

# PIZZA



# DESCRIPTION

This project analyzes pizza sales data using SQL to identify top-selling pizzas, peak sales hours, and revenue trends. It leverages SQL queries for insights into customer preferences, inventory optimization, and sales strategies.

# SCHEMA

## **ORDER\_DETAIL**

- order\_detail\_id
- order\_id
- pizza\_id
- quantity

## **PIZZAS**

- pizza\_id
- pizza\_type\_id
- size
- price

## **PIZZATYPE**

- pizza\_type\_id
- name
- category
- ingredients

## **ORDERS**

- order\_id
- order\_date
- order\_time

# PROBLEMS

- 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.
- 
- 
- 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.

1

Local instance MySQL80 ×

Edit View Query Database Server Tools Scripting Help

SQL File 6\* × SQL File 7\* order\_details pizza\_types pizzas orders

stor:... MAS

er objects

anudb

pizzahut

Tables

► order\_details

► orders

► pizza\_types

► pizzas

Views

Stored Procedures

Functions

sys

1 -- Retrieve the total number of orders placed.

2 • select count(order\_id) as total\_orders

3 from orders;

Result Grid | Filter Rows: Export: Wrap Cell Content: □

total_orders
21350

Result 3 ×

2

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 6\* SQL File 7\* × order\_details pizza\_types pizzas orders

SCHEMAS

Filter objects

anudb

pizzahut

- Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

sys

-- Calculate the total revenue generated from pizza sales.

1     -- Calculate the total revenue generated from pizza sales.

2

3 •    SELECT

4       ROUND(SUM(od.quantity \* p.price), 2) AS revenue

5     FROM

6       order\_details AS od

7       JOIN

8       pizzas AS p USING (pizza\_id);

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content: |

revenue
818176.45

3

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 6\* SQL File 7\* SQL File 8\* x SQL File 9\* order\_details pizza\_types pizzas orders

SCHEMAS

Filter objects

anudb  
pizzahut  
Tables  
order\_details  
orders  
pizza\_types  
pizzas  
Views  
Stored Procedures  
Functions

sys

```
1 -- Identify the highest-priced pizza.
2 • SELECT
3     pt.name, p.price
4 FROM
5     pizza_types AS pt
6     JOIN
7     pizzas AS p USING (pizza_type_id)
8 ORDER BY p.price DESC
9 LIMIT 1;
10
11
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	name	price
▶	The Greek Pizza	35.95

Result Grid

# 4

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* × order\_details pizza\_types pizzas orders

Navigator Filter objects

SCHEMAS

Filter objects

anudb

pizzahut

- Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

sys

```
1 -- Identify the most common pizza size ordered.
2
3 • SELECT
4     p.size, COUNT(o.order_details_id) AS total_order
5     FROM
6         order_details AS o
7             JOIN
8                 pizzas AS p USING (pizza_id)
9     GROUP BY p.size
10    ORDER BY total_order DESC
11    LIMIT 1;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid

size	total_order
L	18558

Administration Schemas

# 5

MySQL Workbench  
Local instance MySQL80 ×

Edit View Query Database Server Tools Scripting Help

Vigator SQL File 10\* × order\_details pizza\_types pizzas orders

Filter objects

SCHEMAS

anudb  
**pizzahut**  
Tables  
Views  
Stored Procedures  
Functions  
sys

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2  • SELECT
3      pt.name AS name, SUM(od.quantity) AS quantities
4  FROM
5      pizza_types AS pt
6          JOIN
7      pizzas AS p USING (pizza_type_id)
8          JOIN
9      order_details AS od USING (pizza_id)
10     GROUP BY name
11     ORDER BY quantities DESC
12     LIMIT 5;
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Form Editor

name	quantities
The Barbecue Chicken Pizza	2455
The Classic Deluxe Pizza	2447
The Pepperoni Pizza	2423
The Hawaiian Pizza	2416
The Thai Chicken Pizza	2375

## 6

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- anedb
- pizzahut
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

SQL File 10\* SQL File 11\* x order\_details pizza\_types pizzas orders

```
1 -- Join the necessary tables to find the total quantity of each pizza category orderer
2
3 * SELECT
4     pt.category AS category, SUM(quantity) AS total_quantity
5     FROM
6         pizza_types AS pt
7             JOIN
8                 pizzas AS p USING (pizza_type_id)
9             JOIN
10                order_details USING (pizza_id)
11        GROUP BY category
12
```

Result Grid | Filter Rows: \_\_\_\_\_ | Export: | Wrap Cell Content: |

category	total_quantity
Veggie	11636
Classic	14913
Supreme	11961
Chicken	11068

Administration Schemas

Information

Result Grid

Form Editor

7

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- anudb
- pizzahut**
- Tables
- Views
- Stored Procedures
- Functions

sys

SQL File 12\* × orders order\_details pizza\_types pizzas

```
1 -- Determine the distribution of orders by hour of the day.
2
3 • SELECT
4     COUNT(order_id) AS total_orders, HOUR(order_time) AS hour
5     FROM
6     orders
7     GROUP BY hour;
8
9
10
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	total_orders	hour
1	1231	11
2	2520	12
3	2455	13
4	1472	14
5	1468	15
6	1920	16
7	2336	17
8	2399	18
9	2009	19
10	1642	20
11	1198	21
12	663	22
13	28	23
14	8	10
15	1	9

Administration Schemas

Information

Schema: pizzahut

Result Grid

Form Editor

Field Types

Query Stats

# 8

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 12\* orders order\_details pizza\_types pizzas SQL File 13\* x

SCHEMAS

Filter objects

anudb  
pizzahut  
Tables  
Views  
Stored Procedures  
Functions  
sys

```
1 -- Join relevant tables to find the category-wise distribution of pizzas.
2 • SELECT
3     COUNT(category) AS total_category, category
4 FROM
5     pizza_types
6 GROUP BY category
```

Administration Schemas Information

Schema: pizzahut

Result Grid | Filter Rows: Export: Wrap Cell Content:

total_category	category
6	Chicken
8	Classic
9	Supreme
9	Veggie

Result Grid Form Editor

9

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator: pizza\_types pizzas orders SQL File 17\* SQL File 18\*

SCHEMAS: pizza hut

Filter objects

anubd  
pizzahut  
Tables: order\_details, orders, pizza\_types, pizzas  
Views  
Stored Procedures  
Functions  
SYS

```
1 -- Group the orders by date and calculate the average number of pizzas ordered per day.
2
3 • SELECT
4     ROUND(AVG(pizza), 0) avg_pizza_per_day
5     FROM
6         (SELECT
7             o.order_date AS date, SUM(od.quantity) AS pizza
8             FROM
9                 orders AS o
10            JOIN order_details AS od USING (order_id)
11        GROUP BY date) AS average_pizza
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid

avg_pizza_per_day
138

10

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 8\* pizza\_types pizzas orders order\_details SQL File 7\* ×

SCHEMAS

Filter objects

anudb

pizzahut

- Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

sys

1 -- determine the top 3 most ordered pizza type based on revenue

2

3 \* SELECT

4     pt.name AS pizzas, SUM(od.quantity \* p.price) AS revenue

5    FROM

6       pizzas AS p

7       JOIN

8       pizza\_types AS pt USING (pizza\_type\_id)

9       JOIN

10      order\_details AS od USING (pizza\_id)

11     GROUP BY pizzas

12    ORDER BY revenue DESC

13    LIMIT 3;

Administration Schemas

Information

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid Form Editor

pizzas	revenue
The Thai Chicken Pizza	43557.25
The Barbecue Chicken Pizza	43169.25
The California Chicken Pizza	41475.75

11

MySQL Workbench

Local instance MySQL80 X

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 8\* pizza\_types pizzas orders order\_details

SCHEMAS

- anudb
- pizzahut
  - Tables
    - order\_details
    - orders
    - pizza\_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sys

```
1 -- Calculate the percentage contribution of each pizza type to total revenue.
2 * with mycte as (select round(sum(od.quantity *p.price),2) as rev
3   from order_details as od
4   join pizzas as p using (pizza_id))
5
6
7   select pt.category as category ,
8   round(sum(od.quantity*p.price)/ (select rev from mycte)*100,2) as total_percentage
9   from order_details as od
10  join pizzas as p using (pizza_id)
11  join pizza_types as pt using(pizza_type_id)
12  group by category
13  order by total_percentage;
14
```

Administration Schemas Information

No object selected

Result Grid Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor

category	tot
Veggie	23.67
Chicken	24
Supreme	25.39
Classic	26.95

12

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- anudb
- pizzahut
  - Tables
    - order\_details
    - orders
    - pizza\_types
    - pizzas
  - Views
  - Stored Procedures
  - Functions
- sys

SQL File 8\* pizza\_types pizzas orders order\_details SQL File 9\* ×

```
1 -- Analyze the cumulative revenue generated over time.
2 select order_date ,
3 round(sum(revenue) over (order by order_date ),2) as sum_revenue
4 from
5 ( select date(o.order_date) as order_date , sum(od.quantity* p.price) as revenue
6 from pizzas as p
7 join pizza_types as pt using ( pizza_type_id )
8 join order_details as od using ( pizza_id )
9 join orders as o using ( order_id )
10 group by order_date ) as sales ;
```

Administration Schemas

Information

No object selected

Result Grid | Filter Rows: \_\_\_\_\_ | Export: \_\_\_\_\_ | Wrap Cell Content:

order_date	sum_revenue
2015-01-01	2524.95
2015-01-02	5097.9
2015-01-03	8392.05
2015-01-04	10644.6
2015-01-05	12539.65
2015-01-06	14684
2015-01-07	16835.35
2015-01-08	20448.9
2015-01-09	22833.65

Result 7 ×

Result Grid

Form Editor

Read Only

13

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

Schemas

Filter objects

anudb

pizzahut

- Tables
  - order\_details
  - orders
  - pizza\_types
  - pizzas
- Views
- Stored Procedures
- Functions

sys

SQL File 8\* pizza\_types pizzas orders order\_details SQL File 10\* x

```
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category
2 select pizza , revenue from
3 (select category , pizza , revenue ,
4 rank () over ( partition by category order by revenue desc ) as rn
5 from
6 (select pt.category as category , pt.name as pizza , sum(od.quantity * p.price) as revenue
7 from pizzas as p
8 join pizza_types as pt using(pizza_type_id)
9 join order_details as od using(pizza_id)
10 group by category , name) as a) as b
11 where rn<= 3;
12
```

Administration Schemas

Information

No object selected

Result Grid | Filter Rows: Export: Wrap Cell Content: Result 6 x Read Only

pizza	revenue
The Thai Chicken Pizza	43557.25
The Barbecue Chicken Pizza	43169.25
The California Chicken Pizza	41475.75
The Classic Deluxe Pizza	38045
The Hawaiian Pizza	32182
The Pepperoni Pizza	30260
The Spicy Italian Pizza	34802
The Italian Supreme Pizza	33232.5
The Sicilian Pizza	30786.25
The Four Cheese Pizza	32239.40000000065
The Mexicana Pizza	26608
The Five Cheese Pizza	26473.5

Object Info Session Output

Form Editor Field Types

Thank  
you!