



PIZZA RESTAURANT

# PIZZA SALE PROJECT

[www.linkedin.com/in/abdul-qadeer-khan-a77b62301](https://www.linkedin.com/in/abdul-qadeer-khan-a77b62301)







# HELLO

My name is Abdul Qadeer Khan. I am a student of Computer Science. In this project I have utilized the SQL queries to solve the questions related to pizza sales



# DATABASE

A database is the structured collection of data that is electrically stored in computer.  
SQL is a language to solve the queries related to structured data.

# DATASET LINK

<https://github.com/Ayushi0214/pizza-sales---SQL>



# RETRIEVE THE TOTAL NUMBERS OF ORDER PLACED

```
SELECT
```

```
    COUNT(order_id) AS total_orders
```

```
FROM
```

```
    orders
```

Result Grid

	total_orders
▶	21350



# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

SELECT

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

AS total\_revenue

FROM

order\_details

JOIN

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

Result Grid

	total_revenue
▶	817860.0499999993

# IDENTIFY THE HIGHEST PRICE PIZZA

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

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

# IDENTIFY MOST COMMONLY ORDERED PIZZA SIZE

```
SELECT
    pizzas.size, COUNT(order_details.quantity) AS order_count
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY size
ORDER BY order_count DESC
LIMIT 1
```

Result Grid			Filter Rows:
	size	order_count	
▶	L	18526	



# LIST THE TOP 5 MOST ORDER PIZZA\_TYPE 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 pizzas.pizza_id = order_details.pizza_id
GROUP BY 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

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 pizzas.pizza_id = order_details.pizza_id
```

GROUP BY category

Result Grid





Filter Rows:

	category	quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

# DETERMINE THE DISTRIBUTION OF ORDERS BY THE HOURS OF DAY

```
SELECT
    HOUR(order_time) AS order_hour,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY order_hour
```

Result Grid     Filter Rows:		
	order_hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399



# JOIN THE RELEVANT TABLES TO FIND THE CATEGORY\_WISE DISTRIBUTION OF PIZZAS

```
SELECT
    pizza_types.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 NUMBERS OF PIZZAS ORDERED PER DAY

```
SELECT
    round(AVG(quantity), 0) as avgPizzas_orderPerDay
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 order_date) AS order_quantity
```

Result Grid		Filter Rows:
	avgPizzas_orderPerDay	
▶	138	



# DETERMINE THE TOP 3 MOST ORDERED PIZZAS BASED ON THE REVENUE

SELECT

    pizza\_types.name,  
    ROUND(SUM(order\_details.quantity \* pizzas.price),  
          2) AS Revenue

FROM

    pizza\_types  
    JOIN  
    pizzas ON pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id  
    JOIN  
    order\_details ON pizzas.pizza\_id = order\_details.pizza\_id

GROUP BY name

ORDER BY revenue DESC

LIMIT n

Result Grid			Filter 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

```
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
        SUM(order_details.quantity * pizzas.price) AS total_revenue
    FROM
        order_details
    JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100),
    2) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY revenue DESC
```

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



# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

```
select order_date,  
       sum(revenue) over(order by order_date) as cumulative_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 order_details.order_id=orders.order_id  
group by order_date) as sales
```

Result Grid			Filter Rows:
	order_date	cumulative_revenue	
▶	2015-01-01	2713.8500000000000004	
	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	

# DETERMINE THE TOP 3 MOST ORDERED PIZZA\_TYPES BASED ON THE REVENUE FOR EACH CATEGORY

```
select name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as top_ranks
from
(SELECT
pizza_types.category,
pizza_types.name,
SUM(order_details.quantity * pizzas.price) AS revenue
FROM
pizza_types
JOIN
pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category , pizza_types.name) as A) as B
where top_ranks<=3;
```

Result Grid	Filter Rows:
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	32265.700000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5





# THANK FOR YOU

LinkedIn  
[www.linkedin.com/in/abdul-qadeer-khan-a77b62301](https://www.linkedin.com/in/abdul-qadeer-khan-a77b62301)