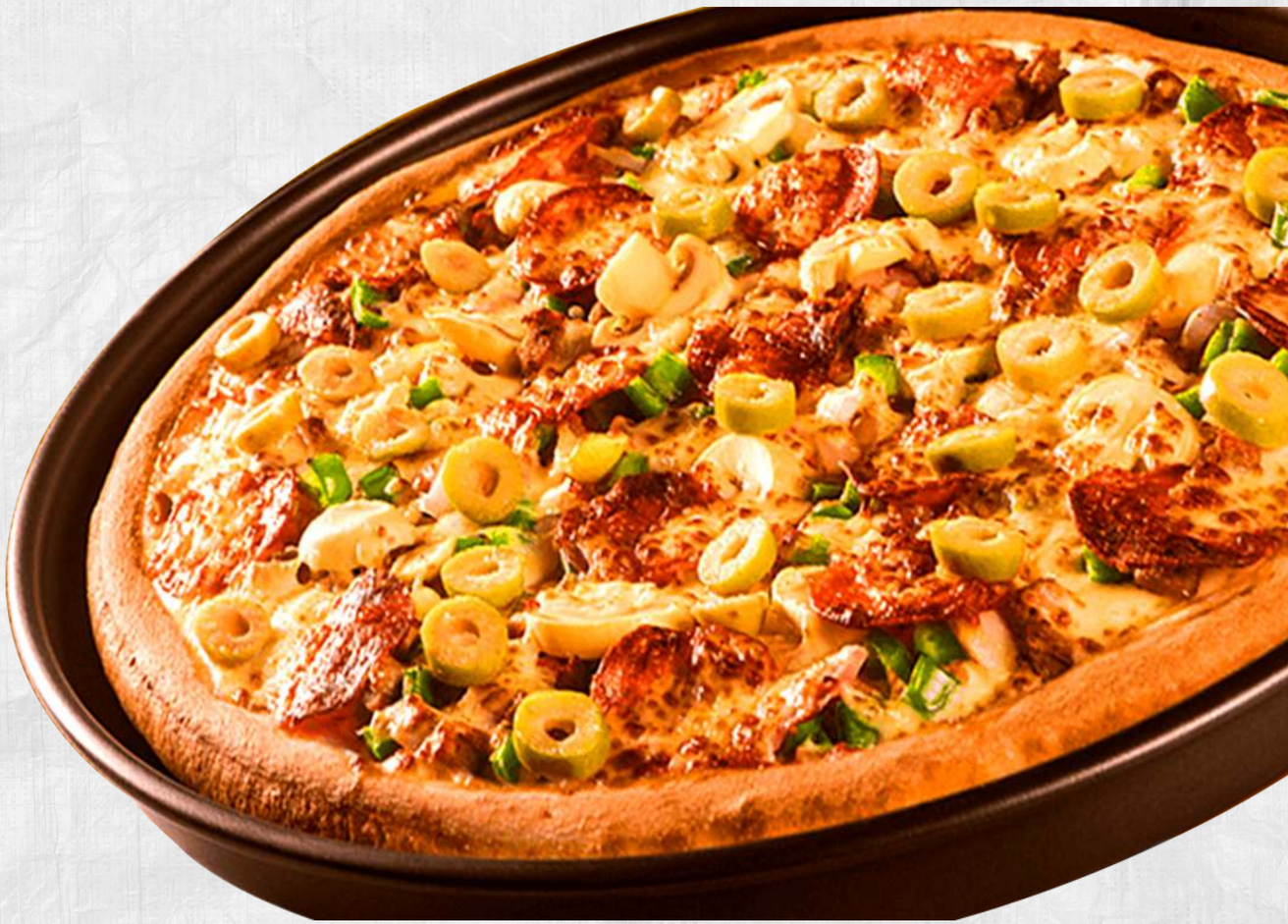
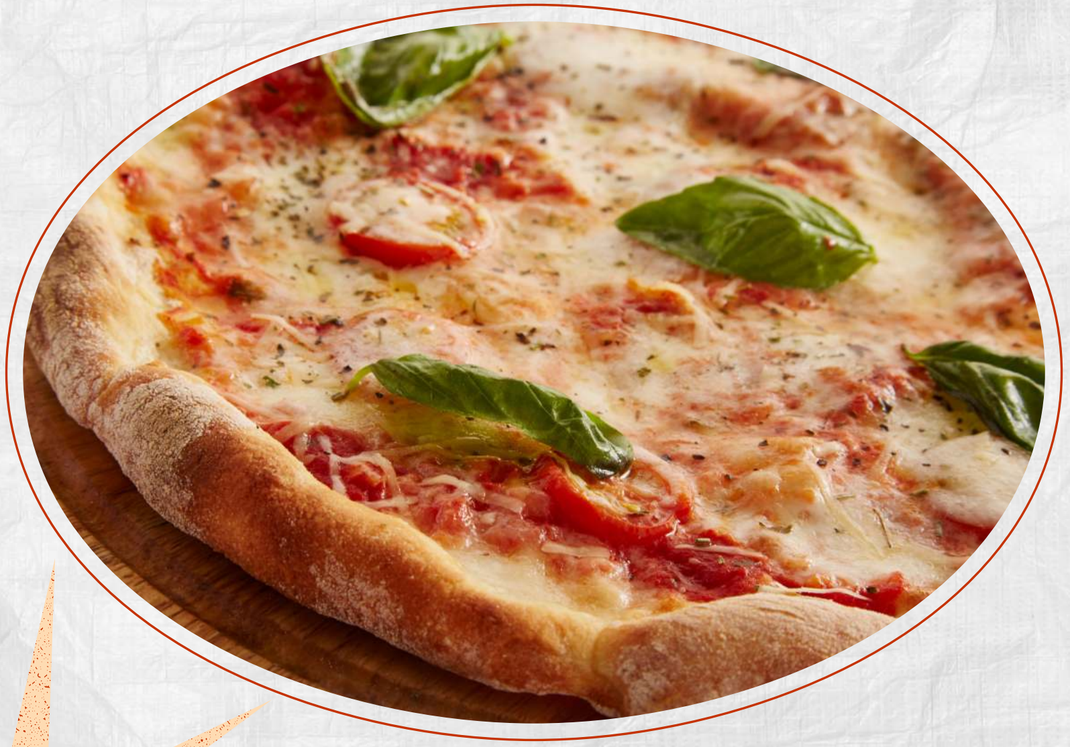
Pizza Sales Analysis Using SQL





Hello there!

The aim of this project is to analyze pizza sales data to uncover insights that can help optimize the business performance and make data-driven decisions.

SCHEMA

PIZZAS

pizza_id	text
pizza_type_id	text
size	text
price	double

ORDER_DETAILS

order_details_id	int
order_id	int
pizza_id	text
quantity	int

PIZZA_TYPES

pizza_type_id	text
name	text
category	text
ingredients	text

ORDERS

order_id	int
order_date	date
order_time	time

1. Retrieve the total number of orders placed.

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

Result Grid

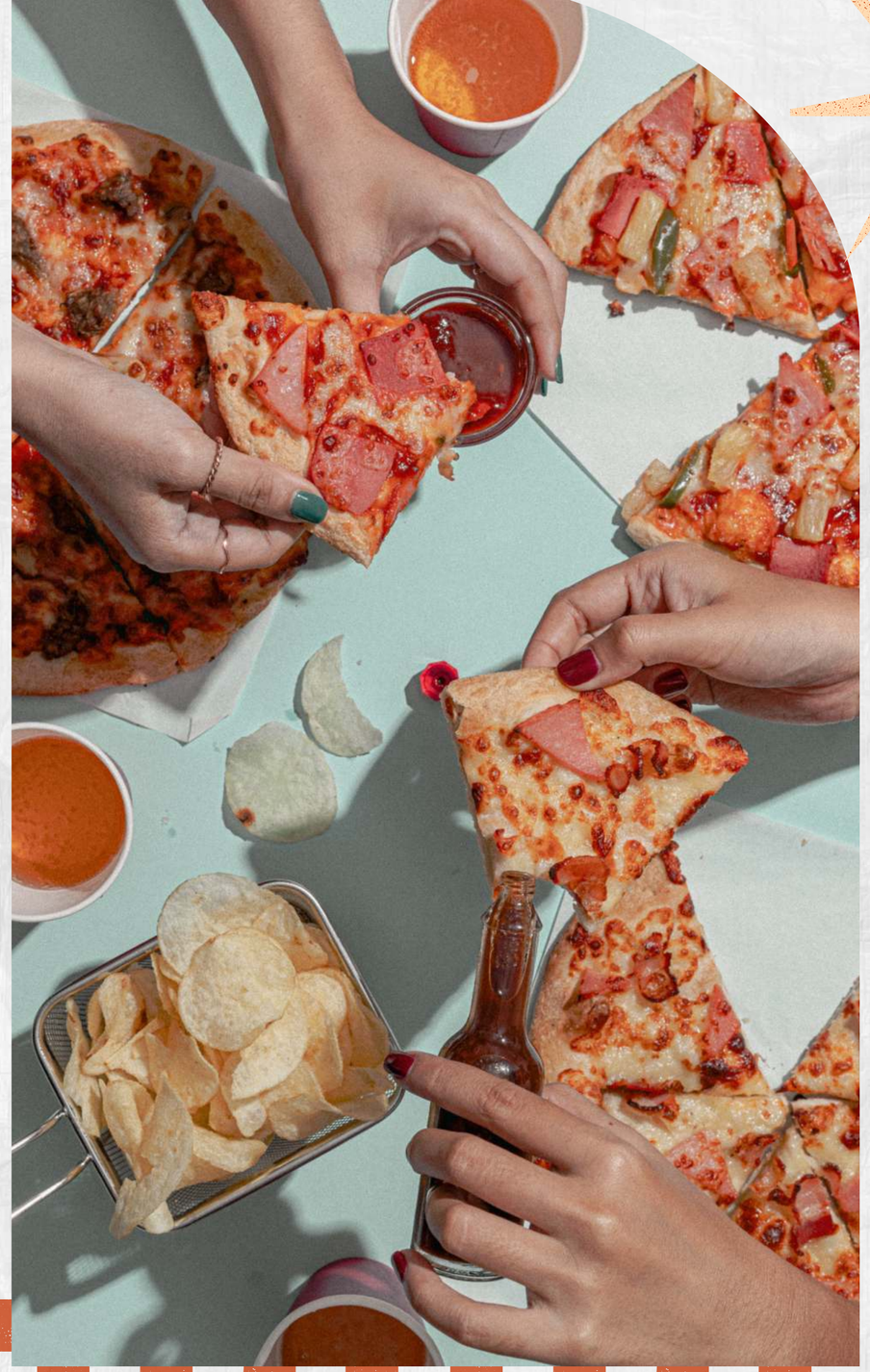
total_orders	21350

2. Calculate the total revenue generated from pizza sales.

```
SELECT
ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
2) AS TOTAL_SALES
FROM
ORDER_DETAILS
INNER JOIN
PIZZAS ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID;
```

Result Grid

total_sales	817860.05



3. Identify the highest-priced pizza.

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

Result Grid

The Greek Pizza	35.95

4. Identify the most common pizza size ordered.

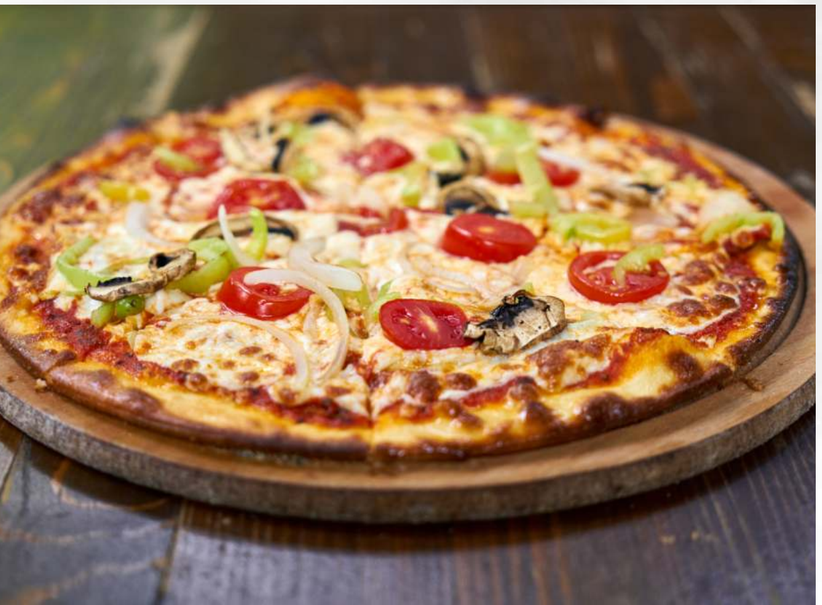
```
SELECT
(pizzas.size),
COUNT(order_details.order_details_id) AS order_count
FROM
pizzas
INNER JOIN
order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid

Size	Order_count
L	18526

5. 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;
```

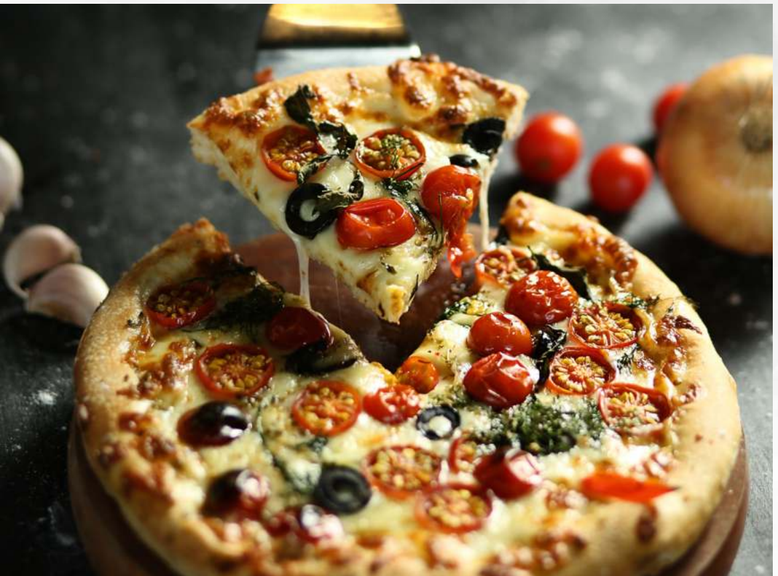


Result Grid

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.

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



Result Grid

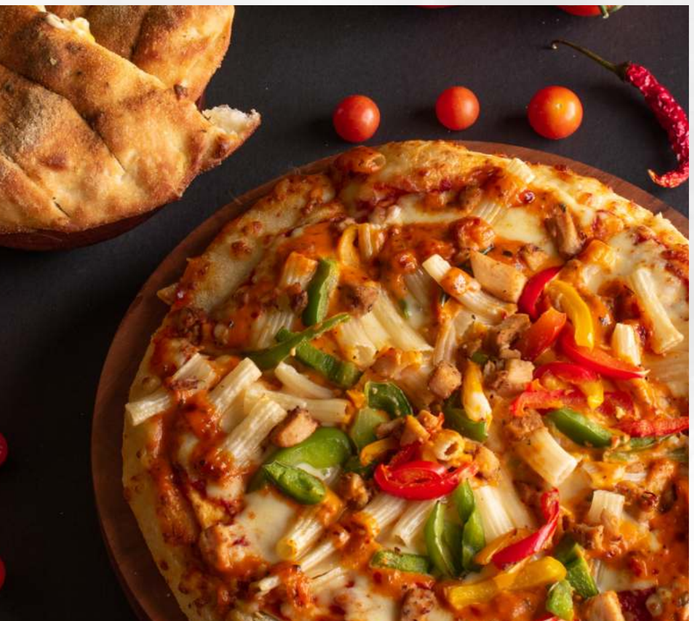
category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

7. 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

hour	order_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

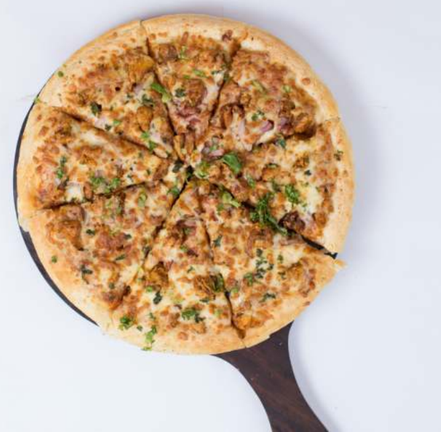


8. Join relevant tables to find the category-wise distribution of pizzas.

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

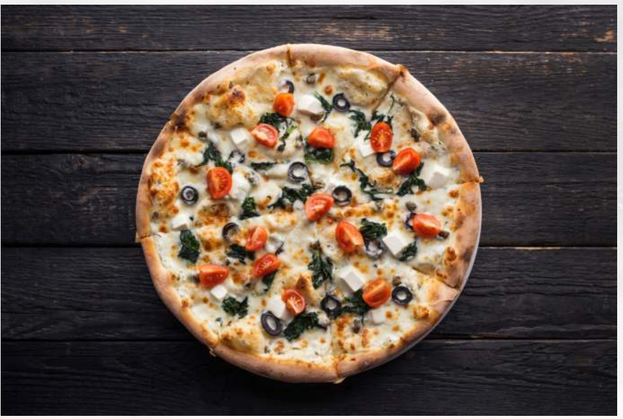
Result Grid

category	count(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 per day.

```
SELECT
ROUND(AVG(quantity), 0) as avg_pizza_ordered_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;
```

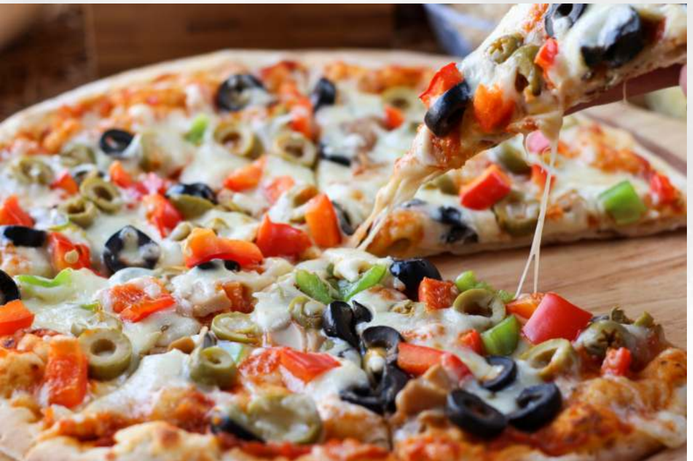


Result Grid

avg_pizza_ordered_per_day
138

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

```
select pizza_types.name, sum(pizzas.price*order_details.quantity) 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 limit 3;
```

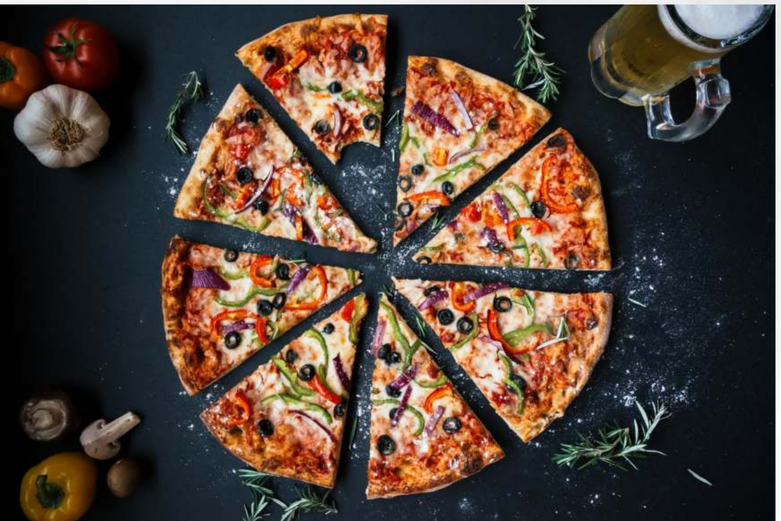


Result Grid

name	revenue
The Brie Carre Pizza	11588.4999999999
The Green Garden Pizza	13955.75
The Spinach Supreme Pizza	15277.75

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

```
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
INNER JOIN
pizzas ON order_details.pizza_id = pizzas.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;
```

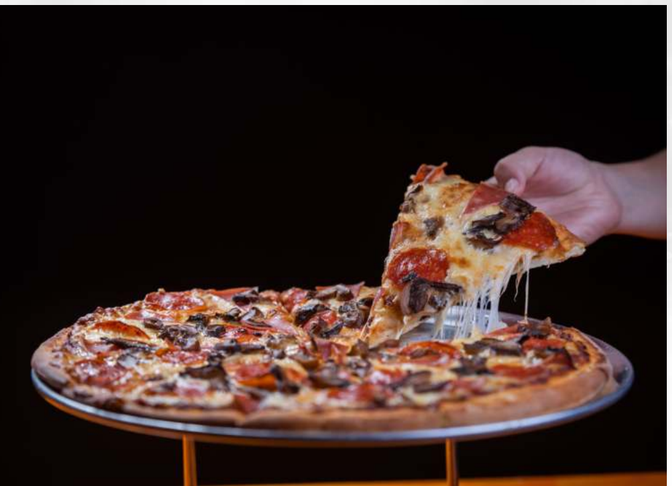


Result Grid

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

12. Analyze 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(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;
```

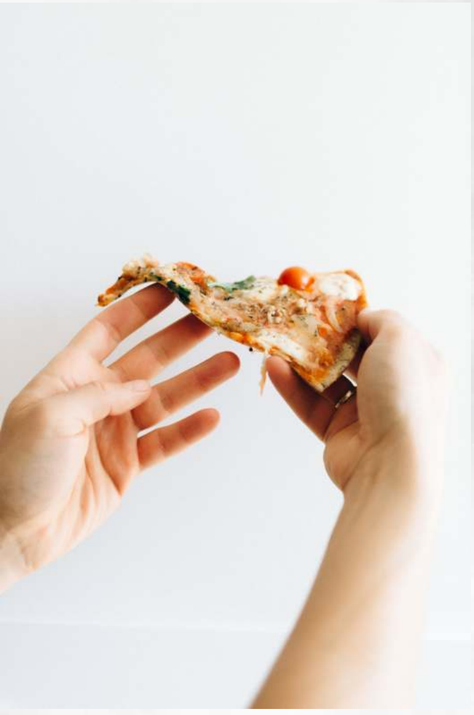


Result Grid

2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15

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

```
select name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) 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;
```



Result Grid

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

THANK YOU!