PIZZA SALES

SQL PROJECT ON PIZZA SALES

Hello!

My name is Kiran Kumari Meena,
in this project I have utilise SQL query to solve the questions related to pizza sales.

Here is the tables that used in pizza sales

Retrieve the total number of orders placed.

```
select
count(order_id) as total_orders
from orders;
```

Result Grid

total_orders
1845

Calculate the total revenue generated from pizza sales.

```
select
round(sum(p.price*o.quantity),2)
as total_revenue
from pizzas as p
join order_details as o
on p.pizza_id=o.pizza_id;
```

Result Grid

total_revenue
69793.3

Identify the highest-priced pizza.

```
select pizza_types.name ,pizzas.price
from pizzas join pizza_types
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

Identify the most common pizza size ordered.

```
select pizzas.size,count(order_details.order_details_id)
as order_count
from pizzas
join order_details
on pizzas.pizza_id=order_details.pizza_id
group by  pizzas.size order by order_count  desc limit 1;
```

Result Grid

size	order_count
L	1599

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

```
select pt.name ,sum(o.quantity)
from pizza_types as pt
join pizzas as p
on p.pizza_type_id=pt.pizza_type_id
join order_details as o
on p.pizza_id=o.pizza_id
group by pt.name
order by count(o.quantity) desc limit 5 ;
```

Result Grid | Filter Rows:

name	sum(o.quantity)
The Pepperoni Pizza	239
The Barbecue Chicken Pizza	211
The Thai Chicken Pizza	199
The California Chicken Pizza	202
The Classic Deluxe Pizza	190

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

```
select pt.category,sum(o.quantity) as total_quantity
from pizza_types as pt
join pizzas as p
on p.pizza_type_id=pt.pizza_type_id
join order_details as o
on p.pizza_id=o.pizza_id
group by pt.category;
```

Result Grid | Filter Rows:

category	total_quantity
Classic	1257
Veggie	1018
Supreme	1044
Chicken	913

Determine the distribution of orders by hour of the day.

```
select hour(order_time) as hours,
count(order_id) as order_count
from orders
group by hours;
```

Result Grid

hours	order_count
11	100
12	206
13	201
14	166
15	132
16	165
17	193
18	208
19	177
20	148
21	96
22	52
23	1

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

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

Result Grid | Filter Rows:

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

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

```
select round(avg(quantity),0) as avg_pizza_order
from (select o1.order_date as days, sum(o2.quantity) as quantity
from orders as o1
join order_details as o2
on o1.order_id=o2.order_id
group by o1.order_date) as order_quantity ;
```

Result Grid

	avg_pizza_order
▶	137

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

```sql
select pt.name, sum(p.price* o.quantity) as total_revenue
from pizzas as p
join pizza_types as pt
on p.pizza_type_id=pt.pizza_type_id
join order_details as o
on o.pizza_id=p.pizza_id
group by pt.name  order by total_revenue desc limit 3 ;
```

Result Grid Filter Rows:

name	total_revenue
The Barbecue Chicken Pizza	3770.25
The Thai Chicken Pizza	3657.25
The California Chicken Pizza	3471.5

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

```
select pt.category ,round((sum(p.price* o.quantity)/(select round(sum(p.price* o.quantity),2)
as total_revenue
from pizzas as p
join order_details as o
on o.pizza_id=p.pizza_id))*100,2) as revenue
from pizzas as p
join pizza_types as pt
on p.pizza_type_id=pt.pizza_type_id
join order_details as o
on o.pizza_id=p.pizza_id
group by pt.category ;
```

Result Grid | Filter

category	revenue
Classic	26.68
Veggie	24.44
Supreme	25.69
Chicken	23.2

Analyze the cumulative revenue generated over time.

```sql
select order_date,sum(revenue) over(order by order_date) as cum_revenue
from (select ORDERS.ORDER_DATE, round(sum(p.price* o.quantity),2) as revenue
from pizzas as p
join order_details as o
on o.pizza_id=p.pizza_id
JOIN ORDERS
ON  ORDERS.ORDER_ID=O.ORDER_ID
GROUP BY ORDERS.ORDER_DATE) AS sales;
```

Result Grid | Filter Rows:

order_date	cum_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	21526.399999999998
2015-01-10	23990.35
2015-01-11	25862.649999999998
2015-01-12	27781.699999999997
2015-01-13	29831.299999999996
2015-01-14	32358.699999999997
2015-01-15	34343.5
2015-01-16	36937.65
2015-01-17	39001.75
2015-01-18	40978.6
2015-01-19	43365.75
2015-01-20	45763.65
2015-01-21	47804.200000000004
2015-01-22	50300.9
2015-01-23	52724.6
2015-01-24	55013.85
2015-01-25	56631.4
2015-01-26	58515.8
2015-01-27	61043.850000000006
2015-01-28	63059.850000000006
2015-01-29	65105.15000000001
2015-01-30	67375.45000000001
2015-01-31	69793.30000000002

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 pt.category,pt.name, round(sum(p.price* o.quantity),2) as revenue
from pizza_types as pt
join pizzas as p
on p.pizza_type_id=pt.pizza_type_id
join order_details as o
on p.pizza_id =o.pizza_id
group by pt.category,pt.name)
as a) as b
where rn<=3;
```

Result Grid | Filter Rows:

category	name	revenue
Chicken	The Barbecue Chicken Pizza	3770.25
Chicken	The Thai Chicken Pizza	3657.25
Chicken	The California Chicken Pizza	3471.5
Classic	The Pepperoni Pizza	3001.25
Classic	The Classic Deluxe Pizza	2941.5
Classic	The Greek Pizza	2539.4
Supreme	The Sicilian Pizza	2987.75
Supreme	The Italian Supreme Pizza	2954.75
Supreme	The Spicy Italian Pizza	2762
Veggie	The Four Cheese Pizza	2671.65
Veggie	The Five Cheese Pizza	2553
Veggie	The Mexicana Pizza	2358.25

Thank YOU