PizzaHut

PIZZHUT SALES REPORT

PizzaHut

INTRODUCTION

This project analyzes Pizza Hut sales data to uncover insights into customer preferences and sales performance. We calculate total orders, revenue, and identify top pizzas and sizes. Intermediate analysis includes pizza category quantities and order patterns, while advanced analysis focuses on revenue contributions and trends over time. The aim is to provide actionable insights for better business decisions.

PizzaHut

AGENDA

01 Basic Analysis

- Total number of orders
- Total revenue from pizza sales
- Highest-priced pizza
- Most common pizza size
- Top 5 most ordered pizza types and quantities

02 Intermediate Analysis

- Quantity of each pizza category ordered
- Distribution of orders by hour of the day
- Category-wise distribution of pizzas
- Average number of pizzas ordered per day
- Top 3 pizza types based on revenue

03 Advanced Analysis

- Percentage contribution of each pizza type to total revenue
- Cumulative revenue over time
- Top 3 pizza types based on revenue per category

PizzaHut

BASIC ANALYSIS

- Total number of orders

File 16* | SQL File 4* | SQL File 5* | SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL F

Limit to 1000 rows

```
1   -- retrieve the total numbers of orders placed
2
3   select count(order_id) as total_orders from orders;
```

Result Grid | Filter R

total_orders
21350

PizzaHut

- calculate total revenue genrated from pizza sale

File 16* | SQL File 4* | SQL File 5* | SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL File 10* | SQL File 11* | SQL File 12* | SQL File 13* | SQL File 14*

Limit to 1000 rows

```
1   -- calculate total revenue genrated from pizza sales
2
3   SELECT
4       ROUND(SUM(orders_details.qunatity * pizzas.price),
5               2) AS total_sales
6   FROM
7       orders_details
8           JOIN
9       pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid | Filter R

	total_sales
▶	817860.05

PizzaHut

- **Identify the highest-priced pizza.**

```sql
-- Identify the highest-priced 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;
```

Result Grid | Filter Rows:

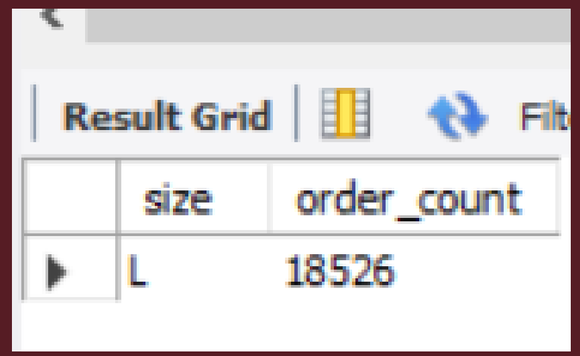
name	price
The Greek Pizza	35.95

PizzaHut

- **Identify the most common pizza size ordered.**

```sql
-- Identify the most common pizza size ordered.

SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```



Result Grid

size	order_count
L	18526

- 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(orders_details.qunatity) AS qunatity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY qunatity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	qunatity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

PizzaHut

Intermediate Analysis

- **Join the necessary tables to find the total quantity of each pizza category ordered.**

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.qunatity) AS qunatity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY qunatity DESC;
```

Result Grid | Filter Rov

category	qunatity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

- **Determine the distribution of orders by hour of the day.**

```
3 •   SELECT
4         HOUR(order_time), COUNT(order_id) AS order_count
5     FROM
6         orders
7     GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

hour(order_time)	order_count
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

8

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

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

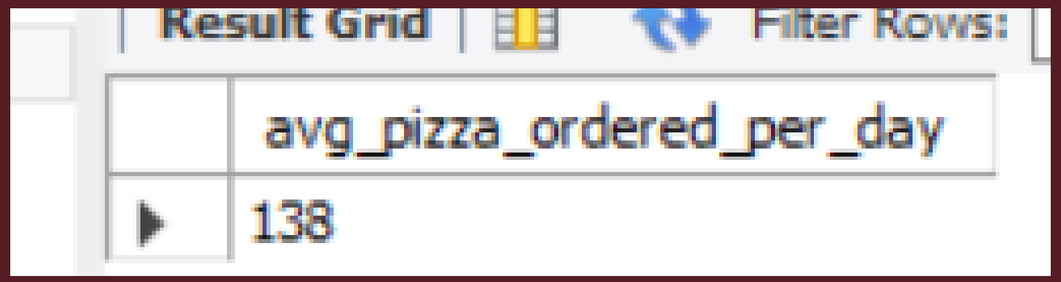
Result Grid | Filter Rows:

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

PizzaHut

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

```sql
SELECT
    ROUND(AVG(qunatity), 0) AS avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.order_date, SUM(orders_details.qunatity) AS qunatity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_qunatity;
```



PizzaHut

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

```sql
SELECT
    pizza_types.name, SUM(qunatity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_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

Advanced Analysis

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.qunatity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.qunatity * pizzas.price),
                                2) AS total_sales
                FROM
                    orders_details
                        JOIN
                    pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
            0) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC
;
```

category	revenue
Classic	27
Supreme	25
Veggie	24
Chicken	24

PizzaHut

- **Analyze the cumulative revenue generated over time.**

```sql
select order_date,
round(sum(revenue) over (order by order_date),0) as cum_revenue from
(select orders.order_date,
sum(orders_details.qunatity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9864

PizzaHut

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

```sql
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(orders_details.qunatity * pizzas.price) AS revenue
FROM
    pizza_types
JOIN
    pizzas
ON
    pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
    orders_details
ON
    orders_details.pizza_id = pizzas.pizza_id
GROUP BY
    pizza_types.category,
    pizza_types.name) as a) as b
    where rn<=3;
```

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

PizzaHut

THANK YOU