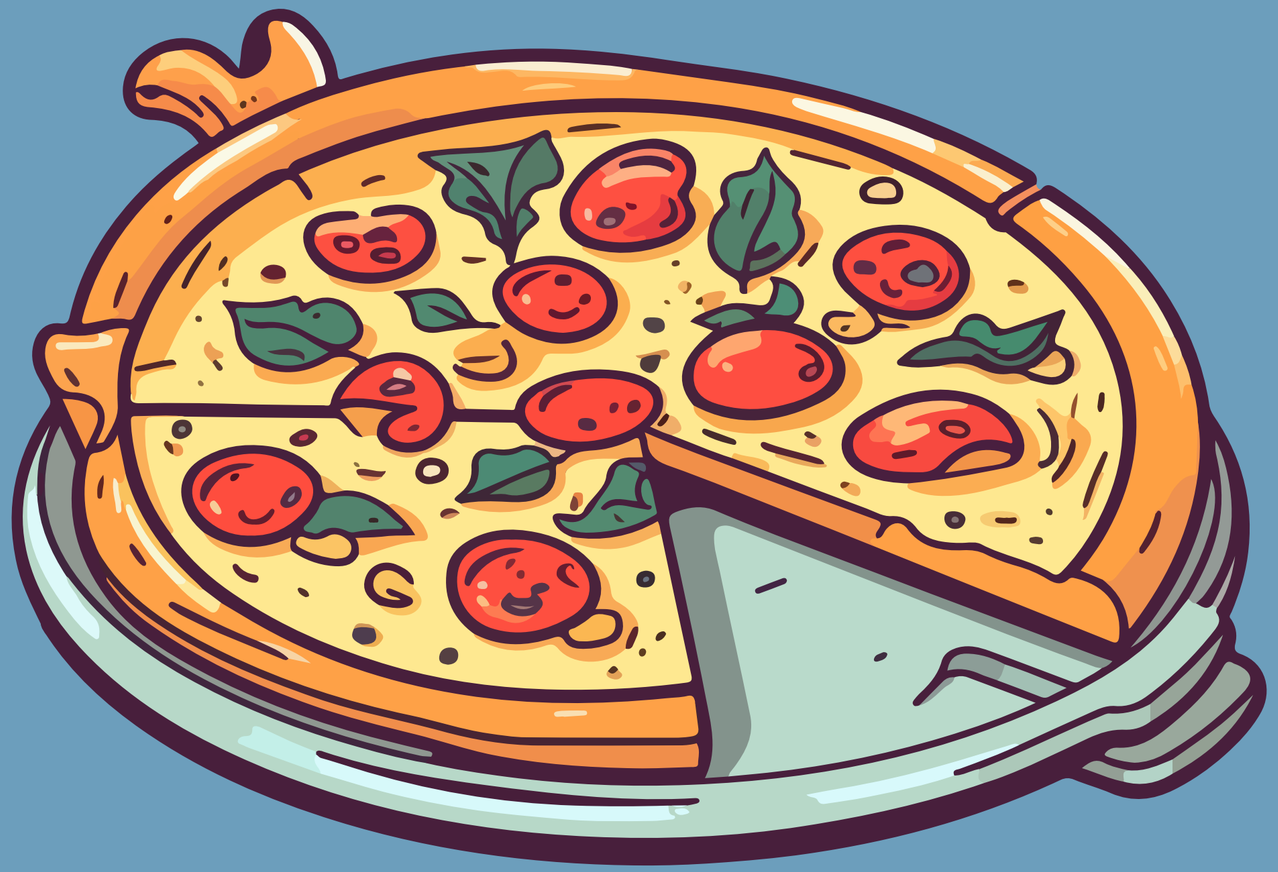
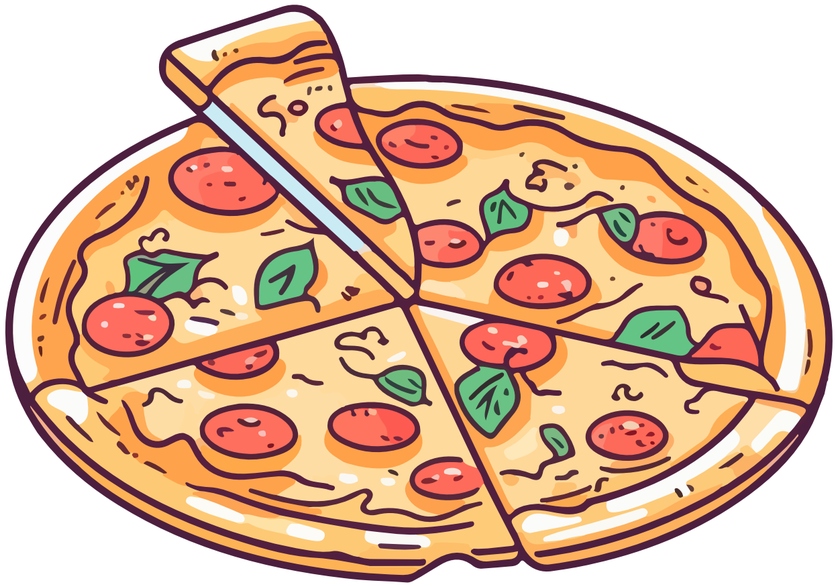


PIZZA SALES PROJECT USING MYSQL

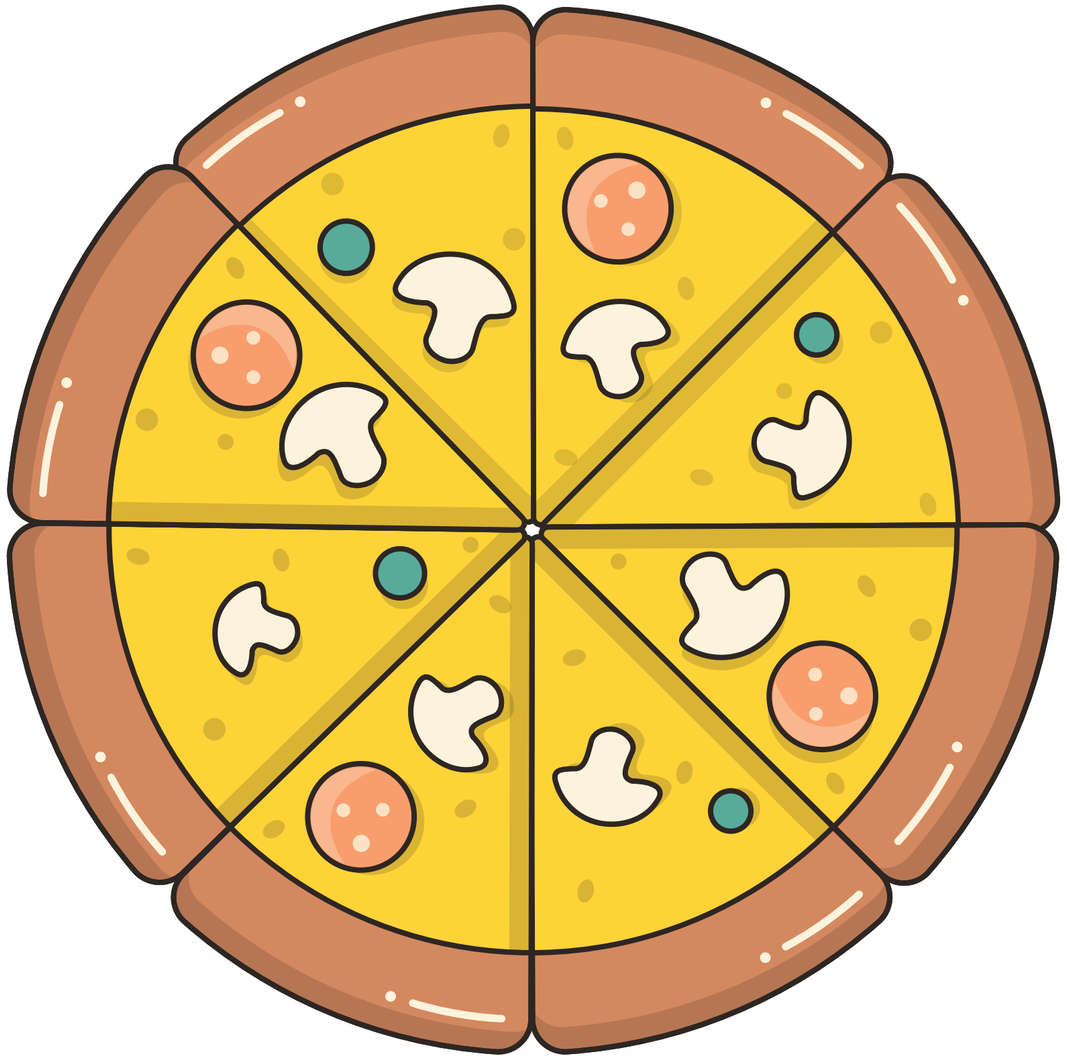
ABOUT THE PROJECT



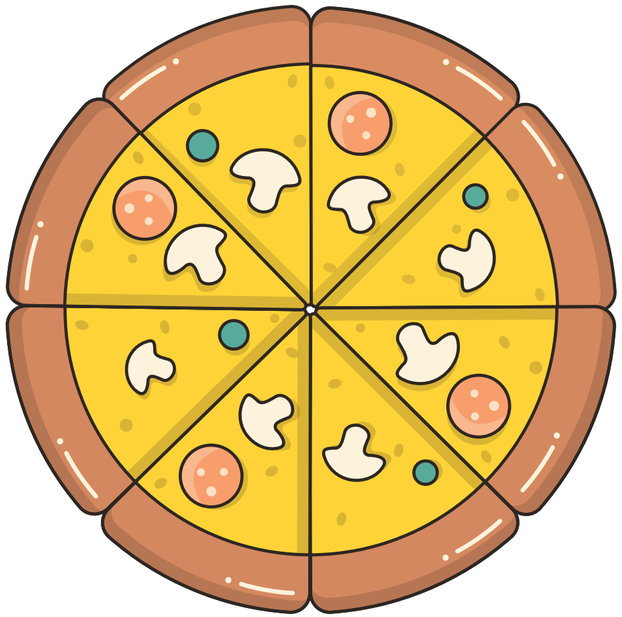
HANDS-ON SQL PORTFOLIO PROJECT USING A PIZZA SALES DATASET THROUGH MYSQL. COVERS DATA CLEANING, AGGREGATION, JOINS, AND REAL-WORLD BUSINESS ANALYSIS.

1

Retrieve the total number of orders placed.



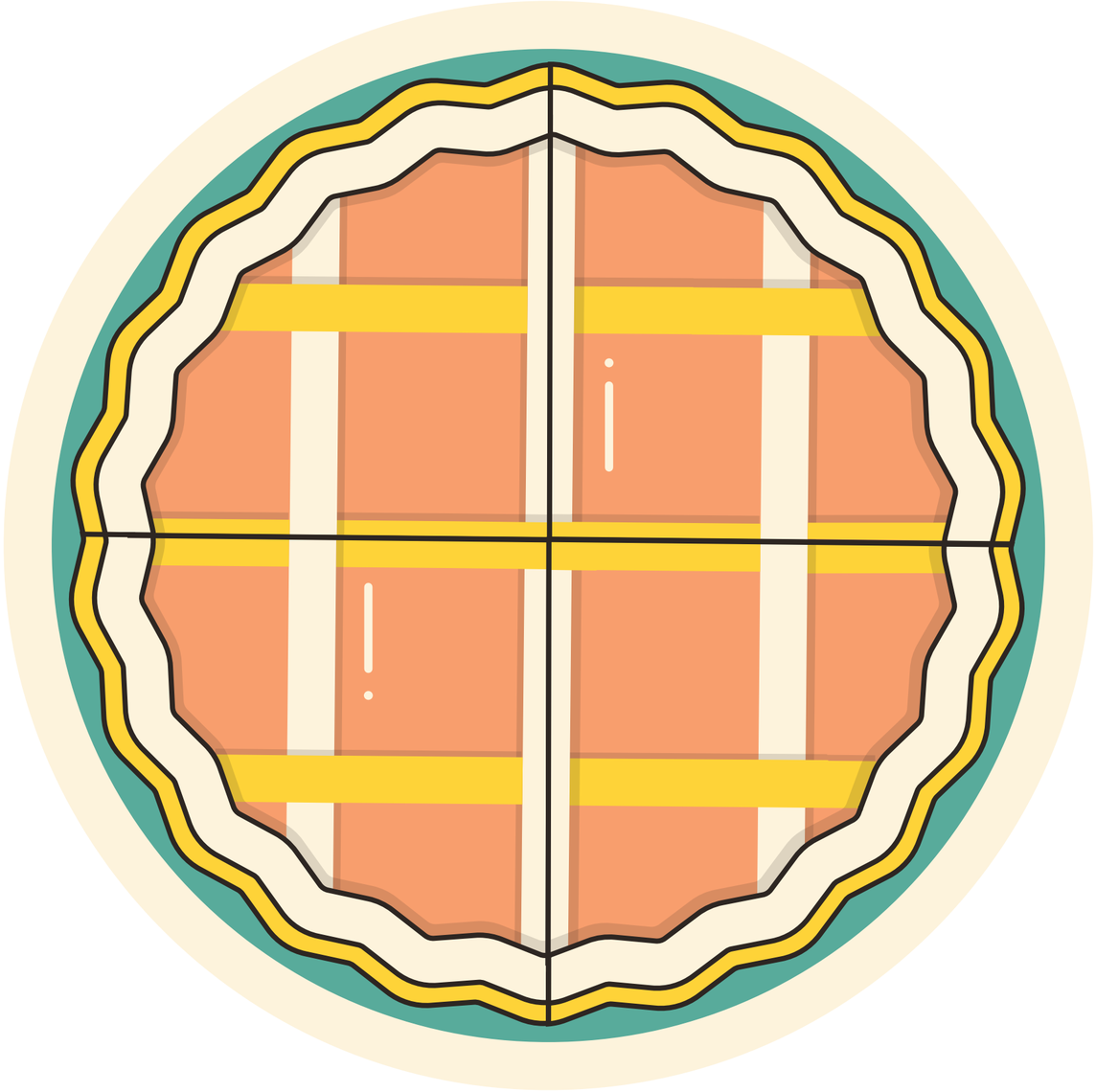
ANSWER



```sql
-- Retrieve the total number of orders placed.

SELECT
    COUNT(order_id) AS totalt_num_oder
FROM
    orders
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

totalt_num_oder
21350



2

Calculate the total Revenue generated from pizza sales

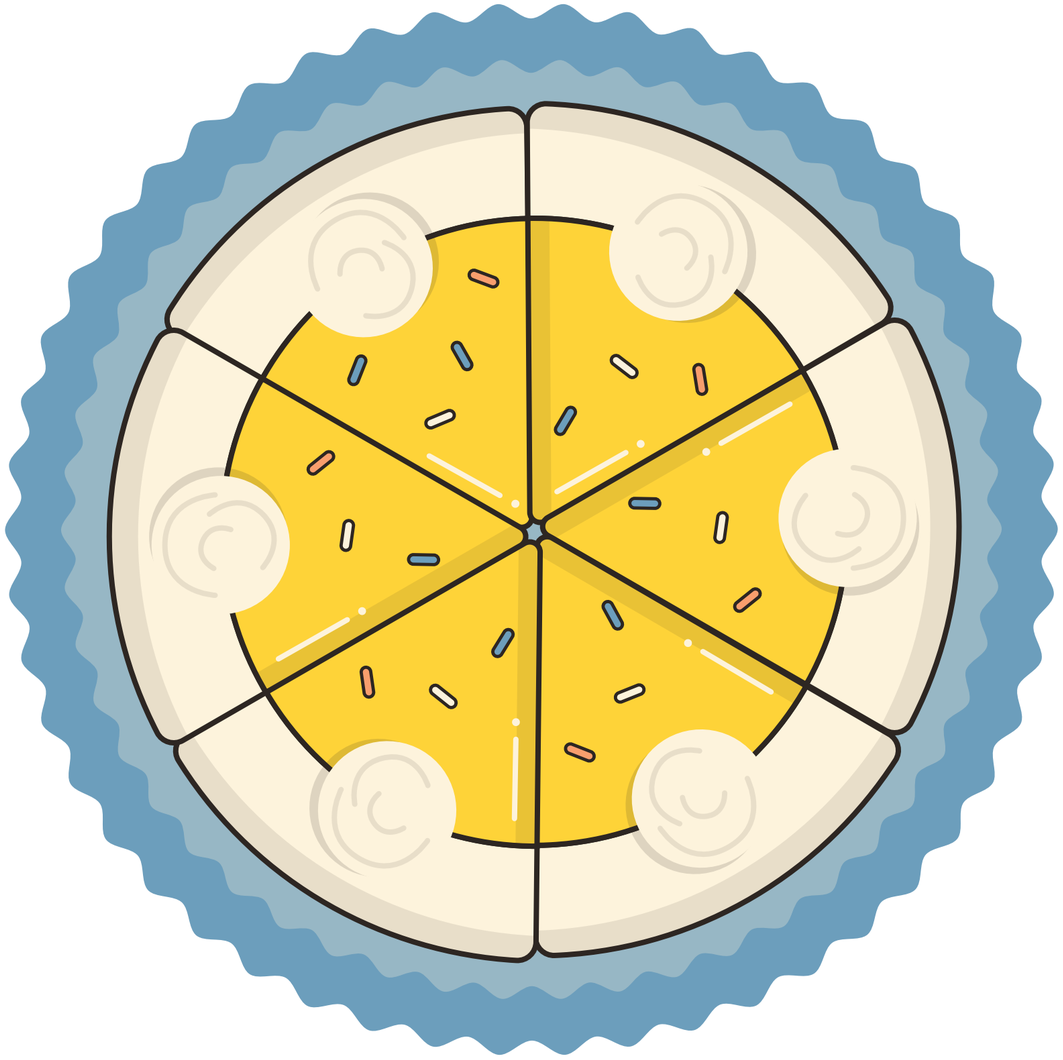
ANSWER

```sql
-- Calculate the total Revenue generated from pizza sales
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

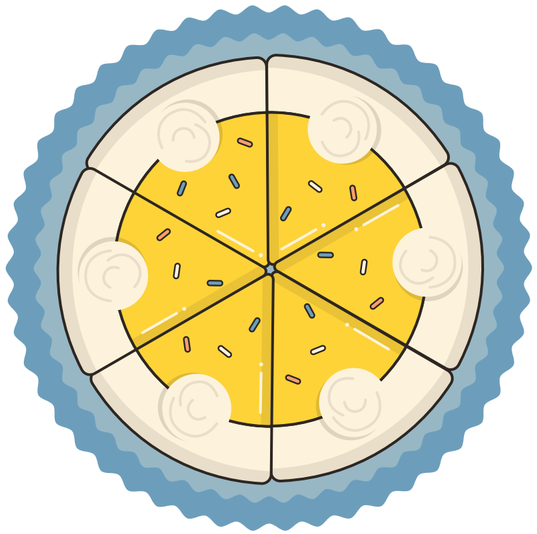
ROUND(SUM(order_details.quantity * pizzas.price), 2)
817860.05

3



Identify the highest priced pizza.

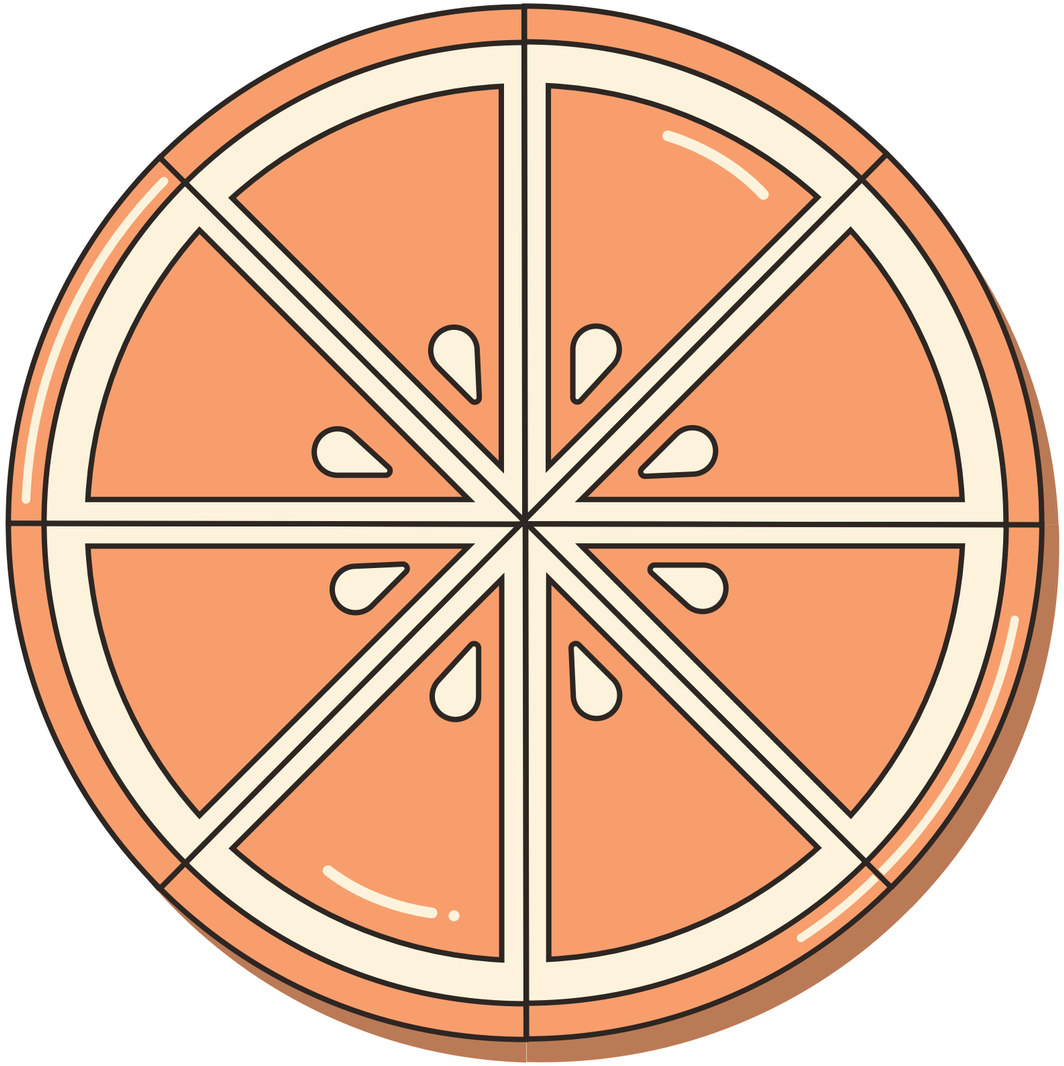
ANSWER



```sql
-- 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: | Export: | Wrap Cell Content: | Fetch rows:

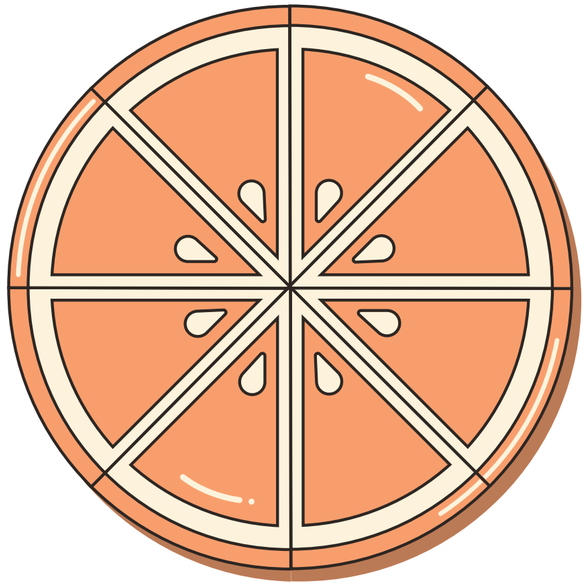
name	price
The Greek Pizza	35.95



4

Identify the most common pizza size ordered

ANSWER



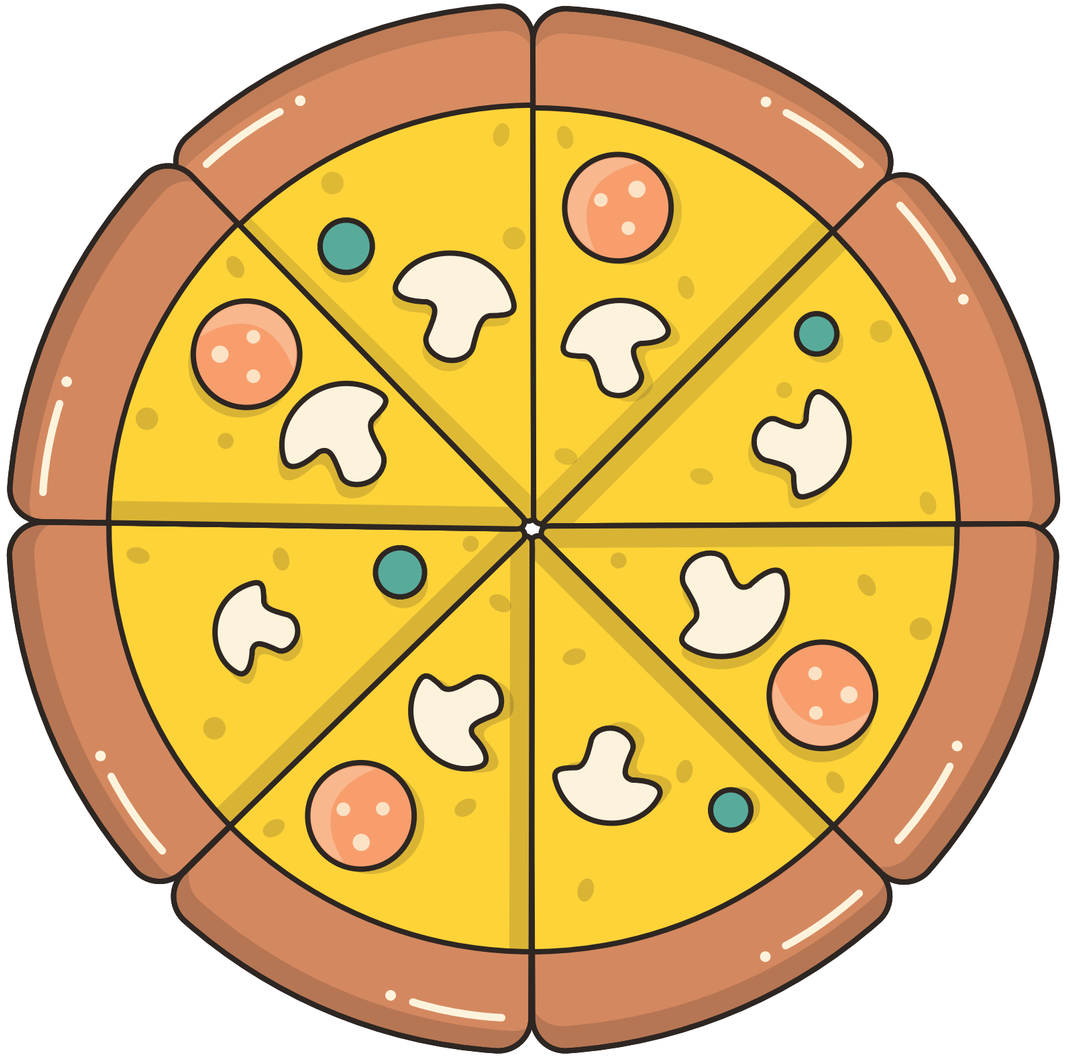
```sql
-- Identify the most common pizza size ordered

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 pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

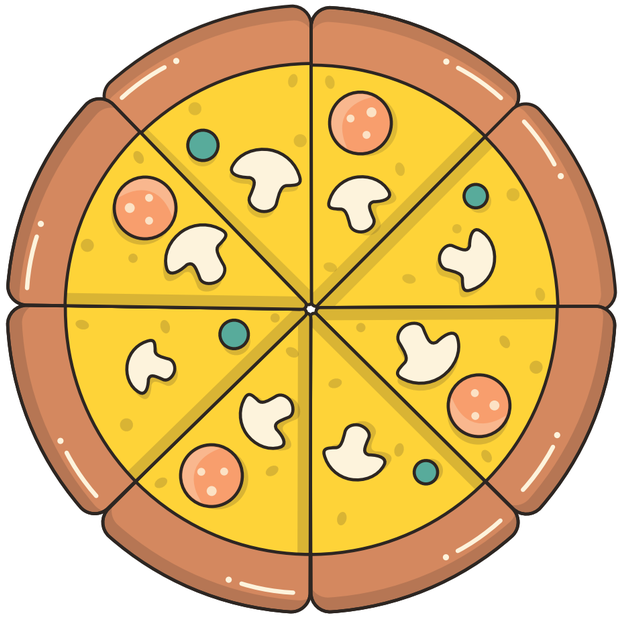
size	order_count
L	18526

5

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



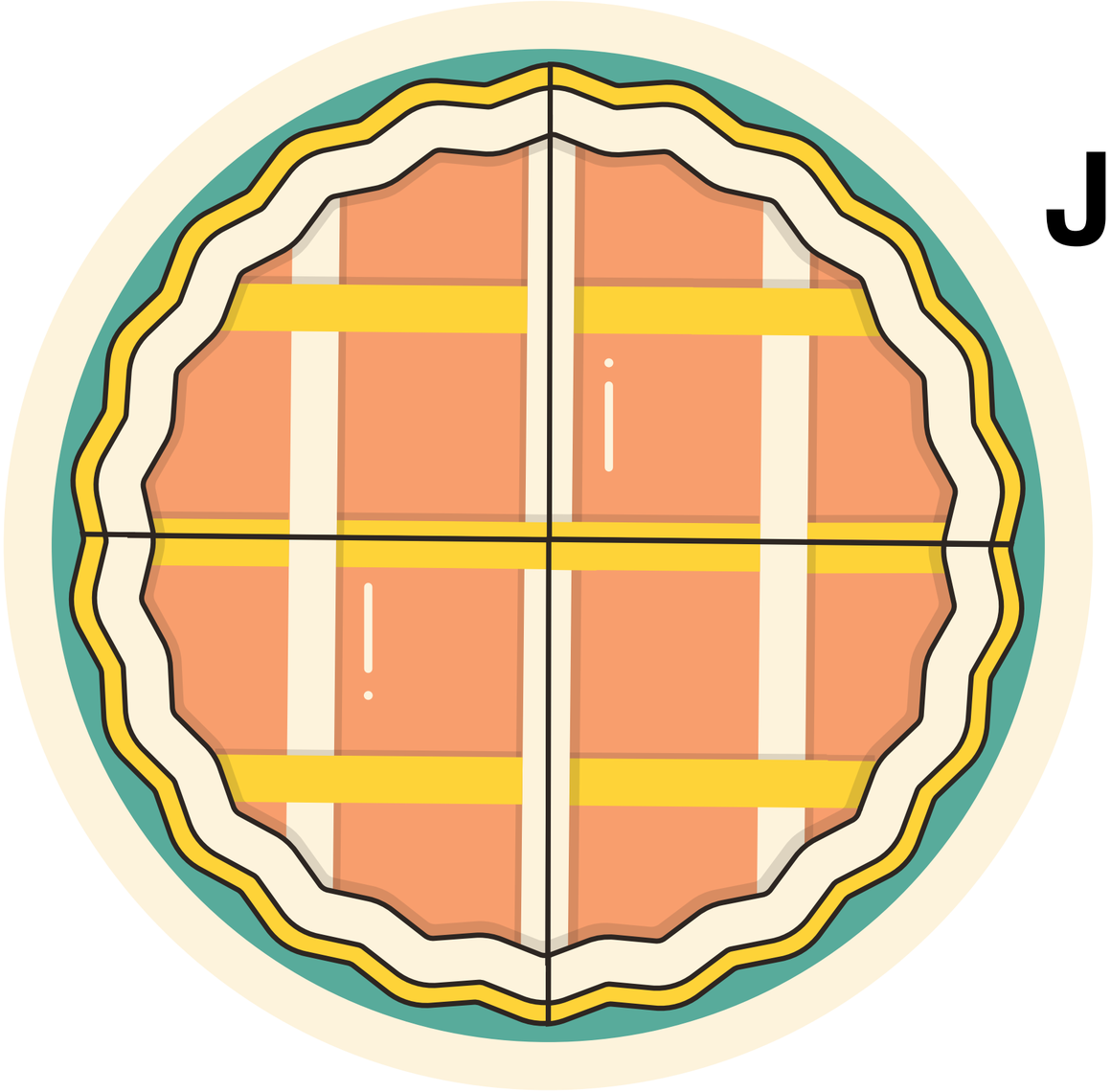
ANSWER



```sql
-- 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

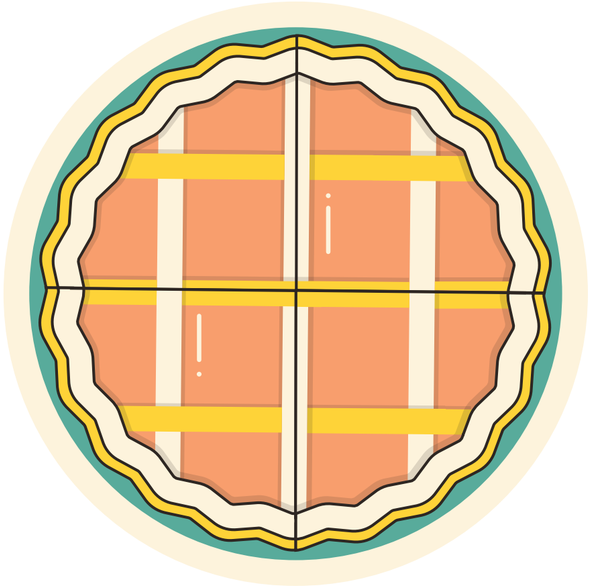
name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



6

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

ANSWER

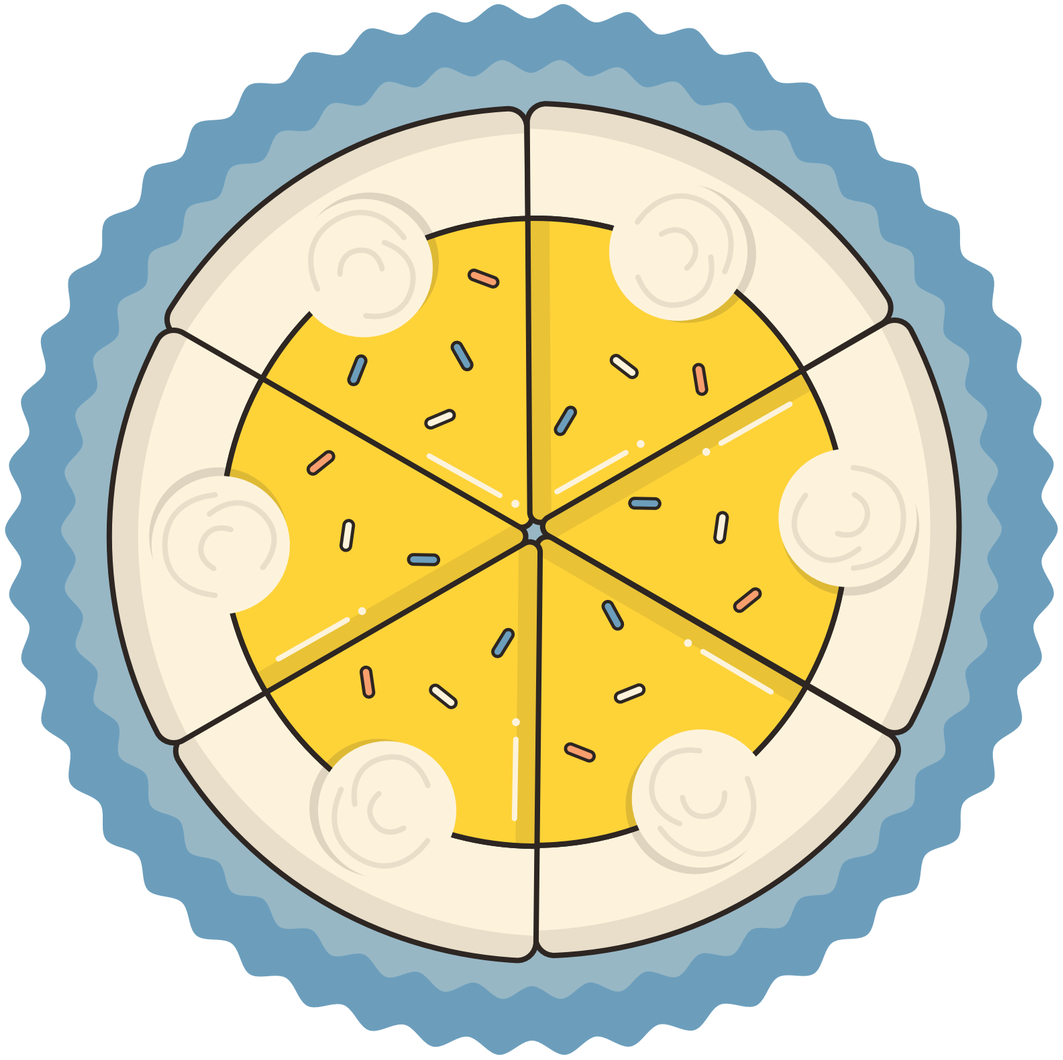


```sql
-- Join the necessary tables to find the total quantity of each pizza category ordered.

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

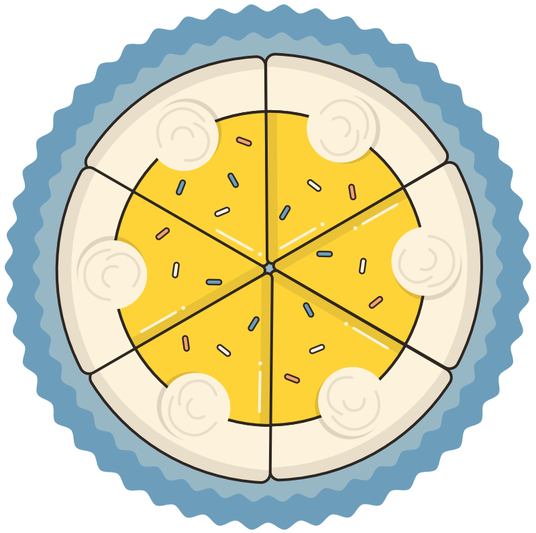
category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7



Determine the distribution of orders by hour of the day

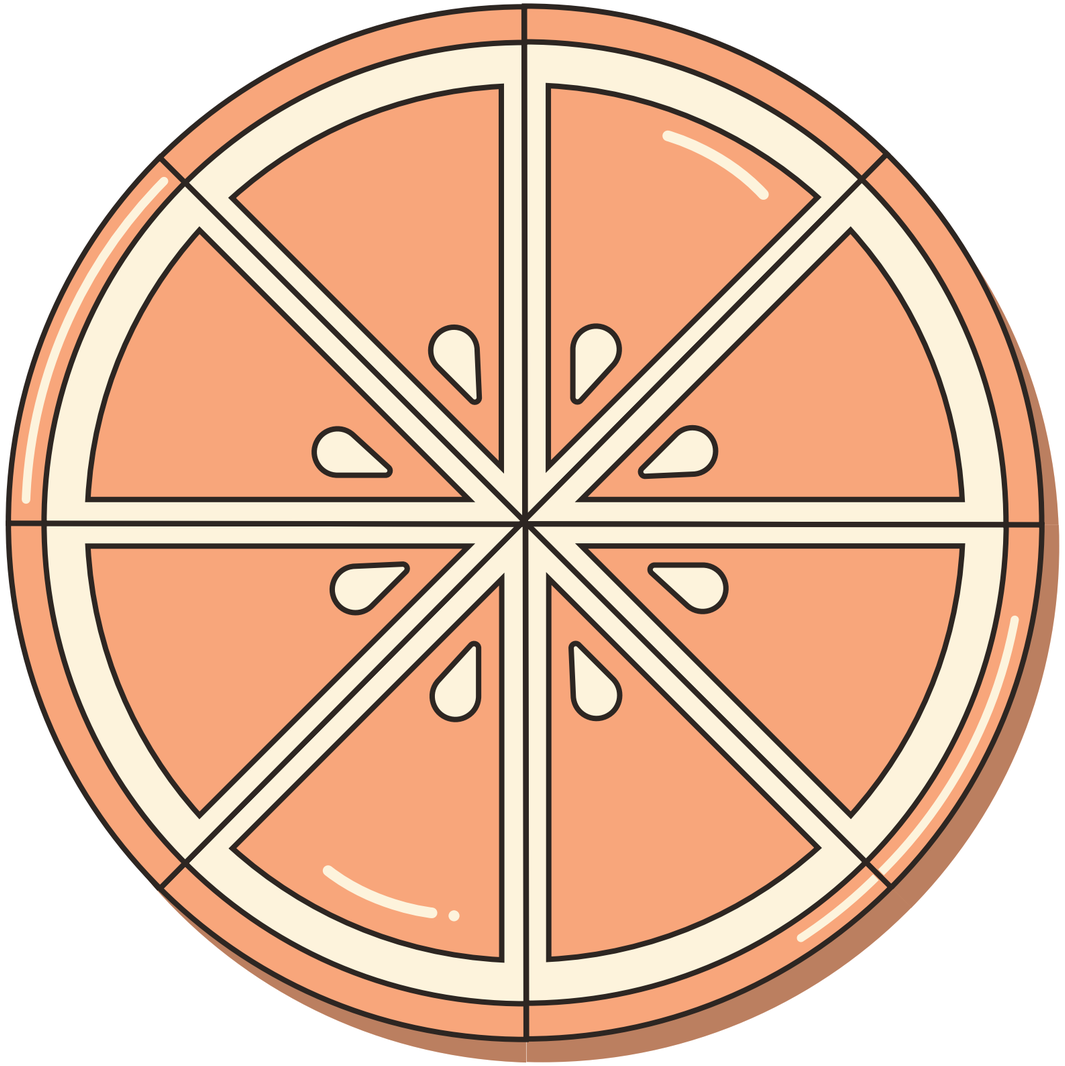
ANSWER



```sql
-- Determine the distribution of orders by hour of the day
SELECT
    HOUR(orders.order_time) AS hour, COUNT(order_id) AS count
FROM
    orders
GROUP BY hour
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hour	count
11	1231
12	2520
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



8

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

ANSWER

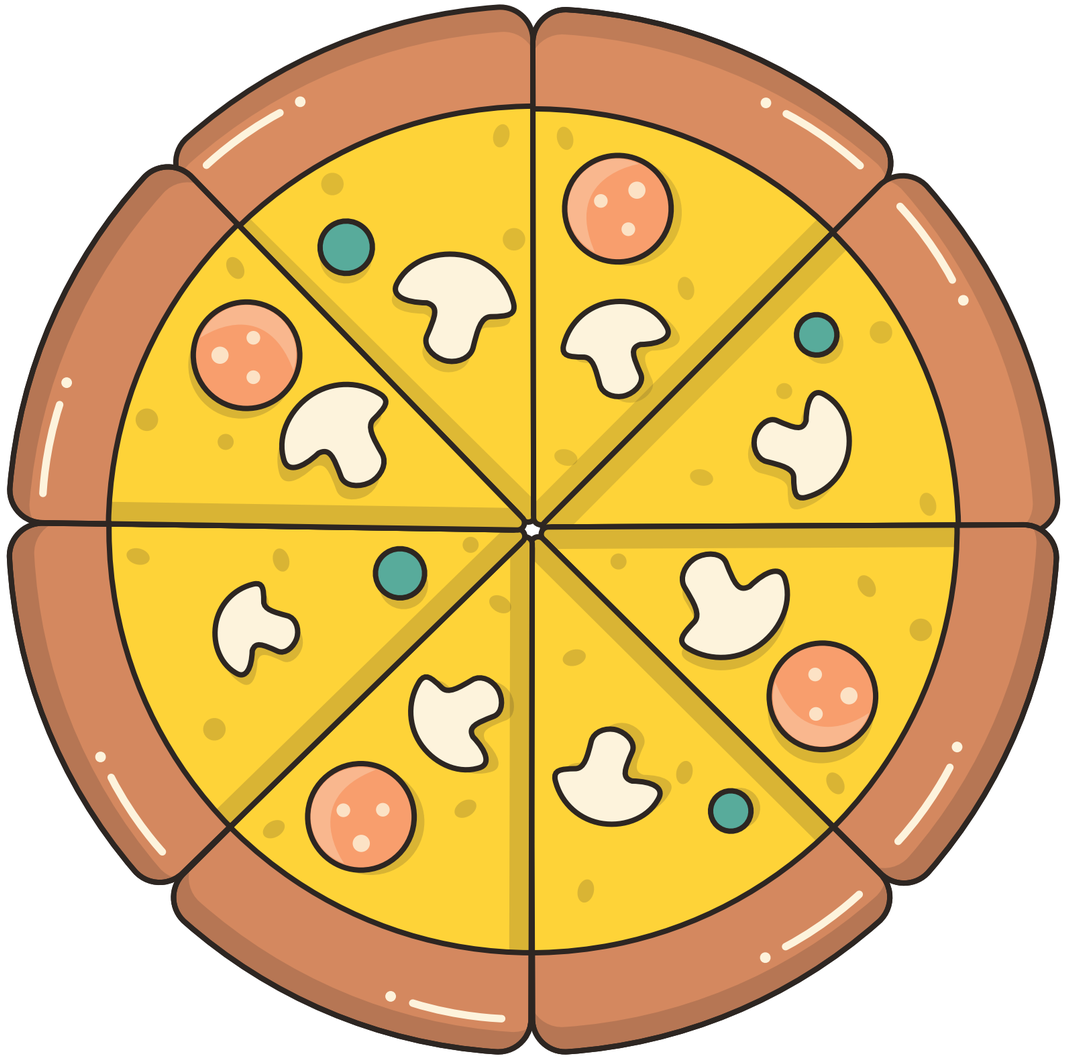
```sql
-- Join relevant tables to find the category-wise distribution of pizzas

SELECT
    pizza_types.category AS category,
    COUNT(pizza_types.pizza_type_id)
FROM
    pizza_types
GROUP BY category
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

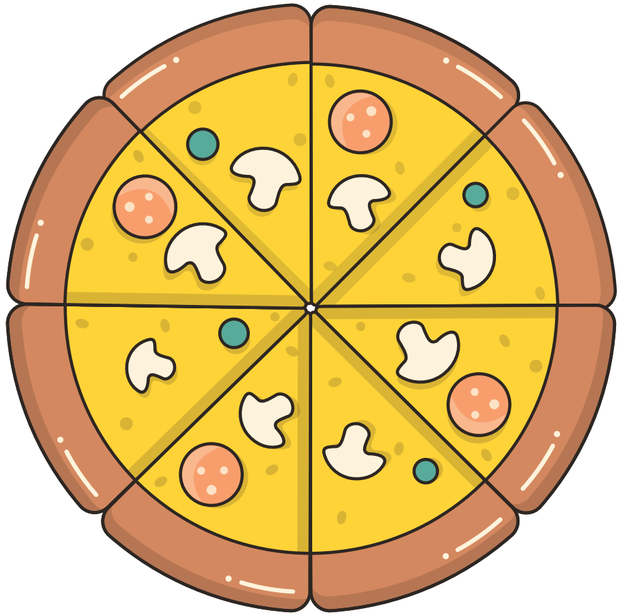
category	COUNT(pizza_types.pizza_type_id)
Chicken	6
Classic	8
Supreme	9
Veggie	9

9

group the orders by date and calculate the average number of pizzas ordered by day



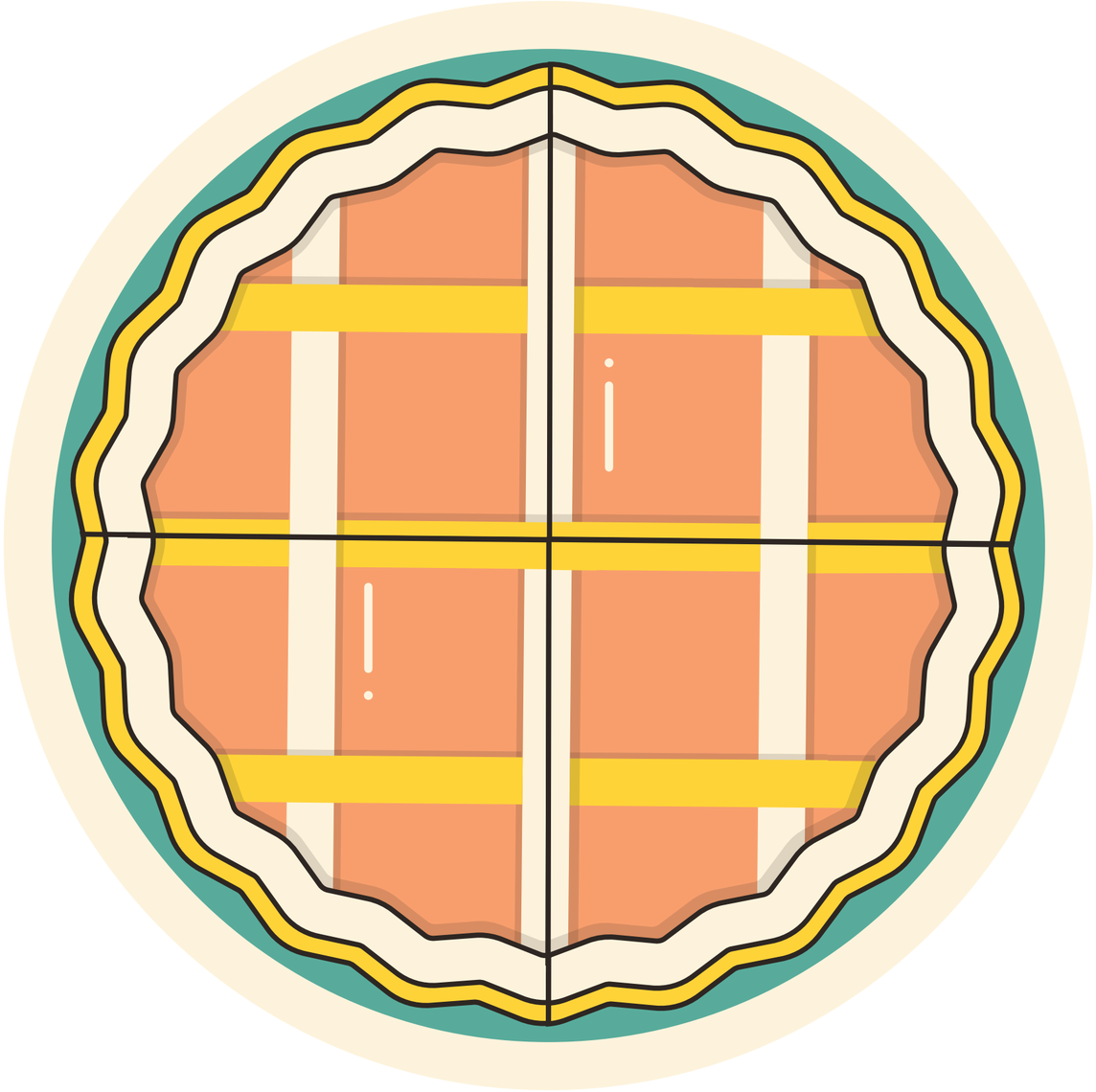
ANSWER



```sql
-- group the orders by date and calculate the average number of pizzas ordered by day

SELECT
    ROUND(AVG(quantity), 0) AS average_pizzas_sold_per_day
FROM
    (SELECT
        orders.order_date AS date,
            COUNT(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) AS sub_table;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

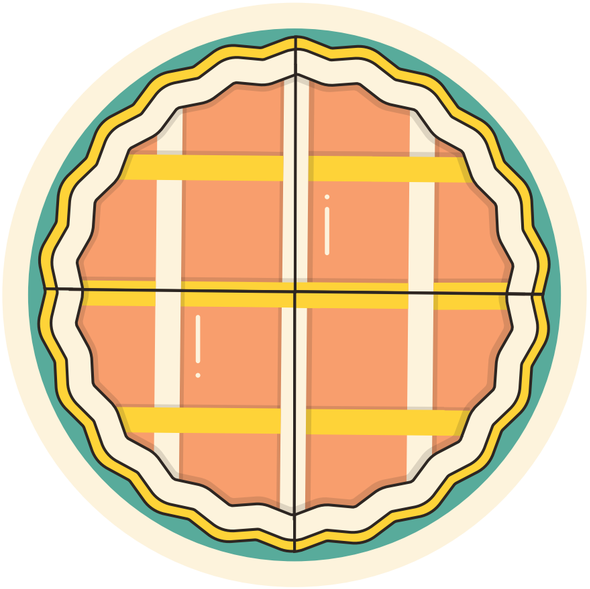
average_pizzas_sold_per_day
136



10

Determine the top 3 most pizzas which has produced the most revenue

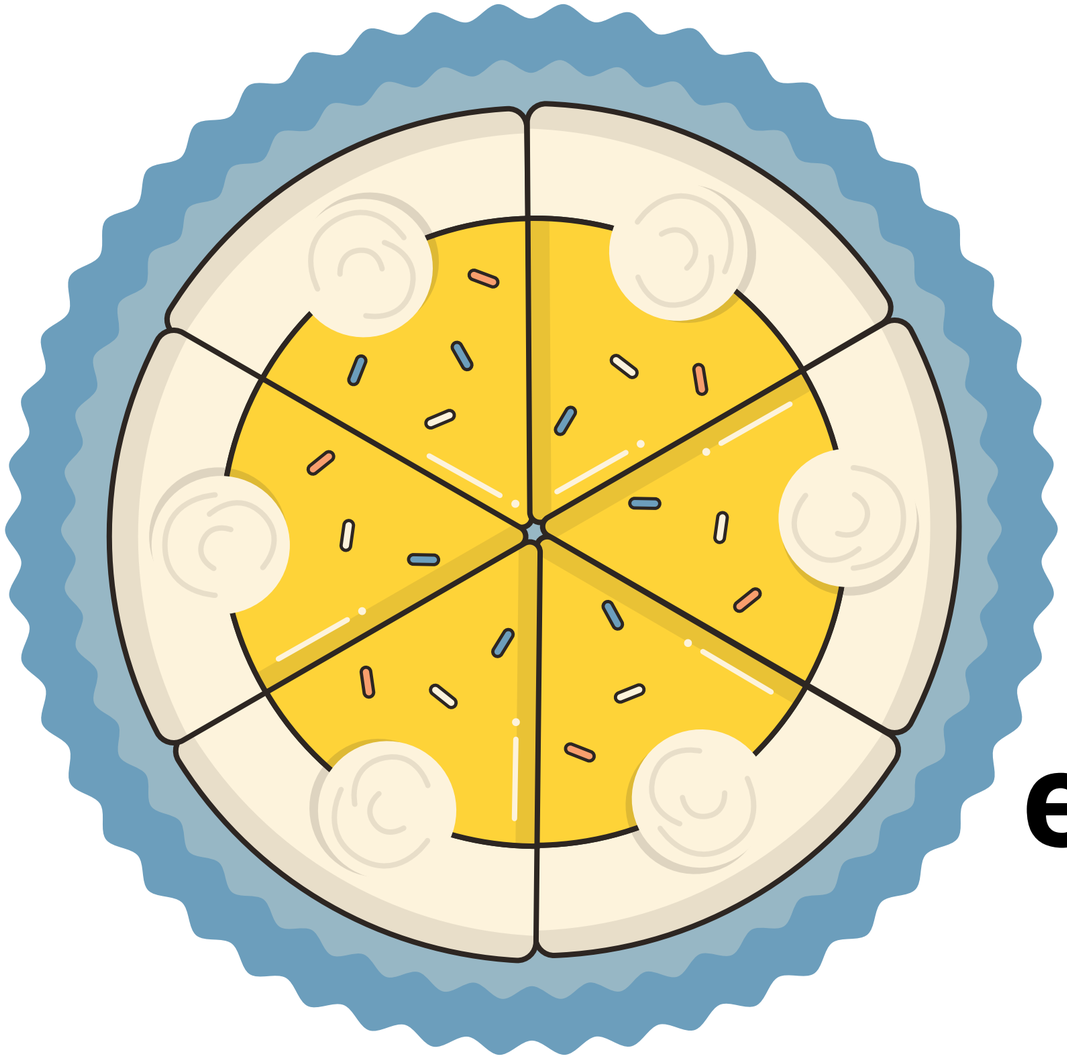
ANSWER



```sql
-- Determine the top 3 most pizzas which has produced the most revenue

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

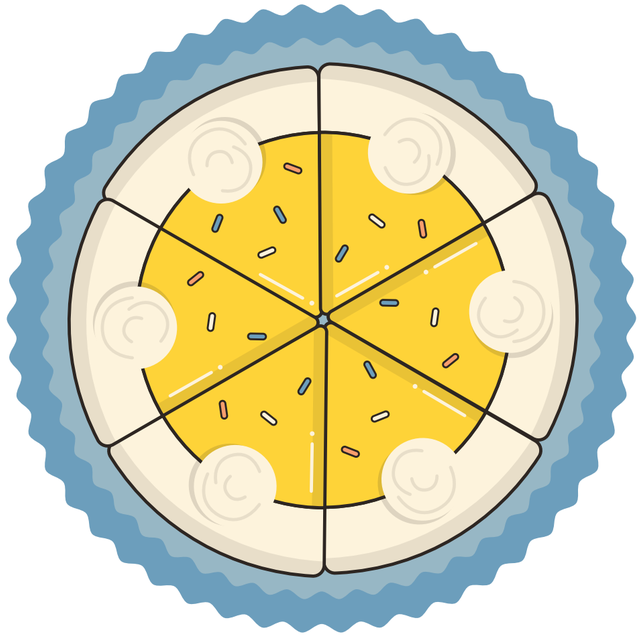
pizza_name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5



11

Calculate the percentage contribution of each pizza category to total revenue

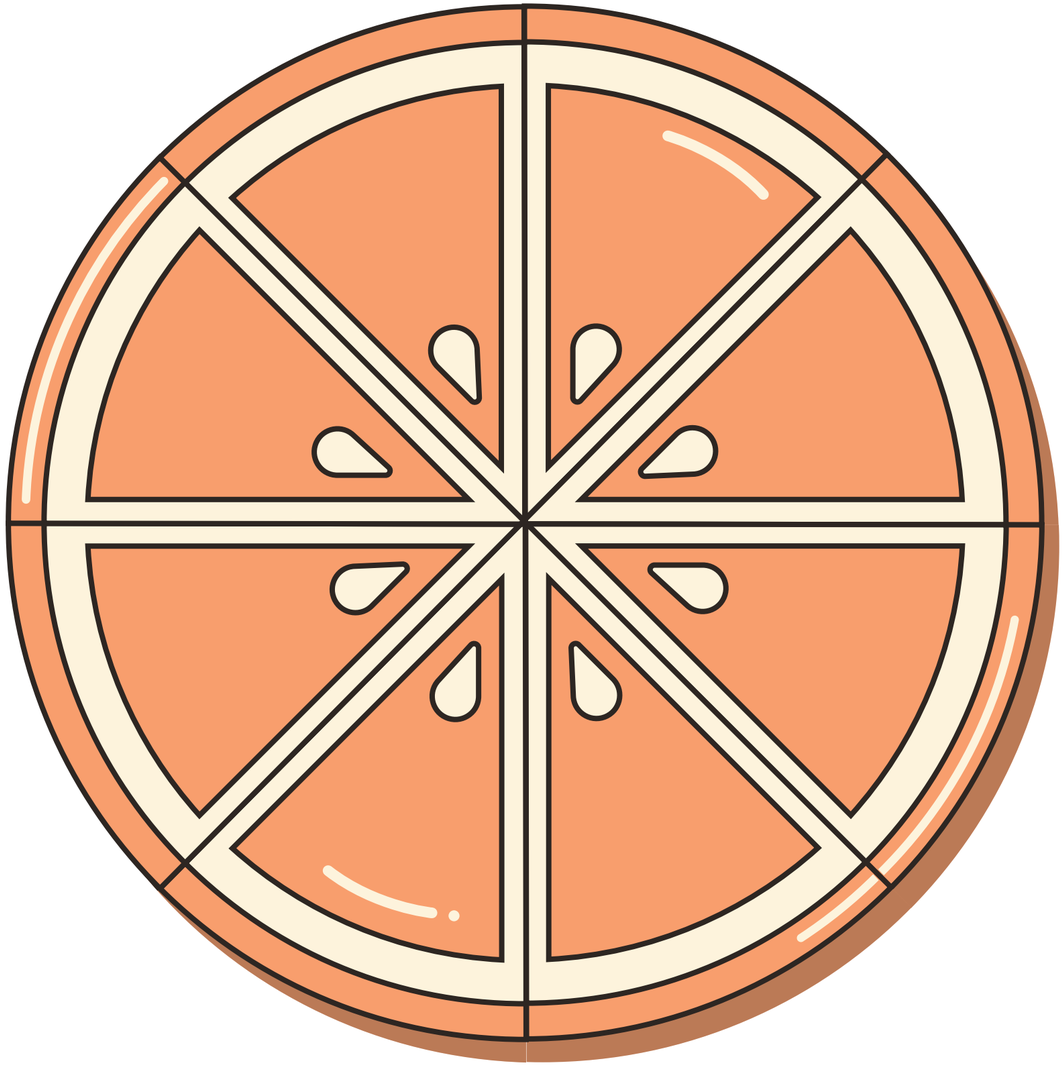
ANSWER



```sql
-- Calculate the percentage contribution of each pizza category 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
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 category
ORDER BY percentage DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

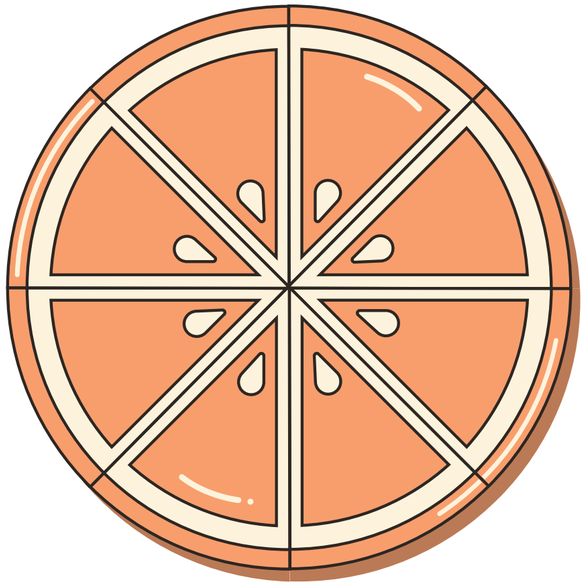
category	percentage
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



12

Analyze the cumulative revenue generated over time

ANSWER



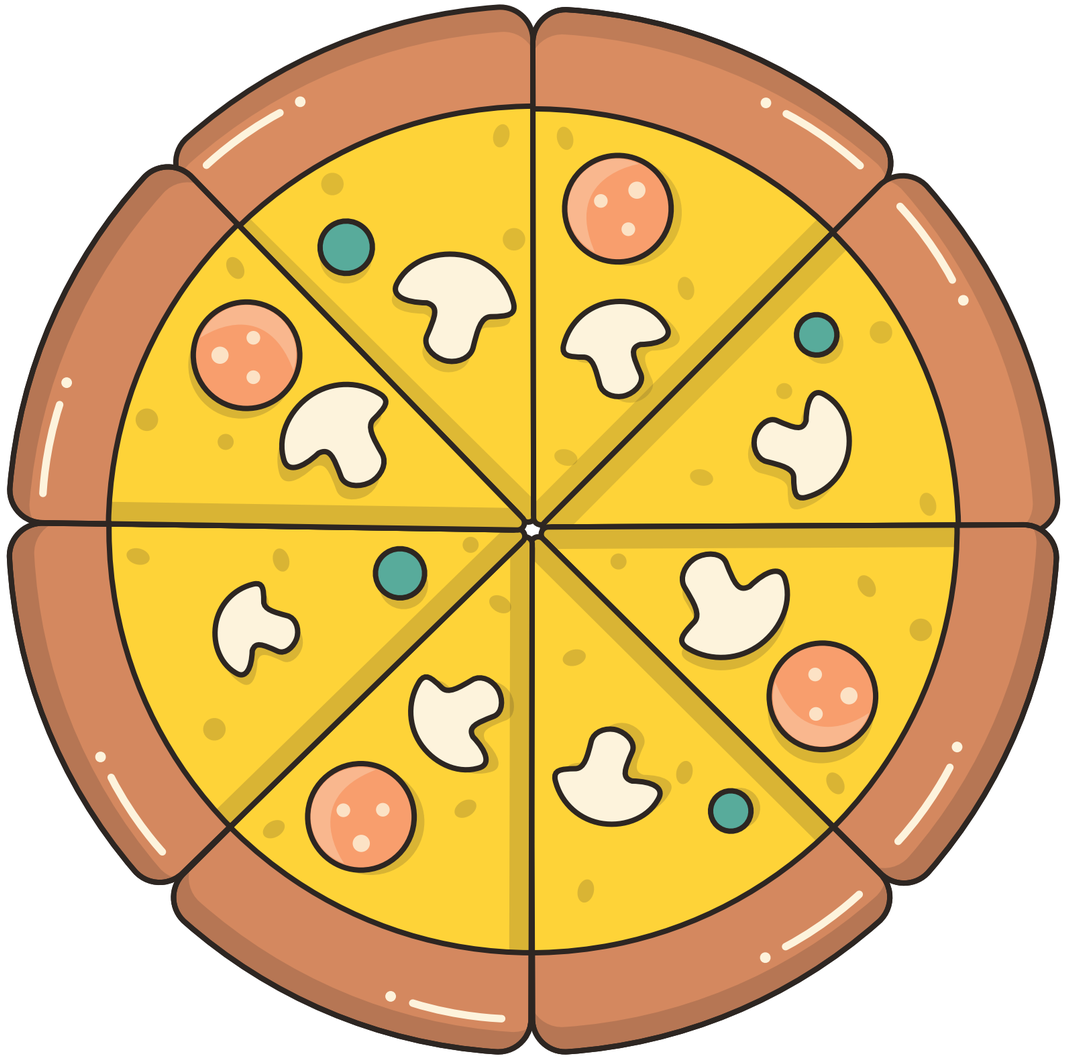
```sql
-- Analyze the cumulative revenue generated over time

SELECT
    day,
    sum(total_revenue) OVER(order by day) AS cummulative_revenue
FROM
    (SELECT
        orders.order_date AS day,
        SUM(order_details.quantity * pizzas.price) AS total_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 day) AS sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

day	cummulative_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	34343.50000000001

13

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



ANSWER

```sql
-- Determine the top 3 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 table_rank from
(SELECT
    pizza_types.category AS category,
    pizza_types.name AS 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 category , name) AS grid) AS final_table
 where table_rank <=3;
```

	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.700
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5