



Pizza Sales report and future goals

TOWARDS NEW HORIZONS: SALES REPORT AND UPCOMING TARGETS

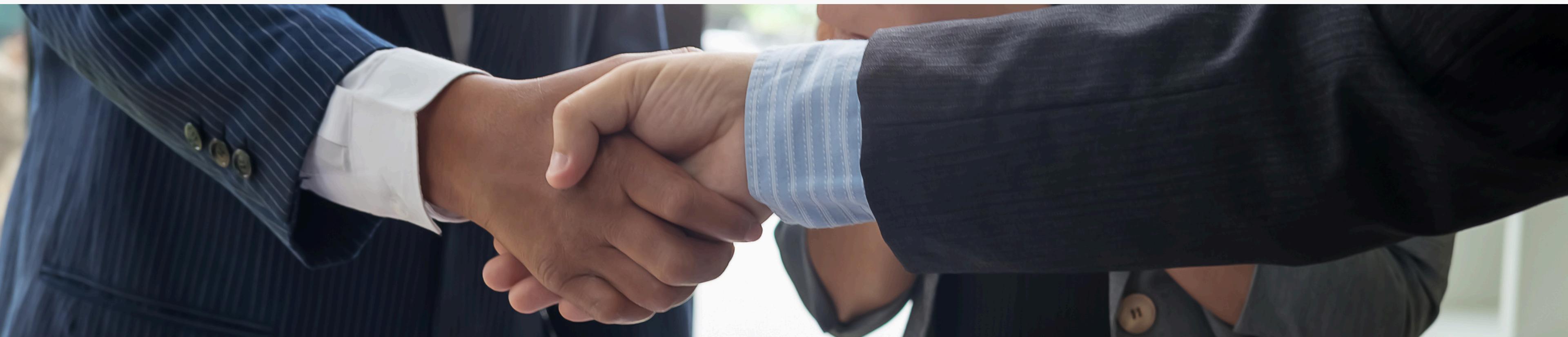


Introduction to the sales report

This script is designed to create a comprehensive sales report for a pizza restaurant using SQL. The report includes various analyses such as total orders, revenue, most popular pizzas, and order distribution over time. The following queries are used to extract and aggregate data from the PIZZA_HUT database.

Sales report methodology

The following methodology outlines the steps and SQL queries used to generate a comprehensive pizza sales report for the PIZZA_HUT database. The report aims to provide insights into sales performance, popular products, revenue distribution, and ordering patterns.



Step 1: Create and Select Database

Step 1: Create and Select Database

CREATE DATABASE PIZZA_HUT;

Use Database: The script switches to the PIZZA_HUT database to perform subsequent operations.

USE PIZZA_HUT;

Total Orders: Count the total number of orders placed

```
-- 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(quantity * price), 2)  
FROM  
    pizzas AS p  
    INNER JOIN  
    order_details AS od ON od.pizza_id = p.pizza_id;
```

Result Grid	
	Revenue
▶	817860.05

Identify the highest-priced pizza.

```
SELECT pt.name AS Name, price FROM pizzas AS p  
INNER JOIN pizza_types AS pt on pt.pizza_type_id = p.pizza_type_id  
ORDER BY PRICE DESC  
LIMIT 1;
```

The screenshot shows a MySQL Workbench interface with a query editor window. The query editor contains the SQL code provided above. Below the code is a result grid titled "Result Grid". The result grid has two columns: "Name" and "price". A single row is displayed, showing "The Greek Pizza" in the "Name" column and "35.95" in the "price" column. There is also a small arrow icon next to the "Name" column header. At the top of the result grid, there are buttons for "Result Grid" and "Filter Rows:", along with a refresh icon.

	Name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

```
SELECT COUNT(order_details_id), size FROM pizzas AS P
INNER JOIN order_details as od on od.pizza_id = p.pizza_id
GROUP BY size
ORDER BY SUM(quantity) DESC;
```

Result Grid |

	Orders	size
▶	18526	L
	15385	M
	14137	S
	544	XL
	28	XXL

List the top 5 most ordered pizza types along with their quantities.

```
SELECT name,pt.pizza_type_id Pizza_id, sum(quantity) Quantity FROM order_details as od
INNER JOIN pizzas AS p on p.pizza_id = od.pizza_id
INNER JOIN pizza_types AS pt on pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.pizza_type_id, name
ORDER BY SUM(quantity) DESC
LIMIT 5;
```

Result Grid | Filter Rows: _____ | Export

	name	Pizza_id	Quantity
▶	The Classic Deluxe Pizza	dassic_dlx	2453
	The Barbecue Chicken Pizza	bbq_ckn	2432
	The Hawaiian Pizza	hawaiian	2422
	The Pepperoni Pizza	pepperoni	2418
	The Thai Chicken Pizza	thai_ckn	2371

Intermediate:

Join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT category, SUM(quantity) AS Quantity FROM pizza_types AS pt
INNER JOIN pizzas AS P ON p.pizza_type_id = pt.pizza_type_id
INNER JOIN order_details AS od on od.pizza_id = p.pizza_id
GROUP BY category
ORDER BY SUM(quantity) DESC;
```

Result Grid | Filter Rows

	category	Quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

- ▶

```
SELECT HOUR(order_time) HOUR_TIME, COUNT(order_id) ORDER_CNT FROM orders  
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

	HOUR_TIME	ORDER_CNT
▶	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

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

```
SELECT category, COUNT(name) cnt_ordered FROM pizza_types  
GROUP BY category;
```

Result Grid | Filter Rows:

	category	cnt_ordered
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

Group the orders by date and calculate the average number of pizzas ordered per day.

```
9 •   SELECT ROUND(AVG(quantity),2) AS AVG_SALE FROM
1   (SELECT o.order_date, SUM(od.quantity) AS quantity FROM orders AS o
2   JOIN order_details AS od ON od.order_id = o.order_id
3   GROUP BY order_date) AS order_quantity;
```

Result Grid	
	AVG_Sale
▶	138.47

Determine the top 3 most ordered pizza types based on revenue.

```
• SELECT name, SUM(quantity * price) Rev FROM pizza_types AS pt
  INNER JOIN pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
  INNER JOIN order_details AS od ON od.pizza_id = p.pizza_id
  GROUP BY name
  ORDER BY Rev DESC
  LIMIT 3;
```

Result Grid | Filter Rows:

	name	Rev
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

Advanced: Calculate the percentage contribution of each pizza type to total revenue.

```
5 • SELECT pt.category, CONCAT(ROUND(SUM(quantity*price) / (SELECT
5           ROUND(SUM(quantity * price), 2)
7
8           FROM
9               pizzas AS p
9
10          INNER JOIN
11              order_details AS od ON od.pizza_id = p.pizza_id) * 100,2),"%")AS Rev
12
13           FROM pizza_types AS pt
14
15          INNER JOIN pizzas AS p ON p.pizza_type_id = pt.pizza_type_id
16
17          INNER JOIN order_details AS od ON od.pizza_id = p.pizza_id
18
19          GROUP BY pt.category
20
21          ORDER BY Rev DESC;
```

Result Grid | Filter Rows:

	category	Rev
▶	Classic	26.91%
	Supreme	25.46%
	Chicken	23.96%
	Veggie	23.68%

Analyze the cumulative revenue generated over time.

- ```
SELECT order_date, SUM(Rev) OVER(ORDER BY order_date) AS CUM_Rev
 FROM
 (SELECT order_date, SUM(quantity * price) Rev FROM pizzas AS P
 INNER JOIN order_details AS od ON od.pizza_id = p.pizza_id
 INNER JOIN orders AS o ON o.order_id = od.order_id
 GROUP BY order_date) AS sales;
```

| Result Grid |            |                    |
|-------------|------------|--------------------|
|             | order_date | CUM_Rev            |
| ▶           | 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 | 23990.35000000002  |
|             | 2015-01-11 | 25862.65           |
|             | 2015-01-12 | 27781.7            |
|             | 2015-01-13 | 29831.30000000003  |
|             | 2015-01-14 | 32358.70000000004  |
|             | 2015-01-15 | 34343.5000000001   |
|             | 2015-01-16 | 36937.6500000001   |
|             | 2015-01-17 | 39001.7500000001   |
|             | 2015-01-18 | 40978.60000000006  |
|             | 2015-01-19 | 43365.75000000001  |

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

```
98 • SELECT name, revenue, Rank_
99 (SELECT category, name, revenue,
10 RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS Rank_
11 FROM
12 (SELECT pt.category, pt.name, SUM(od.quantity * p.price) AS revenue
13 FROM pizza_types AS pt
14 JOIN pizzas AS p ON pt.pizza_type_id = p.pizza_type_id
15 JOIN order_details AS od ON od.pizza_id = p.pizza_id
16 GROUP BY pt.category, pt.name) AS a) AS b
17 WHERE Rank_ <=3;
```

|   | name                         | revenue           | Rank_ |
|---|------------------------------|-------------------|-------|
| ▶ | The Thai Chicken Pizza       | 43434.25          | 1     |
|   | The Barbecue Chicken Pizza   | 42768             | 2     |
|   | The California Chicken Pizza | 41409.5           | 3     |
|   | The Classic Deluxe Pizza     | 38180.5           | 1     |
|   | The Hawaiian Pizza           | 32273.25          | 2     |
|   | The Pepperoni Pizza          | 30161.75          | 3     |
|   | The Spicy Italian Pizza      | 34831.25          | 1     |
|   | The Italian Supreme Pizza    | 33476.75          | 2     |
|   | The Sicilian Pizza           | 30940.5           | 3     |
|   | The Four Cheese Pizza        | 32265.70000000065 | 1     |
|   | The Mexicana Pizza           | 26780.75          | 2     |
|   | The Five Cheese Pizza        | 26066.5           | 3     |