



# SQL Project on Pizza Sales

# Overview



Hello, I am Devesh Mishra, and I am creating a SQL project to analyze pizza sales data, covering topics from basic to advanced SQL techniques. This project involves setting up a database with tables for customers, orders, pizzas, order details, and stores. It includes performing CRUD operations, using SELECT queries, filtering, sorting, and aggregating data. Advanced techniques such as subqueries, window functions, CTEs, indexing, stored procedures, and triggers are also employed. The goal is to derive business insights like customer preferences, sales trends, and popular products, and to generate comprehensive sales reports and real-time dashboards for effective decision-making.

# Questions:-

#BASIC:-

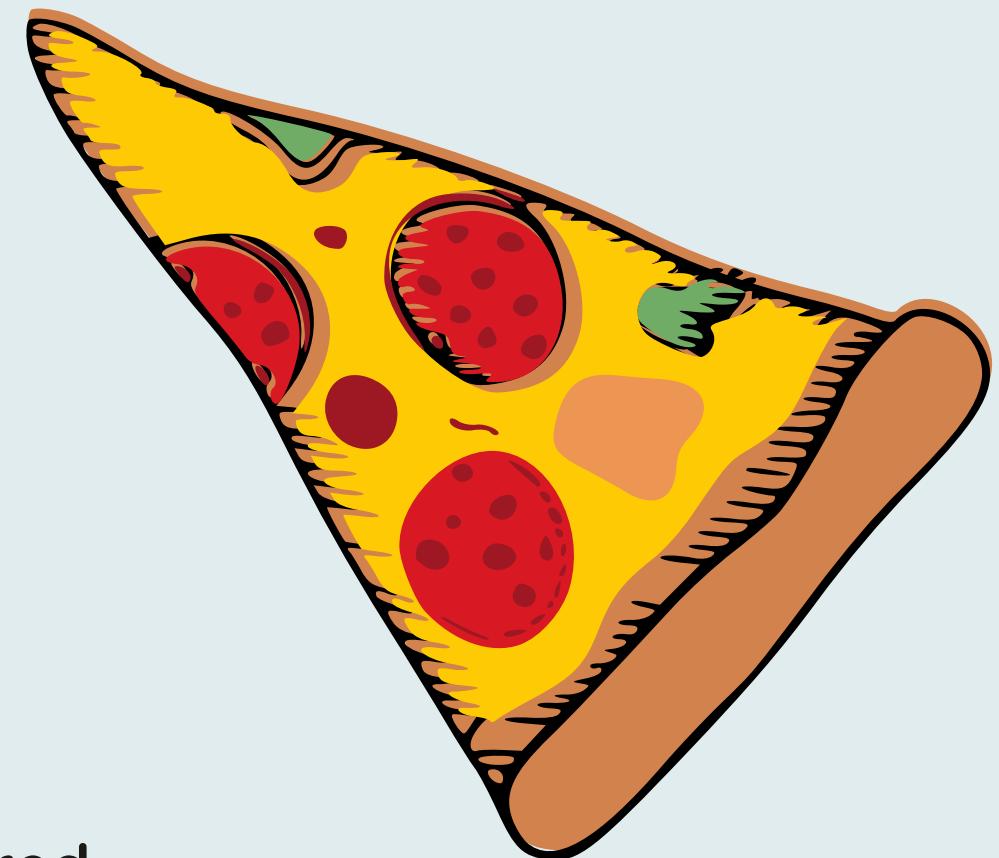
#Q-Retrieve the total number of orders placed.

#Q-Calculate the total revenue generated from pizza sales.

#Q-Identify the highest-priced pizza.

#Q-Identify the most common pizza size ordered.

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



#Intermediate:

#Q-Join the necessary tables to find the total quantity of each pizza category ordered.

#Q-Determine the distribution of orders by hour of the day.

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

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

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

#Advanced:

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

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

Q-Retrieve the total number of orders placed.

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

Result Grid	
	total_orders
▶	21350



# Q-Calculate the total revenue generated from pizza sales.

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;
```

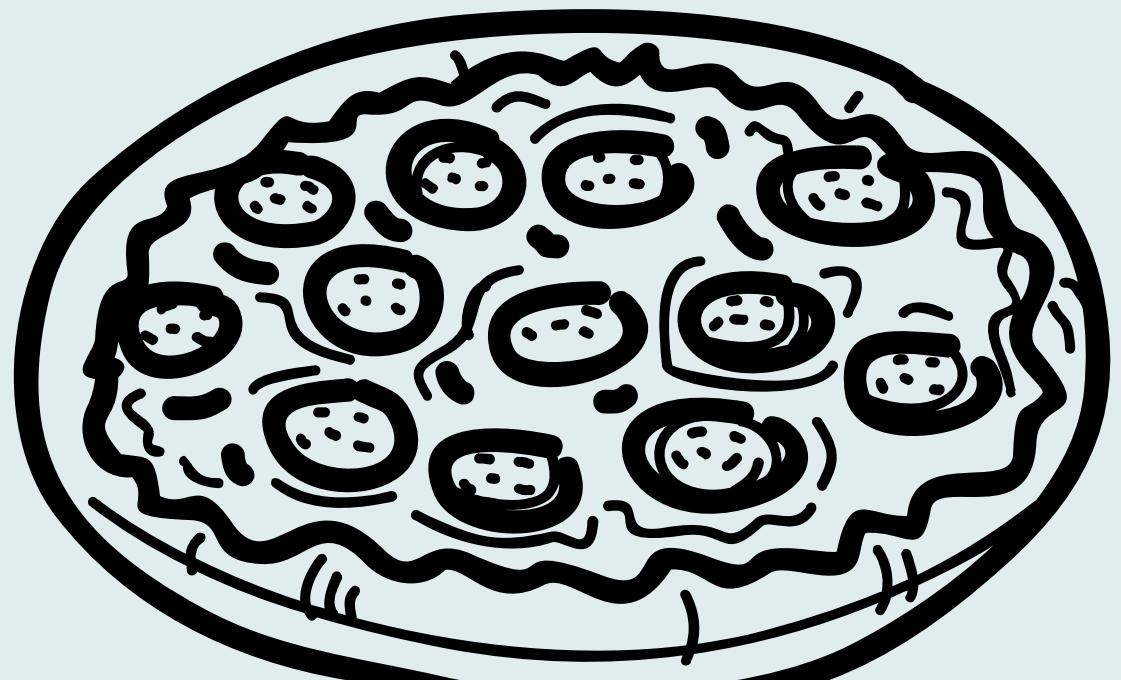
	total_sales
▶	817860.05



# Q-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;
```

	name	price
▶	The Greek Pizza	35.95



Q-Identify the most common pizza size ordered.

```
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
LIMIT 1;
```

Result Grid | Filter

	size	order_count
▶	L	18526

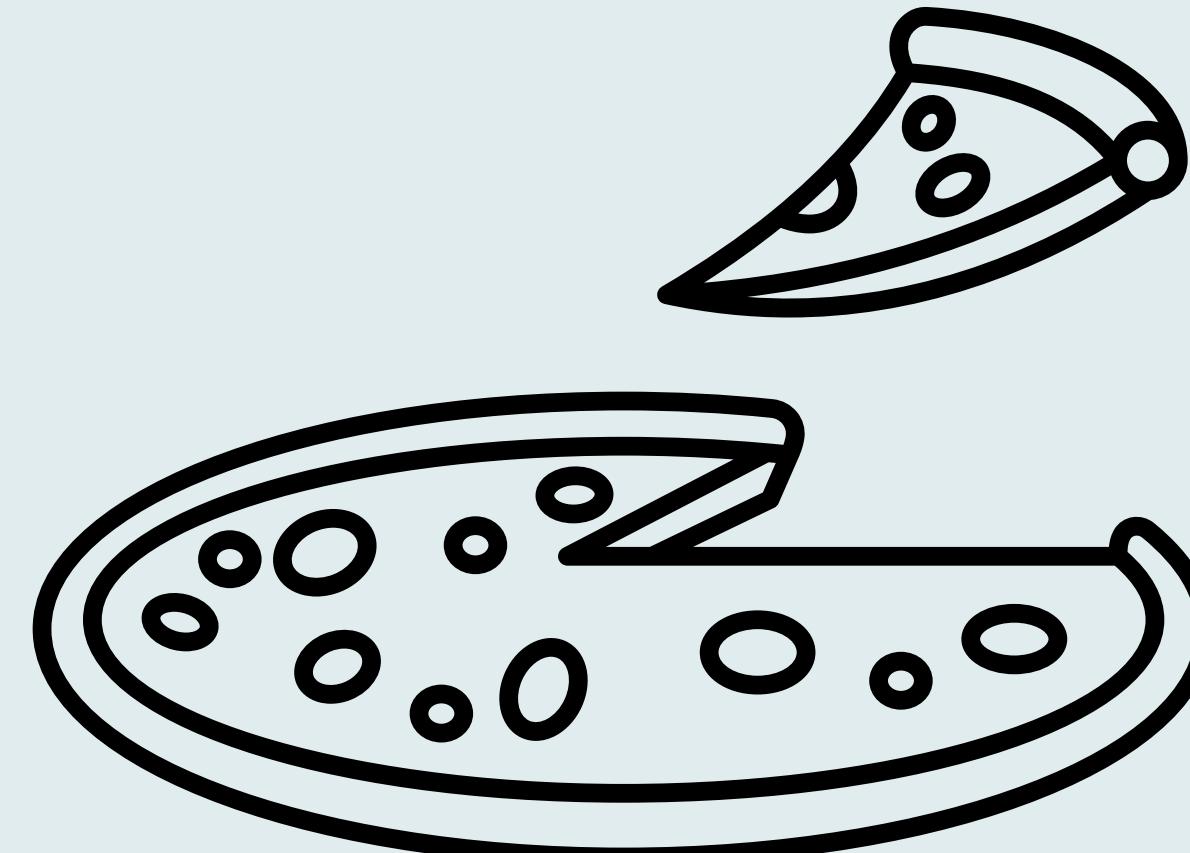


Q-List the top 5 most ordered pizza types along with their quantities.

```
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;
```

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



Q-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 quantity DESC;
```

Result Grid | Filter R

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



Q-Determine the distribution of orders by hour of the day.

SELECT

HOUR(order\_time) AS hour, COUNT(order\_id) AS order\_count

FROM

orders

GROUP BY HOUR(order\_time);

Result Grid |

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468



Q-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



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

SELECT

ROUND(AVG(quantity), 0) as avg\_pizza\_order\_per\_day

FROM

(SELECT

oders.order\_date, SUM(order\_details.quantity) AS quantity

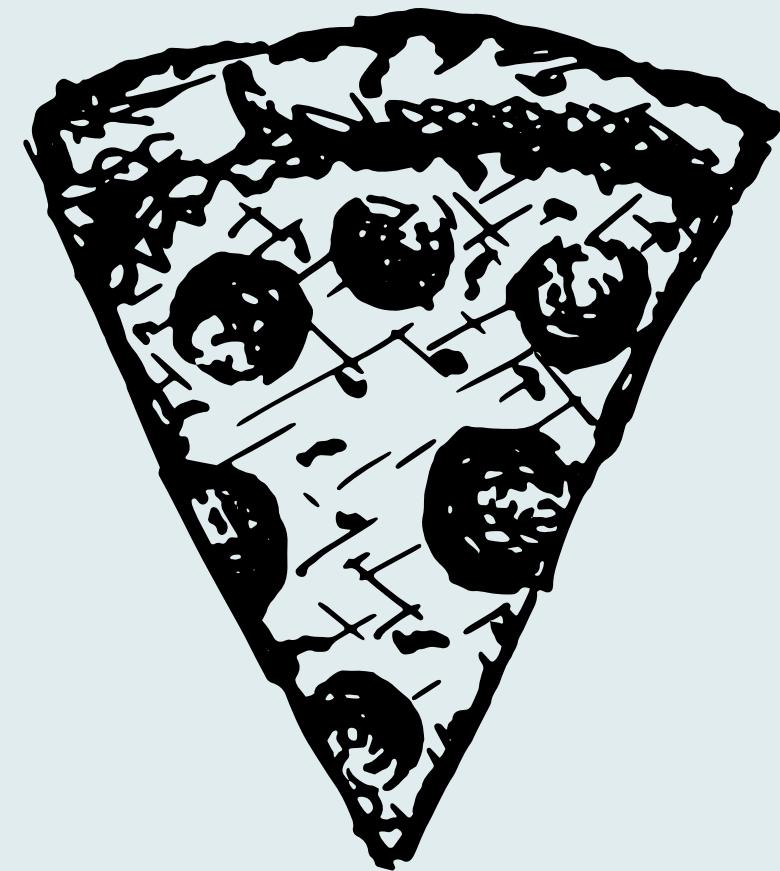
FROM

oders

JOIN order\_details ON oders.order\_id = order\_details.order\_id

GROUP BY oders.order\_date) AS order\_quantity;

	Result Grid			Filter Rows:
	avg_pizza_order_per_day			
▶	138			



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

SELECT

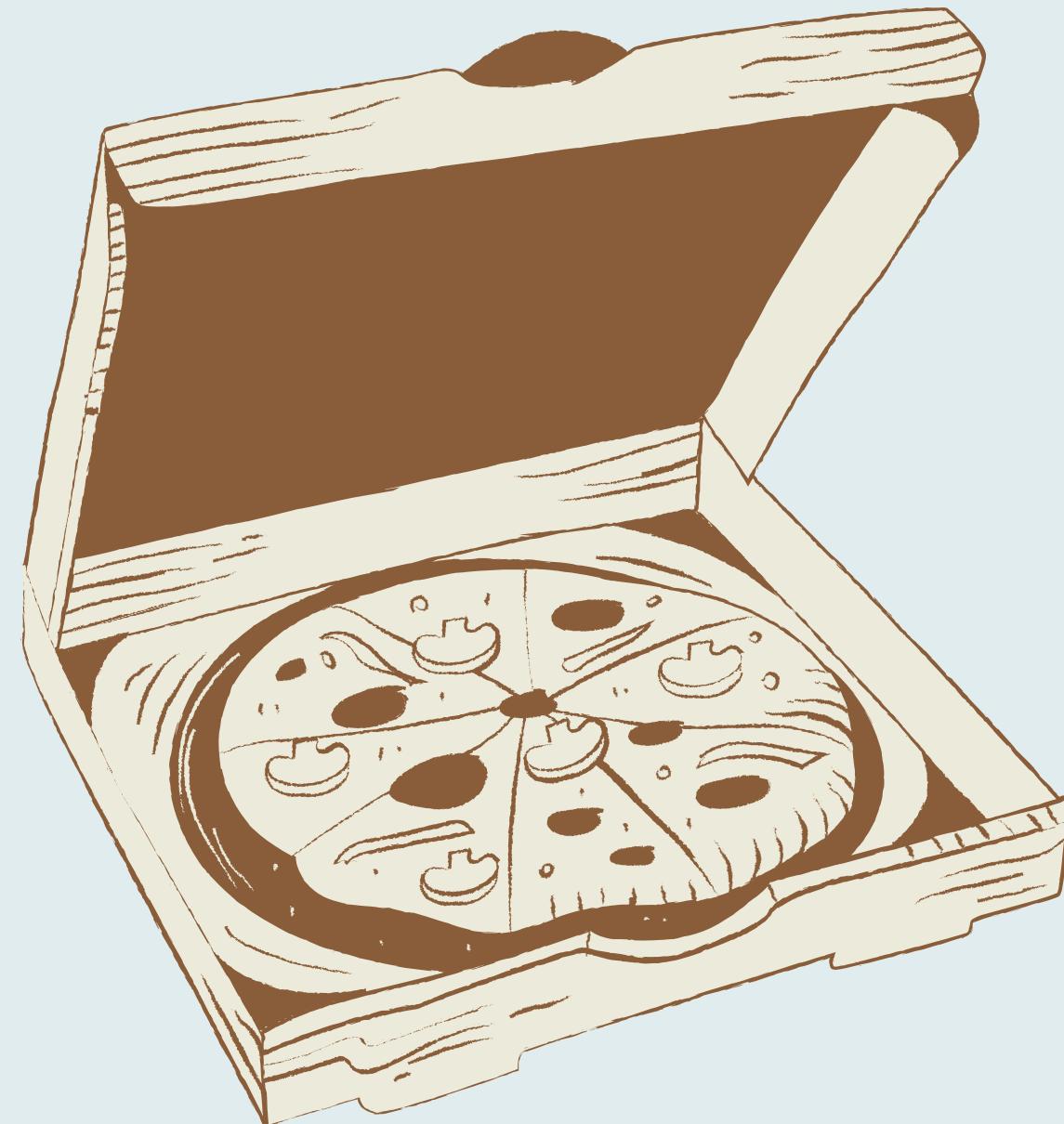
```
    pizza_types.name,  
    SUM(order_details.quantity * pizzas.price) AS revenue
```

FROM

```
    pizza_types  
        JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.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



Q-Calculate the percentage contribution of each 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;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q-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 rr  
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;
```

Result Grid		Filter Rows:	Exp
	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	

*Thank you*