IBM Software Group

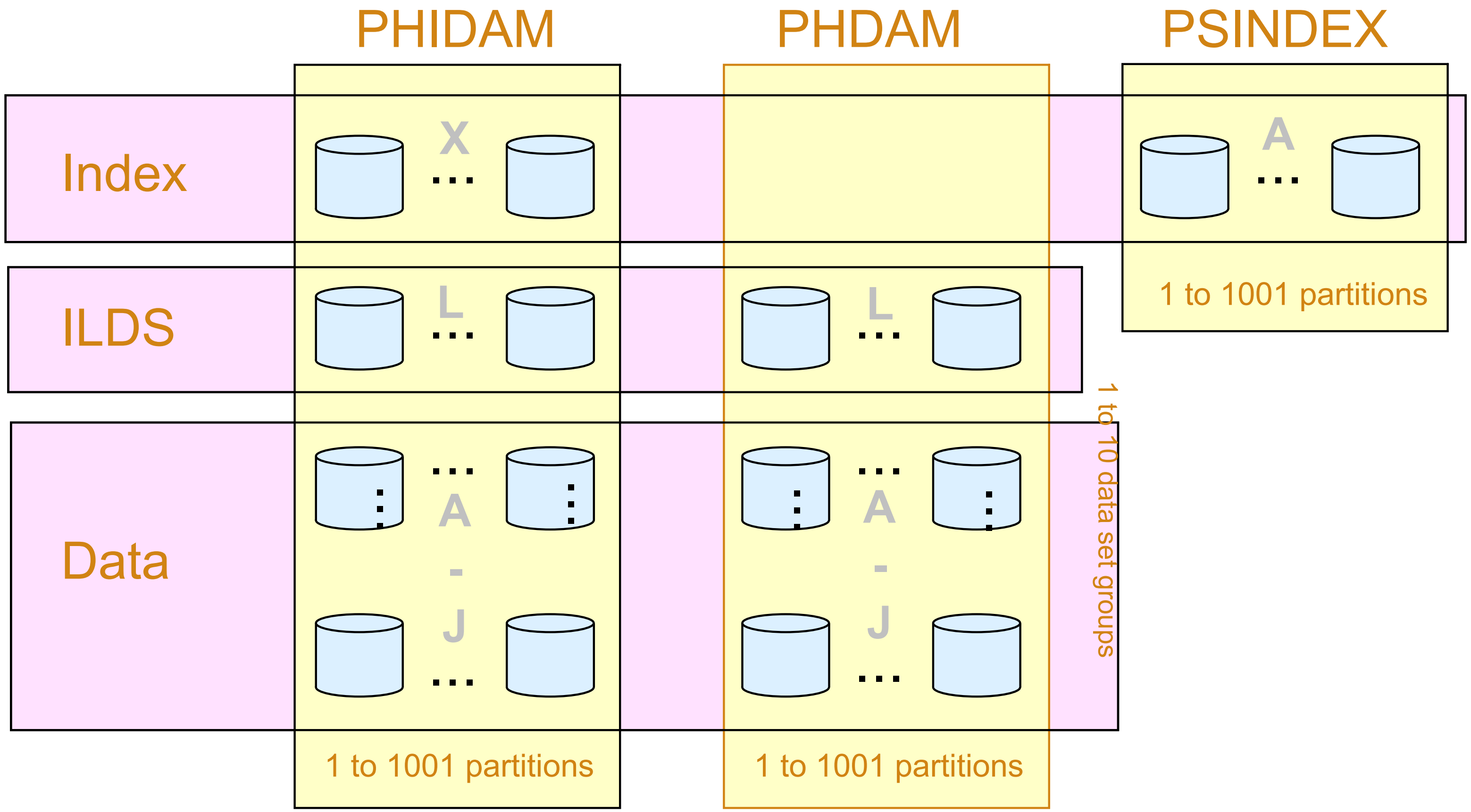
IMS13
IMS HALDB Administration

Cornelia Hallmen
Rich Lewis

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

HALDB Database Data Sets

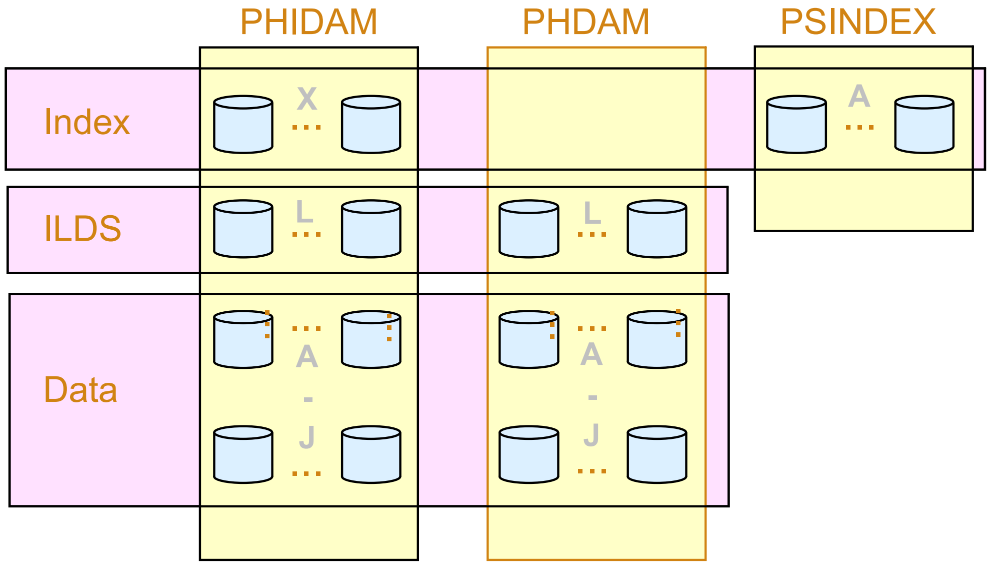


The data sets in a partition have generated data set names and DDNAMEs.
Letters are used to distinguish them.

X - PHIDAM index
L - ILDS
A through J - Data data sets
A - PSINDEX

HALDB Database Data Sets

- Each PHDAM or PHIDAM partition **requires an ILDS (L)**
  - ILDS is empty if there are no logical relationships or secondary index entries
- Each PHIDAM partition has an index data set (X)
- Each PHDAM or PHIDAM partition has an A data set
  - Root segments are in the A data sets
- Each PHDAM or PHIDAM partition may have B-J data sets
  - Used for multiple data set groups
- Each PSINDEX partition has an A data set



Partition Names and IDs

- Each partition has a name
  - Unique in the RECONs
    - Partitions in different databases cannot have the same name
    - Partitions cannot have the same name as a database
  - Choices:
    - Name signifies the data in the partition
      - Could cause problems when partitions are modified
    - Name is arbitrarily chosen
- Each partition has an ID
  - Number assigned by IMS when partition is created
    - Assigned in creation order
    - Not in key sequence
    - Not reused

HALDB Database Data Sets

- Data set names
  - Begin with data set name prefix for the partition
    - Up to 37 characters assigned by user
  - Letter and Partition ID are used as suffix
    - X for PHIDAM index
    - L for ILDS
    - A for PSINDEX
    - A through J for data
  - Example:
    - Partition data set name prefix IMP0.DB.INV23
    - Partition ID: 00004
    - Data set names:
      - PHIDAM index: IMP0.DB.INV23.X00004
      - PHIDAM ILDS: IMP0.DB.INV23.L00004
      - PHIDAM first data data set: IMP0.DB.INV23.A00004

- Each partition in a database may have the same data set name prefix.
  - Partition IDs make names unique.

HALDB DDNAMEs

- DDNAMEs
  - Begin with the partition name
    - Up to 7 characters assigned by user
  - Letter is used as suffix
    - X for PHIDAM index
    - L for ILDS
    - A for PSINDEX
    - A through J for data
  - Example:
    - Partition name LBAD112
    - DDNAME for PHIDAM Index: LBAD112X
    - DDNAME for PHIDAM ILDS: LBAD112L
    - DDNAME for first 'data' data set: LBAD112A

Dynamic Allocation

- Dynamic allocation uses RECON information
  - All HALDB databases are registered in RECONs
  - DFSMDA members are never used for HALDB
- If you use a DD statement:
  - If DD statement conflicts with RECON information, allocation fails
  - If DD statement matches RECON information, allocation succeeds
    - It works as if you had not used a DD statement
- THEREFORE, do not include DD statements for HALDB

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Partition Initialization

- Partition initialization
  - Prepares partition data sets for use
  - Ensures that partitions with no data are usable
  - Initialization is done either by
    - HALDB Partition Data Set Initialization utility (DFSUPNT0) or
    - Database Prereorganization utility (DFSURPR0)
  - Database is specified to the utility
    - Partitions with 'partition initialization required' DBRC flag (PINIT) are initialized
      - Exception: DFSUPNT0 has unconditional partition initialization function
        - Invokes initialization for all partitions in the database with or without flag set
        - Specified with INITALL control statement in DFSOVRDS DD data set
        - Introduced by PQ49638 (IMS V7) and PQ55002 (IMS V8)

Partition Initialization

- Partition initialization process
  - Makes high-used RBA non-zero
    - Writes and erases a record in PSINDEX
  - Writes reorg number and partition ID in PHDAM and PHIDAM
  - Creates first bit map block in PHDAM and PHIDAM
  - Writes high-key (x'FF...FF') record in PHIDAM

Partition Initialization

- Partition initialization is only required in three cases:
  1. Before initial load (PROCOPT=L) of partition
  2. Before migration reload of partition
     - Input to reload was created by unload of non-HALDB database with MIGRATE=YES or MIGRATX=YES option
  3. Before a partition may be used without containing any data
     - Initial load or reload does not insert any segments in the partition
- Partition initialization is not required with reorganizations
  - Not required even when data sets are deleted and redefined
    - Unless the partition is empty
- 'Partition Initialization Required' flag in RECONs
  - Turned 'on' by partition definition or DBRC command
  - Turned 'off' by partition initialization or DBRC command
  - Authorization fails if the flag is 'on'

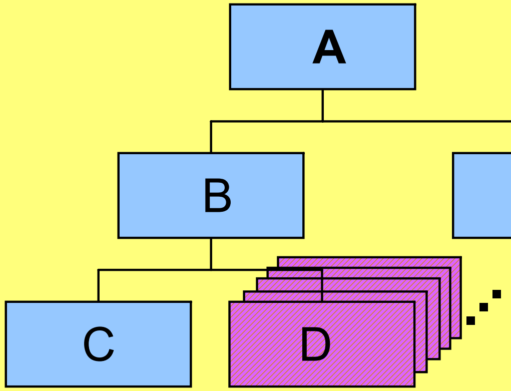
Number of Partitions and Their Sizes

- Things to consider when choosing the number of partitions
  - Number of partitions affects the sizes of partitions
  - Time required to reorg partitions in parallel
    - Smaller partitions shorten the process
  - Time required to image copy and recover partition data sets
    - Smaller partitions shorten these processes
  - Smaller partitions may avoid multivolume data sets
  - Management of the data sets
  - More data sets require more management
- Multiple data set groups
    - May be advantageous to have only one data set per partition
    - May be advantageous to have multiple data sets

Multiple Data Set Groups

- HALDB supports multiple data set groups
  - Multiple data set groups place different segment types in different data sets
  - Should you use them?
- Multiple data set groups were used for two reasons with non-HALDB
  1. Avoid data set size limitations - Not required with HALDB
  2. Place infrequently used segments in another data set - Also applies to HALDB

- Example:
  - Place D segments in a data set group
  - Increases likelihood that E will be in the same block with A



Database Compression

- HALDB supports segment edit/compression routines
  - Should you use them for compression?
- Reasons to use compression with HALDB:
  - Saves DASD space
  - May improve performance
    - Reduces I/Os required to retrieve and write data
- Reasons not to use compression with HALDB:
  - Not needed to avoid data set size limitation
  - May hurt performance
    - CPU costs for compression and expansion of segments
      - Probably not significant

Adding, Deleting, and Changing Partitions

- Databases change over time
  - The sizes of partitions may change over time
    - Data added and deleted
  - The high keys of partitions may need to be adjusted over time
    - Different amounts of data added or deleted to different partitions
      - Example: Root keys based on date
- Databases need to be adjusted over time
  - Partitions may need to be split, consolidated, created, or deleted
  - Partition boundaries (high keys) may need to be adjusted

HALDB Migration Aid Utility

- HALDB Migration Aid utility (DFSMAID0)
  - Reads HDAM, HIDAM, Secondary Index databases
    - Provides sizing and high key information for migration planning
    - Secondary index support
      - Provides key range boundaries and numbers of records
        - Secondary index 'bytes' and 'prefix-incr' information are inaccurate in the report!
        - Number of segments and high key values are accurate in the report
        - Sizes are easily calculated from the numbers of records
  - Reads PHDAM, PHIDAM, and PSINDEX databases
    - Provides sizing and high key information for repartitioning planning

HALDB Migration Aid Utility

- Sample report:

```
 partition 1 :

    minimum key =

       +0000  d2c1c1f1 f1f2f3f4    |KAA11234 |

    maximum key =

       +0000  d2f2f3f9 f9f2f3f4    |K2399234 |

                      segments          bytes   prefix-incr   length-incr
    1) 'PRODUCT '        31567        4040576        252536             0
    2) 'INVENT'         103781        8094918        830248             0
    3) 'ORDQTY'         171182       10955648       1369456             0
    4) 'MFGSPECS'        51115       10938610        408920             0
  SUM)                  357645       34029752       2861160             0
```

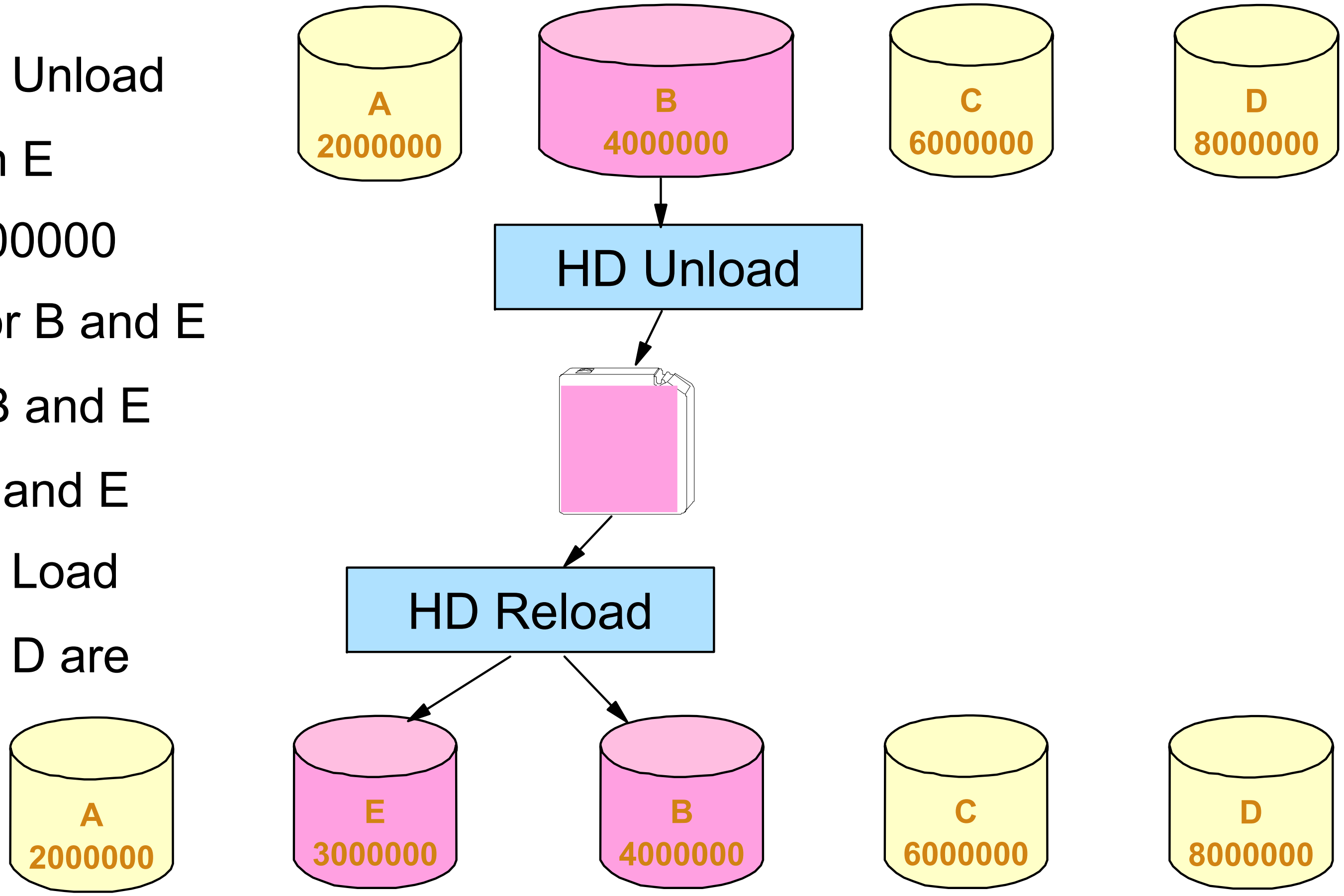
segments - **number of segments**
bytes - **number of bytes for the segments**
prefix-incr - **additional bytes due to increased prefix size**
length-incr - **additional bytes required for paired logical relationships**

HALDB Migration Aid Utility

- Using the Migration Aid utility
  - You may specify one of the following:
    - Number of equal sized partitions
    - Number of segment bytes per partition
    - High keys for partitions
  - Report for each partition and the entire database
  - Bytes in reports do not include free space, bit maps, RAPs, or FSEAPs
    - You must adjust for these!

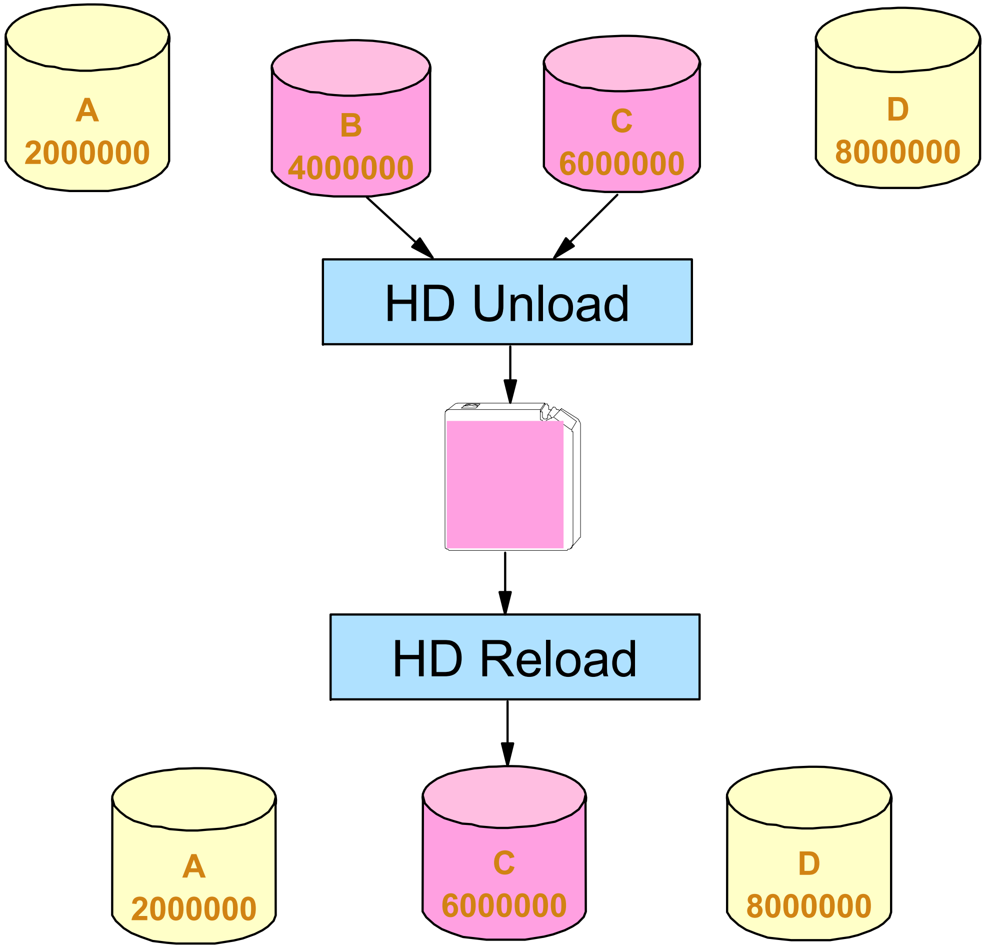
Splitting a Partition

- If partition B with high key 4000000 needs to be split
  - Unload partition B
    - HD Unload or HP Unload
  - Define new partition E
    - With high key 3000000
    - Sets PINIT flag for B and E
  - Initialize partitions B and E
  - Reload partitions B and E
    - HD Reload or HP Load
  - Partitions A, C, and D are not affected



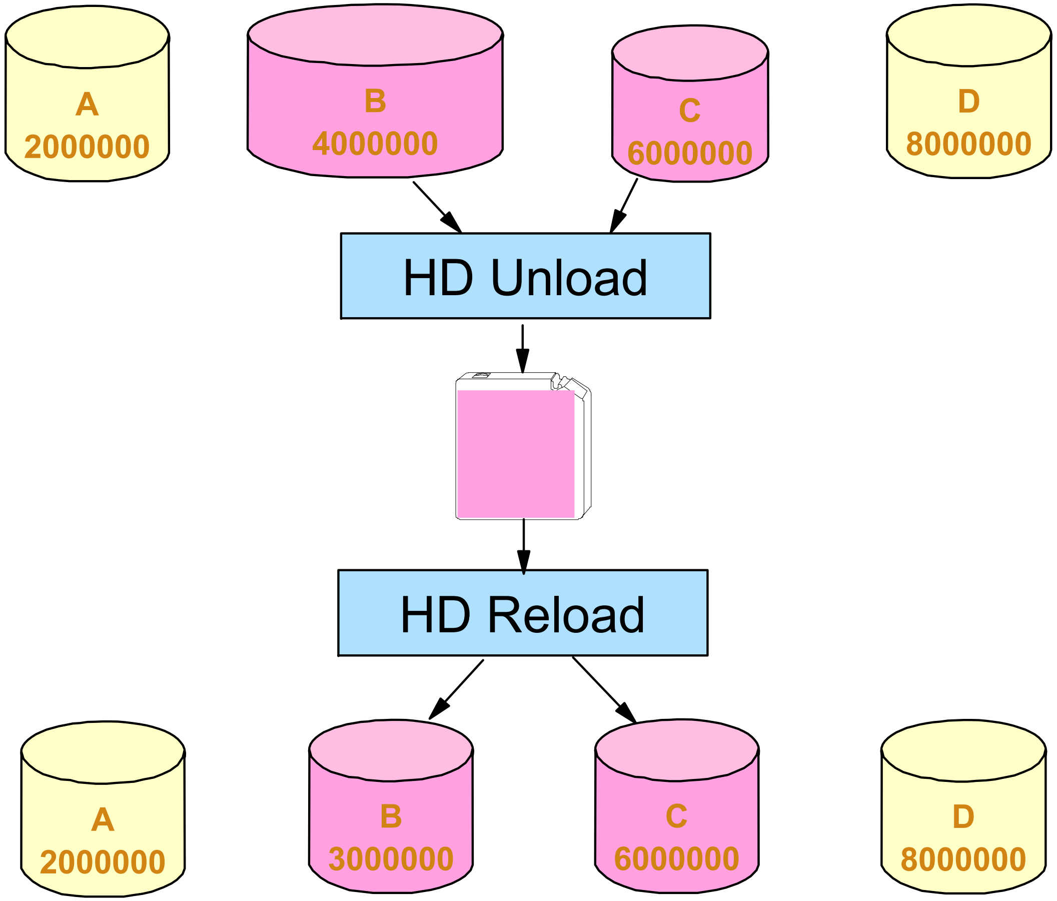
Combining Partitions

- If partitions B and C with high keys 4000000 and 6000000 need to be combined
  - Unload partitions B and C
    - HD Unload or HP Unload
  - Delete definition of partition B
    - Sets PINIT flag for C
  - Initialize partition C
  - Reload partition C
    - HD Reload or HP Load
  - Partitions A and D are not affected



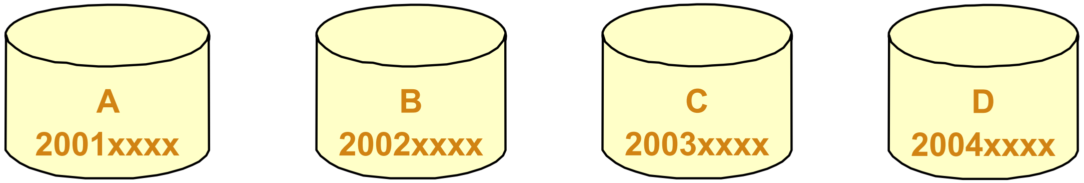
Modifying Partition Boundaries

- If records need to be moved from partition B to C
  - Unload partitions B and C
    - HD Unload or HP Unload
  - Change high key for partition B
    - From 4000000 to 3000000
    - Sets PINIT flag for B and C
  - Initialize partitions B and C
  - Reload partitions B and C
    - HD Reload or HP Load
  - Partitions A and D are not affected

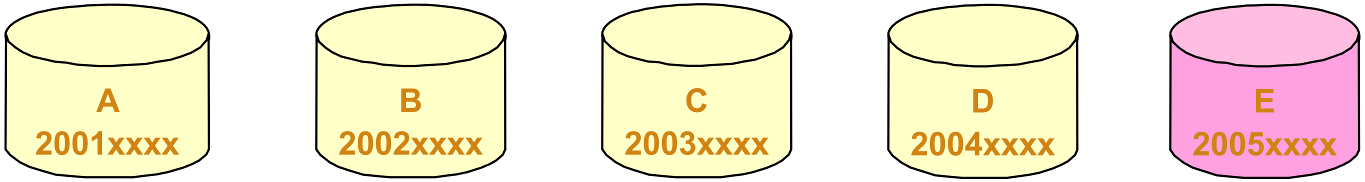


Databases with Dates for Keys

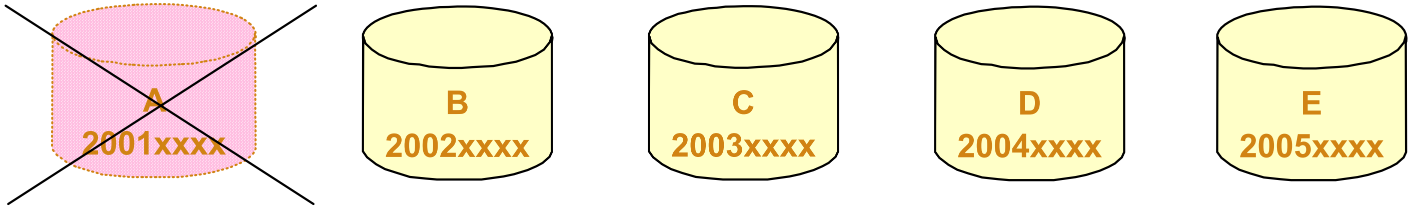
- Some databases have dates as the high-order part of the key



- ► To add a partition for a set of dates (higher keys)
  - Define it and initialize it



- ► To delete the partition with the lowest dates (keys) and all of its data
  - Delete the partition definition



- ► Unloads and reloads are not required for these changes

Disabling and Enabling Partitions

- Disabling and enabling of partitions was introduced by APARs
  - PQ48421 for IMS V7
  - PQ73858 for IMS V8
- Disabling partitions
  - Definitions and information remain in RECONs
    - Includes partition IDs, DSN prefixes, and recovery information
  - Partitions are not used
    - Partitions are ignored
- Disabled partitions may be enabled
  - Enabled partitions are made active
  - Enabled partitions are marked 'recovery needed'

Enabling and Disabling Partitions

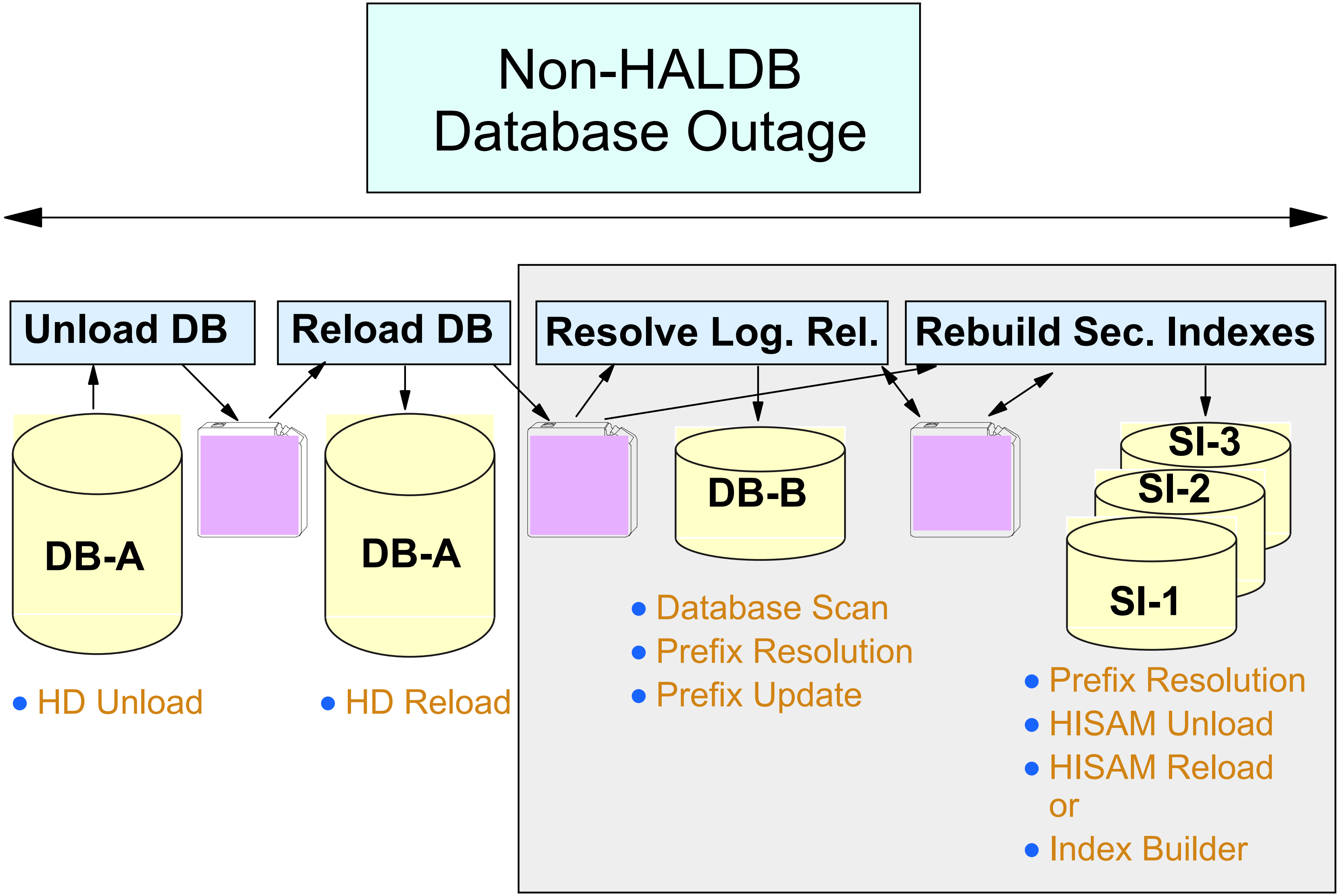
- Use of disabling and enabling of partitions
  - Disabling is normally done prior to deleting a partition
    - Keeps recovery information, partition ID, DSN prefix, etc.
  - If testing is successful, partition is deleted
    - Deletion removes all information
  - If testing is not successful, partition is enabled
    - Partition is recovered and becomes active
      - Other partitions may require timestamp recovery
- PDU support for disabling and enabling
  - New 'Partition status' field on 'Change Partition' panel
- DBRC commands for disabling and enabling

```
CHANGE.PART DBD(dname) PART(pname) DISABLE
CHANGE.PART DBD(dname) PART(pname) ENABLE
```

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Non-HALDB Reorganizations



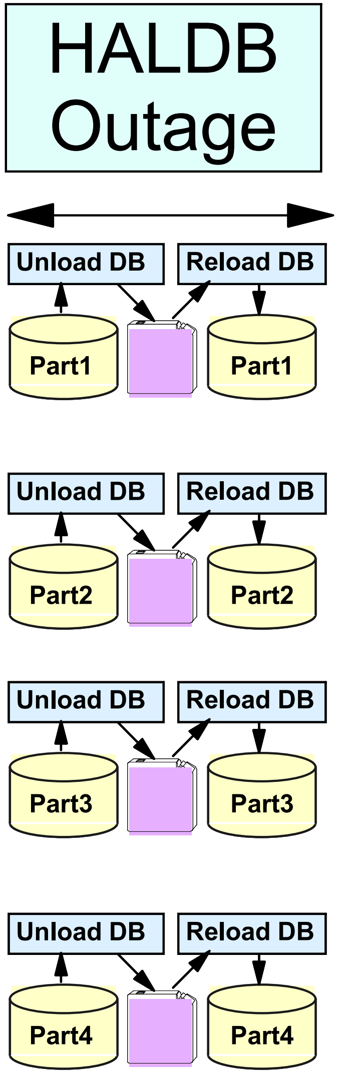
- HD Unload

- HD Reload

- Database Scan
- Prefix Resolution
- Prefix Update

- Prefix Resolution
- HISAM Unload
- HISAM Reload
  or
- Index Builder

HALDB Reorganizations



- **Shorten the reorg time to <u>your window</u>**
- **Reorg partitions in parallel**
  - Create enough partitions to meet your requirement
- **Eliminate rebuilds of secondary indexes**
  - Prefix Resolution, HISAM Unload, HISAM Reload, or Index Builder are not required
- **Eliminate updates to logical relationships**
  - DB Scan, Prefix Resolution, and Prefix Update are not required

Healing Pointers After Reorgs

- After a reorganization sec. index and log. rel. pointers are "broken"
  - Normal processing heals them efficiently
    - Only heals pointers that are used
    - Reads of pointers are "free"
      - They are being read for normal processing
    - ILDS reads are efficient
      - ILDS CIs hold many entries
      - ILDS CIs are maintained in the buffer pools
  - Optionally, you can heal them
    - Extends the reorganization process
    - Typically, uses more resources
      - Heals all pointers
      - More total I/Os
    - HALDB Conversion and Maintenance Aid includes pointer healing utility
- Our recommendation: Let normal processing heal the pointers

Data Set Delete and Define for Reorgs

- HALDB database data sets may be reused
  - Delete and redefine are not required for reorganization
    - VSAM REUSE attribute is honored by HD Reload
      - Non-HALDB VSAM required DELETE and DEFINE
    - OSAM allows reuse with both HALDB and non-HALDB
  - Delete and define are required to move data sets
- REUSE attribute is required for HALDB VSAM data sets
  - Except ILDS
    - Parameter is allowed but not honored for ILDS
      - ILDS will not be reused by Index/ILDS Rebuild utility (DFSPREC0)

Partition Initialization During Reorgs

- Partition initialization is not required during reorganizations
  - Data sets may be deleted and redefined without partition initialization
    - Exception: A partition which contains no data must be initialized

- Reorganization steps:
  - Unload partition
  - Delete partition data sets (optional)
  - Define partition data sets (optional)
  - Reload partition

Reorganizations and Secondary Indexes

- Reorganization of a HALDB database does not require rebuild of its secondary indexes
  - Self-healing pointer scheme eliminates this requirement

Good!

- Many installations never reorganize non-HALDB secondary indexes
  - They are rebuilt (and organized) with every reorg of the indexed databases
- HALDB secondary indexes may become disorganized
  - They may require reorganization

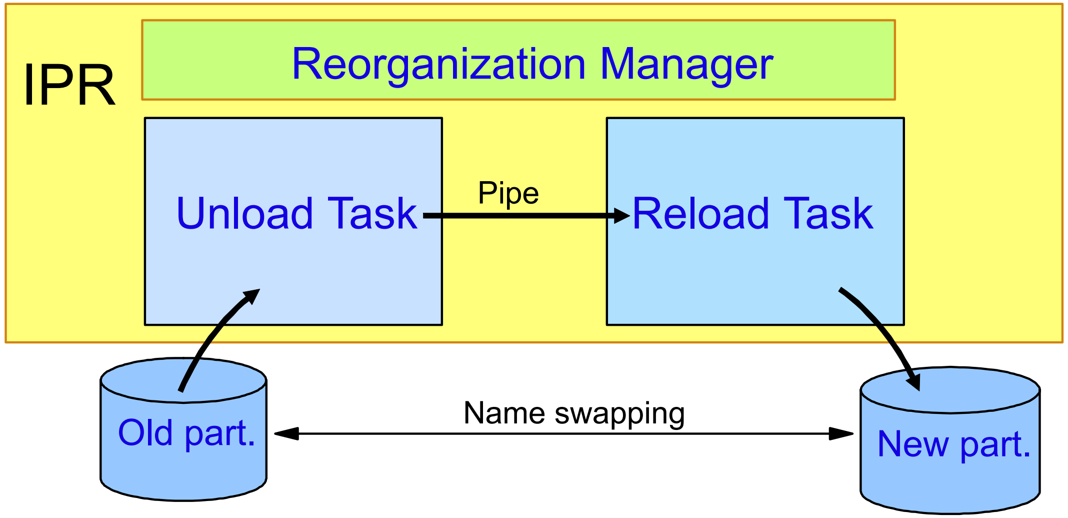
This is a change in procedures!

Reorganization Alternatives

- HD Unload and HD Reload
  - Utilities provided with IMS
  - PHDAM, PHIDAM, and PSINDEX
    - Also non-HALDB support
  - Partitions must be off-line
- High Performance Unload and High Performance Load
  - Tools from IBM
  - PHDAM, PHIDAM, and PSINDEX
    - Also non-HALDB support
  - Partitions must be off-line

Reorganization Alternatives

- IMS Parallel Reorganization (IPR)
  - Single job step reorganization
  - Uses HP Unload and HP Load
  - Unload and load are done in parallel
    - Segment is read for unload and immediately passed to load
    - Reorg time is about equal to the slower process



Reorganization Alternatives

- IMS V9 Online Reorganization (OLR)
  - Utility provided with IMS V9
  - PHDAM and PHIDAM
  - Absolutely no outage
    - Data is available throughout the reorg process
  - Supports: data sharing, XRF, logical relationships, secondary indexes, ...

Eliminating the Need for Reorgs

- Free space
  - Rule of thumb of 20% free space is 25+ years old
    - Developed when DASD was very expensive
    - Developed when the nightly window was 12 hours
    - Out of date?
  - HALDB allows you to have as much free space as you need (and can afford)
    - DASD space is cheap
    - Reorganizations are expensive
  - More free space could eliminate the need for some reorganizations!

Reorganization Summary

- Size partitions to meet reorganization window needs
  - Reorganize partitions in parallel
  - Largest partition determines reorganization time
- Pointer healing
  - Typically, normal processing heals pointers most efficiently
- Database data sets may be reused
  - VSAM REUSE attribute is honored
- Partition initialization is not required for reorganizations
  - Even when data sets are redefined
- Secondary indexes are not rebuilt
  - May need to be reorganized

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Backup and Recovery

- HALDB A-J data sets (not the ILDS or PHIDAM index)
  - Backup
    - Image Copy utility (DFSUDMP0)
      - Including CIC option
    - Image Copy 2 utility (DFSUDMT0)
  - Updates are logged
    - Change Accum may be used
  - Recovery
    - Database Recovery utility (DFSURDB0)
    - Online Recovery Service (ORS) tool
    - Database Recovery Facility (DRF) tool
  - DBRC
    - GENJCL.IC
    - GENJCL.CA
    - GENJCL.RECOV

Like other IMS database data sets

Backup and Recovery

- HALDB ILDS (L) and PHIDAM Index (X) data sets
  - Backup
    - No image copies
  - Updates are not logged
    - ILDS is only updated by reorganization reload
    - PHIDAM Index is treated like a non-recoverable database
  - Recovery
    - Index/ILDS Rebuild utility (DFSPREC0)
      - Rebuilds the data set(s) from the database
  - DBRC
    - GENJCL.USER MEMBER(DSPUPJCL)
      - May be used to generate DFSPREC0 JCL to rebuild an ILDS or PHIDAM index

Timestamp Recoveries

- All data sets of a partition must be recovered to the same time
  - PHIDAM index must be rebuilt
    - A data set must be recovered first
    - Rebuild with Index/ILDS Rebuild utility (DFSPREC0)
  - ILDS may need to be rebuilt
    1. If secondary indexes or logical relationships are used and
    2. If recovery is to time before last reorganization
       - ILDS is only changed by reorganizations
    - May be rebuilt with Index/ILDS Rebuild utility (DFSPREC0)
  - Alternative for ILDS
    - After reorganization
      - Copy ILDS with REPRO
    - If ILDS needs to be restored
      - Use copy produced by REPRO

Timestamp Recoveries

- Must all partitions of a database be recovered to the same time?
  - Almost always
  - User must understand when this is not required
    - For example, offending program updated only one partition
- Secondary index implications
  - Usually, database with secondary index forces recovery of all partitions to the same time
    - All partitions of the indexed database
    - All partitions of its secondary indexes
- Logical relationship implications
  - Usually, database with logical relationships forces recovery of all partitions to the same time
    - All partitions in the logically related databases

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Test Databases

- Non-HALDB test databases
  - Often, not registered in RECONs
  - Each programmer may have one or more versions of a database
- All HALDB databases are registered in RECONs
  - Multiple versions of a database must be defined in different RECONs
    - DBRC does not allow multiple databases with the same name
  - Multiple test versions of a database require multiple RECONs
  - Plan your batch test environments

Defining Test Databases

- Use the same DBD as production
  - DBD does not include partition or data set information
  - Place in test DBDLIB and ACBLIB
- Create test partition definitions
  - Define partitions for test environment

    or
  - Use Partition Definition Utility EXPORT and IMPORT functions
    - Moves partition definitions between RECONs
      - They may be modified after IMPORT
        - Data set name prefix, RAA, etc.
      - APARs PQ48421 (V7) and PQ73858 (V8) maintain partition IDs

Creating Test Databases

- Alternatives for creating a test database from a production database
  - Unload and Reload
    - HD Unload (HP Unload) of production
    - HD Load (HP Load) to test
      - You may create a different partition configuration
        - Partition IDs will generally be different
        - Partition names may be changed
        - Partition boundaries may be changed
  - Image Copy and restore
    - Export and import partition definitions
      - Maintains partition IDs (with APARs PQ48421 or PQ73858)
    - Image copy production database data sets and restore to test
      - Partition IDs are stored in database data sets
    - Change database data set names of test database
      - Change data set name prefixes
  - Use application programs

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Secondary Indexes

- Plan the partitions for the secondary indexes
  - How many partitions do you need?
    - Space requirements
      - HALDB secondary index entries are much larger than those for non-HALDB sec. ind.
        - Pointers are larger
        - Root key of target is stored in the entry
    - Reorganization requirements
  - Will they need to be adjusted during life of the database?
    - Keys based on date, etc.
- Plan to reorganize them
  - They are not rebuilt with each reorganization of their indexed databases
- Don't make them non-recoverable unless you have a tool to rebuild them (e.g. Index Builder)
  - They are not rebuilt by IMS utilities

Agenda

- Review of database data sets
- Partitions
  - Initialization
  - Sizing
  - Adding, deleting, and modifying partitions
- Reorganizations
- Recoveries
  - Timestamp recoveries
- Test databases
- Secondary indexes
  - Sizing, recoveries, and reorganizations
- Performance

Performance

- HALDB processing is tuned like other full function database processing
  - Buffer pools
    - ILDSs also use buffer pools
  - Reorganizations and free space
  - OSAM sequential buffering
  - PHDAM root addressable area (RAA) and RAPs
    - Make your RAA large enough to hold all of your data with free space
      - In each partition
    - Give yourself a lot more RAPs than roots
      - In each partition
- HALDB has some new options
  - Parallel processing of partitions

Assigning Data Sets to Buffer Pools

- HALDB database data sets may be assigned to separate buffer pools
  - DFSVSMxx member or DFSVSAMP data set

```
DBD=dbdname(data set identifier,id)
```

dbdname - partition name or master database name

data set identifier - Letter A-J, L, or X

A-J for user data sets

A for secondary index

L for Indirect List Data Set (ILDS)

X for PHIDAM primary index

id - subpool id

Parallel Processing of Partitions

- Parallel processing of partitions
  - Different jobs may process different partitions
  - Could shorten elapsed times
  - Control statement may be used to limit PCB access to one partition
    - Batch (DLI or DBB), BMP, or JBP region
    - DFSHALDB DD statement:

```
HALDB PCB=(nnn|dddddddd,ppppppp)

   nnn - DBPCB number
   dddddddd - DBPCB label or name
   ppppppp - partition name
```

HALDB Database Administration

- Partitioning
  - Sizing, naming, and modifying
- Reorganization
  - Parallel processing and alternatives
- Backup and recovery
  - Special considerations for ILDSs and PHIDAM indexes
- Secondary indexes
  - Partition sizing and reorganization requirements
- Redbook:
  - *The Complete IMS HALDB Guide: All You Need to Know to Manage HALDBs*, SG24-6945

Things to Remember

- HALDB Migration Aid utility can analyze existing HALDB databases
  - Useful when planning repartitioning
- Deleting a partition definition deletes its recovery information
  - Disabling a partition keeps its recovery information
- Secondary indexes may require reorganizations
  - They are not rebuilt when the indexed database is reorganized
- Secondary index cannot be rebuilt from database with IMS utilities
  - Don't make them non-recoverable unless you have a tool like the IBM Index Builder

Things to Remember

- PHIDAM indexes and ILDSs have a different recovery process
  - They are rebuilt with Index/ILDS Rebuild Utility (DFSPREC0)
- Plan your scheme for creating HALDB test databases
  - DBRC registration is required for all databases