

SQL Project on Pizza Sales







About this Project

- Analyzed Total orders, Revenue, and identified Top-selling Pizzas using SQL.
- Conducted in-depth analysis of order patterns, including pizza size Popularity and Revenue contribution.
- Showcased advanced SQL techniques, including joins and aggregation, to extract actionable Business Insights.



Retrieve the Total number of Orders placed. . .

```
1 • SELECT
2     COUNT(order_id)
3 FROM
4 orders;
```




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

	count(order_id)
▶	21350

Your paragraph text

Calculate the Total Revenue generated from Pizza Sales.

```
1  #Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4  ROUND(SUM(order_details.quantity * pizzas.price),
5        2) as total_revenue
6  FROM
7      order_details
8      JOIN
9      pizzas ON pizzas.pizza_id = order_details.pizza_id
10
```

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

	total_revenue
▶	817860.05

Your paragraph text

Identify the Highest-Priced Pizza.



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

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

	name	price
▶	The Greek Pizza	35.95

Your paragraph text

Identify the Most common Pizza size ordered.

1

•

SELECT

2

pizzas.size,

3

COUNT(order_details.order_details_id) AS order_count

4

FROM

5

pizzas

6

JOIN

7

order_details ON pizzas.pizza_id = order_details.pizza_id

8

GROUP BY pizzas.size

9

ORDER BY order_count DESC;

10

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

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

Your paragraph text

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

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

Your paragraph text

Join the necessary tables to find the Total Quantity of each Pizza Category ordered. ...

```
1 • SELECT
2     pizza_types.category,
3     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 pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizza_types.category
11 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.

1 • SELECT

2 HOUR(time), COUNT(order_id)

3 FROM

4 orders

5 GROUP BY HOUR(time);

Result Grid



Filter Rows:

	hour(time)	count(order_id)
▶	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

Your paragraph text

Join relevant tables to find the Category-wise Distribution of Pizzas.

```
1 • SELECT
2     category, COUNT(name)
3 FROM
4     pizza_types
5 GROUP BY category;
```

Result Grid   Filter Rows:

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

Your paragraph text

Group the Orders by Date and Calculate the Average Number of Pizzas ordered per Day.

```
1 • SELECT
2     ROUND(AVG(quantity), 0)
3 FROM
4     (SELECT
5         orders.date, SUM(order_details.quantity) AS quantity
6     FROM
7         orders
8     JOIN order_details ON orders.order_id = order_details.order_id
9     GROUP BY orders.date) AS order_quantity;
10
```

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

	round(avg(quantity),0)
▶	138

Your paragraph text

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

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

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

	name	revenue
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

Your paragraph text

Calculate the Percentage contribution of each Pizza Type to Total Revenue.

1

●

SELECT

2

pizza_types.category,

3

ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT

4

ROUND(SUM(order_details.quantity * pizzas.price),

5

0) AS total_sales

6

FROM

7

order_details

8

JOIN

9

pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,

10

2) AS revenue_percentage

11

FROM

12

pizza_types

13

JOIN

14

pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id

15

JOIN

16

order_details ON order_details.pizza_id = pizzas.pizza_id

17

GROUP BY pizza_types.category

18

ORDER BY revenue_percentage DESC;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	category	revenue_percentage
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Your paragraph text

Analyze the Cumulative Revenue generated over Time...

```
1 • select date,
2     sum(revenue) over(order by date) as cum_revenue
3 from
4     (select orders.date,
5        round(sum(pizzas.price* order_details.quantity),0) as revenue
6     from order_details join pizzas
7     on order_details.pizza_id= pizzas.pizza_id
8     join orders
9     on order_details.order_id= orders.order_id
10    group by orders.date) as sales;
```

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

	date	cum_revenue
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9863
	2015-01-05	11929
	2015-01-06	14358
	2015-01-07	16560
	2015-01-08	19398
	2015-01-09	21525
	2015-01-10	23989
	2015-01-11	25861

Your paragraph text

Determine the Top 3 Most Ordered Pizza types based on Revenue for each Pizza Category

```
1 • select name, revenue
2   from
3   (select category, name, revenue,
4    rank() over(partition by category order by revenue desc) as rn
5   from
6   (select pizza_types.category, pizza_types.name,
7    ROUND(SUM(pizzas.price * order_details.quantity), 0) 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, pizza_types.name) as a) as b
13  where rn <=3;
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	name	revenue			
▶	The Thai Chicken Pizza	43434			
	The Barbecue Chicken Pizza	42768			
	The California Chicken Pizza	41410			
	The Classic Deluxe Pizza	38180			
	The Hawaiian Pizza	32273			
	The Pepperoni Pizza	30162			

Your paragraph text

Key findings of the Project

- Total Revenue: \$817,860.05 from 21,350 orders.
- Most Common Size: Large size ordered 18,526 times.
- Top-Selling Pizza: Classic Deluxe Pizza with 2,453 orders.
- Most Ordered Category: Classic Pizza with 14,888 orders.
- Revenue Contribution: Classic Pizza at 26.91% of total revenue.



Thank you

