How to Use Oracle at CIMS

Contents

1 General Instructions

Please note that in your homework *you will not use* any GUIs for working with Oracle.

If the steps below do not work for you please send email to Guang Yang (`mailto:gy552@nyu.edu`).

In your home directory at CIMS, a file called `oracle_pass.txt` has been placed. It contains your password for accessing Oracle.

If say, your user name is `abc123` and the string in `oracle_pass.txt` is `qA3c89` these are your credentials for logging into Oracle server: we will call them your Oracle credentials.

Do not communicate with the CIMS helpdesk concerning any issues dealing with Oracle or your credentials. Communicate only with the grader as listed above.

You need to know about two CIMS machines:

1. `access.cims.nyu.edu`
2. `odb1.cims.nyu.edu`

These two machines share your file system. So what you put in your directory on one, appears on the other.

More details about the two machines:

1. You have previously been issued a CIMS computer account; if you have not, you can obtain it by following the instructions at `http://cims.nyu.edu/webapps/content/systems/userservices/accounts/obtain`.

2. odb1.cims.nyu.edu can be accessed by ssh to it from anywhere within CIMS computer network. If you want to connect from outside of CIMS, you will need first to connect securely to access.cims.nyu.edu and then to odb1.cims.nyu.edu. More details below.

2 Set-up Steps

We will assume that you work from your own machine not on an NYU network. Therefore you will need to connect to the gateway machine access.cims.NYU.edu.

1. Secure Romote Access

   Assuming that you will connect from outside CIMS, you first need a secure shell client to access the gateway.

   - For windows, we recommend putty, which you can find it at http://cims.nyu.edu/systems/userservices/netaccess/putty.exe.

     Download it and run. In the configuration panel, enter access.cims.nyu.edu and choose connection type as SSH. Click open and then enter your CIMS account and password. After connecting to the gateway, you can use essentially the same commands as UNIX. If you are not familiar with UNIX commands, you can find useful and basic commands at following link http://cims.nyu.edu/webapps/content/systems/platforms/sun/basicunix, but possibly you will not need to know practically anything other than how to connect to the Oracle server odb1.cims.NYU.edu.

   - For Mac OS, you can open the Terminal application through Applications, then Utities, Terminal. From here you can use the ssh commands. You can execute ssh username@access.cims.nyu.edu to log in.

   - For Linux/Unix, similarly to Mac.

2. File Transfer Programs

   File transfer programs are used to transfer files securely between local and remote servers.

   - For windows, we recommend winscp, which you can find it at http://cims.nyu.edu/systems/userservices/netaccess/winscp406setupintl.exe. Download and install it. Enter the Host Name, User Name and Password, then click login. After connecting, you will have two views, the local and the remote file systems. Then you can transfer files between the two file systems.

   - For Mac OS, you can use scp in Terminal. The syntax for the scp command is scp [options] user@sourcehost:dir1/file1 user@destinationhost:dire2/file2

     For example, if you want to copy a file called demo.txt from your home directory to the home directory in your account on the access.cims.nyu.edu, enter scp ~/demo.txt username@access.cims.nyu.edu:~/

   - For Linux/Unix, similarly to Mac.

3. Copy the example file to the CIMS network.

Enclosed is a file `ExamplePCI.sql`, which deals with database defined in `PlantCustomerInvoice.vsd`. Move it to your home directory on `access.cims.nyu.edu`. (For completeness, this file is also listed in this document as Appendix A.)

4. Log into Oracle

   From `access.cims.NYU.edu` securely log into `odb1.cims.nyu.edu` using command

   ```
   ssh odb1.cims.nyu.edu
   ```

   using your standard CIMS credentials. Once on `odb1.cims.NYU.edu`, log into Oracle *using your Oracle credentials*

   ```
   sqlplus abc123/qA3c89
   ```

   as in our example.

5. Record and execute SQL script

   Execute

   ```
   spool Name.txt
   ```

   and use the file name you like. This will start recording. Then execute

   ```
   @ExamplePCI
   ```

   to execute the script `ExamplePCI.sql`.

6. Stop Recording and Quit

   Execute

   ```
   spool off
   ```

   and

   ```
   quit
   ```

   to stop recording and to log off from Oracle.

7. Save the spool file

   Copy file `Name.txt` from `access.cims.nyu.edu` to your personal machine. Recall that `access.cims.nyu.edu` and `odb1.cims.nyu.edu` share your file system.

A ExamplePCI.sql

```
--   This SQL DDL script was generated by Microsoft Visual Studio (Release Date: LOCAL BUILD).

--    Driver Used : Microsoft Visual Studio - Oracle Server Driver.
--    Time Created: February 17, 2008 10:41 AM.
--    Operation   : From Visio Generate Wizard.
--    Connected data source : No connection.
--    Connected server      : No connection.
--    Connected database    : Not applicable.
```

```

-- Some parts commented out by Zvi


-- Create PCI database.
-- connect internal ;
-- startup nomount pfile=   ;
-- spool create_db_PCI ;

-- create database PCI
--      NOARCHIVELOG;

-- create rollback segment PCI_r0 tablespace system ;

-- alter rollback segment  PCI_r0 online ;

-- Create Oracle exception file.
-- create table exceptions ( row_id rowid, owner varchar2(30), table_name varchar2(30), constraint v

-- End parts commented out by Zvi


drop table INVOICE;

drop table CUSTOMER;

drop table PLANT;

-- Create new table INVOICE.
-- INVOICE : Table of Invoice
--      I# : I# identifies Invoice
--      AMT : Amt is of Invoice
--      IDATE : Idate is of Invoice
--      C# : C# is of Invoice
create table INVOICE (
        I# INTEGER not null,
        AMT INTEGER null,
        IDATE DATE null,
        C# INTEGER not null, constraint INVOICE_PK primary key (I#) );

-- Create new table CUSTOMER.
-- CUSTOMER : Table of Customer
--      C# : C# identifies Customer
--      CNAME : Cname is of Customer
--      CITY : City is of Customer
--      P# : P# is of Customer
```

```
create table CUSTOMER (
        C# INTEGER not null,
        CNAME VARCHAR2(10) null,
        CITY VARCHAR2(10) null,
        P# INTEGER null, constraint CUSTOMER_PK primary key (C#) );

-- Create new table PLANT.
-- PLANT : Table of Plant
--      P# : P# identifies Plant
--      PNAME : Pname is of Plant
--      CITY : City is of Plant
--      MARGIN : Margin is of Plant
create table PLANT (
        P# INTEGER not null,
        PNAME VARCHAR2(10) null,
        CITY VARCHAR2(10) null,
        MARGIN FLOAT null, constraint PLANT_PK primary key (P#) );


-- Populate the database (added by Zvi)



Insert into Plant values(1,'alpha','Boston',null);
Insert into Plant values(2,'alpha','Boston',0.2);
Insert into Plant values(3,'beta','New York',0.5);

Insert into Customer(C#,Cname,P#) values(11,'a',1);
Insert into Customer values(12,null,null,null);

Insert into Invoice values (21,50,'12-JAN-2009',11);


-- Back to Visual Studio script

-- Add foreign key constraints to table INVOICE.
alter table INVOICE
        add constraint CUSTOMER_INVOICE_FK1 foreign key (
                C#)
         references CUSTOMER (
                C#);

-- Add foreign key constraints to table CUSTOMER.
alter table CUSTOMER
        add constraint PLANT_CUSTOMER_FK1 foreign key (
                P#)
         references PLANT (
                P#);


-- This is the end of the Microsoft Visual Studio generated SQL DDL script.


-- Print description of the database


desc user_tables;

desc table_name from user_tables;

select table_name, constraint_name, constraint_type from user_constraints;

desc Plant;

desc Customer;

desc Invoice;

-- Print the database

select * from Plant;

select * from Customer;

select * from Invoice;

-- Run a trivial query

drop TABLE Answer0;

create Table Answer0 AS
select I#, Plant.City
from Plant, Customer, Invoice
where Plant.P# = Customer.P# and Customer.C# = Invoice.C#;

select * from Answer0;
```