



SQL

PROJECT ON

PIZZA SALES

- WHERE EVERY SLICE TELLS A STORY



Contact

ABOUT PROJECT



PIZZA SALES ANALYSIS USING SQL

Analyzed PizzaHut sales data to extract key insights using SQL. Covered total orders, revenue, top-selling pizzas, size trends, hourly order patterns, and category-wise performance. Included advanced revenue analysis and cumulative sales tracking.



SQL QUERY CHALLENGES SOLVED USING PIZZA SALES DATA - FROM BASIC TO ADVANCED

Contact



Basic:

Retrieve the total number of orders placed.

Calculate the total revenue generated from pizza sales.

Identify the highest-priced pizza.

Identify the most common pizza size ordered.

List the top 5 most ordered pizza types along with their quantities.

Intermediate:

Join the necessary tables to find the total quantity of each pizza category ordered.

Determine the distribution of orders by hour of the day.

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

Group the orders by date and calculate the average number of pizzas ordered per day.

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

Advanced:

Calculate the percentage contribution of each pizza type to total revenue.

Analyze the cumulative revenue generated over time.

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



RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

Contact



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

Result Grid

	count(order_id)
▶	21350



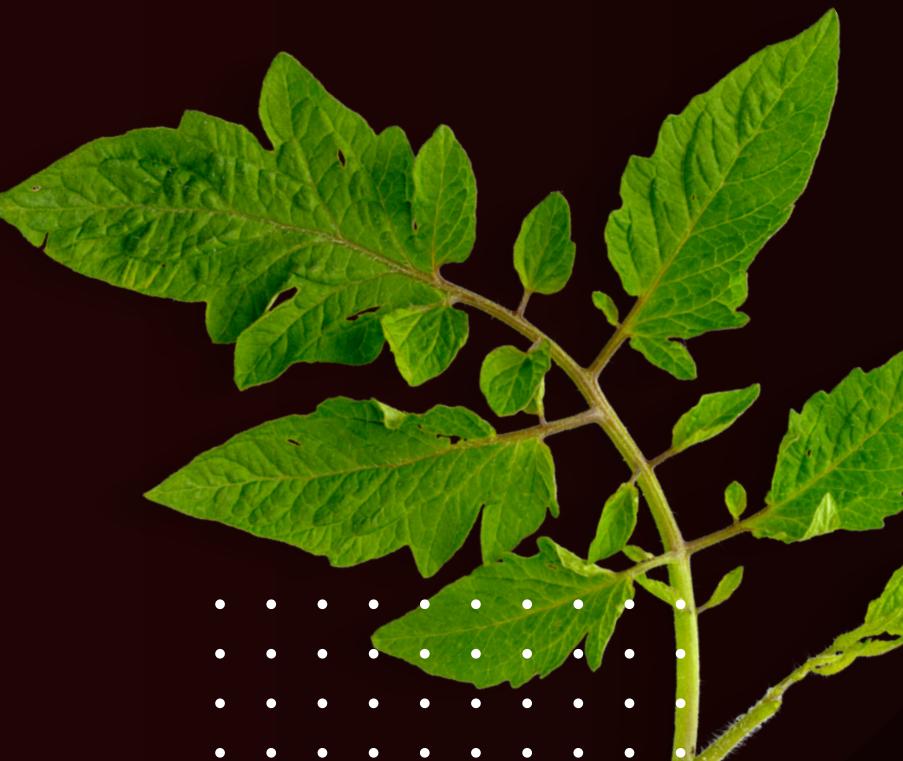
CALCULATE A TOTAL REVENUE GENERATED FROM PIZZA SALES

Contact



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

Result Grid	
	total_revenue
▶	817860.05



IDENTIFY THE HIGHEST PRIZED PIZZA

[Home](#)[Contact](#)

```
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.

Contact



```
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 pizzas.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 PIZZA TYPES ALONG WITH THEIR QUANTITIES.

[Home](#)[Contact](#)

```
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		
	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

Contact



```
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;
```

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

[Home](#)

Contact



SELECT

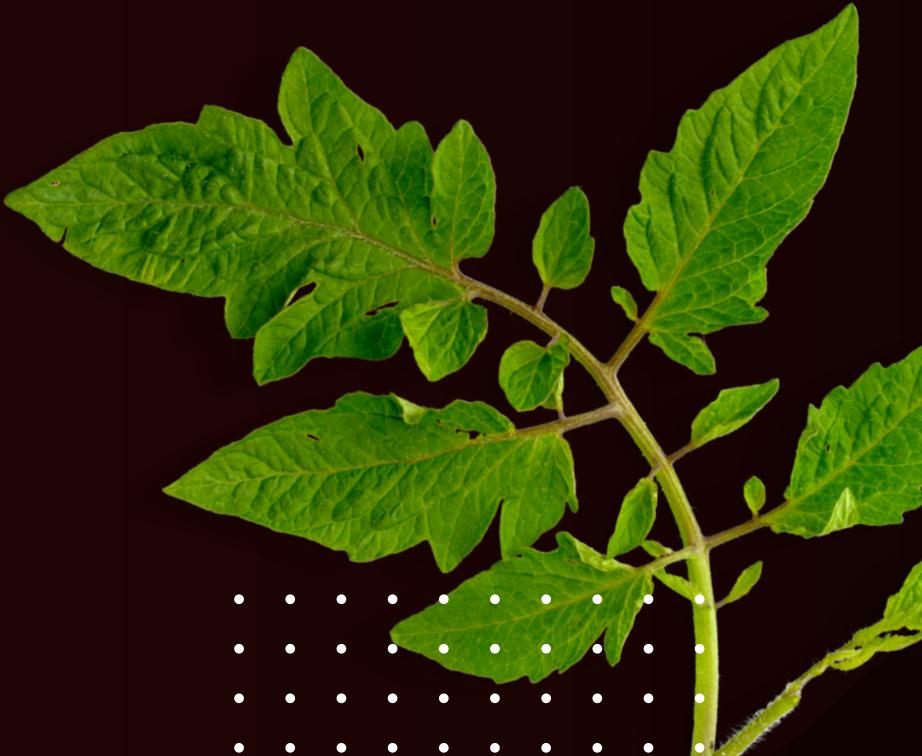
```
HOUR(order_time) AS hour, COUNT(order_id) AS order_count
```

FROM

```
orders
```

GROUP BY hour;

Result Grid		
	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642



JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS

Home

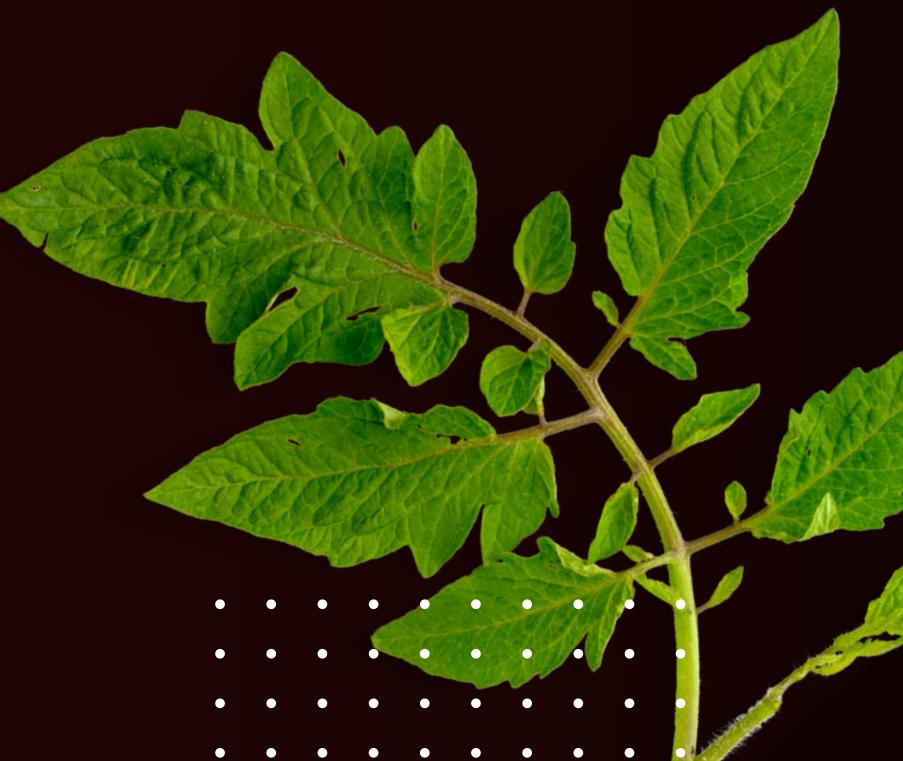
Contact



```
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



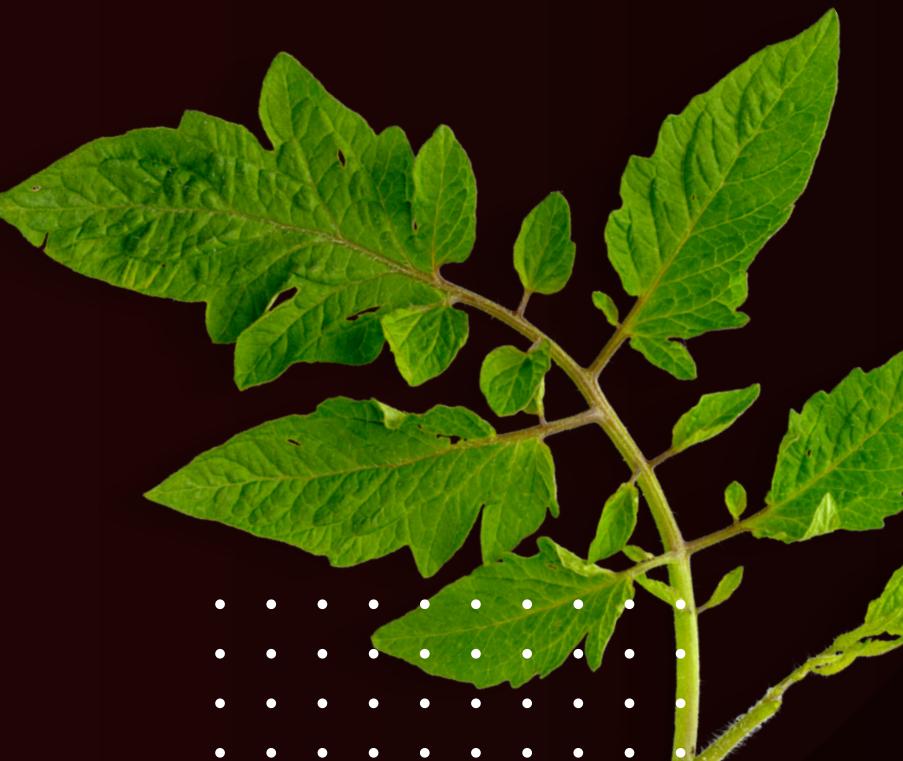
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY

Contact



```
SELECT  
    ROUND(AVG(quantity), 0) as avg_pizzas_per_day  
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 orders.order_date) AS order_quantity;
```

Result Grid	
	avg_pizzas_per_day
▶	138



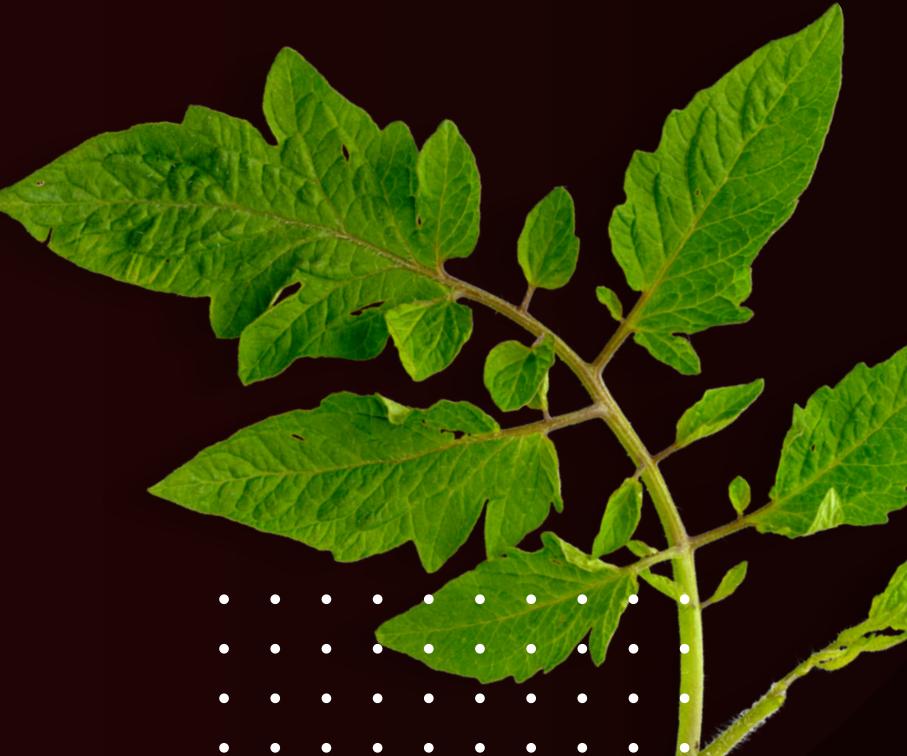
DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

[Home](#)[Contact](#)

```
SELECT
    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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 5;
```

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

[Home](#)

Contact



```
select pizza_types.category,  
round((sum(order_details.quantity*pizzas.price) / (select round(sum(order_details.quantity * pizzas.price),2) as total_sales  
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 order_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

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



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

Home

Contact



```
select order_date,  
sum(revenue) over (order by order_date) as cum_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 orders.order_id = order_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



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

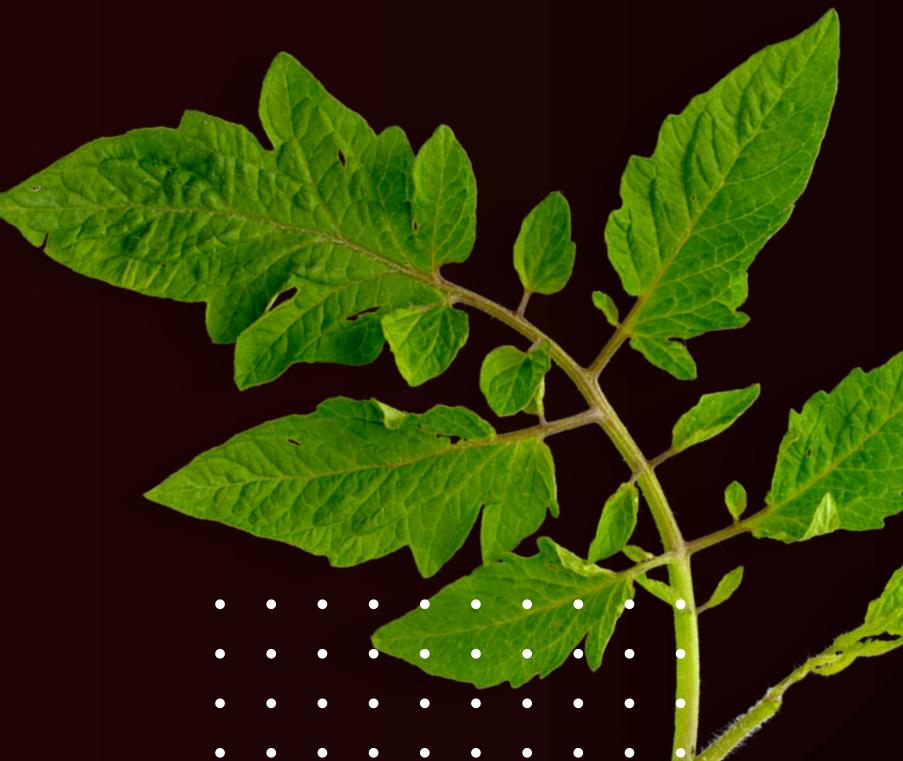
Contact



```
select name, revenue from
(select category, name , revenue,
rank() over(partition by category order by revenue) as rn
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 order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as a) as b
where rn <=3 ;
```



name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087



THANK YOU