



LICERIA
& CO.

PIZZA SALES

PRESENTED BY
ANKITA SINGH VATS

DESCRIPTION:
EXPLORE OUR SQL DATA ANALYSIS
PROJECT PRESENTATION:

UNVEIL INSIGHTS FROM DATABASE QUERIES
DISCOVER TRENDS AND PATTERNS
CLEAR METHODOLOGIES, ACTIONABLE FINDINGS
CAPTIVATING VISUALS, PRECISE RECOMMENDATIONS
UNLOCK THE POWER OF SQL FOR INNOVATION!

INTRODUCTION

- >GREETINGS! WELCOME TO OUR SQL DATA ANALYSIS PROJECT PRESENTATION.
- >TODAY, WE EMBARK ON A JOURNEY THROUGH DATA-DRIVEN DISCOVERY, POWERED BY SQL.
- >IN THIS PRESENTATION, WE'LL EXPLORE HOW SQL ENABLES US TO EXTRACT ACTIONABLE INSIGHTS FROM RAW DATA.
- >FROM UNCOVERING TRENDS AND PATTERNS TO FORMULATING STRATEGIC RECOMMENDATIONS, WE'LL SHOWCASE THE TRANSFORMATIVE POTENTIAL OF SQL IN DATA ANALYSIS

OBJECTIVES

OUR OBJECTIVES:

- TO DEMONSTRATE THE EFFECTIVENESS OF SQL IN DATA ANALYSIS.
- TO PRESENT KEY INSIGHTS DERIVED FROM OUR ANALYSIS.
- TO PROVIDE ACTIONABLE RECOMMENDATIONS BASED ON THESE INSIGHTS.
- BY THE END OF THIS PRESENTATION, YOU'LL HAVE A CLEAR UNDERSTANDING OF THE POWER OF SQL IN UNLOCKING VALUABLE INTELLIGENCE FOR DECISION-MAKING AND INNOVATION.

-- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SELECT

COUNT(order_id)

FROM

orders;

COUNT(order_id)

21350

-- 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;



	total_sales
▶	817860.05

-- 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;
```

	name	price
▶	The Greek Pizza	35.95

-- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
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;
```

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(orders_details.quantity) AS total_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 total_quantity  
LIMIT 5;
```

	name	total_quantity
▶	The Brie Carre Pizza	1470
	The Mediterranean Pizza	2802
	The Calabrese Pizza	2811
	The Spinach Supreme Pizza	2850
	The Soppressata Pizza	2883

-- JOIN THE NECESSARY TABLES TO FIND THE CATEGORY OF EACH PIZZA ORDERED

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;
```

	category	quantity
▶	Classic	44664
	Supreme	35961
	Veggie	34947
	Chicken	33150

-- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

SELECT

HOUR(order_time) AS hour, COUNT(*) AS order_count

FROM

orders

GROUP BY HOUR(order_time);

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

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

SELECT

category, COUNT(name)

FROM

pizza_types

GROUP BY category;

category	count(name)
----------	-------------

Chicken	18
---------	----

Classic	24
---------	----

Supreme	27
---------	----

Veggie	27
--------	----

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

```
SELECT ROUND(AVG(order_quantity), 0) AS average_quantity
FROM (
    SELECT order_date, SUM(quantity_order) AS order_quantity
    FROM (
        SELECT orders.order_date, SUM(orders_details.quantity) AS quantity_order
        FROM orders
        JOIN orders_details ON orders.order_id = orders_details.order_id
        GROUP BY orders.order_date
    ) AS order_quantity_per_day
    GROUP BY order_date
) AS aggregated_orders;
```

average_quantity

138

-- 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;

name	revenue
The Thai Chicken Pizza	130302.75
The Barbecue Chicken Pizza	128304
The California Chicken Pizza	124228.5

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

```
select pizza_types.category,  
round(sum(orders_details.quantity * pizzas.price) / (SELECT SUM(orders_details.quantity * pizzas.price) AS total_sales  
FROM orders_details JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 180, 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;
```

	category	revenue
▶	Classic	145.29
	Supreme	137.46
	Chicken	129.36
	Veggie	127.89

-- 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;
```

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
2015-01-09	21526.4
2015-01-10	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004
2015-01-15	34343.50000000001
2015-01-16	36937.65000000001
2015-01-17	39001.75000000001
2015-01-18	40978.60000000006
2015-01-19	43365.75000000001
2015-01-20	45763.65000000001
2015-01-21	47804.20000000001
2015-01-22	50300.90000000001
2015-01-23	52724.60000000006
2015-01-24	55013.85000000006
2015-01-25	56631.40000000001

-- 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;
```

	name	revenue
▶	The Thai Chicken Pizza	130302.75
	The Barbecue Chicken Pizza	128304
	The California Chicken Pizza	124228.5
	The Classic Deluxe Pizza	114541.5
	The Hawaiian Pizza	96819.75
	The Pepperoni Pizza	90485.25
	The Spicy Italian Pizza	104493.75
	The Italian Supreme Pizza	100430.25
	The Sicilian Pizza	92821.5
	The Four Cheese Pizza	96797.09999999325
	The Mexicana Pizza	80342.25
	The Five Cheese Pizza	78199.5