

# **ANALYSING PIZZAS SALES DATA USING SQL QUERIES**



[www.reallygreatsite.com](http://www.reallygreatsite.com)

# **WELCOME MESSAGE**

Hi this is Adnan khan , Data Analyst, Python developer  
i have created this project for my portfolio  
Analysing Pizzas sales data using sql queries



# CALCULATING ORDERS PER HOUR

```
• SELECT  
    HOUR(orders.order_time) AS hours,  
    SUM(orders_detail.quantity) AS quantity  
FROM  
    orders  
    JOIN  
    orders_detail ON orders.order_id = orders_detail.order_id  
GROUP BY hours  
ORDER BY quantity DESC;
```

hours	quantity
12	6776
13	6413
18	5417
17	5211
19	4406
16	4239
14	3613
20	3534
15	3216
11	2728
21	2545
22	1386
23	68
10	18
9	4

# MOST ORDERED PIZZA SIZE BASED ON SIZE

```
SELECT
    pizzas.size, COUNT(pizzas.size) AS most_ordered_pizza_size
FROM
    orders_detail
    JOIN
        pizzas ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY pizzas.size ASC ;
```

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

# CALCULATING PIZZAS SALES ON BASIS OF CATEGROY

```
SELECT
    pizza_type.category, SUM(orders_detail.quantity) AS quantity
FROM
    orders_detail
    JOIN
    pizzas ON orders_detail.pizza_id = pizzas.pizza_id
    JOIN
    pizza_type ON pizzas.pizza_type_id = pizza_type.pizza_type
GROUP BY pizza_type.category
ORDER BY quantity
```

Result Grid		
	category	quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	13529

# ORDERS OF PIZZA SALES

- SELECT

```
 pizza_type.pizza_name,  
 SUM(orders_detail.quantity) AS quantity  
 FROM  
 pizza_type  
 JOIN  
 pizzas ON pizza_type.pizza_type = pizzas.pizza_type_id  
 JOIN  
 orders_detail ON orders_detail.pizza_id = pizzas.pizza_id  
 GROUP BY pizza_type.pizza_name  
 ORDER BY pizza_name;
```

result Grid	Filter Rows:
pizza_name	quantity
The Barbecue Chicken Pizza	2432
The Big Meat Pizza	1914
The Brie Carre Pizza	490
The Calabrese Pizza	937
The California Chicken Pizza	2370
The Chicken Alfredo Pizza	987
The Chicken Pesto Pizza	973
The Classic Deluxe Pizza	2453
The Five Cheese Pizza	1409
The Four Cheese Pizza	1902

Result 8 ×

# TOTAL REVENUE

```
SELECT
    ROUND(SUM(orders_detail.quantity * pizzas.price),3) AS total_revenue
FROM
    orders_detail
    JOIN
        pizzas ON pizzas.pizza_id = orders_detail.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05

# DISTRIBUTION OF TOP 3 SELLING PIZZAS ON BASIS OF REVENUE INTO PERCENTAGE

```
SELECT
    pizza_type.category,
    ROUND(SUM(orders_detail.quantity * pizzas.price) / (SELECT
        ROUND(SUM(orders_detail.quantity * pizzas.price), 3) AS total_revenue
    FROM
        orders_detail
        JOIN
            pizzas ON pizzas.pizza_id = orders_detail.pizza_id) * 100, 2) AS revenue_in_percentage
FROM
    pizzas
    JOIN
        pizza_type ON pizzas.pizza_type_id = pizza_type.pizza_type
    JOIN
        orders_detail ON orders_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_type.category
ORDER BY revenue_in_percentage DESC;
```

Result Grid	
category	revenue_in_percentage
Supreme	25.46
Classic	24.6
Chicken	23.96
Veggie	23.68

# CALCULATING CUMMULATIVE REVENUE ON ORDER DATE

```
SELECT order_date,SUM(revenue) OVER(ORDER BY order_date) as cum_revenue
FROM
(SELECT orders.order_date,SUM(orders_detail.quantity*pizzas.price) as revenue
FROM orders_detail
JOIN
pizzas ON orders_detail.pizza_id=pizzas.pizza_id
JOIN orders ON orders.order_id=orders_detail.order_details_id
GROUP BY orders.order_date) AS sales;
```

Result Grid		
	order_date	cum_revenue
▶	2015-01-01	1171.45
	2015-01-02	2316.1000000000004
	2015-01-03	3433.8
	2015-01-04	4341.8
	2015-01-05	5247.25

Result 46 ×