

Pizza sales analysis



INTRODUCTION

Project Overview

The Pizza Sales Analysis project focuses on analyzing pizza sales data to identify key trends and insights. By querying the database, we aim to uncover customer preferences, popular pizza types, and revenue patterns.

Objectives

Determine the total number of orders placed.

Calculate the total revenue generated.

Identify the most popular pizza types and sizes.

Analyze sales trends based on time and categories.

Tools Used

MySQL Workbench: For querying and analyzing the pizza sales data.

Canva: To design and present the findings in this report.

Retrieve the total numbers of order placed

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

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

▶	21350
---	-------

calculate the total revenue generated from pizza sales

```
3 •   SELECT
4   ROUND(SUM(order_details.quantity * pizzas.price),
5       2) AS total_sales
6
7   FROM
8   order_details
9   JOIN
10  pizzas ON pizzas.pizza_id = order_details.pizza_id
```



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

	total_sales
▶	817860.05

identify the highest priced pizza

```
3 • SELECT
4     pizza_types.name, pizzas.price
5 FROM
6     pizza_types
7     JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9     ORDER BY pizzas.price DESC
10    LIMIT 1;
```

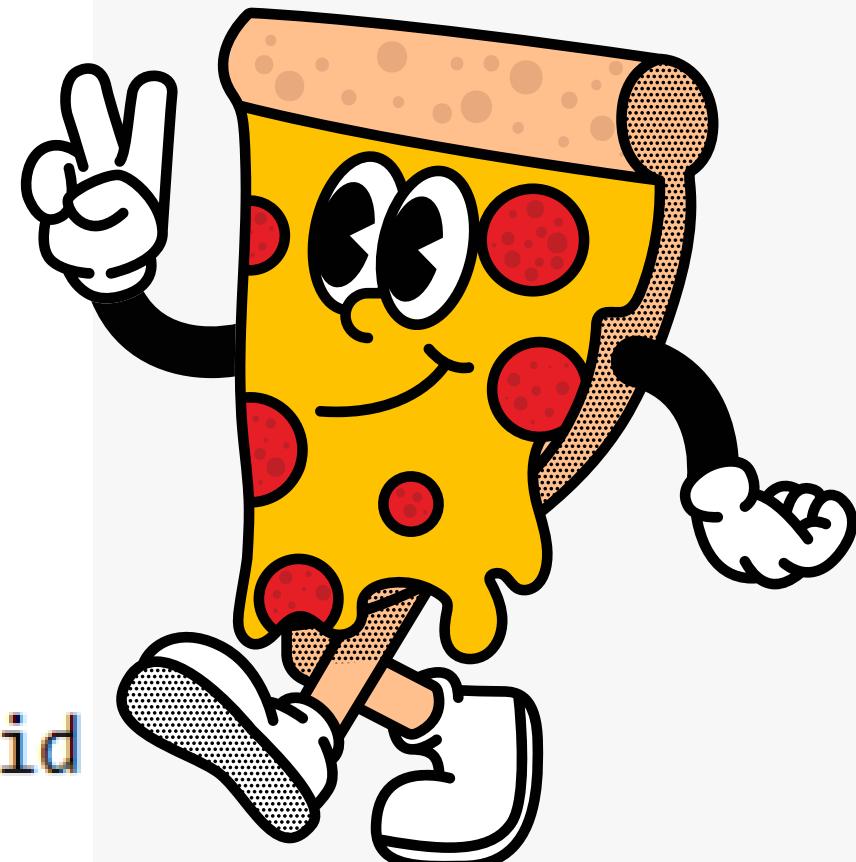


Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	name	price
▶	The Greek Pizza	35.95

identify THE MOST COMMON PIZZA SIZE ORDERED

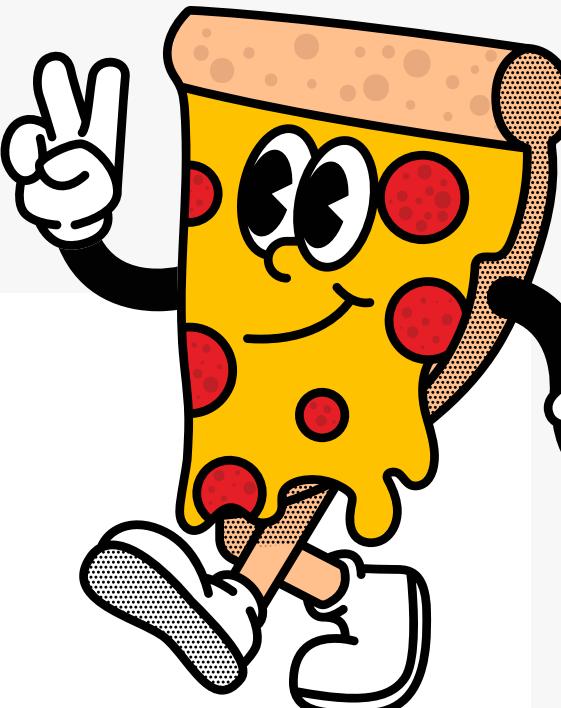
```
3 •   SELECT
4       pizzas.size,
5       COUNT(order_details.order_details_id) AS order_count
6   FROM
7       pizzas
8       JOIN
9       order_details ON pizzas.pizza_id = order_details.pizza_id
10  GROUP BY pizzas.size
11  ORDER BY order count DESC;
```



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

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

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



```
3 • SELECT
4     ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
5 FROM
6     (SELECT
7         orders.order_date, SUM(order_details.quantity) AS quantity
8     FROM
9         orders
10    JOIN order_details ON orders.order_id = order_details.order_id
11    GROUP BY orders.order_date) AS ordeer_quantity;
```

determine the top 3 most ordered pizza types based on revenue

```
3 • SELECT
4     pizza_types.name,
5         SUM(order_details.quantity * pizzas.price) AS revenue
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

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

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



analyze the cumulative revenue generated over time

```
3 • select order_date,  
4   sum(revenue) over(order by order_date) as cum_revenue  
5   from  
6   (select orders.order_date,  
7     sum(order_details.quantity * pizzas.price) as revenue  
8     from order_details join pizzas  
9     on order_details.pizza_id = pizzas.pizza_id  
10    join orders  
11    on orders.order_id = order_details.order_id  
12    group by orders.order_date)as sales;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
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			



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.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;
```



	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

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_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

list the top 5 most ordered pizza types along with their quantites

```
3  SELECT
4      pizza_types.name, 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
0      order_details ON order_details.pizza_id = pizzas.pizza_id
1  GROUP BY pizza_types.name
2  ORDER BY quantity DESC
3  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