DB2 Certification

Data Management

Objectives

After completing this unit, you should be able to:

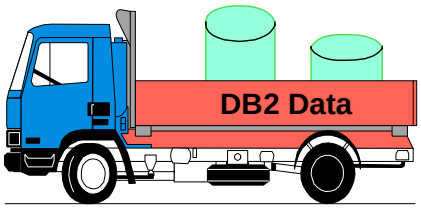
- talk about the methods of populating tables
  - LOAD, IMPORT & EXPORT
- discuss the advantages and disadvantages of these methods
- utilise the DB2 tools to maintain your tables
  - REORGCHK, REORG, RUNSTATS & REBIND
- discuss the strategies for BACK-UP & RECOVERY
- understand the processes & types of Logging

Data Management

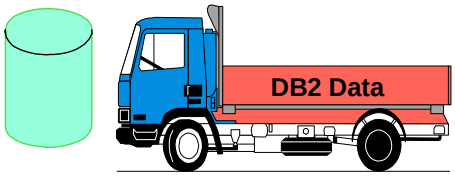
Moving Your Data

Moving Data

- Import/Export
- Load Utility
  - No Logging
  - Uses Parallel I/O
  - Uses CPU parallelism
  - Suspends Constraint Checking
- Types of Data
  - Non-Delimited/Delimited ASCII
  - PC/IXF
  - WSF (Import/Export Only)



Three Phases of Load



1. **LOAD** data into tables
   Collects index keys / sort
   Records consistency points
   Places Invalid data rows/messages in file

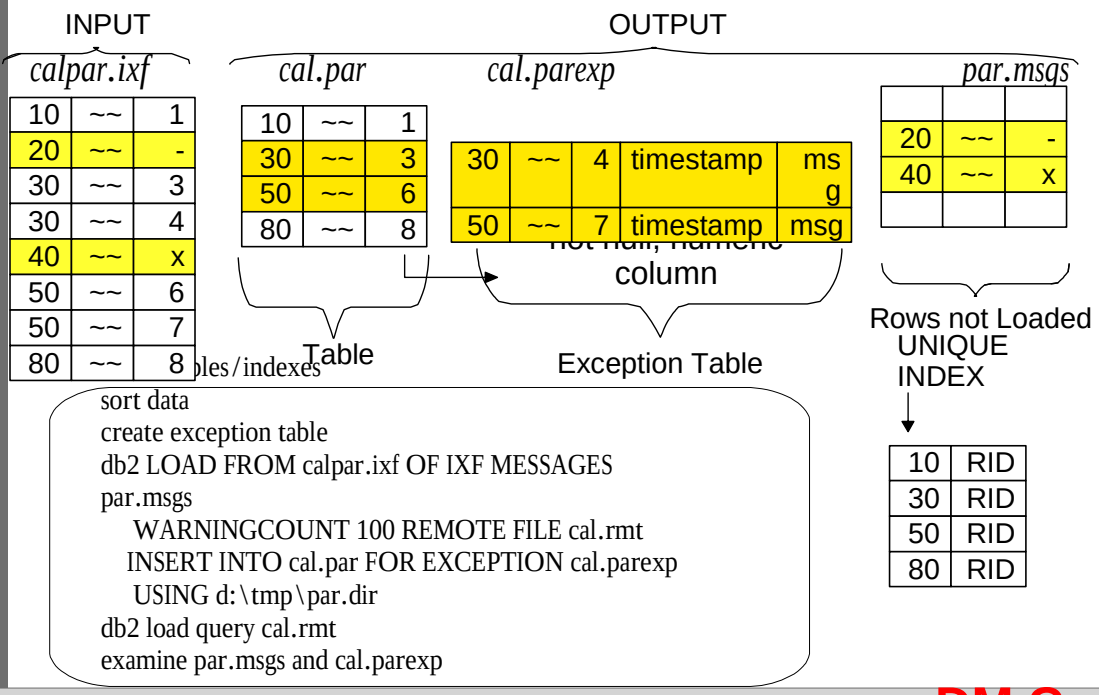
2. **BUILD**
   Indexes Created
   Unique Key Violations placed in Exception Table

3. **DELETE**
   Deletes Unique Key Violation Rows

Creating Target & Exception Tables

- Target Table must exist prior to LOAD
- Use modified DDL for the Exception Table
  - Drop indexes
  - Drop constraints
  - Alter DDL to add timestamp and message columns

LOAD Scenario



What happens if a LOAD Fails

- An SQLCODE and a short explanation are returned
- Check Messages Files
- Optionally use LOAD QUERY command
- Check the DB2DIAG.LOG file

Recovering From LOAD Failure

- **Restart the Load**
- **Check Messages Files**
- **Use Restart with RESTARTCOUNT Option**

1. LOAD Phase

- RESTARTCOUNT n
- (Restarts at n + 1;
- n is Last SAVECOUNT Point)

- Only Person doing the
- LOAD can Access Table
- if LOAD Pending State

2. BUILD Phase

- RESTARTCOUNT B

3. DELETE Phase

- RESTARTCOUNT D
- Removes "Delete Pending State"

Recovering From LOAD Failure (2)

- Create Backup Copy of Existing Table space before Loading Data
- To Return to Prior State Before Load
  - Attempt LOAD with RESTART Option
  - LOAD with TERMINATE Option to Change to Recovery Pending State
  - Restore Table space Backup Image created before LOAD
  - Roll Forward to End of Logs

LOAD Performance Considerations

- Best Performance in Replace Mode
- Use Parallelism if SMP machine
- Creating Index During Load Reduces Performance
  - Especially When Adding Data to Table
- COPY YES Reduces Performance of load, but overall performance may improve
- Use COPY NO in "Read Only" Environment
- Savecount
  - Converted to Page Count
  - Rounded Up to Intervals of EXTENTSIZE
- Nonrecoverable does not force backup or copy

SET CONSTRAINTS

SYSCAT.TABLES

TABNAME	DEFINER	STATUS	CONST_CHECKED
par	cal	C	N ...
for	cal	C	N ...

BEFORE

cal.par

10		1
30		3
50		6
80		8

Primary Key

cal.for

10		X
10		Y
20		Y
50		Z
80		X
90		A

Foreign Key

AFTER

cal.for

10		X
10		Y
50		Z
80		X

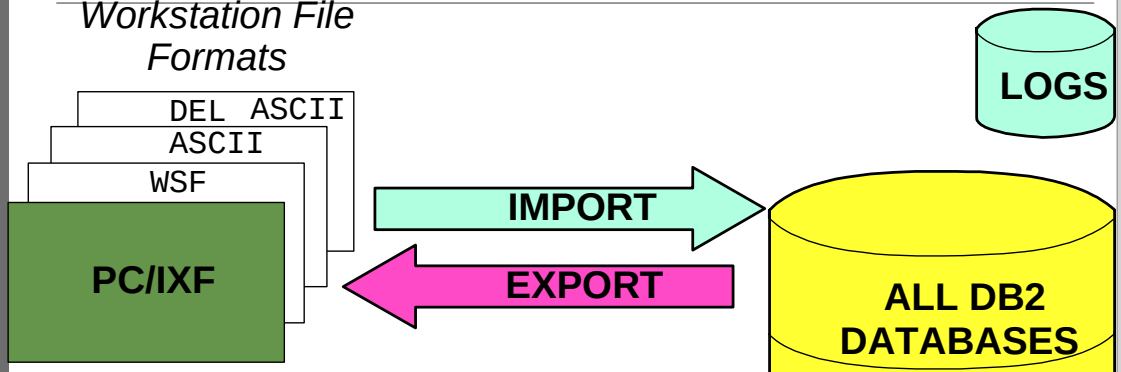
forexp

20		Y	timestamp	msg
90		A	timestamp	msg

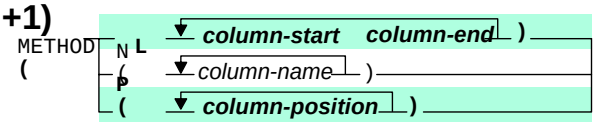
db2 SET CONSTRAINTS for cal.par, cal.for IMMEDIATE CHECKED
FOR EXCEPTION
IN cal.par USE parexp,
IN cal.for USE forexp

Data Exchange

Workstation File Formats



- **Must be connected prior to call**
- **IMPORT COMMITCOUNT n keeps log sizes manageable**
- **IMPORT RESTARTCOUNT n allows to restart from last commit (n+1)**

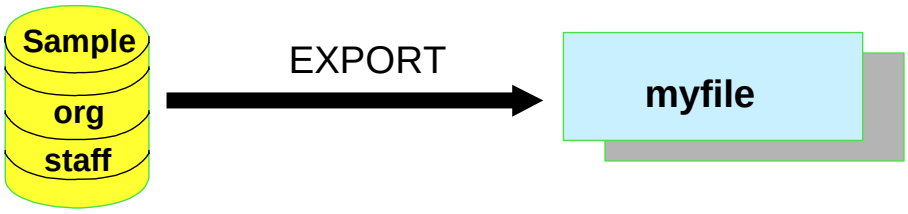


EXPORT uses only

N

Export Utility

```
db2  connect  to  sample
db2  export  to  myfile
     of ixf messages msg
     select staff.name, staff.dept, org.location
          from org,staff
          where org.deptname=staff.dept
```



- Exports data from database table(s) to file.
- Check msg for error or warning messges.
- Only PC/IXF file format supported by DDCS.
- Must have SYSADM, DBADM, CONTROL, or SELECT privilege on table(s).

Export Considerations

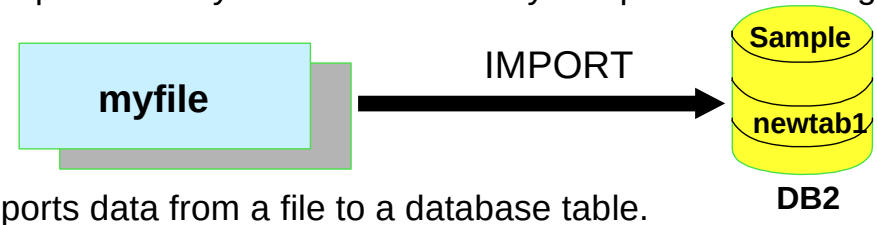
- The following information is required when exporting data:-
  - A SELECT statement specifying data
  - Path & Name of export file
  - Format of the export file
  - A message file (OPTIONAL but RECOMMENDED)
- Optional information
  - New column names for IXF/WSF
  - File type modifier for DEL/WSF

DM CofC

Import Utility

```
db2 connect to sample
db2 import from myfile of ixf
  messages msg [ create | insert | insert_update | replace | replace_create ] into newtab1
.....
```

db2 import from myfile of ixf modified by compound 5 messages ...



- Imports data from a file to a database table.
- Check msg for error or warning messges.
- Only PC/IXF file format supported by DDCS.
- Must have SYSADM, DBADM, or underlying privileges (SELECT, INSERT, CONTROL, or CREATETAB) on newtab1.

DM CofC

Import Considerations

- The following information is required when importing data:-
  - Path & Name of import file
  - Name or Alias of the table or view
  - Format of import file
  - A message file (OPTIONAL but RECOMMENDED)
- Optional information
  - Commitcount
  - Restartcount
  - Column names

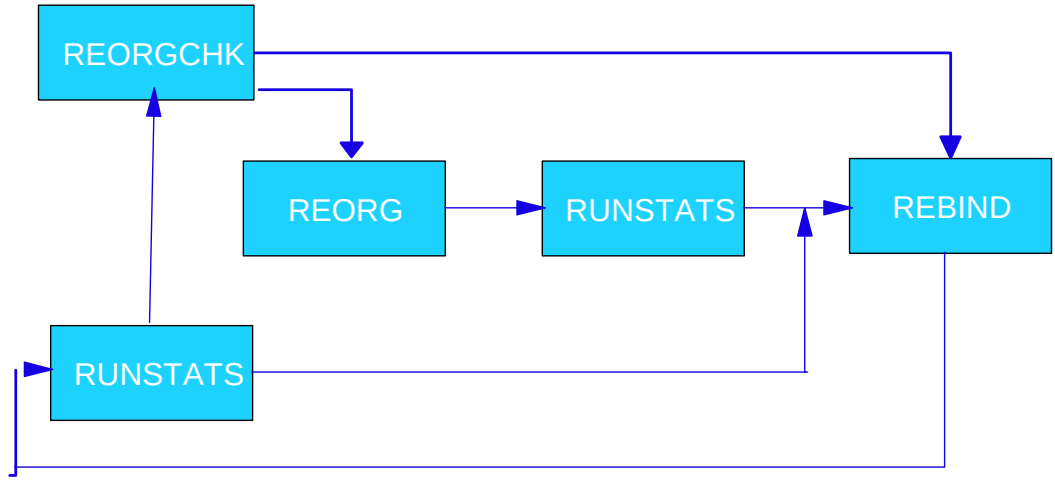
IMPORT vs. LOAD

IMPORT	LOAD
Slower on large amounts of data	Faster on large loads - writes formatted pages
Creation of tables & indexes with IXF format	Tables and indexes must exist
WSF supported	WSF not suppoted
Import into tables and views (Aliases supported)	Load tables only(Aliases supported)
Table space(s) On-line during import	Table space(s) Off-line during load
All rows logged	Minimal logging performed
Triggers will be fired	Triggers not supported
If interrupted table is usable with data upto the last commit point.	If interrupted the table is held in LOAD PENDING state. Either restart or restore tables effected.
Temporary space used within the database. Largest index plus 10%(approx).	Temporary space used outside the database. Sum of all indexes(approx).
Constraints validated during import	All Unique key is verified during load. (SET CONSTRAINTS)
Index keys are inserted individual	Index built after load.
Run RUNSTATS after import for Statistics	Statistics gathered during Load.
Import into host via DDCS	Cannot load into host.
Files must reside on the same node as the import	Files/Pipes must reside on the database node
No back-up image required	Backup can be created during load.

Data Management

Maintaining Your Data

Data Maintenance Process



REORGCHK Utility

```
REORGCHK ON TABLE db2.candidate
Table statistics:
F1: 100*OVERFLOW / CARD < 5
F2: 100*TSIZE / ((FPAGES-1) * 4020) > 70
F3: 100*NPAGES / FPAGES > 80
CREATOR     NAME          CARD   OV   NP   FP   TSIZE   F1   F2   F3   REORG
DB2         CANDIDATE        3    0    1    1      27    0    -  100  - - -

Index statistics:
F4: CLUSTERRATIO or normalised CLUSTERFACTOR > 80
F5: 100*(KEYS*(ISIZE+10)+(CARD-KEYS)*4) / (NLEAF*4096) > 50
F6: 90*(4000 / (ISIZE+10)**(NLEVELS-2))*4096 / (KEYS*(ISIZE+10)+(CARD-KEYS)*4)
<100
CREATOR   NAME                    CARD   LEAF  LVLS ISIZE   KEYS  F4   F5   F6
REORG
Table: DB2.CANDIDATE
SYSIBM    SQL960226095048000       3      1     1     9      3    100  -    -
```


DM CofC

REORG Utility

- REORG TABLE db2.candidate
  - Must use qualified table name or alias
  - If an index is specified you must use a fully qualified name
  - Use temporary table space if the table is very large
  - If REORG fails ***DO NOT DELETE*** the temporary files

DM CofC

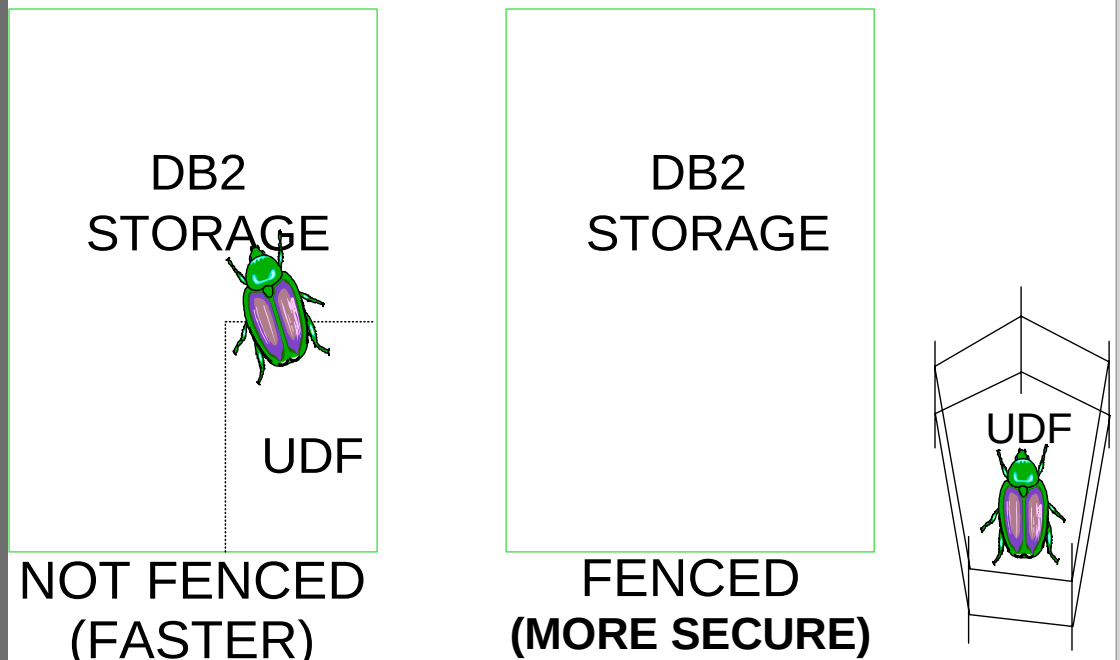
RUNSTATS Utility

- RUNSTATS ON TABLE db2.candidate
  - Must use qualified table name or alias
  - You can specify table and/or indexes
  - If requested distribution statistics can be collected based on db cfg parameters
  - SHRLEVEL can be set
    - *Change - other users can read & write to table*
    - *Reference - other users have read-only access*

REBIND Utility

- REBIND PACKAGE db2cert
  - Must use qualified package name or it will assume the current authorization ID.
  - Does not automatically commit unless auto-commit is enabled
  - Provides a quick way to recreate a package
  - db2rbind tool will rebind all packages

User-Defined Functions - Fencing



NOT FENCED
(FASTER)

FENCED
(MORE SECURE)

DM CofC

Application Management

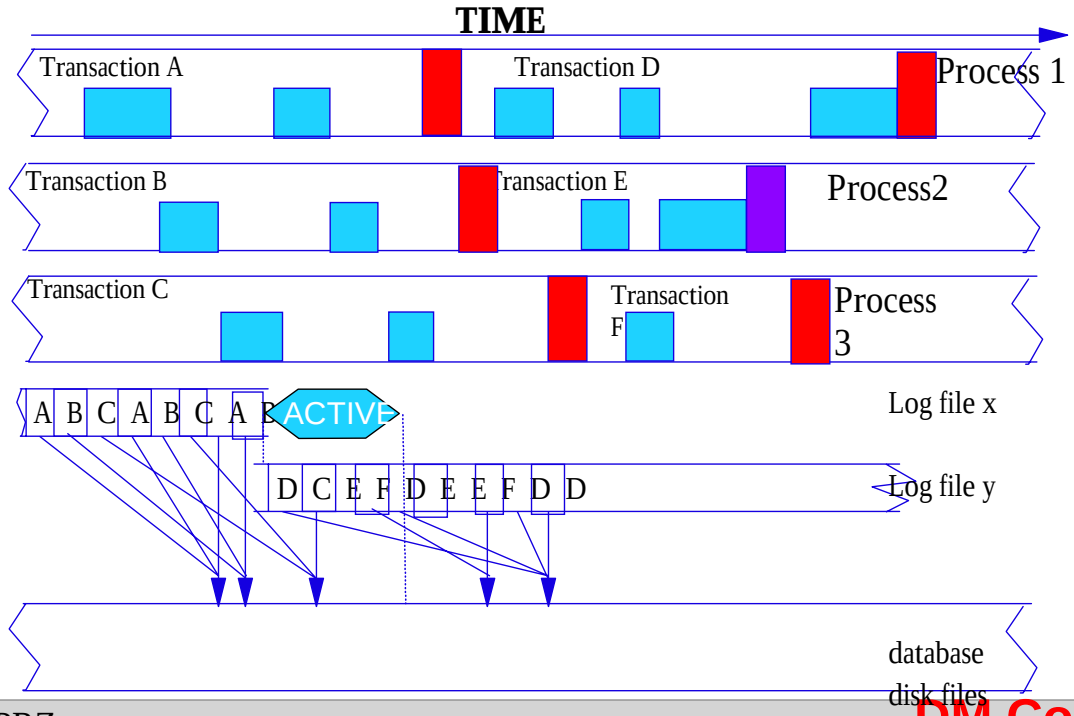
```
db2 list applications

Auth Id  Application Name  Agent Id  Application Id                DB
Name
DB2      db2bp_32          9878      *LOCAL.db2.960426161832
DB2CERT
DB2      db2bp_32          27134     *LOCAL.db2.960426161005
DB2CERT

db2 "force application  ( 9878 ) "
DB20000I The FORCE APPLICATION command completed successfully.
DB21024I This command is asynchronous and may not be effective immediately.

Auth Id  Application Name  Agent Id  Application Id                DB
Name
DB2      db2bp_32          27134     *LOCAL.db2.960426161005
DB2CERT
```


DM CofC

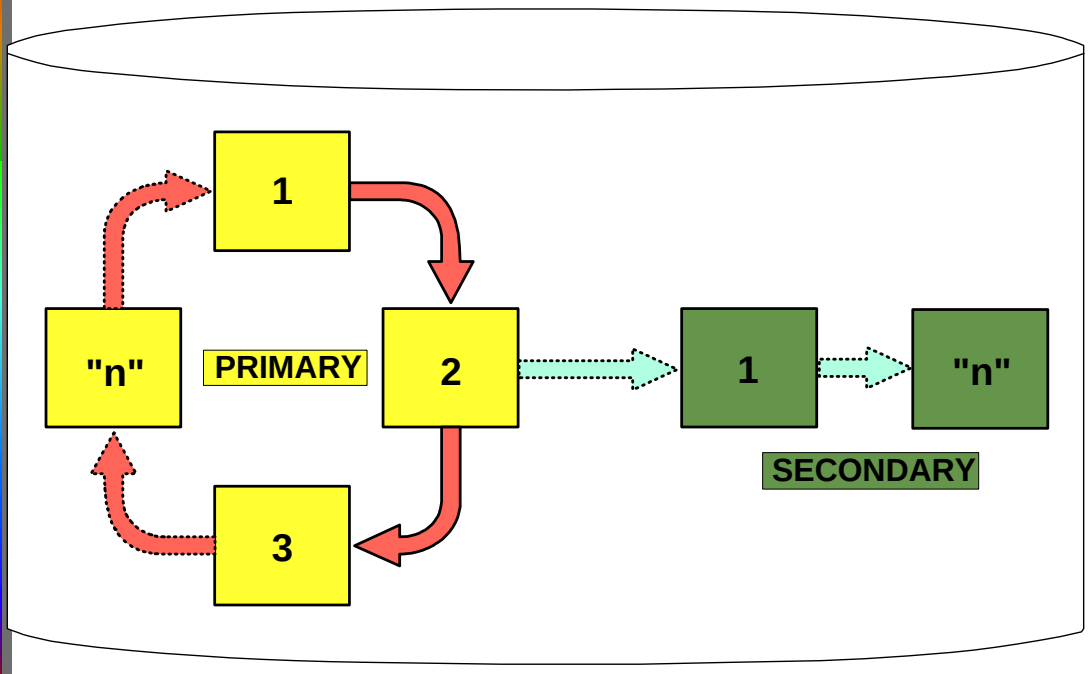
Data Management

Back-up & Recovery

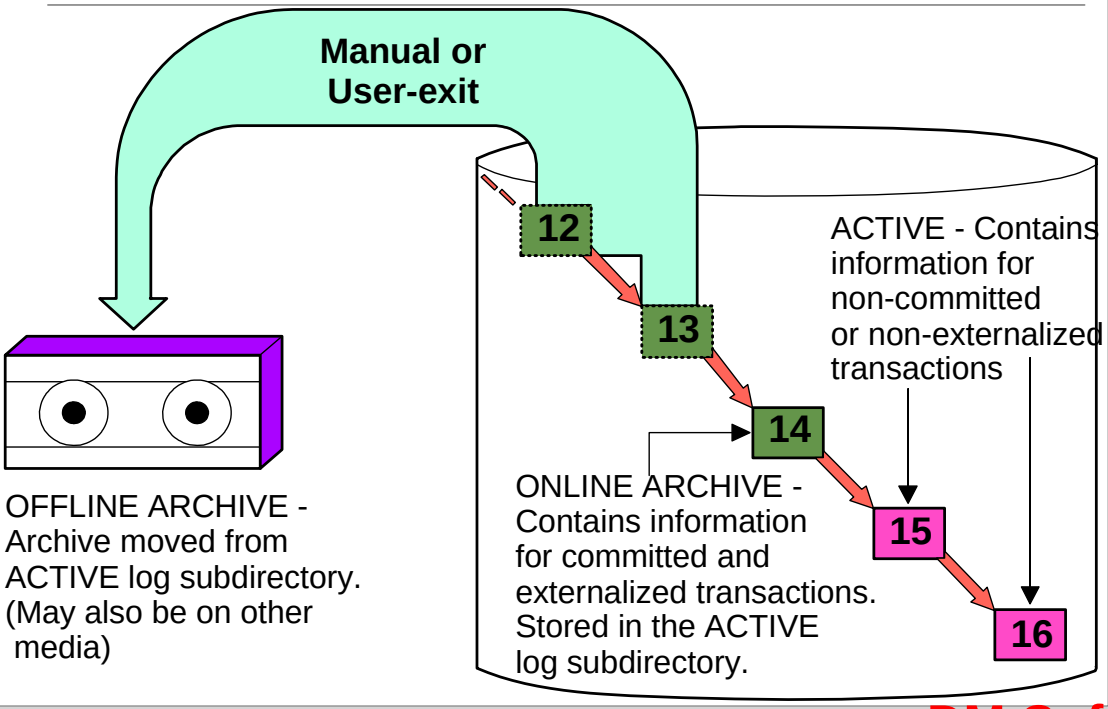
Transaction Log File Use



Circular Logging



Archival Logging/Log Retain

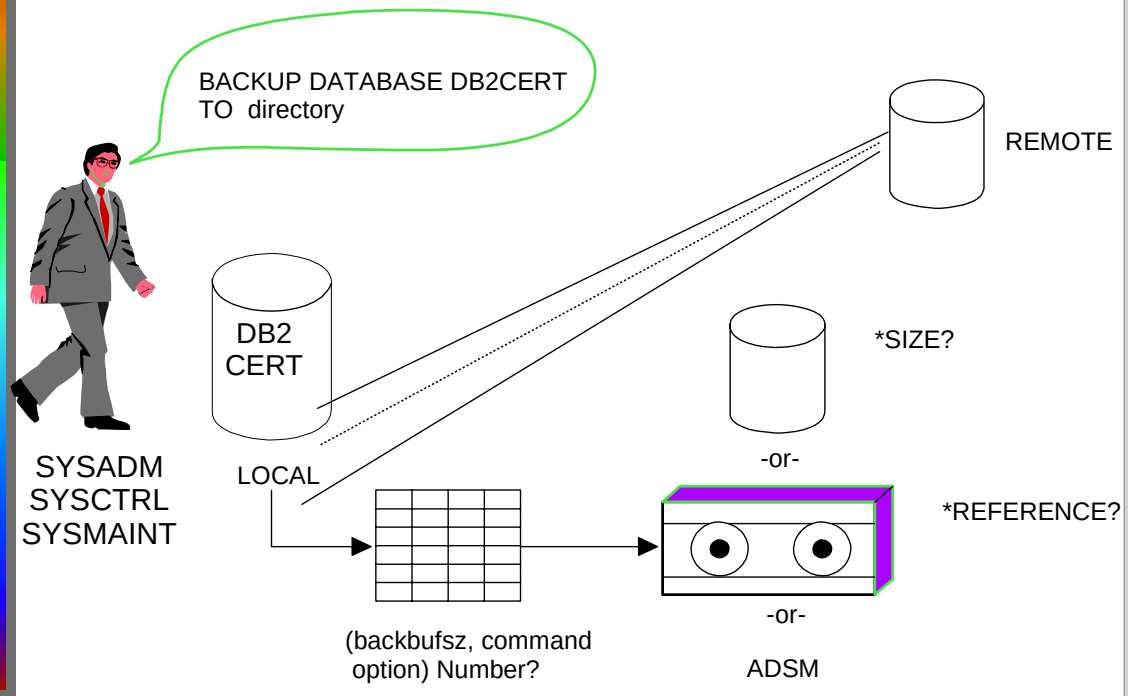


Log File Usage

- Rollback
  - Uses log files to terminate a unit of work and back out any changes
- Crash Recovery
  - Consists of 2 phases
    - Reapply all transactions (regardless of commit)
    - Rollback changes NOT committed
  - DB config parameter **AUTORESTART** (default ON)
- Roll Forward Recovery
  - Command can be applied to DB or Table space
    - Min PIT for tablespaces

DM CofC

Backup Utility



DM CofC

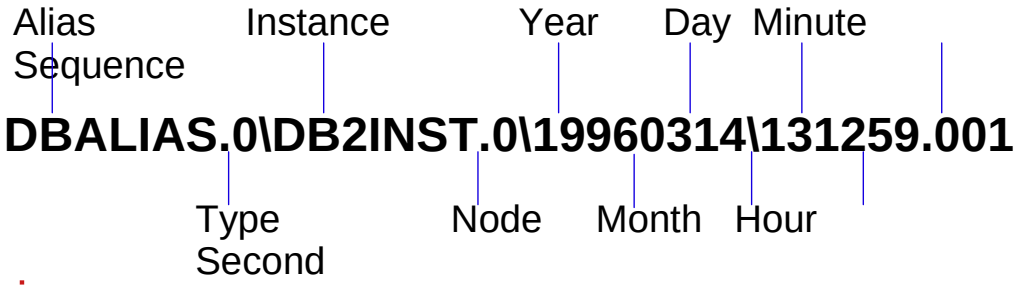
The Backup Files

- File name for images on disk or diskette contains:
  - database alias
  - type of backup (0=FULL, 3=TS, 4=Copy from LOAD)
  - instance name
  - database node (always 0for non EEE)
  - timestamp of backup
  - sequence number
- Exact naming convention varies slightly by platform.
- Tape images are not named, but contain the same information in the backup header for verification purposes.
- Backup History provides key information in easy to use format.

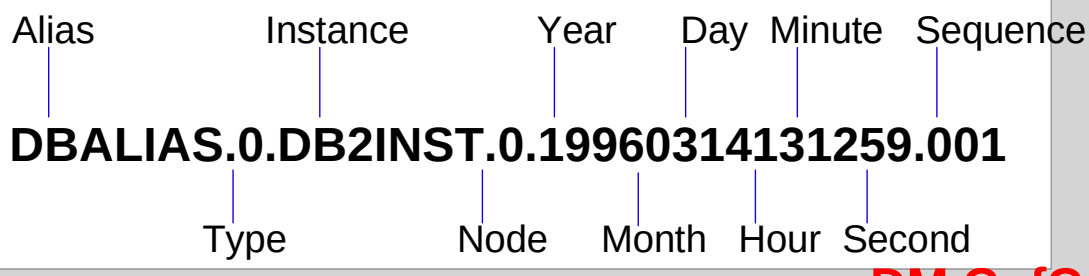
DM CofC

The Backup Files

Intel

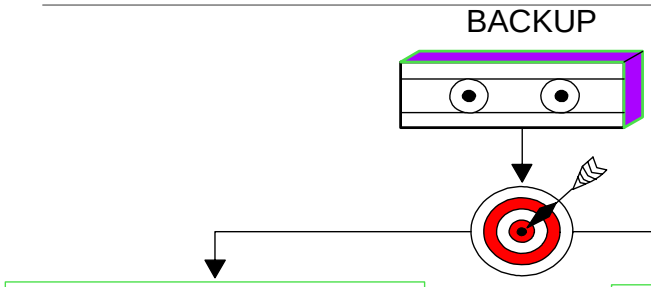


Unix



DM CofC

Restoring a Backup Image



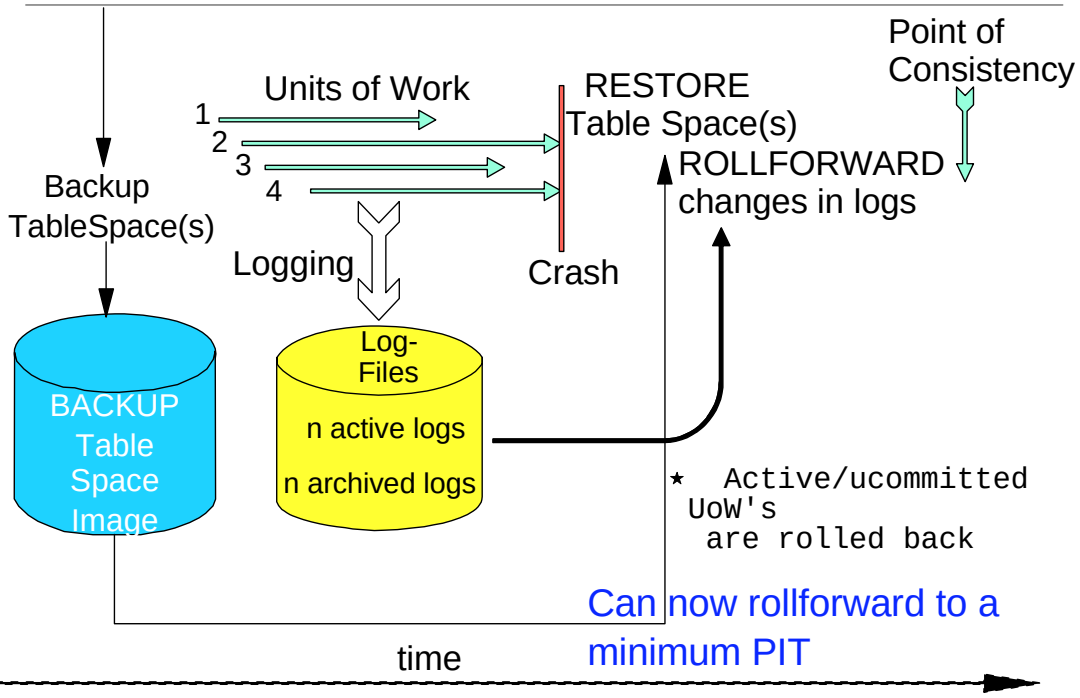
EXISTS

- DELETE TABLE, INDEX, LONG FIELD FILES
- RETAIN AUTHENTICATION
- RETAIN DATABASE DIRECTORIES
- REPLACE TABLE SPACE ENTRIES
- RETAIN HISTORY
- CHECK DATABASE SEED *

NEW

- CREATE NEW DATABASE
- RESTORE AUTHENTICATION
- RESTORE DATABASE CONFIGURATION FILE
- SET DEFAULT LOG PATH
- RESTORE COMMENTS

Table Space Recovery



Min Tablespace PIT Recovery

- Tablespaces can now be rolled forward to a PIT
- There is a minimum PIT for each tablespace
- The min PIT will initially be when the backup occurred
- May be increased by:
  - Changes which cause system catalog updates
    - Alter table
    - Create index
    - tablespace definiton change

Backup & Recovery Considerations

	Full Database Back-up off-line	Full Database Back-up off-line	Full Database Back-up on-line	Table Space Back-up off-line	Table Space Back-up on-line
Logging Type	Archival	Circular	Archival	Archival	Archival
Access allowed during Back-up	N/A	N/A	Full	None	Full
Database state after restore	Rollforward Pending	Consistent	Rollforward Pending	TS in Rollforward Pending	TS in Rollforward Pending
Rollforward Required	Any Point in Time after backup	N/A	Any Point in Time past Back-up	Min PIT	Min PIT

Backup History File

Column Name	Type	Description
OPERATION	Char(1)	B=backup, R=restore, L=load
OBJECT	Char(1)	D=Full, P=Table Space, T=Table
OBJECT_PART	Char(17)	yyyymmddhhmmssnnn (nnn=001 for restore/load)
OPTYPE	Char(1)	F=off-line, N=on-line, R=Load Replace,A=Load Append,C=Load Copy otherwise blank
DEVICE_TYPE	Char(1)	D=Disk, K=Diskette, T=Tape, A=ADSM, U=Userexit, O=other vendor device support
FIRST_LOG	Char(12)	First Log file id
LAST_LOG	Char(12)	Latest Log file id
BACKUP_ID	Char(14)	yyyymmddhhmmss
SCHEMA	Char(8)	Qualifier for Loaded table
TABLE_NAME	Char(18)	Name of Loaded tale
NUM_TABLESPACES	Char(3)	Number of Table Spaces
LOCATION	Char(255)	Dependent on Device_Type
COMMENT	Char(30)	Free-form Text

Data Management

Graphical Management Tools

Graphical Tools (Control Center) Includes

- Script Center- Write / Execute / schedule script
- Journal - Monitor, Alerts, Messages, Recovery History
- Replication - Setup Replication Subscriptions
- Smartguides for normal DB operations
  - backup
  - restore
  - load
  - etc.

Unit Summary

- After completing this unit, you should be able to:
- **talk about the methods of populating tables**
  - **LOAD, IMPORT & EXPORT**
- **discuss the advantages and disadvantages of these methods**
- **utilise the DB2 tools to maintain your tables**
  - **REORGCHK, REORG, RUNSTATS & REBIND**
- **discuss the strategies for BACK-UP & RECOVERY**
- **understand the processes & types of Logging**