Data Analysts

SQL PROJECT OF PIZZA SALES

Start Your Slide

ABOUT

About Our Pizza project short summary

In my Pizza Sales project, I use SQL to explore a database of pizza orders. I answer important business questions by calculating total revenue, finding the best and worst-selling pizzas, and discovering ordering trends throughout the week. This analysis helps understand what customers like and how to improve sales.

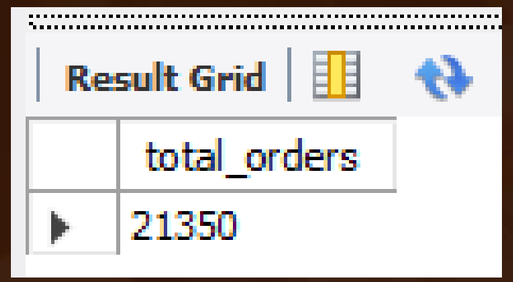
QUESTIONS

1: Retrieve the total number of orders placed.
2: Calculate the total revenue generated from pizza sales.
3: Identify the highest-priced pizza.
4: Join the necessary tables to find the total quantity of each pizza category ordered.

5: Determine the distribution of orders by hour of the day.
6: Join relevant tables to find the category-wise distribution of pizzas.
7:Calculate the percentage contribution of each pizza type to total revenue.
8: Analyze the cumulative revenue generated over time.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
-- 1st question
-- Retrieve the total number of orders placed.
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
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;
```

Result Grid

	Total_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95

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

```sql
select pizza_types.category,
sum(order_details.quantity) as order_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 order_quantity desc;
```

Result Grid | Filter Rows:

category	order_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS count_order
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid

hour	count_order
11	1231
12	2520
13	2455
14	1472
15	1468

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 Ro

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

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_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id) ) * 100, 2) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
```

Result Grid | Filter R

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED: OVER TIME.

```sql
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 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
```

Result Grid | Filter Rows:

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

Result 9

IF YOU WANT MORE SQL PROJECT

Follow My Github Account

Pizza Project Presentation

THANK YOU FOR ATTENTION

See You Next

https://github.com/shahlarafiq12