

PROOF OF TECHNOLOGY

DB2 ADMINISTRATION FOR ORACLE DBA'S

BACKUP / RESTORE / RECOVERY

AND

CONFIGURE DATABASE LOGGING

DOCUMENT REVISION HISTORY

Version Number	Revision Description	By Whom	When
1.0	Backup / Restore / Recovery Configure Database Logging	Vikram Khatri vkhatr@us.ibm.com	10/16/2004

TABLE OF CONTENTS

Introduction.....	5
Training	5
Architecture	5
Oracle Architecture.....	5
Data files and Data blocks	5
Redo Logs	6
Undo Segments.....	6
Control Files.....	6
DB2 Architecture	6
Recovery Logs.....	8
Recovery History Files.....	8
Table space Change History File.....	8
Offline Backup.....	8
History File	9
Oracle	9
DB2	9
Restore History File	10
Offline Restore	10
Oracle	10
DB2	10
Configure Database Logging	12
Oracle	12
DB2	12
LOGRETAIN.....	12
LOGFILSZ	12
LOGBUFSZ.....	12
NEWLOGPATH.....	13
MIRRORLOGPATH	13
LOGPRIMARY	13
LOGSECOND	13
BLK_LOG_DSK_FUL.....	13
OVERFLOWLOGPATH	13
Update Logging Parameters.....	13
Online Backup.....	14
Oracle	14
DB2	15
Active Logs	15
Online Archive Logs.....	15
Offline Archive Logs.....	15
Online Tablespace Backup.....	16
Comparison of Online Backup for Oracle and DB2	16
Online Restore	17
Oracle	17
Control and SPFILE are intact but all data files are lost	17
Restore and Recovery of Individual Tablespaces.....	17
Restore the control file from backup	17
Restore the Server Parameter File (SPFILE) from backup.....	18
Restore / Recover Data Files and Tablespaces to new Location	18
Perform Media Recovery of a Database, Tablespace or Data file	18
DB2	18

Incremental Backup and Restore.....	19
Oracle.....	19
DB2.....	19
Restoring from Incremental Backup.....	20
Recover Dropped Table.....	21
Oracle.....	21
DB2.....	21
Conclusion.....	23

INTRODUCTION

Backup and Recovery are essential part of database administration regardless of database. Backup and Restore are not conspicuous until there is a serious loss of data. In today's world, when hardware reliability is extremely good with hot swapping of failed disks etc, the restore or recovery do not get to practice except for specific purposes.

The role of backup and restore is now even more important from high availability standpoint. We will discuss this specifically in High Availability and Disaster Recovery section separately.

This Proof of Technology session on Backup and Restore provides a detailed comparison between Oracle and DB2.

TRAINING

If we compare backup and recovery in Oracle with DB2, it is complicated in Oracle. Oracle offers a three day course in [backup and recovery](#). DB2 backup and restore is also complicated but it is hidden from end users. The IBM offers a fast paced four days training course for relational database administrators in all major aspects of [DB2 administration](#) including backup and restore.

ARCHITECTURE

The underlying architecture of Backup and Recovery differ greatly in Oracle and DB2 but principles are very much similar.

Oracle Architecture

Oracle uses Recovery manager (RMAN) tool to perform range of backup and recovery activities. In another solution, the traditional user managed backup and recovery methods using operating system utilities are used. This requires detailed knowledge of different Oracle component to back them up appropriately. The cold backup in Oracle is done using traditional methods.

The physical database that makes up an Oracle database is as follows:

Data files and Data blocks

An Oracle database consists of one or more logical storage units called tablespaces. Each tablespace in an Oracle database consists of one or more files called datafiles, physical files under the host operating system in which the database is running.

A database's data is collectively stored in the datafiles that constitute each tablespace of the database. The simplest Oracle database would have one tablespace, stored in one datafile. The database manages the storage space in the datafiles of a database in units called data blocks. A data block is the smallest unit of data used by a database. Data blocks are the smallest units of storage that the database can use or allocate.

Redo Logs

Redo logs record all changes made to database's data files. Any time a data is changed in database, a change is recorded in online redo logs first, before it is applied to data files. An Oracle database requires atleast two online redo log groups, and in each group there is atleast one online redo log member. The database rotates through the online redo log groups, storing changes in current online redo logs while the group not in use can be archived based upon setting of ARCHIVELOG and NOARCHIVELOG mode. In NOARCHIVELOG mode, the online redo logs are overwritten.

Undo Segments

When a data in a datafile is updated, "before image" of that data are written into uodo segments. If a transaction is rolled back, this undo information can be used to restore the original datafile contents. From recovery prospective, the undo segments are required to rollback effects of uncommitted transactions.

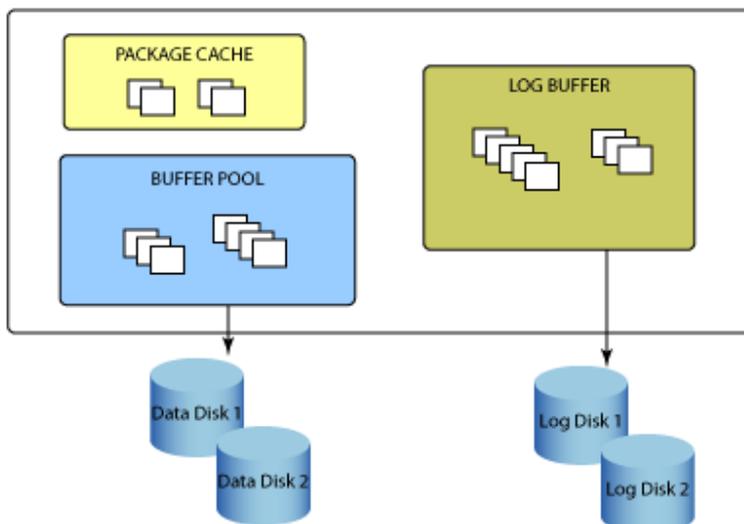
Control Files

The control file contains a crucial record of the physical structures of the database and their status. Several types of information stored in the control file are related to backup and recovery:

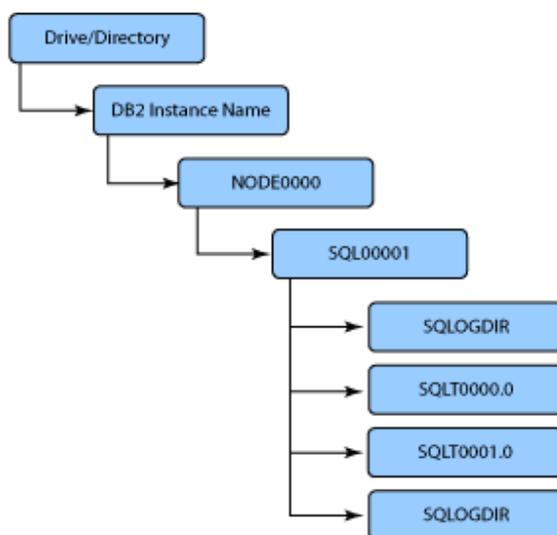
- Database information (RESETLOGS SCN and time stamp)
- Tablespace and datafile records (filenames, datafile checkpoints, read/write status, offline ranges)
- Information about redo threads (current online redo log)
- Log records (log sequence numbers, SCN range in each log)
- A record of past RMAN backups
- Information about corrupt datafile blocks

DB2 Architecture

DB2 has buffer pool and log buffer, which are used to store changes in memory before they are flushed to disks. The following picture explains this.



The DB2 database structure is given as below:



DB2 provides BACKUP, RESTORE and ROLLFORWARD commands that can do cold backup, online backup, restore and rolling forward. The complete backup of database using traditional methods can be done by shutting down the database also known as cold backup. But DB2 backup command can also do cold as well as online backup of database.

There are no external files that need to be backed up like control or parameters files in Oracle. The DB2 backup takes care of all internal files that are required for a recovery.

The physical database in DB2 consists of following from backup and recovery perspective.

Recovery Logs

Each DB2 database includes *recovery logs*, which are used to recover from application or system errors. In combination with database backups, they are used to recover the consistency of the database right upto the point in time when the error occurred.

Recovery History Files

The *recovery history files* contains a summary of the backup information that can be used to determine recovery options, if all or part of the database must be recovered to a given point in time. It is used to track recovery-related events such as backup and restore operations, among others.

Table space Change History File

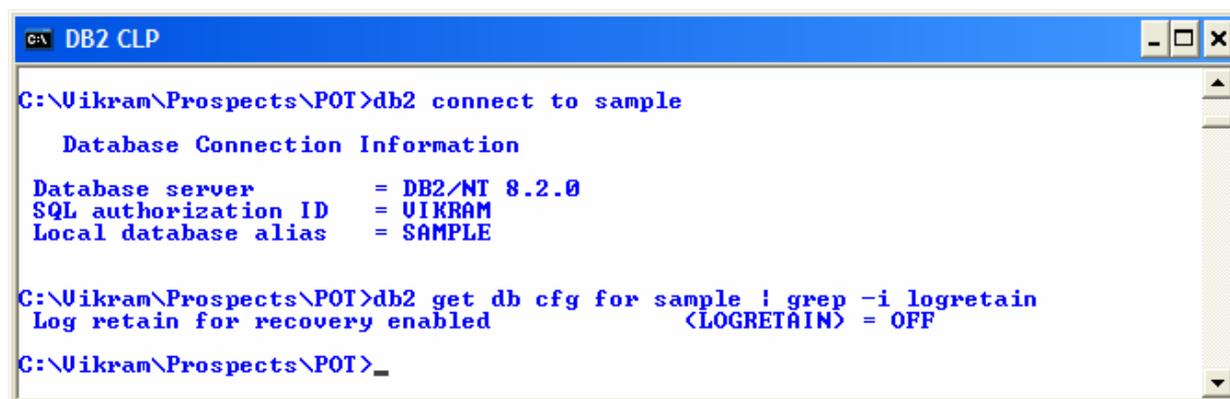
The table space change history file contains information that can be used to determine which log files are required for the recovery of a particular table space.

OFFLINE BACKUP

The offline backup requires that all applications disconnect from database, whereas online backup allows transactions to carry on while backup is in progress.

An offline backup in Oracle is normally done using NOARCHIVELOG option. Using this option, the changes made after the backup are not recoverable if there is a media failure. In Oracle, to perform an offline backup, the database is shutdown in normal mode. Copy all data files, control files, redo logs and undo logs to backup media.

The offline backup is the simplest form of backup in DB2. The offline backup is governed by LOGRETAIN database parameters. By default, this is off when you create a database for the first time. This means that circular logging is in effect in which log files are overwritten and not archived. This is equivalent to NOARCHIVELOG option of Oracle.



```

C:\> DB2 CLP
C:\Uikram\Prospects\POT>db2 connect to sample

Database Connection Information

Database server      = DB2/NT 8.2.0
SQL authorization ID = UIKRAM
Local database alias = SAMPLE

C:\Uikram\Prospects\POT>db2 get db cfg for sample | grep -i logretain
Log retain for recovery enabled          <LOGRETAIN> = OFF

C:\Uikram\Prospects\POT>_

```

The offline backup in DB2 can be done using command.

```
C:\>db2 backup database sample to c:\db2backup
```

Backup successful. The timestamp for this backup image is : 20040901142723

The backup name in Windows and UNIX is slightly different. The backup filename in windows will look like:

SAMPLE.0\DB2\NODE0000\CATN0000\20040901\142723.001

In UNIX, the backup file will be created as:

SAMPLE.0.DB2.NODE0000.CATN0000.20040901.142723.001

HISTORY FILE

Oracle

Oracle RMAN allows you to list all the backups taken by using LIST BACKUP command through RMAN. The V\$BACKUP_FILES also contain information for backups.

DB2

DB2 allows you to query history file to know about history ob backup taken.

```
C:\>db2 list history backup all for sample
```

The output of this command will look something like as we took two backups for SAMPLE database.

```

List History File for sample
Number of matching file entries = 2
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20040901142723001 F D S0000000.LOG S0000000.LOG
-----
Contains 2 tablespace(s):
00001 SYSCATSPACE
00002 USERSPACE1
-----
Comment: DB2 BACKUP SAMPLE OFFLINE
Start Time: 20040901142723
End Time: 20040901142750
Status: A
-----
EID: 1 Location: c:\db2backup\SAMPLE.0\DB2\NODE0000\CATN0000\20040901
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
B D 20040901144022001 F D S0000000.LOG S0000000.LOG
-----
Contains 2 tablespace(s):
00001 SYSCATSPACE
00002 USERSPACE1

```

```
-----
Comment: DB2 BACKUP SAMPLE OFFLINE
Start Time: 20040901144022
End Time: 20040901144044
Status: A
-----
```

```
EID: 2 Location: c:\db2backup\SAMPLE.0\DB2\NODE0000\CATN0000\20040901
```

The LIST HISTORY command shows two backup. We can restore from the backup using TIMESTAMP of the backup. You can see the timestamp from the LIST HISTORY command output. All applications need to disconnect to restore from an offline backup.

Restore History File

The history file is always backed up with the database or tablespace backup. You can recover history file from the backup. There might be instances when one can purge the history by using CLP PRUNE HISTORY command. You can recover the history file from database or tablespace backup by using following command:

```
C:\>db2 restore database sample history file from c:\db2backup taken at
20040901230402
DB20000I The RESTORE DATABASE command completed successfully.
```

OFFLINE RESTORE

Oracle

The offline backup can be restored as offline only. In Oracle, copy back the all files in its right location and start the database. In cases, when you need to re-locate the data files, you can update the new location with ALTER command.

```
ALTER DATABASE RENAME filename <oldpath> to <newpath>
```

DB2

In DB2, each backup will create a timestamp. The restore is done using the timestamp of the backup. You can see the backup history with the command as:

The command to restore the offline backup is:

```
CONNECT TO SAMPLE;
QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS;
CONNECT RESET;
RESTORE DATABASE SAMPLE FROM "c:\db2backup" TAKEN AT 20040901144022 WITH 2
BUFFERS BUFFER 1024 PARALLELISM 1 WITHOUT PROMPTING;
```

If there is a media failure and you want to move the database from one physical path to another, this could be done using RESTORE-REDIRECT command.

The command to move the SAMPLE database backup from current location C:\DB2\NODE0000\SQL00001 to a new path C:\NEWDB2\NODE0000\SQL00001 will be as follows:

```

CONNECT TO SAMPLE;
QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS;
CONNECT RESET;

RESTORE DATABASE SAMPLE FROM "c:\db2backup" TAKEN AT 20040901144022 WITH 2
BUFFERS BUFFER 1024 REPLACE HISTORY FILE REDIRECT PARALLELISM 1 WITHOUT
PROMPTING;

SET TABLESPACE CONTAINERS FOR 0 IGNORE ROLLFORWARD CONTAINER
OPERATIONS USING (PATH "C:\DB2\NODE0000\SQL00001\SQLT0000.0",
PATH "C:\NEWDB2\NODE0000\SQL00001\SQLT0000.0")

SET TABLESPACE CONTAINERS FOR 1 IGNORE ROLLFORWARD CONTAINER
OPERATIONS USING (PATH "C:\DB2\NODE0000\SQL00001\SQLT0001.0",
PATH "C:\NEWDB2\NODE0000\SQL00001\SQLT0001.0")

SET TABLESPACE CONTAINERS FOR 2 IGNORE ROLLFORWARD CONTAINER
OPERATIONS USING (PATH "C:\DB2\NODE0000\SQL00001\SQLT0002.0",
PATH "C:\NEWDB2\NODE0000\SQL00001\SQLT0002.0")

RESTORE DATABASE SAMPLE CONTINUE;

CONNECT TO SAMPLE;
UNQUIESCE DATABASE;
CONNECT RESET;

```

The major difference between Oracle and DB2 offline backup and restore are:

In Oracle, the offline backup creation and proper labeling is DBA's duty to make sure that a wrong copy of backup is not restored. The restore can not be done to another database. The database files can be renamed and relocated using ALTER DATABASE command.

In DB2, the history of backup is maintained and one can choose from a desired timestamp for restoring a database. The tablespace containers can be redefined using RESTORE-REDIRECT, SET TABLESPACE and RESTORE-CONTINUE commands.

DB2 allows you to restore a database to another database, which is not possible in Oracle. For example, you can restore database SAMPLE to SAMPLE2 database using following command.

```

C:\>db2 RESTORE DATABASE SAMPLE FROM "c:\db2backup" TAKEN AT
20040901153436 INTO SAMPLE2

```

The DB2 BACKUP command does not backup external shared libraries used in Stored Procedures or User Defined Functions (UDF). These are generally stored under FUNCTION directory under SQLLIB and these needs to be copied to new location manually.

CONFIGURE DATABASE LOGGING

Oracle

In Oracle, online backup requires that the database operate in ARCHIVELOG mode. This is similar to DB2 LOGRETAIN parameter. To switch to an online with auto archive from the default offline backup in Oracle, you need to do the following:

Fill in the parameters LOG_ARCHIVE_START, LOG_ARCHIVE_DEST, LOG_ARCHIVE_FORMAT and LOG_ARCHIVE_DUPLEX_DEST, LOG_ARCHIVE_DEST_N in the init.ora with the appropriate values

```
SQLPLUS> connect sys as sysdba
SQLPLUS> startup mount exclusive;
SQLPLUS> alter database archivelog;
SQLPLUS> archive log start;
SQLPLUS> alter database open;
```

DB2

The configuration of database logging is an important step in enabling ONLINE backup of databases or tablespaces.

There are few database parameters, which need to be configured for ONLINE backup. How each one of the parameter affects ONLINE backup or HIGH AVAILABILITY is explained below:

LOGRETAIN

If *logretain* is set to `RECOVERY`, archived logs are kept in the database log path directory, and the database is considered to be recoverable, meaning that roll-forward recovery is enabled.

LOGFILSZ

This parameter specifies the size of each configured log, in number of 4-KB pages. The size of the log file has a direct bearing on performance. There is a performance cost for switching from one log to another. The larger log file size is better from a pure performance perspective. This parameter also indicates the log file size for archiving. In this case, a larger log file is size it not necessarily better, since a larger log file size may increase the chance of failure or cause a delay in log shipping scenarios. When considering active log space, it may be better to have a larger number of smaller log files.

LOGBUFSZ

This parameter allows you to specify the amount of memory to use as a buffer for log records before writing these records to disk. The log records are written to disk when either transaction commits or log buffer becomes full. Increasing the log buffer size results in more efficient input/output (I/O) activity associated with logging, because the log records are written to disk less frequently, and more records are written each time.

NEWLOGPATH

The database logs are initially created in SQLOGDIR, which is a subdirectory of the database directory. You can change the location in which active logs and future archived logs are placed by changing the value of this configuration parameter to point to a different directory or to a device. Since logs involve mainly WRITE performance, the high performance WRITE disks are better options to change the ACTIVE LOG path.

MIRRORLOGPATH

To protect the logs on the primary log path from disk failure or accidental deletion, you can specify that an identical set of logs be maintained on a secondary (mirror) log path. To do this, change the value of this configuration parameter to point to a different directory.

LOGPRIMARY

This parameter specifies the number of primary logs of size *logfilsz* that will be created. A primary log, whether empty or full, requires the same amount of disk space. Thus, if you configure more logs than you need, you use disk space unnecessarily. If you configure too few logs, you can encounter a log-full condition. As you select the number of logs to configure, you must consider the size you make each log and whether your application can handle a log-full condition.

LOGSECOND

This parameter specifies the number of secondary log files that are created and used for recovery, if needed. If the primary log files become full, secondary log files (of size *logfilsiz*) are allocated, one at a time as needed, up to the maximum number specified by this parameter. If this parameter is set to -1, the database is configured with infinite active log space. There is no limit on the size or number of in-flight transactions running on the database.

BLK_LOG_DSK_FUL

This configuration parameter can be set to prevent disk full errors from being generated when DB2 cannot create a new log file in the active log path. Instead, DB2 will attempt to create the log file every five minutes until it succeeds.

OVERFLOWLOGPATH

This parameter can be used for several functions, depending on your logging requirements. You can specify a location for DB2 to find log files that are needed for a roll forward operation. It is similar to the OVERFLOW LOG PATH option of the ROLLFORWARD command; however, instead of specifying the OVERFLOW LOG PATH option for every ROLLFORWARD command issued, you can set this configuration parameter once. If both are used, the OVERFLOW LOG PATH option will overwrite the *overflowlogpath* configuration parameter for that roll forward operation.

Update Logging Parameters

The database logging parameters are updated using following command. First we need to change the logging method from circular to recovery by updating LOGRETAIN to ON.

```
C:\>db2 UPDATE DB CFG FOR SAMPLE USING logretain ON
```

```
C:\>db2 connect to sample
SQL1116N  A connection to or activation of database "SAMPLE" cannot be
made because of BACKUP PENDING.  SQLSTATE=57019

C:\>db2 backup database sample to c:\db2backup
```

After taking the backup, the other database logging parameters can be changed to affect the new log path, a new mirror log path and log archival path.

```
C:\>db2 UPDATE DB CFG FOR SAMPLE USING logarchmeth1
"DISK:C:\DB2PrimaryLogArch" logprimary 10 logsecond -1 logfilsiz 1000
newlogpath C:\DB2LOG mirrorlogpath C:\DB2MirrorLog failarchpath
"C:\DB2FailureLogArch"
```

After we update above logging parameters, on activation of database , the new log path, mirror log path, primary archival log path and failure archival log path will be used by the database.

```
C:\>db2 activate db sample
```

The activation of database can occur on a new connection also. The ACTIVATE command activates all database related tasks without establishing database connections.

ONLINE BACKUP

Oracle

In Oracle, each tablespace that needs to be backed-up must be switched into backup mode before copying the files to storage.

```
ALTER TABLESPACE xyz BEGIN BACKUP;
!cp xyfile1 /backupdir/
ALTER TABLESPACE xyz END BACKUP;
```

When a tablespace is put in backup mode, following three things happen:

1. Oracle checkpoints the tablespace, flushing all changes from shared memory to the disk.
2. The SCN markers for each data file in that tablespace are frozen at their current values. Even though further updates will be sent to the data files, the SCN markers will not be updated until the tablespace is taken out of the backup mode.
3. Oracle switches to logging full images of changed database blocks to the redologs. Instead of recording how it changed a particular block (the change vector), it will log the entire image of the block after the change. This is why the redo logs grow at a much faster rate while hot backups are going on.

When *Begin backup* is issued, an operating system copy should follow, and then quickly end by an *End backup*. The backup is restricted to sequential backup meaning after backing up data files in tablespace1 then go on for data files in tablespace2.

Oracle recommends the minimum time elapsed between *Begin* and *End backup* commands. Also, minimum user activities should be allowed during this time for the fact that Oracle logs the entire block images on the first change to the block.

For online backup, special care needs to be taken for the SYSTEM table space and rollback segments table space. You need to also make sure that the control files are backed up once there are schema changes to the database. The backup scripts should also reflect the latest schema.

DB2

The mechanism for DB2 to do the recovery to the point in time is similar to Oracle. When logretain parameter is set to ON, roll forward recovery is enabled. Log files are kept and not reused, so they can be applied when performing roll forward recovery. There are three definitions of logs in DB2 that we need to familiar with:

Active Logs

Active logs contain transactions that are not committed or rolled back or committed transactions not yet flushed out to disks.

Online Archive Logs

Logs contain information for committed and flushed transactions.

Offline Archive Logs

Archive logs that are being moved from active log directory to either another directory in another media.

The online backups are allowed while the database is started and is in use. If online database backups are performed and users are updating it, it is possible that the data backed up will be inconsistent. The database uses log files during recovery process to recover the database to a fully consistent state. This requires that you retain all the active log files.

The following is an example of online full database backup, while database is in use.

```
C:\>db2 backup database sample online to c:\db2backup
```

After a complete online backup, the log files need to be also backed up. Since the log files are in active state, they can not be backed up. The ARCHIVE LOG can be used after a successful backup to close and archives all active log files.

```
C:\>db2 archive log for database sample
```

After issuing above command, the archive log files can be backed up and keep with the recent database backup files. You can use these backup files (database + archive logs) to restore the database point in time when you took the online backup even though you lose all the archive logs in production server.

To overcome a complete site loss, DB2 provides two alternative locations to keep the archive logs. We configured these two parameters in previous section.

The backup command includes history files, system files, log files, control files and so on.

Online Tablespace Backup

DB2 provides online tablespace backup. If database is very large, you may consider the combination of database and tablespace backup. You can take one tablespace or several tablespaces together in one backup file. If the size of one backup file exceeds the operating system limit, DB2 will create another backup file and number them sequentially.

From the recovery point of view, you should consider backing up all tablespaces all together, of which tables have referential integrity or triggers with each other. The related tablespaces at the application level should be backed up together even if they do not have RI and triggers for the data integrity. This way, you can recover where all the data is consistent.

The tablespace backup file also stores tablespace layout information automatically when a tablespace backup is taken.

The following command takes online backup of two tablespaces.

```
C:\>db2 backup database sample tablespace (userspace1,systoolspace)
online to c:\db2backup
C:\>db2 archive log for database sample
```

Comparison of Online Backup for Oracle and DB2

Oracle	DB2
Operation in ARCHIVELOG mode	Operation in LOGRETAIN mode
Redo logs contain both committed and uncommitted data. Oracle uses archived logs to roll forward to consistent state and Undo table space to rollback any uncommitted data.	Logs contain both committed and uncommitted data. In DB2, both roll forward and rollback are done using logs.
Online Redo Logs are written round robin manner namely when the last log is full, it will overwrite log 1. During log switch, online redo logs will be archived.	Logs are not written in round robin manner. Archived logs will not be overwritten. An active log will become an archived log upon closing by DB2
For a long running transaction when all logs are active, online redo logs will only be overwritten if either you manually archive the non-active online redo or that you have the ARCH process to do it for you.	For a long running transaction when all logs are active, secondary logs will be allocated according to parameter LOGSECOND
LOG_ARCHIVE_DUPLEX_DEST, LOG_ARCHIVE_DEST_N can be used to specify more than one location Oracle write logs to. Maximum value for LOG_ARCHIVE_DEST_N is 10. When one path is down, Oracle will ignore it and continue functioning as long as there is one path that works. However, a message will be logged in audit trail. No synchronization of the paths will happen.	MIRRORLOGPATH configuration parameter is used to multiplex logs. Maximum value for MIRRORLOGPATH is 1. When one path is down, DB2 will write to DB2 administration notification log and continue functioning as long as there is one path that works. However, DB2 will continually trying to write to the bad path when the current log is completed. No synchronization of the paths will happen.
Enabling ArchiveLog mode will require a	Enabling LOGRETAIN will only be effective

Oracle	DB2
database rebound, and a close and clean full backup.	after a DB2 database restart. You will need a full offline backup
Table space level backup	Both table space and database level backups
Oracle 10g introduced compression in Backup.	Backup compression can be specified while taking the database.
Database restore and roll forward does not necessary require the database be shutdown.	Database restore and roll forward must be done offline. Restore and roll forward for table space can be done online.

ONLINE RESTORE

Oracle

The Oracle RESTORE and RECOVER is mainly dependent upon what is lost. There are many scenarios in which RESTORE and RECOVER can be applied differently.

Control and SPFILE are intact but all data files are lost.

Oracle RMAN has two methods called RESTORE and RECOVER. The restore is used to bring up database from backup. The RECOVER applies redo logs to the database to roll it forward. One can roll-forward until a specific point in time (before the disaster occurred) or roll-forward until the last transaction recorded in the log file.

```
RMAN> RESTORE DATABASE;
RMAN> RECOVER DATABASE DELETE ARCHIVELOG MAXSIZE 25MB;
RMAN> RESTORE DATABASE CHECK READONLY;
RMAN> RECOVER DATABASE DELETE ARCHIVELOG;
RMAN> ALTER DATABASE OPEN;
```

Restore and Recovery of Individual Tablespaces

In this scenario, some but not all of the data files are damaged.

```
RMAN> SQL 'ALTER TABLESPACE users OFFLINE IMMEDIATE';
RMAN> RESTORE TABLESPACE users;
RMAN> RECOVER TABLESPACE users;
RMAN> SQL 'ALTER TABLESPACE users ONLINE';
```

Restore the control file from backup

In this scenario, loss of control files can be restored from backup by using RESTORE CONTROLFILE command. There are limitations in using backup control file and you must follow procedures to make sure that you do not run into documented problems.

Restore the Server Parameter File (SPFILE) from backup

In this scenario, loss of SPFILE can be restored from backup by using the command RESTORE SPFILE FROM command. The database instance need to be shut down and restarted after restore of SPFILE.

Restore / Recover Data Files and Tablespaces to new Location

Oracle allows you to restore data files and tablespaces to new locations.

Perform Media Recovery of a Database, Tablespace or Data file

Media recovery reapplies all changes from the archived and online redo logs and available incremental backups to data files restored from backup.

```
RMAN> RECOVER DATABASE;
```

DB2

The DB2 has a simple database restore and recovery (also known as roll-forward in DB2). The database can be restored fully on media failure and roll forward to point in time.

The database RESTORE needs to be done in an offline mode.

```
C:\>db2 restore database sample from c:\db2backup taken
at 20040901194626 without prompting
```

```
C:\>db2 rollforward database sample to end of logs and complete
```

Rollforward Status

```
Input database alias           = sample
Number of nodes have returned status = 1

Node number                    = 0
Rollforward status             = not pending
Next log file to be read       =
Log files processed            = S0000000.LOG - S0000003.LOG
Last committed transaction     = 2004-09-02-00.14.37.000000
```

```
DB20000I The ROLLFORWARD command completed successfully.
```

The tablespace can be restored online using RESTORE and ROLLFORWARD commands.

```
C:\>db2 restore database sample tablespace
(userspace1,systoolspace) online from c:\db2backup
taken at 20040901201428 without prompting
```

```
DB20000I The RESTORE DATABASE command completed successfully.
```

```
C:\>db2 rollforward database sample to end of logs
and complete tablespace(userspace1,systoolspace) online
```

```

                                Rollforward Status

Input database alias              = sample
Number of nodes have returned status = 1

Node number                       = 0
Rollforward status                 = not pending
Next log file to be read           =
Log files processed                 = -
Last committed transaction         = 2004-09-02-00.14.37.000000

DB20000I  The ROLLFORWARD command completed successfully.

```

INCREMENTAL BACKUP AND RESTORE

As database size grows very large, it may not be feasible to take database backup everyday. In a data warehouse environment, it may be just better to backup the changed pages instead of the entire database or tablespaces.

Oracle

Oracle's Recovery Manager is used to do incremental backup. The incremental backup option instructs the tool to back up blocks that have changed since the last incremental backup at the same level or below. For instance, a full backup (level_0) is taken on day 1 and two incrementals of level_1 are taken on days 2 and 3. The latter two merely back up the changed blocks between days 1 and 2 and days 2 and 3, not across the entire backup time. This strategy reduces backup size, requiring less space, and narrows the backup window, reducing the amount of data moving across the network.

RMAN uses a file, analogous to journals in filesystems, to track the blocks that have changed since the last backup. RMAN reads this file to determine which blocks are to be backed up.

You can enable this tracking mechanism by issuing the following command:

```
SQL> alter database enable block change tracking
using file '/rman_bkups/change.log';
```

This command creates a binary file called /rman_bkups/change.log for tracking purposes. Conversely, you can disable tracking with

```
SQL> alter database disable block change tracking;
```

To see whether change tracking is currently enabled, you can query:

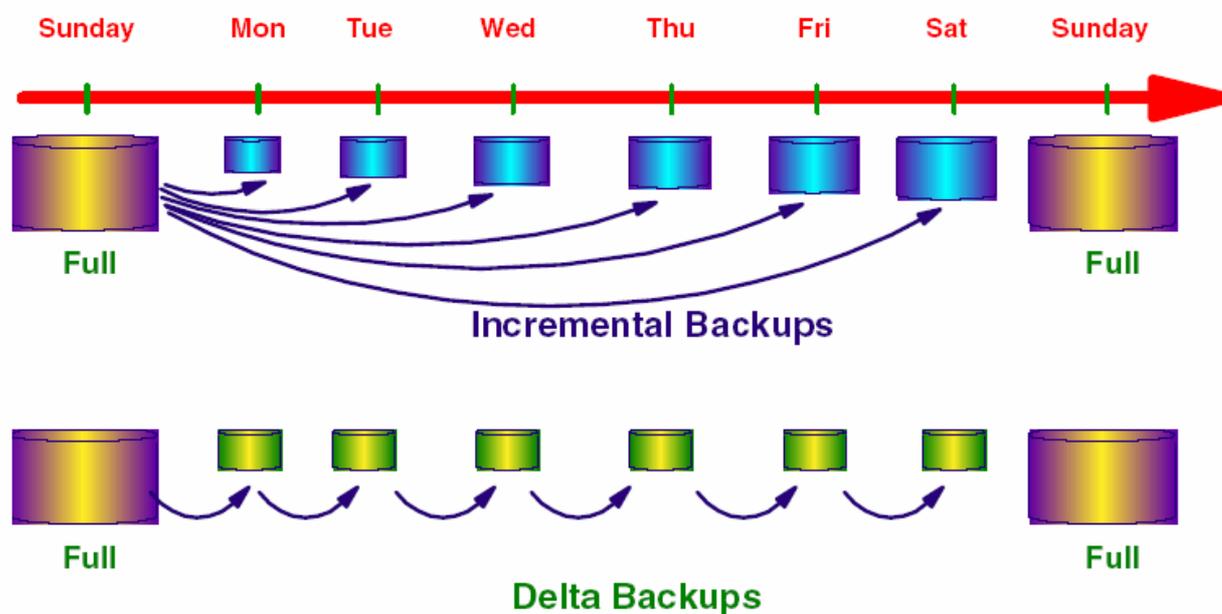
```
SQL> select filename, status from v$block_change_tracking;
```

Oracle RMAN can do the incremental merge from the incremental backups.

DB2

There are two types of incremental backup in DB2:

Incremental Backup: This is a cumulative backup; all changes since the last full database backup are backed up.



Delta Backup: This is a non cumulative backup; all the changes since the most recent backup which includes full, incremental or delta backup are backed up.

The database parameter TRACKMOD should be set to YES to use the incremental backup feature.

Use UPDATE command to set this parameter:

```
C:\>db2 UPDATE DB CFG FOR SAMPLE USING TRACKMOD YES
```

After changing this database parameter, taking a backup of database or tablespace is a good point to start an incremental backup. A full database or a tablespace backup, before incremental backups are taken, is necessary to recover a database from the incremental backups.

Full database backup for incremental backup consistency:

```
C:\>db2 backup database sample to c:\db2backup
```

Incremental delta backup and incremental backup:

```
C:\>db2 backup database sample online incremental delta to c:\db2backup
C:\>db2 backup database sample online incremental to c:\db2backup
```

Restoring from Incremental Backup

The restore without losing the data, first the full backup needs to be applied. You can use automatic keyword to automatic restore from the full backup and apply later days' incremental backup image. At this moment, you will still be in roll forward pending mode since there will be transactions to be roll forward.

```

C:\>db2 restore database sample incremental automatic
from c:\DB2Backup taken at 20040901230402 without prompting

SQL2540W  Restore is successful, however a warning "2539" was encountered
during Database Restore while processing in No Interrupt mode.

C:\>db2 rollforward database sample to end of logs and complete

                                Rollforward Status

Input database alias              = sample
Number of nodes have returned status = 1

Node number                       = 0
Rollforward status                = not pending
Next log file to be read          =
Log files processed                = S0000005.LOG - S0000006.LOG
Last committed transaction        = 2004-09-02-03.04.38.000000

DB20000I  The ROLLFORWARD command completed successfully.

```

RECOVER DROPPED TABLE

Sometime we need to recover a dropped table from database. Both Oracle and DB2 provide ways to retrieve a dropped table. Oracle has a simpler method but it is not foolproof and to retrieve a dropped table from backup is not easy in Oracle as you need to create another database to retrieve the table. DB2 provides a simpler method to recover a dropped table.

Oracle

If FLASHBACK option is enabled, Oracle provides a simple method to retrieve a dropped table by using FLASHBACK Query. The sample command to reclaim a dropped table is:

```
SQL> FLASHBACK TABLE RECYCLETEST TO BEFORE DROP;
```

The above is simple but it has an associated cost as dropped table are not dropped but are put in a recycle bin. The space is not reclaimed with a dropping of a table. To permanently delete the table, use the command:

```
DROP TABLE RECYCLETEST PURGE;
```

But if above command is used accidentally and if one needs to reclaim permanently dropped table, you have to re-create the database on another server and perform point-in-time recovery of that copy to a time before the table was dropped, export the dropped table using Oracle export utility, and re-import in into your primary database using an Oracle import utility.

DB2

DB2 allows you to recover a one table at a time. The following procedure is an example to recover a dropped table.

Turn on the tablespace recover option for the tablespace.

```
C:\>db2 select TBSPACE, DROP_RECOVERY from SYSCAT.TABLESPACES
```

TBSPACE	DROP_RECOVERY
SYSCATSPACE	N
TEMPSPACE1	N
USERSPACE1	Y

3 record(s) selected.

For USERSPACE1 tablespace, the drop recovery is ON. If it is off, you could turn it on using command:

```
C:\>db2 ALTER TABLESPACE userspace1 DROPPED TABLE RECOVERY ON
DB20000I The SQL command completed successfully.
```

For USERSPACE1 tablespace, the drop recovery is ON. If it is off, you could turn it on using command:

Turn on Recovery

The *logretain* database parameter should be on to enable RECOVERY. If it is not enabled, use following command to enable it.

```
C:\>db2 update db cfg for sample using logretain on
C:\>db2 get db cfg for sample | grep -i logretain
Log retain for recovery enabled          (LOGRETAIN) = RECOVERY
```

Backup the database:

```
C:\>db2 backup db sample to c:\db2backup
```

Create a table to be recovered:

```
C:\>db2 create table recoverme (c1 int not null)
C:\>db2 insert into recoverme values 91),(2),(3),(4),(5),(6),(7)
```

Drop the Table:

```
C:\>db2 drop table recoverme
```

See History of Dropped Tables:

```
C:\>db2 list history dropped table all for sample

List History File for sample

Number of matching file entries = 1
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log Backup ID
-----
D T 20040902101135          00000000000002010002000d
-----
"VIKRAM"."RECOVERME" resides in 1 tablespace(s):

00001 USERSPACE1
-----
Comment: DROP TABLE
Start Time: 20040902101135
```

```
End Time: 20040902101135
Status: A
```

```
-----
EID: 4
```

```
DDL: CREATE TABLE "VIKRAM"."RECOVERME" ( "C1" INTEGER NOT NULL ) IN
"USERSPA
CE1" ;
-----
```

Restore Tablespace from Previous Backup Image

From above history, we need the backup ID of the dropped table and DDL to create the table. Now we can restore database from the backup image we made earlier. This can be either a database-level restore or a tablespace-level restore. The tablespace-level restore requires exclusive access only to the tablespace and not the entire database during the recovery. The following syntax restores the database tablespace (USERSPACE1) using the backup image we created before:

```
C:\>db2 restore db sample tablespace(userspace1) from c:\db2backup taken
at 20040902100931 without rolling forward without prompting
```

Unload Dropped Table Data from Backup

After restoring the tablespace, we need to roll-forward to recover the dropped table data in our backup directory. We use the backup ID of the dropped table to unload the data.

```
C:\>db2 rollforward db sample to end of logs and complete recover dropped
table 00000000000002010002000d to c:\db2backup
```

Re-create Table using DDL from History File

```
C:\>db2 CREATE TABLE "VIKRAM"."RECOVERME" ( "C1" INTEGER NOT NULL )
IN "USERSPACE1"
```

Load Data in Table

```
C:\>db2 load from c:\db2backup\node0000\data OF DEL METHOD p(1)
insert into recoverme
```

After restoring the tablespace, we need to roll-forward to recover the dropped table data in our backup directory. We use the backup ID of the dropped table to unload the data.

CONCLUSION

The Backup and Restore are essential part of any relational database. Oracle and DB2 both provide mechanisms to offline and online backup and restore for both small and large databases. Both the databases also provide compression, tablespace backup, restore and re-location. For offline backup, the exclusive use is required to do a backup whereas online backup allow users to connect while backup is in progress. Both databases provide the roll forward until a certain point in time and to the end of the logs.

Dealing with huge databases such as those in data warehouse environment, both Oracle and DB2 provide incremental and differential backup to avoid copying unchanged data pages.

Oracle and DB2 UDB both provide effective mechanisms and capabilities for securing data from loss due to media corruption or human error. Using the information provided in this article, you should find it easy to transfer your Oracle skills to the DB2 UDB environment.