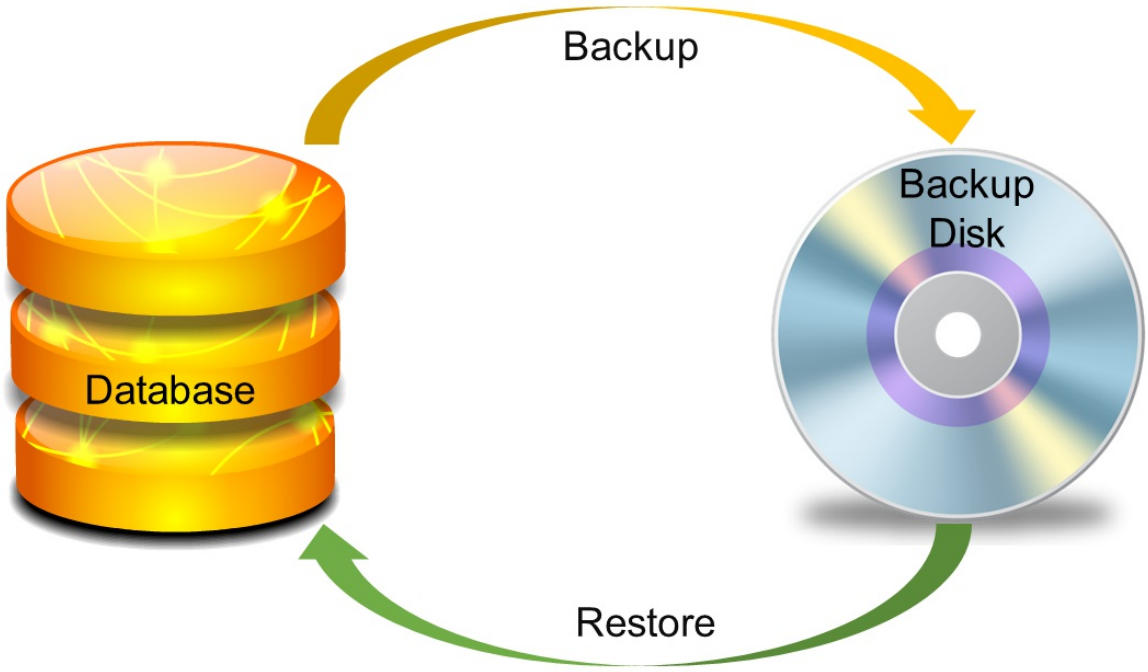
DB2 - BACKUP AND RECOVERY

This chapter describes backup and restore methods of database.



Introduction

Backup and recovery methods are designed to keep our information safe. In Command Line Interface *CLI* or Graphical User Interface *GUI* using backup and recovery utilities you can take backup or restore the data of databases in DB2 UDB.

Logging

Log files consist of error logs, which are used to recover from application errors. The logs keep the record of changes in the database. There are two types of logging as described below:

Circular logging

It is a method where the old transaction logs are overwritten when there is a need to allocate a new transaction log file, thus erasing the sequences of log files and reusing them. You are permitted to take only full back-up in offline mode. i.e., the database must be offline to take the full backup.

Archive logging

This mode supports for Online Backup and database recovery using log files called roll forward recovery. The mode of backup can be changed from circular to archive by setting logretain or userexit to ON. For archive logging, backup setting database require a directory that is writable for DB2 process.

Backup

Using **Backup** command you can take copy of entire database. This backup copy includes database system files, data files, log files, control information and so on.

You can take backup while working offline as well as online.

Offline backup

Syntax: [To list the active applications/databases]

```
db2 list application
```

Output:

```
Auth Id  Application    Appl.      Application Id
DB       # of
         Name           Handle
Name    Agents
-------- -------------- ---------- --------------------
-------------------------------------- -------- -----
DB2INST1 db2bp          39
*LOCAL.db2inst1.140722043938
ONE      1
```

Syntax: [To force application using app. Handled id]

```
db2 "force application (39)"
```

Output:

```
DB20000I  The FORCE APPLICATION command completed
successfully.

DB21024I  This command is asynchronous and may not
be effective immediately.
```

Syntax: [To terminate Database Connection]

```
db2 terminate
```

Syntax: [To deactivate Database]

```
db2 deactivate database one
```

Syntax: [To take the backup file]

```
db2 backup database <db_name> to <location>
```

Example:

```
db2 backup database one to /home/db2inst1/
```

Output:

```
Backup successful. The timestamp for this backup image is :
20140722105345
```

Online backup

To start, you need to change the mode from **Circular logging** to **Archive Logging**.

Syntax: [To check if the database is using circular or archive logging]

```
db2 get db cfg for one | grep LOGARCH
```

Output:

```
First log archive method (LOGARCHMETH1) = OFF
 Archive compression for logarchmeth1  (LOGARCHCOMPR1) = OFF
 Options for logarchmeth1              (LOGARCHOPT1) =
 Second log archive method             (LOGARCHMETH2) = OFF
 Archive compression for logarchmeth2  (LOGARCHCOMPR2) = OFF
 Options for logarchmeth2              (LOGARCHOPT2) =
```

In the above output, the highlighted values are [logarchmeth1 and logarchmeth2] in off mode, which implies that the current database in “CIRCULLAR LOGGING” mode. If you need to work with ‘ARCHIVE LOGGING’ mode, you need to change or add path in the variables logarchmeth1 and logarchmeth2 present in the configuration file.

Updating logarchmeth1 with required archive directory

Syntax: [To make directories]

```
mkdir backup
mkdir backup/ArchiveDest
```

Syntax: [To provide user permissions for folder]

```
chown db2inst1:db2iadm1 backup/ArchiveDest
```

Syntax: [To update configuration LOGARCHMETH1]

```
db2 update database configuration for one using LOGARCHMETH1
'DISK:/home/db2inst1/backup/ArchiveDest'
```

You can take offline backup for safety, activate the database and connect to it.

Syntax: [To take online backup]

```
db2 backup database one online to
/home/db2inst1/onlinebackup/ compress include logs
```

Output:

```
db2 backup database one online to
/home/db2inst1/onlinebackup/ compress include logs
```

Verify Backup file using following command:

Syntax:

```
db2ckbkp <location/backup file>
```

Example:

```
db2ckbkp
/home/db2inst1/ONE.0.db2inst1.DBPART000.20140722112743.001
```

Listing the history of backup files

Syntax:

```
db2 list history backup all for one
```

Output:

```
                    List History File for one
```

```
Number of matching file entries = 4

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log
Backup ID
 -- --- ------------------ ---- --- ------------ ------------
 --------------
  B  D  20140722105345001   F    D  S0000000.LOG S0000000.LOG

 ------------------------------------------------------------
 ----------------

 Contains 4 tablespace(s):
 00001 SYSCATSPACE

 00002 USERSPACE1

 00003 SYSTOOLSPACE

 00004 TS1
  ------------------------------------------------------------
  ----------------
  Comment: DB2 BACKUP ONE OFFLINE

 Start Time: 20140722105345

   End Time: 20140722105347

     Status: A
 ------------------------------------------------------------
 ----------------
 EID: 3 Location: /home/db2inst1


 Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log
 Backup ID
 -- --- ------------------ ---- --- ------------ ------------
 --------------
  B  D  20140722112239000   N       S0000000.LOG S0000000.LOG
 ------------------------------------------------------------
 ------------------------------------------------------------
 --------------------------------

 Comment: DB2 BACKUP ONE ONLINE

 Start Time: 20140722112239

   End Time: 20140722112240

     Status: A
 ------------------------------------------------------------
 ----------------
  EID: 4 Location:
SQLCA Information

 sqlcaid : SQLCA     sqlcabc: 136   sqlcode: -2413   sqlerrml: 0

 sqlerrmc:
 sqlerrp : sqlubIni
 sqlerrd : (1) 0               (2) 0               (3) 0

           (4) 0               (5) 0               (6) 0

 sqlwarn : (1)      (2)      (3)      (4)        (5)       (6)

           (7)      (8)      (9)      (10)       (11)
 sqlstate:

 Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log
 Backup ID
```

```
 -- --- ------------------ ---- --- ------------ ------------
 --------------
  B  D  20140722112743001   F    D  S0000000.LOG S0000000.LOG

 ------------------------------------------------------------
 ----------------
 Contains 4 tablespace(s):

 00001 SYSCATSPACE

 00002 USERSPACE1

 00003 SYSTOOLSPACE

 00004 TS1
  ------------------------------------------------------------
  ----------------
  Comment: DB2 BACKUP ONE OFFLINE

 Start Time: 20140722112743

   End Time: 20140722112743

     Status: A
 ------------------------------------------------------------
  ----------------
 EID: 5 Location: /home/db2inst1

 Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log
 Backup ID
  ------------------------------------------------------------
  ----------------

R  D  20140722114519001   F
20140722112743

 ------------------------------------------------------------
 ----------------
 Contains 4 tablespace(s):

 00001 SYSCATSPACE

  00002 USERSPACE1

 00003 SYSTOOLSPACE

 00004 TS1
 ------------------------------------------------------------
 ----------------
Comment: RESTORE ONE WITH RF

 Start Time: 20140722114519

   End Time: 20140722115015
     Status: A

 ------------------------------------------------------------
 ----------------
  EID: 6 Location:
```

Restoring the database from backup

To restore the database from backup file, you need to follow the given syntax:

Syntax:

```
db2 restore database <db_name> from <location>
taken at <timestamp>
```

Example:

```
db2 restore database one from /home/db2inst1/ taken at
20140722112743
```

Output:

```
SQL2523W  Warning!  Restoring to an existing database that is
different from

the database on the backup image, but have matching names.
The target database

will be overwritten by the backup version.  The Roll-forward
recovery logs

associated with the target database will be deleted.

Do you want to continue ? (y/n) y

DB20000I  The RESTORE DATABASE command completed successfully.
```

Roll forward all the logs located in the log directory, including latest changes just before the disk drive failure.

Syntax:

```
db2 rollforward db <db_name> to end of logs and stop
```

Example:

```
db2 rollforward db one to end of logs and stop
```

Output:

```
                                 Rollforward Status
 Input database alias                   = one
 Number of members have returned status = 1
 Member ID                               = 0
 Rollforward status                     = not pending
 Next log file to be read               =
 Log files processed                    = S0000000.LOG -
 S0000001.LOG
 Last committed transaction             = 2014-07-22-
 06.00.33.000000 UTC
DB20000I  The ROLLFORWARD command completed successfully.
```