

Analysis of pizza sales using SQL

WELCOME TO MY PROJECT

by Harshit chugh

Start Now

READ
NOW





HELLO

A LITTLE ABOUT ME

Hii everyone i'm Harshit chugh a student studying in data science .

I'm particularly interested in using data to uncover insights and trends, and in this project I've been exploring this through SQL.

In this project I utilize sql queries to solve the questions that were related to pizza sales

PROJECT OVERVIEW

Slicing into Pizza Sales Data

- * This project analyzes pizza sales data using SQL to gain a better understanding of sales performance, customer preferences, and key business metrics.



- * The data used for this analysis is a company's sales database
- * The main goals of this project were to identify top-selling pizza types, analyze sales trends over time, determine peak sales periods





|||||

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```
select count(order_id) as total_orders  
from orders;
```

Result Grid |  

	total_orders
▶	21350

IDENTIFY THE HIGHEST-PRICED PIZZA

```
select pt.name,  
       p.price  
  from pizza_types as pt  
  join pizzas as p  
    on pt.pizza_type_id = p.pizza_type_id  
order by p.price desc  
limit 1;
```

Result Grid		Filter Rows:
	name	price
▶	The Greek Pizza	35.95



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

```
select round(avg(quantity),0)
from
(select o.order_date,
sum(od.quantity) as quantity
from orders as o
join order_details as od
on o.order_id = od.order_id
group by o.order_date) as order_quantity;
```

	round(avg(quantity),0)
▶	138



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

```
select pt.name ,  
       sum(od.quantity) as quantity  
  from pizza_types as pt  
  join pizzas as p  
    on pt.pizza_type_id = p.pizza_type_id  
  join order_details as od  
    on p.pizza_id = od.pizza_id  
 group by 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
	The Thai Chicken Pizza	2371

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED

```
select p.size ,  
       count(o.order_id) as total_count  
  from pizzas as p  
  join order_details as ot  
    on p.pizza_id = ot.pizza_id  
  join orders as o  
    on o.order_id = ot.order_id  
 group by p.size  
order by total_count desc  
limit 1 ;
```

	size	total_count
▶	L	18526





CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
select round(sum(o.quantity*p.price),2) as revenue  
from order_details as o  
join pizzas as p  
on o.pizza_id = p.pizza_id;
```

Result Grid	
	revenue
▶	817860.05

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

```
select pt.category,  
       sum(od.quantity) as quantity  
  from pizza_types as pt  
  join pizzas as p  
    on p.pizza_type_id = pt.pizza_type_id  
  join order_details as od  
    on od.pizza_id = p.pizza_id  
 group by pt.category  
order by quantity desc;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
select pt.name,  
sum(od.quantity*p.price) as revenue  
from pizza_types as pt  
join pizzas as p  
  on pt.pizza_type_id = p.pizza_type_id  
join order_details as od  
  on p.pizza_id = od.pizza_id  
group by pt.name  
order by revenue desc  
limit 3;
```

Result Grid | Filter Rows: _____

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



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
select hour(order_time) as hours ,count(order_id)  
from orders  
group by hours;
```

hours	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
9	1



CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

```
select pt.category ,  
round((sum(p.price*od.quantity)/  
      (select sum(p.price*od.quantity)  
       from pizzas as p  
       join order_details as od  
         on p.pizza_id = od.pizza_id))*100,2) as revenue  
from pizzas as p  
join pizza_types as pt  
  on p.pizza_type_id = pt.pizza_type_id  
join order_details as od  
  on od.pizza_id = p.pizza_id  
group by pt.category;
```

Result Grid		Filter Rows:
	category	revenue
▶	Classic	26.91
	Veggie	23.68
	Supreme	25.46
	Chicken	23.96

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date ,  
round(sum(revenue) over(order by order_date),2) as com_revenue  
from  
(select orders.order_date ,  
sum(order_details.quantity * pizzas.price) as revenue  
from order_details  
join orders  
    on order_details.order_id = orders.order_id  
join pizzas  
    on pizzas.pizza_id = order_details.pizza_id  
group by order_date) as sales;
```

Result Grid		Filter Rows:
	order_date	com_revenue
▶	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.35
	2015-01-11	25862.65
	2015-01-12	27781.7
	2015-01-13	29831.3
	2015-01-14	32358.7
	2015-01-15	34343.5
	2015-01-16	36937.65
	2015-01-17	39001.75
	2015-01-18	40978.6
	2015-01-19	43365.75
	2015-01-20	45763.65
	2015-01-21	47804.2
	2015-01-22	50300.9
	2015-01-23	52724.6
	2015-01-24	55013.85
	2015-01-25	56631.4
	2015-01-26	58515.8
	2015-01-27	£1042.05

Result 4 ×

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

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

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



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

```
select name ,revenue  
from  
(select category ,  
name,  
revenue,  
rank() over(partition by category order by revenue desc ) as r  
from  
(select pizza_types.category,  
pizza_types.name ,  
sum(pizzas.price * order_details.quantity) as revenue  
from pizza_types  
join pizzas  
    on pizzas.pizza_type_id = pizza_types.pizza_type_id  
join order_details  
    on order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category ,pizza_types.name) as a) as b  
where r<=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	32265.7000000065
	The Mexicana Pizza	26780.75
	The Five Cheese Pizza	26066.5

MEET OUR FOUNDER

Mr. Harshit Chugh

profession : Data Scientist

student in Dr. MPS collage ,Agra

mb.no. : 7665056712

[Read More](#)



Pizza sales Presentation

**THANK YOU
FOR ATTENTION**

See You Next

www.linkedin.com/in/harshitchugh2810