

# SQL-Project

**Pizza\_sales**





Hello everyone, my name is Ajay Singh Rana. I am passionate about data analysis and database management. Recently, I completed a comprehensive SQL project focused on analyzing pizza sales. This project involved working with four key tables:

Order\_details: Contains detailed information about each order.

Orders: Includes general information about each order.

Pizza\_types: Lists the different types of pizzas available.

Pizzas: Contains specific details about each pizza.

Through this project, I gained valuable insights into database design, data manipulation, and SQL querying techniques. I am excited to share my findings and the skills I have developed during this project.

# Questions

## Basic level

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

## Intermediate

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

## Advance level

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

## Retrive the total number of orders placed

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

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
total_orders					
21350					

## Calculate the total revenue generated from pizza sale






```
2 ● SELECT
3   ROUND(SUM(order_details.quantity * pizzas.price),
4         2) AS total_sales
5 FROM
6   order_details
7   JOIN
8   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

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

Result Grid			Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 	Fetch rows: 
name	price					
The Greek Pizza	35.95					

## Identify the most common pizza size ordered

```
select pizzas.size, count(order_details.order_details_id) as order_count
from
order_details
join
pizzas on order_details.pizza_id = pizzas.pizza_id
group by pizzas.size order by order_count desc
limit 1;
```

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

	order_count
	18526

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

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

Result Grid



Filter Rows:

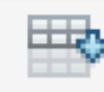
Export:



Wrap Cell Content:



Fetch 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



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

```
2  ●  SELECT
3      pizza_types.category,
4      SUM(order_details.quantity) AS quantity
5  FROM
6      pizzas
7      JOIN
8      pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9      JOIN
10     order_details ON order_details.pizza_id = pizzas.pizza_id
11  GROUP BY pizza_types.category
12  ORDER BY quantity DESC;
```

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

```
2 • SELECT
3     HOUR(order_time) AS hour, COUNT(order_id) AS order_count
4 FROM
5     orders
6 GROUP BY HOUR(order_time);
```

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

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

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

```
2 • Select category, count(name) from pizza_types
3 group by category;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



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

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

```
2
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 order_quantity;
```

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

avg_pizza_ordered_per_day
138



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

```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2  • select pizza_types.name,
3     sum(order_details.quantity * pizzas.price) as revenue
4  from
5     pizza_types join pizzas
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id
7     join order_details
8     on pizzas.pizza_id = order_details.pizza_id
9     group by pizza_types.name order by revenue desc 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			

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

```
2 ● select pizza_types.category,  
3 round(sum(order_details.quantity * pizzas.price) /  
4 (select round(sum(order_details.quantity * pizzas.price),2) as total_sales  
5 from  
6 order_details  
7 join pizzas on pizzas.pizza_id = order_details.pizza_id)*100,2) as revenue  
8 from pizza_types join pizzas  
9 on pizza_types.pizza_type_id = pizzas.pizza_type_id  
10 join order_details  
11 on order_details.pizza_id = pizzas.pizza_id  
12 group by pizza_types.category order by revenue desc;  
13
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	category	revenue
•	Classic	26.91
	Supreme	25.46
	Chicken	23.96

## Analyze the cumulative revenue generated over time

```
2 ● select order_date,
3     sum(revenue) over (order by order_date) as cumulative_revenue
4     from
5     (select orders.order_date,
6      sum(order_details.quantity * pizzas.price) as revenue
7      from order_details join pizzas
8      on order_details.pizza_id =pizzas.pizza_id
9      join
10     orders on
11     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	cumulative_revenue
2015-01-01	2713.85000000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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

```
2 • select pizza_types.name,  
3      sum(order_details.quantity * pizzas.price) as revenue  
4      from  
5      pizza_types join pizzas  
6      on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7      join order_details  
8      on pizzas.pizza_id = order_details.pizza_id  
9      group by pizza_types.name order by revenue desc 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



Thanking you





