SQL PROJECT ON PIZZA SALES

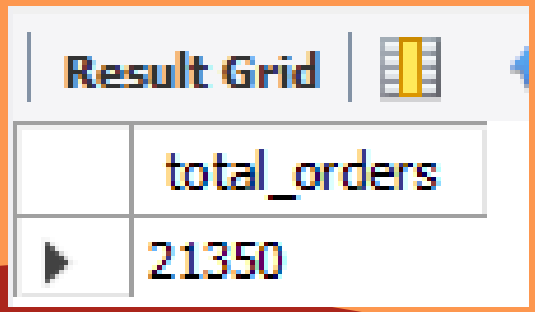
Devyani Rasekar

HELLO!

I'm Devyani Rasekar. Additionally I used a SQL query in this project to answer questions about pizza sales.

Retrieve the total number of orders placed.

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



Result Grid

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales.

```sql
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;
```

Result Grid

total_revenue
817860.05

Identify the highest-priced pizza.

```sql
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 R

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

```sql
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;
```

Result Grid

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

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

```sql
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 5;
```

Result Grid | Filter Rows:

name	quantity
The Thai Chicken Pizza	2371
The Pepperoni Pizza	2418
The Hawaiian Pizza	2422
The Barbecue Chicken Pizza	2432
The Classic Deluxe Pizza	2453

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

```sql
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 | Filte

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

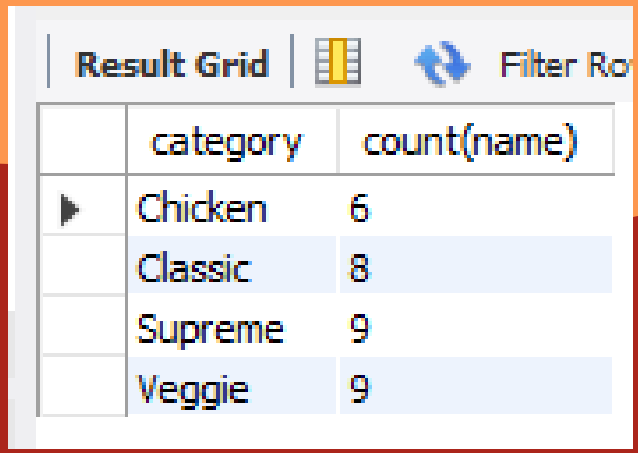
```sql
select hour(order_time) as hour,
count(order_id) as order_count
from orders
group by hour(order_time);
```

Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	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 Ro

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.

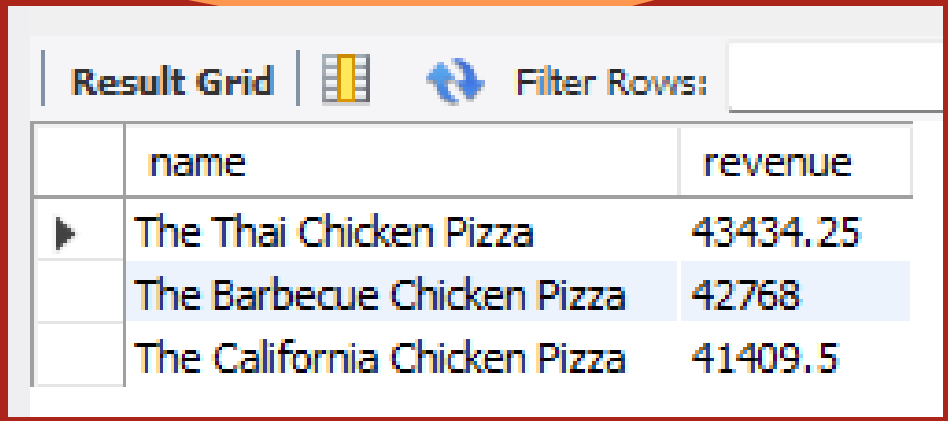
```sql
select round(avg(quantity),0) from
(select orders.order_date ,
sum(order_details.quantity) as quantity
from orders
join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```

Result Grid | Filter R

	round(avg(quantity),0)
▶	138

Determine the top3 most ordered pizza types based on revenue.

```sql
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;
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

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

```sql
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 pizzas.pizza_id = order_details.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

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 , 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;
```

Result Grid | Filter Rows:

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
2015-01-14	32358.700000000004

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

```sql
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

Thank you!