Database project part 3

E-commerce database

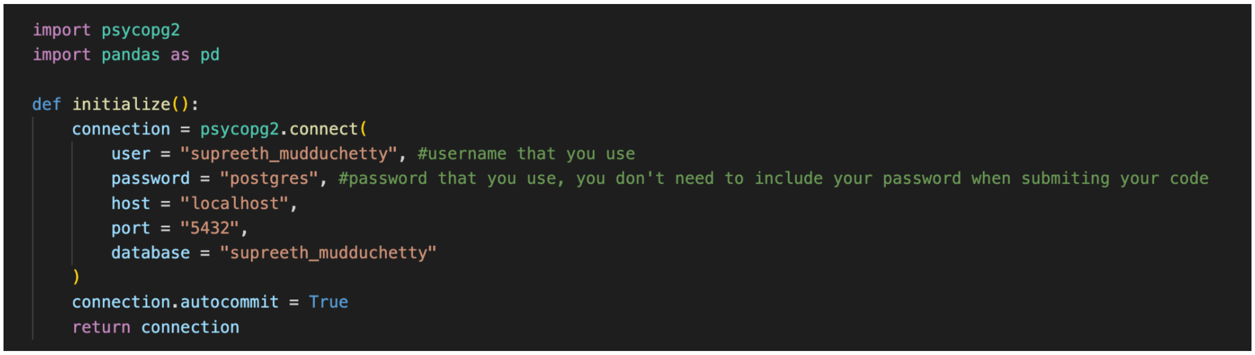
Name: Supreeth Mudduchetty
Email - smuddu2@pdx.edu

New questions designed

1. Which order has the most no of items? Return the order_id and the customer who made it too.
2. Which all orders would not get affected due to the availability of one of the products in the order

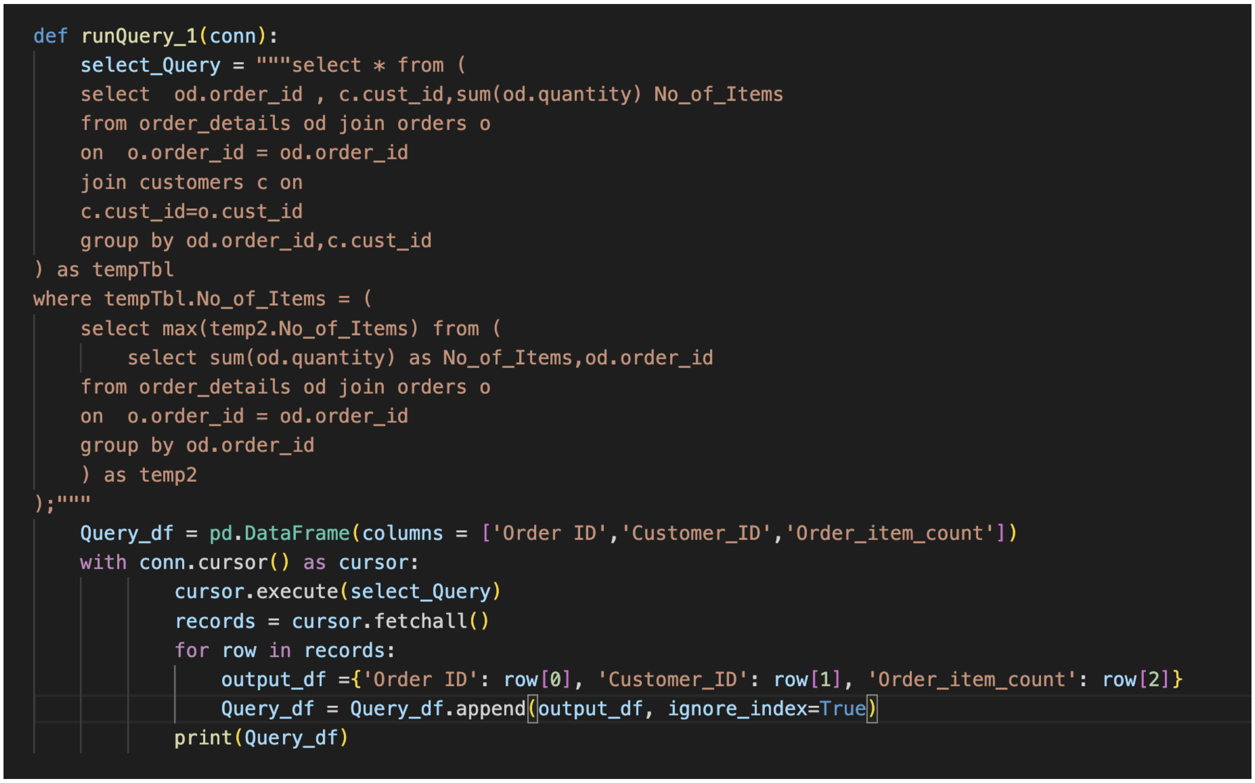
Code

Code to connect to Postgres



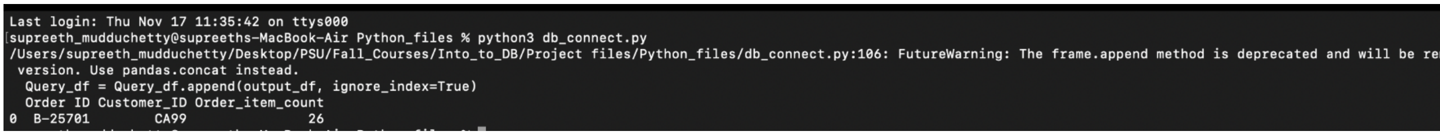
```
import psycopg2
import pandas as pd

def initialize():
    connection = psycopg2.connect(
        user = "supreeth_mudduchetty", #username that you use
        password = "postgres", #password that you use, you don't need to include your password when submiting your code
        host = "localhost",
        port = "5432",
        database = "supreeth_mudduchetty"
    )
    connection.autocommit = True
    return connection
```

Function that runs my query first query



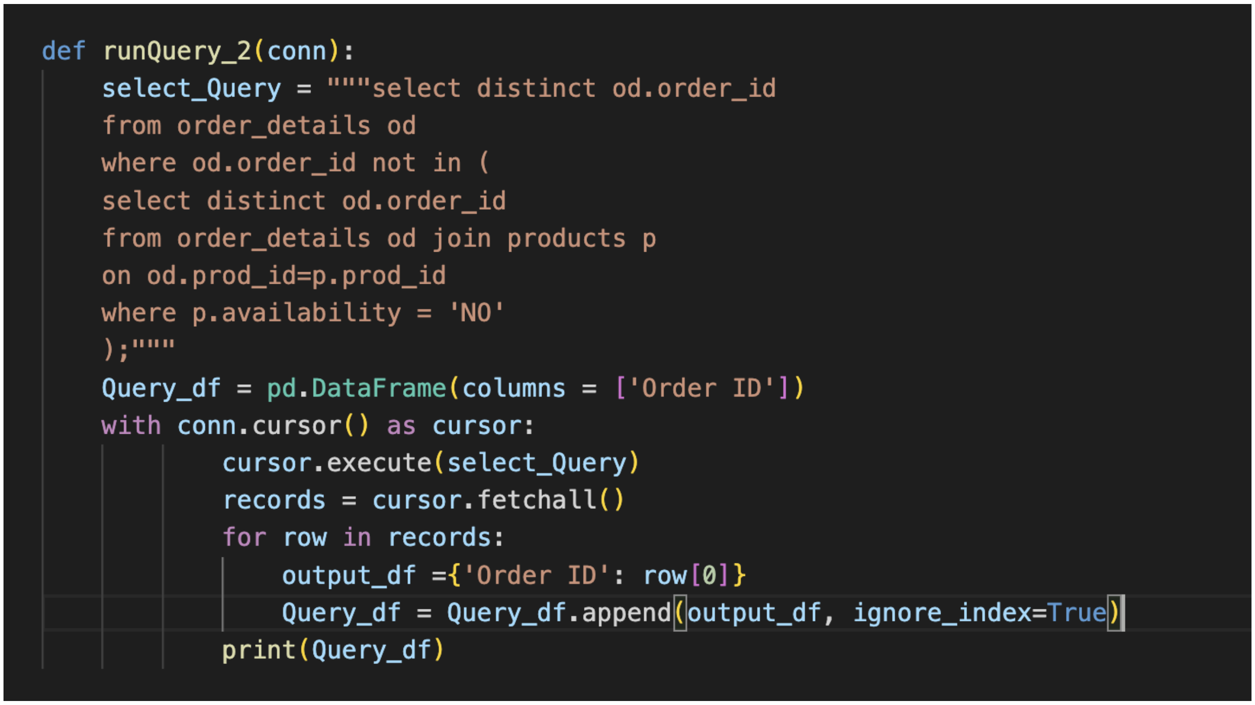
```
def runQuery_1(conn):
    select_Query = """select * from (
    select  od.order_id , c.cust_id,sum(od.quantity) No_of_Items
    from order_details od join orders o
    on  o.order_id = od.order_id
    join customers c on
    c.cust_id=o.cust_id
    group by od.order_id,c.cust_id
) as tempTbl
where tempTbl.No_of_Items = (
    select max(temp2.No_of_Items) from (
        select sum(od.quantity) as No_of_Items,od.order_id
    from order_details od join orders o
    on  o.order_id = od.order_id
    group by od.order_id
    ) as temp2
);"""
    Query_df = pd.DataFrame(columns = ['Order ID','Customer_ID','Order_item_count'])
    with conn.cursor() as cursor:
            cursor.execute(select_Query)
            records = cursor.fetchall()
            for row in records:
                output_df ={'Order ID': row[0], 'Customer_ID': row[1], 'Order_item_count': row[2]}
                Query_df = Query_df.append(output_df, ignore_index=True)
            print(Query_df)
```

Result of the above query



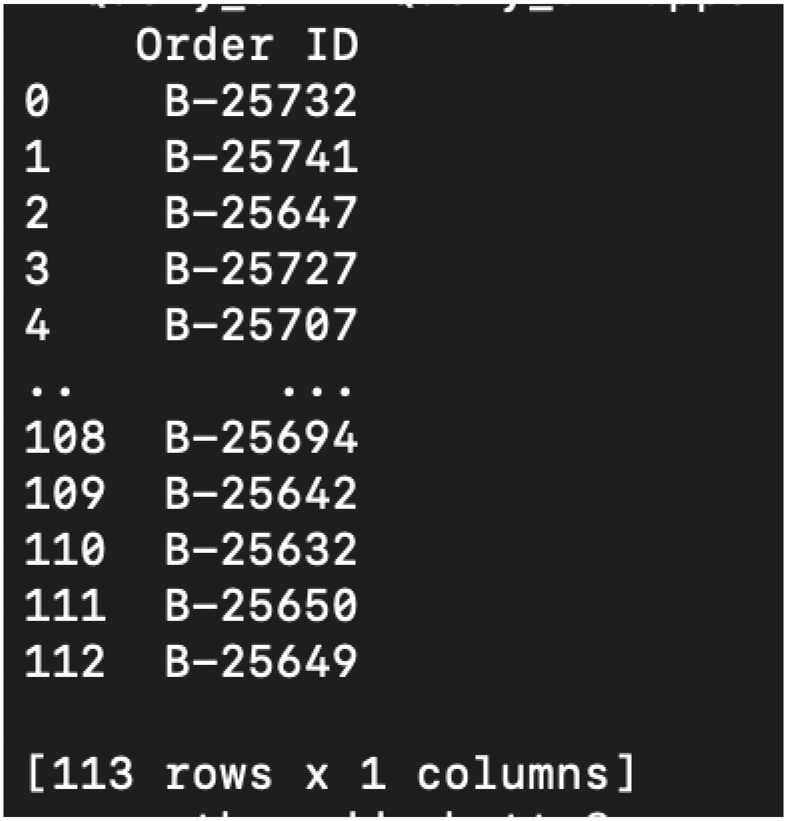
```
Last login: Thu Nov 17 11:35:42 on ttys000
[supreeth_mudduchetty@supreeths-MacBook-Air Python_files % python3 db_connect.py
/Users/supreeth_mudduchetty/Desktop/PSU/Fall_Courses/Into_to_DB/Project files/Python_files/db_connect.py:106: FutureWarning: The frame.append method is deprecated and will be re
 version. Use pandas.concat instead.
  Query_df = Query_df.append(output_df, ignore_index=True)
  Order ID Customer_ID Order_item_count
0  B-25701        CA99               26
```

Function that runs my query first query



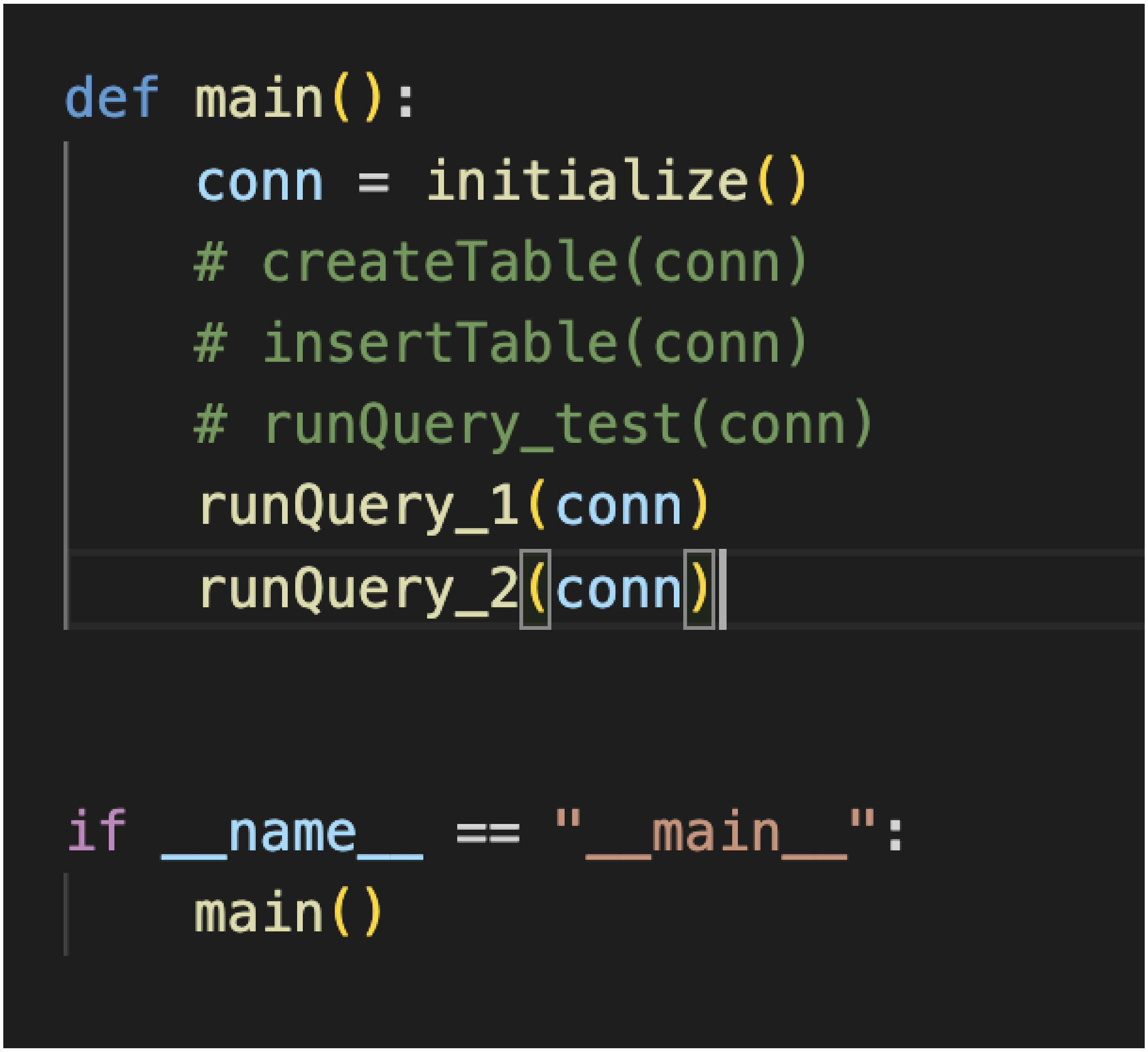
```
def runQuery_2(conn):
    select_Query = """select distinct od.order_id
    from order_details od
    where od.order_id not in (
    select distinct od.order_id
    from order_details od join products p
    on od.prod_id=p.prod_id
    where p.availability = 'NO'
    );"""
    Query_df = pd.DataFrame(columns = ['Order ID'])
    with conn.cursor() as cursor:
            cursor.execute(select_Query)
            records = cursor.fetchall()
            for row in records:
                output_df ={'Order ID': row[0]}
                Query_df = Query_df.append(output_df, ignore_index=True)
            print(Query_df)
```

Result of the above query



```
    Order ID
0     B-25732
1     B-25741
2     B-25647
3     B-25727
4     B-25707
..        ...
108   B-25694
109   B-25642
110   B-25632
111   B-25650
112   B-25649

[113 rows x 1 columns]
```

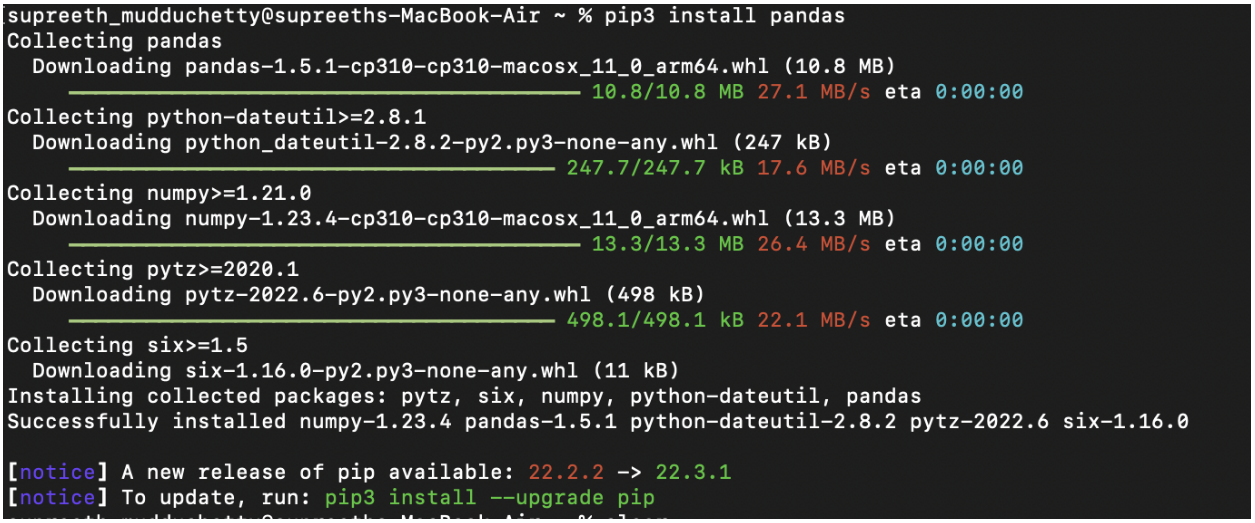
Screenshot of main function that calls my establish connection function and my execute query functions.



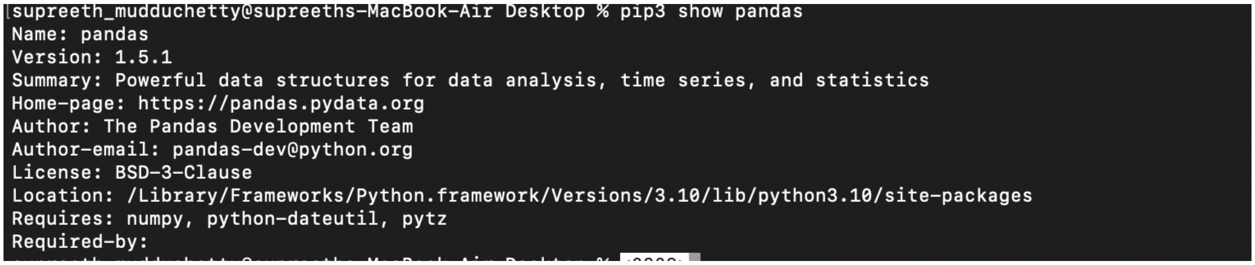
```python
def main():
    conn = initialize()
    # createTable(conn)
    # insertTable(conn)
    # runQuery_test(conn)
    runQuery_1(conn)
    runQuery_2(conn)


if __name__ == "__main__":
    main()
```

Screenshot of libraries imported

Pandas

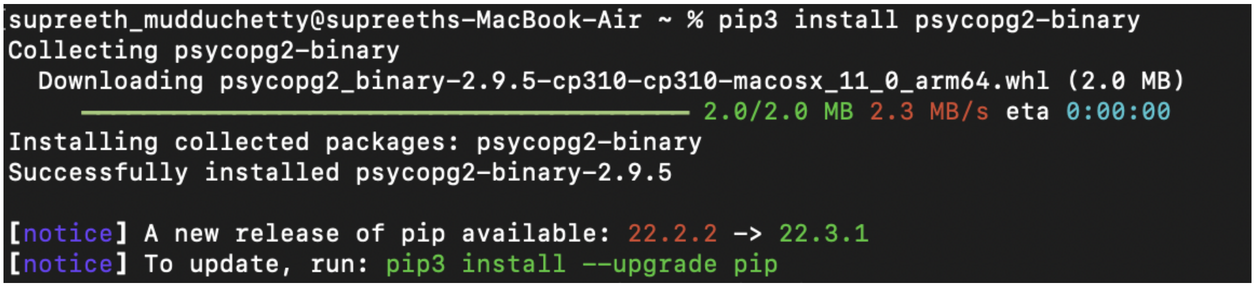


```
[supreeth_mudduchetty@supreeths-MacBook-Air ~ % pip3 install pandas
Collecting pandas
  Downloading pandas-1.5.1-cp310-cp310-macosx_11_0_arm64.whl (10.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.8/10.8 MB 27.1 MB/s eta 0:00:00
Collecting python-dateutil>=2.8.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.7/247.7 kB 17.6 MB/s eta 0:00:00
Collecting numpy>=1.21.0
  Downloading numpy-1.23.4-cp310-cp310-macosx_11_0_arm64.whl (13.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.3/13.3 MB 26.4 MB/s eta 0:00:00
Collecting pytz>=2020.1
  Downloading pytz-2022.6-py2.py3-none-any.whl (498 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 498.1/498.1 kB 22.1 MB/s eta 0:00:00
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, six, numpy, python-dateutil, pandas
Successfully installed numpy-1.23.4 pandas-1.5.1 python-dateutil-2.8.2 pytz-2022.6 six-1.16.0

[notice] A new release of pip available: 22.2.2 -> 22.3.1
[notice] To update, run: pip3 install --upgrade pip
```

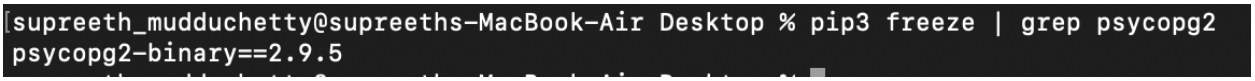


```
[supreeth_mudduchetty@supreeths-MacBook-Air Desktop % pip3 show pandas
Name: pandas
Version: 1.5.1
Summary: Powerful data structures for data analysis, time series, and statistics
Home-page: https://pandas.pydata.org
Author: The Pandas Development Team
Author-email: pandas-dev@python.org
License: BSD-3-Clause
Location: /Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages
Requires: numpy, python-dateutil, pytz
Required-by:
```

psycopg2-binary



```
[supreeth_mudduchetty@supreeths-MacBook-Air ~ % pip3 install psycopg2-binary
Collecting psycopg2-binary
  Downloading psycopg2_binary-2.9.5-cp310-cp310-macosx_11_0_arm64.whl (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 2.3 MB/s eta 0:00:00
Installing collected packages: psycopg2-binary
Successfully installed psycopg2-binary-2.9.5

[notice] A new release of pip available: 22.2.2 -> 22.3.1
[notice] To update, run: pip3 install --upgrade pip
```



```
[supreeth_mudduchetty@supreeths-MacBook-Air Desktop % pip3 freeze | grep psycopg2
psycopg2-binary==2.9.5
```