



SHODWE

Pizza Sales Report



SQL PROJECT ON PIZZA SALES

- WHERE EVERY QUERY TELLS A STORY

Home

About

Contact





SHODWE

Pizza Sales Report

[Home](#)[About](#)[Contact](#)

ABOUT US

MY NAME IS SAHIL CHAVAN

This MySQL project analyzes pizza sales data to generate key business insights through SQL queries. It covers basic, intermediate, and advanced levels of analysis — from finding total orders, revenue, highest-priced pizzas, and most popular sizes to performing category-wise analysis, time-based order distribution, and revenue comparisons. Advanced queries calculate each pizza type's revenue contribution, cumulative sales trends, and top-performing pizzas by category. The project demonstrates practical skills in database design, joins, aggregations, and data-driven decision-making using MySQL.



Retrieve the total number of orders placed

```
3 •   SELECT  
4       COUNT(order_id) AS total_orders  
5  
6   FROM  
    orders;
```

Result Grid	
	total_orders
▶	21350



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 Row

	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;
```

	size	order_count
▶	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 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



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 |

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



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

```
• SELECT
    pizza_types.category, COUNT(pizza_types.name)
FROM
    pizza_types
GROUP BY pizza_types.category;
```

Result Grid | Filter Rows:

	category	COUNT(pizza_types.name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



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

```
SELECT
    round(AVG(quantity),0)
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;
```

Result Grid		Filter R
round(AVG(quantity),0)		
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 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



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

```
* SELECT
    pizza_types.category,
    round((SUM(pizzas.price * order_details.quantity) / 817860.049999993) * 100,2) AS percentage
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;
```

	category	percentage
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96



Analyze the cumulative revenue generated over time.

```
• select order_date,  
sum(Sales) over(order by order_date) as cum_sales  
from  
(select orders.order_date, sum(order_details.quantity * pizzas.price) as Sales  
from order_details join orders  
on order_details.order_id = orders.order_id  
join pizzas  
on pizzas.pizza_id = order_details.pizza_id  
group by orders.order_date) as Sales;
```

	order_date	cum_sales
▶	2015-01-01	2713.850000000004
	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



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

```
select category, name, revenue, rn
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 < 4;
```

	category	name	revenue	rn
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2



SHODWE

Pizza Sales Report

[Home](#)

[About](#)

[Contact](#)

THANK YOU