

# PIZZA SALE

## ANALYSIS

TheKartikAnalytics



Sales  
And

Product Analysis

Using MySQL Workbench And Queries



## Operations Done:

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 from orders;
```

	total_orders
▶	21350

2. Calculate the total revenue generated from pizza sales

```

SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
          2)
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id

```

	ROUND(SUM(order_details.quantity * pizzas.price), 2)
▶	817860.05

### 3. 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

#### 4. 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;
```

Result Grid     Filter Rows:		
	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

5. 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 Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

6. 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

## 7. Determine the distribution of orders by hour of the day.

Result Grid | Filter Rows:

	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

```
select hour(order_time) as hour, count(order_id) as order_count from orders  
group by hour(order_time);
```

## 8. 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)
1	Chicken	6
2	Classic	8
3	Supreme	9
4	Veggie	9

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;
```

Result Grid	
	avg_pizza_order_per_day
▶	138

10.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

## 11. 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_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

## 12. 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	2713.850000000004
	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

# 13.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
	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	32265.70000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5



# Thank You

Thekartikanalytics

thekartikanalytics21@gmail.com

Upadhyayk629@gmail.com