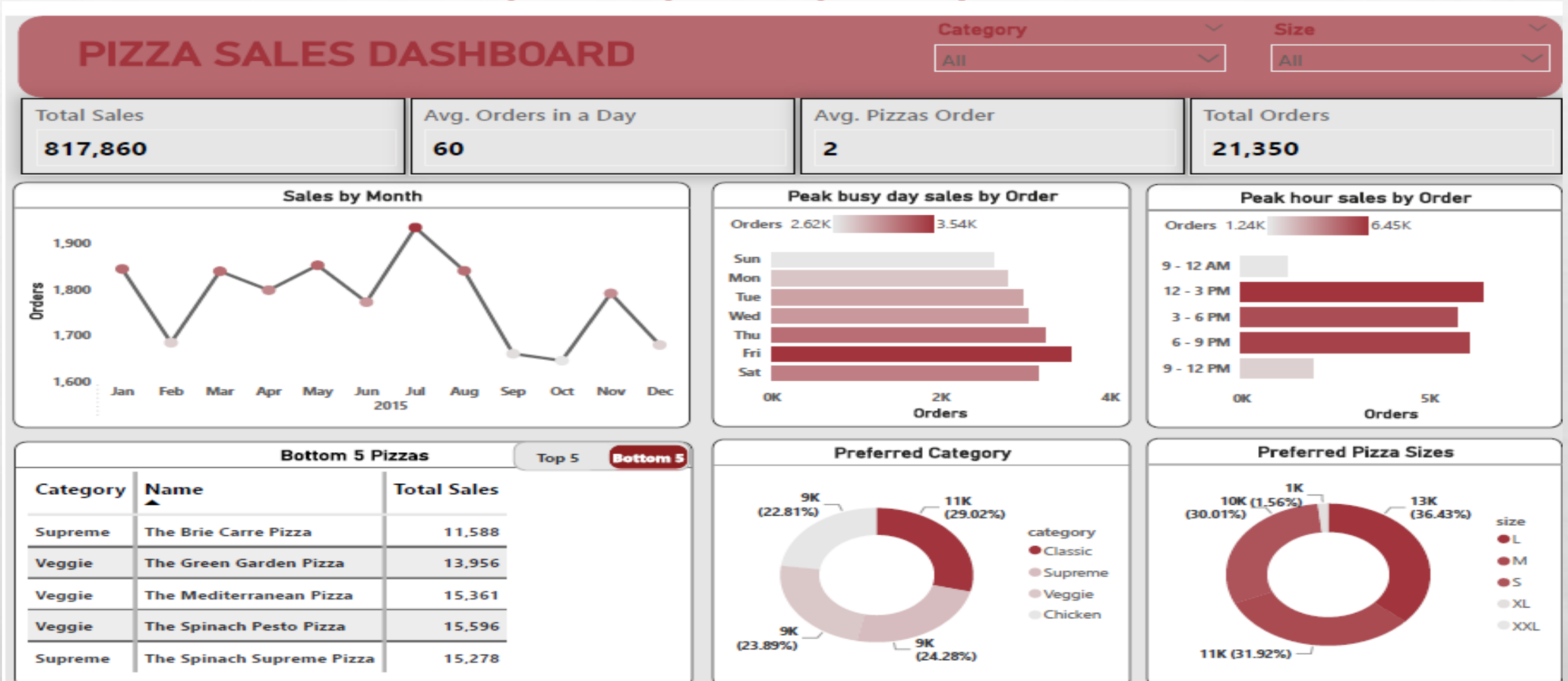
SQL QUESTION OF PIZZA SALES

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

```
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;
```

Result Grid			Filter Rows:
	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	
	The Hawaiian Pizza	32273.25	
	The Pepperoni Pizza	30161.75	
	The Spicy Italian Pizza	34831.25	
	The Italian Supreme Pizza	33476.75	
	The Sicilian Pizza	30940.5	
	The Four Cheese Pizza	32265.70000000065	
	The Mexicana Pizza	26780.75	
	The Five Cheese Pizza	26066.5	

PIZZA SALES DASHBOARD



THANK YOU

SURESH THAPA

EXCEL MAP SOLUTION INC. ¹⁸

