

Where Every Slice is a Taste of Perfection

WELCOME TO PIZZA HUT



ORDER
NOW



HELLO

My name is Shalini Gupta, and in this project, I utilized SQL queries to analyze and solve questions related to pizza sales. This experience allowed me to enhance my skills in data analysis, query optimization, and deriving meaningful insights from complex datasets. By addressing real-world sales scenarios, I gained a deeper understanding of data-driven decision-making and the role of SQL in streamlining business processes. This project has been a valuable step in honing my technical expertise and problem-solving abilities.



RETRIVE THE TOTAL NUMBER OF ORDER PLCAED

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

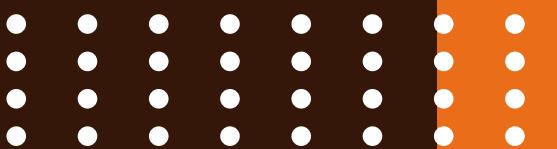
Result Grid	
	total_orders
▶	7487

CALCULATE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

Result Grid	
	total_revenue
▶	15591.5

IDENTIFY THE HIGHEST PRICED PIZZAS

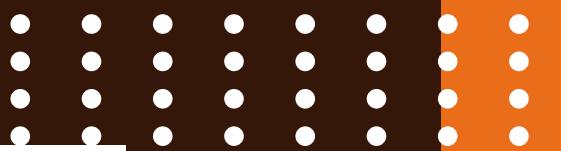


```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED



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

Result Grid |

	order_count	size
371	L	
286	M	
255	S	
9	XL	

LIST TOP MOST 5 ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES

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

	name	quantity
▶	The Thai Chicken Pizza	54
	The Pepperoni Pizza	54
	The Hawaiian Pizza	48
	The Barbecue Chicken Pizza	47
	The Classic Deluxe Pizza	46

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

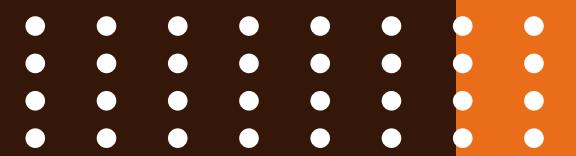


```
SELECT
    SUM(orders_details.quantity) AS quantity,
    pizza_types.category
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category order by quantity desc ;
```

Result Grid |

	quantity	category
▶	290	Classic
	218	Veggie
	218	Chicken
	216	Supreme

THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



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

Result Grid |

	hour	order_count
▶	11	409
	12	910
	13	839
	14	557
	15	506
	16	672
	17	882
	18	808

JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS



```
SELECT
```

```
    COUNT(pizza_type_id) as distribution ,category
```

```
FROM
```

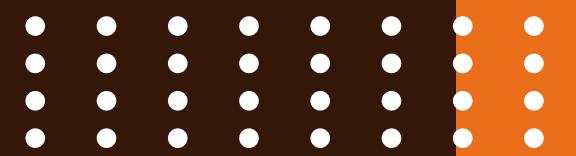
```
    pizza_types
```

```
GROUP BY category;
```

Result Grid | Filter R

	distribution	category
▶	6	Chicken
	8	Classic
	9	Supreme
	9	Veggie

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



```
SELECT  
    ROUND(AVG(QUANTITY), 0) AS AVG_ORDERED_PIZZA_PER_DAY  
FROM  
    (SELECT  
        ORDERS.ORDER_DATE, SUM(orders_details.quantity) AS QUANTITY  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
```

Result Grid	
	AVG_ORDERED_PIZZA_PER_DAY
▶	135

DETERMINE THE TOP 3 MOST ORDERED PIZZA BASED ON REVENUE



```
SELECT
    pizza_types.name,
    SUM(orders_details.quantity * pizzas.price) AS REVENUE
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizza_types.NAME
ORDER BY REVENUE DESC
LIMIT 3;
```

Result Grid | Filter Rows:

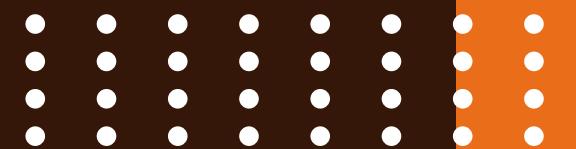
	name	REVENUE
▶	The Thai Chicken Pizza	996.5
	The Barbecue Chicken Pizza	843.25
	The Italian Supreme Pizza	782.25

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE

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

	category	revenue
▶	Classic	27.13
	Chicken	25.09
	Supreme	24.24
	Veggie	23.54

ANALYSE 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(orders_details.quantity * pizzas.price) AS revenue  
FROM  
    orders_details  
    JOIN  
    pizzas ON orders_details.pizza_id = pizzas.pizza_id  
    JOIN  
    orders ON orders.order_id = orders_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
	2015-01-07	15591.5

Pizza hut

THANK YOU
FOR ATTENTION