

SPECIAL OFFER
PIZZA TIME

CHEESE
PIZZA

**30%
OFF**



FREE DELIVERY
1234567890

ORDER NOW

Hello !

Myself Abhishek Kumar. In this project I utilize SQL query to solve questions that are related to pizza sales.

RETRIEVE TOTAL NUMBER OF ORDER PLACED

```
SELECT
```

```
    COUNT(order_id) AS total_order
```

```
FROM
```

```
    orders;
```

Result Grid	
	total_order
▶	21350



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid	
	total_revenue
▶	817860.05

IDENTIFY THE HIGHEST PRICE PIZZAS

```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
    JOIN
        pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY 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 size
ORDER BY order_count DESC;
```

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



LIST THE TOP 5 MOST ORDERED PIZZAS TYPE ALONG WITH THEIR QUANTITY

```
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: <input type="text"/>		
	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 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;
```

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

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

```
SELECT
    HOUR(orders.order_time) AS hour,
    COUNT(orders.order_id) AS order_count
FROM
    orders
GROUP BY hour
ORDER BY order_count DESC;
```

Result Grid			Filter
	hour	order_count	
▶	12	2520	
	13	2455	
	18	2399	
	17	2336	
	19	2009	
	16	1920	
	20	1642	
	14	1472	
	15	1468	
	11	1231	
	21	1198	
	22	663	
	23	28	
	10	8	
	9	1	

JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS

SELECT



 pizza_types.category **AS** category,
 COUNT(pizza_types.category) **AS** category_count

FROM

 pizza_types



GROUP BY category

ORDER BY category_count **DESC**;

Result Grid   Filter Rows		
	category	category_count
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6



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

```
SELECT
    ROUND(AVG(quantity), 0) average_order_per_day
FROM
    (SELECT
        DATE(orders.order_date) AS date,
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS quantity;
```

Result Grid			 Filter R
	average_order_per_day		
▶	138		

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

```
SELECT
    pizza_types.name AS name,
    SUM(pizzas.price * order_details.quantity) 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 3;
```

Result Grid   Filter Rows: <input type="text"/>		
	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) AS category,
    ROUND((SUM(pizzas.price * order_details.quantity) / (SELECT
        SUM(pizzas.price * order_details.quantity)
    FROM
        pizzas
        JOIN
            order_details ON pizzas.pizza_id = order_details.pizza_id)) * 100,
    2) AS percentage_contribution
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 percentage_contribution DESC;
```

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




ANALYZE CUMULATIVE REVENUE GENERATED OVER TIME

```
SELECT order_date,  
SUM(revenue)  
OVER(ORDER BY order_date) AS cum_revenue  
FROM (SELECT orders.order_date,  
SUM(pizzas.price*order_details.quantity) AS revenue  
FROM pizzas  
JOIN order_details  
ON pizzas.pizza_id=order_details.pizza_id  
JOIN orders  
ON order_details.order_id=orders.order_id  
GROUP BY orders.order_date) AS sales order by order_date desc limit 5;
```

Result Grid   Filter Rows:		
	order_date	cum_revenue
▶	2015-12-31	817860.05
	2015-12-30	814944.05
	2015-12-29	813606.25
	2015-12-28	812253
	2015-12-27	810615.8

DETERMINE THE TOP 2 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
select category,name,revenue
from (select category,name,revenue,rank()
over(partition by category order by revenue desc) as rn
from (select pizza_types.category as category,
pizza_types.name as name,
sum(pizzas.price*order_details.quantity) as revenue
from pizzas
join order_details
on pizzas.pizza_id=order_details.pizza_id
join pizza_types
on pizzas.pizza_type_id=pizza_types.pizza_type_id
group by category,name)
as a)
as b
where rn<3;
```

Result Grid   Filter Rows: <input type="text"/> Export: 			
	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75