

# Pizza Sales Analysis

PIZZA IS ONE OF THE FAVORITE FOODS OF ALMOST EVERYONE IN THE WORLD.

**50%**  
**DISCOUNT**



# Project Description:

---

This project uses SQL to analyze the dataset of an Pizza Sales.

.The objective of the project is to gain valuable insights into the sales operations,

Customer Segmentation, trends, sales growth, customer interest and money spent by customer.

.The goal of the project is to answer a set of questions about the pizza sales business performance and help in its growth by making better decisions.

ToolUsed: MySQL

Tables Used: Pizza\_Sales



## Project Overview

---

The primary objectives of this project are:

- To determine the most profitable days and times.
- To identify the average order values and revenue contributions by pizza categories.
- To understand customer preferences for different pizza sizes.
- To track monthly revenue growth.
- To identify the most popular and most expensive pizzas



## QUERIES OF THE PROJECT:

- 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.
  - 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.





Navigator:

9\*

SQL File 10\*

SQL File 11\*

SQL File 12\*

SQL File 13\*

SQL File 14\* x

SQL File 15\*

SQL File 16\*

pizza\_types

order\_details

pizzas

pizzas

SQLAdditions

&lt; &gt; | I? | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## SCHEMAS

Filter objects

pizzahut

Tables

order\_details

- Columns
- Indexes
- Foreign Keys
- Triggers

orders

- Columns
- Indexes
- Foreign Keys
- Triggers

pizza\_types

- Columns
- Indexes
- Foreign Keys
- Triggers

pizzas

- Columns
- Indexes
- Foreign Keys
- Triggers

Administration

Information

## Table: order\_details

## Columns:

<u>order_details_id</u>	int	PK
order_id	int	text
pizza_id	text	int

Formatted 0 statements, 1 unsupported statement types skipped.

```

1 -- Calculate the percentage contribution of each pizza type to total revenue.
2
3
4 • select pizza_types.category,
5   round(sum(order_details.quantity * pizzas.price) / (select
6     round(sum(order_details.quantity * pizzas.price),
7       2) AS total_sales
8   from
9   order_details join
10  pizzas on pizzas.pizza_id = order_details.pizza_id) *100,2) as revenue
11  from pizza_types join pizzas
12  on pizza_types.pizza_type_id = pizzas.pizza_type_id
13  join order_details
14  on order_details.pizza_id = pizzas.pizza_id
15  group by pizza_types.category order by revenue desc ;

```

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Result 2 x

Result Grid  
Read Only  
Context Help  
Snippets

Output

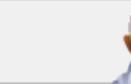
Action Output

#	Time	Action	Message
124	00:42:37	select category, name, revenue, rank() over(partition by category order by revenue desc) as m from (select pizz...	32 row(s) returned
125	00:44:12	select name, revenue from (select category, name, revenue, rank() over(partition by category order by revenue ...	12 row(s) returned

Activate Windows Duration / Fetch  
Go to Settings to activate Windows. 0.250 sec / 0.000 sec  
0.266 sec / 0.000 sec



Type here to search

15:24  
02-10-2024


**pizzahut**  
 Tables

- order\_details
- Columns
- Indexes
- Foreign Keys
- Triggers

- orders
- Columns
- Indexes
- Foreign Keys
- Triggers

- pizza\_types
- Columns
- Indexes
- Foreign Keys
- Triggers

- pizzas
- Columns
- Indexes
- Foreign Keys
- Triggers

```

1 -- Analyze the cumulative revenue generated over time.
2
3
4 • select order_date,
5   sum(revenue) over ( order by order_date) as cum_revenue
6   from
7   (select orders.order_date,
8    sum(order_details.quantity * pizzas.price) as revenue
9    from order_details join pizzas
10   on order_details.pizza_id = pizzas.pizza_id
11   join orders
12   on orders.order_id = order_details.order_id
13   group by orders.order_date) sales;

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: **order\_details**

## Columns:

<u>order_details_id</u>	int	PK
order_id	int	
pizza_id	text	
quantity	int	

	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55



Result Grid



Read Only

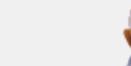
Context Help Snippets

#	Time	Action	Message
124	00:42:37	select category, name, revenue, rank() over(partition by category order by revenue desc) as m from (select pizza...	32 row(s) returned
125	00:44:12	select name, revenue from (select category, name, revenue, rank() over(partition by category order by revenue ...	12 row(s) returned

Activate Windows Duration / Fetch  
 Go to Settings to activate Windows. 0.250 sec / 0.000 sec  
 0.266 sec / 0.000 sec



Type here to search



34°C Mostly cloudy

15:26  
02-10-2024