Connecting to oracle database using Python

* Needs cx\_Oracle module
* To connect to Oracle database,

import cx\_Oracle

con=cx\_Oracle.connect('pythonhol/welcome@127.0.0.1/orcl')

print con.version

con.close()

* connect() accepts username and password
* It also accepts IP address of the machine and the database name.
* Close() terminates the connection.
* Connection will be terminated automatically, at the end of the script.
* When we run the script, if the connection is successful, version number is printed. Otherwise, an exception is thrown.

Creating a simple query

import cx\_Oracle

con=cx\_Oracle.connect('pythonhol/welcome@127.0.0.1/orcl')

cur = con.cursor()

cur.execute('select \* from departments order by department\_id')

for result in cur:

print result

cur.close()

con.close()

* cursor() opens a cursor for the statements to use.
* execute() parses and executes the statements.
* The for loop fetches each row from the cursor and prints them.
* The results of the query are displayed as Python tuples, which are arrays that can’t be changed.

Fetching data

import cx\_Oracle

con=cx\_Oracle.connect('pythonhol/welcome@127.0.01/orcl')

cur = con.cursor()

cur.execute('select \* from departments order by department\_id')  **row = cur.fetchone()  
print row  
row = cur.fetchone()  
print row**

#for I in cur:

# row=i.fetchone()

# print row

cur.close()

con.close()

* fetchone() just returns a single row as a tuple.
* When it is called multiple times, consecutive rows are returned.

Fetching many rows

import cx\_Oracle

con=cx\_Oracle.connect('pythonhol/welcome@127.0.0.1/orcl')

cur = con.cursor()

cur.execute('select \* from departments order by department\_id')

**res = cur.fetchmany(numRows=3)  
print res**

cur.close()

con.close()

* fetchmany(n) returns multiple rows.
* fetchall() returns all the rows.
* Fetchone() returns a single row

import cx\_Oracle

con=cx\_Oracle.connect('pythonhol/welcome@127.0.0.1/orcl')

cur = con.cursor()

cur.execute('select \* from departments order by department\_id')  **res = cur.fetchall()  
print res**

cur.close()

con.close()

Another way of connecting

sqlplus [pythonhol/welcome@127.0.0.1/orcl](mailto:pythonhol/welcome@127.0.0.1/orcl)

We will get directly sql> prompt, where we can execute SQL commands directly.

To get back the python prompt, say ‘exit’.

* We can also use PL/SQL stored procedures.