Name: - Vishal Patel

Gmail: - vishal.patelssm1992@gmail.com

**Project: - SQL Project On Pizzas Sales Analysis** 

### Question

1. Retrieve the total number of orders placed.

```
SELECT

COUNT(order_id) AS total_orders

FROM

orders;
```



2. calculate the total revenue generated from pizza sales

```
SELECT

ROUND(SUM(order_details.quantity * pizzas.price),

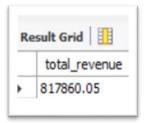
2) AS total_revenue

FROM

order_details

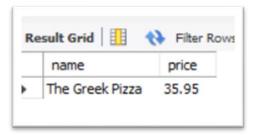
JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

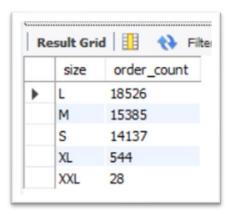


# 3. Identify the heighest-priced pizza.

```
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;
```

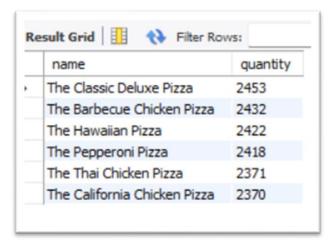


# 4. Identify the most common pizza size ordered.



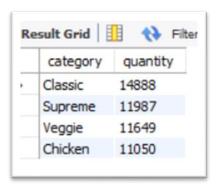
5. List the top most 5 ordered pizza type along with their quantities.

```
SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 6;
```



6. 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;
```



7. Determine the distribution of orders by hour of the day.

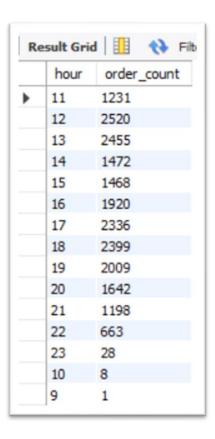
```
SELECT

HOUR(time) AS hour, COUNT(order_id) AS order_count

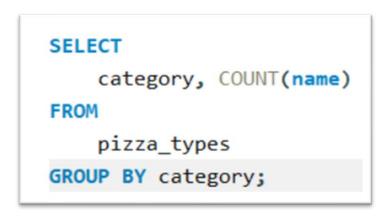
FROM

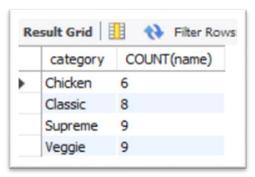
orders

GROUP BY HOUR(time);
```



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





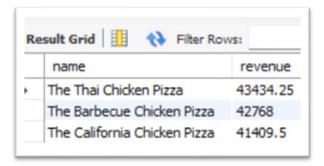
9. Group the order by data and calculate the average number of pizzas ordered per day.

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
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 order_quantity;
```



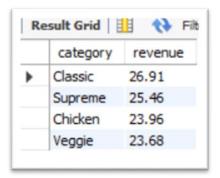
### 10. Determine the top 3 most ordered pizza types based on revenue.

```
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;
```



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

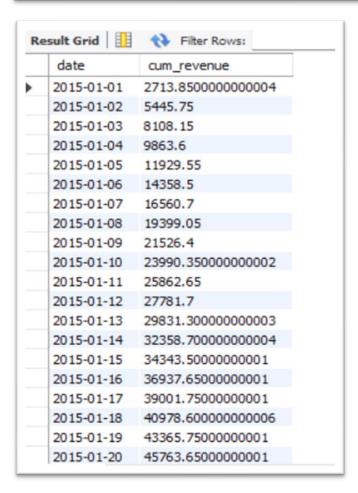
```
SELECT
   pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2) AS total_revenue
                FROM
                    order_details
                       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



12. Analyse the cumulative revenue generated over time.

```
select date,
sum(revenue) over(order by date) as cum_revenue
from

(select orders.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.date) as sales;
```



13. Determine the top 3 most ordered pizza type 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
  on order_details.pizza_id=pizzas.pizza_id
  group by pizza_types.category,pizza_types.name) as a) as b
  where rn <=3;</pre>
```

	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