

```

import psycopg2

import test

try:

    connection = psycopg2.connect(user="postgres", password="36@gmail.", host="localhost",
port="5432", database="test")

    cursor = connection.cursor()


    # cursor.execute('SELECT * FROM parts')

    # result = cursor.fetchall()

    # print(result)

#   cursor.execute("""
#   CREATE TABLE Rating (UserID INTEGER NOT NULL, MovieID INTEGER NOT NULL,PRIMARY KEY
#   (UserID , MovieID), Rating FLOAT)
#   """)
# #
#   connection.commit()
#   print("Table create successfully")


filePath = 'ratings.dat'
fp = open(filePath)
for l in fp.readlines():
    value = l.strip().split("::")
    uid = (value[0])
    mid = (value[1])
    r = (value[3])
    # print(value[0],value[1],value[2])
    cursor.execute("INSERT INTO rating ( UserID, MovieID, Rating) VALUES (" +uid+", " + mid + ", " + r+")")
    connection.commit()
print("Data Insert successfully")

```

```
fp.close()
```

```
# with open(test.txt) as myfile:
```

```
#     myfile.read()
```

```
    cursor.close()
```

```
#
```

```
except:
```

```
    print("Database not connected")
```

The screenshot shows a PyCharm IDE window with a Python script named `1st.py` open. The script is located at `C:\Users\TS\Downloads\Compressed\18191203020\1st.py`. The script imports `psycopg2` and `test`, and attempts to connect to a PostgreSQL database. It then reads data from a file named `ratings.dat` and inserts it into a table named `rating` in a database named `test`. The script uses a `try` block to handle the database connection and data insertion. The output of the script is `Data Insert successfully`.

```
1 import psycopg2
2 import test
3 try:
4     connection = psycopg2.connect(user="postgres", password="36@gmail.", host="localhost", port="5432",
5                                     database="test")
6     cursor = connection.cursor()
7     ...
16
17     filePath = 'ratings.dat'
18     fp = open(filePath)
19     for l in fp.readlines():
20         value = l.strip().split("::")
21         uid = (value[0])
22         mid = (value[1])
23         r = (value[3])
24         # print(value[0],value[1],value[2])
25         cursor.execute("INSERT INTO rating ( UserID, MovieID, Rating) VALUES ('"+uid+"'" + mid + "'" + r+"')")
26         connection.commit()
27     print("Data Insert successfully")
28     fp.close()
29
30     ...
32     cursor.close()
33 #
34 except:
```

The bottom of the screenshot shows the Run console with the command `E:\python\project1\venv\Scripts\python.exe C:/Users/TS/Downloads/Compressed/18191203020/1st.py` and the output `Data Insert successfully`.

pgAdmin 4

File Object Tools Help

Browser

- Foreign Data Wrappers
- Languages
- Publications
- Schemas (2)
 - public
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Procedures
 - 1.3 Sequences
 - Tables (5)
 - part_drawings
 - parts
 - rating
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - vendor_parts
 - vendors
 - Trigger Functions

Dashboard Properties SQL Statistics Dependencies Dependents public.rating/te... public.rating/te... test/postgres(< > x

test/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

```
1 SELECT * FROM public.rating
2 ORDER BY userid DESC, movieid DESC
```

Data Output Explain Messages Notifications

	userid [PK] integer	movieid [PK] integer	rating double precision
1	3658	2021	913797071
2	3658	2017	913797872
3	3658	2014	913798055
4	3658	2012	913796265
5	3658	2011	913798211
6	3658	2009	913797920
7	3658	2003	913798245
8	3658	2001	913797431
9	3658	2000	913797474
10	3658	1968	913797777

ENG
12:04 AM
Friday
1/14/2022