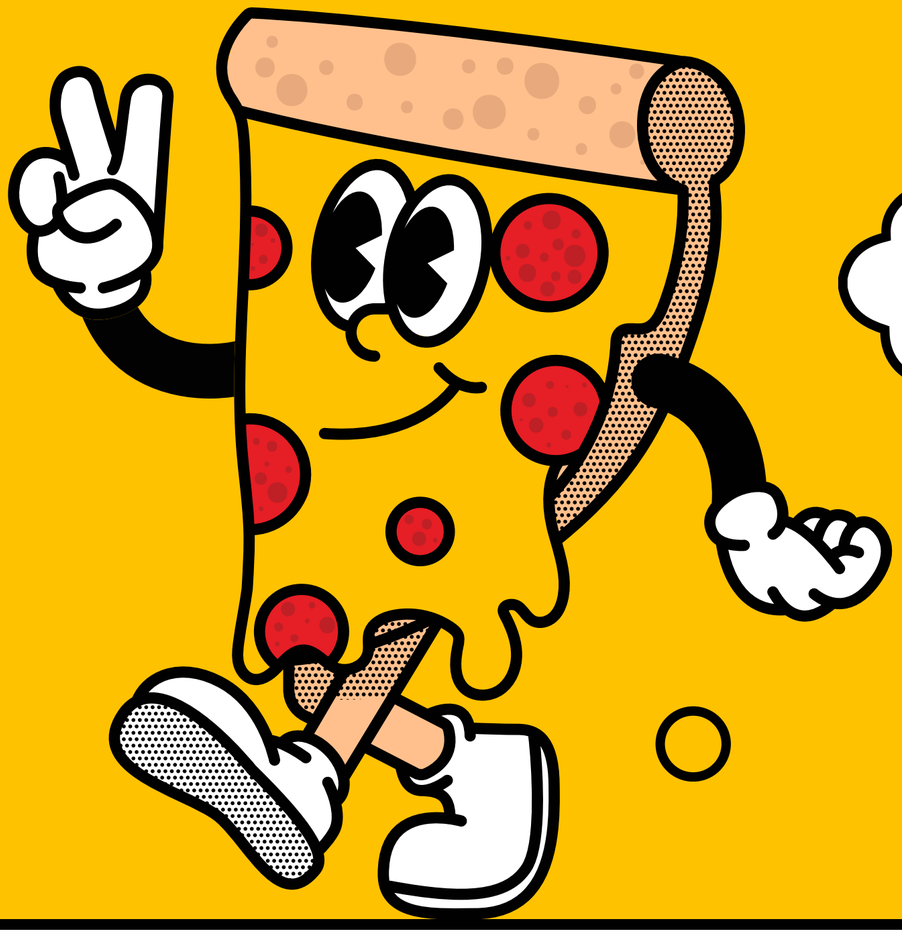
Pizza Sales Analysis

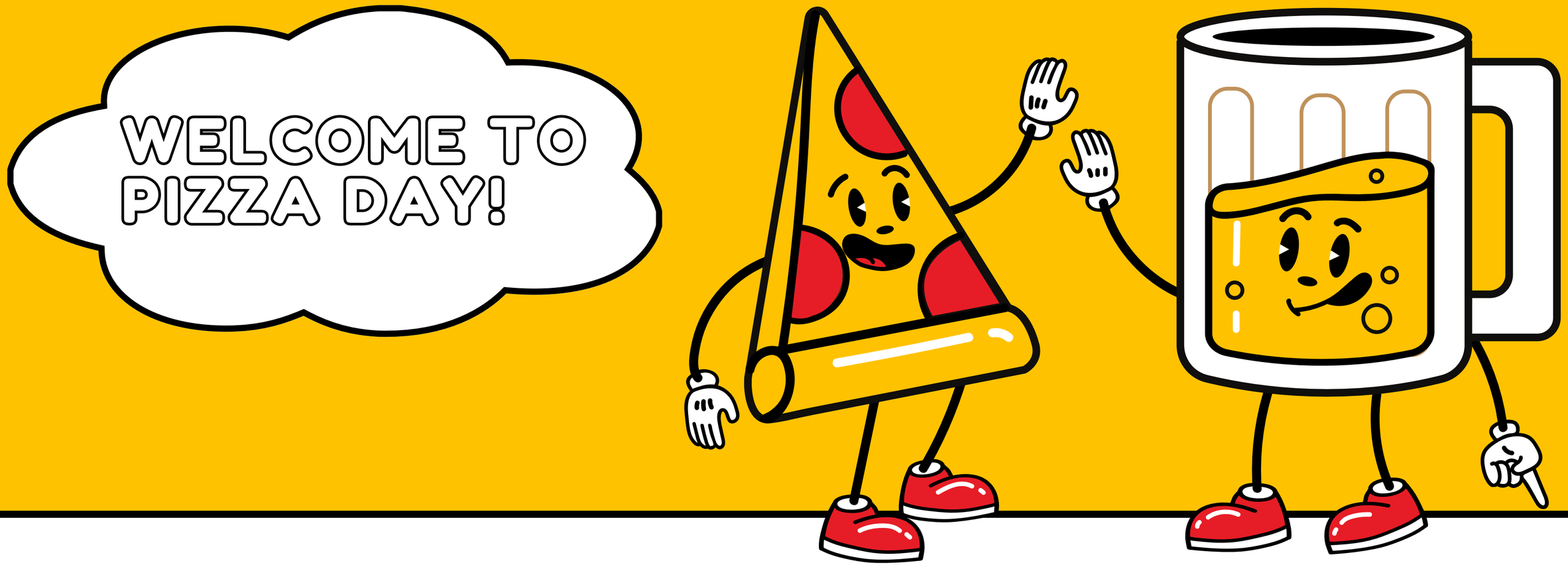
A SLICE OF HAPPINESS



SQL PROJECT

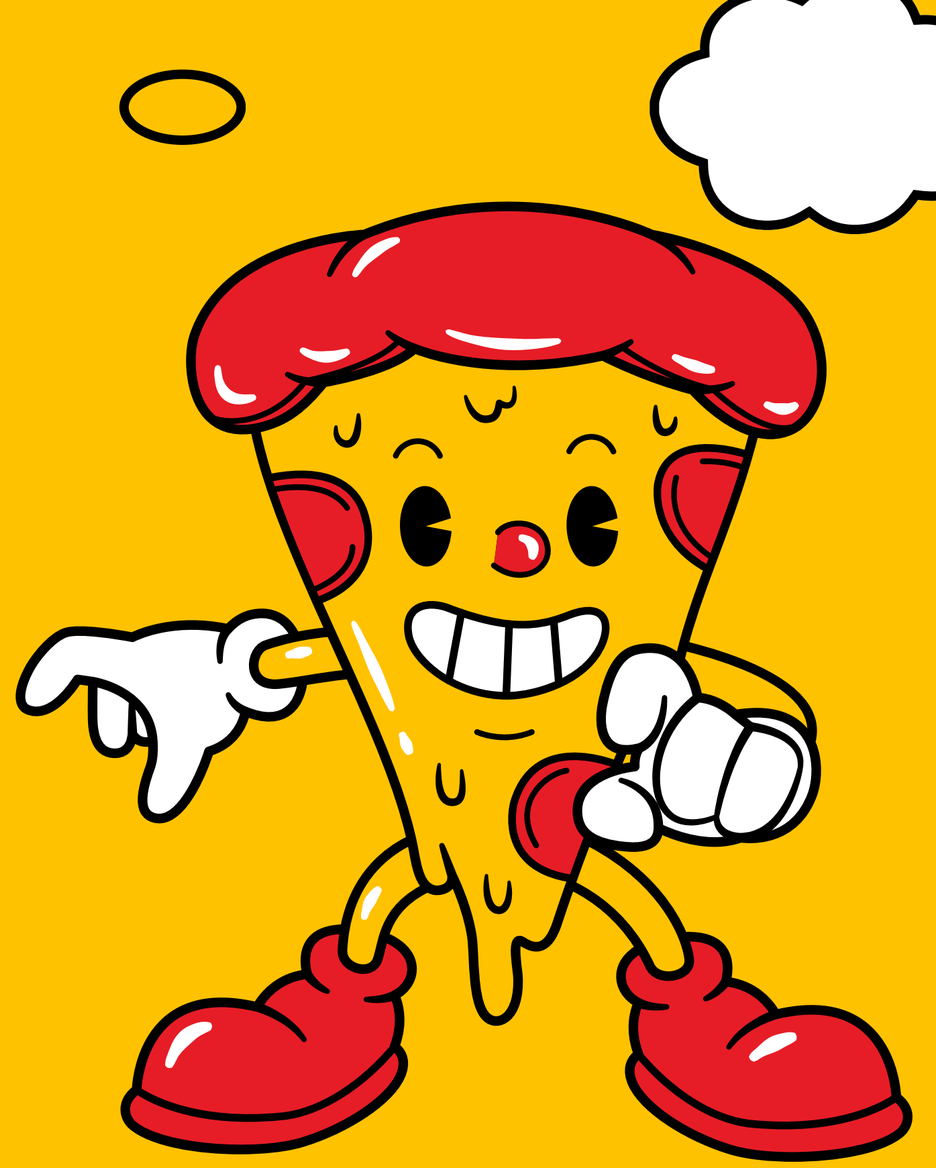
BY: Puja Ghosh

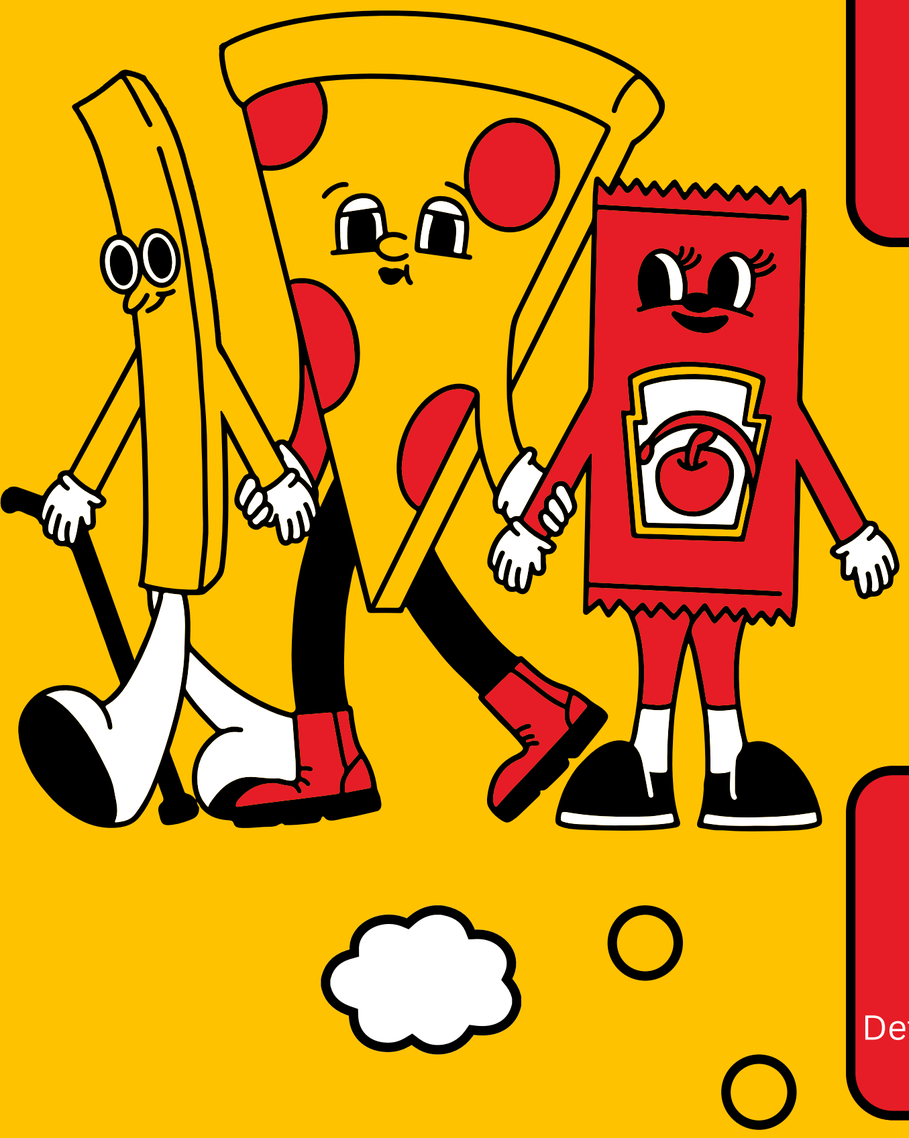
Hello....my name is Puja Ghosh....
in this project i have utilized SQL query to solve questions related to pizza sales.



Aim of Project

The aim of the project is to analyze pizza sales data to identify trends and provide actionable insights that can help to increasesalers and object to uncover key matrics and patterns within the sales data by laveraging SQL queries in MYSQL.





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.

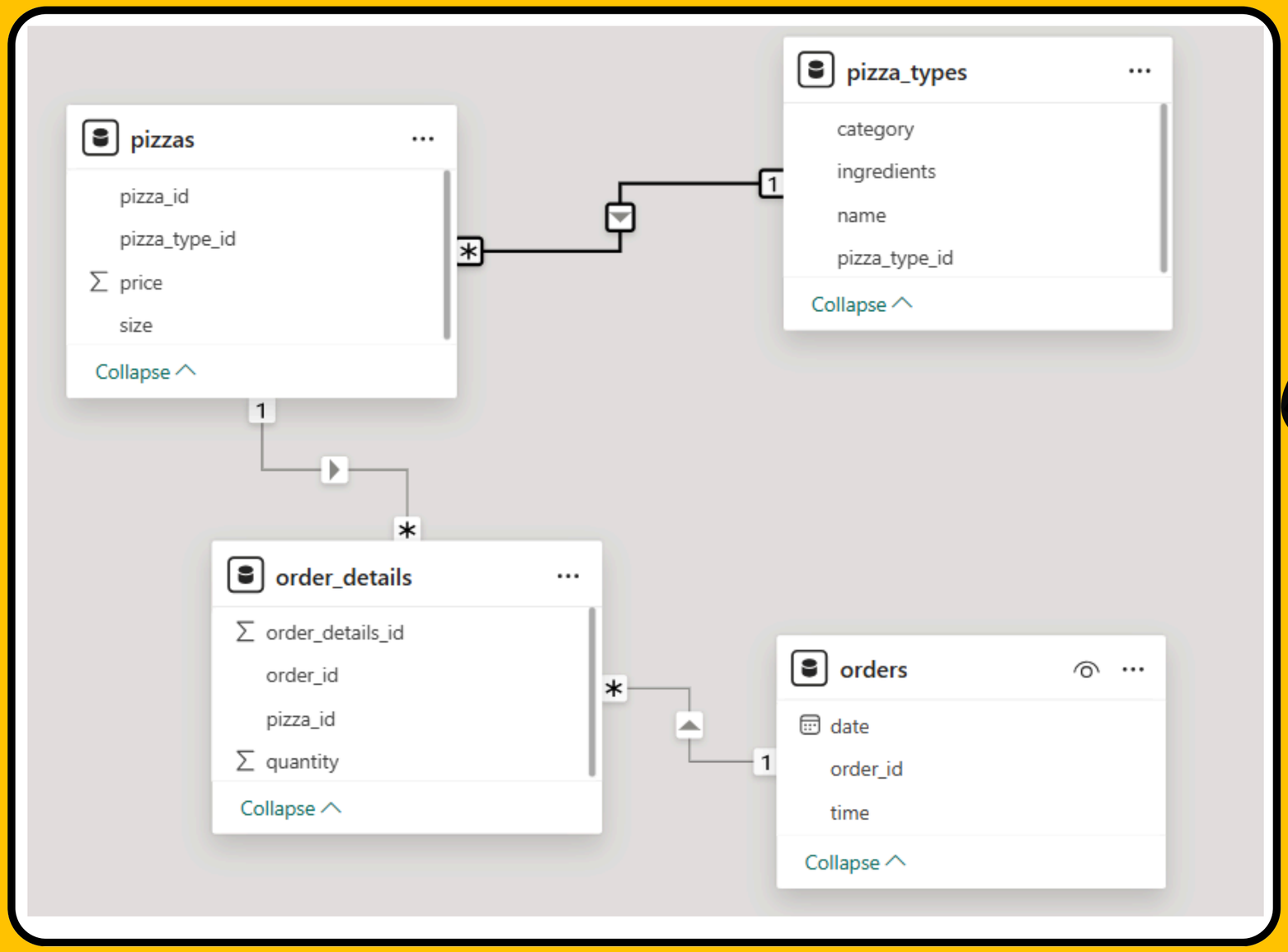
Intermediate:

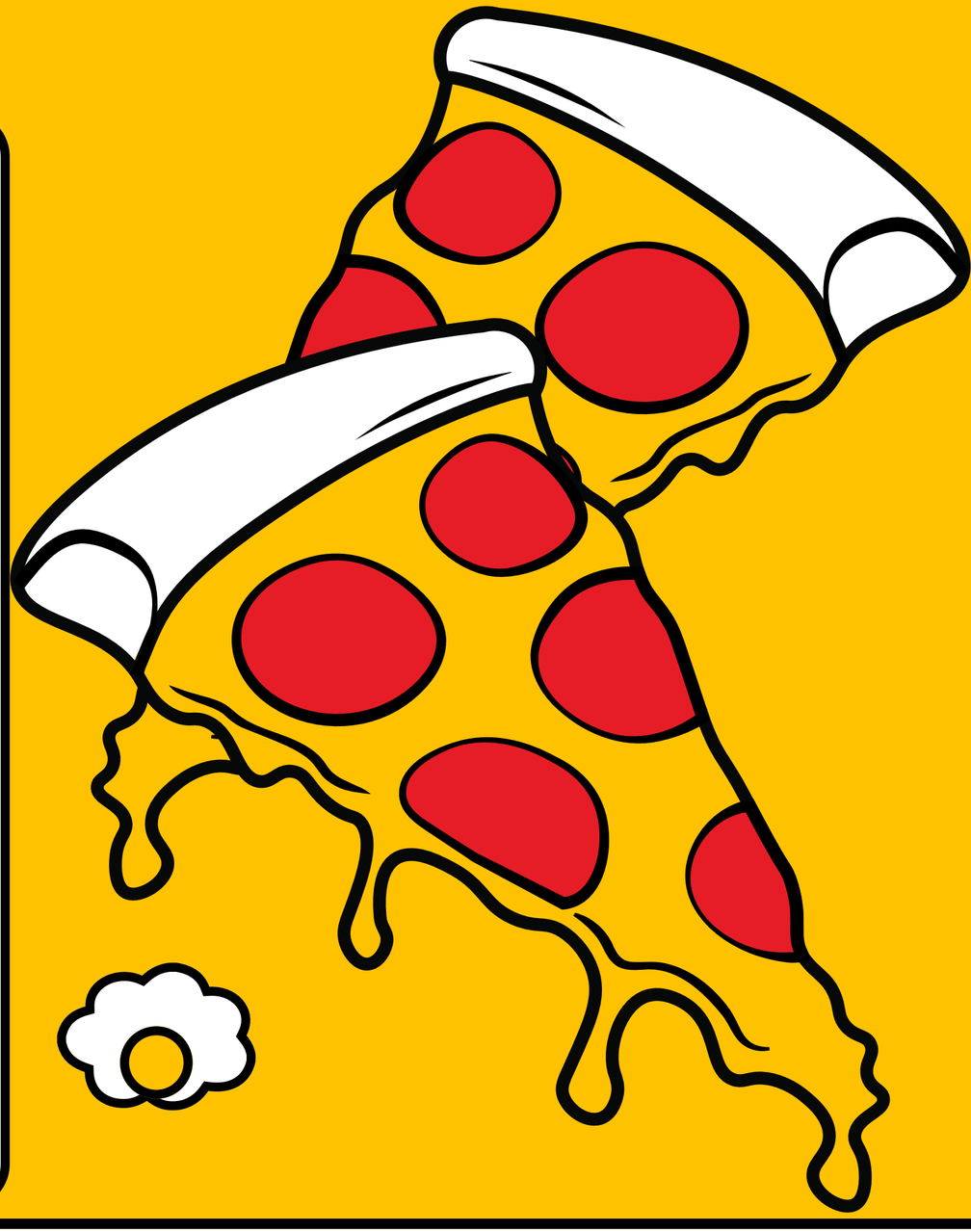
Join the necessary tables to find the total quantity of each pizza category ordered.
Determine the distribution of orders by hour of the day.
Join relevant tables to find the category-wise distribution of pizzas.
Group the orders by date and calculate the average number of pizzas ordered per day.
Determine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

SCHEMA MODEL





Basic Questions:

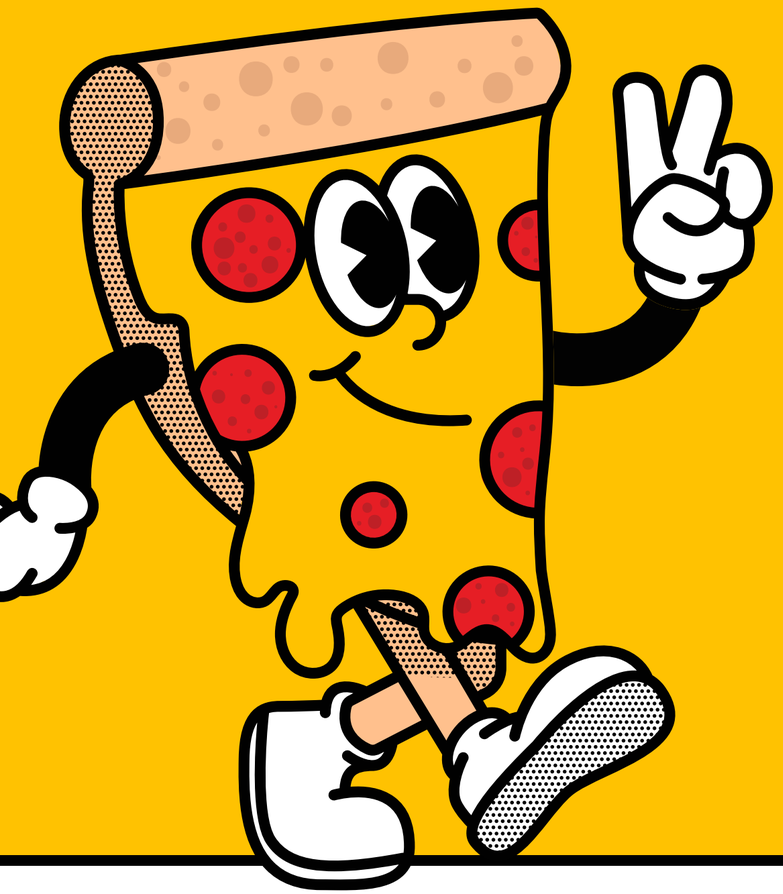
1.Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS Total_orders
FROM
    orders;
```

Result Grid

Total_orders
21350

2.Calculate the total revenue generated from pizza sales.



```sql
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;
```

Result Grid

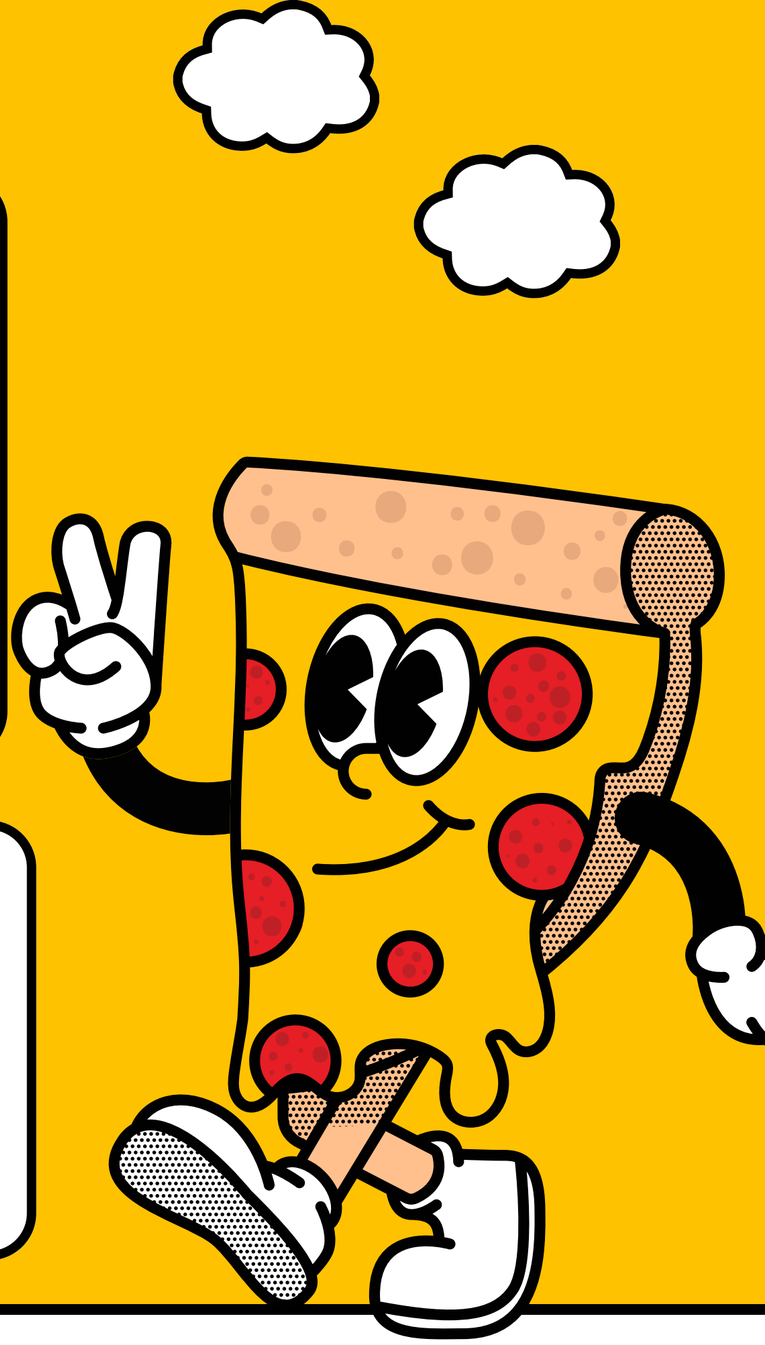
	Total_Sales
▶	817860.05

3.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 Rows:

name	price
The Greek Pizza	35.95

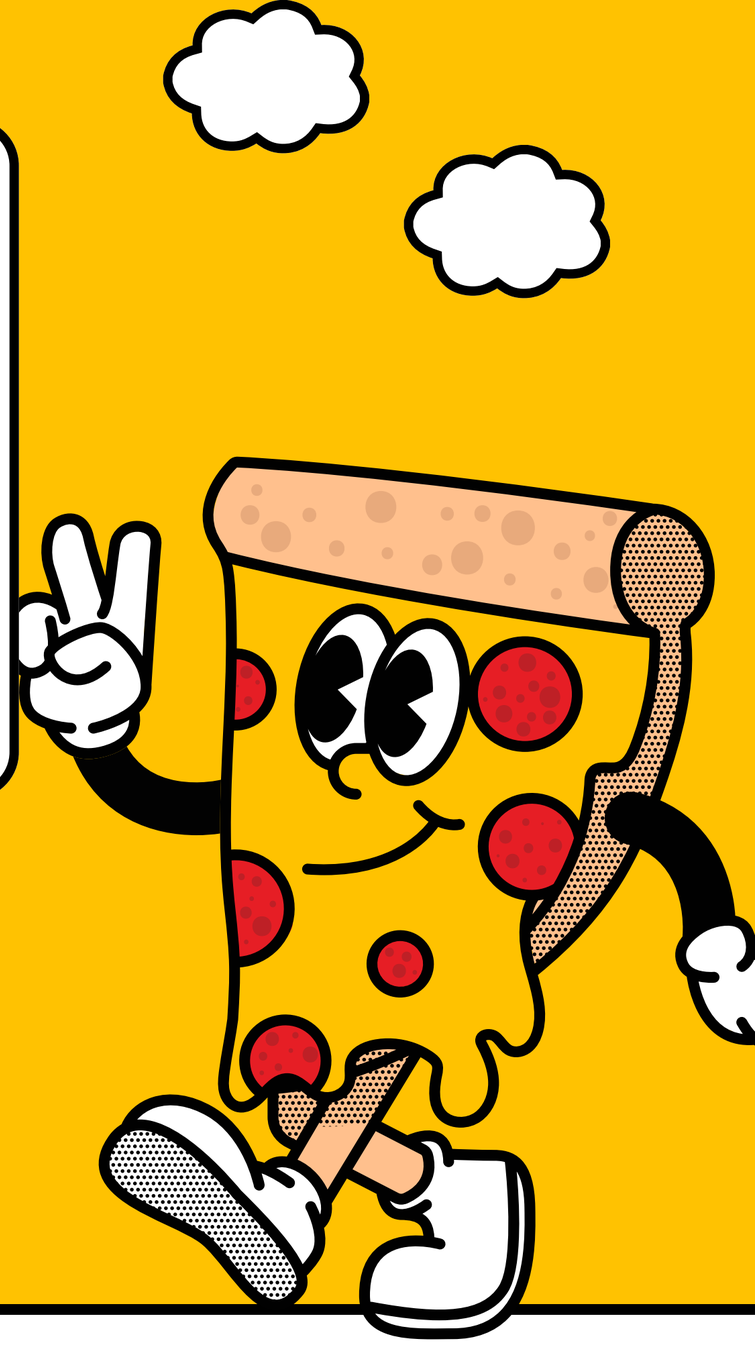


4.Identify the most common pizza size ordered.

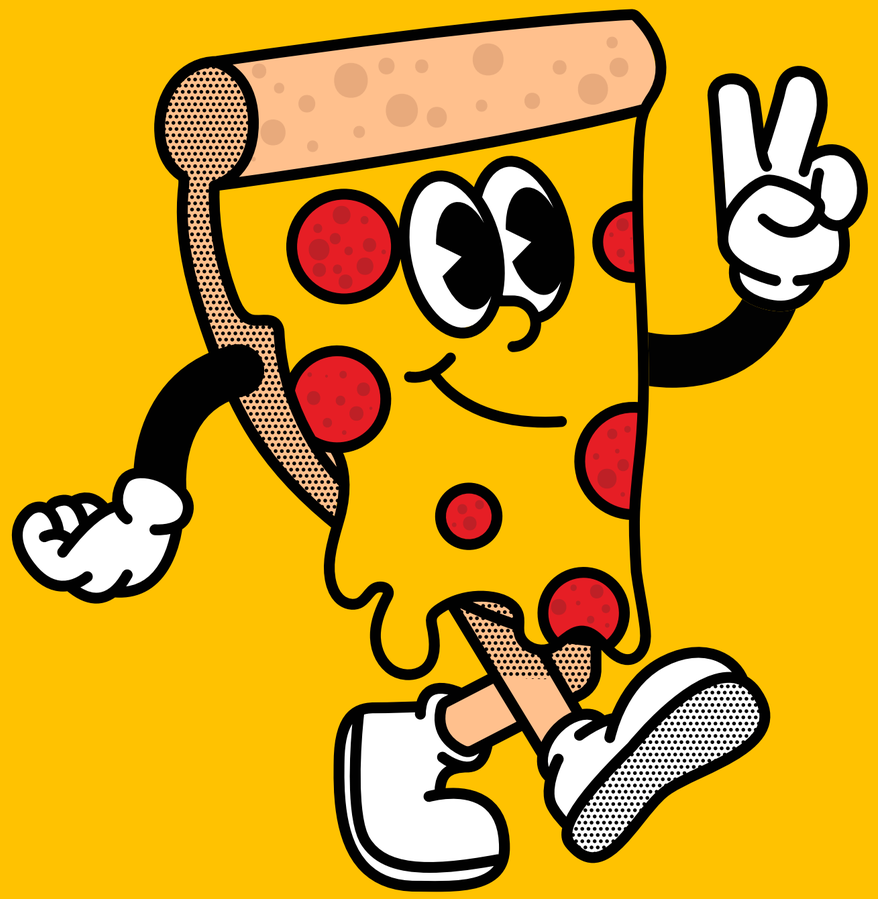
```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

Result Grid | Fi

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



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



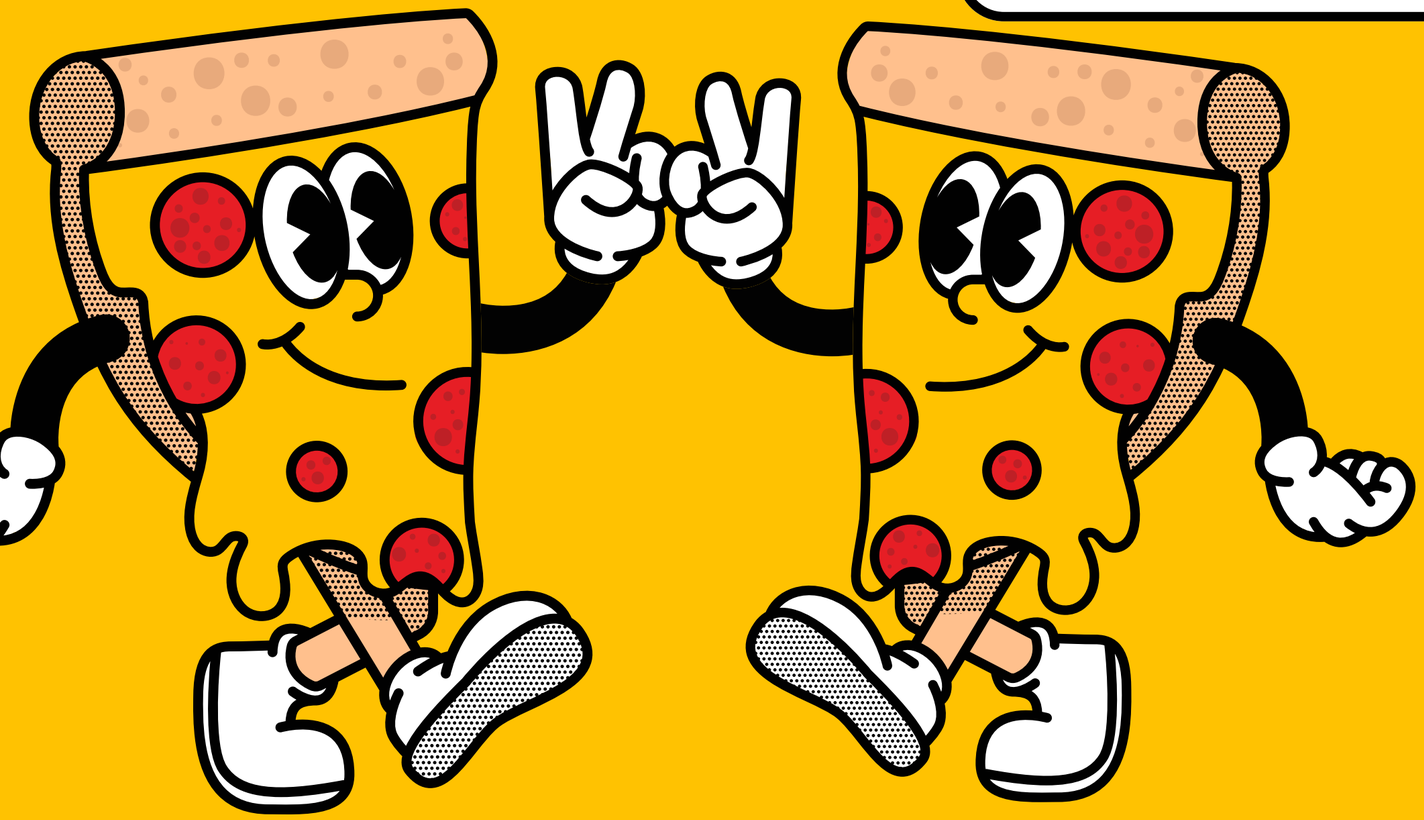
```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT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

Result Grid | Filter Rows:

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

6.Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY hour;
```



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

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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

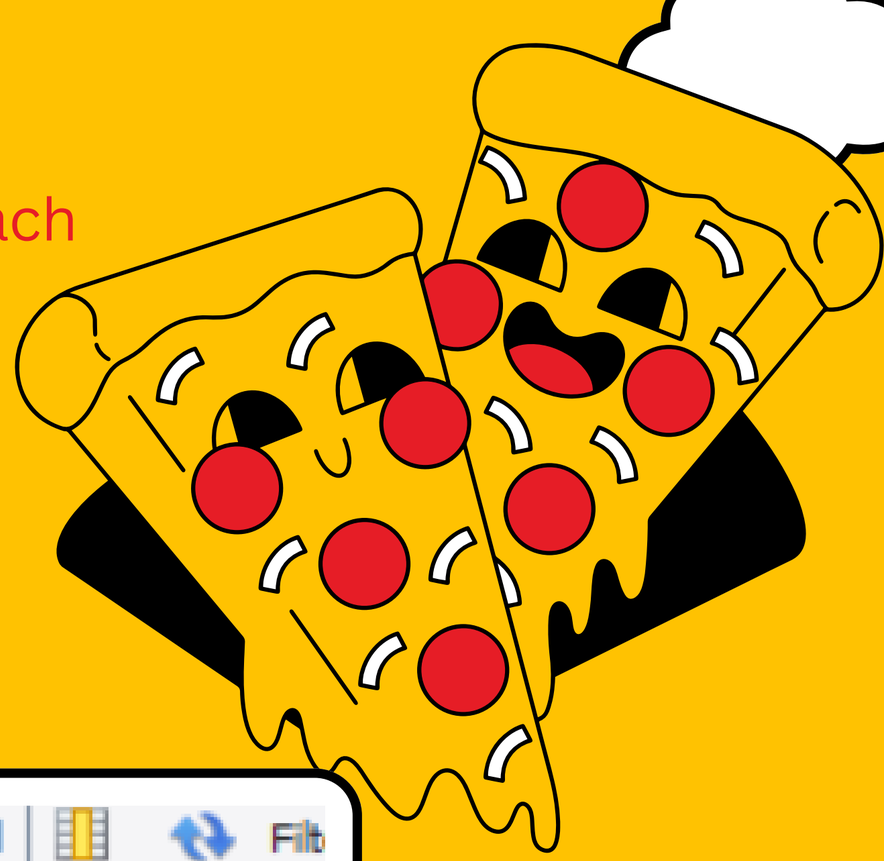
Result Grid | Filter Rows:

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

Intermediate Questions:

1.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;
```

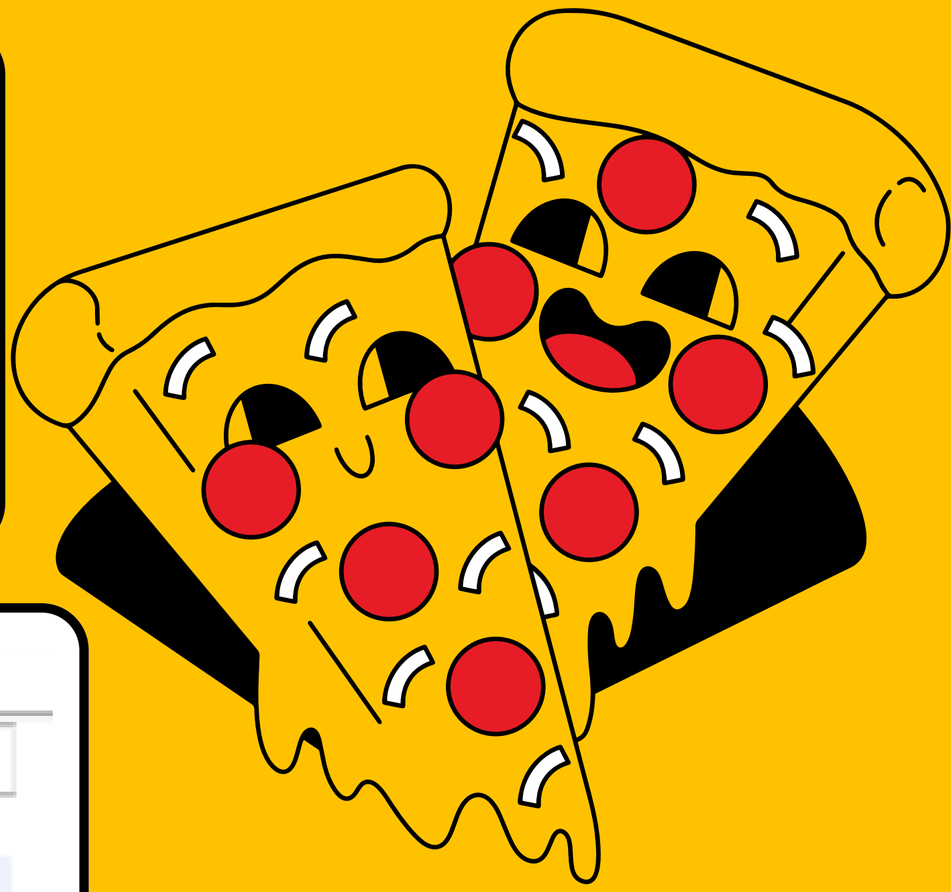


Result Grid | Filt

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

2.Determine the top 3 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 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 revenue DESC
LIMIT 3;
```

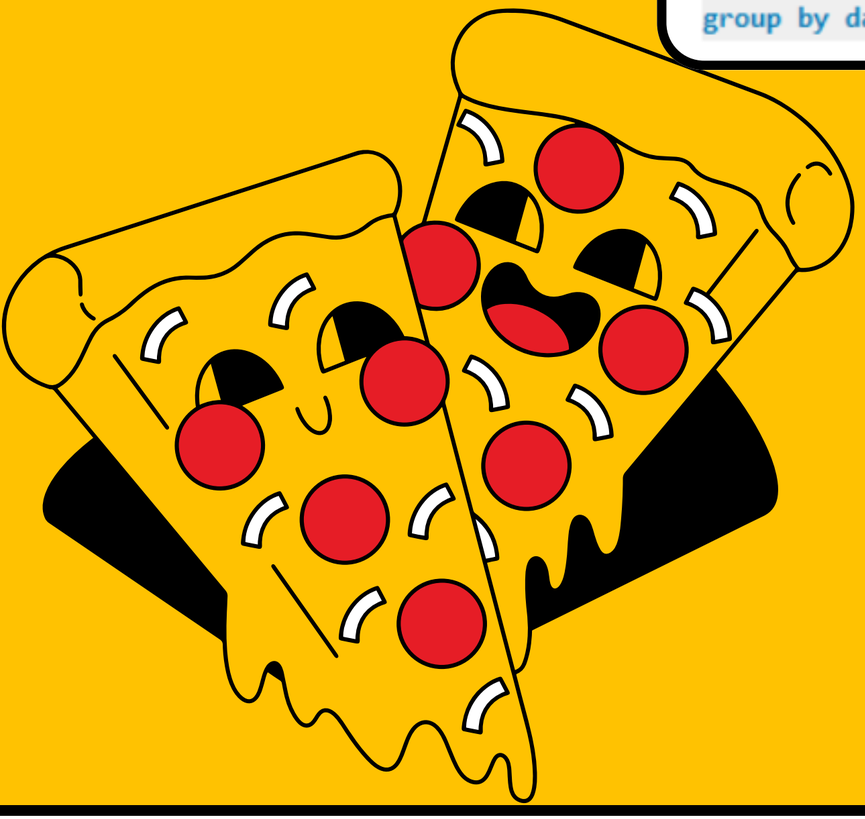


Result Grid | Filter Rows:

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

3.Analyze the cumulative revenue generated over time.

```sql
select date, sum(revenue) over(order by date) as cum_revenue
from(SELECT
   orders.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 date ) as sales ;
```



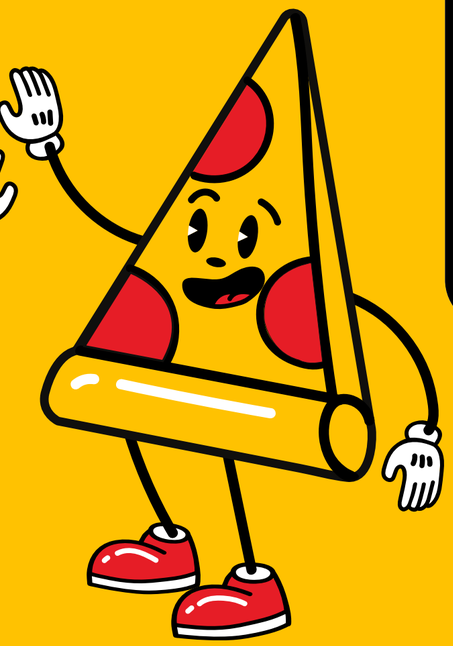
Result Grid | Filter Rows:

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
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001

Advanced Questions:

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

```sql
SELECT
    pizza_types.category,
    ROUND((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,
            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;
```

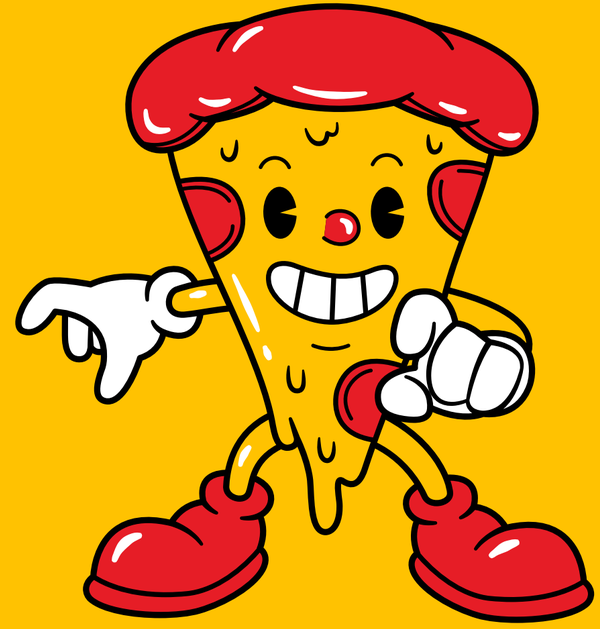


Result Grid

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

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

```sql
select name ,revenue
from
(select name,category,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.name, pizza_types.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	32265.7000000006
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

THANK YOU AND ENJOY A SLICE OF PIZZA!

PROJECT LINK