

PIZZA SALES ANALYSIS

M Y S Q L



OVERVIEW

Here we are analyzing the pizza sales by using MYSQL and finding insightful information that will help to make the business decisions and also for strategic planning.

and we have different questions about the sales report that need to be analyzed one by one. We have a total 4 tables here for the analysis.

Tool: MYSQL

We have a total 4 tables of this whole pizza sales data

- 1.order_details
- 2.orders
- 3.pizzas
- 4.pizza_types

1. Retrieve the total number of orders placed.

SQL Query

```
Query 1 Pizza_sales_report* x
1 • select *from order_details;
2 • select *from orders;
3 • select *from pizza_types;
4 • select *from pizzas;
5
6 -- 1. Retrieve the total number of orders placed.
7 • select count(order_id) as total_orders from orders;
```

Output

total_orders
21350

2. Calculate the total revenue generated from pizza sales.

SQL Query

```
Query 1 Pizza_sales_report* x
8
9      -- 2. Calculate the total revenue generated from pizza sales.
10 •   SELECT
11       round(SUM(pizzas.price * order_details.quantity),2) AS Total_Revenue
12   FROM
13       pizzas
14   JOIN
15       order_details ON pizzas.pizza_id = order_details.pizza_id;
16
```

Output

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	□
Total_Revenue					
817860.05					

3. Identify the highest-priced pizza.

SQL Query

```
Query 1 Pizza_sales_report* 
17      -- 3.Identify the highest-priced pizza.
18 •   SELECT
19       pizzas.price AS highest_price, pizza_types.name
20   FROM
21       pizzas
22       JOIN
23       pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
24   ORDER BY highest_price DESC
25   LIMIT 1;
```

Output

Result Grid		Filter Rows:	Export:	Wrap Cell Content:

	highest_price	name
▶	35.95	The Greek Pizza

3. Identify the highest-priced pizza.

SQL Query

```
Query 1 Pizza_sales_report* 
17      -- 3.Identify the highest-priced pizza.
18 •   SELECT
19       pizzas.price AS highest_price, pizza_types.name
20   FROM
21       pizzas
22       JOIN
23           pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
24   ORDER BY highest_price DESC
25   LIMIT 1;
```

Output

Result Grid		Filter Rows:	Export:	Wrap Cell Content:

	highest_price	name
▶	35.95	The Greek Pizza

4. Identify the most common pizza size ordered.

SQL Query

```
Query 1 Pizza_sales_report* x
26
27      -- 4. Identify the most common pizza size ordered.
28 •  SELECT
29      COUNT(order_details.order_details_id) AS order_count,
30      pizzas.size
31  FROM
32      orders
33          JOIN
34      order_details ON order_details.order_id = orders.order_id
35          JOIN
36      pizzas ON pizzas.pizza_id = order_details.pizza_id
37  GROUP BY pizzas.size
38  ORDER BY order_count DESC
39  LIMIT 1;
```

Output

	order_count	size
▶	18526	L

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

SQL Query

```
Query 1 Pizza_sales_report* x
41  -- 5. List the top 5 most ordered pizza types along with their quantities.
42 • SELECT
43      pizza_types.name, SUM(order_details.quantity) AS ord_qnt
44  FROM
45      pizzas
46      JOIN
47          pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
48      JOIN
49          order_details ON order_details.pizza_id = pizzas.pizza_id
50  GROUP BY pizza_types.name
51  ORDER BY ord_qnt DESC
52  LIMIT 5;
```

Output

JOIN	
Result Grid	
name	ord_qnt
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.

SQL Query

```
Query 1 Pizza_sales_report* x
55
56    -- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
57 • SELECT
58     pizza_types.category,
59     SUM(order_details.quantity) AS ord_qty
60   FROM
61     pizzas
62       JOIN
63     order_details ON pizzas.pizza_id = order_details.pizza_id
64       JOIN
65     pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
66   GROUP BY category
67   ORDER BY ord_qty;
```

Output

Result Grid		
	category	ord_qty
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

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

SQL Query

```
Query 1 Pizza_sales_report* 
70      -- 2. Determine the distribution of orders by hour of the day.
71 •  SELECT
72      HOUR(order_time) AS hr, COUNT(order_id) AS ord_count
73  FROM
74      orders
75  GROUP BY hr
76  ORDER BY ord_count ASC;
77
```

Output

	hr	ord_count
▶	9	1
	10	8
	23	28
	22	663
	21	1198
	11	1231
	15	1468
	14	1472
	20	1642
	16	1920
	19	2009
	17	2226

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

SQL Query

```
Query 1 Pizza_sales_report* ×
79      -- 3.Join relevant tables to find the category-wise distribution of pizzas.
80 •   SELECT
81       category, COUNT(pizza_type_id) AS count
82   FROM
83       pizza_types
84 GROUP BY category
85 ORDER BY count;
```

Output

Result Grid		
	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

9. Group the orders by date and calculate the average number of pizzas ordered (Quantity) per day.

SQL Query

```
Query 1 Pizza_sales_report* 
88      -- 4. Group the orders by date and calculate the average number of pizzas ordered per day.
89 •  SELECT
90      AVG(ord_qty.qnty) AS avg_qty_perday
91  FROM
92  (SELECT
93      orders.order_date AS dates,
94      SUM(order_details.quantity) AS qnty
95  FROM
96      order_details
97  JOIN orders ON order_details.order_id = orders.order_id
98  GROUP BY dates
99  ORDER BY qnty) AS ord_qty;
```

Output

avg_qty_perday
138.4749

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

SQL Query

```
Query 1 Pizza_sales_report* x
100
101    -- 5. Determine the top 3 most ordered pizza types based on revenue.
102 •  SELECT
103      pizza_types.name,
104      SUM(pizzas.price * order_details.quantity) AS revenue
105  FROM
106    pizzas
107    JOIN
108      order_details ON pizzas.pizza_id = order_details.pizza_id
109    JOIN
110      pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
111  GROUP BY pizza_types.name
112  ORDER BY revenue DESC
113  LIMIT 3;
```

Output

	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 category to total revenue.

SQL Query

```
Query 1 Pizza_sales_report* ×
115      -- 1. Calculate the percentage contribution of each pizza category to total revenue.
116 •   SELECT
117      pizza_types.category AS cat,
118      ROUND((SUM(pizzas.price * order_details.quantity) / (SELECT
119                  SUM(pizzas.price * order_details.quantity) AS total_sales
120              FROM
121              pizzas JOIN
122              order_details ON pizzas.pizza_id = order_details.pizza_id JOIN
123              pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id) * 100),
124          2) AS revenue
125  FROM
126      pizzas JOIN
127      pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id JOIN
128      order_details ON pizzas.pizza_id = order_details.pizza_id
129  GROUP BY cat
130  ORDER BY revenue;
```

Output

	cat	revenue
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91

12. Analyze the cumulative revenue generated over time.

SQL Query

```
Query 1 Pizza_sales_report* x
132      -- 2 Analyze the cumulative revenue generated over time.
133 •   select order_date, sum(revenue) over(order by order_date) as cum_rev
134     from
135     (select orders.order_date, round(sum(order_details.quantity * pizzas.price),2) as revenue from
136       order_details join orders on
137         order_details.order_id = orders.order_id
138       join pizzas on
139         pizzas.pizza_id = order_details.pizza_id group by order_date order by revenue) as total_sales;
```

Output

order_date	cum_rev
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

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

SQL Query

```
Query 1 Pizza_sales_report* x
143      -- 3 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
144 •  select category, name, revenue,rk
145    from
146    (select category, name, revenue,
147     rank() over(partition by category order by revenue desc) as rk
148    from
149    (select pizza_types.name, pizza_types.category, sum(pizzas.price*order_details.quantity) as revenue from
150      pizzas join order_details on
151        order_details.pizza_id = pizzas.pizza_id
152      join pizza_types on
153        pizza_types.pizza_type_id = pizzas.pizza_type_id
154      group by pizza_types.name, pizza_types.category) as a) as b
155    where rk<=3;
```

Output

	category	name	revenue	rk
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Classic	The Classic Deluxe Pizza	38180.5	1
	Classic	The Hawaiian Pizza	32273.25	2



THANK YOU

Special Thanks to @WSCUBETECH @Ayushi Jain