

PIZZA SALES DATA ANALYSIS USING SQL



OBJECTIVE OF THE PROJECT

- Analyse pizza sales data to uncover business insights
- Use SQL queries for metrics like revenue, top-selling pizzas, and peak hours
- Help decision-makers optimize inventory, staffing, and marketing strategies



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(orders_details.quantity * pizzas.price),
```

```
2) AS total_revenue
```

FROM

```
orders_details
```

JOIN

```
pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

ORDER BY total_revenue DESC;

Result Grid	
	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS most_common_ordered
FROM
    pizzas
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY most_common_ordered DESC
LIMIT 1;
```

Result Grid | Filter Rows:

	size	most_common_ordered
▶	L	18526

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

```
SELECT
    pizza_types.name, COUNT(orders_details.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 COUNT(orders_details.quantity) DESC
LIMIT 5;
```

Result Grid | Filter Rows: _____ | Export:

	name	COUNT(orders_details.quantity)
▶	The Classic Deluxe Pizza	2416
	The Barbecue Chicken Pizza	2372
	The Hawaiian Pizza	2370
	The Pepperoni Pizza	2369
	The Thai Chicken Pizza	2315

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

```
SELECT
    pizza_types.category, SUM(orders_details.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 COUNT(orders_details.quantity) DESC
```

	category	SUM(orders_details.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);

	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.

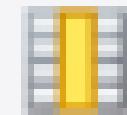
```
SELECT  
    category, COUNT(name)  
FROM  
    pizza_types  
GROUP BY category;
```

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 average_pizza_per_day  
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_quantity;
```

Result Grid |  Filter Rows: 

	average_pizza_per_day
▶	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 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 revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

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

CALCULATE THE PERCENTAGE CONTRIBUTION 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_revenue
    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
LIMIT 10;
```

Result Grid |   Filter Rows: 

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

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  
ORDER 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

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	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
	The Italian Supreme Pizza	33476.75
	The Sicilian Pizza	30940.5
	The Four Cheese Pizza	32265.70000
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

THANK YOU!

THANK YOU FOR TAKING THE TIME TO EXPLORE THIS PIZZA SALES SQL ANALYSIS PROJECT.
I HOPE THIS GAVE YOU A CLEAR VIEW OF HOW STRUCTURED QUERY LANGUAGE CAN TURN
RAW DATA INTO POWERFUL INSIGHTS.

