



In [6]:

```
conda install cx_oracle
```

Collecting package metadata (current\_repodata.json): ...working... done  
Solving environment: ...working... done

## Package Plan ##

environment location: C:\Users\AnilKumar\anaconda3  
  
added / updated specs:  
- cx\_oracle

The following packages will be downloaded:

package	build	
-----	-----	
conda-4.11.0	py39haa95532_0	14.4 MB
cx_oracle-8.3.0	py39h8cc25b3_0	192 KB
-----	-----	
Total:		14.6 MB

The following NEW packages will be INSTALLED:

cx\_oracle pkgs/main/win-64::cx\_oracle-8.3.0-py39h8cc25b3\_0

The following packages will be UPDATED:

conda 4.10.3-py39haa95532\_0 --> 4.11.0-py39h  
aa95532\_0

Downloading and Extracting Packages

Note: you may need to restart the kernel to use updated packages.

conda-4.11.0	14.4 MB		0%
conda-4.11.0	14.4 MB		0%
conda-4.11.0	14.4 MB		1%
conda-4.11.0	14.4 MB	2	2%
conda-4.11.0	14.4 MB	4	5%
conda-4.11.0	14.4 MB	8	9%
conda-4.11.0	14.4 MB	#2	12%
conda-4.11.0	14.4 MB	#6	16%
conda-4.11.0	14.4 MB	##	21%
conda-4.11.0	14.4 MB	##5	26%
conda-4.11.0	14.4 MB	##9	30%
conda-4.11.0	14.4 MB	###4	34%
conda-4.11.0	14.4 MB	###8	39%
conda-4.11.0	14.4 MB	####3	43%
conda-4.11.0	14.4 MB	####8	48%
conda-4.11.0	14.4 MB	#####2	53%
conda-4.11.0	14.4 MB	#####7	57%
conda-4.11.0	14.4 MB	#####1	62%
conda-4.11.0	14.4 MB	#####5	66%
conda-4.11.0	14.4 MB	#####9	70%
conda-4.11.0	14.4 MB	#####3	74%

conda-4.11.0	14.4 MB	#####7	78%
conda-4.11.0	14.4 MB	#####1	81%
conda-4.11.0	14.4 MB	#####4	84%
conda-4.11.0	14.4 MB	#####7	87%
conda-4.11.0	14.4 MB	#####	90%
conda-4.11.0	14.4 MB	#####3	94%
conda-4.11.0	14.4 MB	#####7	98%
conda-4.11.0	14.4 MB	#####	100%

cx_oracle-8.3.0	192 KB		0%
cx_oracle-8.3.0	192 KB	8	8%
cx_oracle-8.3.0	192 KB	#####	100%
cx_oracle-8.3.0	192 KB	#####	100%

Preparing transaction: ...working... done

Verifying transaction: ...working... done

Executing transaction: ...working... done

In [8]:

```
conda install cx_oracle
```

Collecting package metadata (current\_repodata.json): ...working... done

Solving environment: ...working... done

# All requested packages already installed.

Note: you may need to restart the kernel to use updated packages.

In [10]:

```
import cx_oracle
conn=cx_oracle.connection('anildb@//localhost:1521/xe')
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp\ipykernel_144\307189935.py in <module>
----> 1 import cx_oracle
      2 conn=cx_oracle.connection('anildb@//localhost:1521/xe')
```

**ModuleNotFoundError:** No module named 'cx\_oracle'

In [48]:

```
import cx_Oracle
def getconnection():
    connection=cx_Oracle.connect('anildb/root2/localhost:1521/xe')
    return connection
def fetchdata():
    connection = getconnection()
    cursor = connection.cursor()
    sql_fetch_date = "select * from kpi_stg_channel"
    cursor.execute(sql_fetch_date)
    for result in cursor :
        print(result)
    connection.commit()
    cursor.close()
```

In [40]:

```
#import module
import cx_Oracle
#create a table in oracle data base
try:
    con =cx_Oracle.connect('anildb/root@//localhost:1521/xe')
    print(con.version)
# now execute sql query
    cursor = con.cursor()
# creating a table employee
    cursor.execute("create table employee(empid integer primary key,name varchar2(20),salar
    print("table created successfully")
except cx_Oracle.DatabaseError as e:
    print("there is a problem with Oracle", e)
# by writing finally if any error occurs
# then also we can close the all database operation
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()
```

11.2.0.2.0

table created successfully

In [59]:

```

import cx_Oracle
try:
    con = cx_Oracle.connect('anildb/root@//localhost:1521/xe')
    cursor = con.cursor()
    cursor.execute('insert into employee values(10001,\'Rahul\',50000.50)')
    con.commit()
    print('Record inserted successfully')
except cx_Oracle.DatabaseError as e:
    print("There is a problem with Oracle", e)
finally:
    if cursor:
        cursor.close()
    if con:
        con.close()

```

Record inserted successfully

In [65]:

```

import cx_Oracle
try:
    con = cx_Oracle.connect('anildb/root@//localhost:1521/xe')
except cx_Oracle.DatabaseError as e:
    print("there is an error with Oracle data base",e)
else:
    try:
        cur : con.cursor()
        data = [[10007,'vikram',48000],[10008,'Sunil',65000.1],[10009,'sameer',75000.0]]
        cur = con.cursor()
#Inserting multiple records into employee table
        cur.executemany('insert into employee values(:1,:2,:3)',data)
    except cx_Oracle.DatabaseError as e:
        print("there is an error with an Oracle",e)
    except Exception as e:
        print(e)
    else:
        con.commit()
        print('multiple records are inserted succesfully')
finally:
    if cur:
        cur.close()
if con:
    con.close()

```

multiple records are inserted succesfully

In [68]:

```

import cx_Oracle
try:
    con = cx_Oracle.connect('anildb/root@//localhost:1521/xe')
    print(con.version)
except cx_Oracle.DatabaseError as e:
    print("This is the problem with Oracle",e)
else:
    try:
        cur = con.cursor()
        #fetchall() is used to fetch all records from result set.
        cur.execute('select * from employee')
        rows = cur.fetchall()
        print(rows)
        #fetchmany(int) is used to fetch limited number of records from result set based on integer
        cur.execute('select * from employee')
        rows = cur.fetchmany(3)
        print(rows)
        #fetchone() is used to fetch one record from top of the result set
        cur.execute('select * from employee')
        rows = cur.fetchone()
        print(rows)
    except cx_Oracle.DatabaseError as e:
        print("there is an error with Oracle data base : ",e)
    except Exception as e:
        print('Error:'+str(e))
finally:
    if cur:
        cur.close()
    if con:
        con.close()

```

11.2.0.2.0

```

[(10001, 'Rahul', 50000.5), (10007, 'vikram', 48000.0), (10008, 'Sunil', 65000.1), (10009, 'sameer', 75000.0)]
[(10001, 'Rahul', 50000.5), (10007, 'vikram', 48000.0), (10008, 'Sunil', 65000.1)]
(10001, 'Rahul', 50000.5)

```

In [69]:

```
import cx_Oracle
try:
    con=cx_Oracle.connect('anildb/root@//localhost:1521/xe')
except cx_Oracle.DatabaseError as e:
    print('There is an error in Oracle database :',e)
else:
    try:
        cur = con.cursor()
        cur.execute('select * from employee where salary > :sal',{'sal' : 60000})
        rows = cur.fetchall()
        print(rows)
    except Exception as e :
        print('Error :',e)
finally:
    if cur:
        cur.close()
    if con:
        con.close()
```

```
[(10008, 'Sunil', 65000.1), (10009, 'sameer', 75000.0)]
```

In [ ]: