```
pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in c:\users\
atharv\appdata\local\programs\python\python313\lib\site-packages
(9.4.0)
Note: you may need to restart the kernel to use updated packages.
```

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import mysql.connector


db = mysql.connector.connect(host = "localhost",
                    username = "root",
                    password = "Atharv_22310552_new",
                    database = "pizzahut")

cur = db.cursor()
```

Retrieve the total number of orders placed

```python
query = """ SELECT
    COUNT(order_id) AS total_orders
FROM
    orders; """

cur.execute(query)
data = cur.fetchall()

"The total number of ordered placed are ",data[0][0]
```

```
('The total number of ordered placed are ', 21350)
```

Calculate the total revenue generated from pizza sales.

```python
query = """ SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id """

cur.execute(query)
```

```
data = cur.fetchall()

"Total revenue generated from Pizza sales is ",data[0][0]

('Total revenue generated from Pizza sales is ', 817860.05)
```

Identify the highest-priced pizza.

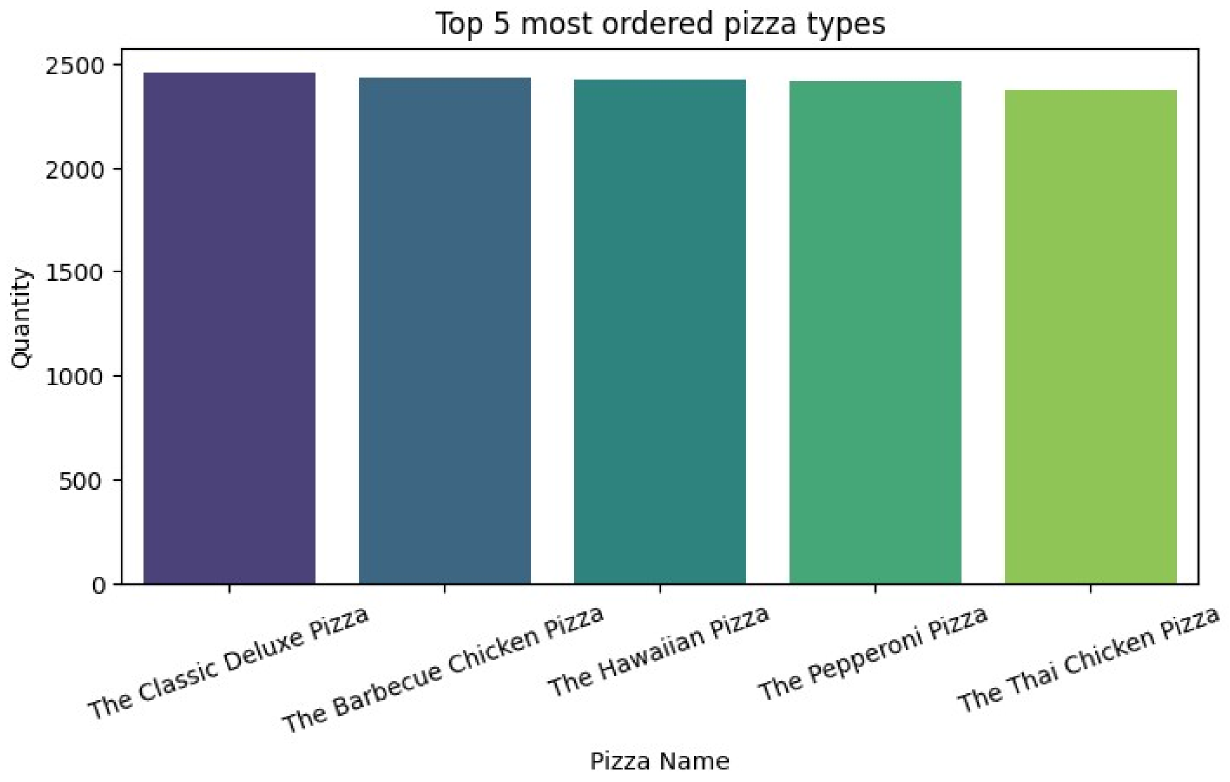
```
query = """ SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1; """

cur.execute(query)
data = cur.fetchall()

"The highest priced pizza is",data[0][0]

('The highest priced pizza is', 'The Greek Pizza')
```

Identify the most common pizza size ordered.

```
query = """ SELECT
    pizzas.size, COUNT(orders_details.order_details_id) AS Total
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY size
ORDER BY total DESC
LIMIT 1; """

cur.execute(query)
data = cur.fetchall()

"The most common pizza size ordered is",data[0][0]

('The most common pizza size ordered is', 'L')
```

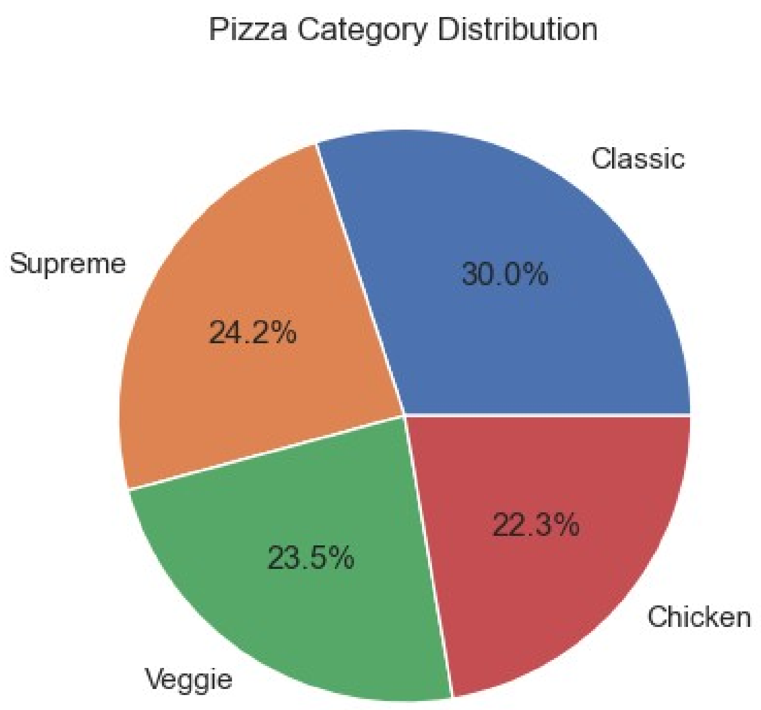
List the top 5 most ordered pizza types along with their quantities.

```
query = """ SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS total_quantity
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 pizza_types.name
ORDER BY total_quantity DESC
LIMIT 5; """

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza Name", "Quantity"])

plt.figure(figsize = (8,4))
sns.barplot(x = df["Pizza Name"], y = df["Quantity"], data = df, hue =
df["Pizza Name"], legend = False, palette = 'viridis')
plt.xticks(rotation = 20)
plt.title("Top 5 most ordered pizza types")
plt.show()
```



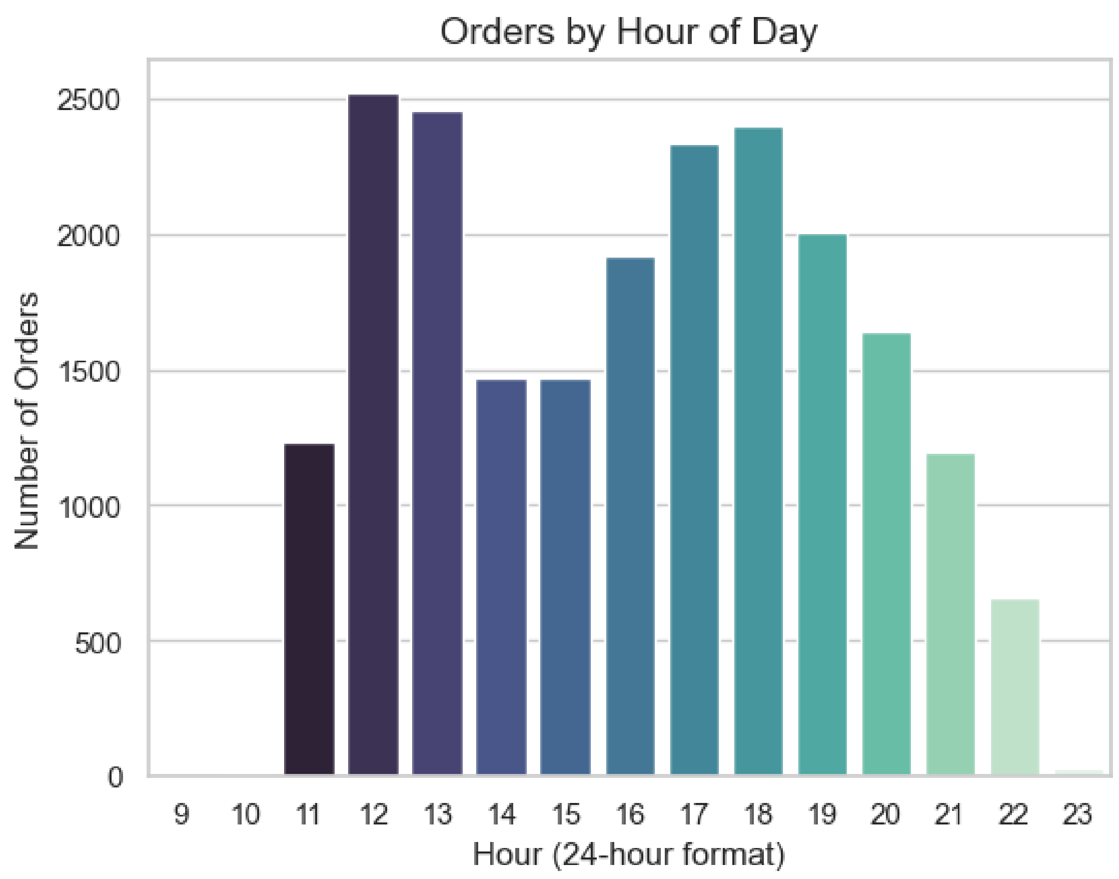
Join the necessary tables to find the total quantity of each pizza category ordered.

```
query = """ SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS category_sum
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 pizza_types.category
ORDER BY category_sum DESC; """

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza Category", "Quantity"])

plt.pie(df["Quantity"], labels=df["Pizza Category"], autopct="%1.1f%
%")
plt.title("Pizza Category Distribution")
plt.show()
```



Determine the distribution of orders by hour of the day.

```
query = """ SELECT
    HOUR(order_time), COUNT(order_id) AS total_orders
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY total_orders DESC;"""

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Hour", "Total Orders"])

sns.barplot(x="Hour", y="Total Orders", data=df,hue = df["Hour"],
legend = False, palette="mako")

plt.title("Orders by Hour of Day", fontsize=14)
plt.xlabel("Hour (24-hour format)")
plt.ylabel("Number of Orders")
plt.show()
```

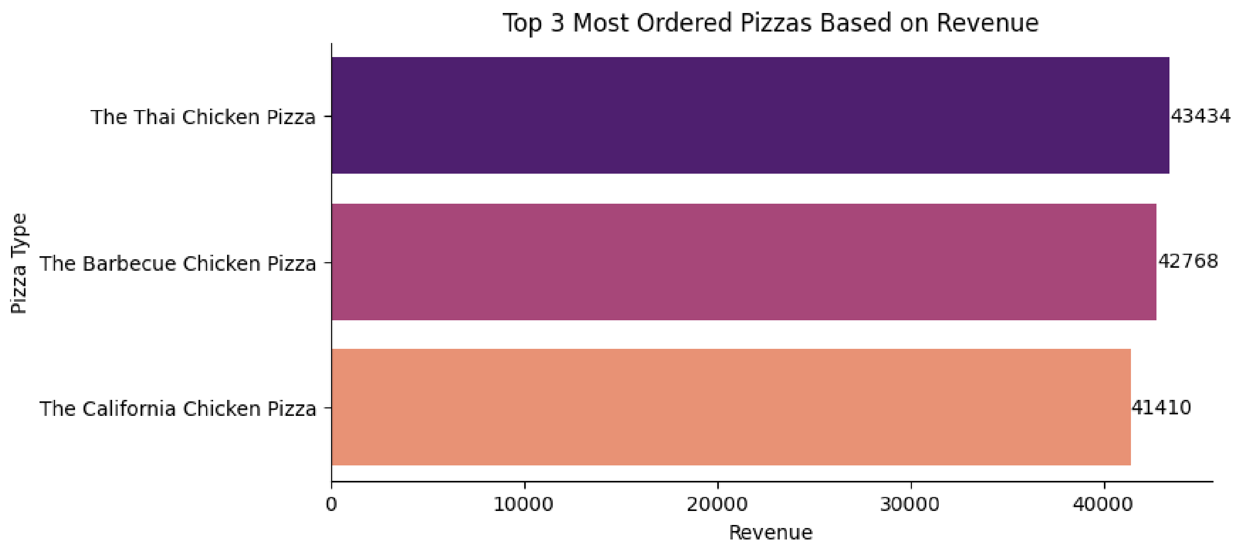


Group the orders by date and calculate the average number of pizzas ordered per day.

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

cur.execute(query)
data = cur.fetchall()
"Average number of pizza orders per day are",int(data[0][0])

('Average number of pizza orders per day are', 138)
```

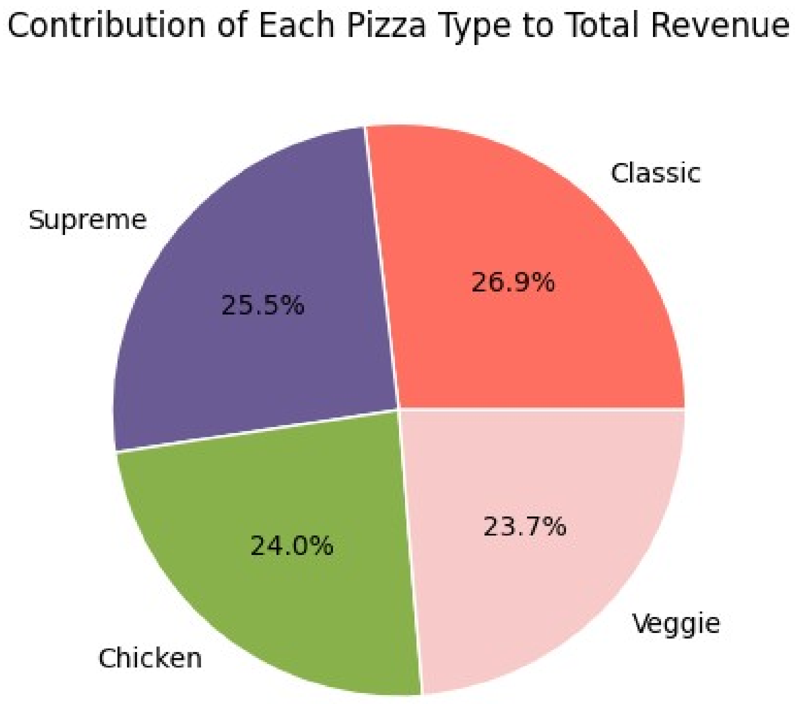
Determine the top 3 most ordered pizza types based on revenue.

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

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza Name", "Revenue"])

plt.figure(figsize = (8,4))
ax = sns.barplot(y="Pizza Name", x="Revenue", data=df, hue=df["Pizza
Name"], legend=False, palette="magma")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
ax.bar_label(ax.containers[2])
sns.despine(left=False, bottom=False)
plt.title("Top 3 Most Ordered Pizzas Based on Revenue")
plt.xlabel("Revenue")
plt.ylabel("Pizza Type")
plt.show()
```



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

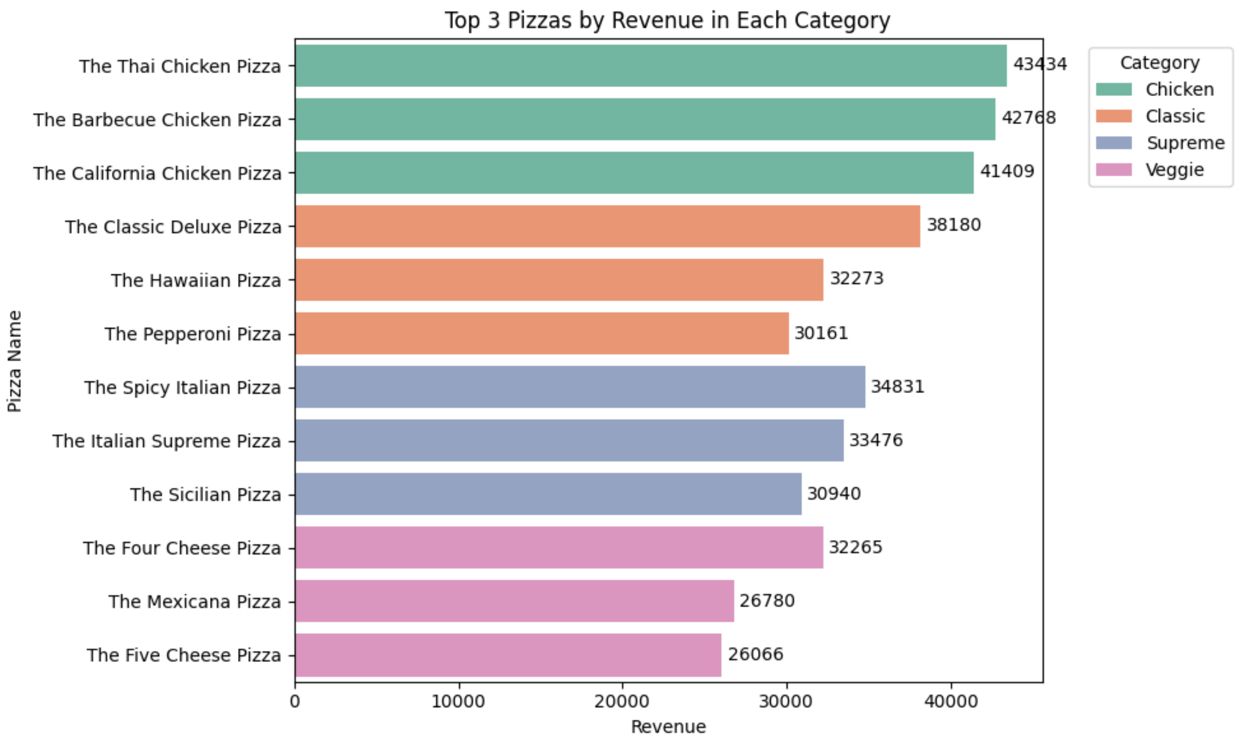
```
query = """ SELECT
    pizza_types.category,
    ROUND(ROUND(SUM(orders_details.quantity * pizzas.price),
                    0) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                0)
                FROM
                    orders_details
                        JOIN
                    pizzas ON pizzas.pizza_id =
orders_details.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
"""

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza Category", "Revenue"])

colors = ['#FF6F61', '#6B5B95', '#88B04B', '#F7CAC9', '#92A8D1',
'#955251', '#B565A7']
plt.pie(df["Revenue"], labels=df["Pizza Category"], autopct="%1.1f%%",
colors=colors[:len(df)],wedgeprops={'edgecolor': 'white'})
plt.title("Contribution of Each Pizza Type to Total Revenue")
plt.show()
```



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

```
query = """ select category, 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(orders_details.quantity*pizzas.price) 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 pizza_types.category, pizza_types.name) as a) as b
where rn <= 3;
```

```
"""

cur.execute(query)
data = cur.fetchall()
df = pd.DataFrame(data, columns = ["Pizza Category", "Pizza Name",
"Revenue"])


plt.figure(figsize=(10, 6))
ax = sns.barplot(
    data=df,
    x="Revenue",
    y="Pizza Name",
    hue="Pizza Category",
    dodge=False,
    palette="Set2"
)


for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=3)


plt.title("Top 3 Pizzas by Revenue in Each Category")
plt.xlabel("Revenue")
plt.ylabel("Pizza Name")
plt.legend(title="Category", bbox_to_anchor=(1.05, 1), loc='upper
left')
plt.tight_layout()
plt.show()
```


Top 3 Pizzas by Revenue in Each Category
Pizza Name
The Thai Chicken Pizza
The Barbecue Chicken Pizza
The California Chicken Pizza
The Classic Deluxe Pizza
The Hawaiian Pizza
The Pepperoni Pizza
The Spicy Italian Pizza
The Italian Supreme Pizza
The Sicilian Pizza
The Four Cheese Pizza
The Mexicana Pizza
The Five Cheese Pizza
43434
42768
41409
38180
32273
30161
34831
33476
30940
32265
26780
26066
0
10000
20000
30000
40000
Revenue
Category
Chicken
Classic
Supreme
Veggie