
Pizza Sales Analysis:
Uncovering Trends

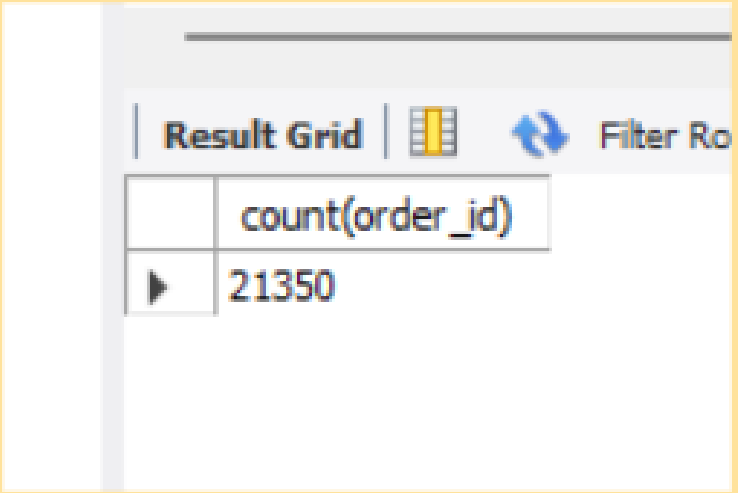

Welcome to my pizza sales analysis project. Through SQL queries, joins, ranks and sub queries, I've explored pizza sales data to uncover insights into consumption patterns, most common pizza types, peak sales time, revenue details and much more.

Tables to be served:

1. Orders
2. Pizzas
3. Order_details
4. Pizza_types

Retrieve the total number of orders placed

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



Calculate the total revenue generated from pizza sales.

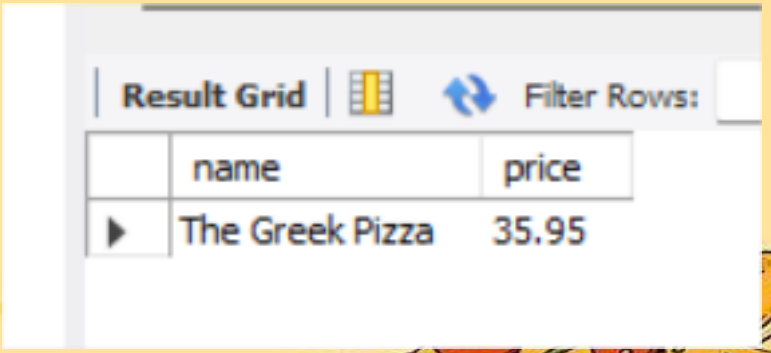
```sql
SELECT
    SUM(order_details.quantity * pizzas.price) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid | Filter R

total_sales
817860.049999993

Identify the highest priced pizza

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    order by pizzas.price desc limit 1;
```



Identify the most common pizza type 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 | Filter Row

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
    pt.name, SUM(od.quantity) AS total_quantity
FROM
    pizza_types AS pt
        JOIN
    pizzas ON pt.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details AS od ON od.pizza_id = pizzas.pizza_id
GROUP BY pt.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

Determine the distribution of orders by hour of the day.

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

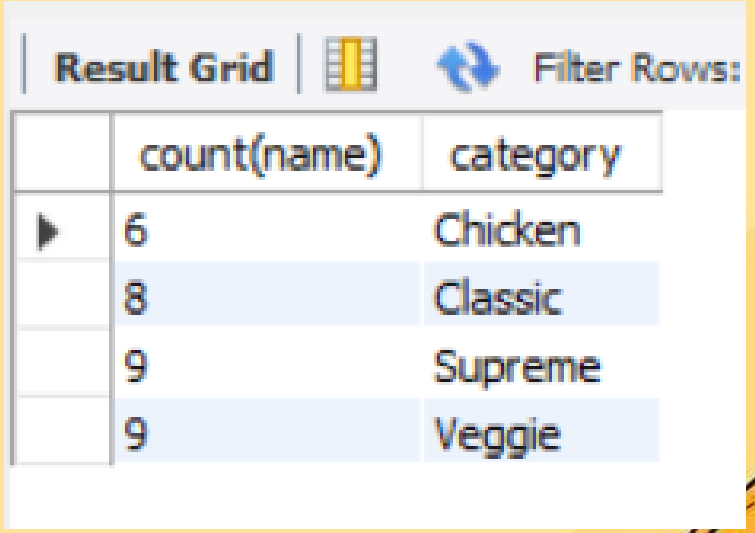
Result Grid | Filter Rov

ho	count(order_id)
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

Result 18

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

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



Result Grid | Filter Rows:

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

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

```sql
select avg(quantity) 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 Rows:

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

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

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

Result Grid | Filter Rows:

name	TR
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 pt.category , sum(od.quantity * pizzas.price)/(select
sum(order_details.quantity*pizzas.price) as TS
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id) * 100 as TR
from pizzas
join pizza_types as pt
on pizzas.pizza_type_id = pt.pizza_type_id
join order_details as od
on pizzas.pizza_id = od.pizza_id
group by pt.category
order by TR desc;
```

Result Grid | Filter Rows:

category	TR
Classic	26.905960255669903
Supreme	25.45631126009884
Chicken	23.955137556847493
Veggie	23.682590927384783

Analyse the cumulative revenue generated over time.

```sql
select order_date, sum(sales) over(order by order_date) as cum_rev
from
(select orders.order_date , sum(od.quantity * pizzas.price) as sales
from order_details as od
join orders
on orders.order_id = od.order_id
join pizzas
on od.pizza_id = pizzas.pizza_id
group by orders.order_date) as revenue;
```

Result Grid | Filter Rows:

category	TR
Classic	26.905960255669903
Supreme	25.4563112600988
Chicken	23.955137556847493
Veggie	23.682590927384783

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

```sql
select name, revenue, rn
from
(
select category, name , revenue,
rank() over(partition by category order by revenue desc)as rn
from
(
select pt.category, pt.name ,sum(od.quantity * pizzas.price) as revenue
from pizzas
join pizza_types as pt
on pizzas.pizza_type_id = pt.pizza_type_id
join order_details as od
on pizzas.pizza_id = od.pizza_id
group by pt.category, pt.name) as a) as b
where rn<=3;
```

Result Grid | Filter Rows: | Export:

name	revenue	rn
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	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3
The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3


Thank you everyone
for joining our little
pizza party!!