

SQL PROJECT ON PIZZA SALES

[Home](#)

[About](#)

[Contact](#)



ABOUT

Hello, my name is Jasmuddin Ansari.

In this project, I have used SQL to analyze pizza sales data and uncover important insights.

The purpose of the project is to understand customer preferences, sales trends, and business performance.

Using SQL queries, I explored key areas such as the most popular pizzas, peak order hours, top-selling categories, and revenue patterns. These insights can help in making better business decisions and improving sales strategies.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
2 • select count(order_id) from orders;
```

```
3
```

Result Grid		Filter Row
	count(order_id)	04
▶	21350	

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

```
3 •   SELECT
4     ROUND(SUM(order_details.quantity * pizzas.price),
5           2) AS total_sales
6   FROM
7     order_details
8   JOIN
9     pizzas ON pizzas.pizza_id = order_details.pizza_id
```

	total_sales
▶	817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

Contact

SELECT

```
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

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

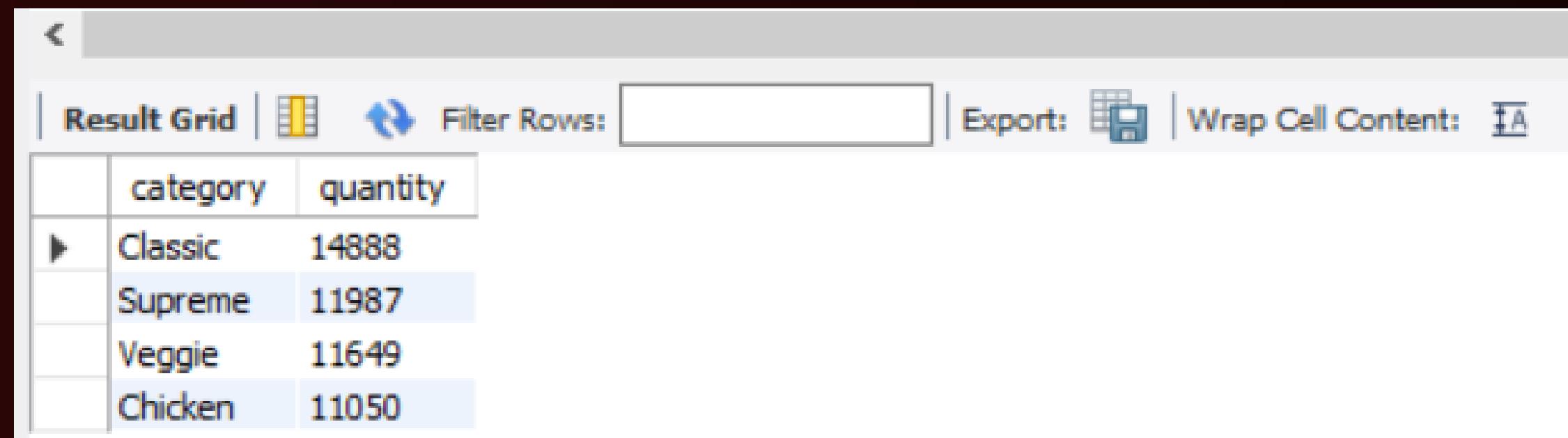
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
3 • select pizza_types.name,  
4     sum(order_details.quantity) as quantity  
5     from pizza_types join pizzas  
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7     join order_details  
8     on order_details.pizza_id = pizzas.pizza_id  
9     group by pizza_types.name order by quantity desc limit 5;
```

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
3 •   select pizza_types.category,  
4     sum(order_details.quantity) as quantity  
5   from pizza_types join pizzas  
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id  
7   join order_details  
8     on order_details.pizza_id = pizzas.pizza_id  
9   group by pizza_types.category order by quantity desc;
```



The screenshot shows a MySQL Workbench interface with a result grid. The grid has two columns: 'category' and 'quantity'. The data is as follows:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
2 • SELECT  
3     HOUR(order_time), COUNT(order_id) AS order_count  
4 FROM  
5     orders  
6 GROUP BY HOUR(order_time);
```

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

	HOUR(order_time)	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

Result 1 ×

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

```
select category, count(name) from pizza_types  
group by category;
```

Result Grid | Filter Rows:

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

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

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

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

avg_pizza_ordered_per_day
138

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

```
3 •      select pizza_types.name,  
4          sum(order_details.quantity * pizzas.price) as revenue  
5      from pizza_types join pizzas  
6      on pizzas.pizza_type_id = pizza_types.pizza_type_id  
7      join order_details  
8      on order_details.pizza_id = pizzas.pizza_id  
9      group by pizza_types.name order by revenue desc limit 3;
```

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

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
3 •  select pizza_types.category,  
4   round(sum(order_details.quantity*pizzas.price) / (SELECT  
5     ROUND(SUM(order_details.quantity * pizzas.price),  
6       2) AS total_sales  
7   FROM  
8     order_details  
9     JOIN  
10    pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2)  as revenue  
11  from pizza_types join pizzas  
12  on pizza_types.pizza_type_id = pizzas.pizza_type_id  
13  join order_details  
14  on order_details.pizza_id = pizzas.pizza_id  
15  group by pizza_types.category order by revenue desc;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

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

Result Grid | Filter Rows: Export:

	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

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

```
3  (select category, name, revenue,
4   rank() over(partition by category order by revenue desc) as rn
5   from
6   (SELECT
7     pizza_types.category,
8     pizza_types.name,
9     SUM(order_details.quantity * pizzas.price) AS revenue
10    FROM
11      pizza_types
12    JOIN
13      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
14    JOIN
15      order_details ON order_details.pizza_id = pizzas.pizza_id
16    GROUP BY
17      pizza_types.category, pizza_types.name) as a) as b
18  where rn <=3;
```

	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



Home

About

Contact

THANK YOU