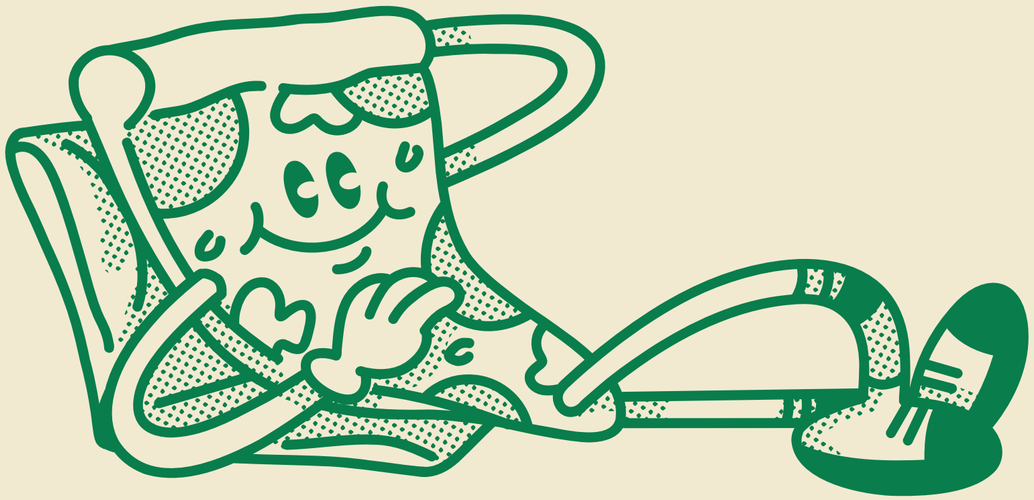
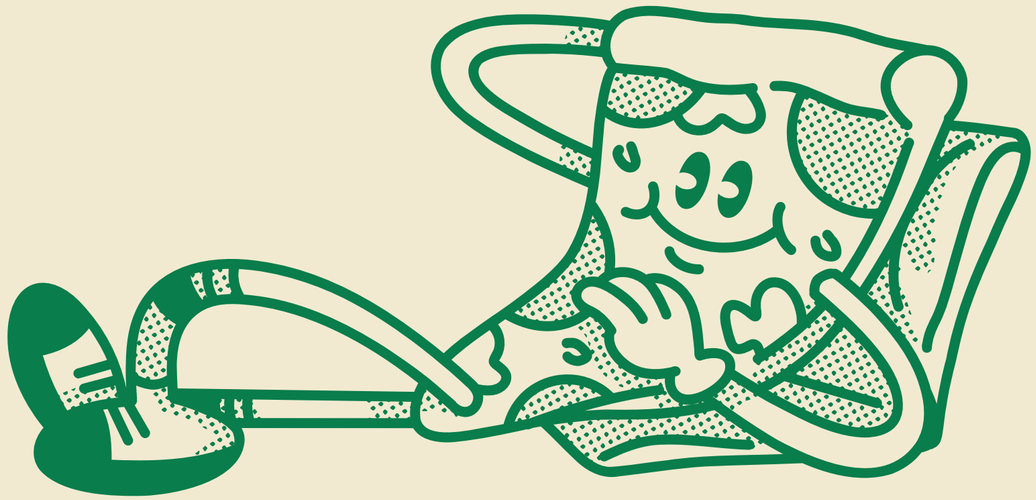
SQL PROJECT ON PIZZA SALES



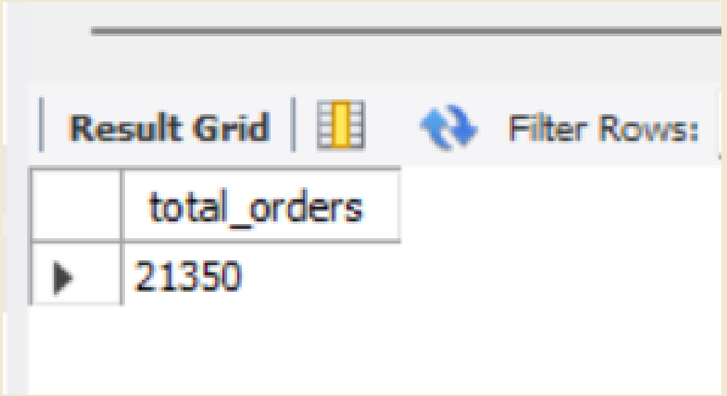


Hello!

My name is Lokeshkumar Madavi.
In this project, I used SQL queries to answer questions related to pizza sales.

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders
```



Calculate the total revenue generated from pizza sales.

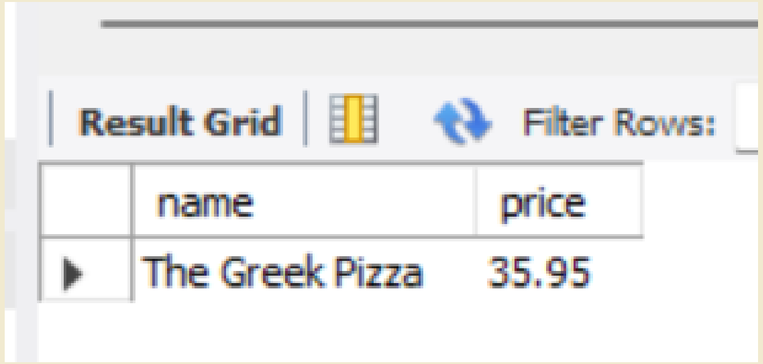
```sql
SELECT
    ROUND(SUM((quantity * price)), 2) AS total_sales
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
```

Result Grid

	total_sales
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    pt.name, p.price
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
ORDER BY p.price DESC
LIMIT 1
```



Identify the most common pizza size ordered.

```sql
SELECT
    p.size, COUNT(od.order_details_id) AS order_count
FROM
    pizzas p
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY p.size
ORDER BY order_count DESC
```

Result Grid | Filter Rows:

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.

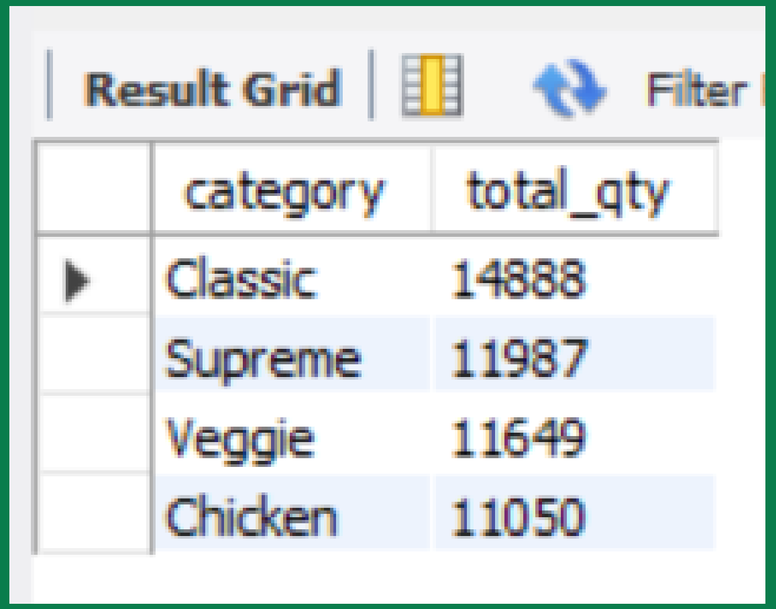
```sql
SELECT
    pt.name, SUM(od.quantity) AS total_qty
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.name
ORDER BY total_qty DESC
LIMIT 5
```

Result Grid | Filter Rows:

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

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

```sql
SELECT
    pt.category, SUM(od.quantity) AS total_qty
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON od.pizza_id = p.pizza_id
GROUP BY pt.category
ORDER BY total_qty DESC
```



category	total_qty
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day.

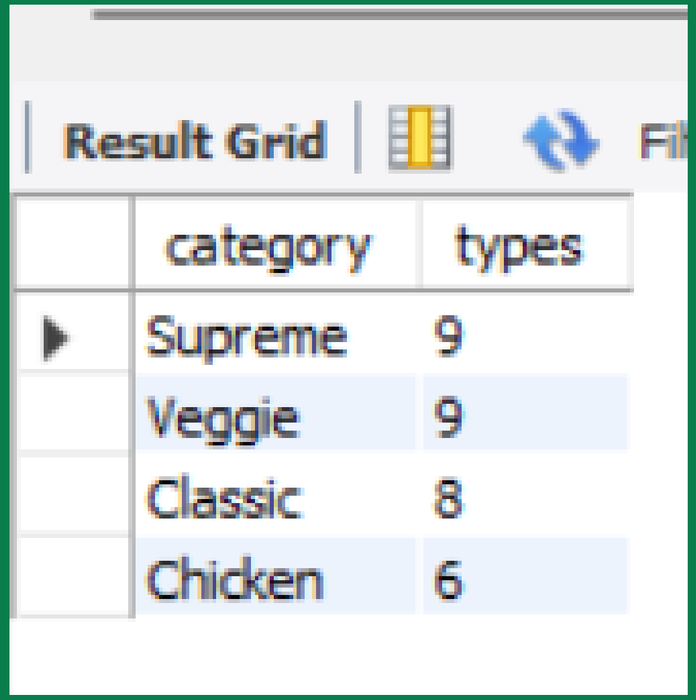
```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY order_count DESC
```

Result Grid | Filter

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

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

```sql
SELECT
    category, COUNT(name) AS types
FROM
    pizza_types
GROUP BY category
ORDER BY COUNT(name) DESC
```

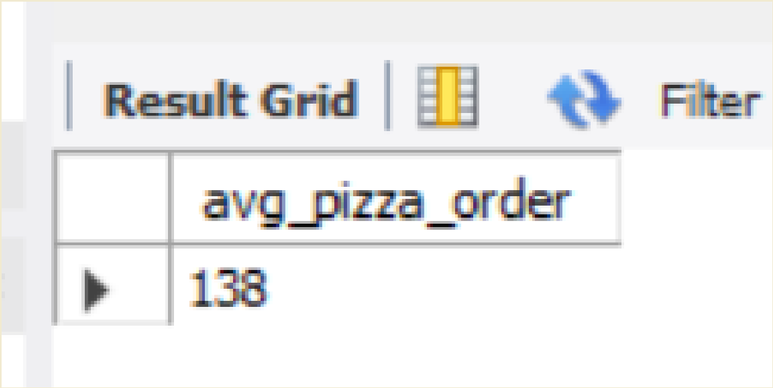


Result Grid

	category	types
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

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

```sql
SELECT
    ROUND(AVG(qty), 0) as avg_pizza_order
FROM
    (SELECT
        o.order_date, SUM(od.quantity) AS qty
    FROM
        orders o
    JOIN order_details od ON o.order_id = od.order_id
    GROUP BY o.order_date) AS order_qty
```



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

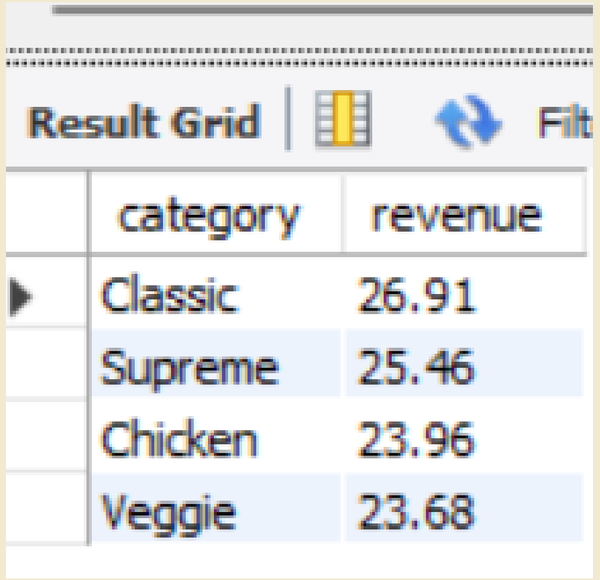
```sql
SELECT
    pt.name, ROUND(SUM(p.price * od.quantity)) AS revenue
FROM
    pizza_types pt
        JOIN
    pizzas p ON pt.pizza_type_id = p.pizza_type_id
        JOIN
    order_details od ON p.pizza_id = od.pizza_id
GROUP BY pt.name
ORDER BY revenue DESC
LIMIT 3
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

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

```sql
SELECT
    pt.category,round((sum(od.quantity*p.price)/(SELECT
    ROUND(SUM((quantity * price)), 2) AS total_sales
FROM order_details o
        JOIN pizzas p ON o.pizza_id = p.pizza_id))*100,2) as revenue
    from pizza_types pt
    join pizzas p on pt.pizza_type_id=p.pizza_type_id
    join order_details od on od.pizza_id=p.pizza_id
    group by pt.category
    order by revenue desc
```



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time

```sql
select
	order_date,round(sum(revenue)
	over(order by order_date),2) as cum_revenue
	from(select
	o.order_date,
	round(sum(od.quantity*p.price),2) as revenue
	from order_details od
	join pizzas p
	on od.pizza_id= p.pizza_id
	join orders o
	on od.order_id= o.order_id
	group by o.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	21526.4
2015-01-10	23990.35
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.3

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
      pt.name,pt.category,round(sum(od.quantity*p.price),2) as revenue
      from pizza_types pt join pizzas p
      on pt.pizza_type_id = p.pizza_type_id
      join order_details od
      on p.pizza_id = od.pizza_id
      group by  pt.name,pt.category
      ) 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	32265.7
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

Thank You!