# Pizza Sales Report using SQL



Raj Bhadwaj

May 2024

#### Introduction

In today's data-driven world, the ability to extract actionable insights from vast amounts of information is paramount. Data analytics, the systematic exploration and analysis of data sets to uncover patterns, trends, and correlations, plays a pivotal role in informing decision-making processes across various industries and domains.

At its core, data analytics encompasses a spectrum of techniques and methodologies, ranging from descriptive analytics, which focuses on summarizing and visualizing data, to predictive and prescriptive analytics, which aim to forecast future outcomes and recommend optimal courses of action.

This brief introduction provides a glimpse into the transformative power of data analytics, offering a glimpse into its potential to drive innovation, enhance efficiency, and unlock new opportunities in an ever-evolving landscape of information.



## About Project

Through a combination of data analysis and market research, this report sheds light on the factors influencing pizza sales, from regional preferences to seasonal variations. By examining key metrics such as sales volume, revenue trends, and consumer demographics, we aim to provide a comprehensive overview of the pizza market and its evolution over time.



### Project Problems with their Solutions



#### Problem 1

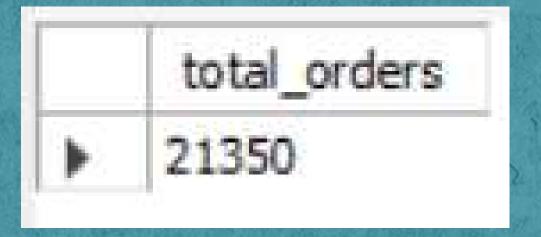
Retrieve the total number of orders placed.

```
SELECT

COUNT(order_id) AS total_orders

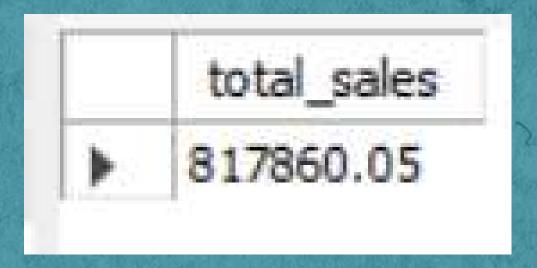
FROM

orders;
```



#### Problem 2

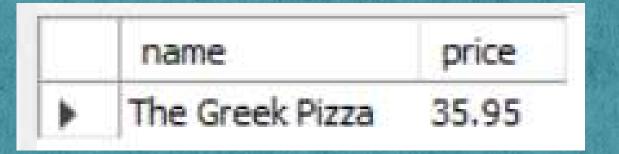
Calculate the total revenue generated from pizza sales.



#### Problem 3

Identify the highestpriced 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;
```



#### Problem 4

Identify the most common pizza size ordered.

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

#### Problem 5

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

```
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

#### Problem 1

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;
```

	category	quantity
<b>&gt;</b>	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

#### Problem 2

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);
```

	hour	order_count
<b>)</b>	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

#### Problem 3

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

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

	category	COUNT(name)
<b>&gt;</b>	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

#### Problem 4

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

```
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
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;
```

```
avg_pizza_ordered_per_day

138
```

#### Problem 5

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;
```

	name	revenue
<b>•</b>	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

#### Advanced Problems

#### Problem 1

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

	category	revenue
>	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```
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 pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
   order details ON order details.pizza id = pizzas.pizza id
GROUP BY pizza types.category
ORDER BY revenue DESC;
```

#### Advanced Problems

#### Problem 2

Analyze the cumulative revenue generated over time.

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

, Ç.	PROBLEM STREET	HERE WAS ALS MAN
	order_date	cum_revenue
١	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
	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
	2015-01-13	29831.300000000003
	2015-01-14	32358.700000000004
	2015-01-15	34343.50000000001
	2015-01-16	36937.65000000001
	2015-01-17	39001.75000000001
	2015-01-18	40978.600000000006
	2015-01-19	43365.75000000001
	2015-01-20	45763.65000000001
	2015-01-21	47804.20000000001
	2015-01-22	50300.90000000001
	2015-01-23	52724.6000000000006
	2015-01-24	55013.8500000000006
	2015-01-25	56631.40000000001
	2015-01-26	58515.80000000001
	2015-01-27	61043.85000000001

#### Advanced Problems

#### Problem 3

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

	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

```
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;
```

#### Contact

PizzaSales





## Thank you!



Raj Bhardwaj 2024