Oracle Basic Commands :

1.You must install Oracle Software [Oracle 11.2 and higher version]

2.By the time of installation we set User name and Password For U R Database

How to Open Oracle

1. open command prompt [window key + R --> Run --> cmd --> OK]
2. type SQLPlus and Press Enter

Enter Username : system | default user account in Oracle

Enter Password : manager | password For System user

3. SQL>

cl scr:

SQL>CL SCR

- To Clear the Screen

SQL>Show user

- It will display the current username

Create :

- It used to create table to store the records

Syn: SQL>Create <table> <tablename>

(<column name> <datatype>(size),.....,

<column n> <datatype>(size));

SQL> create table student

```
 2  (sno number(3),
 3  sname varchar(10),
 4  scity varchar(10) );
```

SQL> select * from tab;

- It will display all the objects existed in current user

```
TNAME                          TABTYPE  CLUSTERID
------------------------------ ------- ----------
BONUS                          TABLE
DEPT                           TABLE
EMP                            TABLE
SALGRADE                       TABLE
STUDENT                        TABLE
```

SQL>DESC[ribe] <tablename>

- It will display all column names and their data types and sizes

SQL> desc student

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------

 SNO                                       NUMBER(3)
 SNAME                                     VARCHAR2(20)
 SCITY                                     VARCHAR2(20)
```

INSERT :

- It is used to insert the data into table

- If you want insert the data into the specified column then we specify the the column names
- If you want insert the data into all the columns then doesn't required to specify the column names but values order should exactly same as order of the columns

Syn: SQL>Insert <into> <tablename>
(<column1>,<Column 2>,....., <column n>)
values
(<value1>,<value 2>,....., <value n>);

```
SQL> insert into student
  2  (sno,scity)
  3  values
  4  (101,'kmm');
1 row created.
```

Note : While passing the values to char| varchar | date data types then those values should be given in ' '

- missing data is represented as null value

```
SQL> insert into student
  2  values
  3  (123,'james','vizag');
1 row created.
```

SELECT :

- it used to read the data from the specified columns or all the columns

from the given table conditionally or unconditionaly.

Syn: SQL>SELECT <columnList> FROM <tablename>
[WHERE <condition>];

```
SQL> select sno from student;
     SNO
----------
     101
     123

SQL> select sno,scity from student;
     SNO SCITY
---------- --------------------
     101 kmm
     123 vizag

SQL> select * from student;
     SNO SNAME           SCITY
---------- -------------------- -----
     101                      kmm
     123 james           vizag
```

Predefine Tables are Existed in the SCOTT User Only:
emp | dept | bonus | salgrade

```
SQL> select * from dept;
   DEPTNO DNAME        LOC
```

```
---------- -------------- ---------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON


SQL> select * from emp;
EMPNO ENAME      JOB        MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ----------
--------------
 7369 SMITH      CLERK      7902 17-DEC-80   800          20
 7499 ALLEN      SALESMAN   7698 20-FEB-81  1600   300     30
 7521 WARD       SALESMAN   7698 22-FEB-81  1250   500     30
 7566 JONES      MANAGER    7839 02-APR-81  2975           20
 7654 MARTIN     SALESMAN   7698 28-SEP-81  1250  1400     30
 7698 BLAKE      MANAGER    7839 01-MAY-81  2850           30
 7782 CLARK      MANAGER    7839 09-JUN-81  2450           10
 7788 SCOTT      ANALYST    7566 19-APR-87  3000           20
 7839 KING       PRESIDENT       17-NOV-81  5000           10
 7844 TURNER     SALESMAN   7698 08-SEP-81  1500     0     30
 7876 ADAMS      CLERK      7788 23-MAY-87  1100           20



SQL> select empno,ename,job,sal
  2  from emp
  3  where JOB='MANAGER';

EMPNO ENAME      JOB        SAL
```

```
----- ---------- --------- -----
 7566 JONES      MANAGER    2975
 7698 BLAKE      MANAGER    2850
 7782 CLARK      MANAGER    2450
```

```
SQL> select empno,ename,job,sal
  2  from emp
  3  where sal>=2500;

EMPNO ENAME      JOB        SAL
----- ---------- --------- -----
 7566 JONES      MANAGER    2975
 7698 BLAKE      MANAGER    2850
 7788 SCOTT      ANALYST    3000
 7839 KING       PRESIDENT  5000
 7902 FORD       ANALYST    3000
```

Update :

- It is used to make the changes in the existed records

Syn: SQL>Update <tablename>
SET <column_name>=<value>,.....
[WHERE <condition>];

```
SQL> select * from student;
 SNO SNAME               SCITY
----- -------------------- -------
```

```
 101                hyd
 123 james          hyd

SQL> update student
  2  set scity='kadapa'
  3  where sno=101;
1 row updated.

SQL> select * from student;
 SNO SNAME              SCITY
----- -------------------- -------
 101                kadapa
 123 james          hyd


SQL> update emp
  2  set comm=3000
  3  where deptno=30;

6 rows updated.

SQL> select * from emp
  2  where deptno=30;

EMPNO ENAME      JOB        MGR HIREDATE    SAL  COMM DEPTNO
----- ---------- --------- ----- --------- ----- ----- ------
 7499 ALLEN      SALESMAN   7698 20-FEB-81  1600  3000     30
 7521 WARD       SALESMAN   7698 22-FEB-81  1250  3000     30
```

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 3000 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 3000 30
7844 TURNER SALESMAN 7698 08-SEP-81 1500 3000 30
7900 JAMES CLERK 7698 03-DEC-81 950 3000 30

Delete :

- It used to delete a record or group of records all the records

Syn: SQL>Delete <from> <tablename>
[WHERE <condition>];

```
SQL> delete from student
  2  where sname='james';
1 row deleted.

SQL> select * from student;
 SNO SNAME                SCITY
----- -------------------- -------
 101                      kadapa

SQL> delete from student;
1 row deleted.

SQL> delete from emp;
11 rows deleted.

SQL> select * from emp;
no rows selected
```

ROLLBACK:

- It used to cancel the last transaction

SQL>Rollback;

COMMIT

- It used to make the transaction to Save
- Once transaction is commited Rollback doesn't work on it.

SQL>Commit;

>>> >>> help('modules')

Please wait a moment while I gather a list of all available modules...

TableDemo	asyncio	hmac	scrolledlist
Test	asyncore	html	search
__future__	atexit	http	searchbase
__main__	audioop	hyperparser	searchengine
_abc	autocomplete	idle	secrets
_ast	autocomplete_w	idle_test	select
_asyncio	autoexpand	idlelib	selectors
_bisect	base64	imaplib	setuptools
_blake2	bdb	imghdr	shelve
_bootlocale	binascii	imp	shlex
_bz2	binhex	importlib	shutil
_codecs	bisect	inspect	sidebar
_codecs_cn	browser	io	signal

_codecs_hk builtins iomenu site
_codecs_iso2022 bz2 ipaddress smtpd
_codecs_jp cProfile itertools smtplib
_codecs_kr calendar json sndhdr
_codecs_tw calltip keyword socket
_collections calltip_w lib2to3 socketserver
_collections_abc cgi linecache sqlite3
_compat_pickle cgitb locale squeezer
_compression chunk logging sre_compile
_contextvars cmath lzma sre_constants
_csv cmd macosx sre_parse
_ctypes code mailbox ssl
_ctypes_test codecontext mailcap stackviewer
_datetime codecs main stat
_decimal codeop mainmenu statistics
_dummy_thread collections marshal statusbar
_elementtree colorizer math string

How to Install the required Modules :

Syn: pip install module_name

Eg: pip install cx_Oracle

- How to install cx_Oracle Module
- Open Scripts Folder of the python at command prompt

```
C:\Users\class\AppData\Local\Programs
            \Python\Python38\Scripts>pip install cx_Ora
cle
Collecting cx_Oracle
  Downloading https://files.pythonhosted.org/packages/b2/6b/b97c28274add4e3e06e4
fa9284c4c3b94fead66f63a9e0a7f1e30a8d028a/cx_Oracle-8.0.0-cp38-cp38-
win_amd64.whl
 (203kB)
    |||||||||||||||||||||||||||||||| 204kB 1.1MB/s
Installing collected packages: cx-Oracle
Successfully installed cx-Oracle-8.0.0
```

PDBC

Steps For PDBC

1.import cx_Oracle and required modules

2.Est .connection with Oracle Database by using connect()

```
from cx_Oracle

connect(username,password:Databaseinfo) -> connection
Database Into :  IPAddress : portnumber/serviceid
                    localhost: 1521/orcl
```

connect( ) will return connection class Object If the connection is established otherwise it will return None

```
SQL> SELECT * FROM GLOBAL_NAME;
GLOBAL_NAME
```

```
------------------------------
ORCL
<class 'cx_Oracle.Connection'>
  Eg:  con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
       if con!=None:
            print("connection is Est with Oracle DB ")
       else:
            print("connection is Gone....!!!! ")
```

Step-2: Create cursor Object to send the Queries to the Database

- To Create Cursor Object we have to " cursor() -> cursor object " from connection class

```
cur=con.cursor()
        print("type is : ",type(cur) )  # <class 'cx_Oracle.Cursor'>
```

Step-3: Execute Queries by using execute() from cursor Class

```
cur.execute(Queries..)
    [ Queries :  Create | Update | Delete | Select | Insert ...]
```

will reading the data from the database by using execute( )
then it will extract all the records and stores them into cursor object only

Step-4: to Read the records from the cursor Object then we have to use

fetchone( ) -> it read and return first row from the cursor in the form of tuple

fetchmany() --> it will read and return the specified no.of.rows from the cursor in the form list of tuples

eg: rows = cur.fetchmany(3)

fetchall( ) --> it will read and return all the records from the ursor in the from of list of tuples

Step-5: Process the Database Result in Python application

Step-6 : close the cursor and connection object by using

close( ) From cursor class

close( ) From Connection Class

Note: Will inserting or updating or deleting records from the database the u must use commit() from connection otherwise those changes are not effected on Database.

Example Prg For Est Connection with Oracle :\

```
import cx_Oracle
connection=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")

if connection!=None:
   print("connection is Est ")
else:
   print("Connection is Gone..!")
```

Example 2:

Program For Creating a table

```
import cx_Oracle

con=cx_Oracle.connect("scott","tiger","localhost:1521/ocl")

if con!=None:
    cur=con.cursor()
    cur.execute("create table student(sno number(3),sname varchar(10),scity varchar(10))")
    print("Table is Created ...!")
    cur.close()
else:
    print("Connection is Failed...")

con.close()
```

Example 3:

Prg For Inserting A Record into Table:

```
import cx_Oracle

con=None
cur=None

try:
    con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
except cx_Oracle.DatabaseError as e:
```

```
    print("Sorry Unable to continue....")
    print("Reason : \n ",e)
else:
    cur=con.cursor()
    cur.execute("INSERT INTO DEPT VALUES(90,'IIIT','KPHB')");
    con.commit()
    print("Rec is inserted ...!")
finally:
    if cur!=None:
        cur.close()

    if con!=None:
        con.close()
```

Example For Reading Data From Database Table :

fetchone()

```
import cx_Oracle
import time
con=None
cur=None
try:
    con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
except cx_Oracle.DatabaseError as e:
    print("Sorry Unable to continue....")
    print("Reason : ",e)
```

```
else:
    cur=con.cursor()
    cur.execute("select ename from emp")
    lst=cur.fetchone()
    print(lst[0])
finally:
    if cur!=None:
        cur.close()
    if con!=None:
        con.close()
```

Example Prg To Read specified No.of.Records From The Table : using fetchmany()

```
import cx_Oracle
import time

con=None
cur=None
try:
    con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
except cx_Oracle.DatabaseError as e:
    print("Sorry Unable to continue....")
    print("Reason : ",e)
else:
    cur=con.cursor()
    cur.execute("select ename from emp")
    lst=cur.fetchmany(5)
```

```
    for i in lst:
        print(i[0])
        time.sleep(.2)
finally:
    if cur!=None:
        cur.close()
    if con!=None:
        con.close()
```

Example For Reading All Records using Fetchall():

```
import cx_Oracle
import time

con=None
cur=None
try:
    con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
except cx_Oracle.DatabaseError as e:
    print("Sorry Unable to continue....")
    print("Reason : ",e)
else:
    cur=con.cursor()
    cur.execute("select ename from emp")
    lst=cur.fetchall()
    for i in lst:
        print(i[0])
        time.sleep(.2)
```

```
finally:
    if cur!=None:
        cur.close()
    if con!=None:
        con.close()
```

Example For Reading All Records without Using fetchone() | fetchmany() or fetchall()

```
import cx_Oracle
import time

con=None
cur=None
try:
    con=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
except cx_Oracle.DatabaseError as e:
    print("Sorry Unable to continue....")
    print("Reason : ",e)
else:
    cur=con.cursor()
    cur.execute("select * from dept ")

    for i in cur:
        time.sleep(.3)
        print("Dno is       : ",i[0])
        print("Dname is : ",i[1])
```

```
        print("Dloc is     :",i[2])
        print("="*20)

finally:
    if cur!=None:
        cur.close()
    if con!=None:
        con.close()
```

Example Prg For Updating Records :

```
import cx_Oracle
connection=None
cur=None

try:
    connection=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
    cur=connection.cursor()
    sal=int( input("enter extra salary : ") )
    con=int(input("enter condition For salary update "))

    query="update emp set sal=sal+%d where sal=%d"
    cur.execute(query %(sal,con))

    connection.commit()
    print("  Rec are updated ...")

except cx_Oracle.DatabaseError as ref:
```

```
    print("sorry unable to continue ...")
    print("Reason ? : ",ref)
finally:
    if cur!=None:
        cur.close()

    if connection!=None:
        connection.close()
```

Prog to Read Data From Database and Store them into CSV File:

```
import cx_Oracle
import time
import csv

connection=None
cur=None

try:
    connection=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
    cur=connection.cursor()
    cur.execute("SELECT * from dept")
    lt=cur.fetchall()

    with open("DataBaseInfo.csv","w",newline='') as f:
        writer=csv.writer(f)
        for row in lt:
            time.sleep(.3)
```

```
            writer.writerow([row[0],row[1],row[2]])

    time.sleep(1)
    print("Records arec copied ...!!!")

except cx_Oracle.DatabaseError as ref:
    print("sorry unable to continue ...")
    print("Reason is : ",ref)
finally:
    if cur!=None:
        cur.close()

    if connection!=None:
        connection.close()
```

Prg For Deleting a Record :

MyDB.py

```
#MyDB.py
import cx_Oracle

def delete_rec(dno):
    connection=None
    cur=None
    try:
        connection=cx_Oracle.connect("scott","tiger","localhost:1521/orcl")
        cur=connection.cursor()
        cur.execute("Delete from dept where deptno=%d" %dno)
```

```
        print(cur.rowcount," Recs are Deleted ...!!! ")
        connection.commit()

    except cx_Oracle.DatabaseError as ref:
        print("Sorry unable to continue....")
        print('Reason : ',ref)

    finally:
        if cur!=None:
            cur.close()
        if connection!=None:
            connection.close()
```

Test.py

```
import MyDB
dno=int( input("Enter deptno for delete ") )
MyDB.delete_rec(dno)
```