- **ORACLE UTILITIES**
  - SQL * LOADER
  - EXPORT & IMPORT

- Practical on EXPORT & IMPORT

- **BACKUP & RECOVERY**
  - COLD BACKUP
  - Practical on COLD BACKUP

  - HOT BACKUP
  - Practical on HOT BACKUP

ORACLE UTILITIES

SQL * LOADER

1. It is a utility to load the data from a textfile to oracle table
2. The textfile is called as flat file. It can be in .txt, .csv, etc format
3. SQL *LOADER contains following components
   a. CONTROL FILE – this will define the configuration parameters which tell how to load the data into the table. It has nothing to do with database control file
   b. INPUT FILE or INFILE – this is the file from which data should be loaded
   c. BADFILE – records which are failed to load into table (due to any reason) will be stored in this file
   d. DISCARDFILE – records which doesn't satisfy the condition will be placed here
   e. LOGFILE – it will record the action of sql * loader and can be used for reference
4. SQL * LOADER can be invoked as follows
   ```
   [oracle@server1 admin]$ sqlldr userid=system/oracle control=control.lst log=track.log
   ```

EXPORT & IMPORT

1. It is the utility to transfer the data between two oracle databases
2. The following levels of export/import are possible
   a. Database level
   b. Schema level
   c. Table level
   d. Row level
3. Apart from database level, we can perform other levels of export and import within the same database
4. Whenever session is running long time, we can check from v$session_longops

Note : To avoid questionable statistics warning, use statistics=none during export

5. Export will convert the command to select statements and the final output will be returned to dumpfile
6. Server process will take the responsibility of writing the data to dumpfile.

7. Export will transfer the data to dumpfile in the size of block. To increase the speed of writing we can set BUFFER=10 * avg row length

Note : avg row length can be obtained from dba_tables

8. DIRECT=Y will make the export process faster by performing in the following way

   Exp → select statement → datafiles → dumpfile

9. By mentioning CONSISTENT=Y, export will take data from only undo tablespace if a DML operation is being performed on the table

Note : while using CONSISTENT=Y, there is a chance of getting ORA-1555 error

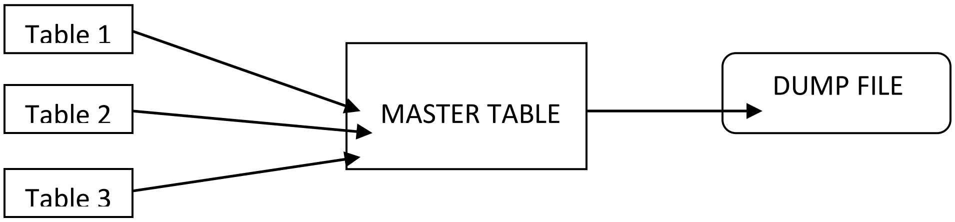
10. Import is the utility to dump the contents from export dumpfile to a schema
11. Import internally converts contents of export dump file to DDL and DML statements
12. SHOW=Y can be used to check corruption in export dump file. This will not actually import the contents
13. IGNORE=Y should be used if already an object exists with the same name. It will append the data if the object exists already

Note : whenever import fails with warning for constraints or grants, do import again with ROWS=N option

14. DBA_DATAPUMP_JOBS view can be used to find the status of datapump export or import process

Note : whenever datapump export is done using PARALLEL option, import also should be done with the same option. Otherwise it will effect the time taking for import

15. Oracle will try to import tables to the tablespace with same name and if tablespace doesn't exist, it will go to users default tablespace
16. Datapump is an extension to traditional exp/imp process
17. During datapump export, oracle will create master table in the corresponding schema and data will be transferred parallely from tables to dumpfile



18. During datapump import this will happen in reverse order i.e from dumpfile a master table will be created and from that original tables
19. After finishing either export or import in datapump, oracle will automatically drops master table
20. Just like exp/imp, datapump also contains 4 (database, schema, table and row) levels

Parameters used in EXPORTS

EXP Option	**Default value**	**Description**
Buffer		Specifies the size, in bytes, of the buffer (array) used to insert the data
Compress	N	When "Y", export will mark the table to be loaded as one extent for the import utility. If "N", the current storage options defined for the table will be used. Although this option is only implemented on import, it can only be specified on export.
Consistent	N	Specifies the set transaction read only statement for export, ensuring data consistency. This option should be set to "Y" if activity is anticipated while the exp command is executing. If 'Y' is set, confirm that there is sufficient undo segment space to avoid the export session getting the ORA-1555 Snapshot too old error.
Constraints	Y	Specifies whether table constraints should be exported with table data.
Direct	N	Determines whether to use direct or conventional path export. Direct path exports bypass the SQL command, thereby enhancing performance.
Feedback	0	Determines how often feedback is displayed. A value of feedback=n displays a dot for every n rows processed. The display shows all tables exported not individual ones.
File		The name of the export file. Multiple files can be listed, separated by commas. When export fills thefilesize, it

		will begin writing to the next file in the list.
Filesize		The maximum file size, specified in bytes.
flashback_scn		The system change number (SCN) that export uses to enable flashback.
flashback_time		Export will discover the SCN that is closest to the specified time. This SCN is used to enable flashback.
Full		The entire database is exported.
Grants	Y	Specifies object grants to export.
Help		Shows command line options for export.
Indexes	Y	Determines whether index definitions are exported. The index data is never exported.
Log		The filename used by export to write messages.
object_consistent	N	Specifies whether export uses SET TRANSACTION READ ONLY to ensure that the data being exported is consistent.
Owner		Only the owner’s objects will be exported.
Parfile		The name of the file that contains the export parameter options. This file can be used instead of specifying all the options on the command line for each export.
Query		Allows a subset of rows from a table to be exported, based on a SQL where clause.
Recordlength		Specifies the length of the file record in bytes. This parameter affects the amount of data that accumulates before it is written to disk. If not specified, this parameter defaults to the value specific to that platform. The highest value is 64KB.
Resumable	N	Enables and disables resumable space allocation. When “Y”, the parameters resumable_name andresumable_timeout are utilized.
resumable_name		User defined string that helps identify a resumable statement that has been suspended. This parameter is ignored unless resumable = Y.
resumable_timeout	2h	The time period in which an export error must be fixed. This parameter is ignored unless resumable = Y.
Rows	Y	Indicates whether or not the table rows should be exported.

Statistics	ESTIMATE	Indicates the level of statistics generated when the data is imported. Other options include COMPUTE and NONE.
Tables		Indicates that the type of export is table-mode and lists the tables to be exported. Table partitions and sub partitions can also be specified.
Tablespaces		Indicates that the type of export is tablespace-mode, in which all tables assigned to the listed tablespaces will be exported. This option requires the EXP_FULL_DATABASE role.
transport_tablespace	N	Enables the export of metadata needed for transportable tablespaces.
Triggers	Y	Indicates whether triggers defined on export tables will also be exported.
tts_full_check	FALSE	When TRUE, export will verify that when creating a transportable tablespace, a consistent set of objects is exported.
Userid		Specifies the userid/password of the user performing the export.
Volsize		Specifies the maximum number of bytes in an export file on each tape volume.

Import utility:

Imports is an oracle software utility which is a platform independent tools uses the dumpfile created by export to import into the database on the object or schema or database level. To import the dumpfile requires the user granted role which is imp_full_database. While importing, the diagnostic information is logged into the log file with .log extension.

Parameters used in IMPORTS

IMP Option	Default value	Description
Buffer		Specifies the size, in bytes, of the buffer (array) used to insert the data
Commit	N	Specifies whether import should commit after each array insert. By default, import commits after each table is loaded, however, this use a lot of the rollback segments or undo space for huge tables.
Compile	Y	Tells import to compile procedural objects when they are imported.
Constraints	Y	Specifies whether table constraints should also be imported with table data.
datafiles (only with transport_tablespace)		This parameter lists data files to be transported to the database.
Destroy	N	Overwrite tablespace data file

Feedback	0	Determines how often feedback is displayed. A value of feedback=100 displays a dot for every 100 rows processed. This option applies to the total tables imported, not individual ones. Another way to measure the number of rows that have been processed is to execute the following query while the import is active:
		SELECT rows_processed
		FROM v$sqlarea
		WHERE sql_text like 'INSERT %INTO "%'
		AND command_type = 2
		AND open_versions > 0;

File		The name of the export file to import. Multiple files can be listed, separated by commas. When export reaches the filesize it will begin writing to the next file in the list.
Filesize		Maximum size of each dump file

Fromuser		A comma delimited list of schemas from which to import. If the export file contains many users or even the entire database, the fromuser option enables only a subset of those objects (and data) to be imported.
Full		The entire export file is imported.
Grants	Y	Specifies to import object grants.
Help		Shows command line options for import imp -help or imp help=y
Ignore	N	Specifies how object creation errors should be handled. If a table already exists and ignore=y, then the rows are imported to the existing tables, otherwise errors will be reported and no rows are loaded into the table.
Indexes	Y	Determines whether indexes are imported.
Indexfile		Specifies a filename that contains index creation statements. This file can be used to build the indexes after the import has completed.
Log		The filename used by import to write messages.
Parfile		The name of the file that contains the import parameter options. This file can be used instead of specifying all the options on the command line.
Recordlength		Specifies the length of the file record in bytes. This parameter is only used when transferring export files between operating systems that use different default values.
Resumable	N	When “Y”, the parameters resumable_name and resumable_timeout are utilized.
resumable_name		User defined string that helps identify a resumable statement that has been suspended. This parameter is ignored unless resumable = Y.
resumable_timeout	2h	The time period in which an error must be fixed. This parameter is ignored unless resumable=Y.
Rows	Y	Indicates whether or not the table rows should be imported.
Show	N	When show=y, the DDL within the export file is displayed.

skip_unusable_indexes	N	Determines whether import skips the building of indexes that are in an unusable state.
Statistics	ALWAYS	Determines the level of optimizer statistics that are generated on import. The options include ALWAYS, NONE, SAFE and RECALCULATE. ALWAYS imports statistics regardless of their validity. NONE does not import or recalculate any optimizer statistics. SAFE will import the statistics if they appear to be valid, otherwise they will be recomputed after import. RECALCULATE always generates new statistics after import.
streams_configuration	Y	Determines whether or not any streams metadata present in the export file will be imported.
streams_instantiation	N	Specifies whether or not to import streams instantiation metadata present in the export file
Tables		Indicates that the type of export is table-mode and lists the tables to be exported. Table partitions and sub partitions can also be specified.
Tablespaces		When transport_tablespace=y, this parameter provides a list of tablespaces.
to_user		Specifies a list of user schemas that will be targets for imports.
transport_tablespace	N	When Y, transportable tablespace metadata will be imported from the export file.
tts_owners		When transport_tablespace=Y, this parameter lists the users who own the data in the transportable tablespace set.
Userid		Specifies the userid/password of the user performing the import.

The common IMP/EXP problems

- **ORA-00001**: Unique constraint ... violated - Perhaps you are importing duplicate rows. Use IGNORE=N to skip tables that already exist (imp will give an error if the object is re-created) or the table could be dropped/ truncated and re-imported if we need to do a table refresh..
- **IMP-00015**: Statement failed ... object already exists... - Use the IGNORE=Y import parameter to ignore these errors, but be careful as you might end up with duplicate rows.
- **ORA-01555**: Snapshot too old - Ask your users to STOP working while you are exporting or use parameter CONSISTENT=NO (However this option could create possible referential problems, because the tables are not exported from one snapshot in time).
- **ORA-01562**: Failed to extend rollback segment - Create bigger rollback segments or set parameter COMMIT=Y (with an appropriate BUFFER parameter) while importing.
- **ORA-01435**:user does not exists- The solution is create user.

$sqlplus '/as sysdba'

>create user raja identified by raja123;

>Grant connect,resource to raja

EXPDP:

```
expdp help=y

Export: Release 10.1.0.2.0 - Production on Tuesday, 23 March, 2004 8:33

Copyright (c) 2003, Oracle.  All rights reserved.


The Data Pump export utility provides a mechanism for transferring data objects
between Oracle databases. The utility is invoked with the following command:

   Example: expdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

You can control how Export runs by entering the 'expdp' command followed
by various parameters. To specify parameters, you use keywords:
  Format:  expdp KEYWORD=value or KEYWORD=(value1,value2,...,valueN)
  Example: expdp scott/tiger DUMPFILE=scott.dmp DIRECTORY=dmpdir SCHEMAS=scott
           or TABLES=(T1:P1,T1:P2), if T1 is partitioned table

USERID must be the first parameter on the command line.

Keyword               Description (Default)
------------------------------------------------------------------------------
ATTACH                Attach to existing job, e.g. ATTACH [=job name].
CONTENT               Specifies data to unload where the valid keywords are:
                      (ALL), DATA_ONLY, and METADATA_ONLY.
DIRECTORY             Directory object to be used for dumpfiles and logfiles.
DUMPFILE              List of destination dump files (expdat.dmp),
                      e.g. DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.
ESTIMATE              Calculate job estimates where the valid keywords are:
                      (BLOCKS) and STATISTICS.
ESTIMATE_ONLY         Calculate job estimates without performing the export.
EXCLUDE               Exclude specific object types, e.g. EXCLUDE=TABLE:EMP.
FILESIZE              Specify the size of each dumpfile in units of bytes.
FLASHBACK_SCN         SCN used to set session snapshot back to.
FLASHBACK_TIME        Time used to get the SCN closest to the specified time.
FULL                  Export entire database (N).
HELP                  Display Help messages (N).
INCLUDE               Include specific object types, e.g. INCLUDE=TABLE_DATA.
JOB_NAME              Name of export job to create.
LOGFILE               Log file name (export.log).
NETWORK_LINK          Name of remote database link to the source system.
NOLOGFILE             Do not write logfile (N).
PARALLEL              Change the number of active workers for current job.
PARFILE               Specify parameter file.
QUERY                 Predicate clause used to export a subset of a table.
SCHEMAS               List of schemas to export (login schema).
STATUS                Frequency (secs) job status is to be monitored where
                      the default (0) will show new status when available.
TABLES                Identifies a list of tables to export - one schema only.
TABLESPACES           Identifies a list of tablespaces to export.
TRANSPORT_FULL_CHECK  Verify storage segments of all tables (N).
TRANSPORT_TABLESPACES List of tablespaces from which metadata will be unloaded.
VERSION               Version of objects to export where valid keywords are:
                      (COMPATIBLE), LATEST, or any valid database version.

The following commands are valid while in interactive mode.
Note: abbreviations are allowed
```

```
Command            Description
------------------------------------------------------------------------------
ADD_FILE           Add dumpfile to dumpfile set.
                   ADD_FILE=dumpfile-name
CONTINUE_CLIENT    Return to logging mode. Job will be re-started if idle.
EXIT_CLIENT        Quit client session and leave job running.
HELP               Summarize interactive commands.
KILL_JOB           Detach and delete job.
PARALLEL           Change the number of active workers for current job.
                   PARALLEL=.
START_JOB          Start/resume current job.
STATUS             Frequency (secs) job status is to be monitored where
                   the default (0) will show new status when available.
                   STATUS=[interval]
STOP_JOB           Orderly shutdown of job execution and exits the client.
                   STOP_JOB=IMMEDIATE performs an immediate shutdown of the
                   Data Pump job.
```

Oracle 10g Release 2 (10.2) added the following parameters.

```
Keyword            Description (Default)
------------------------------------------------------------------------------
COMPRESSION        Reduce size of dumpfile contents where valid
                   keyword values are: (METADATA_ONLY) and NONE.
ENCRYPTION_PASSWORD   Password key for creating encrypted column data.
SAMPLE             Percentage of data to be exported;

The following commands are valid while in interactive mode.
Note: abbreviations are allowed

Command            Description
------------------------------------------------------------------------------
FILESIZE           Default filesize (bytes) for subsequent ADD_FILE commands.
```

Oracle 11g Release 1 (11.1) added the following parameters.

```
Keyword             Description (Default)
------------------------------------------------------------------------------
DATA_OPTIONS         Data layer flags where the only valid value is:
                XML_CLOBS-write XML datatype in CLOB format
ENCRYPTION          Encrypt part or all of the dump file where valid keyword
                values are: ALL, DATA_ONLY, METADATA_ONLY,
                ENCRYPTED_COLUMNS_ONLY, or NONE.
ENCRYPTION_ALGORITHM  Specify how encryption should be done where valid
                keyword values are: (AES128), AES192, and AES256.
ENCRYPTION_MODE      Method of generating encryption key where valid keyword
                values are: DUAL, PASSWORD, and (TRANSPARENT).
REMAP_DATA          Specify a data conversion function,
                e.g. REMAP_DATA=EMP.EMPNO:REMAPPKG.EMPNO.
REUSE_DUMPFILES      Overwrite destination dump file if it exists (N).
TRANSPORTABLE        Specify whether transportable method can be used where
                valid keyword values are: ALWAYS, (NEVER).

The following commands are valid while in interactive mode.
Note: abbreviations are allowed

Command             Description
------------------------------------------------------------------------------
REUSE_DUMPFILES      Overwrite destination dump file if it exists (N).
```

Oracle 11g Release 2 (11.2) altered the format of the help output as well as adding the following parameters.

```
CLUSTER
Utilize cluster resources and distribute workers across the Oracle RAC.
Valid keyword values are: [Y] and N.

SERVICE_NAME
Name of an active Service and associated resource group to constrain Oracle RAC resources.

SOURCE_EDITION
Edition to be used for extracting metadata.
```

IMPDP:

```
impdp help=y

Import: Release 10.1.0.2.0 - Production on Saturday, 11 September, 2004 17:22

Copyright (c) 2003, Oracle.  All rights reserved.

The Data Pump Import utility provides a mechanism for transferring data objects
between Oracle databases. The utility is invoked with the following command:

     Example: impdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

You can control how Import runs by entering the 'impdp' command followed
by various parameters. To specify parameters, you use keywords:

     Format:  impdp KEYWORD=value or KEYWORD=(value1,value2,...,valueN)
     Example: impdp scott/tiger DIRECTORY=dmpdir DUMPFILE=scott.dmp

USERID must be the first parameter on the command line.

Keyword               Description (Default)
------------------------------------------------------------------------------
ATTACH                Attach to existing job, e.g. ATTACH [=job name].
CONTENT               Specifies data to load where the valid keywords are:
                      (ALL), DATA_ONLY, and METADATA_ONLY.
DIRECTORY             Directory object to be used for dump, log, and sql files.
DUMPFILE              List of dumpfiles to import from (expdat.dmp),
                      e.g. DUMPFILE=scott1.dmp, scott2.dmp, dmpdir:scott3.dmp.
ESTIMATE              Calculate job estimates where the valid keywords are:
                      (BLOCKS) and STATISTICS.
EXCLUDE               Exclude specific object types, e.g. EXCLUDE=TABLE:EMP.
FLASHBACK_SCN         SCN used to set session snapshot back to.
FLASHBACK_TIME        Time used to get the SCN closest to the specified time.
FULL                  Import everything from source (Y).
HELP                  Display help messages (N).
INCLUDE               Include specific object types, e.g. INCLUDE=TABLE_DATA.
JOB_NAME              Name of import job to create.
LOGFILE               Log file name (import.log).
NETWORK_LINK          Name of remote database link to the source system.
NOLOGFILE             Do not write logfile.
```

```
PARALLEL            Change the number of active workers for current job.
PARFILE             Specify parameter file.
QUERY               Predicate clause used to import a subset of a table.
REMAP_DATAFILE       Redefine datafile references in all DDL statements.
REMAP_SCHEMA         Objects from one schema are loaded into another schema.
REMAP_TABLESPACE      Tablespace object are remapped to another tablespace.
REUSE_DATAFILES      Tablespace will be initialized if it already exists (N).
SCHEMAS              List of schemas to import.
SKIP_UNUSABLE_INDEXES Skip indexes that were set to the Index Unusable state.
SQLFILE             Write all the SQL DDL to a specified file.
STATUS              Frequency (secs) job status is to be monitored where
                the default (0) will show new status when available.
STREAMS_CONFIGURATION Enable the loading of Streams metadata
TABLE_EXISTS_ACTION   Action to take if imported object already exists.
                Valid keywords: (SKIP), APPEND, REPLACE and TRUNCATE.
TABLES              Identifies a list of tables to import.
TABLESPACES          Identifies a list of tablespaces to import.
TRANSFORM            Metadata transform to apply (Y/N) to specific objects.
                Valid transform keywords: SEGMENT_ATTRIBUTES and STORAGE.
                ex. TRANSFORM=SEGMENT_ATTRIBUTES:N:TABLE.
TRANSPORT_DATAFILES   List of datafiles to be imported by transportable mode.
TRANSPORT_FULL_CHECK  Verify storage segments of all tables (N).
TRANSPORT_TABLESPACES List of tablespaces from which metadata will be loaded.
                Only valid in NETWORK_LINK mode import operations.
VERSION             Version of objects to export where valid keywords are:
                (COMPATIBLE), LATEST, or any valid database version.
                Only valid for NETWORK_LINK and SQLFILE.

The following commands are valid while in interactive mode.
Note: abbreviations are allowed

Command             Description (Default)11g
------------------------------------------------------------------------------
CONTINUE_CLIENT      Return to logging mode. Job will be re-started if idle.
EXIT_CLIENT          Quit client session and leave job running.
HELP                Summarize interactive commands.
KILL_JOB            Detach and delete job.
PARALLEL            Change the number of active workers for current job.
                PARALLEL=.
START_JOB            Start/resume current job.
                START_JOB=SKIP_CURRENT will start the job after skipping
                any action which was in progress when job was stopped.
STATUS              Frequency (secs) job status is to be monitored where
```

```
                 the default (0) will show new status when available.
                 STATUS=[interval]
STOP_JOB                 Orderly shutdown of job execution and exits the client.
                 STOP_JOB=IMMEDIATE performs an immediate shutdown of the
                 Data Pump job.
```

Oracle 10g Release 2 (10.2) added the following parameter.

```
Keyword              Description (Default)
------------------------------------------------------------------------------
ENCRYPTION_PASSWORD   Password key for accessing encrypted column data.
                 This parameter is not valid for network import jobs.
```

Oracle 11g Release 1 (11.1) added the following parameters.

```
Keyword              Description (Default)
------------------------------------------------------------------------------
DATA_OPTIONS          Data layer flags where the only valid value is:
                 SKIP_CONSTRAINT_ERRORS-constraint errors are not fatal.
PARTITION_OPTIONS     Specify how partitions should be transformed where the
                 valid keywords are: DEPARTITION, MERGE and (NONE)
REMAP_DATA            Specify a data conversion function,
                 e.g. REMAP_DATA=EMP.EMPNO:REMAPPKG.EMPNO
```

Oracle 11g Release 2 (11.2) altered the format of the help output as well as adding the following parameters.

```
CLUSTER
Utilize cluster resources and distribute workers across the Oracle RAC.
Valid keyword values are: [Y] and N.

SERVICE_NAME
Name of an active Service and associated resource group to constrain Oracle RAC resources.

SOURCE_EDITION
Edition to be used for extracting metadata.

TARGET_EDITION
Edition to be used for loading metadata.
```

Demo

Utility: Datapump is available from Oracle 10g

- Expdp (export datapump)
- Impdp (import datapump)

COMMANDS

# To create a directory

```
$mkdir -p /u01/expbkp'
```

```
SQL> create directory dpbkp as '/u01/expbkp';

Directory created.
```

# To grant permissions on directory

```
SQL> grant read,write on directory dpbkp to scott;

Grant succeeded.
```

# To view directory information

```
SQL> select * from dba_directories;

OWNER                          DIRECTORY_NAME
------------------------------ ------------------------------
DIRECTORY_PATH

SYS                            DPBKP
/u01/expbkp
```

# To know options of export/import

```
[oracle@server1 ~]$ exp help=y
```

```
[oracle@server1 ~]$ imp help=y
```

```
[oracle@server1 ~]$ expdp help=y
```

```
[oracle@server1 ~]$ impdp help=y
```

# To take database level export (using datapump also)

```
[oracle@server1 ~]$ expdp directory=dpbkp dumpfile=fullprod.dmp logfile=fullprod.log full=y
```

```
[oracle@server1 ~]$ exp file=/u01/fullbkp_prod.dmp log=/u01/fullbkp_prod.log full=y
```

# To take schema level export (using datapump also)

```
[oracle@server1 ~]$ expdp directory=dpbkp dumpfile=scott_bkp.dmp logfile=scott_bkp.log schemas='SCOTT'
```

```
[oracle@server1 ~]$ exp file=/u01/scott_bkp.dmp log=/u01/scott_bkp.log owner='SCOTT'
```

# To take table level export (using datapump also)

```
[oracle@server1 ~]$ expdp directory=dpbkp dumpfile=emp_bkp.dmp logfile=emp_bkp.log tables='SCOTT.EMP'
```

```
[oracle@server1 ~]$ exp file=/u01/emp_bkp.dmp log=/u01/emp_bkp.log tables='SCOTT.EMP'
```

# To take row level export (using datapump also)

```
[oracle@server1 ~]$ expdp directory=dpbkp dumpfile=emprows_bkp.dmp logfile=emprows_bkp.log tables='SCOTT.EMP' query=\"where deptno=10\"
```

```
[oracle@server1 ~]$ exp file=/u01/emp_rows_bkp.dmp log=/u01/emp_rows.log tables='SCOTT.EMP' query=\"where deptno=10\"
```

# To import full database

```
[oracle@server1 ~]$ impdp directory=dpbkp dumpfile=fullprod.dmp logfile=imp_fullprod.log full=y
```

```
[oracle@server1 ~]$ imp file=/u01/fullprod.dmp log=/u01/imp_fullprod.log full=y
```

# To import a schema

```
[oracle@server1 ~]$ impdp directory=dpbkp dumpfile=scott_bkp.dmp logfile=imp_schema.log remap_schema='SCOTT:SCOTT'
```

```
[oracle@server1 ~]$ imp file=/u01/scott_bkp.dmp log=/u01/imp_schema.log fromuser='SCOTT' touser='SCOTT'
```

# To import a table

```
[oracle@server1 ~]$ impdp directory=dpbkp dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP' remap_schema='SCOTT:SCOTT'
```

```
[oracle@server1 ~]$ imp file=/u01/emp_bkp.dmp log=/u01/imp_emp.log fromuser='SCOTT' touser='SCOTT' tables='EMP'
```

# To import a table to another user

```
[oracle@server1 ~]$ impdp directory=dpbkp dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP' remap_schema='SCOTT:SYSTEM'
```

```
[oracle@server1 ~]$ imp file=/u01/emp_bkp.dmp log=/u01/imp_emp.log fromuser='SCOTT' touser='SYSTEM' tables='EMP'
```

# To import tables to another tablespace (only in datapump)

```
[oracle@server1 ~]$ impdp directory=dpbkp dumpfile=emp_bkp.dmp logfile=imp_emp.log tables='EMP' remap_schema='SCOTT:SCOTT' remap_tablespace=’MYDATA:MYTBS’
```

BACKUP & RECOVERY

BACKUP's

There are two types of physical backups

1. Cold Backup
2. Hot Backup

COLD BACKUP

1. Backup is a copy of original data which will be used to recover databases
2. If the data is reproducable and backup not existing, still we can recover the data. But it is a tedious and time consuming task
3. Taking backup after shutting down the database is called cold backup and because no transactions exist, the backup will be consistent
4. In real time, we will perform cold backup very rarely

STEPS to take cold backup

```
SQL> select name from v$datafile;

SQL> select member from v$logfile;

SQL> select name from v$controlfile;

SQL> shutdown immediate

[oracle@server1 ~]$ mkdir /u03/coldbkp

[oracle@server1 ~]$ cp /datafiles/prod/*.dbf /u03/coldbkp

 [oracle@server1 ~]$ cp /datafiles/prod/*.log /u03/coldbkp

 [oracle@server1 ~]$ cp /datafiles/prod/*.ctl /u03/coldbkp

[oracle@server1 ~]$ sqlplus "/ as sysdba"

SQL> startup

SQL> alter database backup controlfile to trace;
```

HOT BACKUP

1. Taking the backup while the database is up and running is called hot backup
2. During hot backup database will be in fuzzy state and still users can perform transactions which makes backup inconsistent
3. Whenever we place a tablespace or database in begin backup mode, following happens
    a. The corresponding datafiles header will be freezed i.e CKPT process will not update latest SCN
    b. Body of the datafile is still active i.e DBWRn will write the dirty blocks to datafiles
4. After end backup, datafile header will be unfreezed and CKPT process will update latest SCN immediately by taking that information from controlfiles
5. During hot backup, we will observe much redo generated because oracle will copy entire data block as redo entry for the first modification of the table. This is to avoid fractured block
6. An oracle data block is a group of OS blocks, depends on the version and bit version of OS, the blocksize of OS will vary
7. The frequency at which oracle will copy the data will be different from OS block copy
8. Database should be in archivelog mode to perform hot backup

<u>STEPS to take hot backup in 9i</u>

```
[oracle@server1 ~]$ mkdir /u03/hotbkp

SQL> select name from v$datafile;

SQL> select name from v$controlfile;

SQL> alter tablespace system begin backup;

SQL> !cp /datafiles/prod/system01.dbf /u03/hotbkp

SQL> alter tablespace system end backup;
```

Repeat above steps for all the tablespaces in the database

```
SQL> !cp /datafiles/prod/*.ctl.dbf /u03/hotbkp

SQL> alter system switch logfile;

SQL> !cp /u03/archives/*.arc /u03/hotbkp/archbkp
```

Taking archive backup is the important step in hot backup

```
SQL> alter database backup controlfile to trace;
```

STEPS to take hot backup in 10g and 11g

```
[oracle@server1 ~]$ mkdir /u03/hotbkp

SQL> select name from v$datafile;

SQL> select name from v$controlfile;

SQL> alter database begin backup;

SQL> !cp /datafiles/prod/*.dbf /u03/hotbkp

SQL> alter database end backup;
```

Since we are placing entire database into begin backup mode, no repetition for all the tablespaces is required

```
SQL> !cp /datafiles/prod/*.ctl.dbf /u03/hotbkp

SQL> alter system switch logfile;

SQL> !cp /u03/archives/*.arc /u03/hotbkp/archbkp
```

Taking archive backup is the important step in hot backup

```
SQL> alter database backup controlfile to trace;
```

Note: In any version, during hot backup we will not take redolog files backup

DATABASE RECOVERY

1. Recover is of 2 types
    a. Complete recovery – recovering database till the point of failure. No data loss
    b. Incomplete recovery – recovering to a certain time or scn. Has data loss
2. We will perform complete recovery if we lost only datafiles
3. We will perform incomplete recovery if we lost either redolog files, controlfiles or archivelog files
4. Recovery process involves two phases
    a. RESTORE – copying a file from backup location to original location as that file is lost now
    b. RECOVER – applying archivelogs and redologs to bring the file SCN in par with latest SCN

Note: Practically, we can do complete recovery even if we lost controlfiles

STEPS for recovering tablespace

```
SQL> alter tablespace mydata offline;

SQL> !cp /u03/hotbkp/mydata01.dbf /datafiles/prod

SQL> recover tablespace mydata;

SQL> alter tablespace mydata online;
```

STEPS for recovering a single datafile

```
SQL> alter database datafile '/datafiles/prod/mydata01.dbf' offline;

SQL> !cp /u03/hotbkp/mydata01.dbf /datafiles/prod

SQL> recover 'datafile /datafiles/prod/mydata01.dbf';

SQL> alter database datafile '/datafiles/prod/mydata01.dbf' online;
```

STEPS for recovering system tablespace

```
SQL> shut immediate

SQL> !cp /u03/hotbkp/system01.dbf /datafiles/prod

SQL> startup mount

SQL> recover tablespace system;

SQL> alter database open;
```

STEPS for recovering database (we will perform this when we lost more than 50% of datafiles)

```
SQL> shut immediate

SQL> !cp /u03/hotbkp/*.dbf /datafiles/prod

SQL> startup mount

SQL> recover database;

SQL> alter database open;
```

Note: we can drop a single datafile using below command

```
SQL> alter database datafile '/datafiles/prod/mydata01.dbf' offline drop;
```

When we use above command, it will delete the file at OS level, but data dictionary will not be updated and never we can get back that file even if we have backup. So don't use this in real time

STEPS to recover datafile in a noarchivelog mode database

```
SQL> shutdown immediate

SQL> !cp /u03/coldbkp/*.dbf /datafiles/prod

SQL> !cp /u03/coldbkp/*.ctl /datafiles/prod

SQL> !cp /u03/coldbkp/*.log /datafiles/prod

SQL>startup
```

STEPS to recover redologfile in a noarchivelog mode database

```
SQL> shutdown immediate

SQL> !cp /u03/coldbkp/*.dbf /datafiles/prod

SQL> !cp /u03/coldbkp/*.ctl /datafiles/prod

SQL> recover database until cancel;

SQL> alter database open resetlogs;
```

STEPS to recover controlfile in a noarchivelog mode database

```
SQL> shutdown immediate

SQL> !cp /u03/coldbkp/*.ctl /datafiles/prod

SQL> startup mount

SQL> recover database using backup controlfile until cancel;

SQL> alter database open resetlogs;
```

STEPS to recover redolog file in archivelog mode

```
SQL> shutdown immediate

SQL> startup mount

SQL> recover database until scn 12345 / until time ‘2011-01-05 11:00:00’;

SQL> alter database open resetlogs;
```

Using RESETLOGS – When used resetlogs option to open the database, it will

1. Create new redolog files at OS level (location and size will be taken from controlfile) if not already existing
2. Resets the log seq number (LSN) to 1, 2, 3 etc for the created files

STEPS to recover a datafile without backup

```
SQL> alter tablespace mydata offline;
```

```
SQL> alter database create datafile ‘/datafiles/prod/mydata01.dbf’ as
‘/datafiles/prod/mydata01.dbf’;
```

```
SQL> recover tablespace mydata;
```

```
SQL> alter tablespace mydata online;
```

Note: All the archives generated from the date of datafile creation should be available to do this

STEPS for controlfile complete recovery

```
SQL> alter database backup controlfile to trace;
```

The above command may not work sometimes, in which case we need to use already taken trace file during backup. This command will generate a controlfile script in udump in the form of trace file

```
SQL> shutdown immediate
```

```
[oracle@server1 ~]$ goto udump location and copy the first create controlfile script to a file
called control.sql
```

```
SQL> startup nomount
```

```
SQL> @control.sql
```

```
SQL> alter database open;
```

Note: After creating control files using above procedure, there will be no SCN in that. So server process will write the latest SCN to control files in this situation by taking info from datafile header