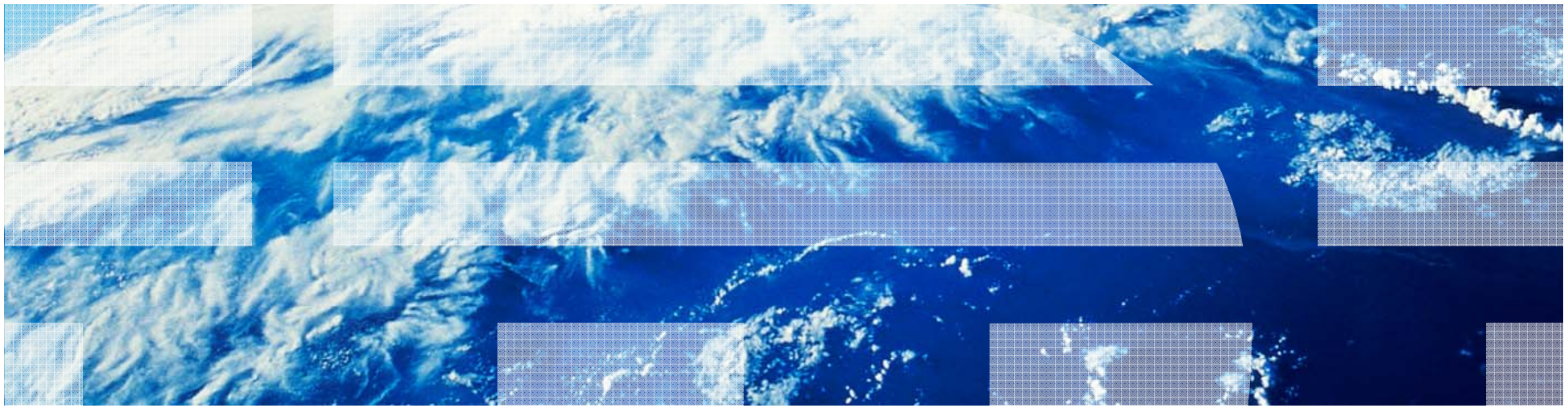
IBM

Information Management

Backup and recovery

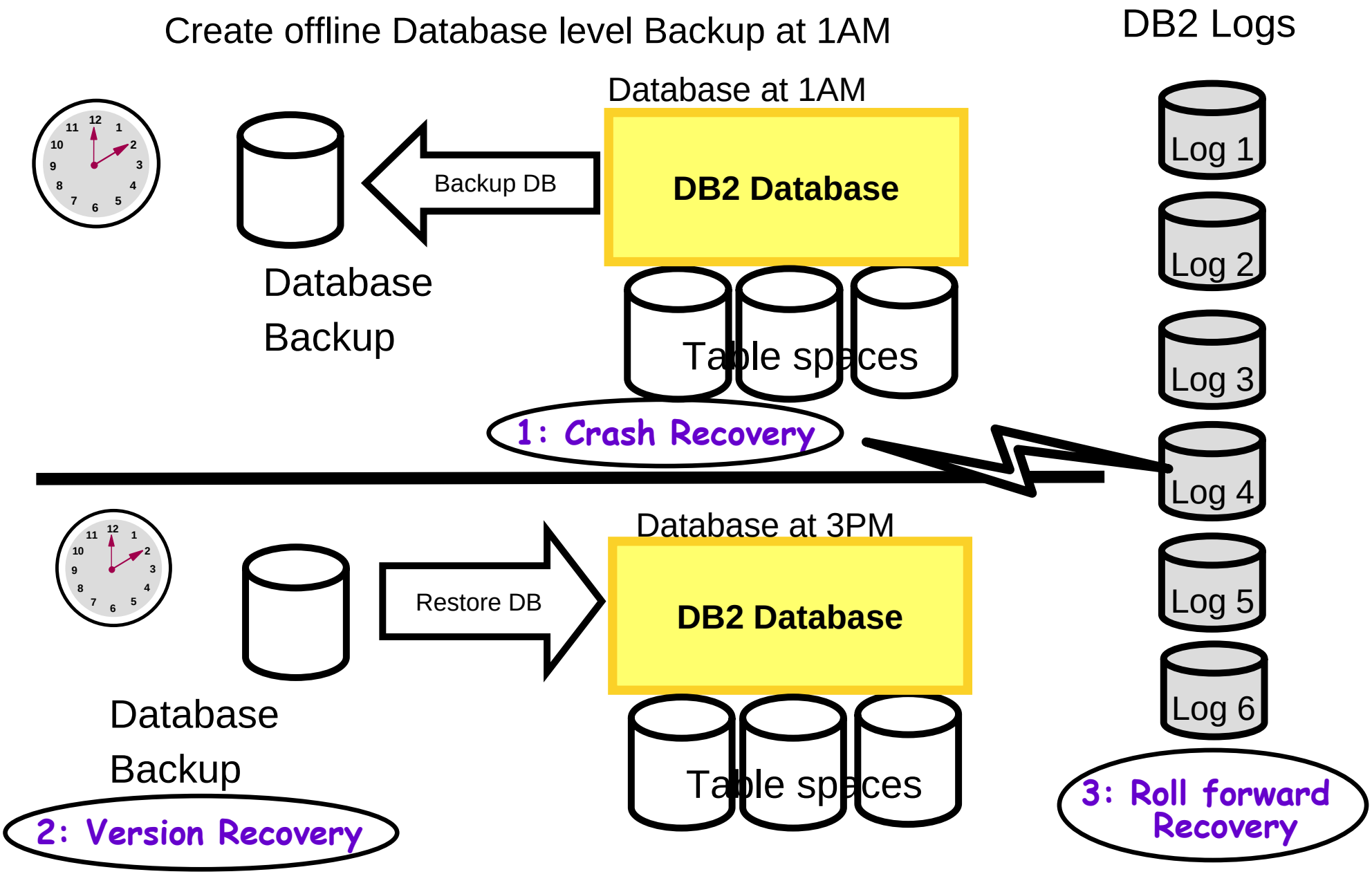


Unit objectives

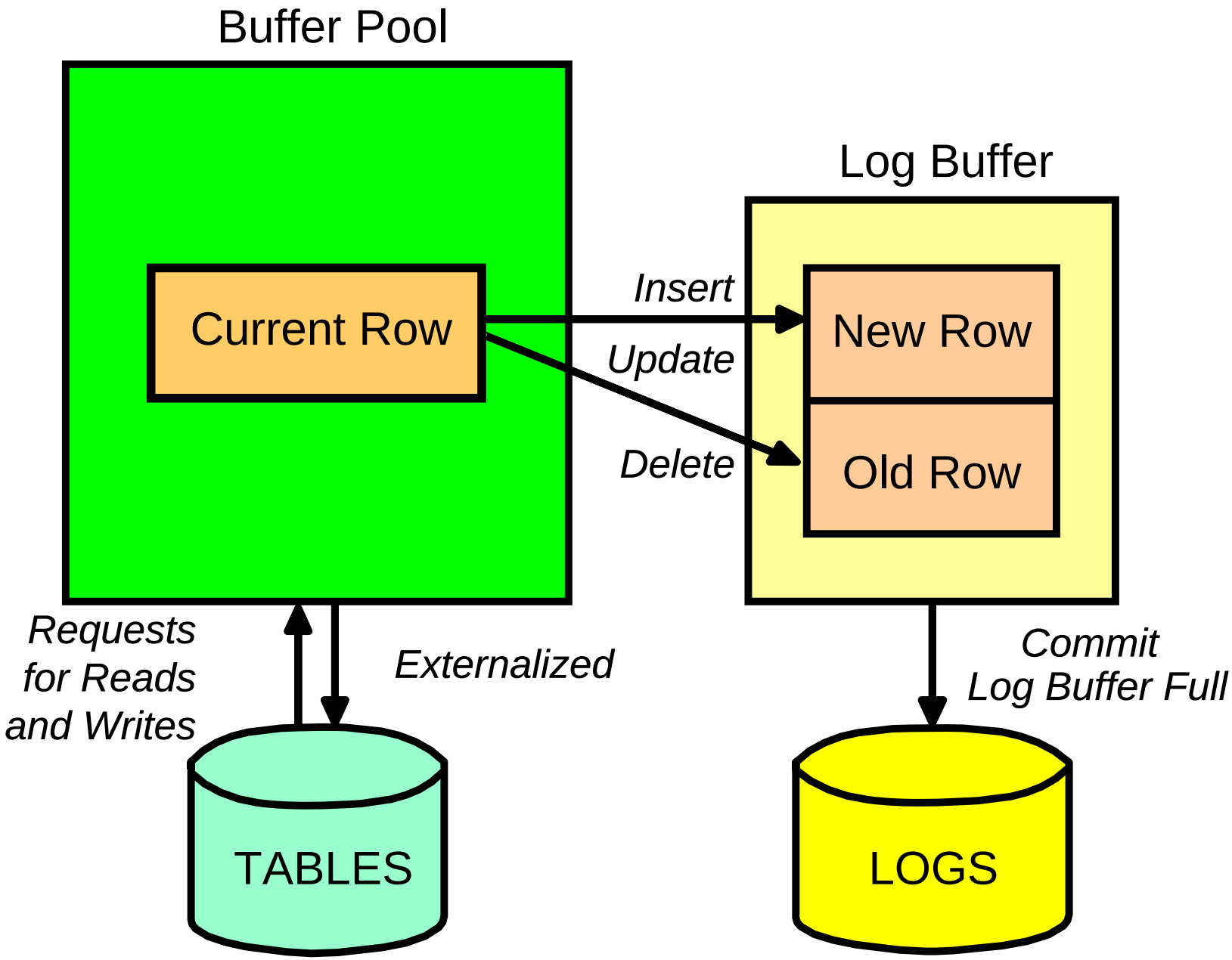
After completing this unit, you should be able to:

- Describe the major principles and methods for backup and recovery
- State the three types of recovery used by DB2
- Explain the importance of logging for backup and recovery
- Describe how data logging takes place, including circular logging and archival logging
- Use the BACKUP, RESTORE, ROLLFORWARD and RECOVER commands
- Perform a table space backup and recovery
- Restore a database to the end of logs or to a point-in-time
- Discuss the configuration parameters and the recovery history file and use these to handle various backup and recovery scenarios

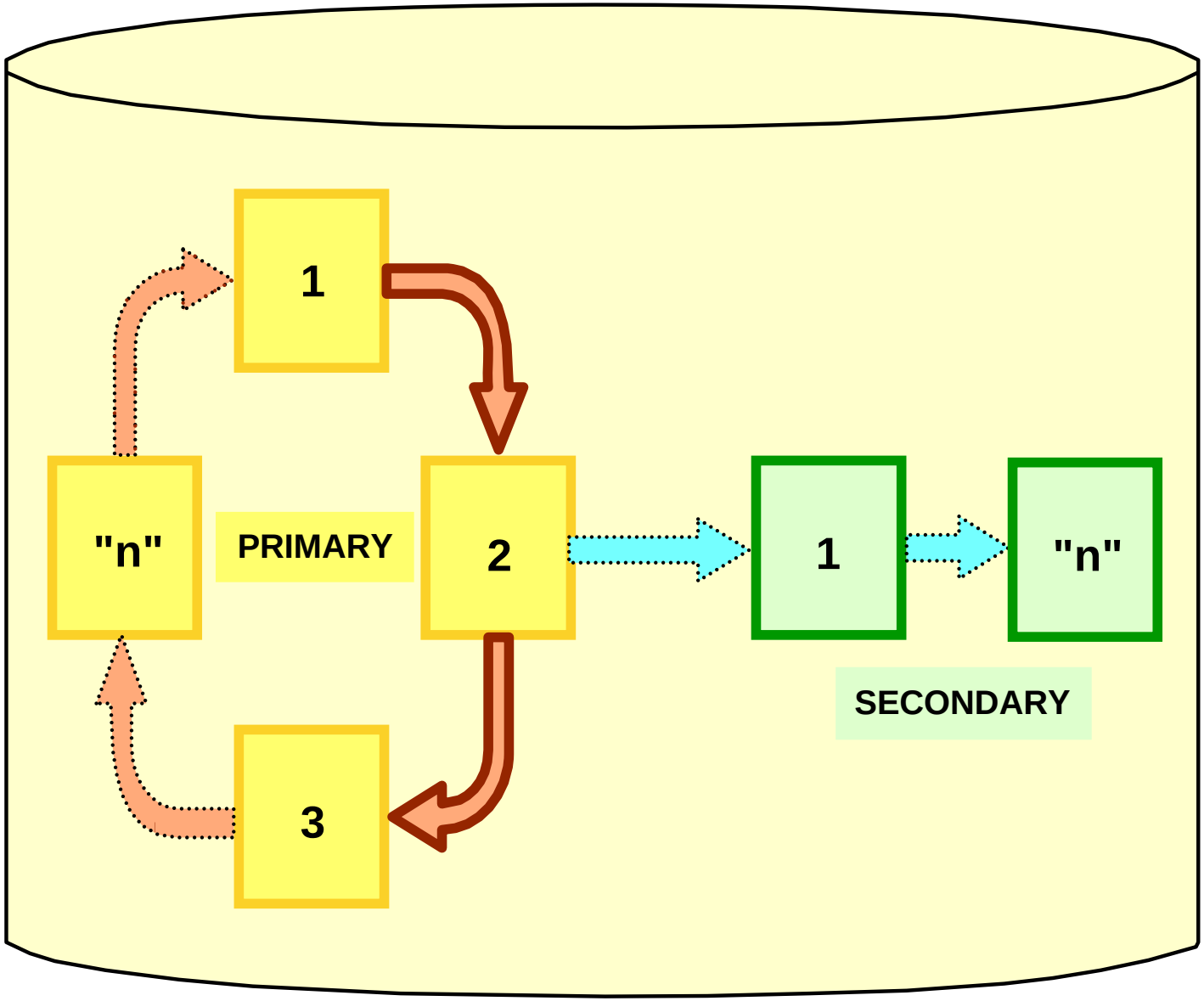
DB2 Database recovery methods



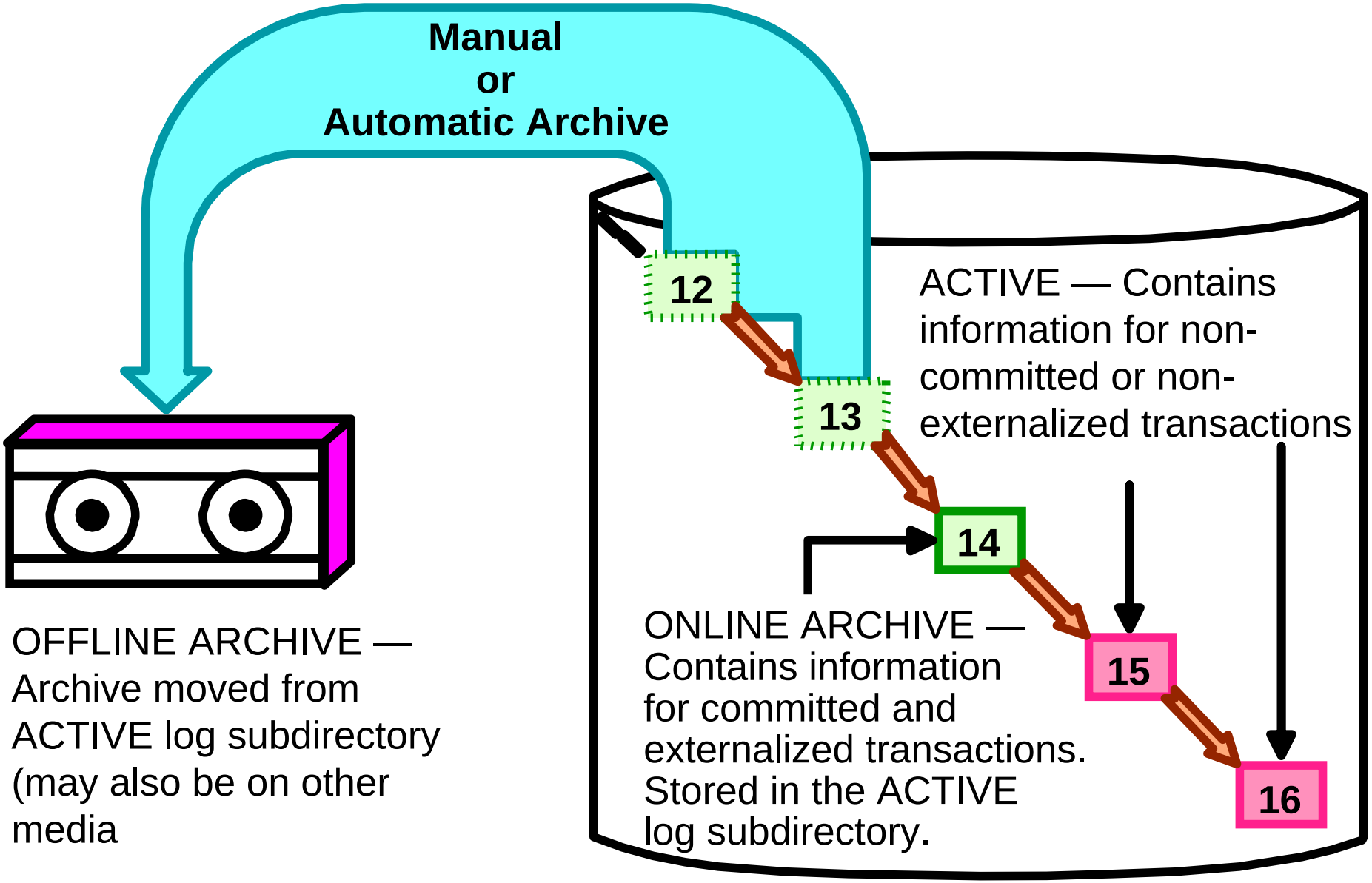
Introduction to logging



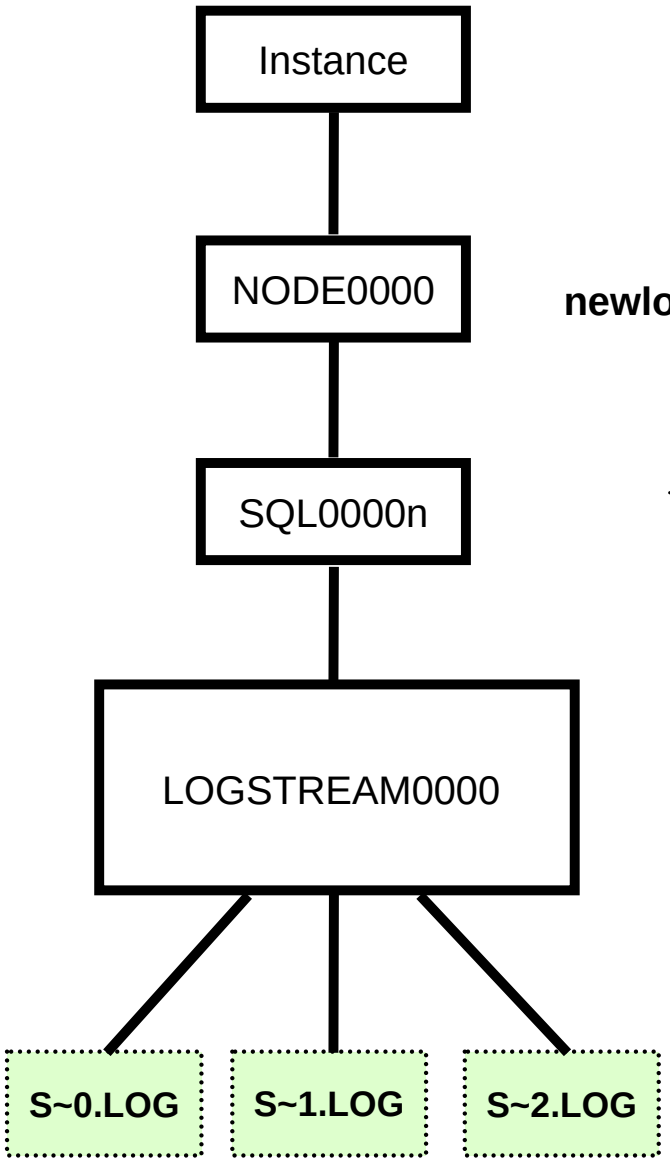
Circular logging (Non-recoverable database)

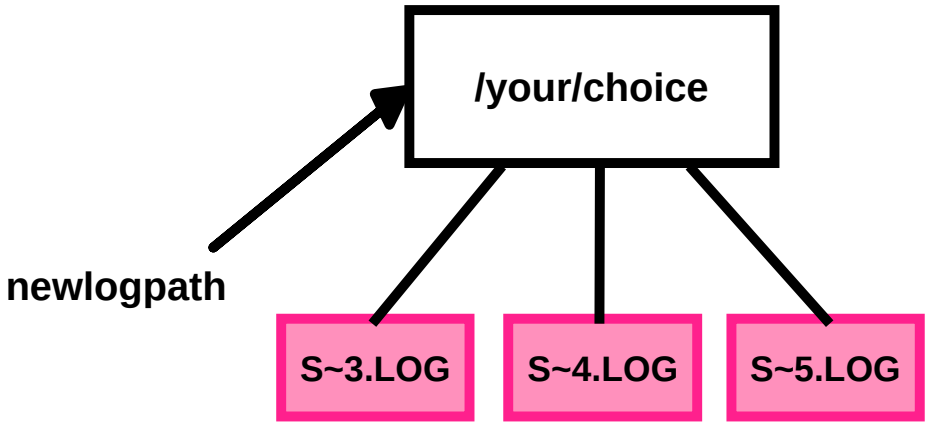


Archival logging (Recoverable database)

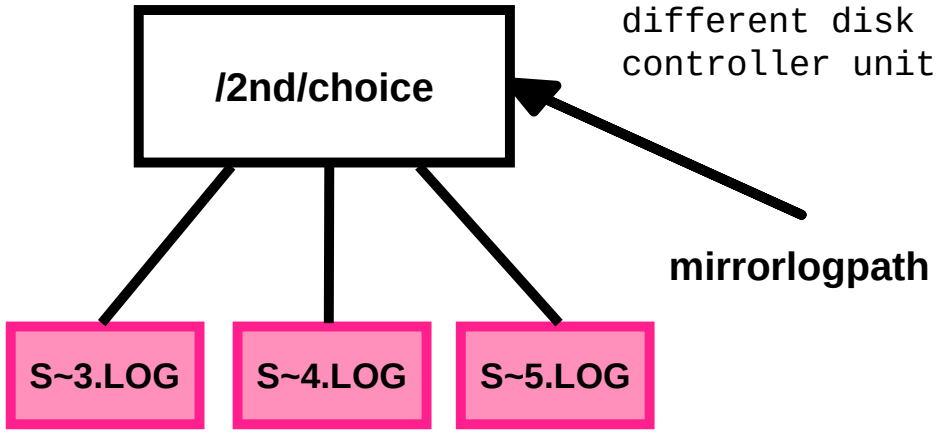


Location of log files





Different disk drive and potentially a different disk controller unit



Configure database logs: Parameters

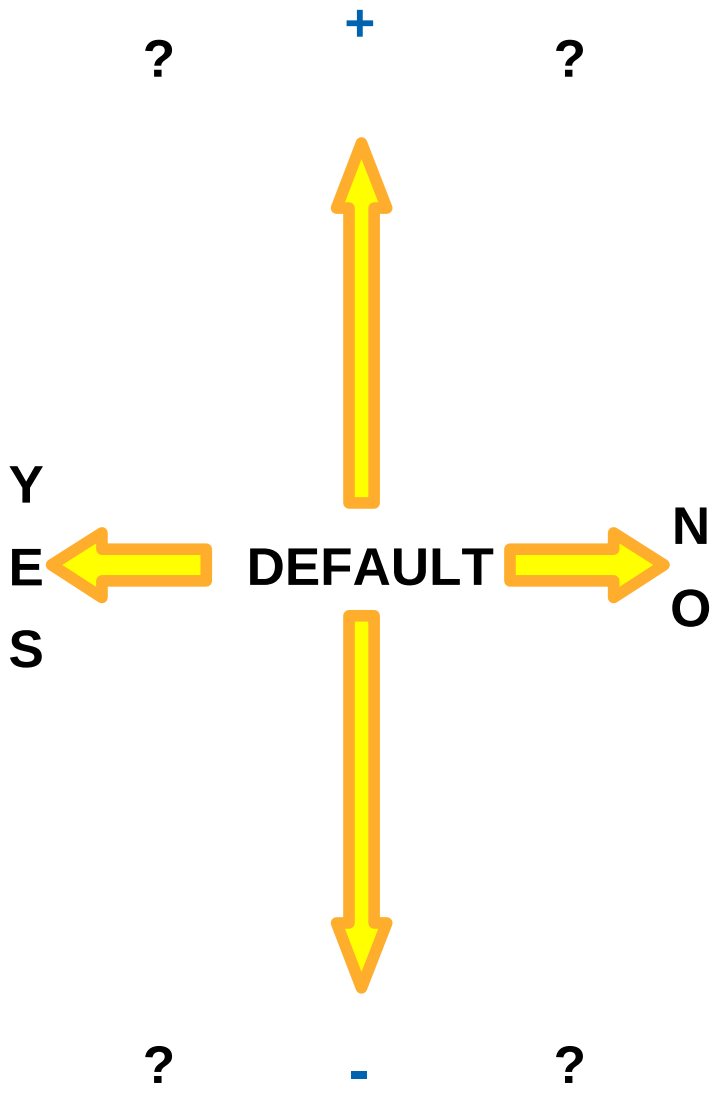
Primary log files (LOGPRIMARY)

Secondary log files (LOGSECOND)

Log size (LOGFILSIZ)

Log Buffer (LOGBUFSIZ)

Overflow log path (OVERFLOWLOGPATH)



New log path (NEWLOGPATH)

Mirror log path (MIRRORLOGPATH)

Primary Log Archive Method (LOGARCHMETH1)

Secondary Log Archive Method (LOGARCHMETH2)

Log records to write before soft checkpoint (SOFTMAX)

DB2 recovery-related system files

- Recovery History File - db2rhist.asc:
  - Created during Create Database command
  - Updated by DB2 Utility execution:
    - Back up database or table space
    - Restore database or table space
    - Roll forward database or table space
    - Load table
    - Reorg table
    - DB2 Log file archival
    - Create/Alter/Rename table space
    - Quiesce table spaces for table
    - Drop Table (optional)
  - Included on each DB2 backup file
  - db2 LIST HISTORY command
- Log Control File – SQLOGCTL.LFH.n and SQLOGCTL.GLFH.n:
  - Used during crash recovery
  - Disk updated at end of each Log File
  - Use softmax value < 100 (% of logfilsiz) to refresh pointers more often
  - Included at end of each DB2 backup

Backup Utility options (partial)

BACKUP DATABASE database-alias [USER username [USING password]]

[TABLESPACE (tblspace-name [{,tblspace-name} ...])] [ONLINE]

[INCREMENTAL [DELTA]]

[USE {TSM | XBSA} [OPEN num-sess SESSIONS]

[OPTIONS {options-string | options-filename}] | TO dir/dev

[{,dir/dev} ...] | LOAD lib-name [OPEN num-sess SESSIONS]
[OPTIONS {options-string | options-filename}]]

[WITH num-buff BUFFERS] [BUFFER buffer-size] [PARALLELISM n]

[COMPRESS [COMPRLIB lib-name [EXCLUDE]] [COMPROPTS options-string]]

[UTIL_IMPACT_PRIORITY [priority]
[{INCLUDE | EXCLUDE} LOGS] [WITHOUT PROMPTING]

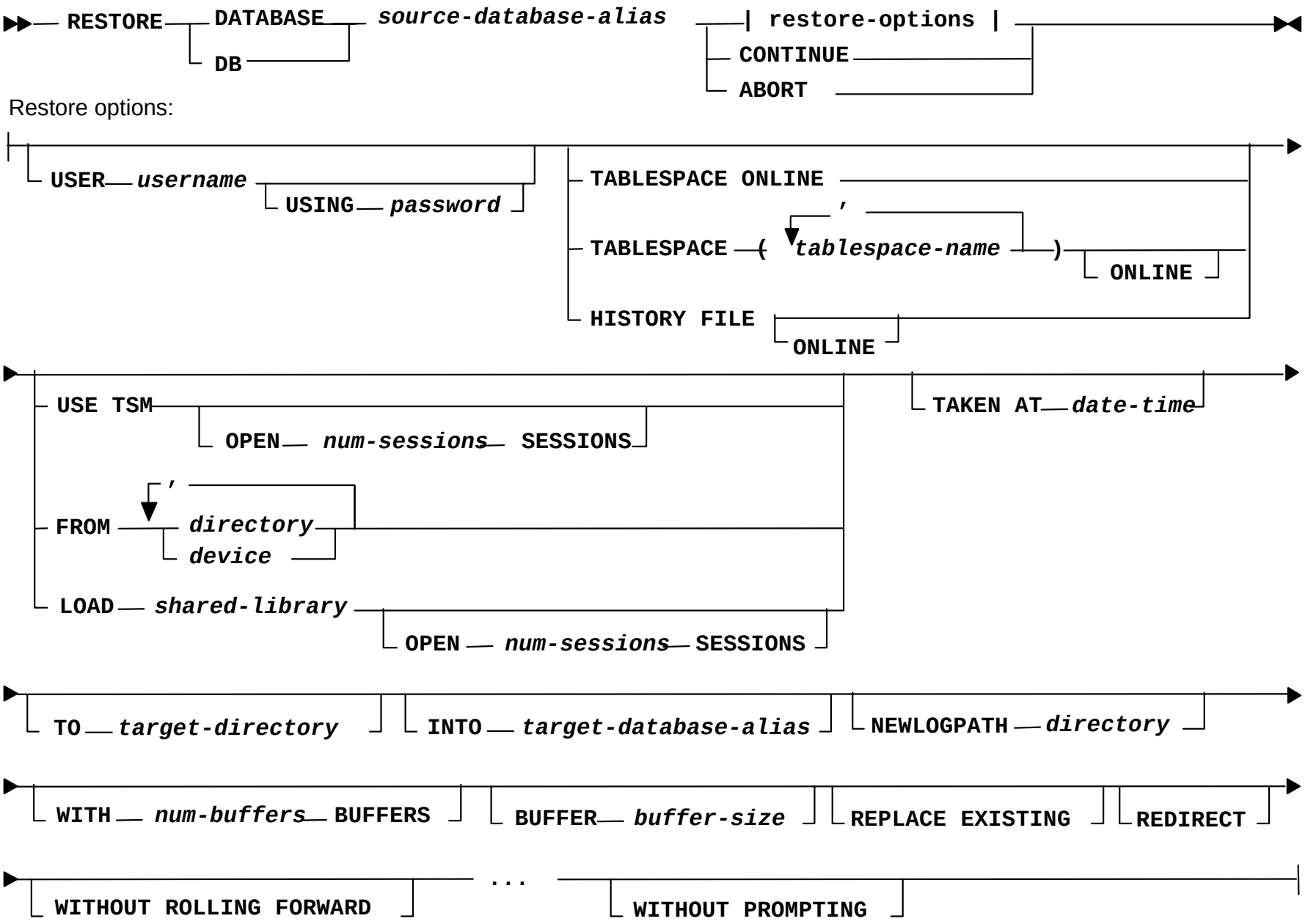
The backup files

- File name for backup images *on disk* has:
  - Database alias
  - Type of backup (0=full, 3=table space, 4=copy from table load)
  - Instance name
  - Database Partition (DBPARTnnn, nnn is 000 for single partition DB)
  - Timestamp of backup (*YYYYMMDDHHMMS*)
  - Sequence number (for multiple files)

MUSICDB.0.DB2.DBPART000. 20120522120112 .001

- Tape images are not named, but internally contain the same information in the backup header for verification purposes
- Backup history provides key information in easy-to-use format

Syntax of the RESTORE command (partial)



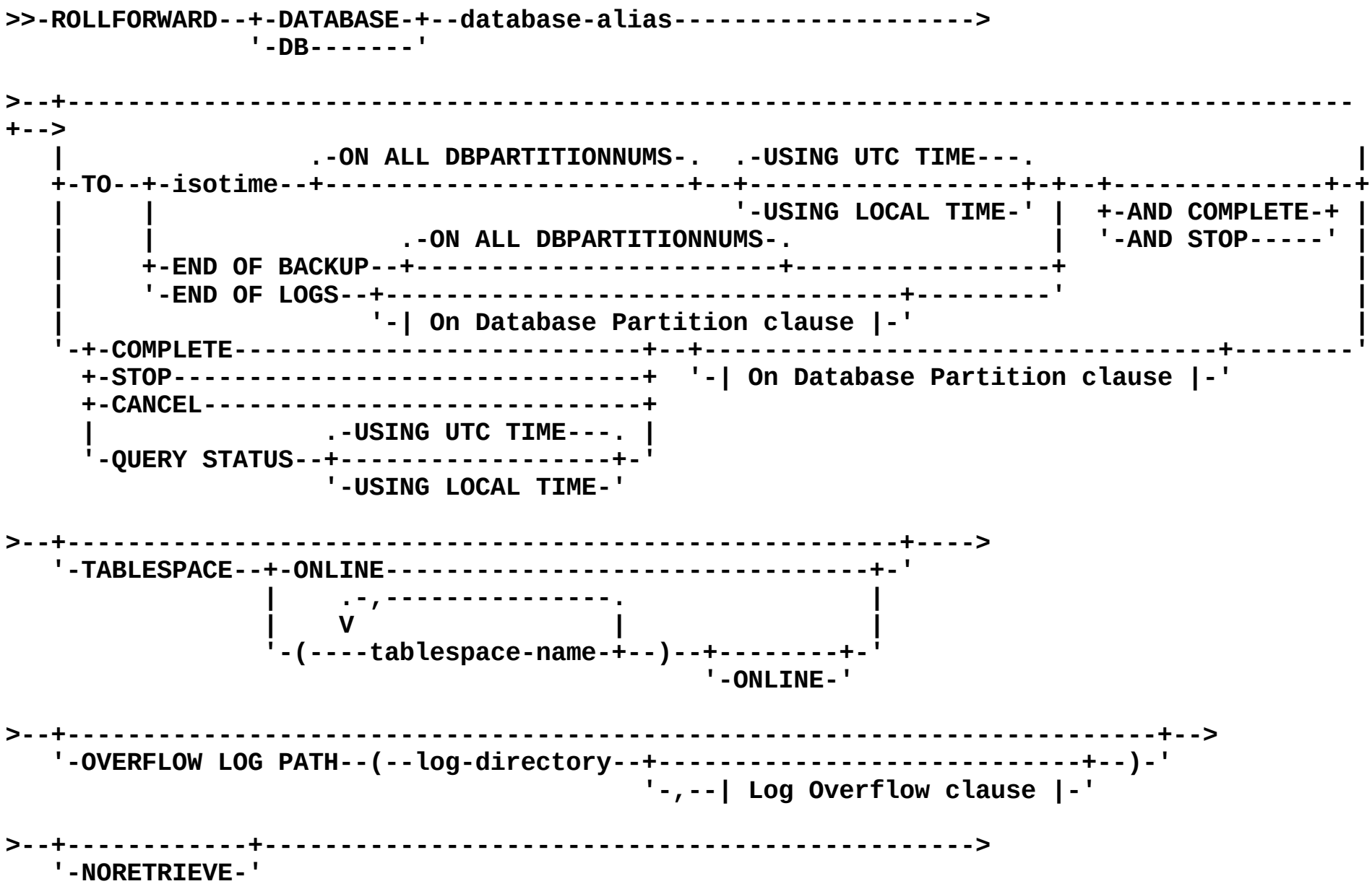
Backup/restore table space considerations

- Recoverable Database - Roll-forward must be enabled
- Can choose to restore a subset of table spaces
- Generally best to put multiple spaces in one backup image:
  - Makes table space recovery strategy easier
  - Provides access to related tables spaces and coherent management of these table spaces
- Handling of long/LOB/XML data requires a correlated strategy
- Point-in-time recovery is supported, but has requirements
- Faster recovery for Catalogs using Tablespace Level backup
- Critical business application tables should obviously be the *focus* of the backup/restore, but other tables are needed in support of these tables

Roll forward pending state

- *Roll forward pending* is set as a result of:
  - Restore of *offline* database backup omitting the command option WITHOUT ROLLING FORWARD
  - Restore of an *online* database backup
  - Restore of *any table space* level backup
  - DB2 detects media failure isolated at a table space
- Scope of pending state managed by DB2:
  - *Database* in pending state will not permit any activity
  - *Table spaces* in pending state will permit access to other table spaces

Syntax of the ROLLFORWARD command



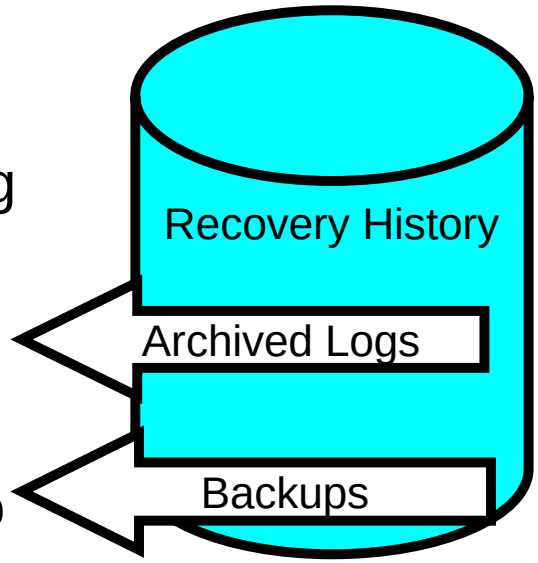
ROLLFORWARD: How far?

- END OF LOGS: (Apply as many changes as possible):
  - Rollforward will apply all available logs beginning with the logs associated with the backup that was restored
  - Archived logs will be retrieved unless NORETRIEVE is specified
- Point-in-time (PIT): (Apply changes up to a specified time):
  - Specified in Coordinated Universal Time (UTC) via command
  - Specified in local time *on server* with USING LOCAL TIME
  - Specified in local time on the client via GUI interface
  - Format: yyyy-mm-dd-hh.mm.ss.nnnnnn
- END OF BACKUP: (Apply as few changes as possible):
  - Allows a Database to be recovered from an online database backup and to end the ROLLFORWARD processing at the earliest point where the database is consistent.
  - Recovery history file (RHF) shows logs associated with online backups
- Table space point-in-time considerations:
  - Minimum roll forward time maintained for each table space – requires roll forward at least to the last DDL change (create, alter, drop) in a table space
  - Table spaces are placed in backup pending when the roll forward completes to insure future recoverability

DB2 RECOVER command

```
db2 recover database salesdb
```

- RECOVER command performs both RESTORE and ROLLFORWARD command processing using Recovery History file data.
- Can use Full or Incremental Database level backup images
- Allows Partitioned DPF database to be recovered using a single command
- Database can be recovered to *End of logs* or a point in time.
- RESTART option forces failed RECOVER command to redo RESTORE even if previous restore completed
- USING HISTORY FILE option allows disaster recovery with no existing database
- No table space level recovery options

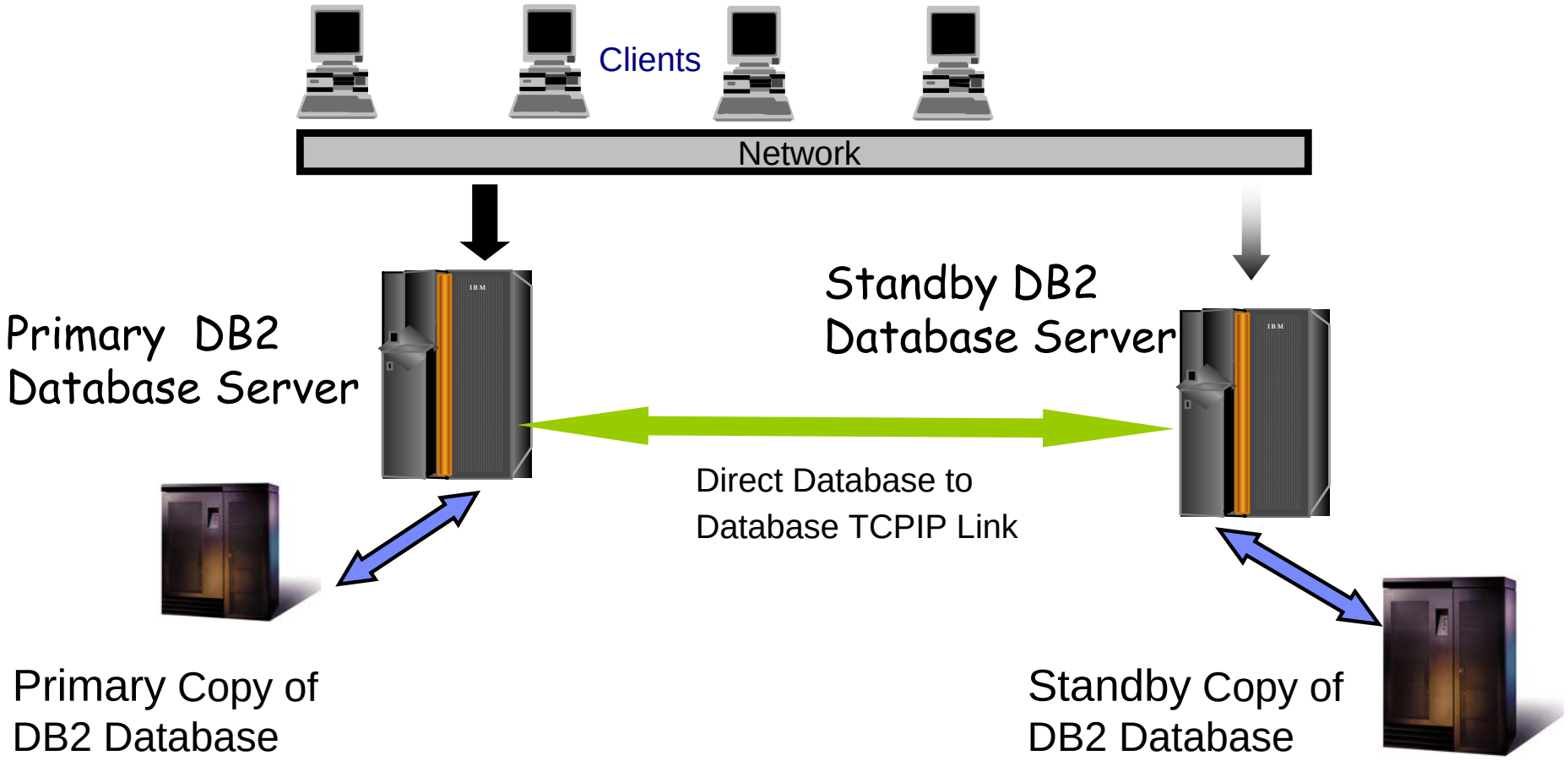


Disaster recovery considerations

Should you use database recovery, table space recovery, or a combination?

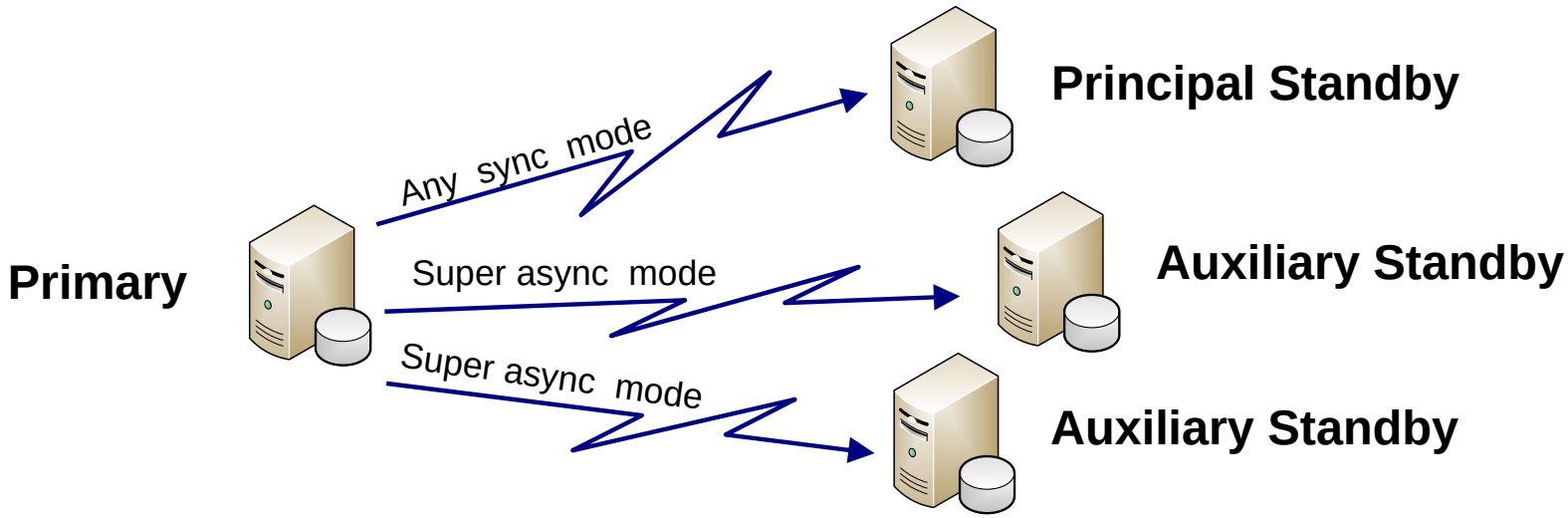
- Database level backups:
  - Offline backup does not require a roll forward
  - Online backup will include the ‘required’ logs for a roll forward TO END OF BACKUP
  - Entire database can be restored on another system
  - Reduces backup management
- Table space level backups:
  - Can be used to supplement database backups
  - A complete set of tablespace backups could be used to REBUILD a database for disaster recovery
  - Increases the number of backup images
- Using the REBUILD WITH option of RESTORE simplifies creating a full or partial copy of a database using either database or table space backup images

High Availability Disaster Recovery: HADR



High Availability Disaster Recovery (HADR) feature is included in all the for-purchase DB2 10.1 editions (Express, Workgroup Server, Enterprise Server, and Advanced Enterprise Server, Database Enterprise Developer).

DB2 10.1 support for multiple active standby databases



- HADR feature in multiple standby mode allows up to three standby databases to be configured

- One Standby is designated the *principal HADR standby database*
- Any additional standby database is an *auxiliary HADR standby database*
- Both types of HADR standbys:
  - Are synchronized with the HADR primary database through a direct TCP/IP connection
  - Support reads on standby
  - Can issue a forced or non-forced takeover

- Other HADR enhancements included in DB2 10.1
  - Log spooling on the Standby database
  - Delayed replay for a Standby database

DB2 Integrated Cluster Manager support

- HA Cluster Manager Integration:
  - Coupling of DB2 and TSA (S AMP) on Linux, Solaris and AIX.
  - TSA can be installed and maintained with DB2 installation procedures.
  - DB2 interface to configure cluster.
  - DB2 to maintain cluster configuration, add node, add table space, and so on. Exploitation of new vendor independent layering (VIL), providing support for any cluster manager.
- NO SCRIPTING REQUIRED!
  - One set of embedded scripts that are used by all cluster managers.
- Automates HADR failover
  - Exploit HA cluster manager integration to automate HADR Takeover on Standby.

DB2: Cluster configuration simplified

- DB2 utility db2haicu:
  - DB2 High Availability Instance Configuration Utility
  - Sets up DB2 with the Cluster Manager
    - For HADR or Shared Storage cluster
  - Interactive or XML file driven interface
  - Sets the DBM CFG option **cluster_mgr**
- DB2 communicates with the Cluster Manager:
  - Keeps the Cluster Manager in sync with DB2 changes:
    - CM needs to be aware of new or changed table spaces, containers, and so on
    - Avoids failover problems due to missing resources
  - Keeps the XML cluster configuration file up to date
  - Automates HADR failover
- Check IBM Developerworks for articles that provide detailed examples and suggestions for DB2 Cluster configuration

DB2 additional recovery facilities

- On-demand log archiving
- Infinite active logs
- Block transactions on log directory full
- Split mirror database copies:
  - SET WRITE SUSPEND/RESUME commands
  - db2inidb command modes:
    - **SNAPSHOT:** Database copy for testing reporting
    - **STANDBY:** Database copy to create standby database for quick recovery
    - **MIRROR:** Use split mirror database copy instead of RESTORE
- Incremental and delta database and table space backups
- Relocating a database or a table space:
  - RESTORE UTILITY with REDIRECT option
  - db2relocatedb command
- Full and partial database REBUILD support
- Integrated Cluster Failover support using TSA

DB2 for LUW Advanced Database Recovery

CL492: *DB2 for LUW Advanced Database Recovery (4 days with Labs)*:

- Automated Recovery log management
- Dropped table recovery
- Table space recovery
- Incremental backup and restore
- Database crash recovery
- Database and table space relocation methods
- Additional recovery facilities
- Recovery history file
- Disaster recovery of DB2 databases
- DB2 high availability and split mirror support
- DB2 partitioned database recovery considerations
- DB2 high availability disaster recovery (HADR) implementation

Unit summary

Having completed this unit, you should be able to:

- Describe the major principles and methods for backup and recovery
- State the three types of recovery used by DB2
- Explain the importance of logging for backup and recovery
- Describe how data logging takes place, including circular logging and archival logging
- Use the BACKUP, RESTORE, ROLLFORWARD and RECOVER commands
- Perform a table space backup and recovery
- Restore a database to the end of logs or to a point-in-time
- Discuss the configuration parameters and the recovery history file and use these to handle various backup and recovery scenarios

Student exercise