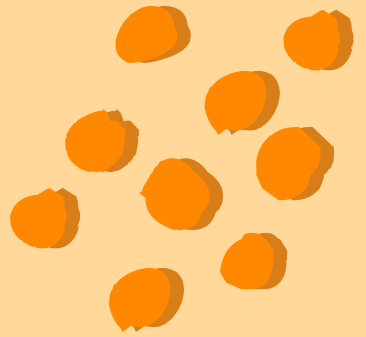


SQL PROJECT SALES





Hello!

My name is Biswasunder Mohapatra, and in this project, I have utilized SQL queries to address various analytical questions related to pizza sales. The goal of this project was to derive meaningful insights from the pizza sales data by writing and executing SQL queries.

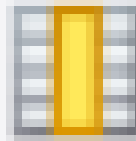




Retrieve the total number of orders placed.

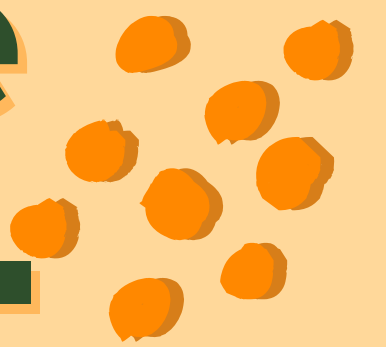


```
SELECT  
    COUNT(order_id) AS total_orders  
FROM  
    pizzahut.order;
```

Result Grid 	
	total_orders
▶	21350

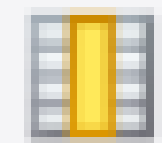


Calculate the total revenue generated from pizza sales.



```
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
```

Result Grid

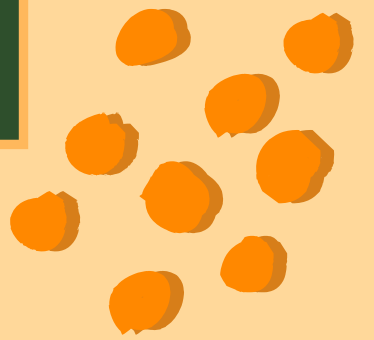


	total_sales
	817860.05





Identify the highest-priced pizza.



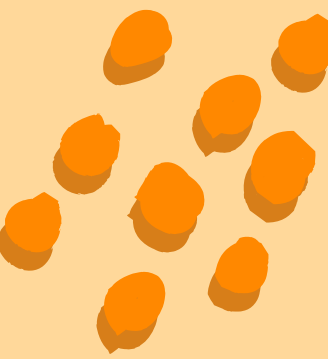
```
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	





Identify the most common pizza size ordered.

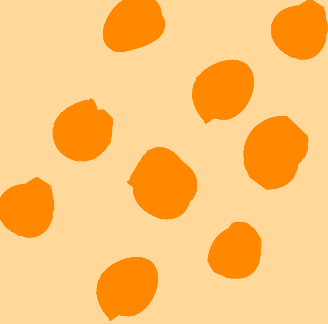


```
select quantity, count(order_details_id)
from pizzahut.order_details group by quantity;

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					F
	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

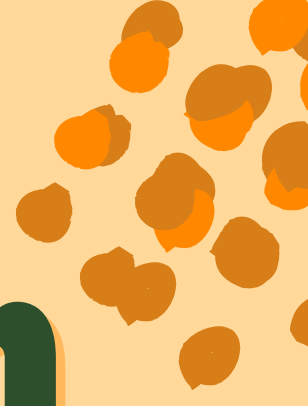
```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

	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
    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.

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

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





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

```
select category, count(name) from pizza_types  
group by category
```

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(order_quantity.total_quantity), 0) AS avg_quantity
FROM
    (SELECT
        `order`.order_date,
        SUM(order_details.quantity) AS total_quantity
    FROM
        `order`
    JOIN order_details ON `order`.order_id = order_details.order_id
    GROUP BY `order`.order_date) AS order_quantity;
```

Result Grid	
	avg_quantity
	138






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

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

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,
    (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 as revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

category	revenue
Classic	26.90596025566967
Supreme	25.45631126009862
Chicken	23.955137556847287
Veggie	23.682590927384577



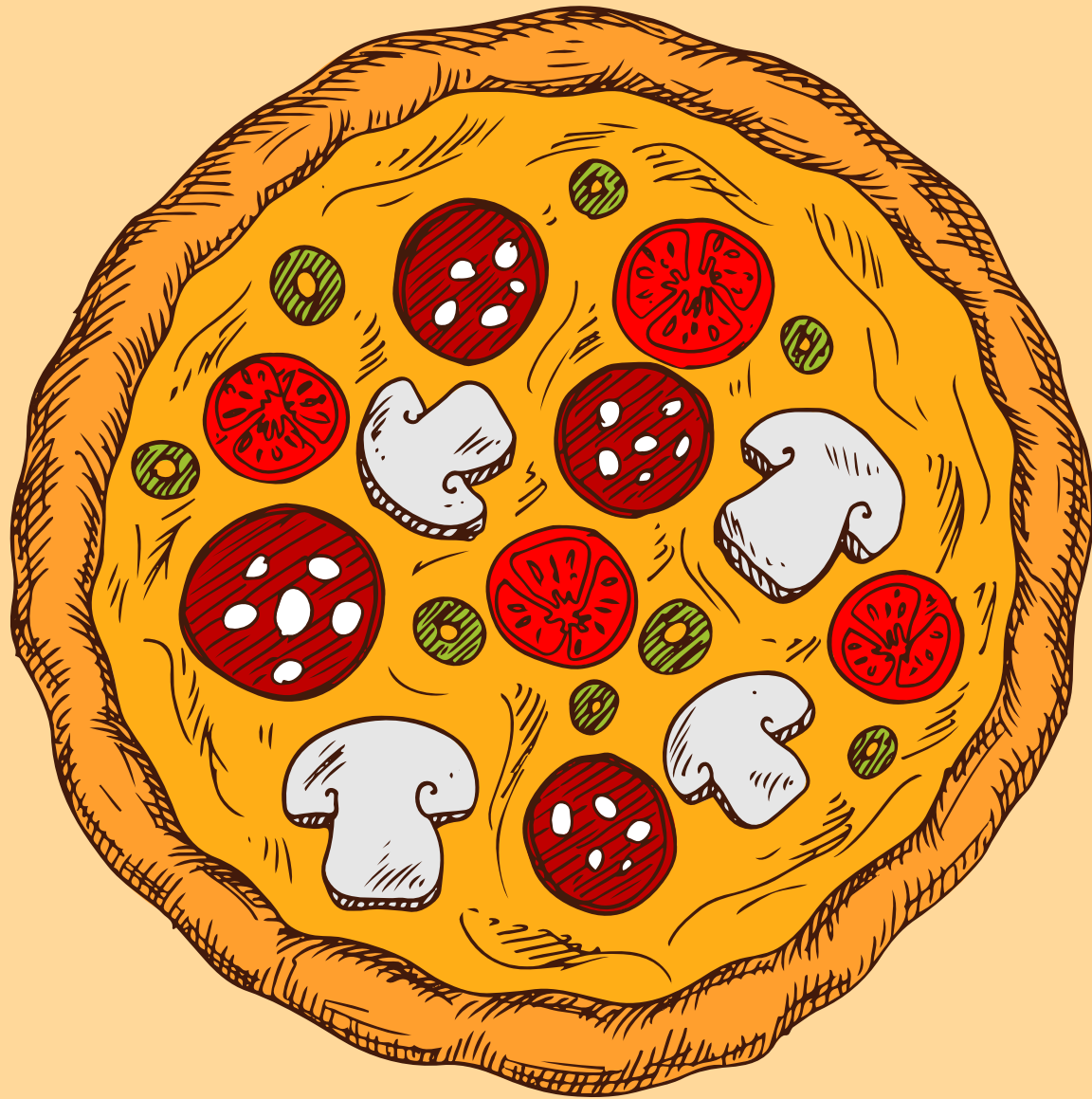
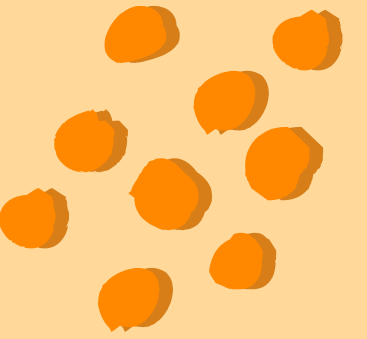


Analyze the cumulative revenue generated over time.

```
select order_date,  
sum(revenue) over (order by order_date) as cum_revenue  
from  
(SELECT `order`.order_date,  
        SUM(order_details.quantity * pizzas.price) AS revenue  
FROM order_details  
JOIN pizzas ON order_details.pizza_id = pizzas.pizza_id  
JOIN `order` ON `order`.order_id = order_details.order_id  
GROUP BY `order`.order_date) as sales;
```

order_date	cum_revenue
2015-01-01	2713.850000000000004
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





**THANK
YOU**

