

DATA-DRIVEN PIZZA SALES INSIGHTS



A COMPREHENSIVE ANALYSIS UNCOVERING TOTAL ORDERS, REVENUE TRENDS, BESTSELLING PIZZAS, CUSTOMER PREFERENCES, AND CATEGORY-WISE PERFORMANCE. FROM BASIC METRICS TO ADVANCED REVENUE INSIGHTS, THIS PROJECT REVEALS PATTERNS THAT HELP OPTIMIZE MENU STRATEGY AND BOOST BUSINESS GROWTH.

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. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.
6. JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.
11. CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
13. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

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

Result Grid	
	Total_Orders_Placed
▶	21350

CALCAULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT
```

```
    ROUND(SUM(order_details.quantity * pizzas.price),  
          1) AS Total_Sales
```

```
FROM
```

```
order_details
```

```
JOIN
```

```
pizzas ON pizzas.pizza_id = order_details.pizza_id;
```

	Result Grid	Filter Rows
	Total_Sales	
▶	817860	

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(order_details.order_details_id) AS order_count
```

FROM

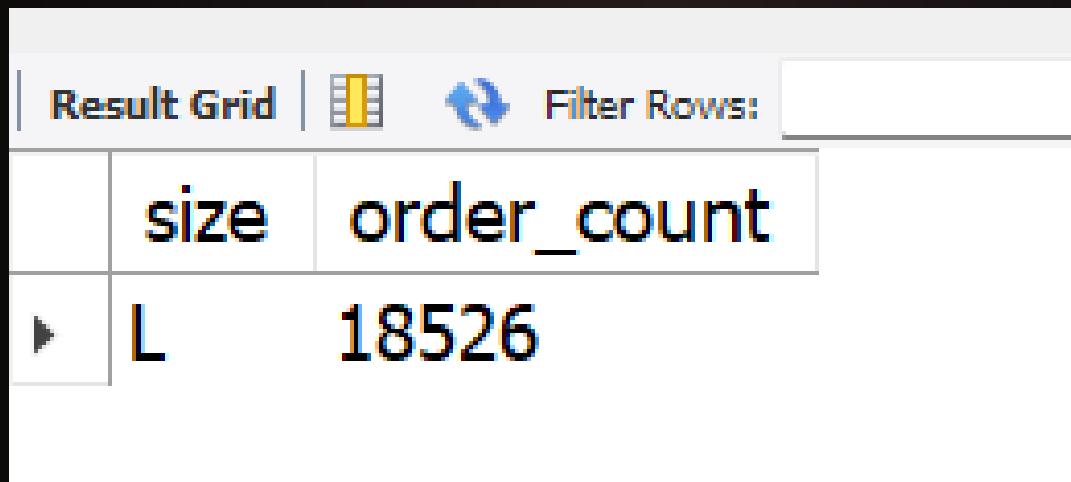
```
pizzas
```

JOIN

```
order_details ON pizzas.pizza_id = order_details.pizza_id
```

GROUP BY pizzas.size

ORDER BY order_count **DESC** ;



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'size' and 'order_count'. There is one row of data: size 'L' with an order count of 18526. The grid includes standard database navigation icons (refresh, filter, etc.) at the top.

	size	order_count
▶	L	18526

IDENTIFY THE MOST COMMON PIZZALIST
THE TOP 5 MOST ORDERED PIZZA TYPES
ALONG WITH THEIR QUANTITIES. SIZE
ORDERED.

```
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;
```

	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**;

	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);

	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

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_order_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;

	avg_pizza_order_per_day
▶	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;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue 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(order_details.quantity * pizzas.price) / (SELECT  
        ROUND(SUM(order_details.quantity * pizzas.price),  
            1) 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 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 revenue DESC;
```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,  
Round(SUM(revenue) over (order by order_date) ,2)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;
```

	order_date	cum_revenue
▶	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6

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((order_details.quantity)* pizzas.price) as revenue
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,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

YOU CAN CONNECT ME ON



WWW.LINKEDIN.COM/IN/WANIUMER-
ANALYTICS