# PIZZA HUT SALES PROJECT SQL

2024

presented by
Md Anwarul Karim

# INTRODUCTION

2024

Welcome to my Pizza Hut sales presentation.  I will analyze and review the Pizza Hut Sales project using SQL to analyze customer orders, sales trends, and profitability within a pizza shop. The project involves querying a database of pizza orders, and menu items to gain insights into customer preferences and peak sales times.

SQL is used to aggregate data, filter records, and perform calculations on sales, allowing analysis of top-selling pizzas, seasonal trends, and customer demographics. Additionally, revenue and profit calculations per order type are performed to optimize menu pricing and marketing strategies.

presented by

**Md Anwarul karim**

# CREATE DATABASE PIZZAHUT

# FIND MY GITHUB TO GET THE FULL PROJECT

SQL File 4* SQL File 5* SQL File 14*

Limit to 1000 rows

```sql
-- CTRL + B then code look beautiful
create database pizzahut;
use pizzahut;

create table orders (
order_id int not null,
order_date date not null,
oder_time time not null,
primary key (order_id)
);

select * from order_details;
select * from orders;
select * from pizza_types;
select * from pizzas;


create table order_details (
order_details_id int not null,
order_id int not null,
pizza_id text not null,
quantity int not null,
primary key (order_details_id)

); -- data taken from excel file, click table then  " Table data import wizard"
```

# 1. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED

2024

```
30    -- 1. Retrieve the total number of orders placed
31 •    select count(order_id) as total_orders from orders;
32
33
34
35
36
```

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

| total_orders |
| --- |
| 21350 |

# 2. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

2024

```
52    -- 2. Calculate the total revenue generated from pizza sales
53 •  SELECT
54        ROUND(SUM(order_details.quantity * pizzas.price),
55                2) AS total_revenue
56    FROM
57        order_details
58            JOIN
59        pizzas ON order_details.pizza_id = pizzas.pizza_id;
60
61
62
63
```

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

| total_revenue |
| --- |
| 817860.05 |

# 3. IDENTIFY THE HIGHEST-PRICED PIZZA

2024

```sql
65
66    -- 3. Identify the highest-priced pizza.
67 •  SELECT
68        pizza_types.name, pizzas.price
69    FROM
70        pizza_types
71            JOIN
72        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
73    ORDER BY pizzas.price DESC
74    LIMIT 1;
75
```

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

| name | price |
|---|---|
| The Greek Pizza | 35.95 |

# 4. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

2024

```
77
78    -- 4. Identify the most common pizza size ordered.
79 •  SELECT
80        pizzas.size,
81        COUNT(order_details.order_details_id) AS order_count
82    FROM
83        pizzas
84            JOIN
85        order_details ON pizzas.pizza_id = order_details.pizza_id
86    GROUP BY pizzas.size
87    ORDER BY order_count DESC
88    LIMIT 3;
89
90
```

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

| size | order_count |
|---|---|
| L | 18526 |
| M | 15385 |
| S | 14137 |

# 5. LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

2024

```
93
94    -- 5. List the top 5 most ordered pizza types along with their quantities.
95 •  SELECT
96        pizza_types.name, SUM(order_details.quantity) AS quantity
97    FROM
98        pizza_types
99            JOIN
100       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
101           JOIN
102       order_details ON order_details.pizza_id = pizzas.pizza_id
103   GROUP BY pizza_types.name
104   ORDER BY quantity DESC
105   LIMIT 5;
106
107
```

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

| 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.

2024

```
112    -- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
113 •  SELECT
114        pizza_types.category,
115        SUM(order_details.quantity) AS quantity
116    FROM
117        pizza_types
118            JOIN
119        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
120            JOIN
121        order_details ON order_details.pizza_id = pizzas.pizza_id
122    GROUP BY pizza_types.category
123    ORDER BY quantity;
124
```

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

| category | quantity |
|---|---|
| Chicken | 11050 |
| Veggie | 11649 |
| Supreme | 11987 |
| Classic | 14888 |

# 7. DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

2024

```
126
127     -- 7. Determine the distribution of orders by hour of the day.
128
129 •   SELECT
130         HOUR(order_time) AS hour, COUNT(order_id) AS order_count
131     FROM
132         orders
133     GROUP BY HOUR(order_time);
134
```

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

| 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 |

# 8. JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

2024

```
136
137     -- 8. Join relevant tables to find the category-wise distribution of pizzas.
138 •   SELECT
139         category, COUNT(name)
140     FROM
141         pizza_types
142     GROUP BY category;
143
```

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

| category | COUNT(name) |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

# 9. GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

2024

```
145
146     -- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
147 •   SELECT
148         ROUND(AVG(quantity), 0) as pizza_order_per_day
149     FROM
150         (SELECT
151             orders.order_date, SUM(order_details.quantity) AS quantity
152         FROM
153             orders
154         JOIN order_details ON orders.order_id = order_details.order_id
155         GROUP BY orders.order_date
156         ORDER BY quantity) AS order_quantity;
157
158
```

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

| pizza_order_per_day |
|---|
| 138 |

# 10. DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

2024

```
158
159     -- 10. Determine the top 3 most ordered pizza types based on revenue.
160 •   SELECT
161         pizza_types.name,
162         SUM(order_details.quantity * pizzas.price) AS revenue
163     FROM
164         pizzas
165             JOIN
166         pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
167             JOIN
168         order_details ON order_details.pizza_id = pizzas.pizza_id
169     GROUP BY 1 -- 1 means select er jeta liksi 'pizza_types.name'
170     ORDER BY revenue DESC
171     LIMIT 3;
172
```

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

2024

```sql
-- 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_revenue
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY 1
ORDER BY revenue DESC;
```

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

| category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

# 12. ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

2024

```
198    -- 12. Analyze the cumulative revenue generated over time.
199 •  SELECT
200        order_date,
201        SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
202    FROM (
203        SELECT
204            orders.order_date,
205            SUM(order_details.quantity * pizzas.price) AS revenue
206        FROM order_details
207        JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id
208        JOIN orders ON order_details.order_id = orders.order_id
209        GROUP BY orders.order_date
210    ) AS sales;
211
```

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

| 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 |
| 2015-01-06 | 14358.5 |
| 2015-01-07 | 16560.7 |
| 2015-01-08 | 19399.05 |
| 2015-01-09 | 21526.4 |
| 2015-01-10 | 23990.350000000002 |

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

2024

```sql
-- 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 profit
    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 category_sales
) AS ranked_sales
WHERE profit <= 3;
```

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

| 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 |

karim247live@gmail.com

# THANK YOU!

Thank you for your attention to my presentation. If you have any questions or would like to discuss the findings in more detail, please don't hesitate to reach out. I appreciate your continued support and partnership.