

PRESENTATION

BY ALTAMASH PATIL

PROJECT TITLE: ANALYZING PIZZA SALES DATA

Introduction: Hello, I'm Altamash Patil, and I'm excited to work on this SQL project analyzing pizza sales data. The goal of this project is to gain insights into customer behavior, sales trends, and product performance. By analyzing the data, we can identify areas of improvement and provide recommendations to stakeholders in the pizza industry.

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;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

Result Grid

	total_orders
21350	

CALCULATE THE TOTAL REVENUE GENERATED 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;
```

Result Grid

	total_sales
817860.05	

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS
    order_count
FROM
    pizzas
    JOIN
    order_details ON pizzas.pizza_id =
order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```



The Result Grid shows the following data:

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

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

```
SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter 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

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

```
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;
```



The Result Grid displays the output of the SQL query. It shows four rows of data with columns labeled 'category' and 'quantity'. The categories are Classic, Supreme, Veggie, and Chicken, with their respective quantities.

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

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

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

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

```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

Result Grid | Filter Rows:

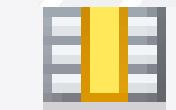
	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

SELECT

```
    ROUND(AVG(quantity), 0) AS  
avg_pizza_ordered_per_day  
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  
order_quantity;
```

Result Grid



Filter Rows:

avg_pizza_ordered_per_day
138

138

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) 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.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

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(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	cum_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

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

```
SELECT  
    pizza_types.name, 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.name  
ORDER BY quantity DESC  
LIMIT 5;
```

Result Grid | Filter Rows:

	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

THANK YOU