

# PIZZA SALES ANALYSIS







# PIZZA SALES ANALYSIS (MYSQL)

**This delicious deep-dive explores pizza sales data — from total orders to top sellers, hourly cravings, and revenue-rich pies. Using SQL, I sliced through the data to answer interesting questions. From basic stats to advanced revenue breakdowns, this report is a full-stack feast — perfect for anyone hungry for insights!**



# The total number of orders placed

```
1  -- Retrieve the total number of orders placed.  
2  
3  •  select count(order_id) as total_orders from orders;
```

100%



53:3

Result Grid



Filter Rows:



Search

Export:



total\_orders ^

21350








# The total revenue generated from pizza sales

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  • SELECT
4      ROUND(SUM(order_details.quantity * pizzas.price),
5              2) AS total_revenue
6  FROM
7      order_details
8      JOIN
9      pizzas ON pizzas.pizza_id = order_details.pizza_id;
10
```

100% 56:9

Result Grid   Filter Rows:  Export: 

total_revenue
817860.05





# Identify the highest-priced pizza

```
1  -- Identify the highest-priced pizza.
2
3  SELECT
4      name, price
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
9  ORDER BY price DESC
10 LIMIT 1;
```

100%



9:10

Result Grid



Filter Rows:



Export:



Fetch rows:



name	price
The Greek Pizza	35.95





# Identify the most common pizza size ordered

```
1  -- Identify the most common pizza size ordered
2
3  •  SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS size_order_count
6  FROM
7      pizzas
8      JOIN
9      order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY size_order_count DESC
12 LIMIT 1;
13
```

100%

1:13

Result Grid



Filter Rows:



Search

Export:



Fetch rows:



size	size_order_count
------	------------------

L	18526
---	-------





# Top 5 most ordered pizza types along with their quantities

```
3 • SELECT
4     pizza_types.name,
5     SUM(order_details.quantity) AS pizza_type_quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY pizza_type_quantity DESC
14 LIMIT 5;
15
16
```

100%



8:3

Result Grid



Filter Rows:



Search

Export:



Fetch rows:



name	pizza_type_quant...
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371





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

```
3 • SELECT
4     pizza_types.category,
5     SUM(order_details.quantity) AS sum_pizza_category
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.category
13 ORDER BY sum_pizza_category DESC;
14
15
```

100% 34:13

Result Grid



Filter Rows:

Search

Export:



category	sum_pizza_categ...
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050





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

```
3 • select hour(order_time) as hour, count(order_id) as orders_by_hour
4   from orders
5   group by hour(order_time);
6
```

100% 1:6

Result Grid



Filter Rows:



Search

Export:



hour	orders_by_hour
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





# category-wise distribution of pizzas.

```
3 • select category, count(*) as category_Distribution
4   from pizza_types
5   group by category;
```

100%



25:3

Result Grid



Filter Rows:



Search

Export:



category	category_Distributi...
----------	------------------------

Chicken	6
---------	---

Classic	8
---------	---

Supreme	9
---------	---

Veggie	9
--------	---








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

```
3 • SELECT
4     ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
5 FROM
6     (SELECT
7         orders.order_date, SUM(order_details.quantity) AS quantity
8     FROM
9         orders
10    JOIN order_details ON orders.order_id = order_details.order_id
11   GROUP BY order_date) AS order_quantity;
```

100% 1:12

Result Grid   Filter Rows:  Export: 

avg_pizza_ordered_per_day
138





# Top 3 most ordered pizza types based on revenue

```
3 • SELECT
4     pizza_types.name,
5     SUM(order_details.quantity * pizzas.price) AS revenue
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY revenue DESC
14 LIMIT 3;
```

100%



9:14

Result Grid



Filter Rows:



Search

Export:



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5





# The percentage contribution of each pizza type to total revenue

```
3 • SELECT
4     pizza_types.category,
5     ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
6         SUM(order_details.quantity * pizzas.price)
7         FROM
8             order_details
9             JOIN
10                pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100,
11         2) AS revenue
12 FROM
13     pizza_types
14     JOIN
15     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
16     JOIN
17     order_details ON order_details.pizza_id = pizzas.pizza_id
18 GROUP BY pizza_types.category
19 ORDER BY revenue DESC;
20
```

100% 24:19

Result Grid Filter Rows: Search Export:

	category	revenue
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68





# Cumulative revenue generated over time

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

100% 38:10

Result Grid



Filter Rows:



Search

Export:



order_date	revenue
2015-01-01	2713.85000000000004
2015-01-02	2731.89999999999996
2015-01-03	2662.39999999999996
2015-01-04	1755.45000000000003
2015-01-05	2065.95
2015-01-06	2428.95
2015-01-07	2202.20000000000003
2015-01-08	2838.34999999999995
2015-01-09	2127.35000000000004
2015-01-10	2463.95
2015-01-11	1872.30000000000002
2015-01-12	1919.05000000000002
2015-01-13	2049.60000000000004
2015-01-14	2527.39999999999996
2015-01-15	1984.80000000000002
2015-01-16	2594.15
2015-01-17	2064.10000000000004
2015-01-18	1976.85000000000001
2015-01-19	2387.14999999999996
2015-01-20	2397.90000000000005





# Top 3 most ordered pizza types based on revenue for each pizza category

```
3 • SELECT
4     category,
5     name,
6     revenue
7 FROM (
8     SELECT
9         category,
10        name,
11        revenue,
12        RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS ranks
13 FROM (
14     SELECT
15         pizza_types.category,
16         pizza_types.name,
17         SUM(order_details.quantity * pizzas.price) AS revenue
18 FROM pizza_types
19 JOIN pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
20 JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
21 GROUP BY pizza_types.category, pizza_types.name
22 ) AS a
23 ) AS b
24 WHERE ranks <= 3;
```

100% 7:23

Result Grid Filter Rows: Search Export:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5





THANK YOU!

