Getting Started With Oracle

Ashikur Rahman
CSE, BUET

July 14, 2016

1 Logging In to Oracle

You should be logged onto one of the Windows 7 machine in the database lab. Open the command promt first by typping ’command’ in the Windows ’Run’ box.

Now, you can log in to Oracle by typing:

```
sqlplus <yourName>
```

Here, sqlplus is Oracle’s generic SQL interface. `<yourName>` refers to your lab login.

You will be prompted for your password. This password is initially changemesoon and *must be* changed as soon as possible. For security reasons, we suggest that you not use your regular BIIS password, because as we shall see there are opportunities for this password to become visible under certain circumstances. After you enter the correct password, you should receive the prompt

```
SQL>
```

2 Changing Your Password

In response to the `SQL>` prompt, type

```
ALTER USER <yourName> IDENTIFIED BY <newPassword>;
```

where `<yourName>` is again your login, and `<newPassword>` is the password you would like to use in the future. This command, like all other SQL commands, should be terminated with a semicolon.

Note that SQL is completely case-insensitive. Once you are in sqlplus, you can use capitals or not in keywords like ALTER; Even your password is case insensitive. We tend to capitalize keywords and not other things.

3 Creating a Table

In sqlplus we can execute any SQL command. One simple type of command creates a table (relation). The form is

```
CREATE  TABLE <tableName> (
       <list of attributes and their types>
       );
```

You may enter text on one line or on several lines. If your command runs over several lines, you will be prompted with line numbers until you type the semicolon that ends any command. An example table-creation command is:

```
CREATE TABLE test (
       i int,
       s char(10)
);
```

This command creates a table named test with two attributes. The first, named i, is an integer, and the second, named s, is a character string of length (up to) 10.

4 Getting Information About Your Database

The system keeps information about your own database in certain system tables. The most important for now is `USER_TABLES`. You can recall the names of your tables by issuing the query:

```
   SELECT TABLE_NAME FROM USER_TABLES;
```

More information about your tables is available from `USER_TABLES`. To see all the attributes of `USER_TABLES`, try:

```
   SELECT *
   FROM USER_TABLES;
```

It is also possible to recall the attributes of a table once you know its name. Issue the command:

```
   DESCRIBE <tableName>;
```

to learn about the attributes of relation `<tableName>`.

5 Inserting Tuples

Having created a table, we can insert tuples into it. The simplest way to insert is with the INSERT command:

```
 INSERT  INTO <tableName>
        VALUES( <list of values for attributes, in order> );
```

For instance, we can insert the tuple (10, 'foobar') into relation test by

```
INSERT INTO test VALUES(10, 'foobar');
```

6 Getting the Value of a Relation

We can see the tuples in a relation with the command:

```
SELECT *
FROM <tableName>;
```

For instance, after the above create and insert statements, the command

```
SELECT * FROM test;
```

produces the result

```
         I          S
    -------- ----------
        10     foobar
```

7 Setting column width

When you display the rows of a table, the values may run out of space in the display screen. To prevent this you may format the column width when displaying rows of a table. The corresponding command is:

```
COLUMN <culmnName> FORMAT <format_specifier>;
```

We suggest you execute

```
column ID format 0000;
```

This will print ID column in 4 digits with leading zero if necessary.

Another useful formatting specifier is a15 (for alpha numeric 15 characters). The corresponding command is:

```
column name format a15;
```

8 Getting Rid of Your Tables

To remove a table from your database, execute

```
DROP TABLE <tableName>;
```

We suggest you execute

```
DROP TABLE test;
```

after trying out this sequence of commands to avoid leaving a lot of garbage around that will be still there the next time you use the Oracle system.

9 Quitting sqlplus

To leave sqlplus, type

```
quit;
```

in response to the `SQL>` prompt.

10 Loading data from data file

Sometimes it is useful to load a bulk data already saved from some external application such as Microsoft Excel. To load such bulk data, wee need to perform the following steps.

10.1 Creating the Control File

A simple control file has the following form:

```
LOAD DATA
INFILE <dataFile>
APPEND INTO TABLE <tableName>
FIELDS TERMINATED BY '<separator>'
(<list of all attribute names to load>)
```

- `<dataFile>` is the name of the data file. If you did not give a file name extension for `<dataFile>`, Oracle will assume the default extension ".dat". Therefore, it is a good idea to name every data file with an extension, and specify the complete file name with the extension.
- `<tableName>` is the name of the table to which data will be loaded. Of course, it should have been created already before the bulk load operation.
- The optional keyword APPEND says that data will be appended to `<tableName>`. If APPEND is omitted, the table must be empty before the bulk load operation or else an error will occur.
- `<separator>` specifies the field separator for your data file. This can be any string. It is a good idea to use a string that you know will never appear in the data, so the separator will not be confused with data fields.

Finally, list the names of attributes of `<tableName>` that are set by your data file, separated by commas and enclosed in parentheses. This list need not be the complete list of attributes in the actual schema of the table, nor must it be arranged in the same order as the attributes when the table was created – sqlldr will match attributes to by their names in the table schema. Any attributes unspecified in the list of attributes will be set to NULL.

As a concrete example, here are the contents of a control file test.ctl:

```
LOAD DATA
INFILE test.dat
INTO TABLE test
FIELDS TERMINATED BY ','
(i, s)
```

10.2 Creating the Data File

Each line in the data file specifies one tuple to be loaded into `<tableName>`. It lists, in order, values for the attributes in the list specified in the control file, separated by `<separator>`. As a concrete example, test.dat might look like:

```
1,foo
2,bar
3,baz
```

Recall that the attribute list of test specified in test.ctl is (i, s), where i has the type int, and s has the type char(10). As the result of loading test.dat, the following tuples are inserted into test:

```
(1, 'foo')
(2, 'bar')
(3, 'baz')
```

10.3 Loading Your Data

The Oracle bulk loader is called `sqlldr`. It is a DOS-level command, i.e., it should be issued directly from your command prompt, rather than within sqlplus. A bulk load command has the following form:

```
sqlldr <yourName> control=<ctlFile> log=<logFile> bad=<badFile>
```

Everything but sqlldr is optional – you will be prompted for your username, password, and control file. `<ctlFile>` is the name of the control file. If no file name extension is provided, sqlldr will assume the default extension ".ctl". The name of the data file is not needed on the command line because it is specified within the control file. You may designate `<logFile>` as the log file. If no file name extension is provided, ".log" will be assumed. sqlldr will fill the log file with relevant information about the bulk load operation, such as the number of tuples loaded, and a description of errors that may have occurred. Finally, you may designate `<badFile>` as the file where bad tuples (any tuples for which an error occurs on an attempt to load them) are recorded (if they occur). Again, if no file extension is specified, Oracle uses

".bad". If no log file or bad file are specified, sqlldr will use the name of the control file with the .log and .bad extensions, respectively.

As a concrete example, if *rahman* wishes to run the control file test.ctl and have the log output stored in test.log, then he should type

```
sqlldr rahman control=test.ctl log=test.log
```

11 Executing SQL From a File

Instead of executing SQL commands typed at a terminal, it is often more convenient to type the SQL command(s) into a file and cause the file to be executed.
To run the file foo.sql, type:

```
@foo
```

sqlplus assumes by default the file extension ".sql" if there is no extension. So you could have entered `@foo.sql` at the `SQL>` prompt, but if you wanted to execute the file bar.txt, you would have to enter `@bar.txt`.

You can also run a file at connection by using a special form on the command line. The form of the command is:

```
sqlplus <yourName>/<yourPassword> @<fileName>
```

For instance, if user *rahman*, whose password is *password*, wishes to execute the file foo.sql, then she would say:

```
sqlplus rahman/password @foo
```

Notice that this mode presents a risk that rahman's password will be discovered, so it should be used carefully.

NOTE: If you are getting an error of the form "Input truncated to 2 characters" when you try to run your file, try putting an empty line at the bottom of your .sql file. This seems to make the error go away.

12 Recording Your Session

sqlplus provides the command spool to save query results to a file. At the`SQL>` prompt, you say:

```
spool foo;
```

and a file called foo.lst will appear in your current directory and will record all user input and system output, until you exit sqlplus or type:

```
spool off;
```

Note that if the file foo.lst existed previously, it will be overwritten, not appended.

13 Help Facilities

SQL*Plus provides internal help facilities for SQL*Plus commands. No help is provided for standard SQL keywords. To see a list of commands for which help is available, type help topics or help index in response to the SQL¿ prompt. To then look up help for a particular keyword (listed in the index), type help followed by the keyword. For example, typing help accept will print out the syntax for the accept command.

The output from help, and in general, the results of many SQL commands, can be too long to display on a screen. You can use

```
set pause on;
```

to activate the paging feature. When this feature is activated, output will pause at the end of each screen until you hit the ”return” key. To turn this feature off, use

```
set pause off;
```

14 Creating User/granting access

You can create a brand new user (after loggig in as `sqlplus sys as sysdba`) as follows:

```
create user ashikur
identified by password
profile default
default tablespace users
temporary tablespace temp
account unlock;
```

After that, grant users the necessary permissions as follows:

```
 grant dba to ashikur with admin option;
```

Acknowledgement

Part of this document was taken from the document written originally for Prof. Jeff Ullman’s CS145 class in Autumn, 1997.