Pizza Hut SQL Data Analysis

An in-depth analysis of Pizza Hut sales data using SQL queries to uncover valuable business insights.

By - Abhishek shukla

LinkedIn - www.linkedin.com/in/avi-shukla-da

GitHub - https://github.com/abhishukkla

Made with GAMMA

Database Schema & Structure

Key Tables

- Orders
- Order_details
- Pizzas
- Pizza_types

Data Relationships

- Orders → Order_details → Pizzas
- 40,000+ transactions
- 1 year of sales data

Made with GAMMA

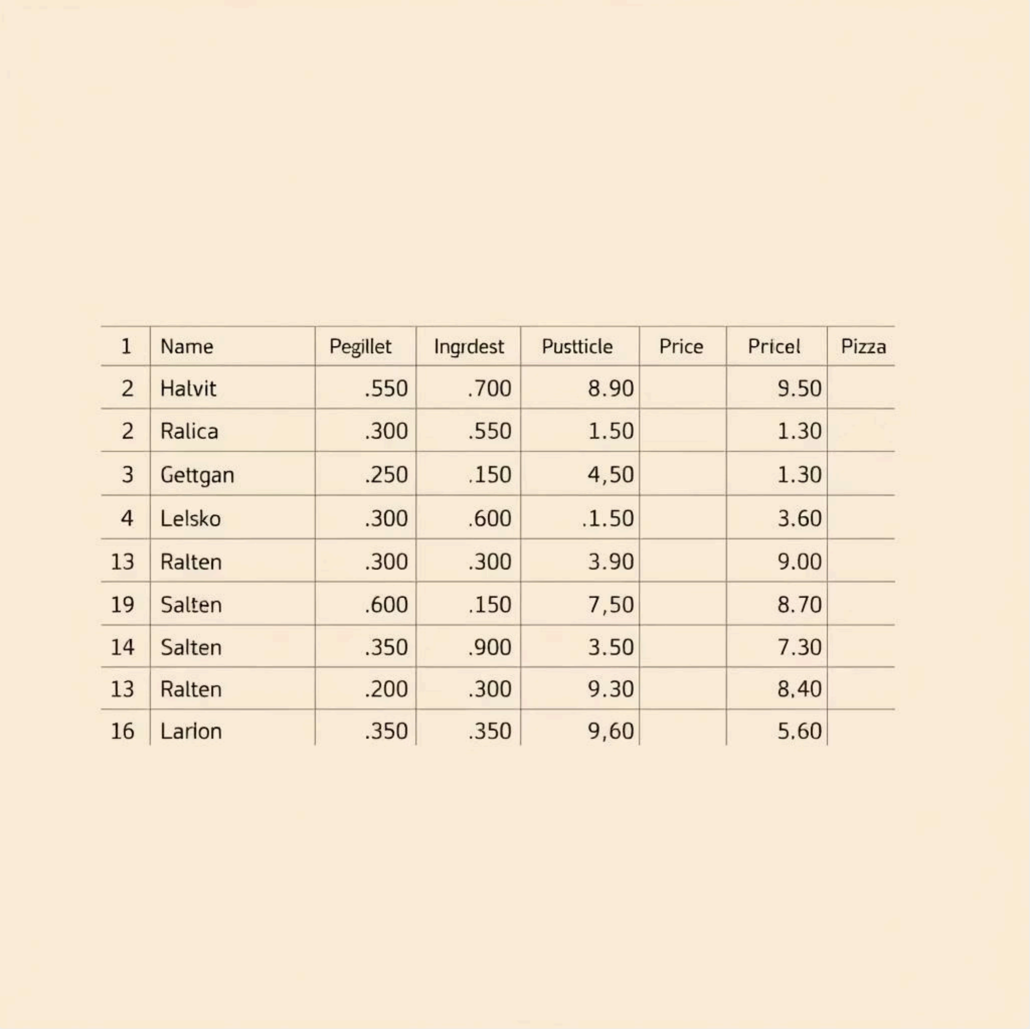
Core Performance Metrics

Total Orders

```
SELECT COUNT(order_id) FROM orders;
```

Total Revenue

```
SELECT ROUND(SUM(quantity * price),2)
FROM order_details JOIN pizzas
```



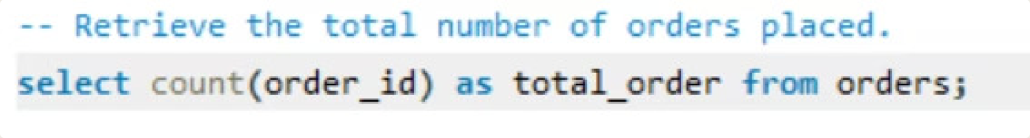
1	Name	Pegillet	Ingrdest	Pustticle	Price	Pricel	Pizza
2	Halvit	.550	.700	8.90		9.50	
2	Ralica	.300	.550	1.50		1.30	
3	Gettgan	.250	.150	4,50		1.30	
4	Lelsko	.300	.600	.1.50		3.60	
13	Ralten	.300	.300	3.90		9.00	
19	Salten	.600	.150	7,50		8.70	
14	Salten	.350	.900	3.50		7.30	
13	Ralten	.200	.300	9.30		8,40	
16	Larion	.350	.350	9,60		5.60	

Popular Pizza Analysis

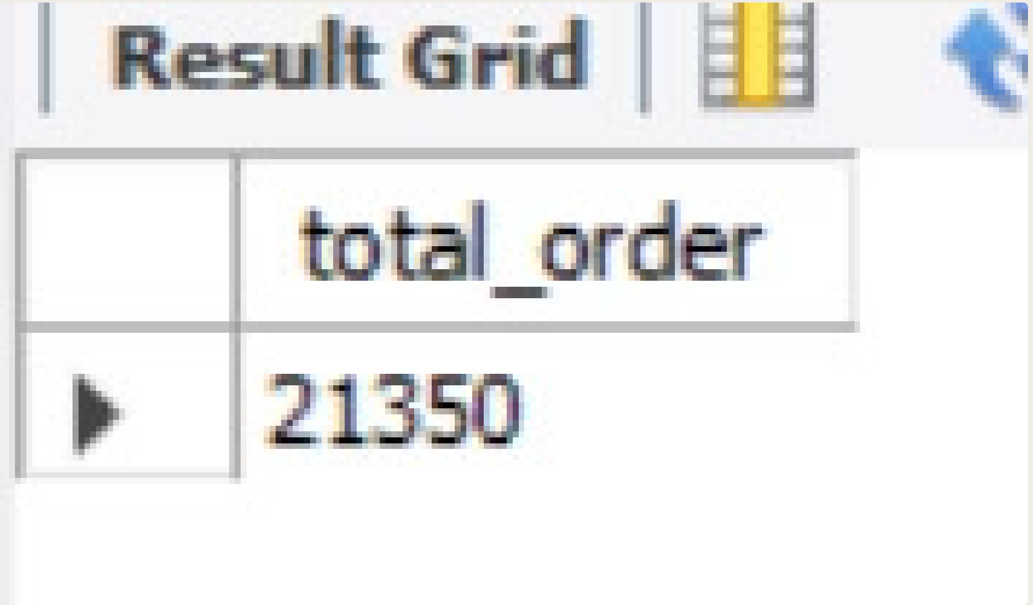
12	5	$24.99
Most ordered pizza size	Top selling pizza types	Highest priced pizza

Made with GAMMA

total number of orders placed



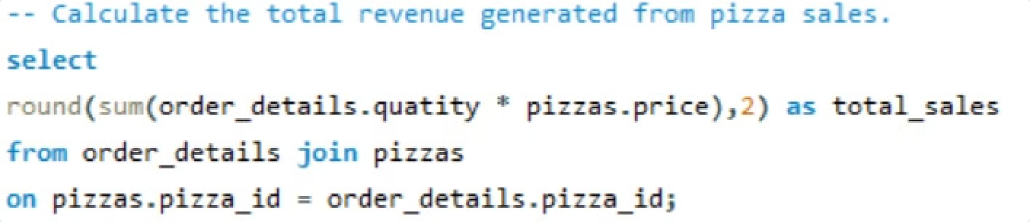
```
-- Retrieve the total number of orders placed.
select count(order_id) as total_order from orders;
```



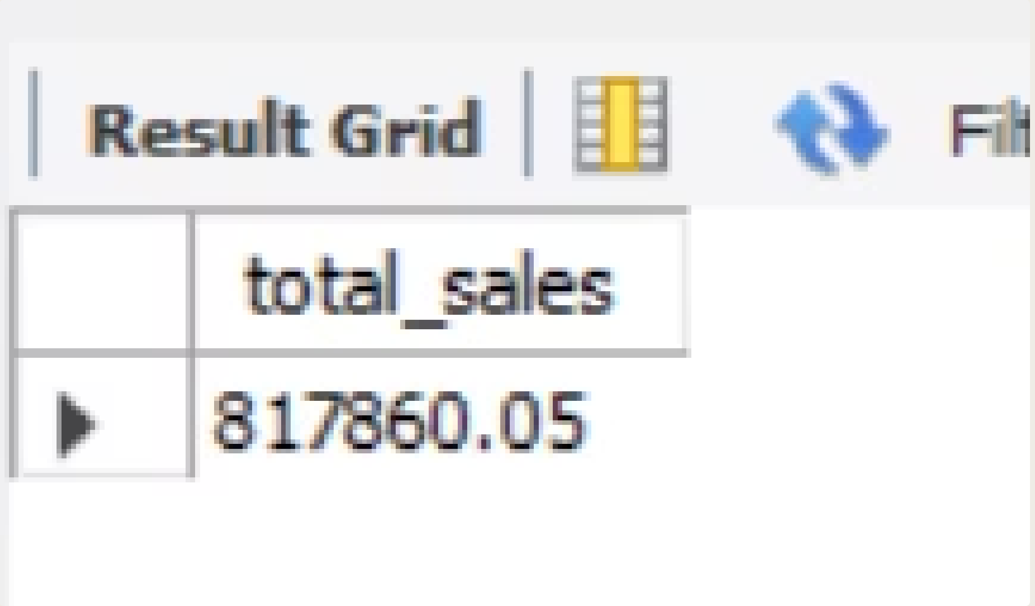
	total_order
▶	21350

Made with GAMMA

total revenue generated



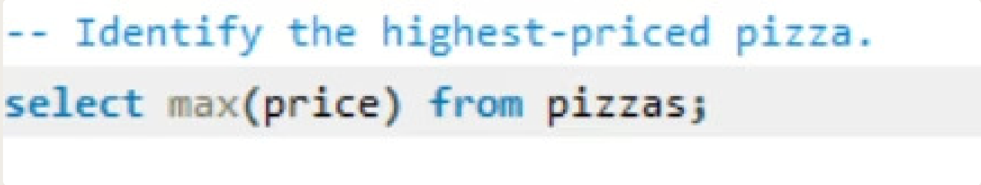
```
-- Calculate the total revenue generated from pizza sales.
select
round(sum(order_details.quatity * pizzas.price),2) as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```



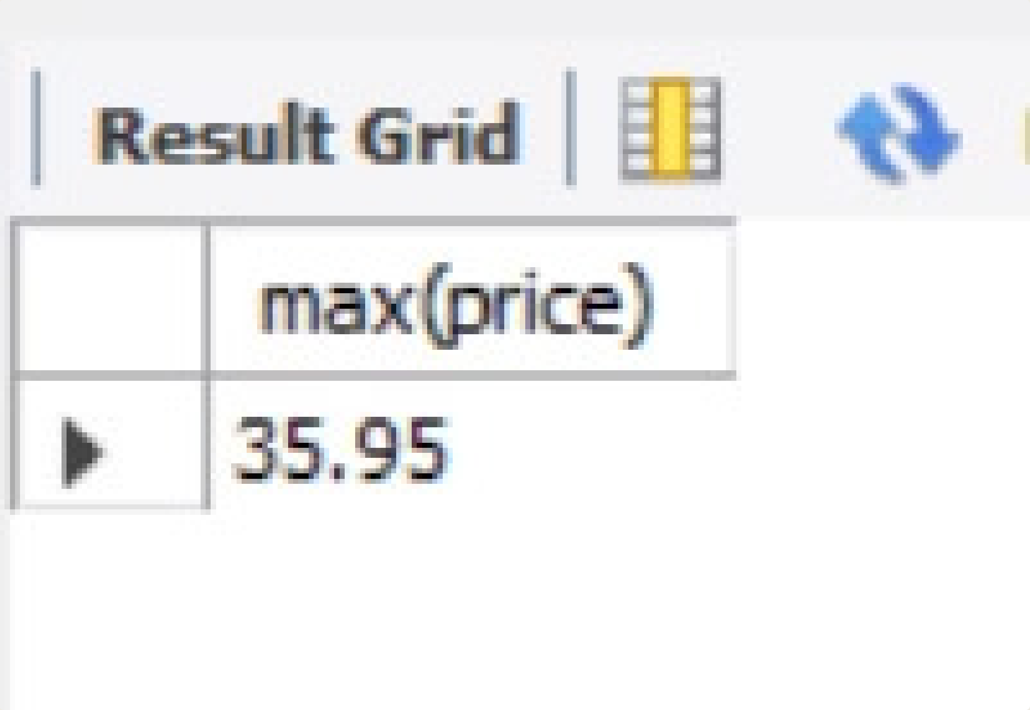
```
total_sales
817860.05
```

Made with GAMMA

highest-priced pizza



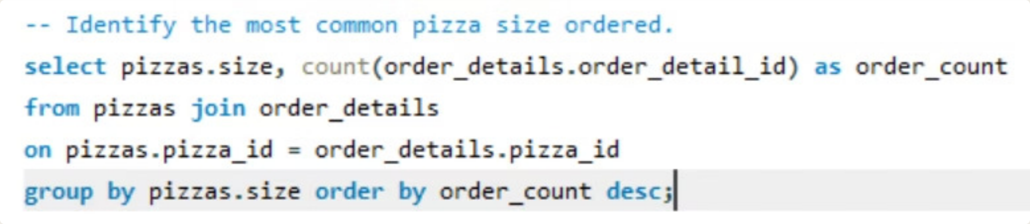
```sql
-- Identify the highest-priced pizza.
select max(price) from pizzas;
```



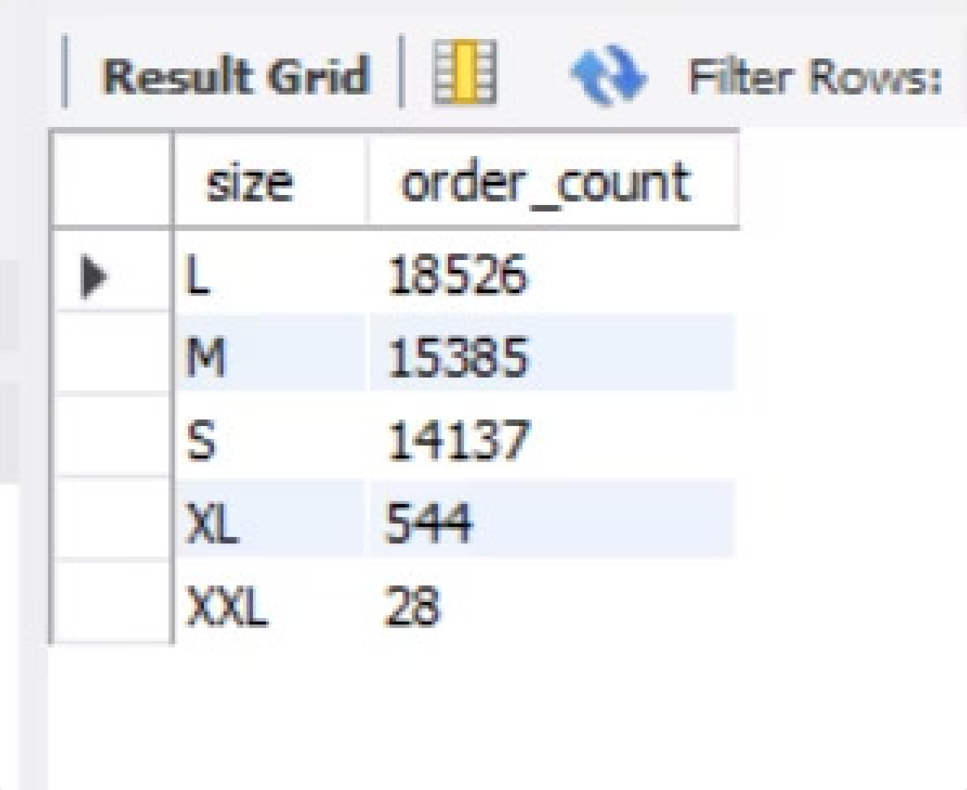
```
max(price)
35.95
```

Made with GAMMA

most common pizza



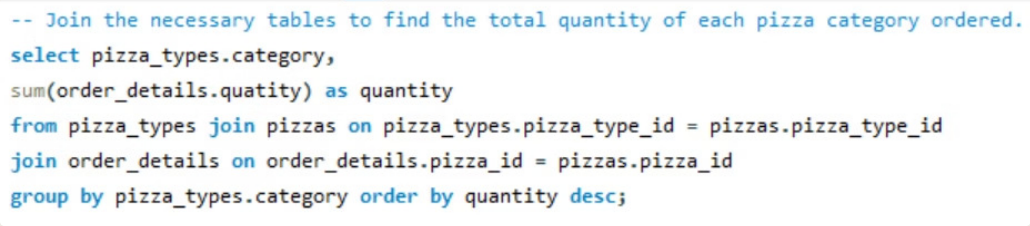
```sql
-- Identify the most common pizza size ordered.
select pizzas.size, count(order_details.order_detail_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```



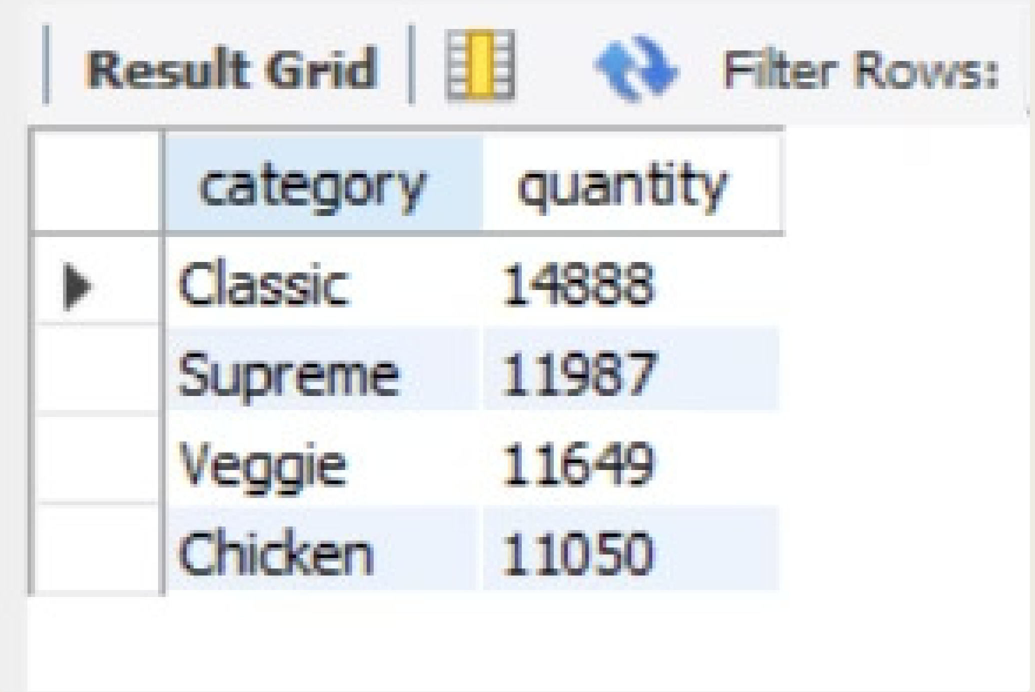
size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28

Made with GAMMA

total quantity of each pizza category



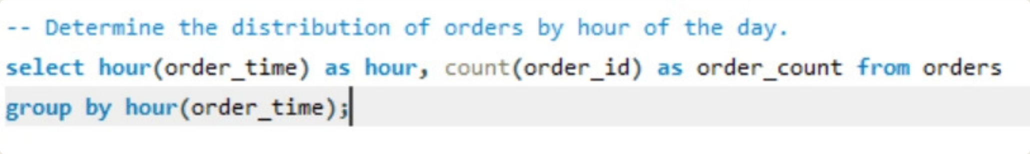
```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.
select pizza_types.category,
sum(order_details.quatity) as quantity
from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by quantity desc;
```



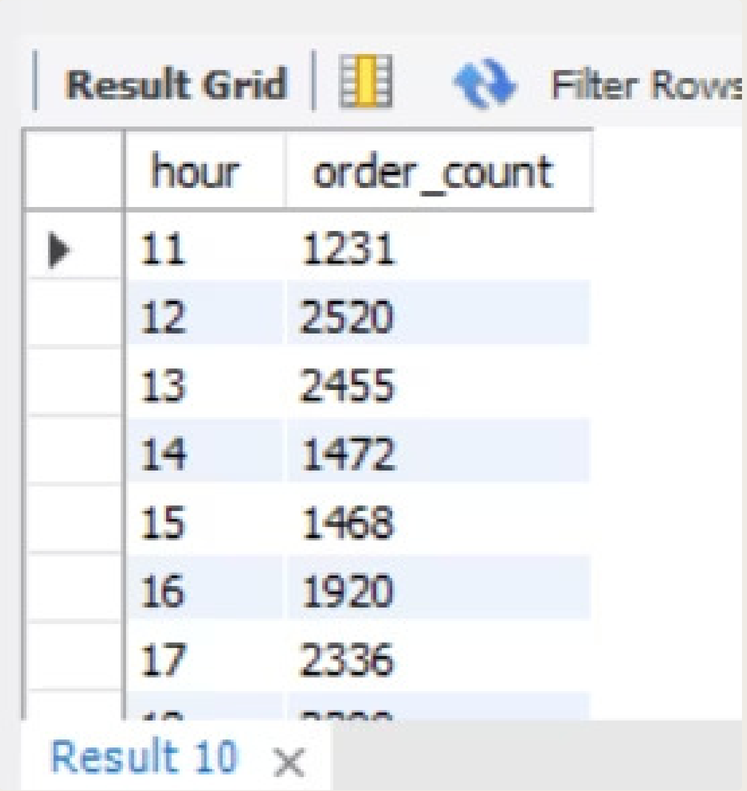
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Made with GAMMA

orders by hour of the day



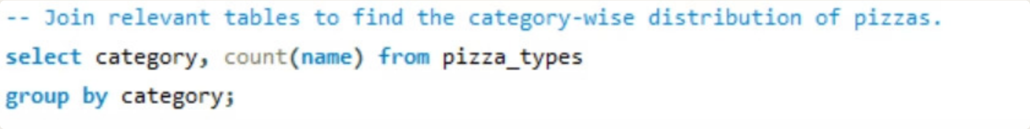
```
-- Determine the distribution of orders by hour of the day.
select hour(order_time) as hour, count(order_id) as order_count from orders
group by hour(order_time);
```



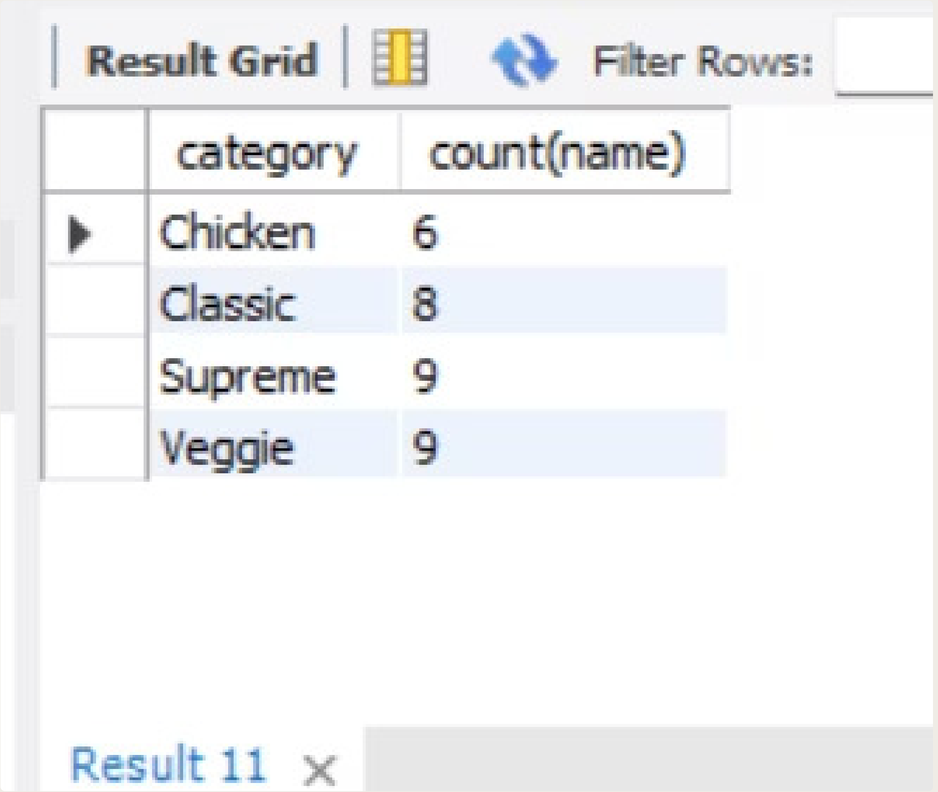
hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

Made with GAMMA

category-wise distribution of pizzas.



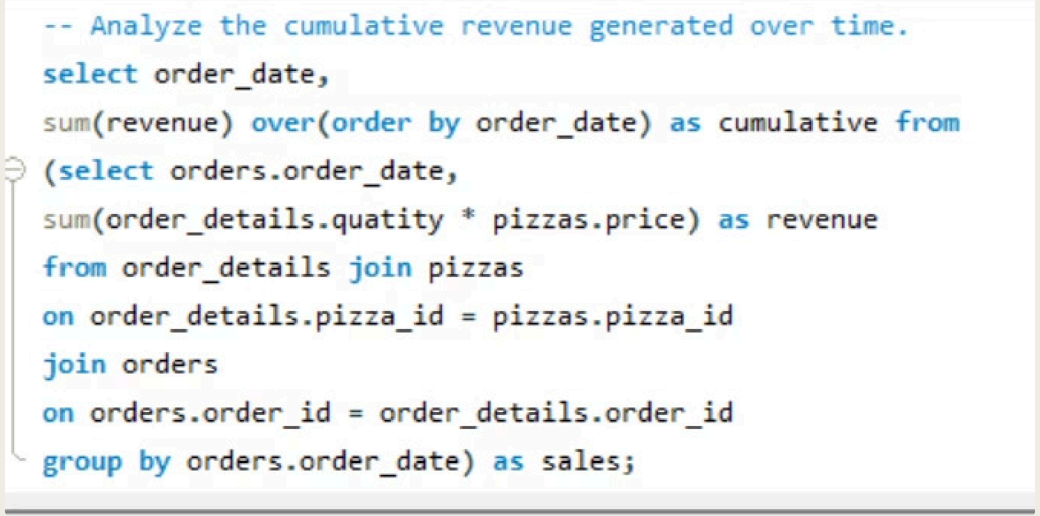
```
-- Join relevant tables to find the category-wise distribution of pizzas.
select category, count(name) from pizza_types
group by category;
```



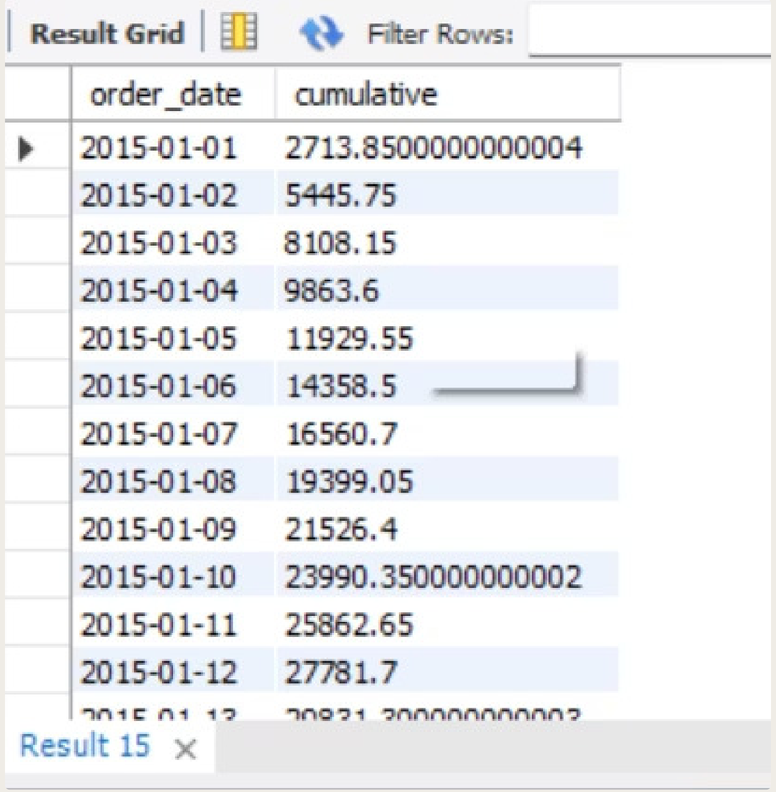
category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Made with GAMMA

Analyze the cumulative revenue generated over time



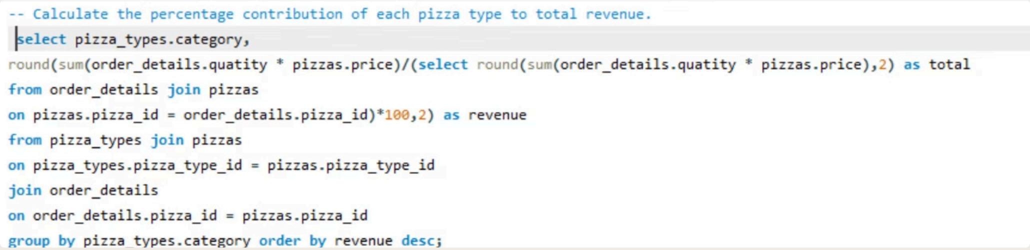
```
-- Analyze the cumulative revenue generated over time.
select order_date,
sum(revenue) over(order by order_date) as cumulative from
(select orders.order_date,
sum(order_details.quatity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```



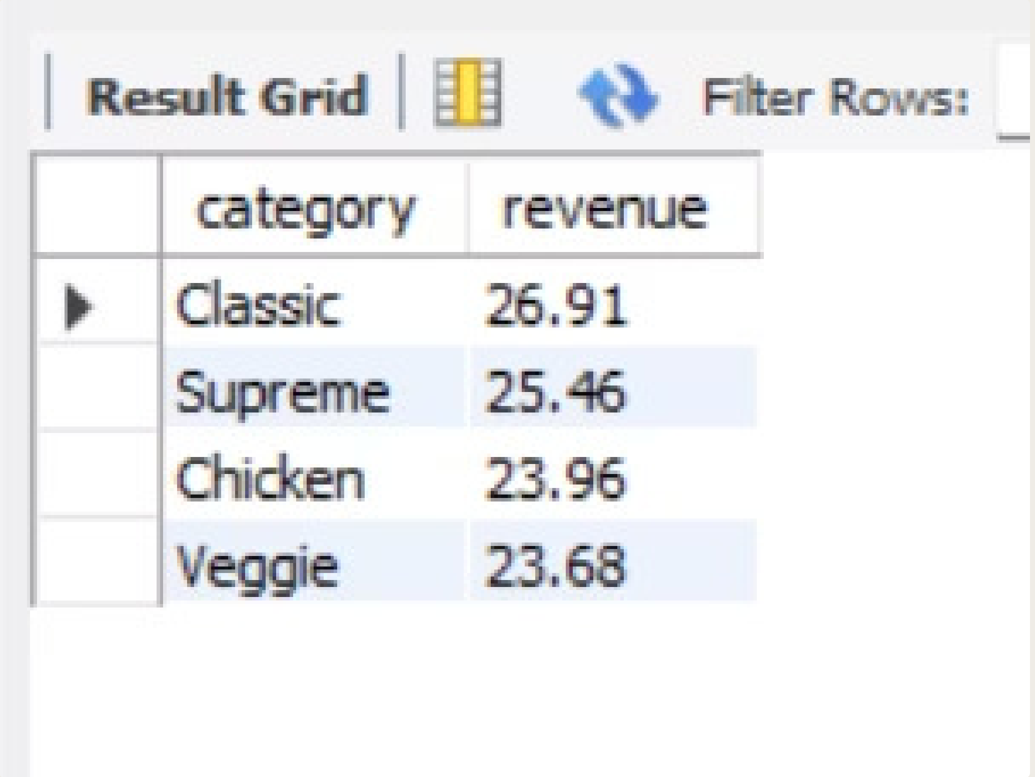
order_date	cumulative
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

Made with GAMMA

percentage contribution of each pizza type to total revenue



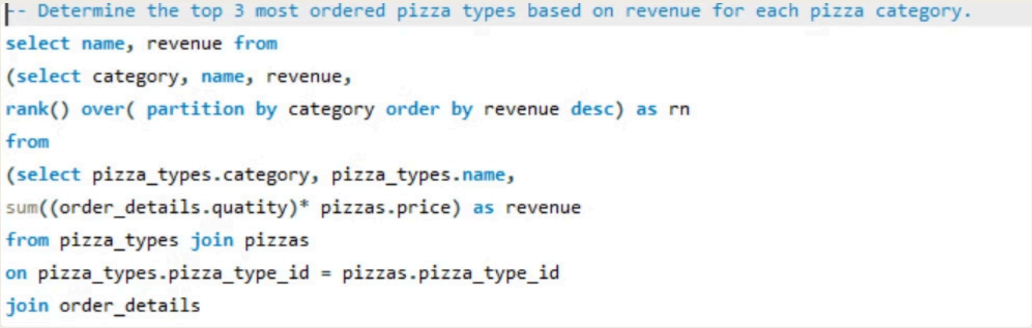
```
-- Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category,
round(sum(order_details.quatity * pizzas.price)/(select round(sum(order_details.quatity * pizzas.price),2) as total
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id)*100,2) 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;
```



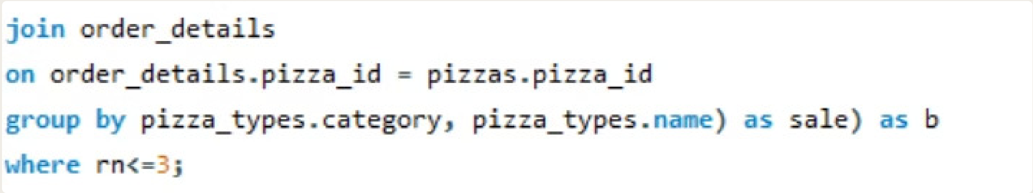
category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Made with GAMMA

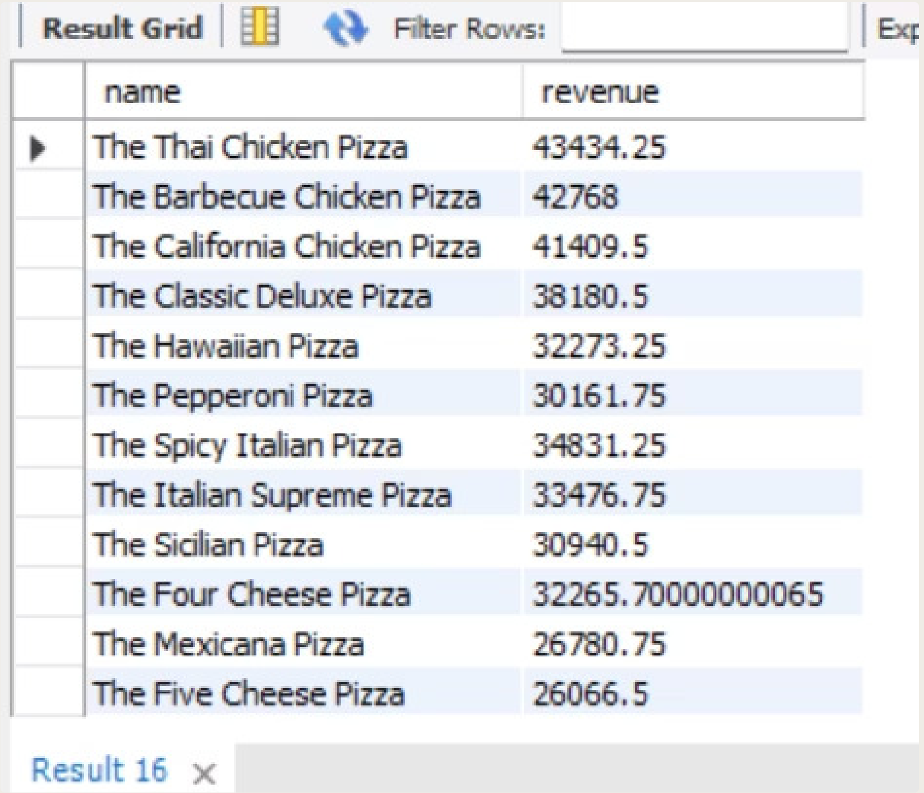
top 3 most ordered pizza types based on revenue



```sql
-- Determine the top 3 most ordered pizza types 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.quatity)* pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
```



```sql
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as sale) as b
where rn<=3;
```



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

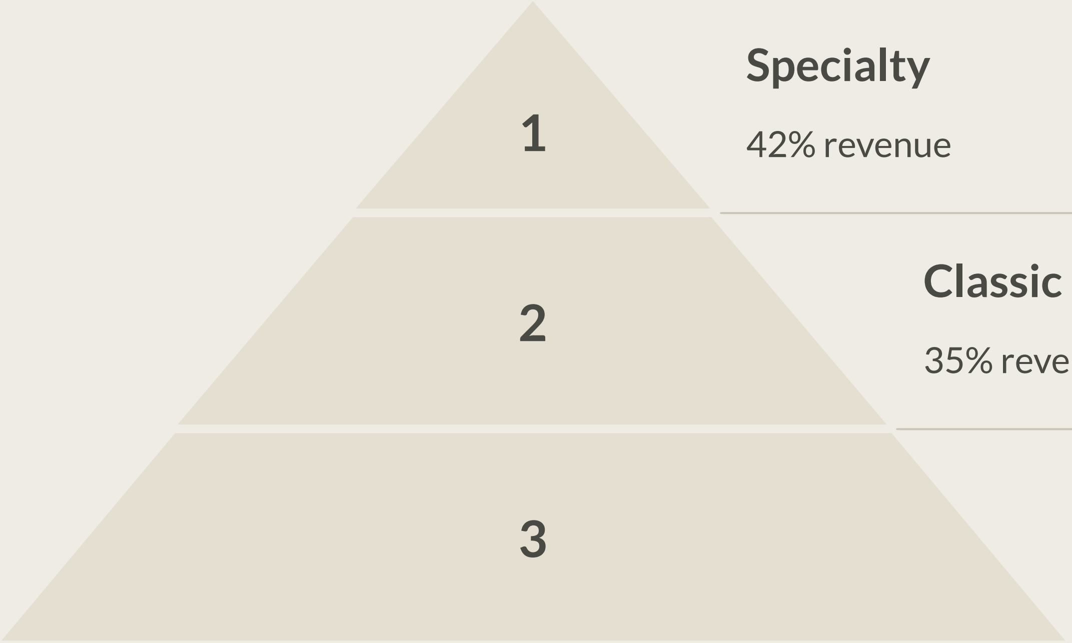
Made with GAMMA

🔍 Key Insights

1. **Large-size pizzas** are the most frequently ordered, indicating customer preference for bigger portions.
2. **The Thai Chicken Pizza** generated the **highest revenue**, closely followed by the Barbecue and California Chicken variants.
3. **Classic category pizzas** contributed the highest share of total revenue (26.91%).
4. The **peak order time** falls between **12 PM to 2 PM**, highlighting lunch as a high-sales window.
5. Daily pizza demand remains consistent with an **average of 138 pizzas sold per day**.
6. The **top 5 pizzas** contributed significantly to the overall sales volume, indicating a narrow product focus.

Made with GAMMA

Revenue Insights



Specialty

42% revenue

Classic

35% revenue

Veggie

23% revenue

Made with GAMMA

💡 Recommendations

1. **Promote bestsellers** like the Thai Chicken and Classic Deluxe pizzas in combo offers and featured deals to maximize revenue.
2. **Bundle large-sized pizzas** with drinks or sides to leverage their popularity and increase average order value.
3. **Optimize kitchen staffing** and delivery resources during the peak lunch hours to ensure fast service.
4. Expand the **Classic and Supreme categories**, as they show strong revenue potential.
5. Create a **loyalty program** focused on frequent buyers of high-revenue pizza types to improve retention.

Made with GAMMA

SQL Techniques Applied

- Multi-table joins for comprehensive analysis
- Window functions for trend calculation
- Aggregate functions for KPIs
- Query optimization techniques

Made with GAMMA

✅ Conclusion

This analysis provided deep insights into customer preferences, revenue drivers, and ordering behavior. By leveraging SQL for data exploration, we successfully identified high-performing products, profitable categories, and optimal business hours. With targeted promotions and operational tweaks based on this data, Pizza Hut can drive both customer satisfaction and revenue growth.

Made with GAMMA