FRADEL AND SPIES

PIZZA HUT SALES ANALYSIS PROJECT

Comprehensive SQL Analysis of Pizza Hut Sales Data

BY JAI SAINI

INTRODUCTION

Project Description

I conducted an in-depth analysis of Pizza Hut's sales data using SQL, focusing on key metrics such as total orders, revenue, and popular items. The analysis covered basic, intermediate, and advanced levels, revealing insights into customer preferences, sales trends, and revenue contributions. Detailed SQL queries were employed to extract meaningful data, driving actionable business recommendations. This project showcases my proficiency in SQL and data analytics.

BASIC ANALYSIS

THE TOTAL NUMBER OF ORDERS PLACED PROVIDES A GENERAL OVERVIEW OF THE BUSINESS VOLUME.

```
2
3 •   select count(order_id) as totel_order from orders;
```

totel_order
21350

Understanding total revenue helps gauge the financial performance

```sql
use pizzahut;
            SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_revenue
            FROM
                order_details
                    JOIN
                pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

total_revenue
817860.05

IDENTIFYING THE HIGHEST-PRICED PIZZA HELPS UNDERSTAND THE PRICING STRATEGY AND CUSTOMER PREFERENCES.

```sql
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;
```

name	price
The Greek Pizza	35.95

KNOWING THE MOST COMMONLY ORDERED PIZZA SIZE ASSISTS IN INVENTORY AND MARKETING STRATEGIES.

```
2
3   SELECT
4       pizzas.size,
5       COUNT(order_details.order_details_id) AS order_count
6   FROM
7       pizzas
8           JOIN
9       order_details ON pizzas.pizza_id = order_details.pizza_id
10  GROUP BY pizzas.size
11  ORDER BY order_count DESC;
```

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

Identifying the most popular pizzas can help in promotional activities.

```
3
4   SELECT
5       pizza_types.name, SUM(order_details.quantity) AS quantity
6   FROM
7       pizza_types
8           JOIN
9       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10          JOIN
11      order_details ON order_details.pizza_id = pizzas.pizza_id
12  GROUP BY pizza_types.name
13  ORDER BY quantity DESC
14  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

INTERMEDIATE ANALYSIS

This helps in understanding which categories (like veg, non-veg) are more popular.

```sql
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
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

ANALYZING ORDERS BY HOUR HELPS IDENTIFY PEAK TIMES FOR BUSINESS.

```
2
3 •   SELECT
4         HOUR(order_time) AS hour, COUNT(order_id) AS order_count
5     FROM
6         orders
7     GROUP BY HOUR(order_time);
```

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

Provides a breakdown of sales by different pizza categories.

```
2
3 •   select category, count(name) from pizza_types
4     group by category;
```

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

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

```
3
4   SELECT
5       ROUND(AVG(quantity), 0) AS avg_pizza_ordered_per_day
6   FROM
7       (SELECT
8           orders.order_date, SUM(order_details.quantity) AS quantity
9       FROM
10          orders
11      JOIN order_details ON orders.order_id = order_details.order_id
12      GROUP BY orders.order_date) AS order_quantity;
```

	avg_pizza_ordered_per_day
▶	138

Determine the top 3 most ordered pizza types based on revenue

```sql
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;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

Calculate the percentage contribution of each pizza type to total revenue

```sql
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price ) / (SELECT
                ROUND(SUM(order_details.quantity * pizzas.price),
                        2) AS total_revenue
            FROM
                order_details
                    JOIN
                pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2) as revanue
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 revanue
desc;
```

	category	revanue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Tracking cumulative revenue over time helps understand revenue growth trends

```sql
select order_date,
sum(revanue) over(order by order_date) as cumulative
from
(select orders.order_date,sum(order_details.quantity* pizzas.price) as revanue
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 limit 10;
```

order_date	cumulative
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

Identifies top-performing pizzas in each category, helping in targeted marketing

```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(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
order by revenue desc)as a) as ab
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
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
The Five Cheese Pizza	26066.5

Conclusion:

Recap: Briefly recap the objectives and outcomes of the project.

SQL Importance: Emphasize the role of SQL in extracting and analyzing data for business insights.

Future Steps: Suggest future analysis or steps that could be taken to further improve Pizza Hut's sales strategy.

THANK YOU