



PIZZA SALES

ANALYSIS

MySQL

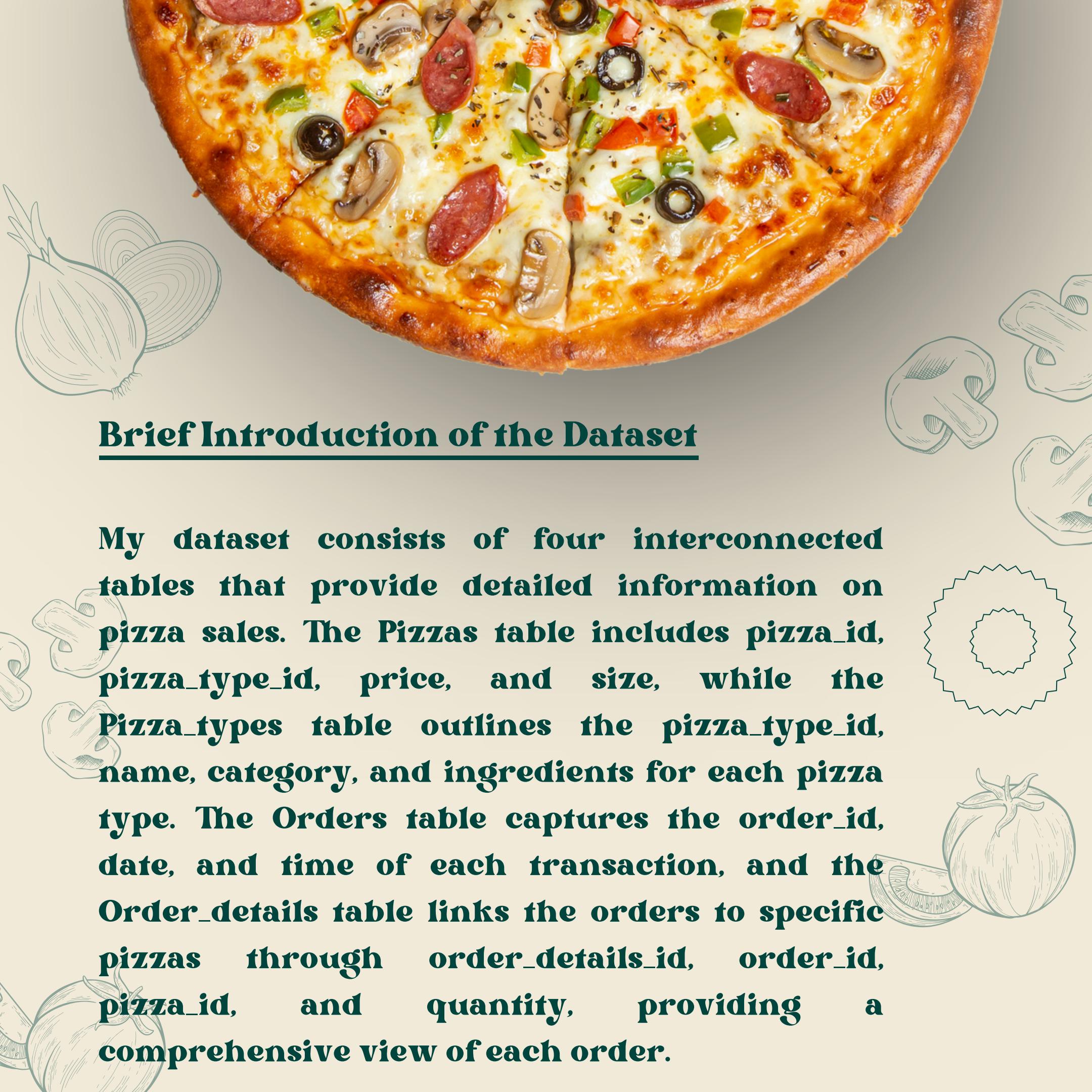
[HTTPS://GITHUB.COM/BUSHRAR1709](https://github.com/bushrar1709)



Hello,

I am Bushra Rahman, and I recently completed a SQL project using MySQL, where I analyzed pizza sales data.

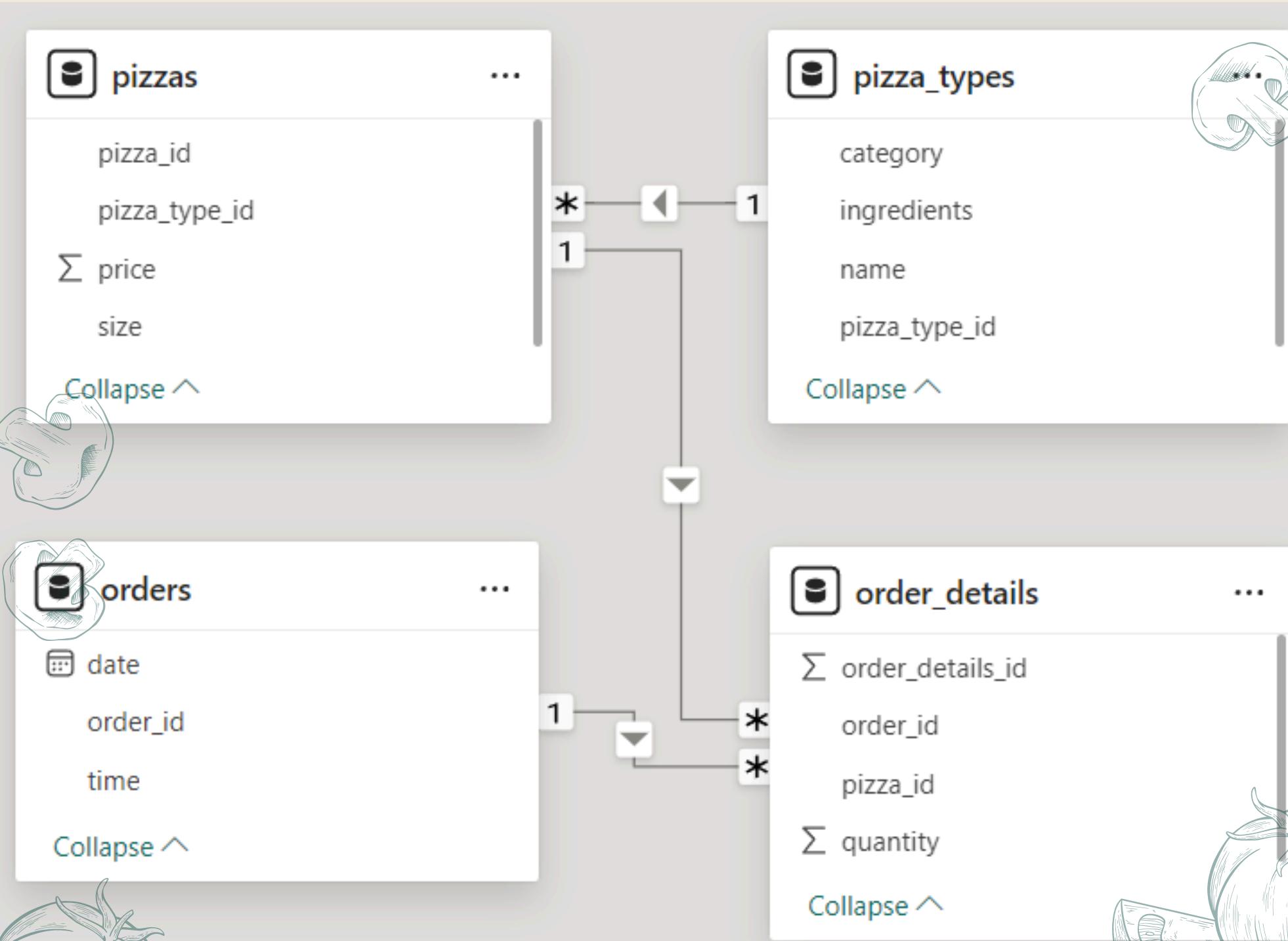
In this project, I tackled a variety of queries, ranging from basic to advanced levels, to uncover valuable insights and trends.



Brief Introduction of the Dataset

My dataset consists of four interconnected tables that provide detailed information on pizza sales. The **Pizzas** table includes **pizza_id**, **pizza_type_id**, **price**, and **size**, while the **Pizza_types** table outlines the **pizza_type_id**, **name**, **category**, and **ingredients** for each pizza type. The **Orders** table captures the **order_id**, **date**, and **time** of each transaction, and the **Order_details** table links the orders to specific pizzas through **order_details_id**, **order_id**, **pizza_id**, and **quantity**, providing a comprehensive view of each order.

Model view of the Dataset



1: Retrieve the total number of orders placed.

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

Result Grid	
	total_orders
▶	21350

2 : Calculate the total revenue generated from pizza sales.

```
SELECT  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        2) AS total_revenue  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

	total_revenue
▶	817860.05

3 : Identify the highest-priced pizza.

```
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 Row

	name	price
▶	The Greek Pizza	35.95

4: Identify the most common pizza size ordered.

```
SELECT
  *
FROM
  order_details;
SELECT
  *
FROM
  orders;
SELECT
  *
FROM
  pizza_types;
SELECT
  *
FROM
  pizzas;

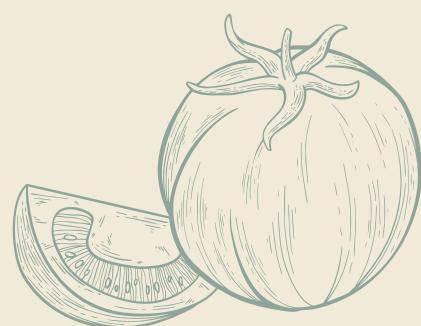
SELECT
  pizzas.size,
  COUNT(order_details.order_details_id) AS max_ordered
FROM
  pizzas
  JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(order_details.order_details_id) DESC;
```

	size	max_ordered
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

```
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;
```

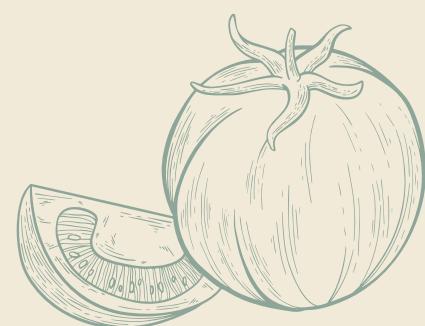
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371



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

```
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 SUM(order_details.quantity) DESC;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



7 : Determine the distribution of orders by hour of the day.

SELECT

HOUR(order_time) AS hours, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time);

	hours	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
	10	8
	9	1

8 : (Join relevant tables if needed) 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

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

SELECT

ROUND(AVG(quantity), 0) AS avg_order_quantity_per_day

FROM

(**SELECT**

orders.order_date, SUM(order_details.quantity) AS quantity

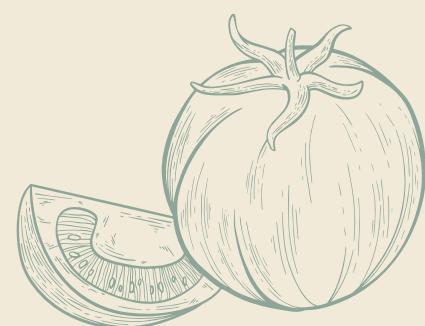
FROM

orders

JOIN order_details **ON** orders.order_id = order_details.order_id

GROUP BY orders.order_date) AS order_quantity;

	avg_order_quantity_per_day
▶	138



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

```
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;
```

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

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

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_revenue
    )
    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

12 : Analyze the cumulative revenue generated over time.

```
SELECT order_date, sum(revenue)
    OVER (ORDER BY order_date) AS cum_revenue
    FROM
        (SELECT orders.order_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 orders.order_date) AS sales;
```

	order_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	36330.50000000001

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

```
SELECT category, 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.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.category, pizza_types.name) AS a) AS b
WHERE rn <= 3 ;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5



THANK YOU