



it's  
Pizza  
Time

ORDER NOW



Hello

This is Abhishek  
Prakash . In this  
project i have  
utilised Sql queries  
that will solve the  
question related to  
pizza Sales.

Retrieve the total number of orders placed

```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    orders;
```

Result Grid | 

total_orders
21350

Calculate the total revenue generated  
from pizza Sales

SELECT

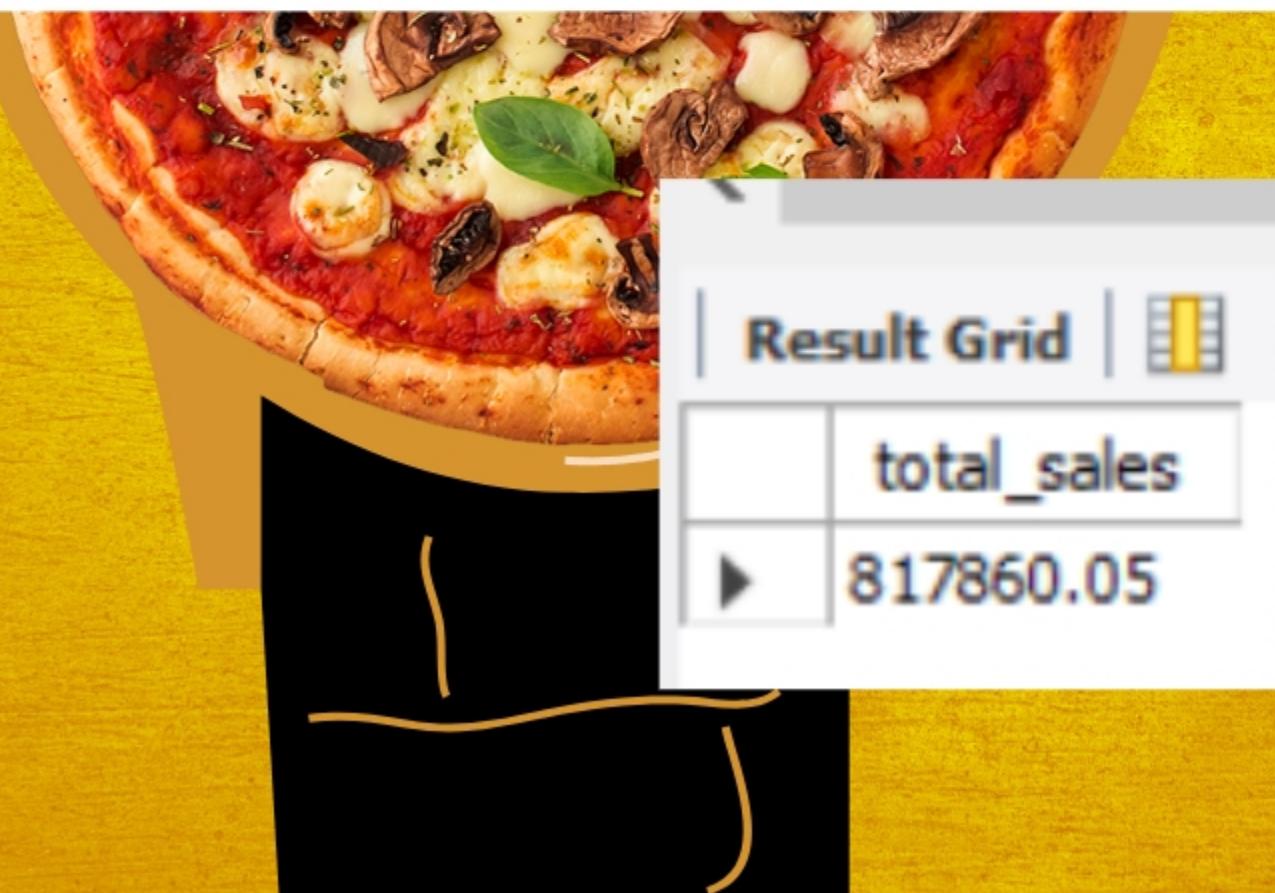
```
ROUND(SUM(orders_details.quantity * pizzas.price),  
2) AS total_sales
```

FROM

orders\_details

JOIN

pizzas ON pizzas.pizza\_id = orders\_details.pizza\_id



# identify the highest-priced pizza

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 R

	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

SELECT

```
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid



Filter 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

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

SELECT

```
pizza_types.category,  
SUM(order_details.quantity) AS quantity
```

FROM

```
pizza_types
```

JOIN

```
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
```

JOIN

```
order_details ON order_details.pizza_id = pizzas.pizza_id
```

GROUP BY pizza\_types.category

ORDER BY quantity DESC;



A photograph of a pizza slice with toppings like pepperoni and cheese.

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

Determine the distribution of orders  
by hour of the day

SELECT

HOUR(order\_time), COUNT(order\_id)

FROM

orders

GROUP BY HOUR(order\_time);



A screenshot of a MySQL Workbench interface showing the results of a SQL query. The results are displayed in a grid with two columns: 'hour(order\_time)' and 'count(order\_id)'. The data shows the distribution of orders by hour of the day.

	hour(order_time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468

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

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

SELECT

ROUND(AVG(quantity), 0) AS avg\_pizza\_ordered\_per\_day

FROM

(SELECT

orders.order\_date, SUM(order\_details.quantity) AS quantity

FROM

orders

JOIN order\_details ON orders.order\_id = order\_details.order\_id

GROUP BY orders.order\_date) AS order\_quantity;



round(avg(quantity),0)
138

# Determine the top 3 most ordered pizza types based on revenue

SELECT

    pizza\_types.name,

    SUM(order\_details.quantity \* pizzas.price) 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.name

ORDER BY revenue DESC

LIMIT 3;

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

# Analyze the cumulative revenue generated over time

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



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