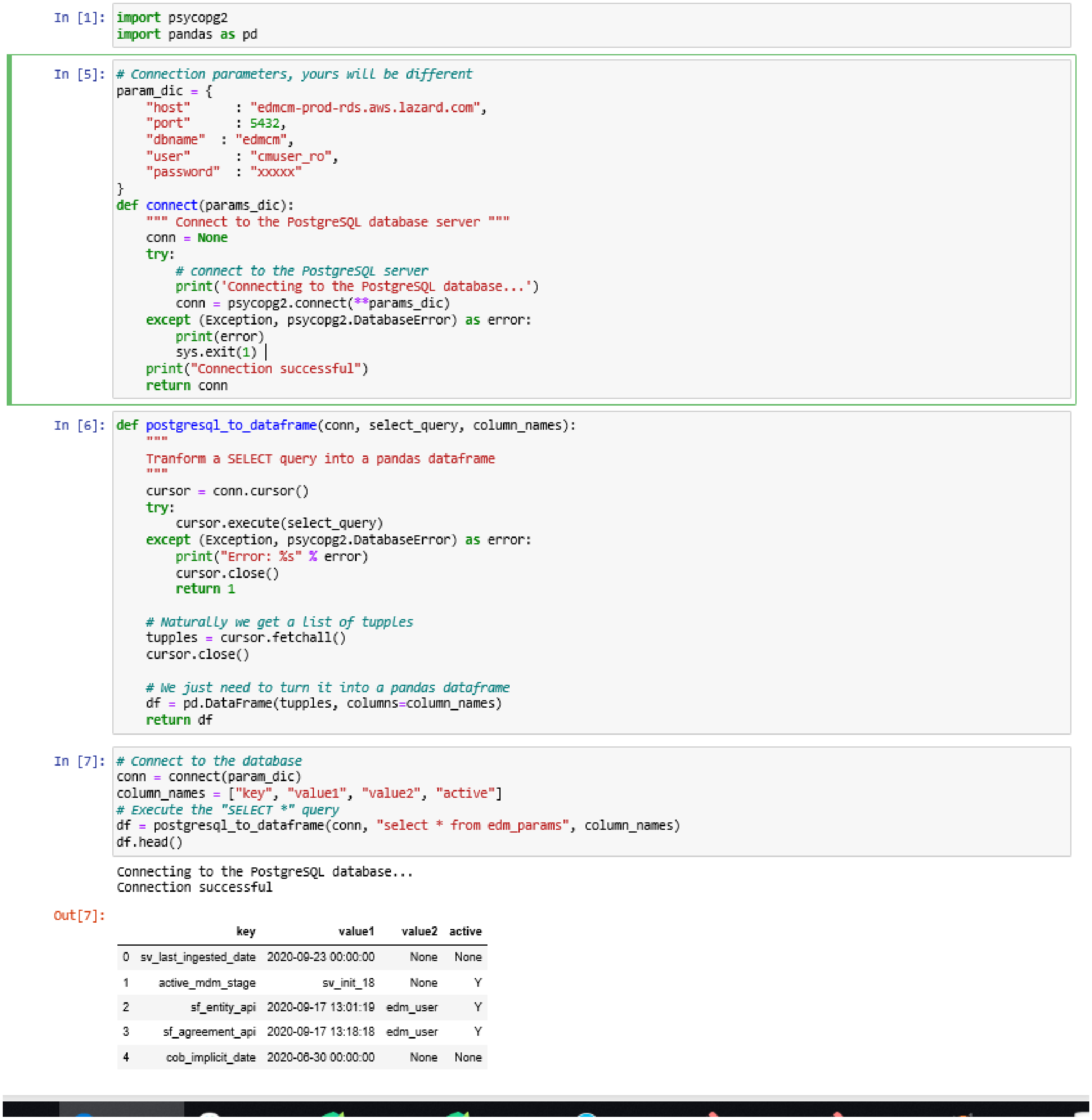
How to connect Jupyter Notbook to PostGregSql, and convert sql result into dataframe

Connect Jupyter Notbook to PostGregSql, and convert sql result into dataframe



In [1]:

```
import psycopg2
import pandas as pd
```

In [5]:

```
# Connection parameters, yours will be different
param_dic = {
    "host"      : "edmcm-prod-rds.aws.lazard.com",
    "port"      : 5432,
    "dbname"  : "edmcm",
    "user"      : "cmuser_ro",
    "password"  : "xxxxx"
}
def connect(params_dic):
    """ Connect to the PostgreSQL database server """
    conn = None
    try:
        # connect to the PostgreSQL server
        print('Connecting to the PostgreSQL database...')
        conn = psycopg2.connect(**params_dic)
    except (Exception, psycopg2.DatabaseError) as error:
        print(error)
        sys.exit(1) 
    print("Connection successful")
    return conn
```

In [6]:

```
def postgresql_to_dataframe(conn, select_query, column_names):
    """
    Tranform a SELECT query into a pandas dataframe
    """
    cursor = conn.cursor()
    try:
        cursor.execute(select_query)
    except (Exception, psycopg2.DatabaseError) as error:
        print("Error: %s" % error)
        cursor.close()
        return 1

    # Naturally we get a list of tupples
    tupples = cursor.fetchall()
    cursor.close()

    # We just need to turn it into a pandas dataframe
    df = pd.DataFrame(tupples, columns=column_names)
    return df
```

In [7]:

```
# Connect to the database
conn = connect(param_dic)
column_names = ["key", "value1", "value2", "active"]
# Execute the "SELECT *" query
df = postgresql_to_dataframe(conn, "select * from edm_params", column_names)
df.head()
```

```
Connecting to the PostgreSQL database...
Connection successful
```

Out[7]:

	key	value1	value2	active
0	sv_last_ingested_date	2020-09-23 00:00:00	None	None
1	active_mdm_stage	sv_init_18	None	Y
2	sf_entity_api	2020-09-17 13:01:19	edm_user	Y
3	sf_agreement_api	2020-09-17 13:18:18	edm_user	Y
4	cob_implicit_date	2020-06-30 00:00:00	None	None