Oracle (PL/SQL)

Lesson 09: SQL * Loader

Capgemini

Lesson Objectives

- **To understand the following topics:**
  - Outline of SQL*Loader, an Oracle-supplied utility
  - SQL * Loader Environment

What Is SQL * Loader?

- SQL*Loader is a bulk loader utility used for moving data from external files(os) into the Oracle database.
- SQL*Loader supports various load formats, selective loading, and multi-table loads.

Usage Of SQL * Loader

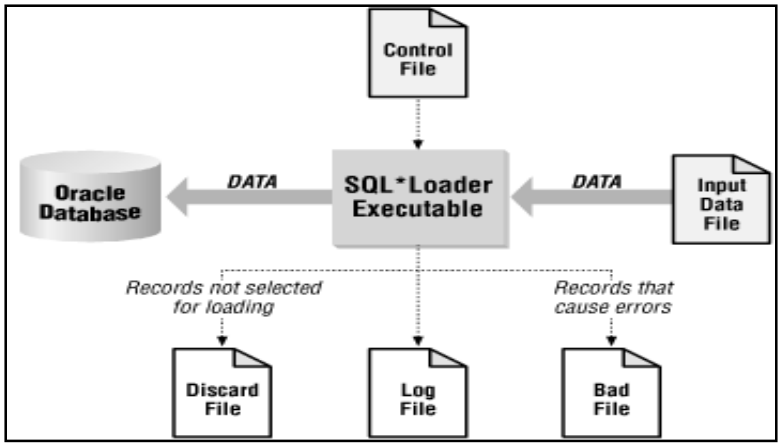
- **SQL*Loader can be used for the following utilities:**
  - To load data from multiple datafiles during the same load session.
  - To load data into multiple tables during the same load session.
  - To specify the character set of the data.
  - To selectively load data (you can load records based on the records' values).
  - To manipulate the data before loading it, using SQL functions.
  - To generate unique sequential key values in specified columns.

Usage Of SQL * Loader

- To use the operating system's file system to access the datafiles
- To load data from disk, tape, or named pipe
- To generate sophisticated error reports, which greatly aids troubleshooting
- To load arbitrarily complex object-relational data
- To use secondary datafiles for loading LOBs and collections

SQL * Loader Environment

- A typical SQL*Loader session takes a control file as input. The input file controls the behavior of SQL*Loader and one or more datafiles.
- The output of SQL*Loader is an Oracle database (where the data is loaded), a log file, a bad file, and potentially a discard file.



The Control File

- **The SQL*Loader Control file is the text file which is a key to any load process.**
- **The Control file provides the following information to Oracle for data load:**
  - Datafile name, location, and format
  - Character sets used in the datafiles
  - Datatypes of fields in those files
  - Information on how each field is delimited
  - Information on the tables and columns to be loaded

The Control File

- You must provide the control file to SQL*Loader so that the tool knows several things about the data it is about to load.
- Data and control file information can be provided in the same file or in separate files.
- Some items in the control file are mandatory, such as which tables and columns to load and how each field is delimited.

The Log File

- **The log file is a record of SQL*Loader's activities during a load session.**
- **The log file contains information as given below:**
  - The names of the control file, log file, bad file, discard file, and data file
  - The values of several command-line parameters
  - A detailed breakdown of the fields and datatypes in the data file that was loaded
  - Error messages for records that cause errors
  - Messages indicating when records have been discarded

The Log File

- A summary of the load that includes the following:
- The number of logical records read from the data file
- The number of rows rejected because of errors
- The number of rows discarded because of selection criteria
- The time elapsed for the load

The Bad File & Discard File

- **Whenever you insert data into a database, you run the risk of that insert failing because of some type of error.**
- **Integrity constraint violations undoubtedly represent the most common type of error.**
  - However, other problems, such as the lack of free space in a tablespace, can also cause insert operations to fail.
- **Whenever SQL*Loader encounters a database error while trying to load a record, it writes that record to a file known as the “Bad file”.**

The Bad File & Discard File

- **Discard files are used to hold records that do not meet selection criteria specified in the SQL*Loader control file.**
  - By default, SQL*Loader will attempt to load all the records contained in the input file.
- **Records that do not meet the specified criteria are not loaded, and are instead written to a file known as the “Discard file”.**

Methods Of Invoking SQL*Loader

- **SQL*Loader can be invoked in one of the following three ways:**
  - sqlldr
  - sqlldr keyword=value [keyword=value ...]
  - sqlldr value [value ...]

Invoking SQL * Loader

- **Given below is a list of Command Line Parameters:**
  - USERID = {username[/password] [@net_service_name]|/}
    - Specifies the username and password to use when connecting to the database. The net_service_name parameter optionally allows you to connect to a remote database. Use a forward-slash character (/) to connect to a local database by using operating system authentication.
  - CONTROL = control_ file_name
    - Specifies the name, which may include the path, of the control file. The default extension is .ctl.
  - LOG = path_ file_name
    - Specifies the name of the Log file to generate for a load session. You may include a path as well. By default, the Log file takes on the name of the Control file, but with a .log extension. The Log file is written to the same directory as the Control file. If you specify a different name, the default extension is still .log.

Invoking SQL * Loader

- BAD = path_ file_name
  - Specifies the name of the Bad file. You may include a path as part of the name. By default, the Bad file takes the name of the Control file, but with a .bad extension. The Bad file is written to the same directory as the Control file.
- DATA = path_ file_name
  - Specifies the name of the file containing the data to load. You may include a path as part of the name. By default, the name of the Control file is used, but with the .dat extension. If you specify a different name, the default extension is still .dat

Invoking SQL * Loader

- DISCARD = path_ file_name
  - Specifies the name of the Discard file. You may include a path as part of the name. By default, the Discard file takes the name of the Control file, but it has a .dis extension. If you specify a different name, the default extension is still .dis.
- DISCARDMAX = logical_record_count
  - Sets an upper limit on the number of logical records that can be discarded before a load will terminate. The limit is actually one less than the value specified for DISCARDMAX.

Invoking SQL * Loader

- SKIP = logical_record_count
  - Allows you to continue an interrupted load by skipping the specified number of logical records.
- LOAD = logical_record_count
  - Specifies a limit on the number of logical records to load. The default is to load all records. Since LOAD only accepts numeric values, it is not possible to explicitly specify the default behavior.
- ROWS = rows_in_bind_array
  - The precise meaning of this parameter depends on whether you are doing a direct path load or a conventional load.

SQL * Loader - Examples

- **Example 1:**
  - sqlldr username@server/password control=loader.ctl
    - The sample control file (loader.ctl) will load an external data file containing delimited data:

```
load data
infile 'c:\data\mydata.csv'
into table emp fields terminated by ","
optionally enclosed by '"' ( empno, empname, sal, deptno )
```

- **The mydata.csv file may look like the sample shown below:**
  - 10001,"Scott Tiger", 1000, 40
  - 10002,"Frank Naude", 500, 20

SQL * Loader - Examples

- **Example 2:**
  - Another sample Control file with in-line data formatted as fix length records is discussed in this example.
  - The trick is to specify “*” as the name of the data file, and use BEGINDATA to start the data section in the control file:

```
load data infile * replace
    into table departments ( dept position (02:05) char(4),
deptname position (08:27) char(20) )
    begindata
    COSC COMPUTER SCIENCE
    ENGL ENGLISH LITERATURE
    MATH MATHEMATICS
    POLY POLITICAL SCIENCE
```

SQL * Loader - Examples

- **Example 1:**
  - Step 1: Create the Relation
    - Create table customer in Oracle database Oracle account with the given structure.

Sql>Desc example

Name	Null ?	Type
custid		number
custname		varchar2(12)
age		number

SQL * Loader Case Examples

- Step 2:
  - Create Data File named customer.dat as shown below:
    - 10,“AAA”,20
    - 20,“BBB”,23
    - 30,“CCC”,41
    - 40,“DDD”,25

SQL * Loader Case Examples

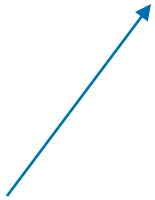
- Step 3:
  - Create sql loader control file named custcontrol.ctl with the given contents:

```
load data
infile 'd:\customer.dat'
into table customer
fields terminated by ',' optionally enclosed by ""
(custid, custname, age)
```

SQL * Loader Case Examples

- Step 4:
  - At command prompt, run SQL Loader by entering the following details:

C:\sqlldr scott/tiger@oracle9i control=d:\custcontrol.ctl



userid/password@Host String

Summary

- **In this lesson, you have learnt:**
  - SQL*Loader as an Oracle utility
  - SQL*Loader environment
    - Control File, Log File, Bad File, and Discard File
    - Invoking SQL*Loader Environment
    - Command Line Parameters

Review Question

- **Question 1: We can use SQL*Loader to load data into multiple tables during the same load session**
  - True / False
- **Question 2: A typical SQL*Loader session takes as input a discard file**
  - True/False
- **Question 3: The SQL*Loader control file is the text file which is a key to any load process.**
  - True/False

Review Question

- Question 4: The ___ file is a record of SQL*Loader's activities during a load session.
- Question 5: SQL*Loader Parameter ___ specifies a limit on the number of logical records to load.