

Backup, Recover & Data Import/Export

The reliability of modern computer hardware has greatly improved. Hardware continues to get faster, better and more reliable. Disk drive failure rates have decreased to about approximately 3% per year in 2002 according to IEEE¹. That means that disk drives were up and running reliably 97% of the time. Implementing technologies such as Self-Monitoring and Reporting Technology (SMART) or using replication techniques such as mirrored or parallel databases may mitigate the risk of a database failure. This would make the system even more reliable. Since disk drives or other hardware can still fail, access to reliable data can be in jeopardy because:

- Access to the database can be interrupted
- Data the database can be put into an inconsistent state

When a database is in an inconsistent state, the data is inaccurate, unreliable and useless for decision making. This situation can be a huge problem for any organization.

Additionally, users are human and humans make mistakes. Mistakes unfortunately can also cause inconsistencies in the database. It is the job of the database administrator to protect the database from failure and to ensure that it is consistent, free from errors. There are various methodologies that can be utilized to ensure that if an error occurs then recovery is possible.

Learning Outcomes

Management Of Information Technology - By completing this lab, you will be able to integrate technical and solution development concepts with the principles of IT governance, strategic alignment, user behavior and financial analysis. You will be able to apply these concepts in the analysis of complex management case studies and problems. You will be able to analyze, compare, evaluate, and clearly articulate the relative value of IT alternatives. **After completing this lab, you will be able to:**

- Administer a database by recommending and implementing procedures including database tuning, backup and recovery

Specifically, *after this lab you will be able to administer a database by recommending and implementing procedures for backup and recovery including the ability to:*

- Determine appropriate backup and recovery techniques
- Compare and contrast various techniques for backing up and restoring large databases
- Explain archiving and archive logs
- Implement Oracle 11g's backup and recovery tools
- Implement Oracle Utilities including
 - Import/Export Utility
 - SQL Loader Utility

¹ http://cmrr.ucsd.edu/smart/tech_papr/SmtPapTransReliFinalWeb.pdf

Backup

Let's explore the several Oracle backup techniques that are possible and could potentially save an organization from disaster. It is important to remember that not only is it important to perform a backup procedure, **it is also important to test the procedure**. This will validate the backup process and ensure recovery is possible.

There are several ways to perform backup procedures. Each method has both advantages and disadvantages. Deciding the best backup technique for your organization or even for specific applications within an organization may depend on several factors such as:

1. Is it possible to take the database down and make it unavailable to users?
2. If the system can be made unavailable, how long can it be down?
3. How volatile is the data?
4. How critical is the data? For example, can it be easily recreated?
5. How quickly must the database be recovered in the event of a failure?
6. Is the data valuable?

Cold vs. Hot Backups

What do I mean **cold** versus **hot**? What's the difference? Why does it matter? These are just a few of the questions that you are probably asking yourself. So let's explore two valid but different backup techniques.

Cold Backup - A cold backup is a backup that takes place when the database has been shut down. If the database has been shut down, then no one can be using the database. This is not a valid option for a database that must be available 24/7. This is one disadvantage of performing a cold backup. A major advantage is that a cold backup is a fairly simple type of backup procedure and requires no other software than what already provided by your operating system.

Let's take a look at what is required to perform a cold backup. The steps include:

1. Shut down the database
2. Copying the appropriate files **to a disk with the same block size**
3. Start up the database

So, let's explore the startup and shut down procedures available in Oracle 9i.

Startup/Shutdown

Sometimes it is necessary to start and stop the Oracle database manually using a command line interface. The can be accomplished using **SQL*Plus**. There are 3 shutdown options. They are:

1. **ABORT** – This option shuts down the database without saving any transactions or rolling back any unsaved transactions. All user sessions are immediately aborted. This option should only be used when the database is not functioning properly, as if you use this option you're going to **lose data**.
2. **IMMEDIATE** – This option waits for all pending transaction to complete or for all users to log off. It is a safer option, but still terminates user sessions.
3. **NORMAL** – This option waits for all users to log off. It is the most graceful and least disruptive. However, it also will take the longest time.

The shutdown command from SQL*Plus is

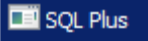
SHUTDOWN **option** where **option** is one of the 3 shutdown options listed above.

To start the database manually, you issue a startup command.

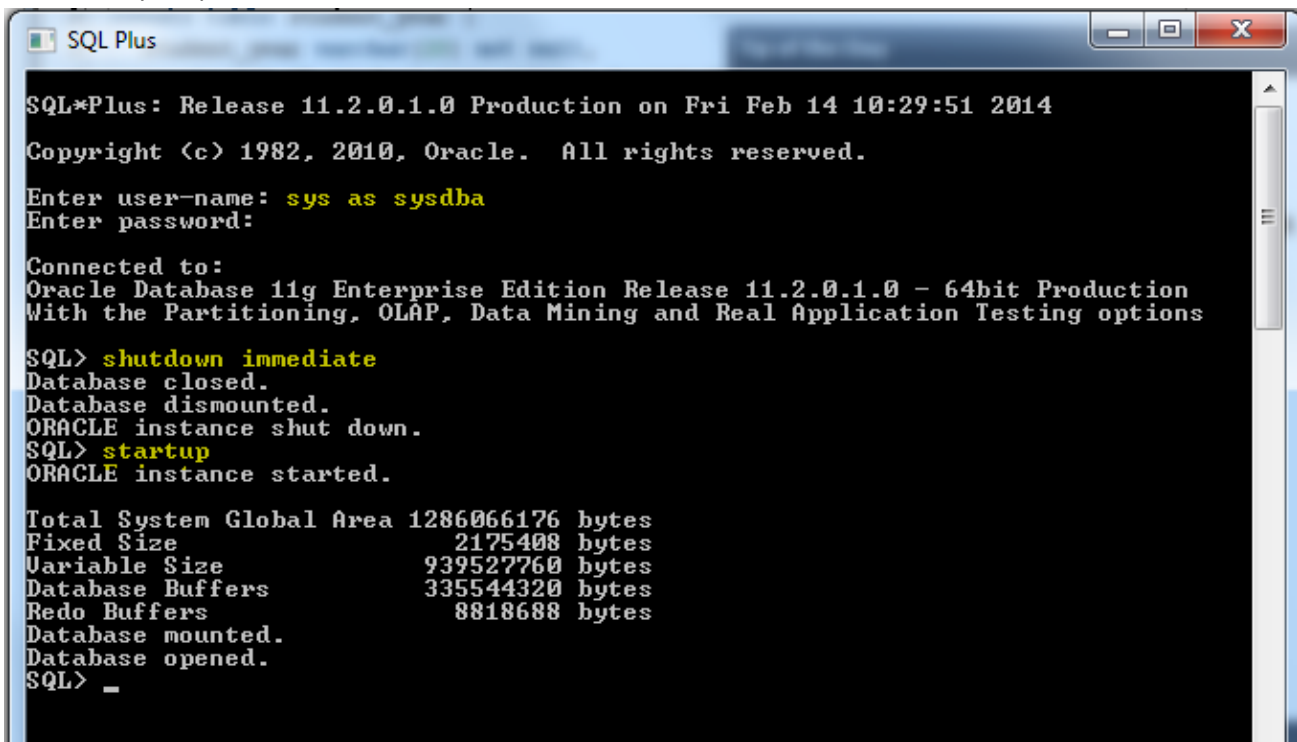
```
STARTUP PFILE = Path-To-Parameter-File  
(this is the init.ora file located in the Oracle Home directory)
```

To perform a manual shutdown or startup using SQL*Plus, you must log on to SQL*Plus as SYSDBA.

Try This:

Click the Windows **Start** button → In the Search box, Type **SQL Plus** → Click on 

This will open the **SQL Plus** command window. To log-on enter **sys as sysdba** at the user-name prompt, then enter your password. Enter the commands as follows.



```
SQL*Plus: Release 11.2.0.1.0 Production on Fri Feb 14 10:29:51 2014  
Copyright (c) 1982, 2010, Oracle. All rights reserved.  
Enter user-name: sys as sysdba  
Enter password:  
Connected to:  
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
SQL> shutdown immediate  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> startup  
ORACLE instance started.  
  
Total System Global Area 1286066176 bytes  
Fixed Size 2175408 bytes  
Variable Size 939527760 bytes  
Database Buffers 335544320 bytes  
Redo Buffers 8818688 bytes  
Database mounted.  
Database opened.  
SQL> _  
SQL> _
```

IMPORTANT: Make sure your database has started back up before you continue (Check the status in OEM). If you have problems, you can always try starting it back up in OEM.

What should you Backup?

The **shutdown** command will shut the database after all users have logged off. **Shutdown Immediate** will log users off and shutdown the database immediately. After the database has been shutdown, none of the database files are being accessed. You can now copy the required database files using any valid copy procedure. This means that you can simply copy the files into a backup folder using Windows Explorer, the copy command at the command prompt or any other valid copy procedure. The files that you must copy include:

1. Parameter file (init.ora) – which store database startup specifications
2. Control files (control01ctl, control02.ctl, control03.ctl) – which contains the data regarding the database status
3. Data files (*userdata.dbf*) – the data values that you or your users have saved in your database tables as well as other user-created database objects

NOTE: Do not copy the log files. You do not need the log file files because you shutdown the database. *The log files are only active when there are outstanding transactions.* The control and data files haven't changed between the shutdown and the backup of these files. You have a complete snapshot of database. You now have a cold backup that can be used to restore the database in case of a failure.

Do This: Perform a cold backup. Note: You'll have to figure out the details and also recall the locations of your database files and control file.

- 1) Make a folder c:\backup
- 2) Shutdown the Oracle database
- 3) COPY all of your .DBF files to c:\backup
- 4) COPY your control file to c:\backup
- 5) Startup the Oracle database

Hot Backup

A hot backup, on the other hand, takes place when the database may still be in use. It does not require that the database be shut down. However, it does require that archiving is taking place which creates overhead and can even cause the database to "hang up" if not done properly.

Creating a hot backup is very similar to creating a cold backup. You must backup the same files (parameter file, control files, and data files). Additionally, you must place the database instance in ARCHIVELOG mode so that archive files are being created. And, the database must be instructed to write the redo log records to the archive log files.

Here's the general procedure for creating a hot backup:

1. Place the database in ARCHIVELOG mode
2. Backup these files:
 - a. Parameter file
 - b. Control files
 - c. Data files
3. Archive the redo log files
4. Backup the archive

Placing the database in ARCHIVELOG mode

To perform a hot backup, you must have Archiving enabled. There are advantages and disadvantages to Archiving. As you recall, archiving backs up the redo log files. The redo log files contain the data required for the database to rollback transactions. This is important data needed for a database to maintain consistency. In effect, archiving the log files creates another level of redundancy and a requirement for performing a hot backup. A disadvantage, however, is that the files require additional disk space and writing to these files requires more processing time. There is a performance price to pay when enabling archiving. However, hot backups are not possible without archiving enabled.

So before you can perform hot backups, your Oracle database must be in ARCHIVELOG mode. If it is not, you get a **media recovery not enabled** error message when you attempt a hot backup. Here's how you check for ARCHIVELOG mode. **Type This into SQL Plus:**

```
15:44:31 SQL> archive log list;
Database log mode       No Archive Mode
Automatic archival      Disabled
Archive destination     USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 79
Current log sequence    81
```

Archive Mode is Disabled.

If it says "No Archive Mode" (as it does in my screenshot) then you must do the following to put it in archive mode: **Type these commands into SQL Plus:**

```
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database archivelog;
SQL> alter database open;
```

Shutdown Oracle

Start in Mount Mode (restricted access)

Change database to Archive Mode

Place Db back in full "open" mode.

After this change the database backup should run ok. This is a one-time setting and should not need to be reconfigured. You can verify your database is configured properly now by executing the **archive log list;** command in SQL Plus. **Try this:**

```
15:50:17 SQL> archive log list;
Database log mode       Archive Mode
Automatic archival      Enabled
Archive destination     USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 79
Next log sequence to archive 81
Current log sequence    81
```

You can also do this in OEM:

OEM → Home → Availability → Recovery Settings

Database Instance: orcl > Logged in As SYS

Recovery Settings Show SQL Revert Apply

Instance Recovery

The fast-start checkpointing feature is enabled by specifying a non-zero desired mean-time to recover (MTTR) value, which will be used to set the FAST_START_MTTR_TARGET initialization parameter. This parameter controls the amount of time the database takes to perform a checkpoint. When fast-start checkpointing is enabled, Oracle automatically maintains the speed of checkpointing so that the recovery time to a consistent state is less than the value to 0 will disable this functionality.

Current Estimated Mean Time To Recover (seconds) 19
 Desired Mean Time To Recover Minutes

Media Recovery

The database is currently in NOARCHIVELOG mode. In ARCHIVELOG mode, hot backups and recovery to the latest time are possible for archived redo log files. If you change the database to ARCHIVELOG mode, you should perform a backup immediately. In NOARCHIVELOG mode, hot backups are possible and data may be lost in the event of database corruption.

ARCHIVELOG Mode

Log Archive Filename Format*

Number	Archived Redo Log Destination	Status	Type
1	USE_DB_RECOVERY_FILE_DEST	VALID	Local

Here you can see that the ArchiveLog Mode is not select.

Check the box then select Apply.

Hot Backup a Control file

To backup a control file that is currently in use, you can create a trace file that contains a script to rebuild the control file. This file can then be viewed using any text editor, like Notepad. To do this, from your logged on copy of SQL Plus and enter the following command:

```
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

Hot Backup Datafiles for Tablespaces

To perform a hot backup, first, you instruct the database to begin the backup of the desired tablespace. Then, using the operating system's copy facilities or copy commands, you copy the data file for that tablespace. Finally, you instruct the database to resume writing to the tablespace.

For example to perform a backup for the IST469 tablespace the steps would be as follows:

Try This:

1. Log onto SQL*Plus with sufficient privileges as SYSDBA (you've already done this)
2. Issue the command: **ALTER TABLESPACE IST469 BEGIN BACKUP;**
3. From the file system Copy all the **IST469##.DBF** files to **c:\backup**
4. Issue the command: **ALTER TABLESPACE IST469 END BACKUP;**

Hot Backup Archive files

Since this is a hot backup, users are still writing to the database while we performed the backup process. Yes, I understand its only YOU accessing this database in the lab, so you'll have to use your imagination. One important thing you should note is we did not have to bring down the database to do this.

While the data files were in the process of being backed up there is a good change some user transactions were logged. Therefore, to capture all of the data, it is now necessary to archive the redo log files and the also backup the archive files.

Therefore to complete the process:

Try This:

1. Log onto SQL*Plus with sufficient privileges SYSBA (you're already done this)
2. Archive the redo log by forcing a log switch. Issue the command:

```
ALTER SYSTEM ARCHIVE LOG CURRENT;
```

3. Copy the archive files (**REDO##**) to **c:\backup**

You now have a complete backup. Since you designate the tablespace to be backed up, it is possible to create hot backups of just specific tablespaces.

VERY IMPORTANT – Turn Off Archive Mode

If you leave archive mode on your control files will eventually balloon out of control, reaching 4TB if you do not routinely execute the **ALTER SYSTEM ARCHIVE LOG CURRENT;** command to flush them. Since this is a development environment (and you are a student) odds are you are not going to remember to do this routinely, so execute these commands in SQL plus to turn off archive mode:

```
SQL> shutdown immediate;

SQL> startup mount;

SQL> alter database noarchivelog;

SQL> alter database open;

SQL> archive log list;
```

The last command should once again display no archive mode:

```
SQL> archive log list;
Database log mode          No Archive Mode
Automatic archival        Disabled
Archive destination       USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 50
Current log sequence      52
```

Recovery

The purpose of creating a backup is to recover or restore the database to a consistent state in the case of a database failure. Recovery using any backup will restore the database to the state that it was in when the backup took place. To recover using a cold backup, this would be the procedure:

1. Shut down the database
2. Restore the control file and data files, by copying over the existing files
3. Restart the database

It sounds simple. However, if all of the files were not backed up at the same time, this may not produce the desired results! All the user table spaces are dependent on the SYS database.

Recovery using a hot backup requires the following steps:

1. Shutdown the database
2. Restore to the latest hot backup
3. Restart the database with restrictions session privileges
4. Recover the changes from the archive files
5. Restart the database

As you can probably tell, this can become a complex task, especially if the organization has many databases. To help manage the backup/recovery process, Oracle provides a utility called RMAN (Recovery Manager.) This is a command line utility and can be extremely useful.

RMAN - an Oracle supplied utility that is executed from the operating system command line using the run command (or the command prompt window). This utility can use either a recovery catalog or the target database's control file to store the recovery process metadata (configuration parameters.)

Oracle EM for Oracle 11g contains a graphical interface that can assist with the configuration and maintenance of back up and recovery.

NOTE: We will not execute any recovery procedures are part of the lab.

Oracle Utilities Import and Export

There are other alternatives and processes that may be useful and serve as alternatives to either a cold or hot backup. Oracle provides several useful utilities that can also be used to save data and/or transport data from one database to another. These utilities are documented in the Oracle Utilities Manual. They are command line utilities and can be extremely useful. They are:

1. **IMP** – IMPORT UTILITY
This utility can be used to import tables from one schema to another.
2. **EXP** – EXPORT UTILITY
This utility can be used to copy tables to a file that can be used to import to another schema.
3. **SQLLDR** – SQLLOADER
This utility can be used to load data into a table from a text file format.

Let's look at each of these utilities.

Export

You can create an export of an Oracle database that can then be imported into another schema. In this screenshot, I created an export of the **customers** table from the **FUDGEMART** schema, and stored the export in the file **fudgemart_customers.dmp**

Feel free to **Try It** for yourself. First open a **command prompt**


```
C:\Command Prompt
C:\Users\admin>exp
Export: Release 11.2.0.1.0 - Production on Fri Sep 21 09:57:23 2012
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Username: FUDGEMART/SU2orange@orc1
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Enter array fetch buffer size: 4096 >

Export file: EXPDAT.DMP > fudgemart_customers.dmp
(1)E(ntire database), (2)U(sers), or (3)T(ables): (2)U > 3

Export table data (yes/no): yes > y
Compress extents (yes/no): yes > y

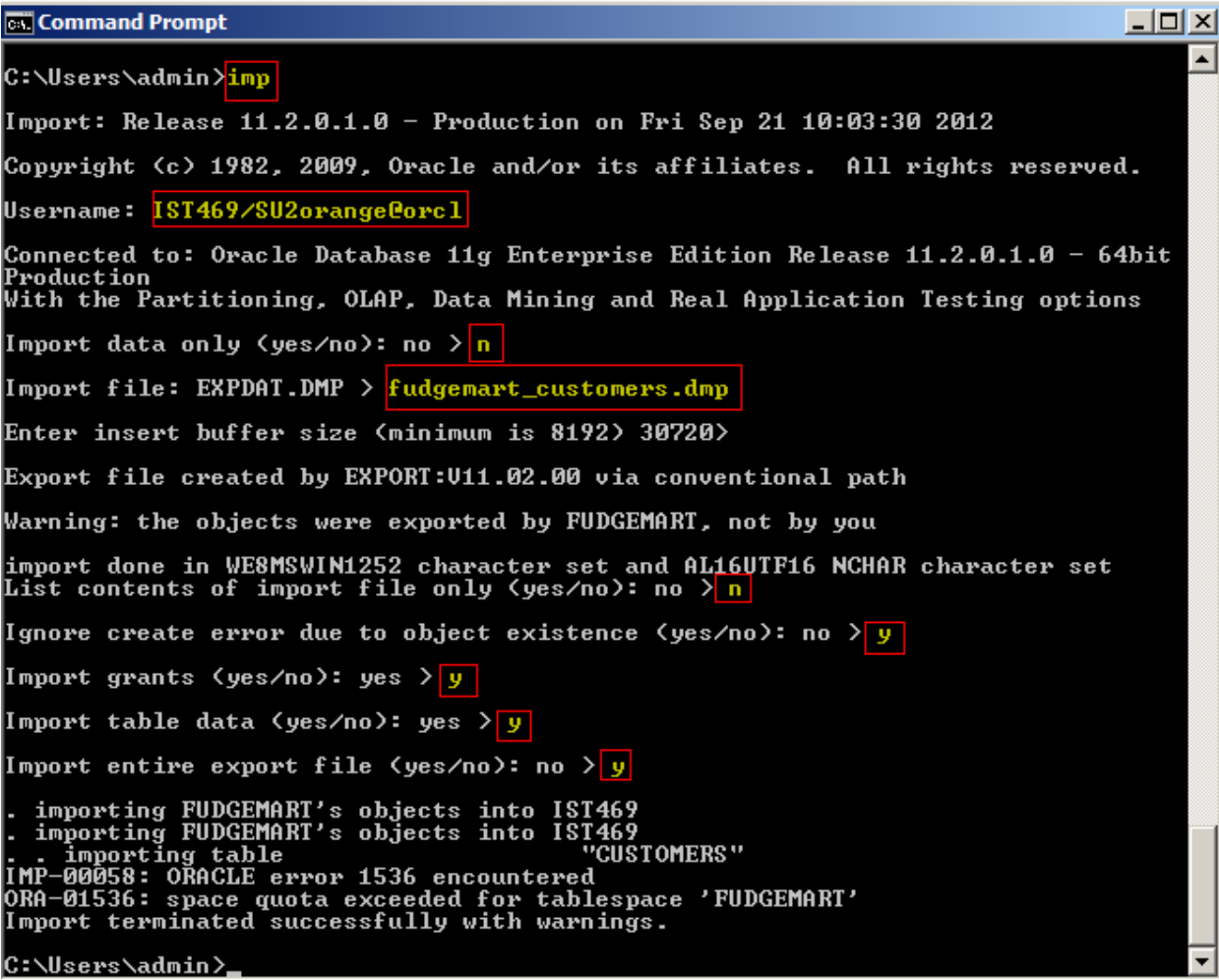
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
About to export specified tables via Conventional Path ...
Table(T) or Partition(T:P) to be exported: (RETURN to quit) > customers
. . exporting table                CUSTOMERS                25 rows exported
Table(T) or Partition(T:P) to be exported: (RETURN to quit) >

Export terminated successfully without warnings.
C:\Users\admin>
```

Import

Now, I demonstrate an import . Using the IMP utility, I will import the **customers** table inside the **fudgemart_customers.dmp** export file into the **IST469** schema.

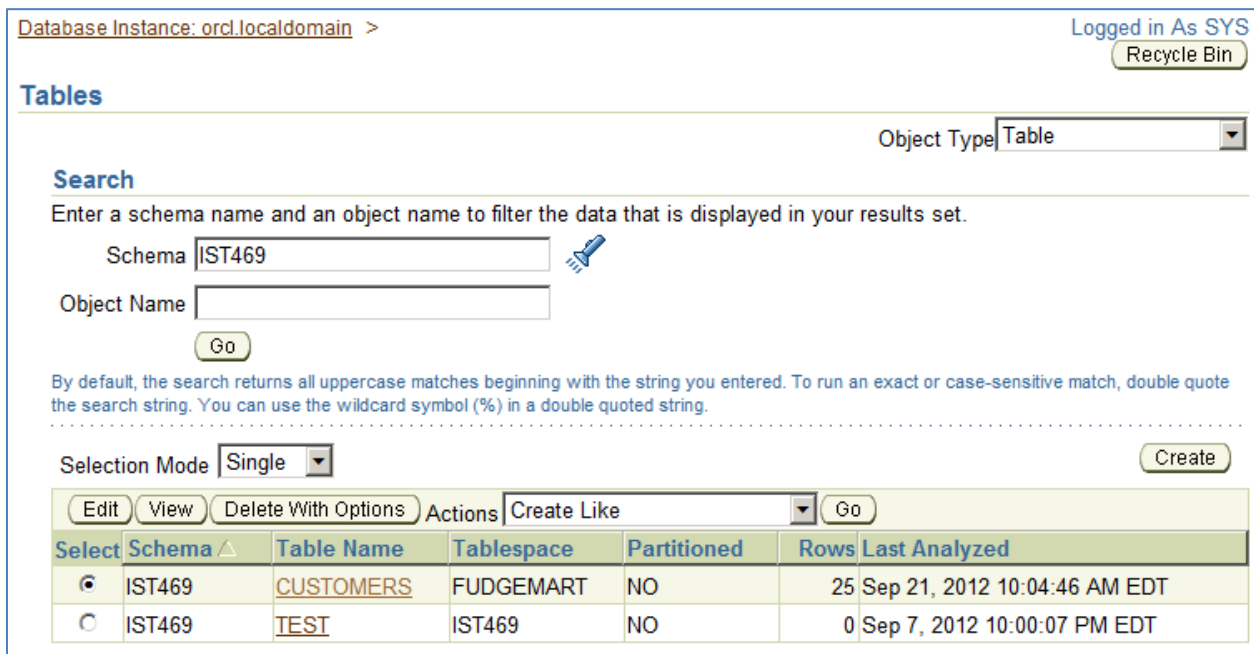
Feel free to **Try It** for yourself. From the same **command prompt**



```
C:\Users\admin>imp
Import: Release 11.2.0.1.0 - Production on Fri Sep 21 10:03:30 2012
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.
Username: IST469/SU2orange@orcl
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit
Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Import data only (yes/no): no > n
Import file: EXPDAT.DMP > fudgemart_customers.dmp
Enter insert buffer size (minimum is 8192) 30720
Export file created by EXPORT:U11.02.00 via conventional path
Warning: the objects were exported by FUDGEMART, not by you
import done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
List contents of import file only (yes/no): no > n
Ignore create error due to object existence (yes/no): no > y
Import grants (yes/no): yes > y
Import table data (yes/no): yes > y
Import entire export file (yes/no): no > y
. importing FUDGEMART's objects into IST469
. importing FUDGEMART's objects into IST469
. . importing table "CUSTOMERS"
IMP-00058: ORACLE error 1536 encountered
ORA-01536: space quota exceeded for tablespace 'FUDGEMART'
Import terminated successfully with warnings.
C:\Users\admin>_
```

NOTE: There's an error at the end because the **IST469** user does not have a quota into the **FUDGEMART** tablespace. If we did assign rights the data would have imported.

You can verify the table is there in OEM: Home → Schema → Tables →




Database Instance: orcl.localdomain > Logged in As SYS
Recycle Bin

Tables

Object Type: Table

Search

Enter a schema name and an object name to filter the data that is displayed in your results set.

Schema: 

Object Name:

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

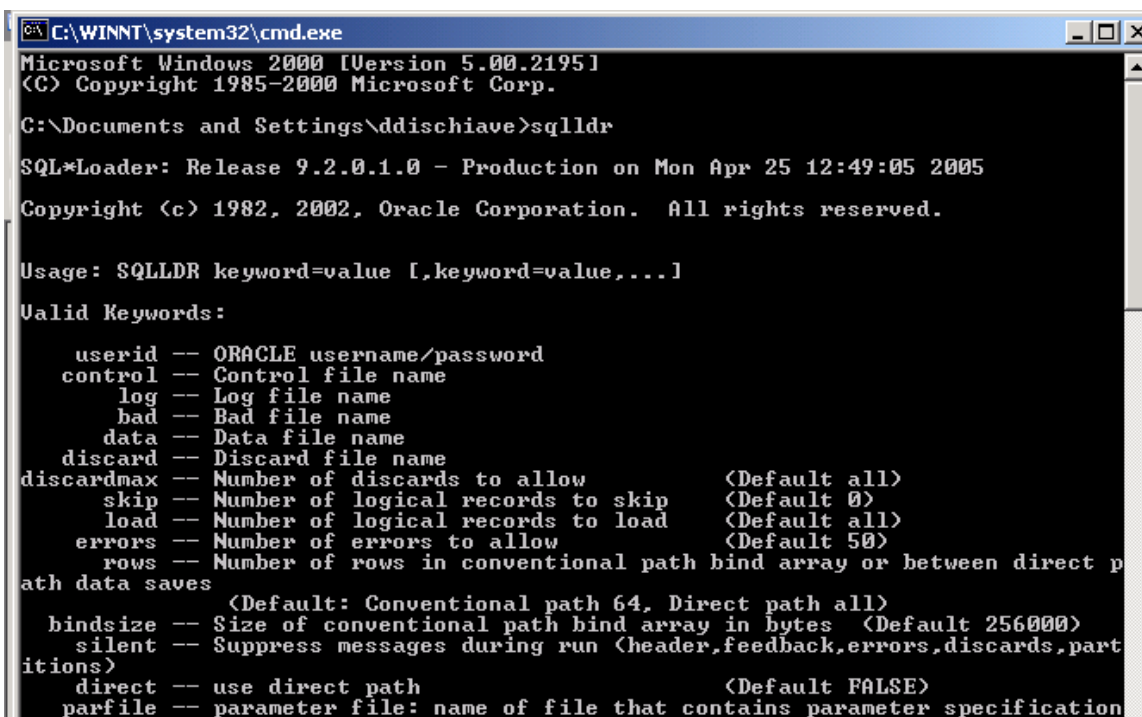
Selection Mode:

Actions:

Select	Schema	Table Name	Tablespace	Partitioned	Rows Last Analyzed
<input checked="" type="radio"/>	IST469	<u>CUSTOMERS</u>	FUDGEMART	NO	25 Sep 21, 2012 10:04:46 AM EDT
<input type="radio"/>	IST469	<u>TEST</u>	IST469	NO	0 Sep 7, 2012 10:00:07 PM EDT

SQL LDR

The Oracle import and export utilities work great if you want to copy database objects from one schema to another. However, what if you have a sequential file and you would like to enter the data into an Oracle table. **SQLLDR** is a utility that allows data to be loaded into tables from a text file. There are a variety of formats available. Parameter and control files can be created to facilitate reuse. Simply entering **sqlldr** at the **command prompt** will display the **help screen**.



```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\ddischiave>sqlldr

SQL*Loader: Release 9.2.0.1.0 - Production on Mon Apr 25 12:49:05 2005
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Usage: SQLLDR keyword=value [,keyword=value,...]

Valid Keywords:

  userid -- ORACLE username/password
  control -- Control file name
  log -- Log file name
  bad -- Bad file name
  data -- Data file name
  discard -- Discard file name
  discardmax -- Number of discards to allow <Default all>
  skip -- Number of logical records to skip <Default 0>
  load -- Number of logical records to load <Default all>
  errors -- Number of errors to allow <Default 50>
  rows -- Number of rows in conventional path bind array or between direct p
ath data saves <Default: Conventional path 64, Direct path all>
  bindsize -- Size of conventional path bind array in bytes <Default 256000>
  silent -- Suppress messages during run <header,feedback,errors,discards,part
itions>
  direct -- use direct path <Default FALSE>
  parfile -- parameter file: name of file that contains parameter specification
```

```

C:\WINNT\system32\cmd.exe
parallel -- do parallel load (Default FALSE)
file -- File to allocate extents from
skip_unusable_indexes -- disallow/allow unusable indexes or index partitions (Default FALSE)
skip_index_maintenance -- do not maintain indexes, mark affected indexes as unusable (Default FALSE)
readsize -- Size of Read buffer (Default 1048576)
external_table -- use external table for load; NOT_USED, GENERATE_ONLY, EXECUTE (Default NOT_USED)
columnarrays -- Number of rows for direct path column array (Default 5000)
streamsize -- Size of direct path stream buffer in bytes (Default 256000)
multithreading -- use multithreading in direct path
resumable -- enable or disable resumable for current session (Default FALSE)
resumable_name -- text string to help identify resumable statement
resumable_timeout -- wait time (in seconds) for RESUMABLE (Default 7200)
date_cache -- size (in entries) of date conversion cache (Default 1000)

PLEASE NOTE: Command-line parameters may be specified either by position or by keywords. An example of the former case is 'sqlldr scott/tiger foo'; an example of the latter is 'sqlldr control=foo userid=scott/tiger'. One may specify parameters by position before but not after parameters specified by keywords. For example, 'sqlldr scott/tiger control=foo logfile=log' is allowed, but 'sqlldr scott/tiger control=foo log' is not, even though the position of the parameter 'log' is correct.

C:\Documents and Settings\ddischiave>

```

Here is an example of a SQLLDR parameter and control file that could be used to import a State table that contains a State abbreviate and State name.

The following SQL created the State table:

```

CREATE TABLE STATE
(
    stateAbbr          CHAR(2),
    stateName          VARCHAR2(30),
    CONSTRAINT State_PK PRIMARY KEY (stateAbbr)
);

```

Examples of the values in the text file:

```

AL ALABAMA
AK ALASKA
AZ ARIZONA
AR ARKANSAS
CA CALIFORNIA
CO COLORADO
CT CONNECTICUT

```

SQLLDR parameter file: This file tells SQLLDR where to find/create all required files

```

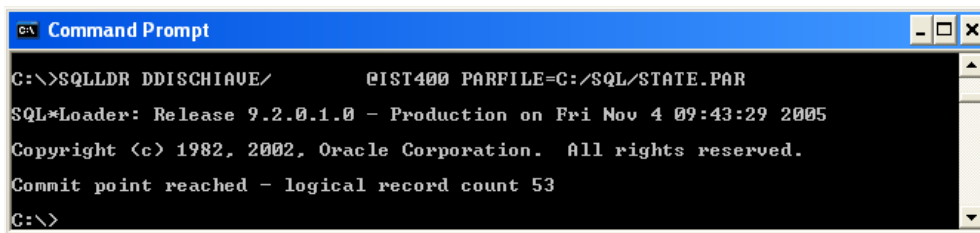
control=c:/SQL/statectl /* control file used by SQLLDR */
log=c:/SQL/LOGstate.txt /* Log file created by SQLLDR */
bad=c:/SQL/BADstate.txt /* file contains any rejected records*/
discard=c:/SQL/DSCstate.txt /* file contains any discrepancies */
SKIP=0

```

SQLLDR control file: This file tells SQLLDR what to do. What table to use, which attributes to populate, where and what type of data exists in the text file.

```
LOAD DATA
INFILE 'C:\SQL\states.txt'
REPLACE
INTO TABLE STATE
  TRAILING NULLCOLS
(
  STATEABBR                POSITION (1:2)                CHAR,
  STATENAME                POSITION (4:34)                CHAR
)
```

Now, if I create the table and execute the SQLLDR program this is the result.



The following is a sample of the results:

```
SQL> SELECT * FROM STATE;

ST STATENAME
-----
AL ALABAMA
AK ALASKA
AZ ARIZONA
AR ARKANSAS
```

The log file contains the following

SQL*Loader: Release 9.2.0.1.0 - Production on Fri Nov 4 09:43:29 2005

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Control File: c:/SQL/state.ctl
Data File: C:\SQL\states.txt
Bad File: c:/SQL/BADstate.txt
Discard File: c:/SQL/DSCstate.txt
(Allow all discards)

Number to load: ALL
Number to skip: 0
Errors allowed: 50
Bind array: 64 rows, maximum of 256000 bytes
Continuation: none specified

Path used: Conventional

Table STATE, loaded from every logical record.
Insert option in effect for this table: REPLACE
TRAILING NULLCOLS option in effect

Column Name	Position	Len	Term	Encl	Datatype
STATEABBR	1:2	2			CHARACTER
STATENAME	4:34	31			CHARACTER

Record 52: Discarded - all columns null.

Record 53: Discarded - all columns null.

Table STATE:
51 Rows successfully loaded.
0 Rows not loaded due to data errors.
0 Rows not loaded because all WHEN clauses were failed.
2 Rows not loaded because all fields were null.

Space allocated for bind array: 2432 bytes(64 rows)

Read buffer bytes: 1048576

Total logical records skipped: 0
Total logical records read: 53
Total logical records rejected: 0
Total logical records discarded: 2

Run began on Fri Nov 04 09:43:29 2005
Run ended on Fri Nov 04 09:43:29 2005

Elapsed time was: 00:00:00.12
CPU time was: 00:00:00.04

Lab Assignment

For each question provide a screenshot of the command or SQL, along with evidence the command worked, such as an OEM screenshot, etc.

1. Create a cold backup of your **Orcl** database and place the backup files in a backup folder named **c:\db_backup**.
2. Create an export of the entire **FUDGEMART** schema. Save it to a file **FUDGEMART.DMP**
3. Create a new user schema called **LAB5** the using SQL Create User command. Make sure the **LAB5** user has unlimited quota on the **FUDGEMART** tablespace, and enough system privileges to import data.
4. Import the **FUDGEMART.DMP** into the new user schema.

Answer these Questions

5. Explain why companies might prefer “hot” backup procedures?
6. What are some of the issues related to “hot” backups?
7. What are some of the issues with ARCHIVELOG mode?
8. Under what circumstances the export (exp) utility could be used as a form of backup?
9. Can IMP and EXP be used to move data from Oracle to another DBMS like SQL Server?
10. How might SQLLDR be used? Describe a “use case” for it.
Hint: think of your answer to question 9.