ANALYSIS OF PIZZA SALES USING SQL

Presented by
Punnya Prasannan

INTRODUCTION

- This presentation provides an overview of a project analyzing pizza sales data using SQL.
- The goal of the project was to uncover valuable insights to help improve business operations and decision-making.
- Key questions addressed include cumulative revenue generated over time, percentage contribution of each pizza types, and customer preferences.

DATA SOURCE

The analysis was conducted using a comprehensive dataset of pizza sales transactions.

Orders Table

Column Name	Data Type	Description
Order_id	INT	Unique identifier for each order
Order_date	DATE	Date the order was placed.
Order_time	TIME	Time the order was placed

Order_Details Table

Column Name	Data Type	Description
Order_details_id	INT	Unique identifier for each order details
Order_id	INT	Unique identifier for each order
Pizza_id	TEXT	Unique identifier for each pizza
Quantity	INT	Quantity ordered

DATA SOURCE

The analysis was conducted using a comprehensive dataset of pizza sales transactions.

Pizzas Table

Column Name	Data Type	Description
Pizza_id	VARCHAR	Unique identifier for each pizza
Pizzatype_id	VARCHAR	Unique identifier for each pizza type
Size	VARCHAR	Size of pizza ordered
Price	DOUBLE	Price of pizza ordered

Pizza_types Table

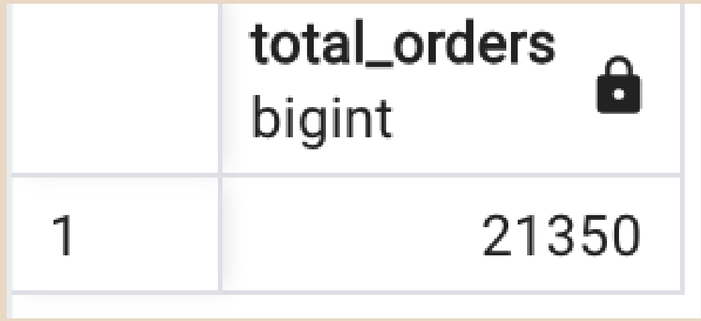
Column Name	Data Type	Description
Pizza_types_id	VARCHAR	Unique identifier for each pizza type
Name	TEXT	Namee of pizza ordered
Category	TEXT	Category of pizza ordered
Ingredients	TEXT	Ingredients used

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

QUERY

```
Select count(order_id) as
total_orders from orders;
```

OUTPUT



	total_orders bigint
1	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

QUERY

```
select sum(order_details.quantity
* pizzas.price)
as total_revenue from
order_details join pizzas
on pizzas.pizza_id =
order_details.pizza_id;
```

OUTPUT

	total_revenue double precision
1	817860.049999993

IDENTIFY THE HIGHEST-PRICED PIZZA.

QUERY

```
select
pizza_types.name,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_types_id =
pizzas.pizzatype_id
order by pizzas.price desc limit 1;
```

OUTPUT

	name text	price double precision
1	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

QUERY

```
select
pizzas.size,count(order_details.or
der_details_id) as common_pizza
from pizzas join order_details
on pizzas.pizza_id =
order_details.pizza_id
group by pizzas.size order by
common_pizza desc limit 1;
```

OUTPUT

	size character varying (5)	common_pizza bigint
1	L	18526

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

QUERY

```
select pizza_types.name,
sum(order_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_types_id =
pizzas.pizzatype_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by
quantity desc limit 5;
```

OUTPUT

	name text	quantity bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

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

QUERY

```
select pizza_types.category,
sum(order_details.quantity) as quantity
from pizza_types join pizzas
on pizza_types.pizza_types_id =
pizzas.pizzatype_id
join order_details
on order_details.pizza_id =
pizzas.pizza_id
group by pizza_types.category order by
quantity desc;
```

OUTPUT

	category text	quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

QUERY

```
select EXTRACT(HOUR FROM order_time) AS hour,count(order_id) from orders group by hour order by hour desc;
```

OUTPUT

	hour numeric	count bigint
1	23	28
2	22	663
3	21	1198
4	20	1642
5	19	2009
6	18	2399
7	17	2336
8	16	1920
9	15	1468
10	14	1472
11	13	2455
12	12	2520
13	11	1231
14	10	8
15	9	1

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

QUERY

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

OUTPUT

	category text	count bigint
1	Supreme	9
2	Chicken	6
3	Classic	8
4	Veggie	9

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

QUERY

```
select ROUND(avg(quantity),0) as
avg_pizzas_ordered_per_day from
(select
orders.order_date,sum(order_details.qua
ntity)as quantity
from orders join order_details
on orders.order_id =
order_details.order_id
group by orders.order_date) as
order_quantity;
```

OUTPUT

	avg_pizzas_ordered_per_day numeric
1	138

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

QUERY

```
select
pizza_types.name,sum(order_details.quantity
* pizzas.price)
as revenue from pizza_types join pizzas
on pizzas.pizzatype_id =
pizza_types.pizza_types_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue
desc limit 3;
```

OUTPUT

	name text	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

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

QUERY

```
select pizza_types.category,
(sum(order_details.quantity * pizzas.price)/(select
sum(order_details.quantity * pizzas.price)
as total_sales
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id))*100 as
revenue
from pizza_types join pizzas
on pizzas.pizzatype_id = pizza_types.pizza_types_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by revenue desc;
```

OUTPUT

	category text	revenue double precision
1	Classic	26.905960255669903
2	Supreme	25.45631126009884
3	Chicken	23.955137556847493
4	Veggie	23.682590927384783

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY

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

OUTPUT

	order_date date	cum_revenue double precision
1	2015-01-01	2713.8500000000004
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.6
5	2015-01-05	11929.55
6	2015-01-06	14358.5
7	2015-01-07	16560.7
8	2015-01-08	19399.05
9	2015-01-09	21526.4
10	2015-01-10	23990.350000000002
11	2015-01-11	25862.65

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

QUERY

```
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 pizzas.pizzatype_id = pizza_types.pizza_types_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category,pizza_types.name) as A)
as B
where rn<=3;
```

OUTPUT

	name text	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5
4	The Classic Deluxe Pizza	38180.5
5	The Hawaiian Pizza	32273.25
6	The Pepperoni Pizza	30161.75
7	The Spicy Italian Pizza	34831.25
8	The Italian Supreme Pizza	33476.75
9	The Sicilian Pizza	30940.5
10	The Four Cheese Pizza	32265.70000000065
11	The Mexicana Pizza	26780.75
12	The Five Cheese Pizza	26066.5

KEY INSIGHTS

Most Ordered Pizza

The analysis revealed that the Classic Deluxe Pizza is the most ordered pizza type.

Peak Sales Time during the day

The peak sales of the day occurred at 12 PM.

The category with the highest number of pizza types.

Supreme and Veggie category has the highest number of pizza types.

CONCLUSION

- The project successfully used SQL queries to analyze pizza sales data and extract valuable business insights.

- Key findings include identifying most ordered pizza type, peak sales time during the day, and the category with the highest number of pizza types.