
Home
About
Contact
SQL PROJECT TO ANALYSE PIZZA SALES
WHERE EVERY SLIDE TELLS A STORY

ABOUT ME & PROJECT OVERVIEW

Hi, I'm Shraddha Pacharne, a data analyst with expertise in SQL, dashboarding, and real-world business analytics. In this project, I conducted a comprehensive Pizza Sales Analysis using SQL to uncover key performance insights across products and categories.

DATA OVERVIEW

DATASET DESCRIPTION:

THE DATASET CONTAINS DETAILED TRANSACTIONAL RECORDS FROM A PIZZA STORE, CAPTURING SALES ACTIVITY OVER A DEFINED PERIOD.

KEY TABLES & FIELDS:

- ORDERS TABLE: ORDER ID, ORDER DATE, TIME, CUSTOMER INFO
- ORDER DETAILS TABLE: PIZZA ID, QUANTITY, AND PRICE PER ITEM
- PIZZA TABLE: PIZZA NAME, SIZE, CATEGORY (E.G., CLASSIC, VEGGIE, CHICKEN), AND UNIT PRICE
- CATEGORY TABLE (IF SEPARATE): PIZZA CATEGORY METADATA FOR SEGMENTATION

QUESTION WE ANALYSE

Home About

BASIC:

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

IDENTIFY THE HIGHEST-PRICED PIZZA.

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

INTERMEDIATE:

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

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

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

ADVANCED:

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

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

ANALYSE OUTPUT

```sql
-- retrive  total number of orders placed.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

total_orders
21350

```sql
-- caluclate the total reveniue genrated from pizza sales.

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

total_sales
817860.05

ANALYSE OUTPUT

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

name	price
The Greek Pizza	35.95

```sql
-- identify the most common pizza size ordered

SELECT
    COUNT(order_details.order_details_id) as order_count, pizzas.size
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
order by order_count desc;
```

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

ANALYSE OUTPUT

```sql
-- list top 5 most ordered pizza types along with their quantites
SELECT
    pizza_types.name, sum(order_details.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.name
    order by sum(order_details.quantity) desc
    limit 5;
```

name	sum(order_details.quantity)
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

ANALYSE OUTPUT

```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

ANALYSE OUTPUT

```sql
-- determine the distribuiton of orders by hours of the days
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GrOUP BY hour(order_time);
```

HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

```sql
-- join relevent tables to find the category wise distribution of pizza

select category,count(name) from pizza_types
group by category;
```

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

ANALYSE OUTPUT

```sql
-- determine the top 3 most ordered pizza types based on revenu


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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue
limit 3;
```

name	revenue
The Brie Carre Pizza	11588.4999999999
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75

```sql
-- group the orders by date and calulate average number of pizza per order

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

ROUND(AVG(quantity), 0)
138

ANALYSE OUTPUT

```sql
-- calculate the percetnage contribution of each pizza type 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 pizzas.pizza_id = order_details.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 pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYSE OUTPUT

```sql
-- analyze the cumlative revenue genrated over time

select order_date,sum(revenue) over(order by order_date) as cu_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 sales;
```

order_date	cu_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

```sql
-- determine the top 3 most ordered pizza types based on revenue for each pizza category

select name,revenue from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ranking
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 pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category , pizza_types.name) as a) as b
where ranking <= 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


THANK YOU
FOR ATTENTION
CREATED BY SHRADDHA PACHARNE