



Pizza Hut
RESTAURANT SALES



SQL PROJECT

Presenting the Pizza Sales Analysis Report, which I've generated using SQL. Goal with this report is to gain insights into the sales performance across various types of pizzas, customer preferences, and trends over time. By analyzing the data, I was able to identify key patterns and opportunities for optimizing the sales strategy. Let's dive into the findings and see how they can help us drive better business decisions."



By Apurv Vats




Retrieve the total
number of orders
placed.

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


Result Grid			Filter Rows
	total_orders_placed		
▶	21350		





Calculate the total
revenue generated
from pizza sales.

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

Result Grid			
	total_sales		
▶	817860.05		






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 Rows
	name	price	
▶	The Greek Pizza	35.95	





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 Rows	
	name	price		
▶	The Greek Pizza	35.95		



Identify the most common pizza size ordered.

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

Result Grid			Filter Rows
	size	total_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(quantity) AS quantity
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.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
    category, SUM(quantity) AS total_quantity
FROM
    pizzas
    JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY total_quantity DESC;
```

Result Grid			Filter Rows:
	category	total_quantity	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	





Determine 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(order_time);
```

Result Grid			Filter
	hour	order_count	
▶	11	123	Resets all
	12	252	
	13	2455	
	14	1472	
	15	1468	
	16	1920	
	17	2336	
	18	2399	
	19	2009	
	20	1642	
	21	1198	
	22	663	
	23	28	
	10	8	
	9	1	





Join relevant tables
to find the category-
wise distribution of
pizzas.

```
SELECT  
    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
number of pizzas ordered
per day.

```
SELECT
    ROUND(AVG(qty), 0) AS average_quantity_per_day
FROM
    (SELECT
        order_date, SUM(quantity) AS qty
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY order_date) AS order_quantity;
```

Result Grid |   Filter Rows:

	average_quantity_per_day
▶	138





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

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

Result Grid			Filter Rows:	
	name	total_sales		
▶	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
	The California Chicken Pizza	41409.5		





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

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

Result Grid			Filter Rows:	
	name	total_sales		
▶	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
    category,
    ROUND((SUM(orders_details.quantity * pizzas.price) / (SELECT
        (ROUND(SUM(orders_details.quantity * pizzas.price),
            2))
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id = orders_details.pizza_id)) * 100,
    2) AS revenue
FROM
    pizzas
    JOIN
    pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    orders_details ON orders_details.pizza_id = pizzas.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 cum_revenue from
(select order_date, sum(orders_details.quantity * pizzas.price) as revenue from
orders join
orders_details
on orders.order_id = orders_details.order_id
join
pizzas
on
pizzas.pizza_id= orders_details.pizza_id
group by order_date) as sales;
```

Result Grid			 Filter Rows: <input type="text"/>
	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	16560.7	
	2015-01-08	19399.05	
	2015-01-09	21526.4	
	2015-01-10	23990.350000000002	
	2015-01-11	25862.65	
	2015-01-12	27781.7	
	2015-01-13	29831.300000000003	
	2015-01-14	32358.700000000004	
	2015-01-15	34343.500000000001	
	2015-01-16	36937.650000000001	
	2015-01-17	39001.750000000001	
	2015-01-18	40978.600000000006	
	2015-01-19	43365.750000000001	
	2015-01-20	45763.650000000001	
	2015-01-21	47804.200000000001	
	2015-01-22	50300.900000000001	





Determine the top 3 most ordered pizza types based on revenue for each pizza category.

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

Result Grid			Filter Rows:
	name	total_sales	
▶	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	32265.70000000065	
	The Mexicana Pizza	26780.75	
	The Five Cheese Pizza	26066.5	

