Not syncing | Home | Exchange_Rates | Stores | Sales | products | customers

localhost:8888/notebooks/jupyter_code/Exchange_Rates.ipynb

jupyter Exchange_Rates Last Checkpoint: 2 days ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11215 entries, 0 to 11214
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      11215 non-null  datetime64[ns]
 1   Currency  11215 non-null  object
 2   Exchange  11215 non-null  float64
dtypes: datetime64[ns](1), float64(1), object(1)
memory usage: 263.0+ KB
```

[11]:
```
df=pd.DataFrame(df)
df.to_csv("Exchange_Rates.csv")
```

[4]:
```
import pymysql
import pandas as pd
myconnection = pymysql.connect(host ='127.0.0.1',user = 'root',passwd = '6381167213',database = 'dataspark')
```

[5]:
```
df = df.astype(object).where(df.notnull(), None)
```

[7]:
```
#exchange_rates
a=",".join(f"{i} {j}"
for i,j in zip(df.columns,df.dtypes)).replace("float64","float").replace("category","text").replace("int64","int").replace("object","text").replace("date
f"create table exchange_rates ({a})"
myconnection.cursor().execute(f"create table exchange_rates ({a})")
sql = "insert into dataspark.exchange_rates values"
for i in range(len(df)):
    myconnection.cursor().execute(f"{sql} {tuple(df.iloc[i])}")
    myconnection.commit()
```

[ ]:

84°F Partly cloudy

Search

ENG IN

23:40 15-08-2024

Not syncing | Home | Exchange_Rates | Stores | Sales | products | customers

localhost:8888/notebooks/jupyter_code/Stores.ipynb

jupyter Stores Last Checkpoint: 2 minutes ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

4	5	Australia	Victoria	2000.0	2015-12-09
...	...	...	...	...	...
62	63	United States	Utah	2000.0	2008-03-06
63	64	United States	Washington DC	1330.0	2010-01-01
64	65	United States	West Virginia	1785.0	2012-01-01
65	66	United States	Wyoming	840.0	2014-01-01
66	0	Online	Online	NaN	2010-01-01

67 rows × 5 columns

[43]:
```
df=pd.DataFrame(df)
df.to_csv("Stores.csv")
```

[44]:
```
import pymysql
import pandas as pd
myconnection = pymysql.connect(host ='127.0.0.1',user = 'root',passwd = '6381167213',database = 'dataspark')
```

[45]:
```
df = df.astype(object).where(df.notnull(), None)
```

[46]:
```
#stores
a=",".join(f"{i} {j}"
for i,j in zip(df.columns,df.dtypes)).replace("float64","float").replace("category","text").replace("int64","int").replace("object","text").replace("date
f"create table stores ({a})"
myconnection.cursor().execute(f"create table stores ({a})")
sql = "insert into dataspark.stores values"
for i in range(len(df)):
    myconnection.cursor().execute(f"{sql} {tuple(df.iloc[i])}")
    myconnection.commit()
```

84°F Partly cloudy

Search

ENG IN

23:40 15-08-2024

Not syncing | Home | Exchange_Rates | Stores | Sales | products | customers

localhost:8888/notebooks/jupyter_code/Sales.ipynb

jupyter Sales Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[15]:
```
order_number     0.0
line_item        0.0
order_date       0.0
CustomerKey      0.0
StoreKey         0.0
ProductKey       0.0
Quantity         0.0
currency_code    0.0
dtype: float64
```

[16]:
```
df = pd.DataFrame(df)
df.to_csv("Sales.csv")
```

[18]:
```
import pymysql
import pandas as pd
myconnection = pymysql.connect(host ='127.0.0.1',user = 'root',passwd = '6381167213',database = 'dataspark')
```

[25]:
```
df = df.astype(object).where(df.notnull(), None)
```

[28]:
```
#sales
a=",".join(f"{i} {j}"
for i,j in zip(df.columns,df.dtypes)).replace("float64","float").replace("category","text").replace("int64","int").replace("object","text").replace("date
f"create table sales ({a})"
myconnection.cursor().execute(f"create table sales ({a})")
sql = "insert into dataspark.sales values"
for i in range(len(df)):
    myconnection.cursor().execute(f"{sql} {tuple(df.iloc[i])}")
    myconnection.commit()
```

[ ]:

84°F Partly cloudy

Search

ENG IN

23:40 15-08-2024

Not syncing | Home | Exchange_Rates | Stores | Sales | products | customers

localhost:8888/notebooks/jupyter_code/products.ipynb

jupyter products Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

```
 9   Category         2517 non-null   object
dtypes: float64(2), int64(3), object(5)
memory usage: 196.8+ KB
```

[14]:
```
df=pd.DataFrame(df)
df.to_csv("products.csv")
```

[15]:
```
import pymysql
import pandas as pd
myconnection = pymysql.connect(host ='127.0.0.1',user = 'root',passwd = '6381167213',database = 'dataspark')
```

[16]:
```
#products
a=",".join(f"{i} {j}"
for i,j in zip(df.columns,df.dtypes)).replace("float64","float").replace("category","text").replace("int64","int").replace("object","text").replace("date
f"create table products ({a})"
myconnection.cursor().execute(f"create table products ({a})")
```

[16]:
```
0
```

•[22]:
```
# Convert all columns in the DataFrame to standard Python types
df = df.astype(object).where(df.notnull(), None)

# SQL insert statement
sql = "insert into dataspark.products values"

# Inserting each row into the database
for i in range(len(df)):
    # Convert each row to a tuple of Python native types
    row = tuple(df.iloc[i].apply(lambda x: int(x) if isinstance(x, np.integer) else float(x) if isinstance(x, np.floating) else x))
    myconnection.cursor().execute(f"{sql} {row}")
    myconnection.commit()
```

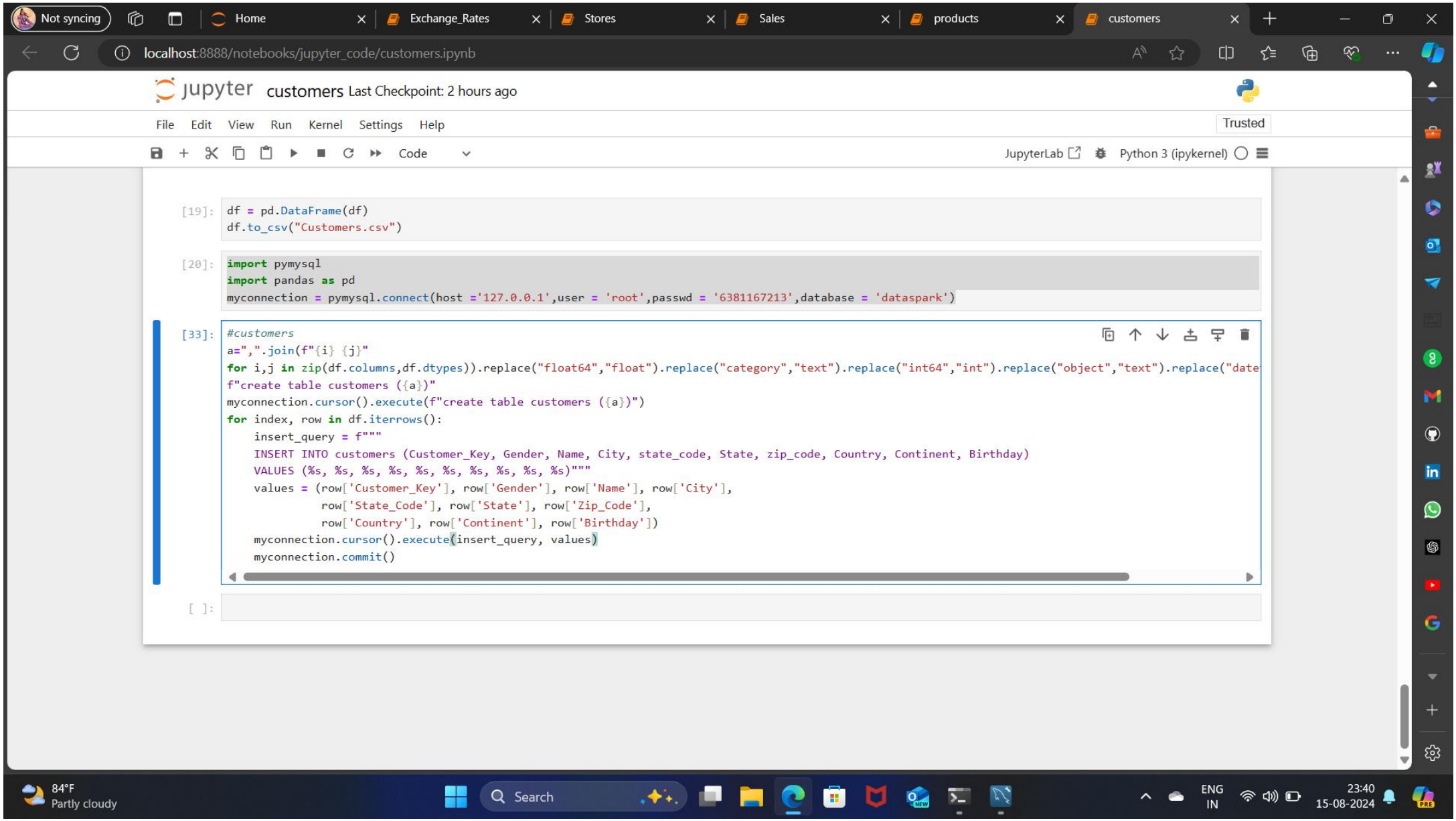
[ ]:

84°F Partly cloudy

Search

ENG IN

23:40 15-08-2024


Not syncing
Home
Exchange_Rates
Stores
Sales
products
customers
localhost:8888/notebooks/jupyter_code/customers.ipynb
jupyter customers Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[19]:
df = pd.DataFrame(df)
df.to_csv("Customers.csv")
[20]:
import pymysql
import pandas as pd
myconnection = pymysql.connect(host ='127.0.0.1',user = 'root',passwd = '6381167213',database = 'dataspark')
[33]:
#customers
a=",".join(f"{i} {j}"
for i,j in zip(df.columns,df.dtypes)).replace("float64","float").replace("category","text").replace("int64","int").replace("object","text").replace("date
f"create table customers ({a})"
myconnection.cursor().execute(f"create table customers ({a})")
for index, row in df.iterrows():
insert_query = f"""
INSERT INTO customers (Customer_Key, Gender, Name, City, state_code, State, zip_code, Country, Continent, Birthday)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"""
values = (row['Customer_Key'], row['Gender'], row['Name'], row['City'],
row['State_Code'], row['State'], row['Zip_Code'],
row['Country'], row['Continent'], row['Birthday'])
myconnection.cursor().execute(insert_query, values)
myconnection.commit()
[ ]:
84°F
Partly cloudy
Search
ENG
IN
23:40
15-08-2024