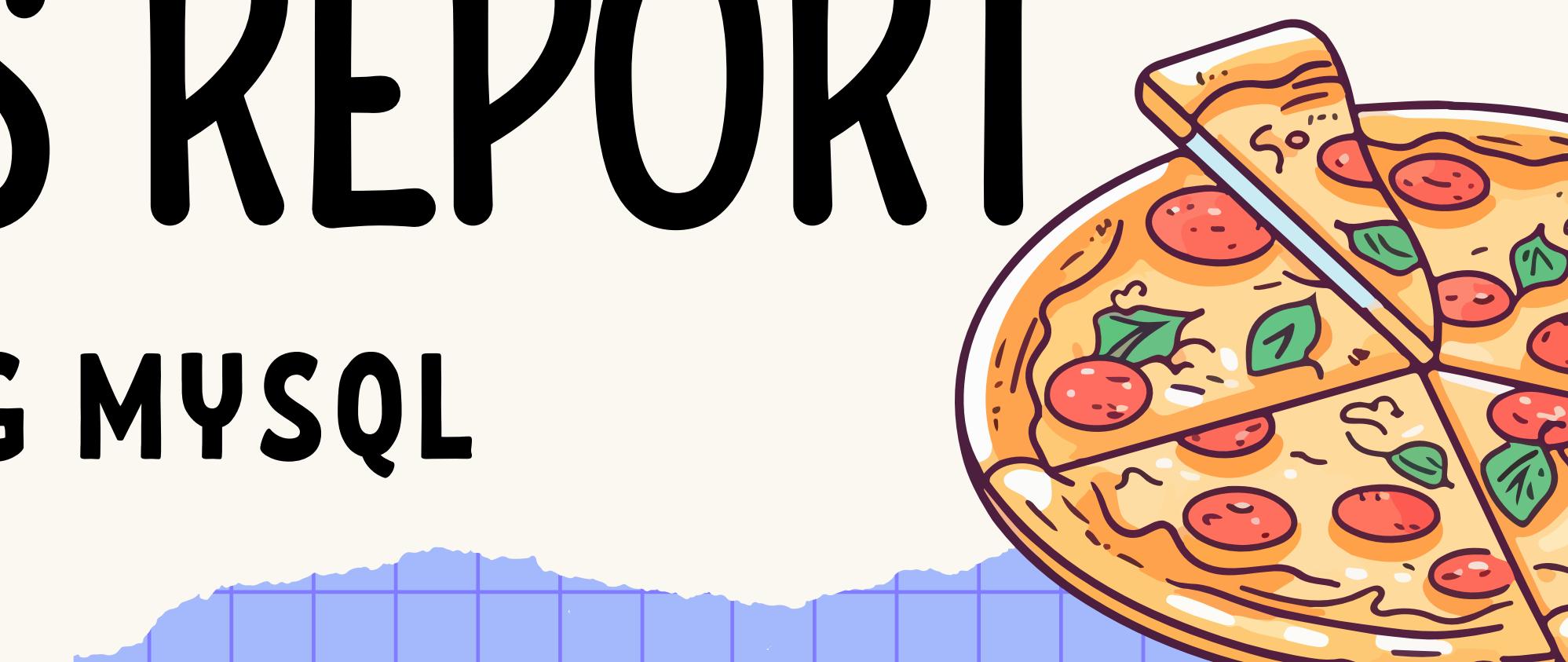
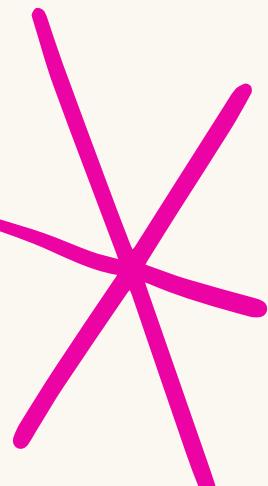
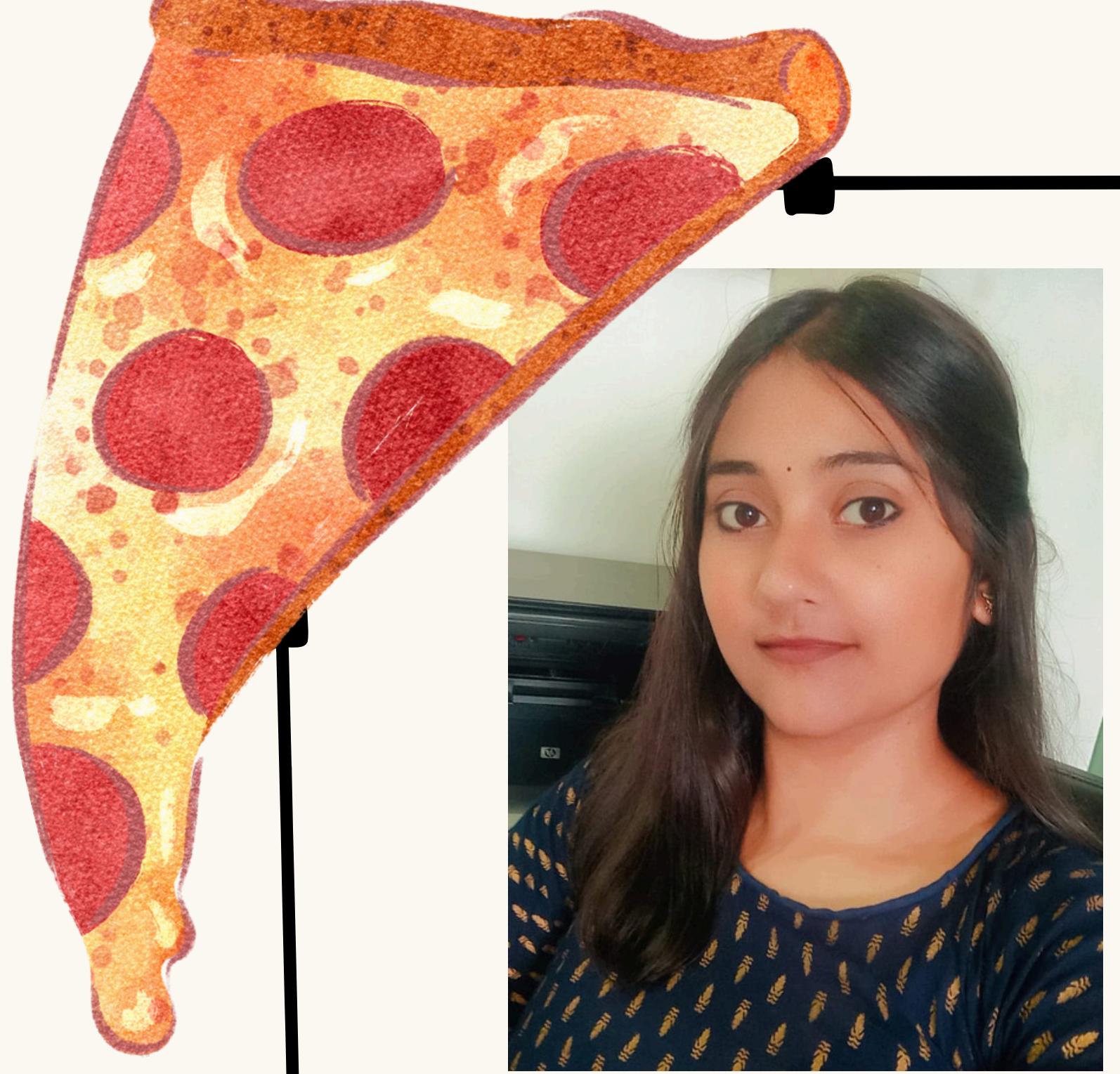


By Varsha Saxena

My_pizza SALES REPORT USING MYSQL

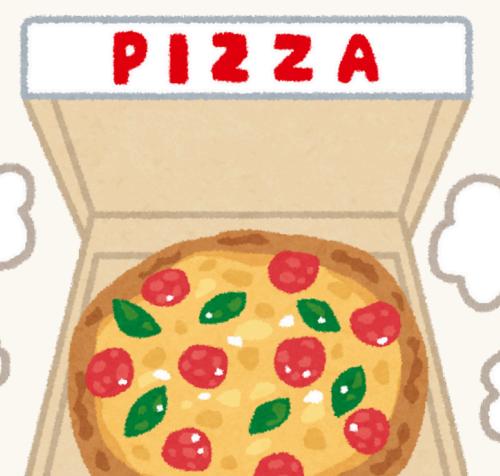
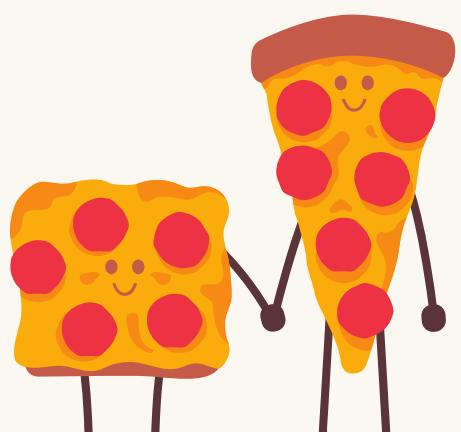
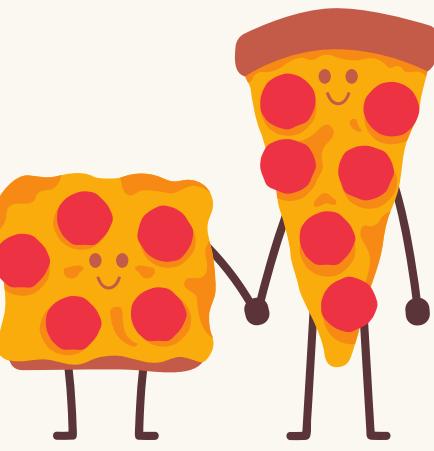




Hi there! I'm Varsha!

Introduction

Welcome to my presentation on SQL project...
I'm Varsha Saxena , currently enhancing my skills in SQL , a vital tool for data analytics . In this project , I have analyzed Pizza Sales data to uncover valuable insights and trends . Through this project , I aim to demonstrate how MySQL can be used to manage , query , and analyze data effectively , contributing to data-driven decision - making processes.



TABLES

TABLE 1

Table: order_details

Columns:

<u>order_details_id</u>	int PK
order_id	int
pizza_id	text
quantity	int

TABLE 3

Table: pizza_types

Columns:

<u>pizza_type_id</u>	text
name	text
category	text
ingredients	text

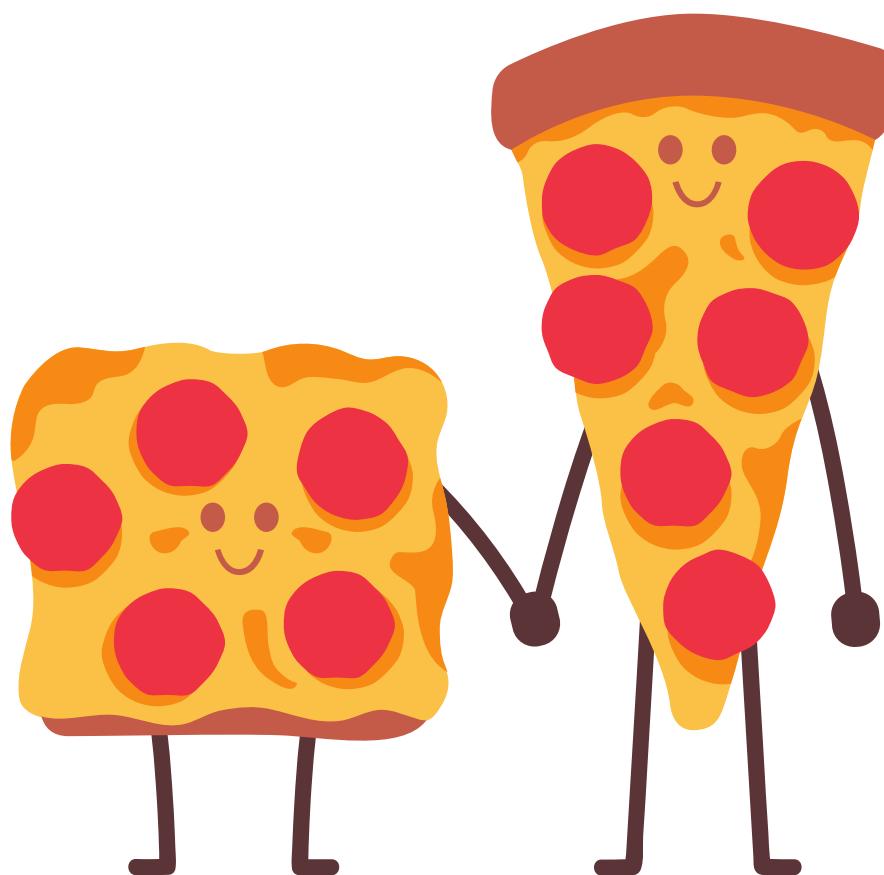


TABLE 2

Table: orders

Columns:

<u>order_id</u>	int PK
order_date	date
order_time	time

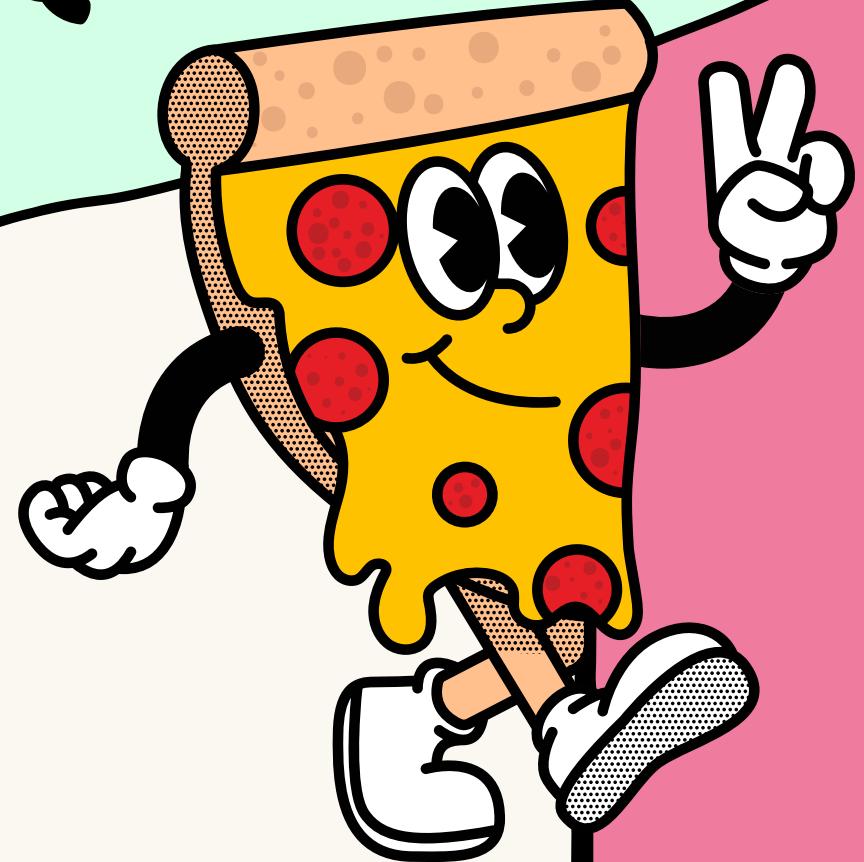
TABLE 4

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

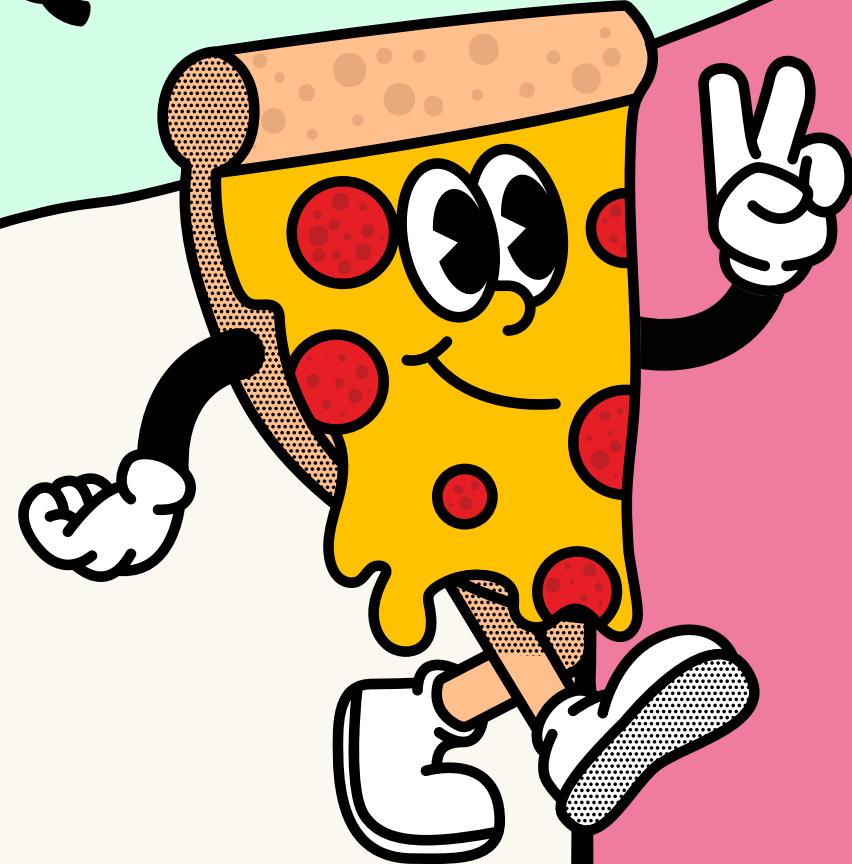
QUERIES



1. Total orders placed.
2. Total revenue.
3. Highest priced pizza.
4. Most common pizza size ordered.
5. Top 5 most ordered pizza types along with their quantities.
6. Total quantity of each pizza category ordered.



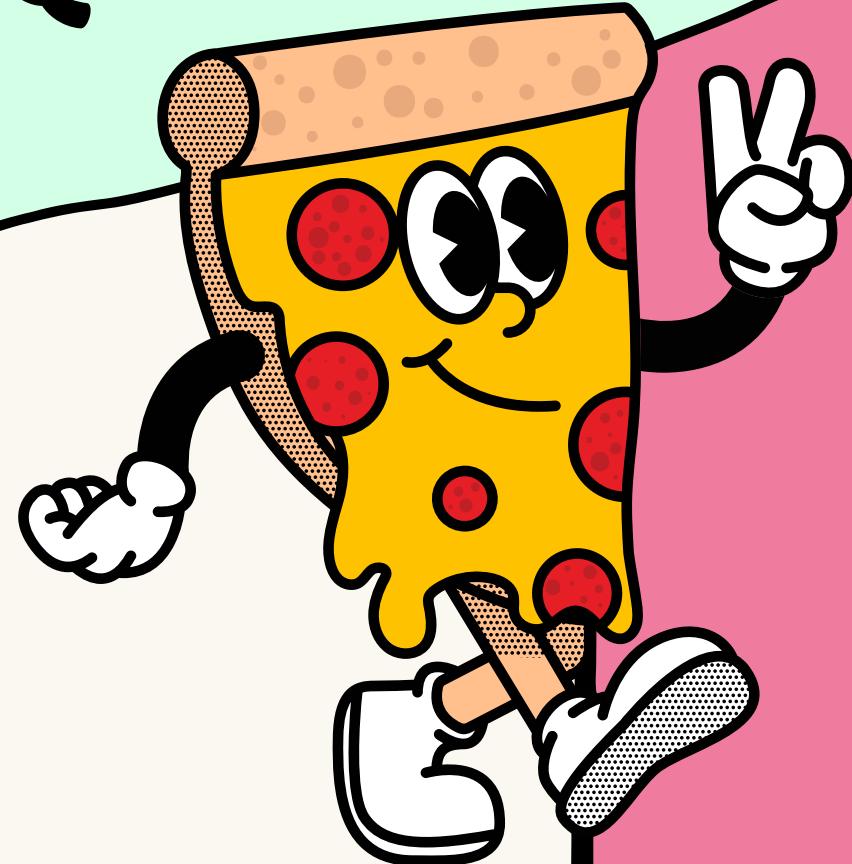
QUERIES



7. Distribution of orders by hour of the day.
8. Category-wise distribution of pizzas.
9. Group the orders by date and calculate the average number of pizzas ordered per day.
10. top 3 most ordered pizza types based on revenue.



QUERIES



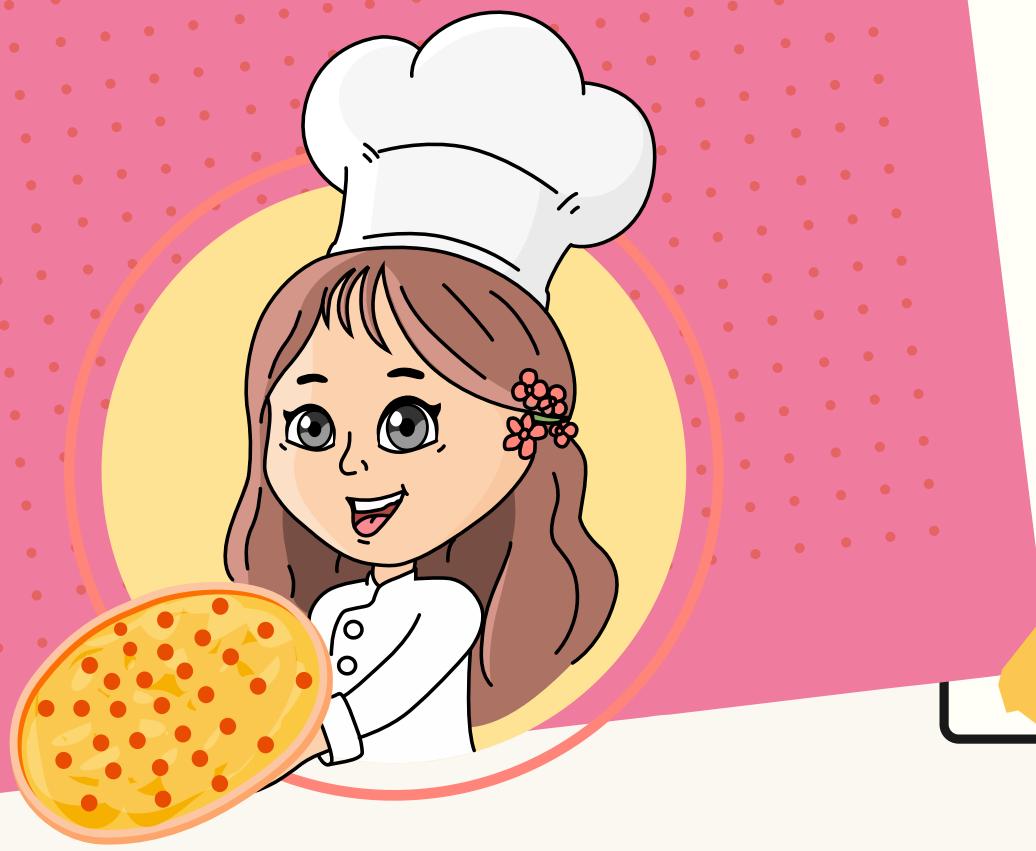
11. Percentage contribution of each pizza type to total revenue.

12. Cumulative revenue generated over time.

13. Top 3 most ordered pizza types based on revenue for each pizza category.



Retrieve the total number of orders placed.

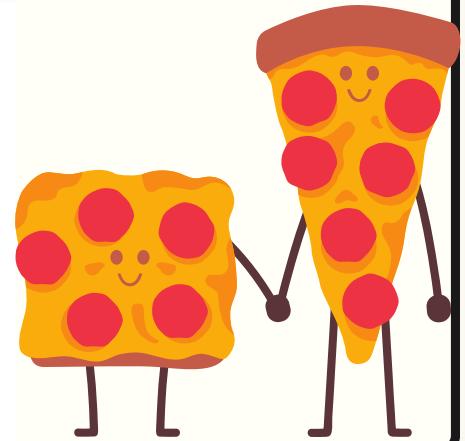


QUERY

```
select count(order_id) as total_orders from orders;
```

OUTPUT

Result Grid	
	total_orders
▶	21350



Calculate the
total revenue
generated
from pizza
sales.



QUERY

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
```

OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	
total_revenue		817860.05			

6

Identify the highest-priced pizza.



QUERY

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

OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	name	price			
▶	The Greek Pizza	35.95			

Identify the most common pizza size ordered.



QUERY

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

OUTPUT

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

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



SELECT

```
pizza_types.name,  
SUM(order_details.quantity) AS ordered_pizzas  
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 ordered_pizzas DESC  
LIMIT 5;
```

QUERY

OUTPUT

	name	ordered_pizzas
▶	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..



SELECT

QUERY

```
    pizza_types.category,  
    SUM(order_details.quantity) AS pizzas_category  
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 pizzas category DESC;
```

OUTPUT

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

	category	pizzas_category
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.



SELECT

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

FROM

orders

GROUP BY HOUR(order_time)

ORDER BY total_orders DESC;

QUERY

OUTPUT



	hours	total_orders
▶	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

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



QUERY

SELECT

category, COUNT(name) AS total_pizzas

FROM

pizza_types

GROUP BY category;

OUTPUT

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

	category	total_pizzas
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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



QUERY

```
SELECT  
    ROUND(AVG(total_quantity), 0) AS pizzas_ordered_per_day  
FROM  
    (SELECT  
        orders.order_date,  
        SUM(order_details.quantity) AS total_quantity  
    FROM  
        orders  
    JOIN order_details ON orders.order_id = order_details.order_id  
    GROUP BY orders.order_date) AS order_quantity;
```

OUTPUT

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

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



SELECT

```
pizza_types.name,  
SUM(order_details.quantity * pizzas.price) AS total_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 total_revenue DESC  
LIMIT 3;
```

QUERY

OUTPUT

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
name	total_revenue				
The Thai Chicken Pizza	43434.25				
The Barbecue Chicken Pizza	42768				
The California Chicken Pizza	41409.5				

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



QUERY

```
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 total_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 total_revenue DESC;
```

OUTPUT

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

category	total_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



Analyze the
cumulative
revenue
generated over
time.



QUERY

```
select order_date,  
       sum( total_revenue ) over(order by order_date ) as cum_revenue  
  from  
(select orders.order_date,  
           sum( order_details.quantity * pizzas.price ) as total_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 ;
```

OUTPUT

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

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

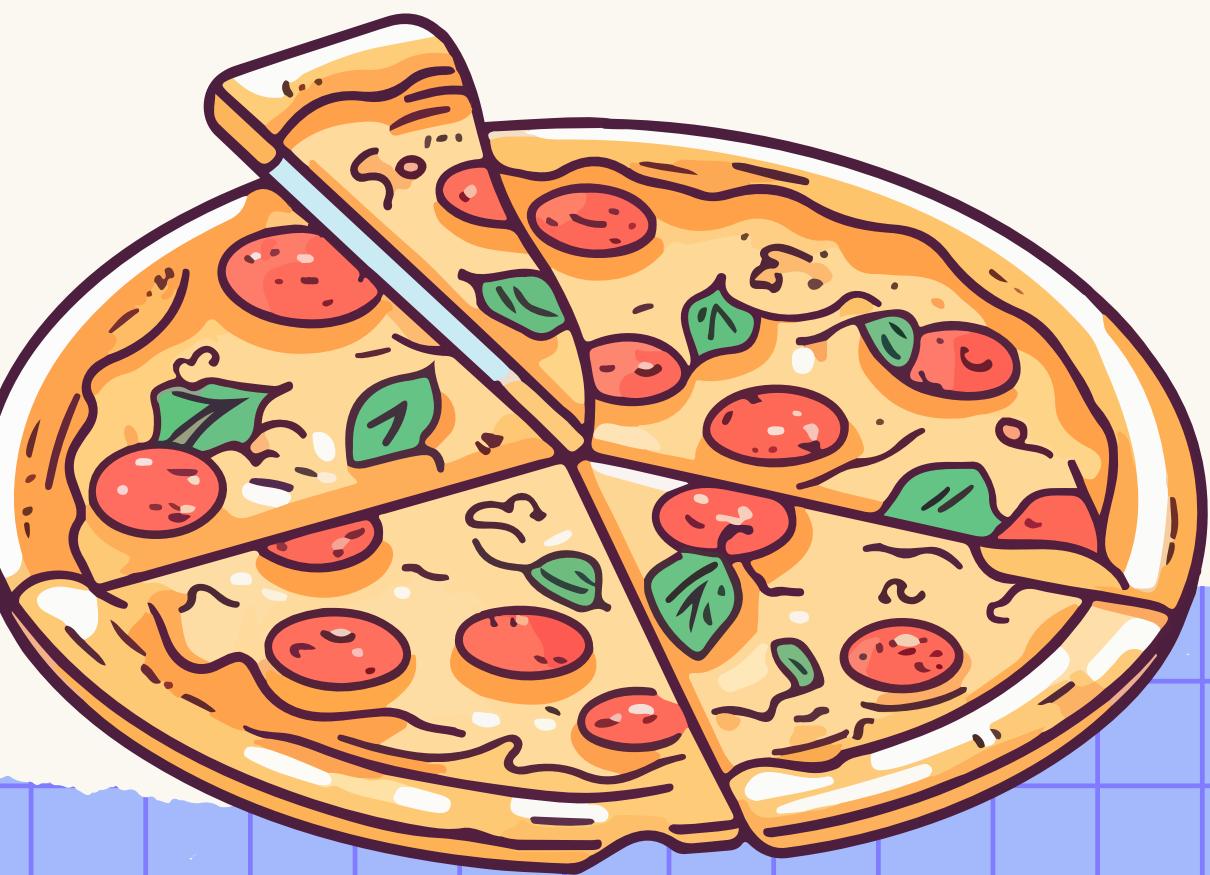
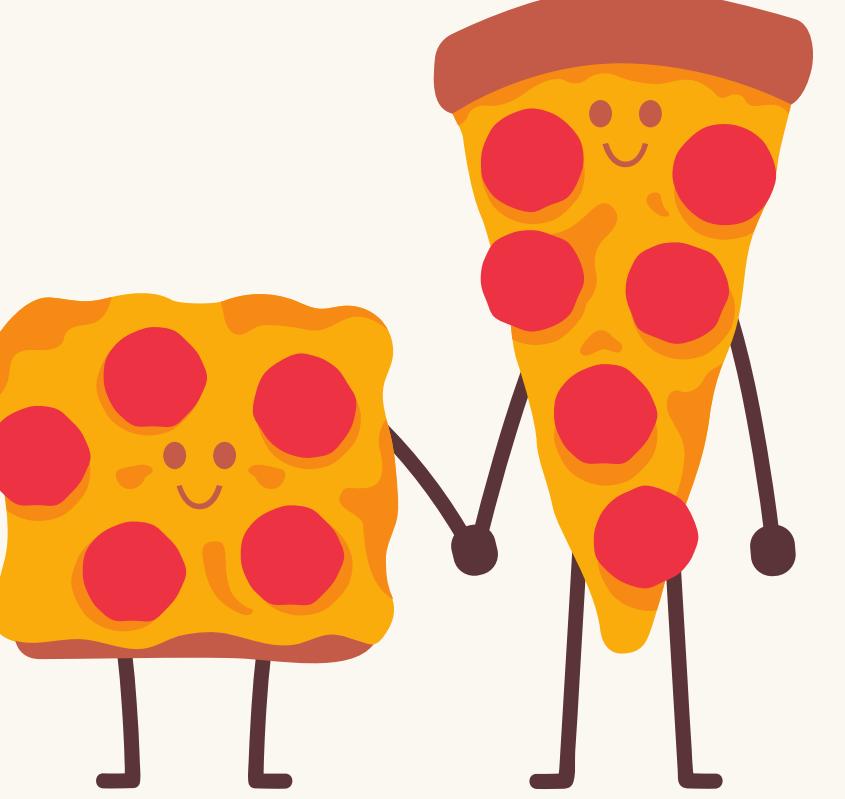
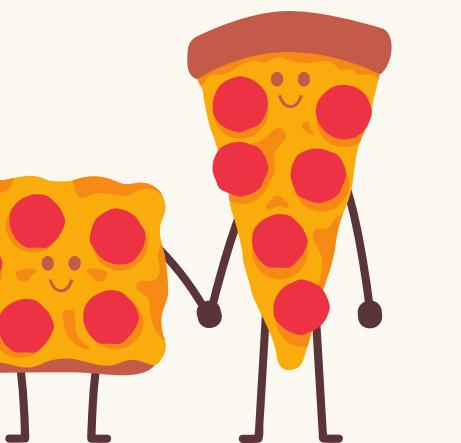
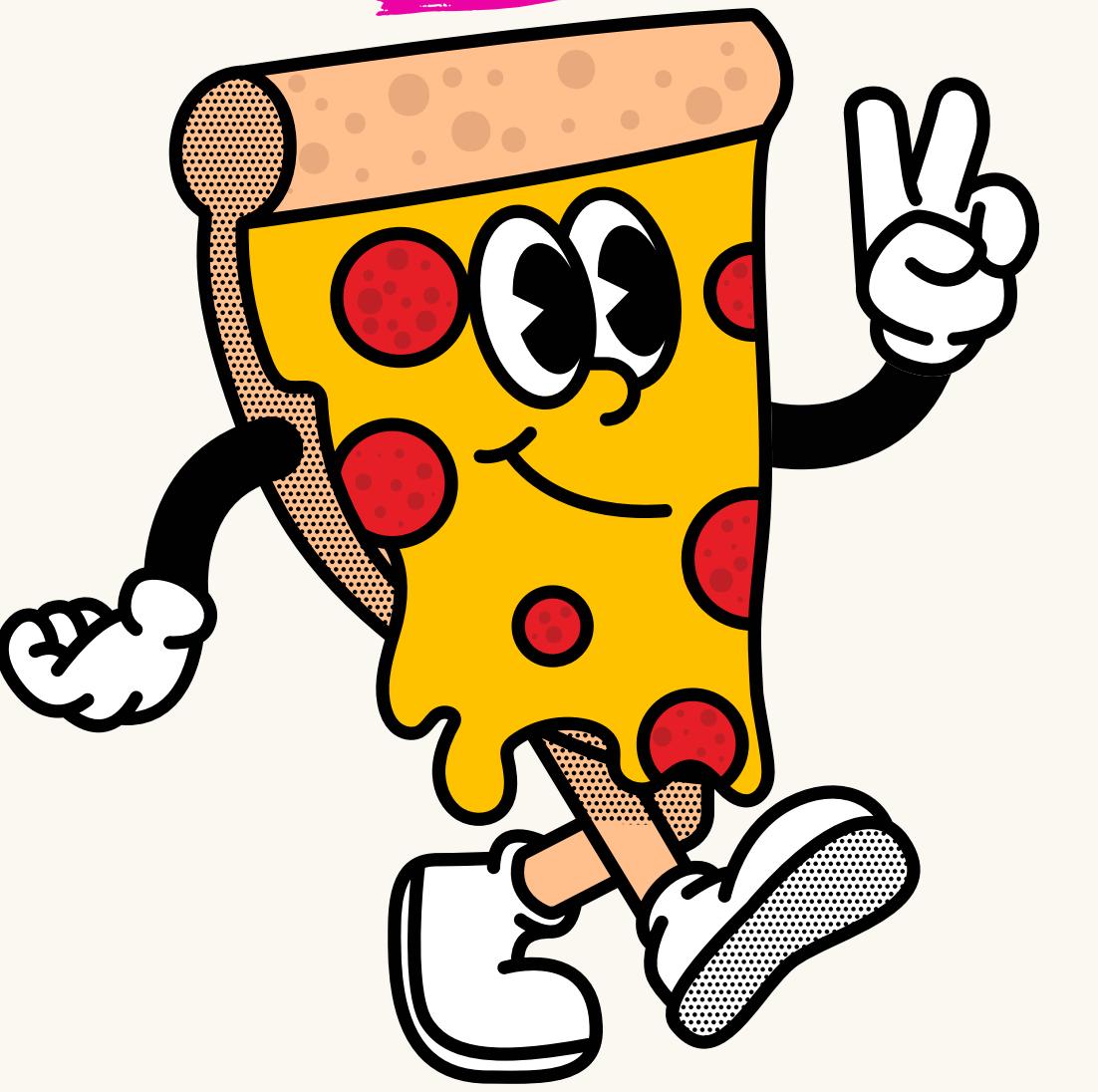


QUERY

```
select name , total_revenue from
(select category , name , total_revenue,
rank() over(partition by category order by total_revenue desc) as rankk
from
(select pizza_types.category , pizza_types.name ,
sum((order_details.quantity) * pizzas.price ) as total_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 ) b
where rankk <= 3 ;
```

OUTPUT

Result Grid		Filter Rows:
	name	total_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
	The Five Cheese Pizza	26066.5



THANK YOU!

By Varsha Saxena