PIZZA SALES -- SQL PROJECT

Project Overview: Pizza Sales Data Analysis

Objective:

This project aims to analyze pizza sales data to extract key business insights, such as order patterns, revenue generation, popular pizza types, and customer ordering behavior. The analysis covers basic, intermediate, and advanced levels of complexity to help optimize business strategies.

Datasets Provided:

- **orders.csv** – Contains details of customer orders, including order ID, order date, and time.
- **order_details.csv** – Provides itemized order details, including pizza ID, quantity, and price.
- **pizzas.csv** – Contains information about different pizza types, including size, category, and price.
- **pizza_types.csv** – Provides details about pizza names and categories.

Questions:

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.

Retrieve the total number of orders placed

Server Tools Scripting Help

SQL File 3* SQL File 10* SQL File 15* SQL File 13* SQL File 10* SQL File 8* SQL File 9*

Limit to 50 rows

```
-- Retrieve the total number of order placed.

select count(order_id)as total_order from orders;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

total_order
21350

Calculate the total revenue generated from pizza sales

Server Tools Scripting Help

SQL File 9* SQL File 10*

Limit to 50 rows

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

Result Grid Filter Rows: Export: Wrap Cell Content:

total_sales
817860.05

Result 1

Identify the highest-priced pizza

er Tools Scripting Help

L File 9* SQL File 10* SQL File 11*

Limit to 50 rows

```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;
```

sult Grid Filter Rows: Export: Wrap Cell Content: Fetch rows:

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered.

Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* ss - Schema

Limit to 50 rows

```sql
-- identity the most common pizza size orderd
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;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

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

Result 3

Output

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

Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13*

Limit to 50 rows

```sql
-- list the top 5 most orderd pizza types along with their quantity
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

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

Result 1

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

Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13* SQL File 14*

Limit to 50 rows

```
-- join the necessary tables to find the total quantity of each pizza category orderded
select pizza_types.category,
sum(orders_details.quantity)as quantity
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 quantity desc;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Result 1

Determine the distribution of orders by hour of the day

Limit to 50 rows

```sql
-- determine the distribution of orders by hour of the day.
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time) ;
```

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

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

Result 1

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

SQL File 9* | SQL File 11* | SQL File 10* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16*

Limit to 50 rows

```
-- join the relevant tables to find the category wise distribution of pizza.
select category,count(name)from pizza_types
group by category;
```

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

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

Result 1

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

SQL File 9* | SQL File 11* | SQL File 10* | SQL File 12* | SQL File 13* | SQL File 14* | SQL File 15* | SQL File 16* | SQL File 17*

Limit to 50 rows

```sql
-- group the orders by date and calculate the average number of pizzas orderd per day.
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quant;
```

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

ROUND(AVG(quantity), 0)
138

Result 1

Determine the top 3 most ordered pizza types based on revenue

erver Tools Scripting Help

QL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13* SQL File 14* SQL

Limit to 50 rows

```sql
-- determine the top 3 most ordered pizza types based on revenue
SELECT
    pizza_types.name,
    sum(orders_details.quantity * 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;
```

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

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

esult 1

utput

Calculate the percentage contribution of each pizza type to total revenue

ase Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13* SQL File 14*

Limit to 50 rows

```
-- calculate the percentage contibution of each pizza type to total revenue
select pizza_types.category,
round(sum(orders_details.quantity * pizzas.price) / (select
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id)*100,2) 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;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Result 1

Output

Analyze the cumulative revenue generated over time

Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13* SQL

Limit to 50 rows

```sql
-- analyze the cumulative revenue generated over time
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(orders_details.quantity * 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;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

order_date	cum_revenue
2015-01-01	2713.850000000004
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

Result 1

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

Server Tools Scripting Help

SQL File 9* SQL File 11* SQL File 10* SQL File 12* SQL File 13* SQL File 14* SQL File 15* SQL Fi

Limit to 50 rows

```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 rn
from
(select pizza_types.category, pizza_types.name,
sum((orders_details.quantity) * 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;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

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

Result 2

Project Summary

This project focuses on analyzing pizza sales data to extract key business insights. By examining order patterns, revenue generation, and customer preferences, the analysis helps optimize business strategies. The data includes order details, pizzas, and their categories, which are used to derive valuable metrics such as total revenue, the most ordered pizza types, and sales distribution by time and category.

The analysis starts with fundamental metrics like total orders, revenue, and top-selling pizzas. It then explores order distribution across different hours of the day, pizza categories, and daily trends. Advanced analysis includes revenue contribution by pizza type, cumulative revenue tracking, and category-wise top-performing pizzas.

The insights gained support decision-making for inventory management, marketing strategies, and pricing optimization. By understanding which pizzas drive the most revenue and when customers place the most orders, businesses can improve efficiency and profitability.