



PIZZA SALES ANALYSIS





Description



I completed a Pizza Sales Analysis project using MySQL with the pizza_hut database, which includes tables for pizzas, orders, order_details, and pizza_types. The project focused on solving various SQL queries, generating insights such as calculating pizza sales revenue, identifying customer preferences, and analyzing sales trends. This work enhanced my SQL skills and provided valuable data-driven insights for optimizing sales strategies.





Questions


Basic:

- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

Intermediate:

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category-wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

Advanced:

- Calculate the percentage contribution of each pizza type to total revenue.
 - Analyze the cumulative revenue generated over time.
 - Determine the top 3 most ordered pizza types based on revenue for each pizza category.
- 



```
-- 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_revenue
```


```
FROM
```

```
    order_details
```

```
      JOIN
```

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

Result Grid		Filter Rows:
	total_revenue	
▶	84616.40	





```
-- Identify the highest-priced pizza.
```

```
SELECT
```

```
    pizza_types.name, MAX(pizzas.price) AS highest_price
```

```
FROM
```

```
    pizza_types
```

```
        INNER JOIN
```

```
        pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
```

```
ORDER BY pizzas.price DESC;
```



	name	highest_price
►	The Barbecue Chicken Pizza	35.95





```
-- Identify the most common pizza size ordered.
```

```
SELECT
```

```
    pizzas.size, COUNT(order_details.order_id) as order_id
```

```
FROM
```

```
    pizzas
```

```
        INNER JOIN
```

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

```
GROUP BY pizzas.size
```

```
order by order_id desc;
```

	size	order_id
▶	L	1936
	M	1568
	S	1480
	XL	57
	XXL	2





```
-- 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;
```

	name	quantity
►	The Pepperoni Pizza	288
	The Barbecue Chicken Pizza	259
	The California Chicken Pizza	248
	The Classic Deluxe Pizza	231
	The Hawaiian Pizza	229





```
-- 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  
    INNER JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
    INNER JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY category  
ORDER BY quantity DESC;
```


	category	quantity
▶	Classic	1534
	Supreme	1274
	Veggie	1227
	Chicken	1100





-- Determine the distribution of orders by hour of the day.


```
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY HOUR(order_time) ASC;
```



	HOUR(order_time)	COUNT(order_id)
▶	9	1
	10	8
	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1100

Result 1 x

Output





```
-- Join relevant tables to find the category-wise distribution of pizzas.
```

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

	category	count(pizza_types.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_order_pizza
```

```
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_order_pizza
▶	139





```
-- 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 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 revenue DESC
```


```
LIMIT 3;
```

	name	revenue
▶	The Barbecue Chicken Pizza	4618.25
	The California Chicken Pizza	4270.00
	The Thai Chicken Pizza	4170.25





```
-- 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),
            2) AS total_revenue
        FROM
            order_details
            JOIN
                pizzas ON order_details.pizza_id = pizzas.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
```



	category	revenue
▶	Classic	26.73
	Supreme	25.93
	Veggie	24.30
	Chicken	23.04



```
-- 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;
```

	order_date	cum_revenue
►	2015-01-01	2713.85
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.60
	2015-01-05	11929.55
	2015-01-06	14358.50





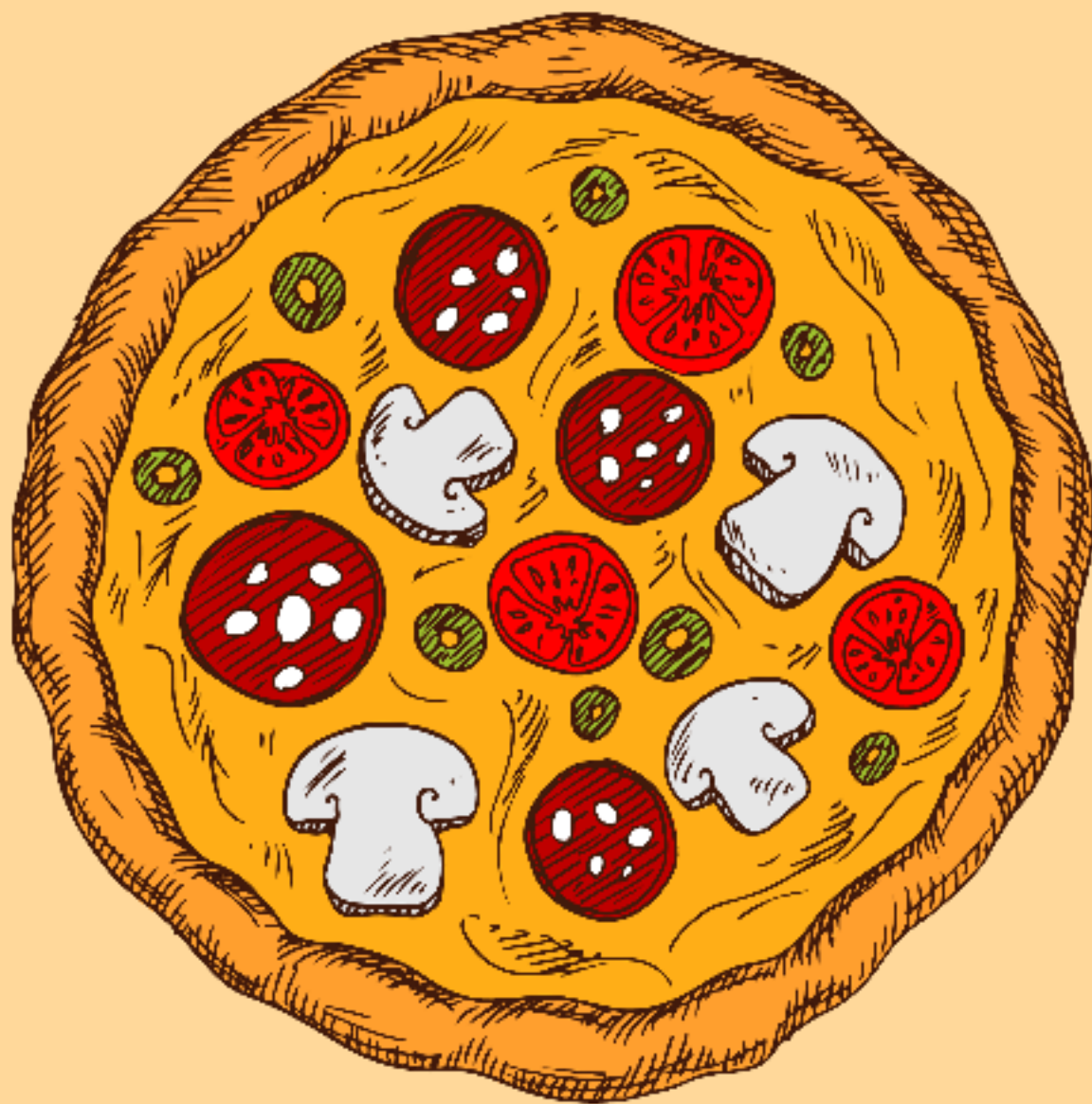
```
-- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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.name , pizza_types.category) as a) as b
where rn<=3;
```

	name	revenue
►	The Barbecue Chicken Pizza	4618.25
	The California Chicken Pizza	4270.00
	The Thai Chicken Pizza	4170.25
	The Pepperoni Pizza	3608.25
	The Classic Deluxe Pizza	3577.50
	The Hawaiian Pizza	3016.25





**THANK
YOU**

