PIZZAHUT SALES ANALAYSIS USING SQL

Objective: To analyze PizzaHut's sales data using SQL to gain insights on performance, customer

preferences, and revenue

Tools: SQL (MySQL), CSV files

Focus Areas: Order volume and timing

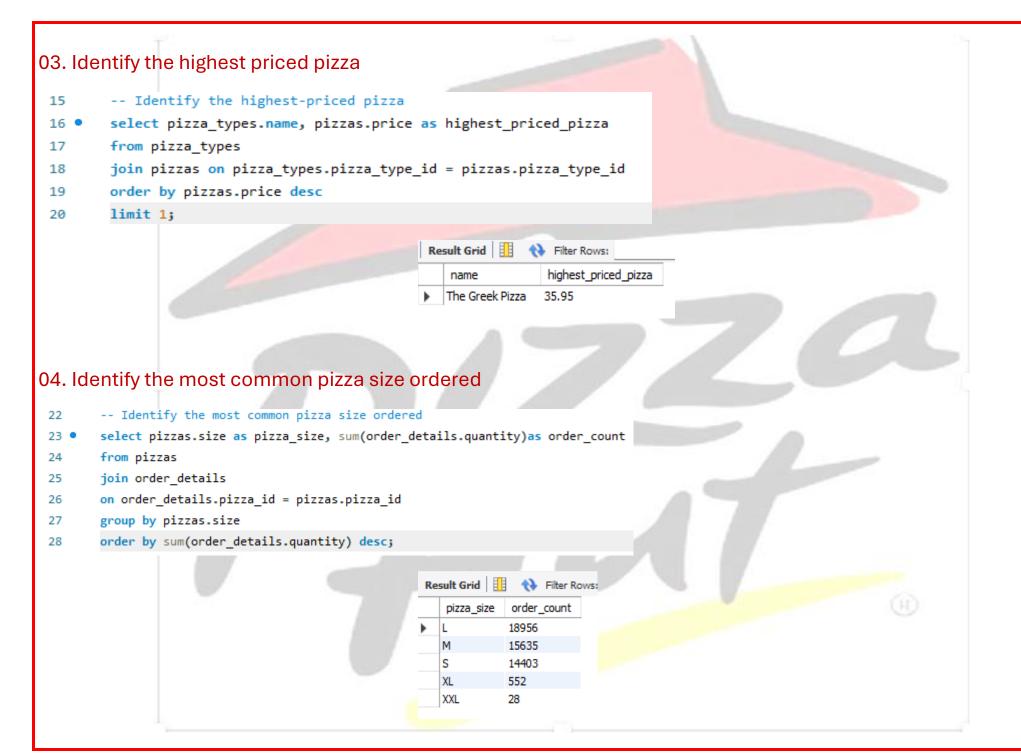
Best-selling pizzas

Revenue patterns

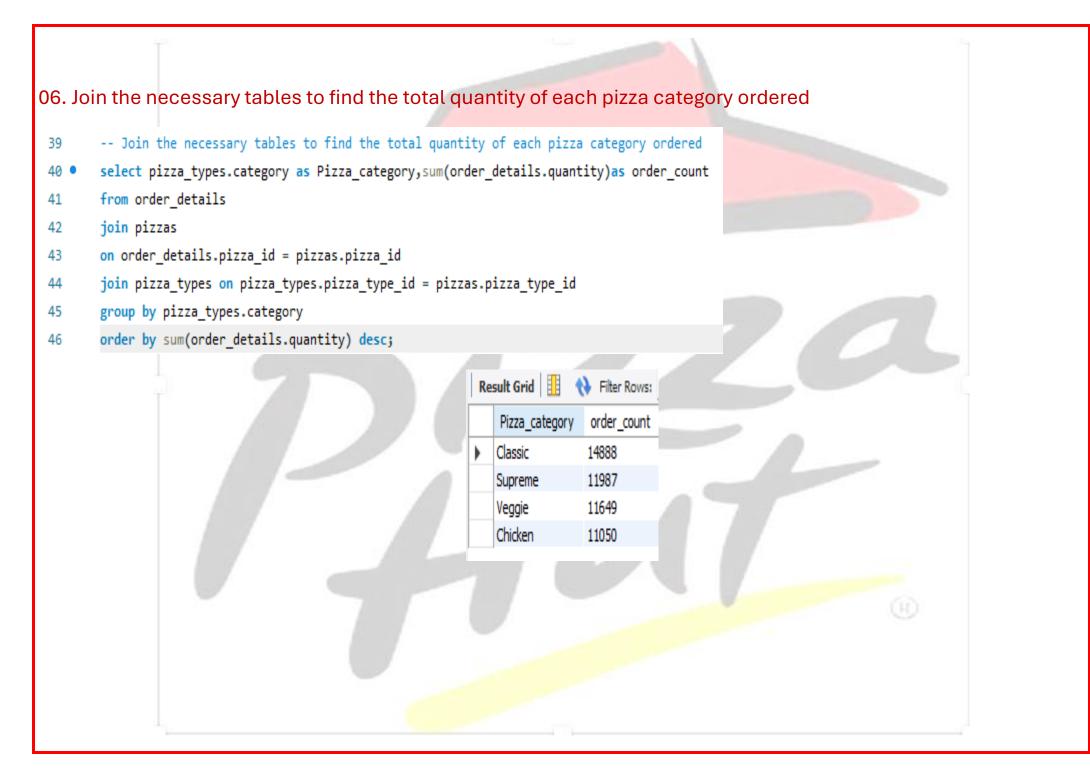
Pizza category breakdowns

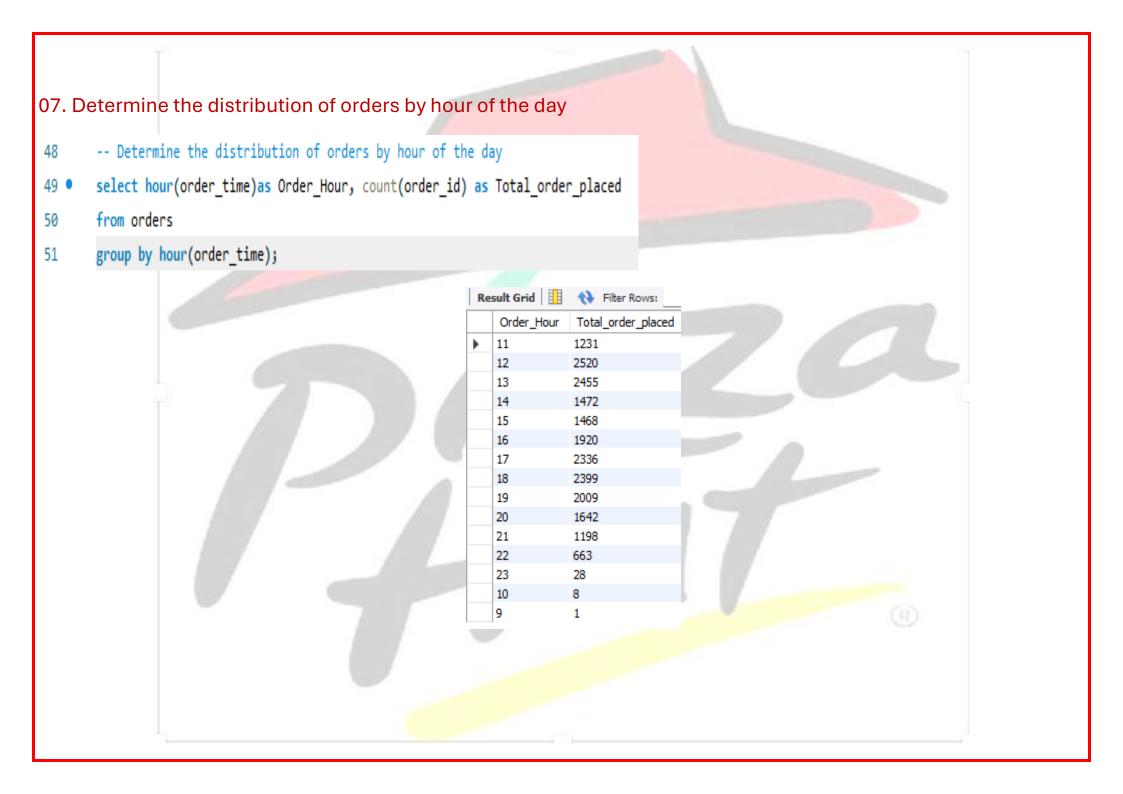












08. Join relevant tables to find the category wise distribution of pizzas

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

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

with cte as(
select orders.order_date as 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)

select round(avg(quantity),0) as avg_pizza_ordered_per_day

from cte;

avg_pizza_ordered_per_day

| 138
```



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

```
-- Calculate the percentage contribution of each pizza type to total revenue
79 • ⊖ with cte as(
       select pizza_types.category as pizza_category, sum(order_details.quantity * pizzas.price) as revenue
80
       from pizza_types
81
       join pizzas
       on pizza_types.pizza_type_id = pizzas.pizza_type_id
83
       join order_details on order_details.pizza_id = pizzas.pizza_id
85
       group by pizza_types.category)
86
       select pizza_category,round(revenue/sum(revenue)over()*100,2) as percent_revenue
87
       from cte
88
       group by pizza_category;
```

R	esult Grid 📙 🛭 Fi	ter Rows:	
	pizza_category	percent_revenue	
١	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

12. Analyze the cumulative revenue generated over time

```
-- Analyze the cumulative revenue generated over time
93 • ⊖ with cte as(
        select orders.order_date as order_date, sum(order_details.quantity * pizzas.price) as revenue
 94
        from orders
        join order_details
 96
        on orders.order_id = order_details.order_id
 97
        join pizzas on pizzas.pizza_id = order_details.pizza_id
 98
        group by orders.order_date)
99
100
        select order_date, round(sum(revenue) over(order by order_date),2)as cumm_revenue
101
        from cte;
102
```

Re	sult Grid	Filter Rows:
	order_date	cumm_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 36937.65
	2015-01-16	39001.75
	2015-01-17	40978.6
	. 2010 01 10	.5570.0

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

```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category
104
105 • ⊖ with cte as(
        select pizza_types.name, sum(order_details.quantity * pizzas.price) as revenue
106
        from pizza types
107
        join pizzas
108
        on pizza_types.pizza_type_id = pizzas.pizza_type_id
109
        join order_details on order_details.pizza_id = pizzas.pizza_id
110
111
        group by pizza_types.name)
112
        select name, revenue,dense_rank() over(order by revenue desc) as rnk
113
        from cte;
```

name	revenue	rnk
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	4
The Spicy Italian Pizza	34831.25	5
The Southwest Chicken Pizza	34705.75	6
The Italian Supreme Pizza	33476.75	7
The Hawaiian Pizza	32273.25	8
The Four Cheese Pizza	32265.70000000065	9
The Sicilian Pizza	30940.5	10
The Pepperoni Pizza	30161.75	11
The Greek Pizza	28454.100000000013	12
The Mexicana Pizza	26780.75	13
The Five Cheese Pizza	26066.5	14
The Pepper Salami Pizza	25529	15
The Italian Capocollo Pizza	25094	16
The Vegetables + Vegetable	24374.75	17
The Prosciutto and Arugula	24193.25	18