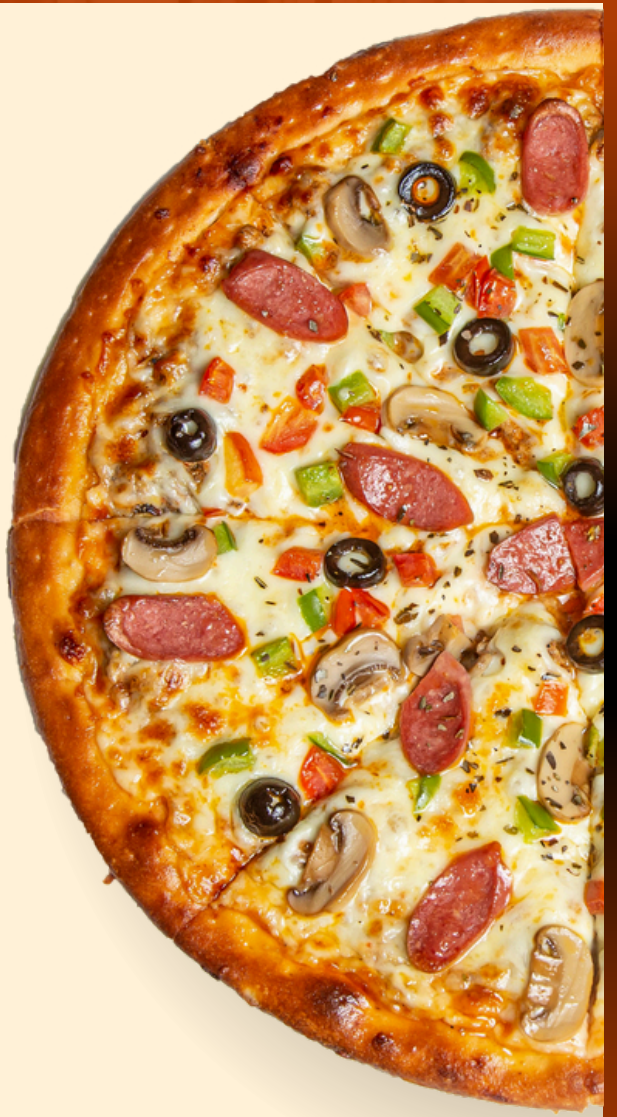
"WE' RE EXCITED TO PRESENT A SLEEK AND SLIGHTFUL REPORT THAT WILL BOOST YOUR UNDERSTANDING AND ENHANCE YOUR PIZZA BUSINESS."

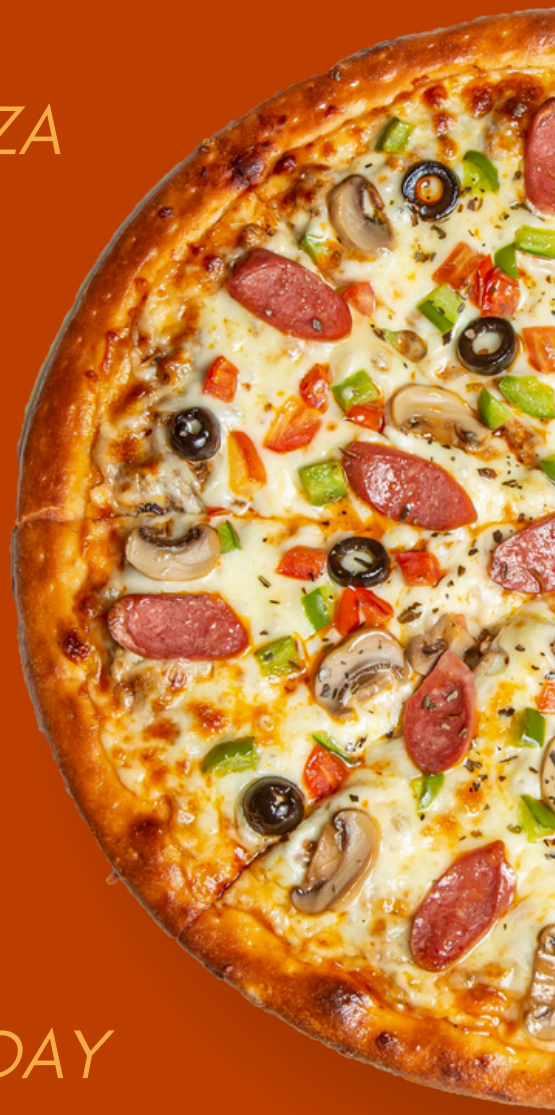
PIZZA SALES
REPORT



PRESENTED BY
ADARSH SAHU

PROJECT OVERVIEW

- *TOTAL REVENUE GENERATED FROM PIZZA SALES*
- *TO IDENTIFY MOST EXPENSIVE PIZZA*
- *WHICH CATEGORIES IN DEMAND*
- *TOP 5 MOST ORDERED PIZZAS*
- *MOST COMMON SIZE PIZZA ORDERED*
- *NUMBER OF ORDERS BY HOUR OF THE DAY*



Q.1 >> RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
1    -- Retrieve the total number of orders placed.
2
3 •  SELECT
4        COUNT(order_id)
5    FROM
6        orders;
```

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

count(order_id)
21350

Q.2 >> CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
1   -- Calculate the total revenue generated from pizza sales.
2   SELECT
3       SUM(order_details.quantity * pizzas.price) AS total_sales
4   FROM
5       order_details
6           JOIN
7       pizzas ON order_details.pizza_id = pizzas.pizza_id;
```

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

total_sales
347935.2000000042

Q.3 >> IDENTIFY THE HIGHEST-PRICED PIZZA.

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

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

name	price
The Greek Pizza	35.95

Q.4 >> IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```sql
# Identify the most common pizza size ordered.
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS most_ordered
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY most_ordered DESC;
```

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

size	most_ordered
L	7898
M	6507
S	6014
XL	249
XXL	15

Q.5 >> LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```sql
# List the top 5 most ordered pizza types along with their quantities.
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS most_ordered
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY most_ordered DESC
LIMIT 5;
```

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

name	most_ordered
The Barbecue Chicken Pizza	1084
The Pepperoni Pizza	1048
The Hawaiian Pizza	1021
The Classic Deluxe Pizza	1007
The Thai Chicken Pizza	981

Q.6 >> JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```sql
# Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS categorywise_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
ORDER BY categorywise_quantity DESC;
```

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

category	categorywise_quantity
Classic	6285
Supreme	5096
Veggie	5022
Chicken	4677

Q.7 >> DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

SELECT
    HOUR(order_time) AS per_hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY per_hour;
```

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

per_hour	order_count
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

Q.8 >> JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
1   # Join relevant tables to find the category-wise distribution of pizzas.
2
3   SELECT
4       category, COUNT(name) as pizzas
5   FROM
6       pizza_types
7   GROUP BY category;
8
```

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

category	pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9

Q.9 >> GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

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

SELECT
    ROUND(AVG(perday_ordered), 0)
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS perday_ordered
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS A;
```

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

round(avg(perday_ordered),0)
139

Q.10 >> DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
# 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;
```

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

name	revenue
The Barbecue Chicken Pizza	19125
The Thai Chicken Pizza	17879.75
The California Chicken Pizza	17215.75

Q.11 >> CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
# Calculate the percentage contribution of each pizza type category to total revenue.
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 order_details.pizza_id = pizzas.pizza_id) * 100,
            2) 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 category
ORDER BY revenue DESC;
```

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

category	revenue
Classic	26.69
Supreme	25.47
Veggie	23.97
Chicken	23.87

Q12 >> ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```sql
# Analyze the cumulative revenue generated over time.

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

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

order_date	cumulative_revenue
2015-01-01	2713.8500000000004
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

Q13 >> DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA 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, 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;
```

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

category	name	revenue
Chicken	The Barbecue Chicken Pizza	19125
Chicken	The Thai Chicken Pizza	17879.75
Chicken	The California Chicken Pizza	17215.75
Classic	The Classic Deluxe Pizza	15698
Classic	The Hawaiian Pizza	13554.75
Classic	The Pepperoni Pizza	13050.5
Supreme	The Spicy Italian Pizza	14898.5
Supreme	The Italian Supreme Pizza	14492
Supreme	The Sicilian Pizza	12968.5
Veggie	The Four Cheese Pizza	14181.30000000011
Veggie	The Five Cheese Pizza	11303.5
Veggie	The Mexicana Pizza	11035.25

CONCLUSION

- *TOTAL REVENUE GENERATED IS AROUND – **"347935"***
- ***"THE GREEK PIZZA"** IS THE MOST EXPENSIVE PIZZA*
- ***"L"** IS THE MOST COMMON SIZE PIZZA ORDERED*
- *TOP 5 MOST ORDERED PIZZA –*

– THE BARBECUE CHICKEN PIZZA – THE PEPPERONI PIZZA
– THE HAWAIIAN PIZZA – THE CLASSIC DELUXE PIZZA
– THE THAI CHICKEN PIZZA

- *TOP CATEGORY WHICH PEOPLE LIKE TO PREFERRED MOST IS **"CLASSIC"** FOLLOWED BY **"SUPREME", "VEGGIE", "CHICKEN"***
- *IN **"AFTERNOON"** AND **"EVENING"** ORDERED OF PIZZA INCREASED MOST*

THANK YOU