



PIZZA SALES

U S I N G M Y S Q L

Data Analysis





PIZZA SALES

ABOUT ME

I am an aspiring data analyst with a passion for transforming data into meaningful insights. Through self-learning and guided projects, I have developed skills in SQL, Python, and Excel. Recently, I completed a MySQL data analysis project, where I gained practical experience in data extraction, cleaning, and analysis and solved questions related to Pizza Sales.





PIZZA SALES

WHY SQL?

SQL is essential for data analysts because it enables efficient retrieval, manipulation, and analysis of data from databases. It allows analysts to query large datasets, clean and prepare data, and perform complex operations, making it a fundamental tool for deriving insights and supporting data-driven decisions.





BASIC

1. Retrieve the total number of orders placed.

```
16 -- retrieve the total number of orders placed.
17 • select count(order_id) as Total_no_of_orders from orders;
18
19
```

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

Total_no_of_orders
21350



BASIC

2. Calculate the total revenue generated from pizza sales.

```
22 •   SELECT
23   ⚒     ROUND(SUM(order_details.quantity * pizzas.price),
24           2) AS Total_sales
25   FROM
26     order_details
27   JOIN
28     pizzas ON pizzas.pizza_id = order_details.pizza_id
29
```

Total_sales
817860.05



BASIC

3. Identify the highest-priced pizza.

```
32
33 •  SELECT
34      pizza_types.name, pizzas.price
35  FROM
36      pizza_types
37      JOIN
38      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
39  ORDER BY pizzas.price DESC
40  LIMIT 1;
```

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

	name	price
▶	The Greek Pizza	35.95



BASIC

4. Identify the most common pizza size ordered.

```
44 • SELECT
45     pizzas.size,
46     COUNT(order_details.order_details_id) AS order_count
47 FROM
48     pizzas
49     JOIN
50         order_details ON pizzas.pizza_id = order_details.pizza_id
51 GROUP BY pizzas.size
52 ORDER BY order_count DESC;
--
```

Result Grid | | Export: | Wrap Cell Content:

size	order_count
L	18526
M	15385
S	14137
XL	544
XXL	28



BASIC

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

```
57 •  SELECT
58      pizza_types.name, SUM(order_details.quantity) AS quantity
59  FROM
60      pizza_types
61      JOIN
62          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
63      JOIN
64          order_details ON order_details.pizza_id = pizzas.pizza_id
65  GROUP BY pizza_types.name
66  ORDER BY quantity DESC
67  LIMIT 5;
```

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

A close-up photograph of a person's hand holding a wooden spoon and stirring a pan of pasta. The pasta appears to be fettuccine with cherry tomatoes and possibly some meat or bacon. The pan is dark-colored and sits on a stove.

INTERMEDIATE

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

```
72 •  SELECT
73      pizza_types.category,
74      SUM(order_details.quantity) AS quantity
75  FROM
76      pizza_types
77      JOIN
78      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
79      JOIN
80      order_details ON order_details.pizza_id = pizzas.pizza_id
81  GROUP BY pizza_types.category
82  ORDER BY quantity DESC;
```

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

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



INTERMEDIATE

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

```
87 •  SELECT
88      HOUR(order_time), COUNT(order_id) AS order_count
89  FROM
90      orders
91  GROUP BY HOUR(order_time);
```

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

hour(order_time)	count(order_id)
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

A close-up photograph of a person's hand holding a wooden spoon and stirring a pan of pasta. The pasta appears to be fettuccine, mixed with cherry tomatoes and some herbs or meat. The pan is dark-colored and sits on a stove.

INTERMEDIATE

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

```
99
100 •   select category , count(name) from pizza_types
101      group by category;
102
103
104
```

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

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



INTERMEDIATE

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

```
104      -- Group the orders by date and calculate the average number of pizzas ordered per day.  
105  
106 •   select round(avg(quantity),0) as average_pizza from  
107     (select orders.order_date, sum(order_details.quantity) as quantity  
108       from orders join order_details  
109         on orders.order_id = order_details.order_id  
110       group by orders.order_date) as order_quantity ;  
111
```

	average_pizza
▶	138



INTERMEDIATE

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

```
117 • select pizza_types.name,  
118     sum(order_details.quantity * pizzas.price) as revenue  
119     from pizza_types join pizzas  
120     on pizzas.pizza_type_id = pizzas.pizza_type_id  
121     join order_details  
122     on order_details.pizza_id = pizzas.pizza_id  
123     group by pizza_types.name order by revenue desc limit 3;  
124  
125  
126
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	name	revenue			
▶	The Vegetables + Vegetables Pizza	817860.049999993			
	The Spinach and Feta Pizza	817860.049999993			
	The Spinach Pesto Pizza	817860.049999993			

ADVANCED

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



```
131 •   SELECT
132     pizza_types.category,
133     (SUM(order_details.quantity * pizzas.price) /
134      (
135          SELECT
136              ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
137          FROM
138              order_details
139              JOIN
140                  pizzas ON pizzas.pizza_id = order_details.pizza_id
141          )
142      ) * 100 AS revenue
143  FROM
144      pizza_types
145      JOIN
146          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
147      JOIN
148          order_details ON order_details.pizza_id = pizzas.pizza_id
149  GROUP BY
150      pizza_types.category
151  ORDER BY
152      revenue DESC;
```

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

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577

ADVANCED

2.Analyze the cumulative revenue generated over time.



```
160  
161 •  select order_date,  
162      sum(revenue) over (order by order_date) as cumulative_revenue  
163      from  
164      (select orders.order_date,  
165          sum(order_details.quantity * pizzas.price) as revenue  
166          from order_details join pizzas  
167            on order_details.pizza_id = pizzas.pizza_id  
168          join orders  
169            on orders.order_id = order_details.order_id  
170          group by orders.order_date) as sales;  
171  
172
```

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

	order_date	cumulative_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
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.300000000003

ADVANCED

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



```
177 •   select name, revenue from
178   (select category, name, revenue,
179    rank() over(partition by category order by revenue desc) as rn
180    from
181   (select pizza_types.category, pizza_types.name,
182    sum((order_details.quantity) * pizzas.price) as revenue
183    from pizza_types
184    join pizzas
185    on pizza_types.pizza_type_id = pizzas.pizza_type_id
186    join order_details
187    on order_details.pizza_id = pizzas.pizza_id
188    group by pizza_types.category, pizza_types.name) as a) as b
189   where rn<=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



PIZZA SALES
DATA ANALYSIS

THANK YOU

