

PIZZA.COM
SALESIQL
PROJECT



INTRODUCTION

HELLO MYSELF SHIVAM MITTAL STUDENT OF PUNJAB
ENGINEERING COLLEGE . I MADE A PROJECT ON ANALYSIS OF
PIZZA SALES WITH THE USE OF SQL IN MY SQL

Let's start our adventure in the world of pizza!

QUESTIONS

Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

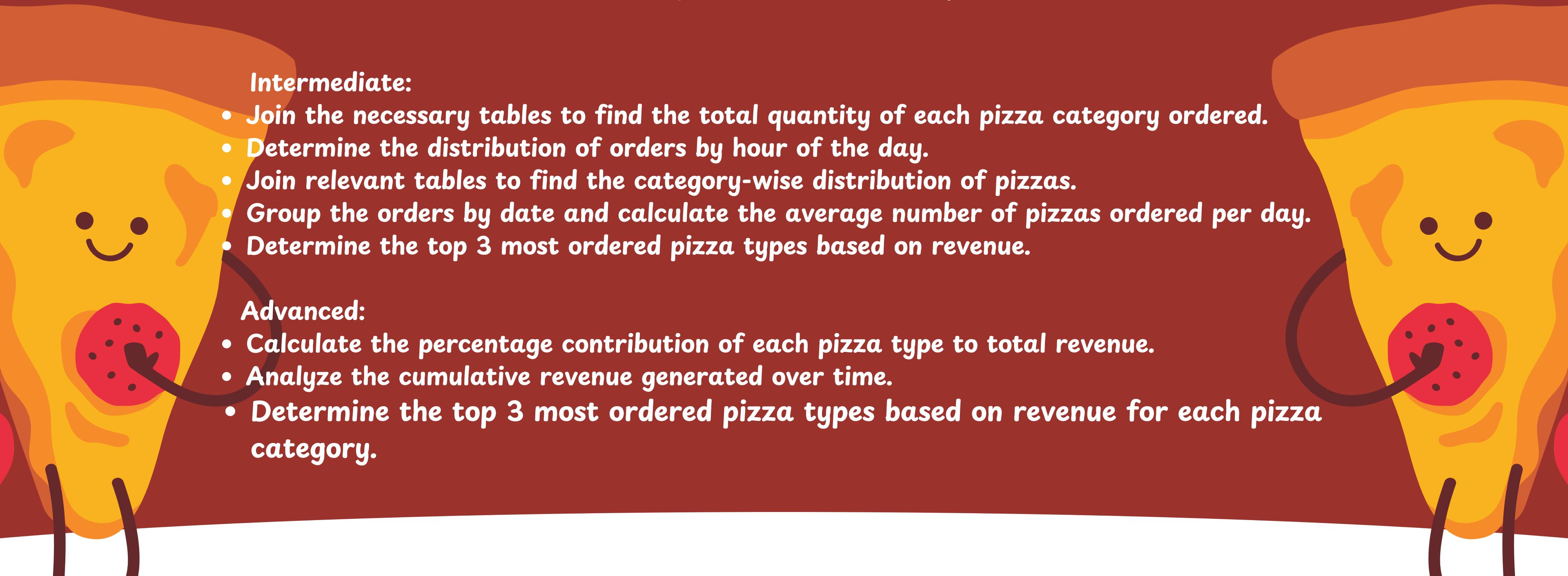


Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

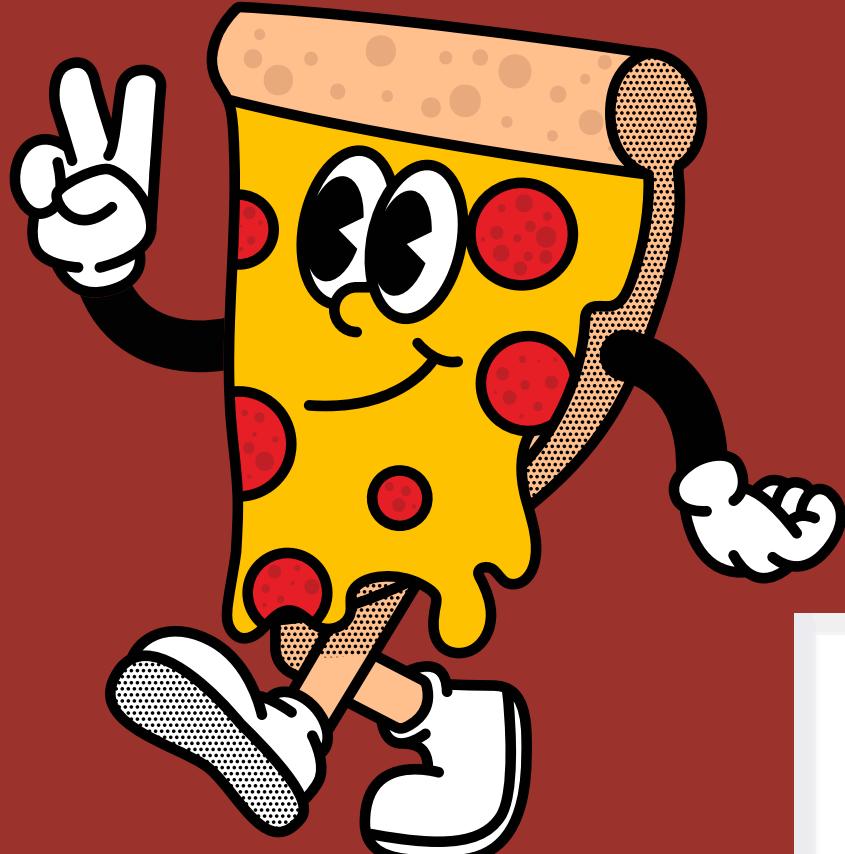


```
1 --- Retrieve the total number of orders placed?
```

```
2 • SELECT count(order_id) as total_orders FROM orders;
```

	total_orders
--	--------------

▶	1708
---	------



```
1 --- Calculate the total revenue generated from pizza sales?
```

```
2 • SELECT  
3     ROUND(SUM(order_details.quantity * pizzas.price),0) AS total_revenue  
4 FROM  
5     order_details  
6     JOIN  
7     pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

Result Grid |



Filter Rows:

Export:



Wrap Cell Content:

	total_revenue
--	---------------

▶	298682
---	--------



```
1 -- Identify the highest-priced pizza?  
2 • SELECT  
3     pizza_types.name, pizzas.price  
4   FROM  
5     pizza_types  
6       JOIN  
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
8 ORDER BY pizzas.price DESC  
9 LIMIT 1;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch

name	price
The Greek Pizza	35.95



```
1 -- Identify the most common pizza size ordered?  
2 • SELECT  
3     pizzas.size AS size, COUNT(pizzas.size) AS quantity  
4   FROM  
5     order_details  
6       JOIN  
7     pizzas ON order_details.pizza_id = pizzas.pizza_id  
8 GROUP BY pizzas.size  
9 ORDER BY quantity DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch

size	quantity
L	6759
M	5602
S	5180
XL	214
XXL	13

```
1 -- List the top 5 most ordered pizza types along with their quantities?
2 • SELECT
3     pizza_types.name, SUM(order_details.quantity) AS quantity
4 FROM
5     pizza_types
6     JOIN
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8     JOIN
9     order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizza_types.name
11 ORDER BY SUM(order_details.quantity) DESC
12 LIMIT 5;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

name	quantity
The Barbecue Chicken Pizza	931
The Hawaiian Pizza	890
The Pepperoni Pizza	885
The Classic Deluxe Pizza	857
The California Chicken Pizza	857



```
1 -- Join the necessary tables to find the total quantity of each pizza category ordered:  
2 • SELECT  
3     pizza_types.category,  
4     SUM(order_details.quantity) AS quantity  
5 FROM  
6     pizza_types  
7     JOIN  
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9     JOIN  
10    order_details ON order_details.pizza_id = pizzas.pizza_id  
11 GROUP BY pizza_types.category  
12 ORDER BY quantity DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

category	quantity
Classic	5375
Supreme	4347
Veggie	4336
Chicken	4044



```
1   --- Determine the distribution of orders by hour of the day.  
2 • SELECT  
3       HOUR(order_time) AS hours,  
4       COUNT(order_id) AS distribution_of_orders  
5   FROM  
6       orders  
7   GROUP BY hours;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	hours	distribution_of_orders
▶	11	93
	12	201
	13	194
	14	155
	15	127
	16	153
	17	182
	18	187
	19	164
	20	123
	21	81
	22	47
	23	1



```
1 --- Join relevant tables to find the category-wise distribution of pizzas.  
2 • SELECT  
3     COUNT(name), category  
4 FROM  
5     pizza_types  
6 GROUP BY category;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

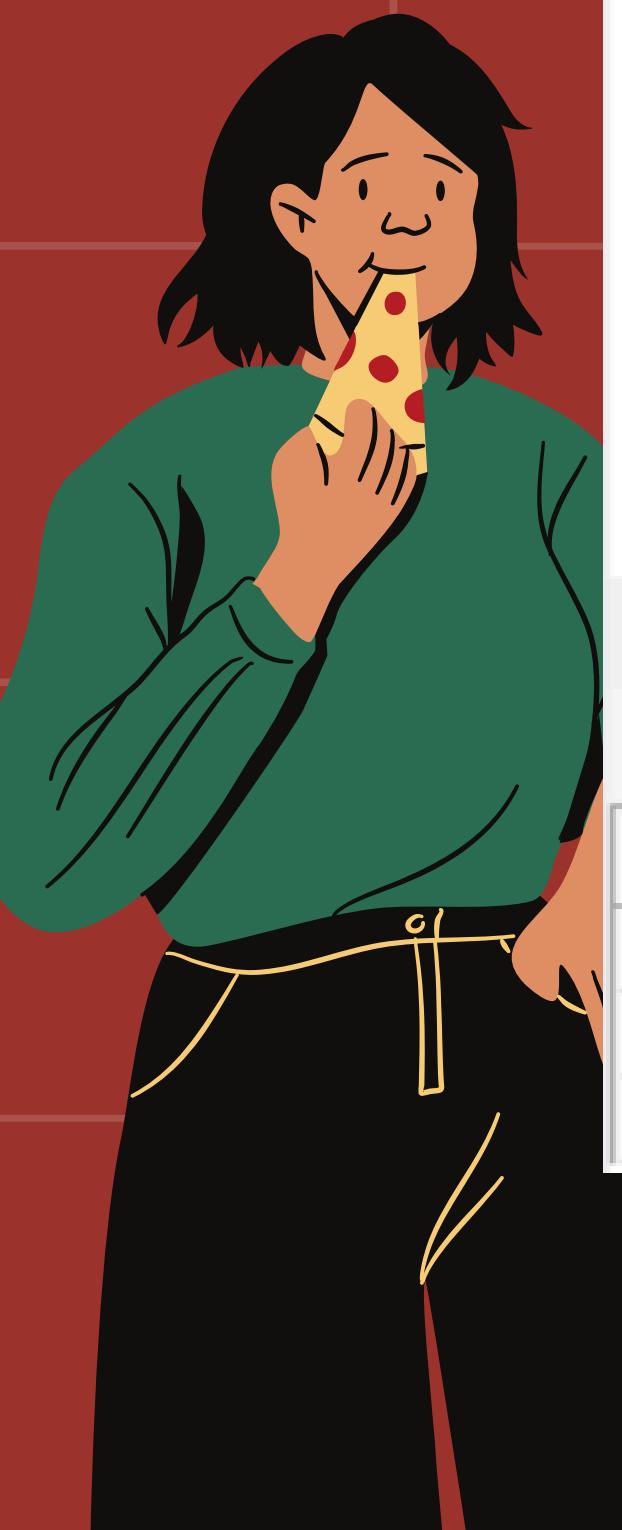
	COUNT(name)	category
6		Chicken
8		Classic
9		Supreme
9		Veggie

```
1  -- Group the orders by date and
2  -- calculate the average number of pizzas ordered per day.
3 • SELECT
4      ROUND(AVG(daily_order_sum), 0) as avg_pizza_ordered_per_day
5  FROM
6  (SELECT
7      orders.order_date, SUM(order_details.quantity) as daily_order_sum
8  FROM
9      orders
10     JOIN order_details ON
11         orders.order_id = order_details.order_id
12     GROUP BY orders.order_date) as order_quantity;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	avg_pizza_ordered_per_day
135	135

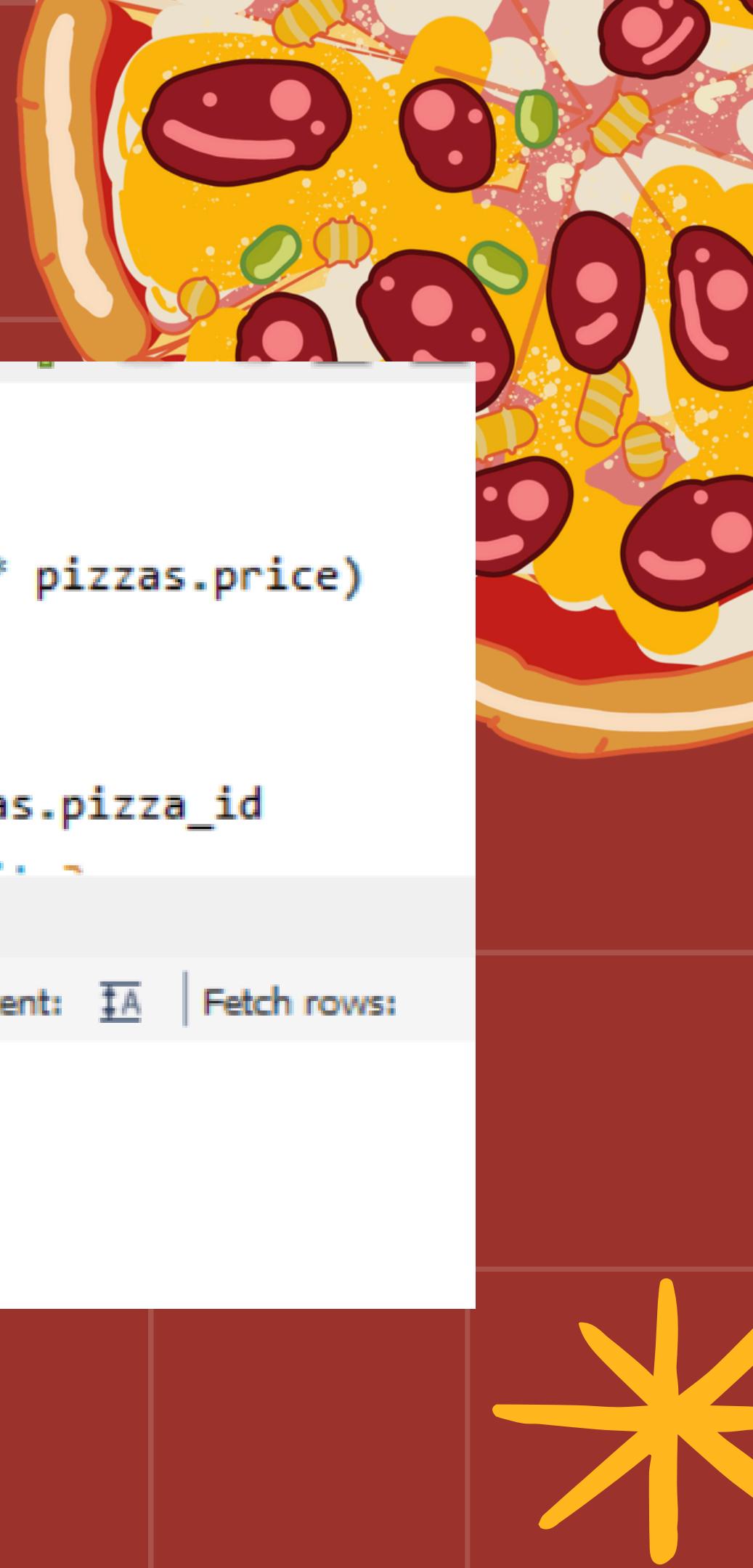




```
1 --- Determine the top 3 most
2 --- ordered pizza types based on revenue.
3 • SELECT pizza_types.name, sum(order_details.quantity * pizzas.price)
4   as revenue from pizza_types join pizzas on
5     pizzas.pizza_type_id = pizza_types.pizza_type_id
6   join order_details on order_details.pizza_id = pizzas.pizza_id
7   . . .
8
9
10
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	name	revenue
▶	The Barbecue Chicken Pizza	16430.25
	The Thai Chicken Pizza	15439.25
	The California Chicken Pizza	14990.75



```
1 -- Calculate the percentage contribution of each pizza type to total revenue
2
3 • SELECT
4     pizza_types.category,
5     CONCAT((ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
6             ROUND(SUM(order_details.quantity * pizzas.price),
7             0) AS total_revenue
8
9             FROM order_details
10            JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id),2) * 100),'%') AS revenue
10
11    FROM
12        pizza_types
13        JOIN
14            pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
15        JOIN
16            order_details ON order_details.pizza_id = pizzas.pizza_id
17    GROUP BY pizza_types.category
18    ORDER BY revenue DESC;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

category	revenue
Classic	27%
Supreme	25%
Veggie	24%
Chicken	24%



```
1 -- Analyze the cumulative revenue generated over time.  
2 • select order_date, round(sum(revenue) over (order by order_date),1 ) as cum_revenue  
3   from (select orders.order_date , sum(order_details.quantity * pizzas.price ) as revenue  
4     from order_details join pizzas on order_details.pizza_id = pizzas.pizza_id join  
5      orders on orders.order_id = order_details.order_id group by orders.order_date) as sales
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	order_date	cum_revenue
▶	2015-01-01	2713.9
	2015-01-02	5445.8
	2015-01-03	8108.2
	2015-01-04	9863.6
	2015-01-05	11929.6
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399
	2015-01-09	21526.4
	2015-01-10	23990.4
	2015-01-11	25862.6
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.7
	2015-01-17	39001.8
	2015-01-18	40978.6
	2015-01-19	43365.8
	2015-01-20	45763.7
	2015-01-21	47804.2
	2015-01-22	50300.9

Result 1



```
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name , revenue , rnk from
3   ( select category , name , revenue , rank() over ( partition by category order by revenue desc )
4     as rnk from
5   (select pizza_types.category , pizza_types.name , sum((order_details.quantity) * pizzas.price )
6     as revenue
7   from pizza_types join
8     pizzas  on pizza_types.pizza_type_id = pizzas.pizza_type_id join
9     order_details
10    on order_details.pizza_id = pizzas.pizza_id
11   group by pizza_types.category , pizza_types.name) as a) as b
12 where rnk <= 3;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	name	revenue	rnk
▶	The Barbecue Chicken Pizza	16430.25	1
	The Thai Chicken Pizza	15439.25	2
	The California Chicken Pizza	14990.75	3
▶	The Classic Deluxe Pizza	13318	1
	The Hawaiian Pizza	11773.5	2
	The Pepperoni Pizza	11037.75	3
	The Spicy Italian Pizza	12579.5	1
	The Italian Supreme Pizza	12210.75	2
	The Sicilian Pizza	11059	3
	The Four Cheese Pizza	12052.050000000056	1
	The Mexicana Pizza	9676.25	2
	The Five Cheese Pizza	9675.5	3



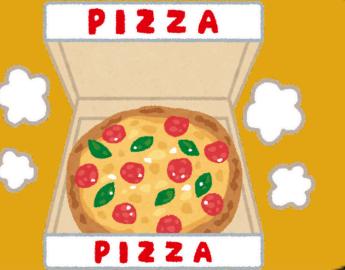
POWER BI DASHBOARD

NEXT ➔





PIZZA.COM SALES WITH MY SQL



CATEGORY

TOTAL ORDERS

1708

TOTAL REVENUE

298.68K

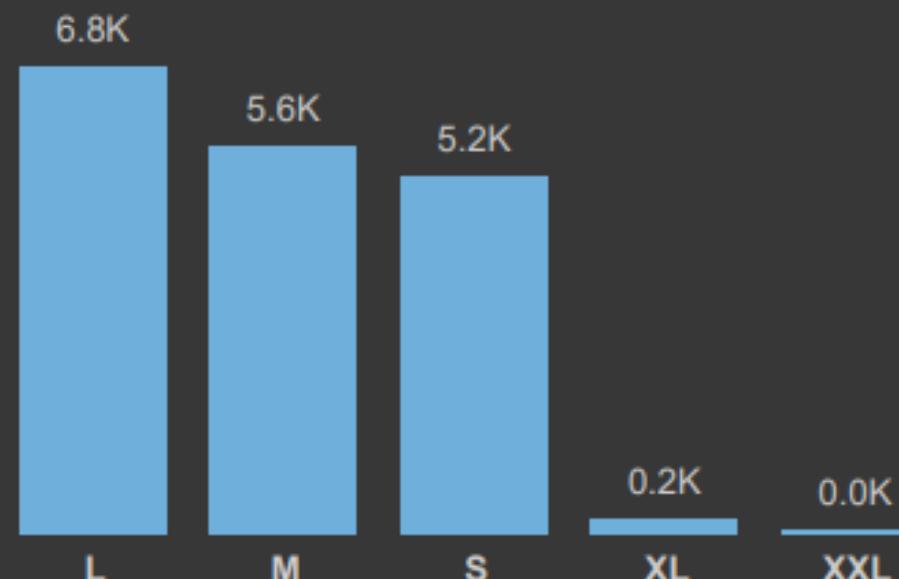
HIGHEST PRICE PIZZA
GREEK PIZZA

36.0

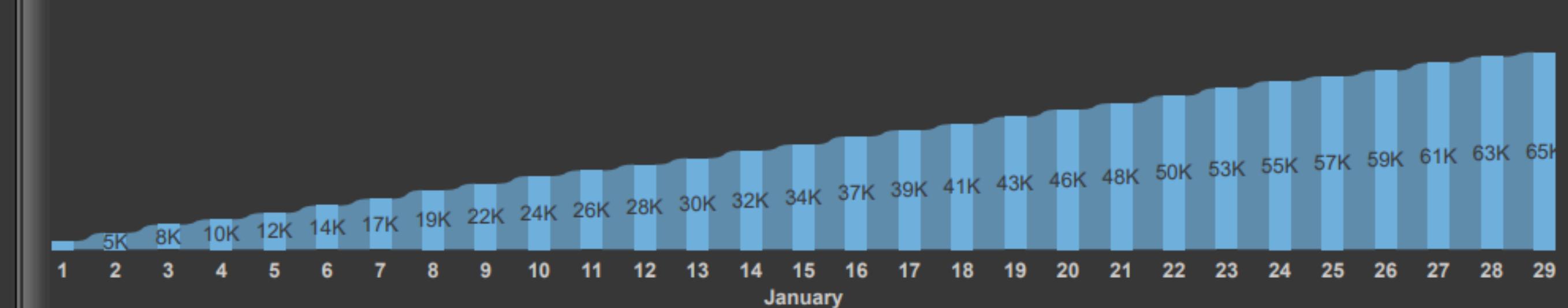
AVG PIZZA ORDERED
PER DAY

135.0

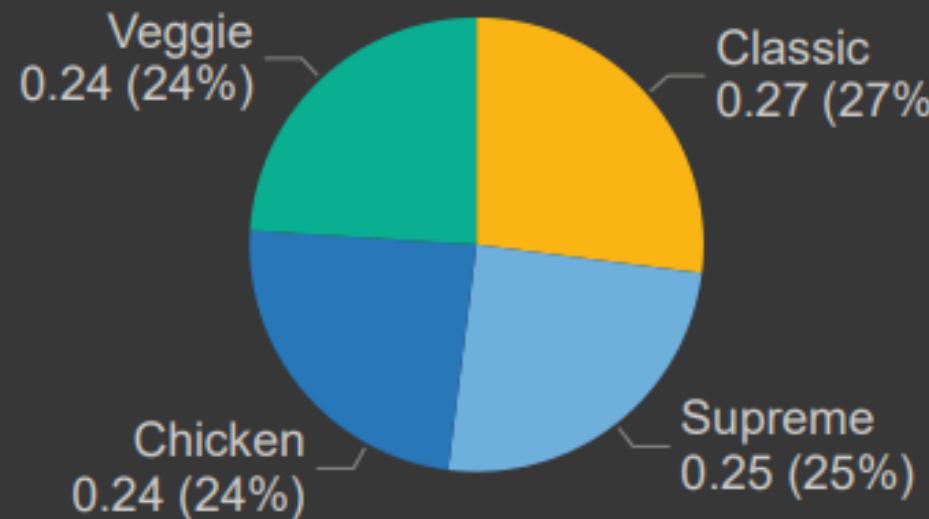
MOST ORDERED SIZE



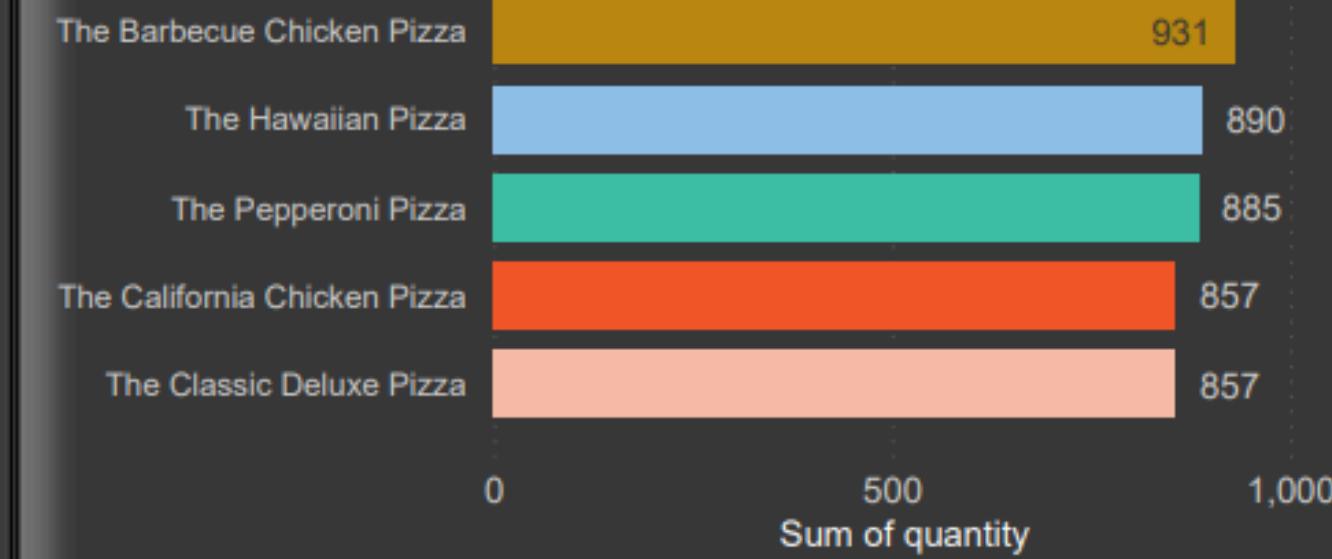
CUMULATIVE REVENUE OF JANUARY



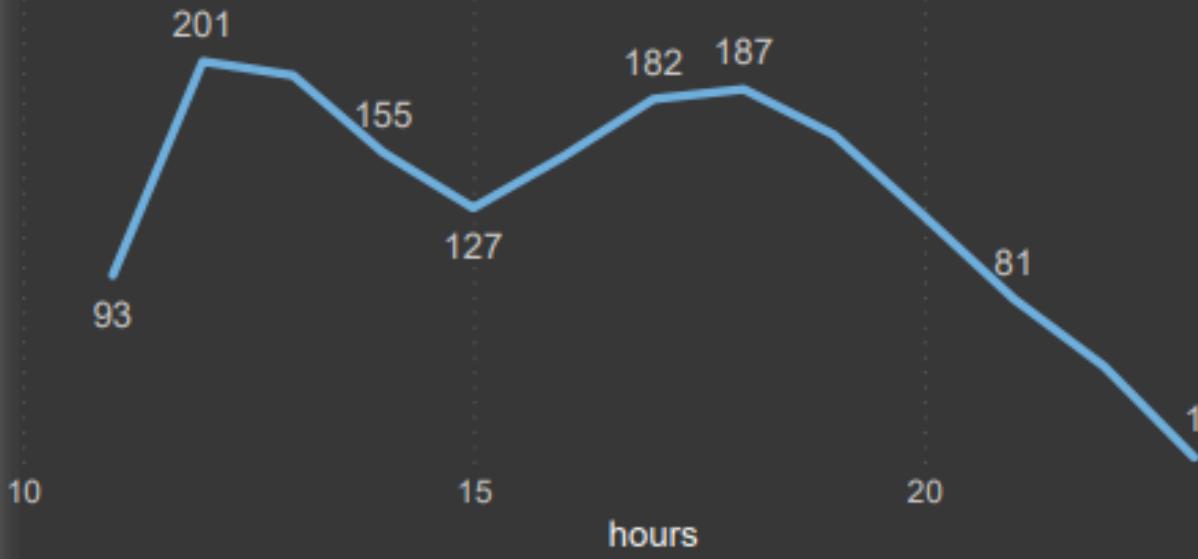
Sum of revenue by category



TOP 5 PIZZA ORDERED



PEAK HOURS OF ORDERS



A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green sweater holds a slice of pizza. To the right, another person with glasses and a green sweater holds a small wrapped gift. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck. The reindeer has large, expressive eyes and a small smile. The background is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU