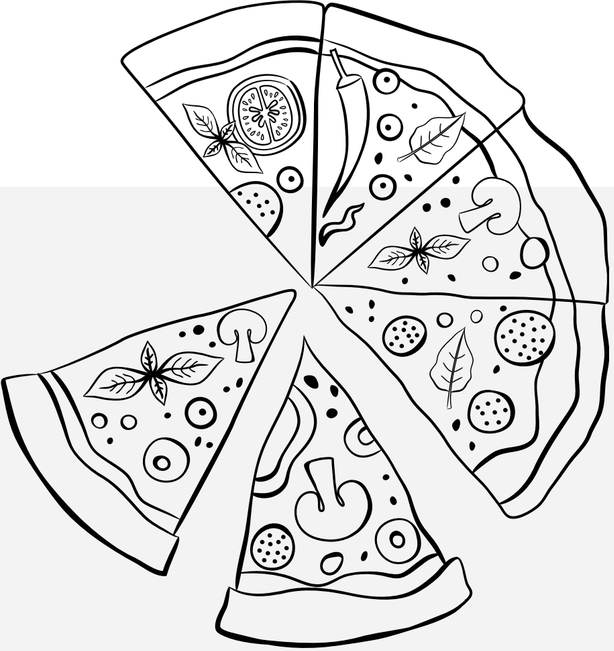


PIZZA SALES DATA ANALYSIS USING SQL

HELLO!

I conducted a data analysis project on pizza sales using SQL. The goal was to uncover key insights such as top-selling pizzas, customer preferences, and revenue trends. The analysis ranged from basic metrics like total orders and revenue to more advanced insights like revenue contribution by pizza type.

TOOLS AND TECHNOLOGIES USED

The project utilized SQL for data querying and analysis, while initial data cleaning and preparation were performed using Excel. This combination ensured efficient data handling and accurate insights.

DATA DESCRIPTION

Four Tables named :

- order_details
- orders
- pizza_types
- pizzas

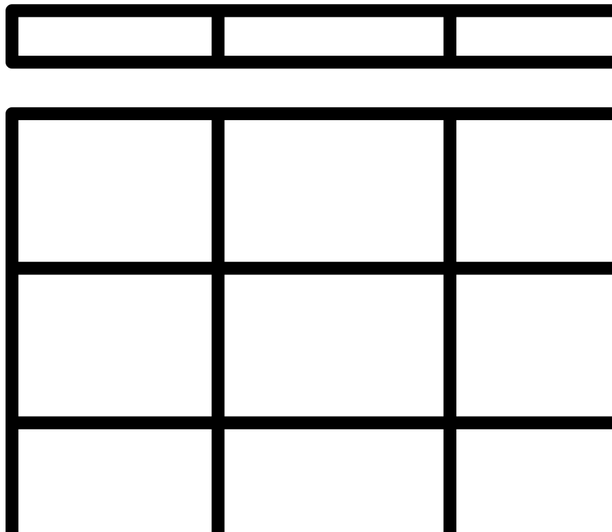


Table: order_details

Columns:

order_details_id	int PK
order_id	int
pizza_id	text
quantity	int

Table: orders

Columns:

order_id	int PK
order_date	date
order_time	time

Table: pizza_types

Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

Table: pizzas

Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

FOR DATA CLEANING

For data cleaning, I removed duplicate rows, deleted rows with missing values, filled in missing data using adjacent values, standardized text entries, and trimmed extra whitespace using Excel.

QUERIES

- Total Number of Orders Placed
- Total Revenue Generated
- Highest-Priced Pizza
- Most Common Pizza Size Ordered
- Top 5 Most Ordered Pizza Types
- Total Quantity of Each Pizza Category Ordered

QUERIES

- Distribution of Orders by Hour of the Day
- Category-Wise Distribution of Pizzas
- Average Number of Pizzas Ordered Per Day
- Top 3 Most Ordered Pizza Types by Revenue
- Percentage Contribution of Each Pizza Type to Total Revenue

QUERIES

- Cumulative Revenue Over Time
- Top 3 Most Ordered Pizza Types by Revenue for Each Category

QUERY-1

Total Number of Orders Placed.

```
-- 1.Retrieve the total number of orders placed.--
select count(order_id) as Total_Orders from orders;
```

Result Grid

	Total_Orders
▶	21350

QUERY-2

Total Revenue Generated

```sql
-- 2.Calculate the total revenue generated from pizza sales.
SELECT
    ROUND(SUM(od.quantity * p.price), 2) AS Total_Revenue
FROM
    order_details AS od
        LEFT JOIN
    pizzas AS p ON od.pizza_id = p.pizza_id;
```

Result Grid

	Total_Revenue
▶	817860.05

QUERY-3

Highest-Priced Pizza

```sql
-- 3.Identify the highest-priced pizza--
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

QUERY-4

Most Common Pizza Size Ordered

```
-- 4.Identify the most common pizza size ordered.
select p.size,sum(quantity) as net_quantity from
order_details as od left join pizzas as p on od.pizza_id=p.pizza_id
group by p.size order by net_quantity desc limit 1;
```

Result Grid | Filter Ro

size	net_quantity
L	18956

QUERY-5

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

```sql
-- 5.List the top 5 most ordered pizza types along with their qua
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

QUERY-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

QUERY-6

Total Quantity of Each Pizza Category Ordered

```
-- 6.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
pizzas join pizza_types on pizzas.pizza_type_id=pizza_types.pizza_type_id join
order_details on pizzas.pizza_id=order_details.pizza_id
group by pizza_types.category;
```

Result Grid | Filter

category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

QUERY-7

Distribution of Orders by Hour of the Day

```sql
-- 7.Determine the distribution of orders by hour of the day.
SELECT HOUR(order_time) AS Hour,COUNT(orders.order_id) AS number_of_orders
FROM orders JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY Hour
ORDER BY Hour ASC;
```

QUERY-7

Result Grid | Filter Rows:

Hour	number_of_orders
9	4
10	17
11	2672
12	6543
13	6203
14	3521
15	3170
16	4185
17	5143
18	5359
19	4350

Result 8

QUERY-8

Category-Wise Distribution of Pizzas

```
-- 8.Join relevant tables to find the category-wise distribution of pizzas.
select category,count(name) as Count from pizza_types group by category;
```

Result Grid | Filter

category	Count
Chicken	6
Classic	8
Supreme	9
Veggie	9

QUERY-9

Average Number of Pizzas Ordered Per Day

```
-- Group the orders by date and calculate the average number of pizzas
-- ordered per day.
select round(avg(quantity),0) as Avg_per_day from
(select orders.order_date as date,sum(order_details.quantity) as quantity
from orders
join order_details
on orders.order_id=order_details.order_id group by date) as order_quantity;
```

Result Grid

	Avg_per_day
▶	138

QUERY-10

Top 3 Most Ordered Pizza Types by Revenue

```
-- 10.Determine the top 3 most ordered pizza types based on revenue.--
select pizza_types.name,round(sum(order_details.quantity*pizzas.price),1)
as revenue
from
pizzas join pizza_types 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;
```

QUERY-10

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.2
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Spicy Italian Pizza	34831.2
The Southwest Chicken Pizza	34705.8
The Italian Supreme Pizza	33476.8
The Hawaiian Pizza	32273.2
The Four Cheese Pizza	32265.7
The Sicilian Pizza	30940.5
The Pepperoni Pizza	30161.8

Result 3

QUERY-11

Percentage Contribution of Each Pizza Type to Total Revenue

```sql
with total_revenue as
( select round(sum(order_details.quantity*pizzas.price),0) as total
from pizzas join pizza_types on pizzas.pizza_type_id=pizza_types.pizza_type_id
join order_details on order_details.pizza_id=pizzas.pizza_id),

revenuepizzatype
as(select pizza_types.name,
round(sum(order_details.quantity * pizzas.price),1)  as revenue from
pizzas join pizza_types 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
)

select revenuepizzatype.name,
round((revenuepizzatype.revenue/total_revenue.total)*100,1)
as percentage_contribution from revenuepizzatype join total_revenue;
```

QUERY-11

Result Grid | Filter Rows: | Export:

name	percentage_contribution
The Hawaiian Pizza	3.9
The Classic Deluxe Pizza	4.7
The Five Cheese Pizza	3.2
The Italian Supreme Pizza	4.1
The Mexicana Pizza	3.3
The Thai Chicken Pizza	5.3
The Prosciutto and Arugula Pizza	3
The Barbecue Chicken Pizza	5.2
The Greek Pizza	3.5
The Spinach Supreme Pizza	1.9
The Green Garden Pizza	1.7

Result 4

QUERY-12

Cumulative Revenue Over Time

```
nalyze the cumulative revenue generated over time.
order_date,sum(revenue) over (order by order_date) as cum_revenue from
 orders.order_date,
um(order_details.quantity*pizzas.price),0) as revenue
der_details join pizzas on order_details.pizza_id=pizzas.pizza_id
ders
rs.order_id=order_details.order_id group by orders.order_date) as sales;
```

QUERY-12

Result Grid | Filter Rows:

order_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

Result 5

QUERY-13

Top 3 Most Ordered Pizza Types by Revenue for Each Category

```sql
-- Determine the top 3 most ordered pizza types based on revenue
-- for each pizza category.--
select category,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,
round(sum(order_details.quantity*pizzas.price),1) as revenue
from pizzas join pizza_types 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,pizza_types.name) as a) as b where rn<=3;
```

QUERY-13

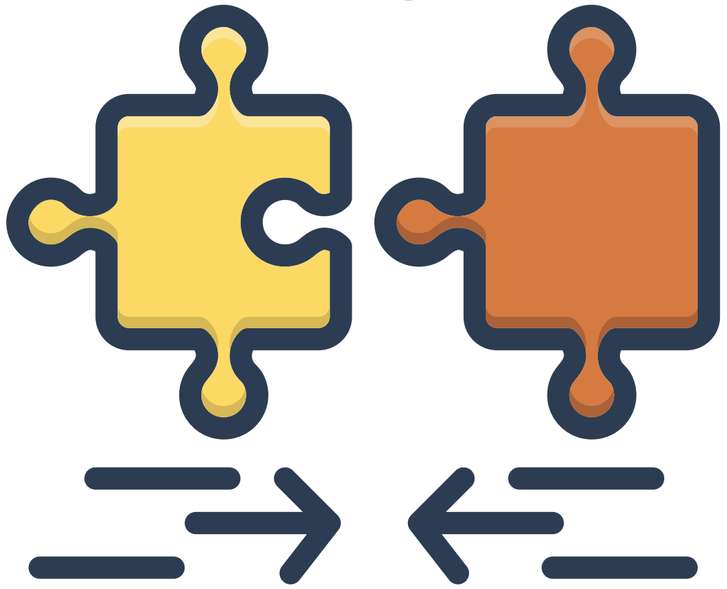
Result Grid | Filter Rows:

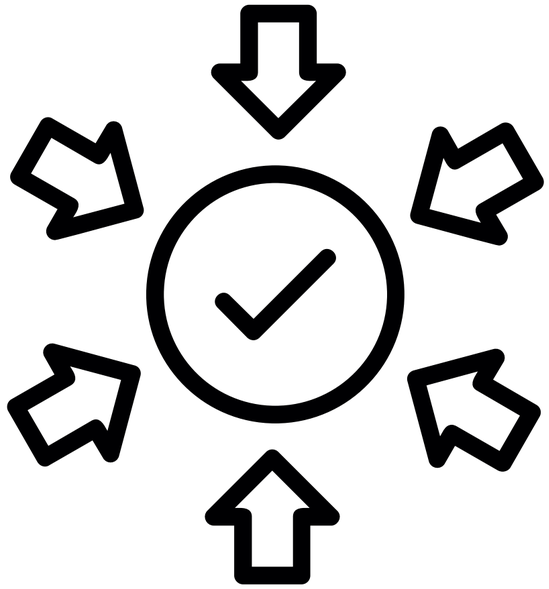
category	name	revenue
Chicken	The Thai Chicken Pizza	43434.2
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.2
Classic	The Pepperoni Pizza	30161.8
Supreme	The Spicy Italian Pizza	34831.2
Supreme	The Italian Supreme Pizza	33476.8
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.7
Veggie	The Mexicana Pizza	26780.8

Result 7

BUSINESS IMPLICATIONS

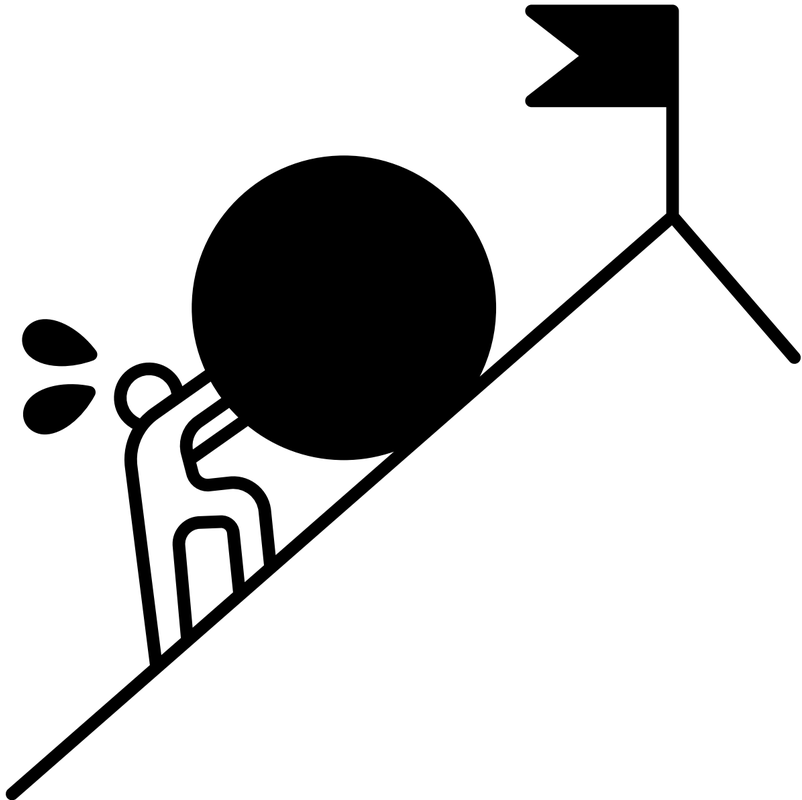
These insights suggest opportunities for targeted promotions during peak hours and optimizing inventory for popular pizza types, potentially increasing profitability and customer satisfaction.





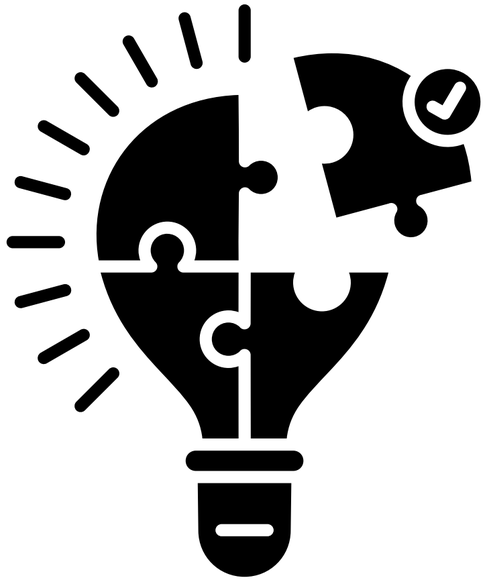
CHALLENGE

One of the main challenges was handling inconsistent data entries and missing values, which could have skewed the analysis.



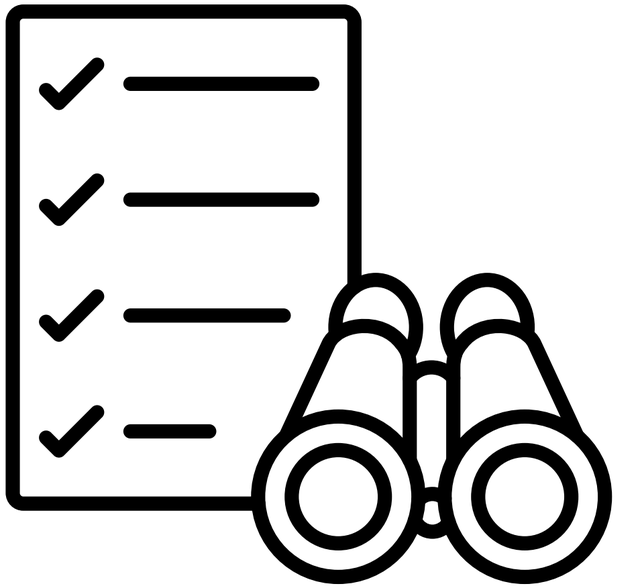
SOLUTION

I addressed this issue by performing data cleaning in Excel, including removing duplicates, standardizing text entries, and filling in missing data, ensuring the dataset was accurate and reliable for analysis.



FUTURE SCOPE

Future Scope: Further analysis could include exploring customer segmentation, seasonal trends, and the impact of promotions on sales, offering deeper strategic insights.



THANK YOU!