1. **CAST OBJECTS TO A DATA TYPE**

SELECT customerNumber,

    COUNT(\*) AS number\_payments,

    MIN(CAST(amount AS INT)) AS min\_purchase,

    MAX(CAST(amount AS INT))  AS max\_purchase,

    AVG(CAST(amount AS INT)) AS avg\_purchase,

    SUM(CAST(amount AS INT)) AS total\_spent

FROM payments

pd.read\_sql('''

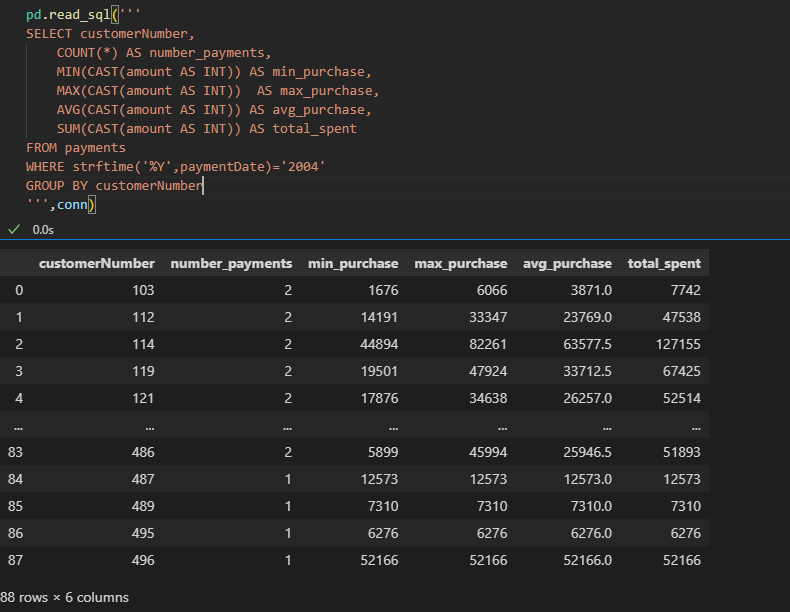
select cast(round(priceEach) as INTEGER) as rounded\_price\_int

        from orderDetails

            ''',conn)

**2.Strip year or month from date as a string object**

WHERE strftime('%Y',paymentDate)='2004'



pd.read\_sql('''

select orderDate,

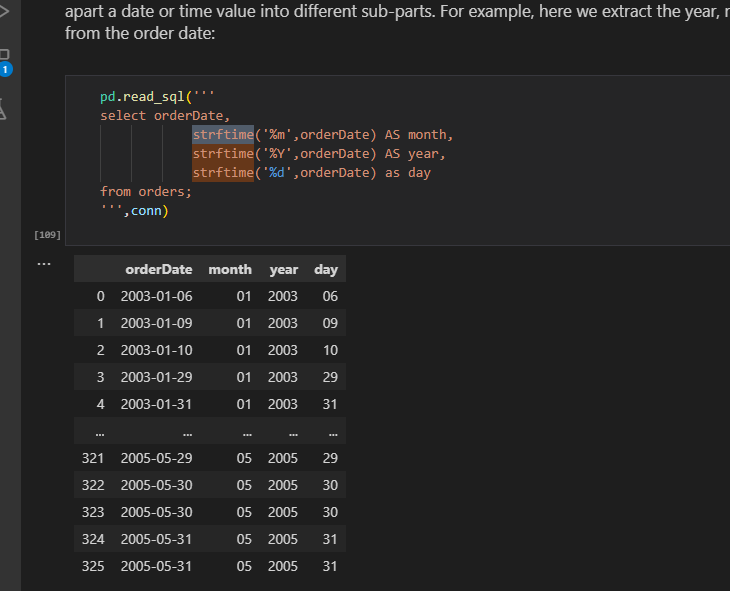
            strftime('%m',orderDate) AS month,

            strftime('%Y',orderDate) AS year,

            strftime('%d',orderDate) as day

from orders;

''',conn)



**Or use substr method**

pd.read\_sql('''

SELECT customerNumber,

    COUNT(\*) AS number\_payments,

    MIN(CAST(amount AS INT)) AS min\_purchase,

    MAX(CAST(amount AS INT))  AS max\_purchase,

    AVG(CAST(amount AS INT)) AS avg\_purchase,

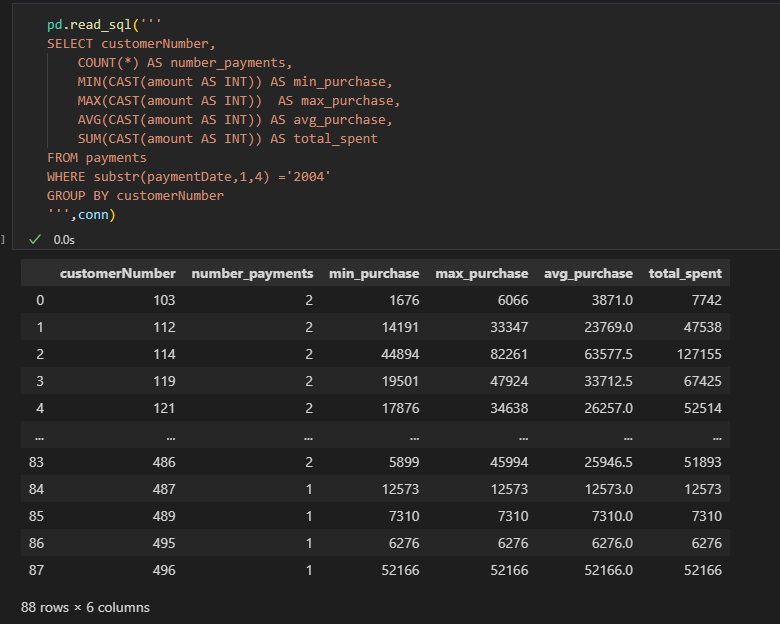
    SUM(CAST(amount AS INT)) AS total\_spent

FROM payments

WHERE substr(paymentDate,1,4) ='2004'

GROUP BY customerNumber

''',conn)



**3.Convert select statement to dataframe**

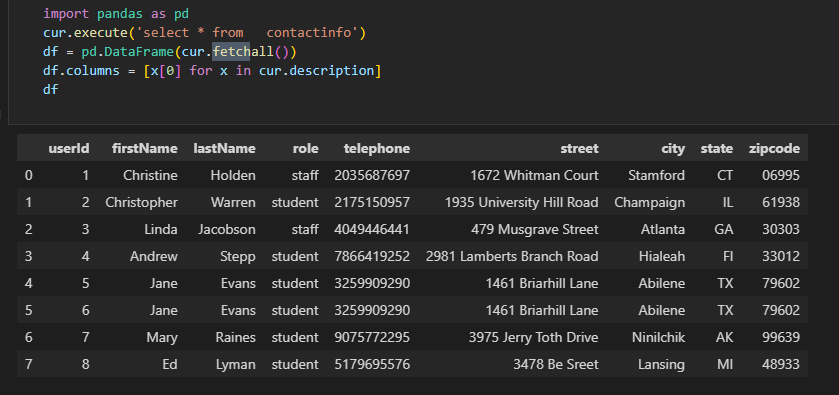
import pandas as pd

cur.execute('select \* from   contactinfo')

df = pd.DataFrame(cur.fetchall())

df.columns = [x[0] for x in cur.description]

df



* 4. Highest -**altitude**
* Southern/northern – **latitude**

**5.Pandasql Error**

**----> 6** passenger\_names **=** pysqldf**(**q**)**

**ImportError**: Unable to find a usable engine; tried using: 'sqlalchemy'.

A suitable version of sqlalchemy is required for sql I/O support.

Trying to import the above resulted in these errors:

**- Pandas requires version '1.4.0' or newer of 'sqlalchemy' (version '1.3.19' currently installed).**

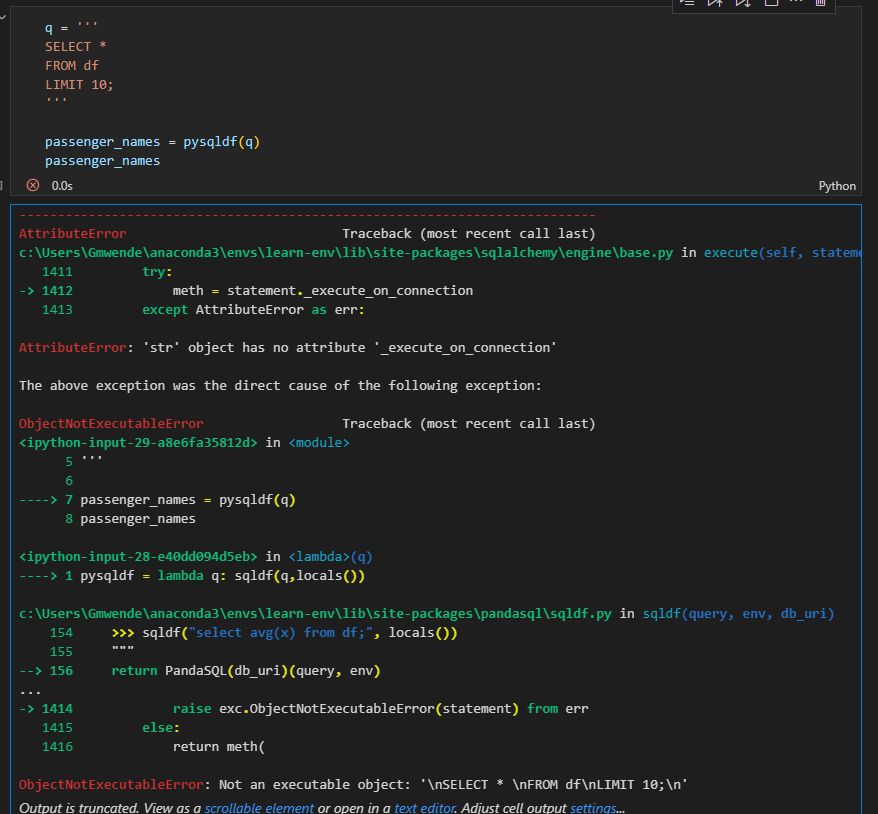
**TO update use**

conda update sqlalchemy

Check version if updated

pip show sqlalchemy

6.



Works well in colab

**7.Put dataframe in memory as to use conn**

#put df to memory

import sqlite3

conn = sqlite3.connect(':memory:')

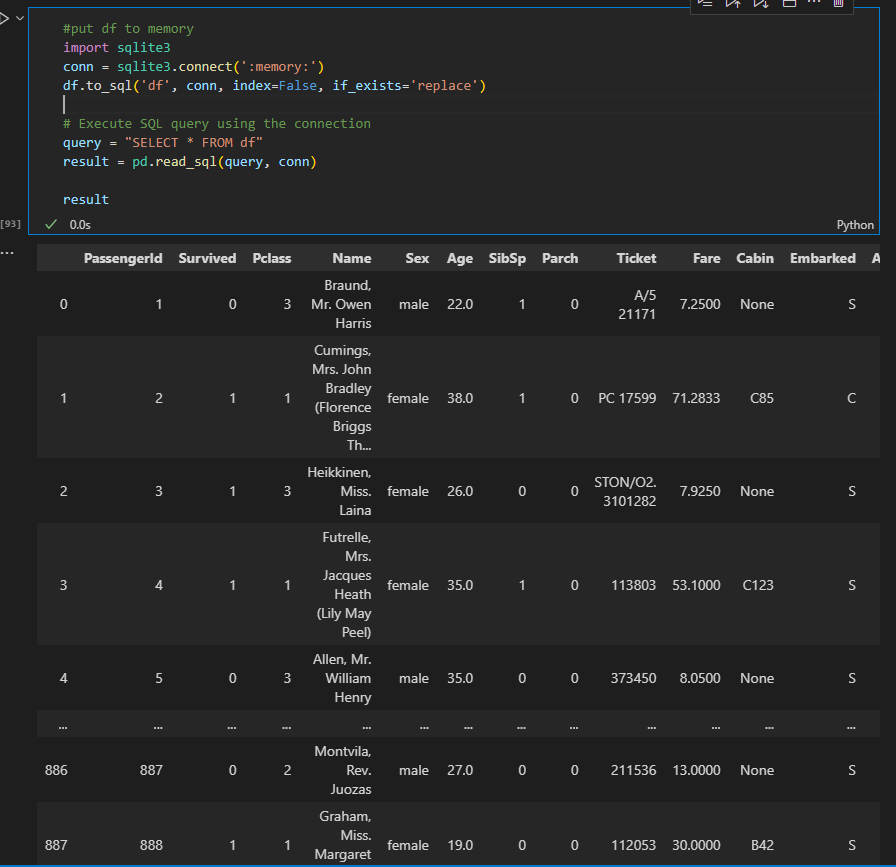
df.to\_sql('df', conn, index=False, if\_exists='replace')

# Execute SQL query using the connection

query = "SELECT \* FROM df"

result = pd.read\_sql(query, conn)

result



**8.Get female and children that is female and male less than or equal to 15**

df[(df['Sex'] == 'female') | (df['Age'] <= 15)]