

COSC 344 Lab for Week 3

Overview

The purpose of this lab is to learn ways of loading data into Oracle. You will also find out some tips on using the DATE data type. You will need this knowledge for the assignments.

How to Load Data into Oracle

You will practice the following three ways to load data into Oracle:

1. Use SQL INSERT commands interactively
2. Use a script containing SQL INSERT commands
3. Use the Oracle SQL Loader

The first approach should be obvious to you, or it will be after we cover the SQL lectures. The other two will be explored to various levels in the following sections.

Using a Script Containing SQL INSERT Commands

In the first lab, you created a file (or script) of SQL commands and then read the script into Oracle. Part of that file was a series of three INSERT commands. That is the way we use a script to load Oracle.

If you are completely loading a table, it is often handy to include a DROP TABLE and CREATE TABLE command to make sure you start with a blank table.

Type the following to create a table to experiment with:

```
DROP TABLE lab2a;  
  
CREATE TABLE lab2a  
  (i int,  
   r number(6,2),  
   txt varchar2(20),  
   when date);
```

Now use your favorite editor to create a script to load about 10 rows into the table. You might want to start with one row until you get the syntax correct. Look at the first lab for the form of the insert command.

You will probably have some trouble with the date, in particular, the TO_DATE function, which is used to convert date/time from string format to DATE data type. Notes on the DATE format and TO_DATE function are given on page 4, and the sample script on page 5 gives some specific examples.

Using the Oracle SQL Loader

The Oracle SQL Loader is a program like SQL. It is **executed from the UNIX prompt**, not from within SQL. It uses a control file and a data file to load data into tables.

When data is loaded from a script with SQL commands, the SQL parsing system is involved. Use of the Loader bypasses this route and results in a more efficient load. This part of the lab will describe a simple use of the SQL Loader. It has many more capabilities than we will use here.

For the examples below, assume we have a table *xyz*, which has three attributes as follows:

Attribute Name	Data type
i	int
s	varchar(18)
d	date

Syntax for the control file

A simple control file is:

```
LOAD DATA
INFILE 'loader1.dat'
INTO TABLE xyz
FIELDS TERMINATED BY ','
(i, s, d DATE "dd-mon-yyyy")
```

The LOAD DATA line is required at the beginning of the control file. The line, INFILE 'loader1.dat' tells the SQL Loader to get the data from the file *loader1.dat*. The next line indicates to load the data into table, *xyz*, which must already exist. The FIELDS TERMINATED line tells the SQL Loader to expect the fields (attributes) to be separated by a comma. The last line tells the loader the order of the fields in the data file. The syntax for the date is required in that form. The order of the fields does not have to be the same as the order used when the table was created. The orders of the fields **MUST** be the same as the order of the data in the data file.

Contents of a data file

A possible data file for the above example is:

```
1,string 1,01-may-2000
2, string 2 ,02-may-2002
3, 'string 3', 03-may-2003
```

A comma separates the fields (that's what we said in the control file). The dates are in the format specified in the control file.

There is a gotcha with respect to character strings. **EVERYTHING** between the commas is interpreted to be data. You do not use quote to delimit a string in the SQL Loader. So in line

2, the data loaded into the table includes the leading and trailing space as well. In line 3, the leading space and the two single quotes appear as data in the table. You've been warned. ☺

Including data in the control file

There is also an option to combine the data and control files. We will not explore that option. If you are interested, consult the documentation.

Invoking the SQL Loader

Before we can invoke the SQL loader, the table must exist. Type the following at the SQL> prompt to create the table.

```
CREATE TABLE xyz (i INT, s VARCHAR2(20), d DATE);
```

The SQL Loader is **started from the Unix prompt**, not from within SQL. Type:

```
sqlldr control=control_filename log=log_filename
```

where `control_filename` is the name you gave your control file and `log_filename` is the name of a file that messages will be written into. Pick the name for the logfile carefully as it is overwritten without warning if it already exists.

The SQL Loader will prompt for your Oracle username and password. It will then attempt to process the data.

You need to look at the log file to get the results. It will tell you the number of rows processed, loaded, and rejected. If you are lucky, you might make sense of the error messages. Basically if you have rejected rows, something went wrong and some of the data was not loaded.

The use of extensions on control files, log files, and/or data files is up to you. Oracle does not specify any defaults. We suggest to reserve `.sql` for SQL scripts, and use `.dat` for data file, `.log` for log files, and `.ctl` for control file. You can use the same basic name of all three. For example, `test.ctl`, `test.dat` and `test.log`. It is your choice, but you should be consistent.

DATE and TO_DATE

SQL specifies

- DATE: Year, Month and Day in the form YYYY-MM-DD
- TIME: Hour, Minute and Second in the form HH:MM:SS
- TIMESTAMP: include both DATE and TIME plus a minimum of six positions for decimal fractions of seconds in the form YYYY-MM-DD HH:MI:SS[.sssss]

Oracle uses DATE for both date and time

- DD - number of days in month (e.g., 24)
- MM – numeric month (e.g., 07)
- MON – abbreviated month name (e.g., JUL)
- MONTH – full month name (e.g., JULY)
- YY - last two digits of year (e.g., 12)
- YYYY - all digits of year (e.g., 2012)
- HH (or HH12) - hours of the day in 12 hour format
- HH24 - hours of the day in 24 hour format
- MI - minutes of the hour
- SS - seconds of the minute

Use TO_DATE to specify the format

- TO_DATE('string', 'format')
- Must be used when entering dates
- e.g, TO_DATE('20-03-2000','dd-mm-yyyy')

```
INSERT INTO x VALUES (..., ..., TO_DATE('27-Jul-2010', 'dd-mon-yyyy'));
```

Use TO_CHAR to format the output of a date

- TO_CHAR(attr, 'format')
- TO_CHAR(bdate, 'dd/mon/yy')

```
SELECT TO_CHAR (bdate, 'dd/mon/yy') FROM EMPLOYEE;
```

Sample Script

```
DROP TABLE lab2a;

CREATE TABLE lab2a
  (i int,
   r number(6,2),
   txt varchar2(20),
   when date);

INSERT INTO lab2a VALUES
  (1, 1.1, 'String 1',
   TO_DATE('01-May-2001', 'dd-mon-yyyy') );
INSERT INTO lab2a VALUES
  (2, 1.1, 'String 2',
   TO_DATE('02-jun-2001', 'dd-mon-yyyy') );
INSERT INTO lab2a VALUES
  (5, 5.5, 'Long String 5',
   TO_DATE('25-Dec-2001', 'dd-mon-yyyy') );
INSERT INTO lab2a VALUES
  (4, 4.4, 'Short String 4',
   TO_DATE('11-Nov-2005', 'dd-mon-yyyy') );
INSERT INTO lab2a VALUES
  (3, 3.3, 'Nothing',
   TO_DATE('01-jan-2000', 'dd-mon-yyyy') );

COMMIT;
```

Assessment

The purpose of this assessment is to test if you can use the Oracle SQL loader to load data into a table.

To prepare for this assessment, you need to make sure the table `lab2a` is created, and then prepare the control file and the data file. There should be exactly 10 rows of data in the data file. After all of these have been done, test it and make sure there is no error in loading data by checking the log file.

[Marking] This assessment will be marked in the labs in Week 4, and it is worth 10 marks. You may ask the lab demonstrator to get it marked in this lab if you have completed it.

When you attend the lab in Week 4, just do the lab work given in the labnote as usual, and the lab demonstrator will approach you to mark this assessment. You need to show the lab demonstrator the control file and the data file, and demonstrate how to load the data into table `lab2a` using the Oracle SQL loader. If you haven't finished this assessment when the lab demonstrator approaches you to mark it, it will be marked based on what you have done by that time, and you may lose some marks. If there is any error in terms of loading data, you may also lose marks.