



B80

Introduction to HALDB and Its Benefits

Rich Lewis

IMS
Technical Conference

Sept. 27-30, 2004

Orlando, FL

HALDB (High Availability Large Database)

- **Large Database**

- ▶ Databases are partitioned
 - Up to 1001 partitions per database
 - Partitions have up to 10 data set groups

Capacity

Performance

Greater than 40 terabytes!

- **High Availability Database**

- ▶ Partition independence
 - Allocation, authorization, and reorganization are by partition
- ▶ Self healing pointers
 - Reorganization of partition does not require changes to secondary indexes or logically related databases which point to it
 - Prefix Resolution, Prefix Update, and secondary index rebuilds are eliminated
- ▶ Much shorter offline reorganization times
- ▶ Online reorganization in IMS V9

Availability

Compatibility

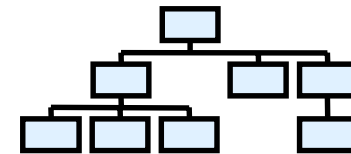
HALDB Highlights

- **Database types**

- ▶ PHDAM - partitioned HDAM
- ▶ PHIDAM - partitioned HIDAM
 - Index is also partitioned
- ▶ PSINDEX - partitioned secondary index

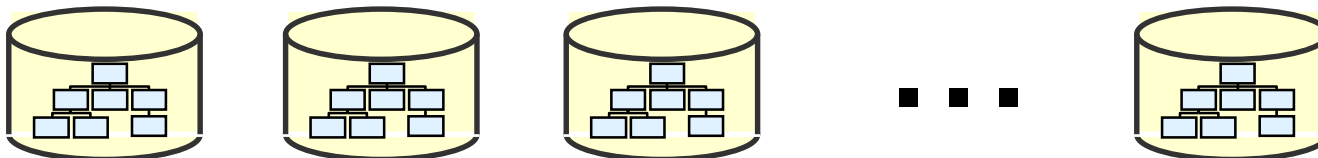
- **Hierarchic structure is maintained**

- ▶ A database record resides in one partition



- **Partition selection**

- ▶ By key range or by user exit routine



HALDB Highlights

- **Parallel Processing**
 - ▶ Reorganizations
 - Partitions may be reorganized independently
 - Partitions may be reorganized in parallel
 - ▶ Application processing
 - Partitions may be processed in parallel
 - DBRC authorization is by partition (not entire database)

HALDB Support

- HALDB is a generalized database including support for:
 - ▶ Logical relationships
 - ▶ Secondary indexes
 - ▶ OSAM and VSAM
 - ▶ Online and batch
 - ▶ Standard backup, recovery, and reorganization utilities
 - ▶ Data sharing
 - ▶ Remote Site Recovery (RSR)
 - ▶ Extended Recovery Facility (XRF)
 - ▶ Database Recovery Facility (DRF)
 - ▶ Online Recovery Service (ORS)
 - ▶ Online Change
 - ▶ OSAM Sequential Buffering
 - ▶ IMS Monitor and IMS Performance Analyzer
 - ▶ ...

- Any full function database may be migrated to HALDB

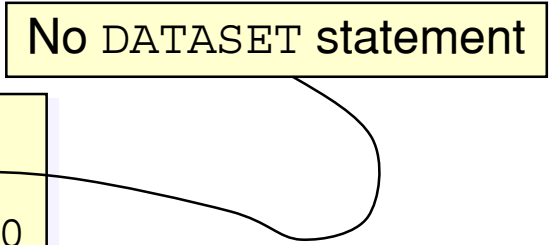
HALDB Highlights

- **DBD defines the database structure**
 - ▶ Does not define partitions or data sets

```
DBD    NAME=SKILLINV, ACCESS=PHIDAM

SEGM   NAME=SKILL, BYTES=32, PTR=NT, PARENT=0
FIELD  NAME=(TYP1, SEQ, U), BYTES=6, START=1
FIELD  NAME=STCODE, BYTES=8, START=7
SEGM   NAME=PER, BYTES=90, PARENT=SKILL
FIELD  NAME=(PERNO, SEQ, U), BYTES=8, START=1
DBDGEN
FINISH
END
```

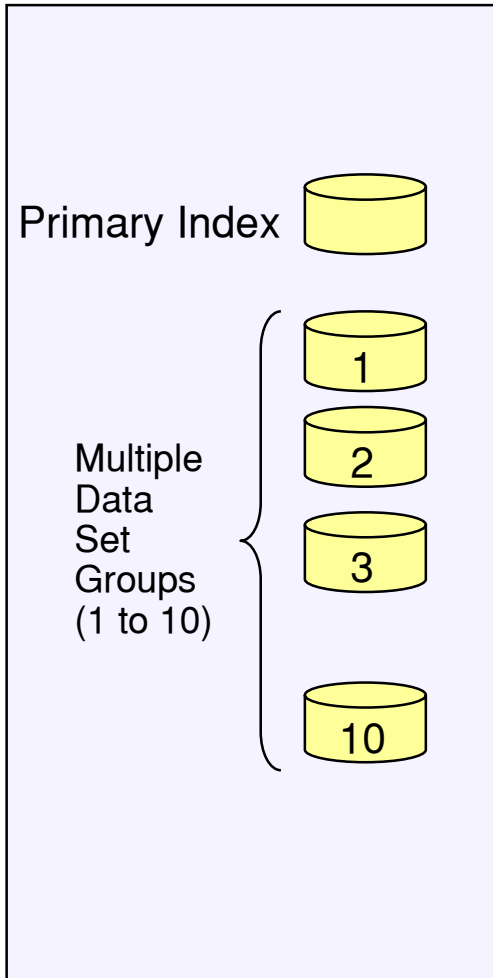
No DATASET statement



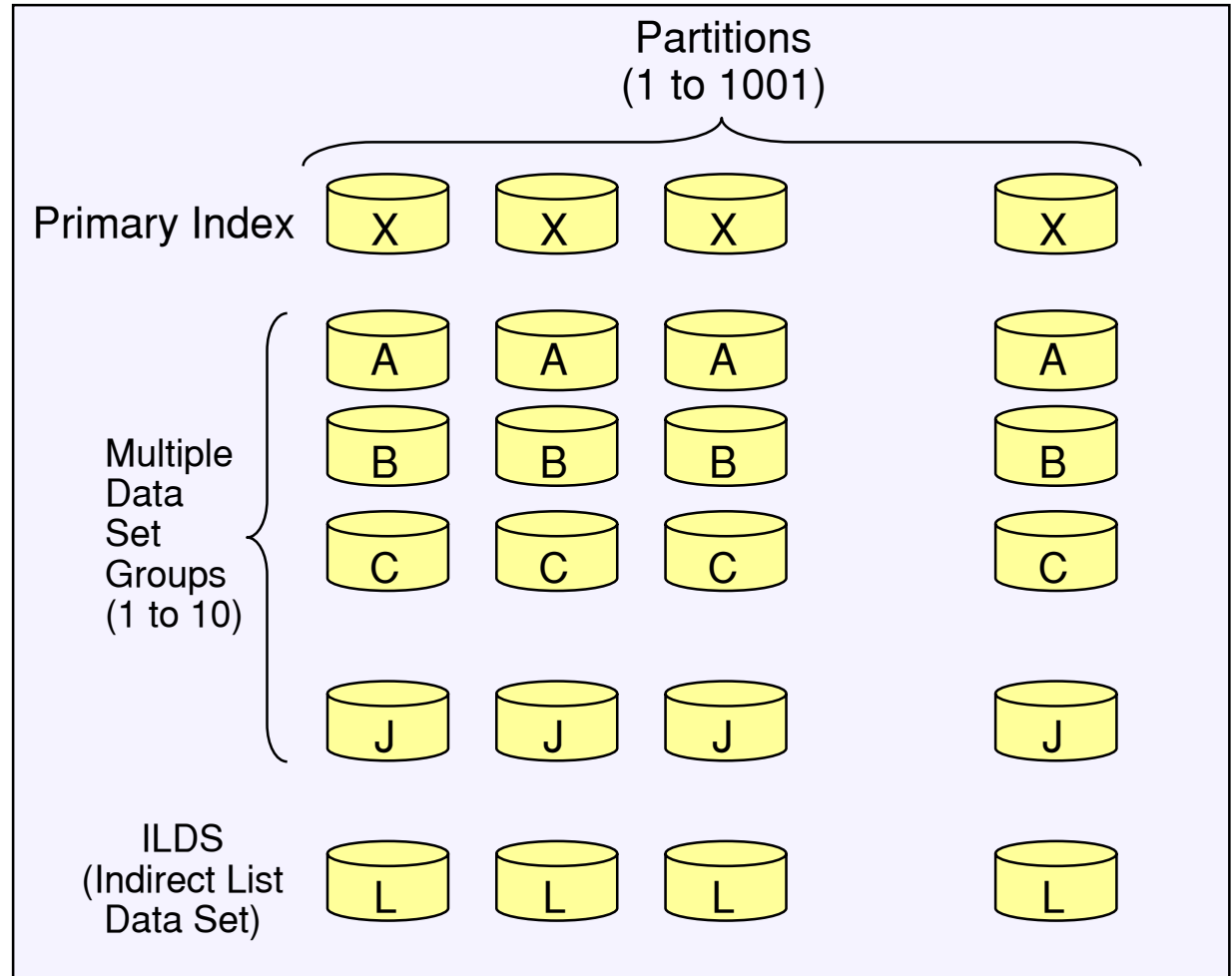
- **Partitions and data sets are defined in the RECONs**
 - ▶ INIT.PART command
 - or
 - ▶ Partition Definition Utility (ISPF)

HIDAM versus PHIDAM Databases

HIDAM Database
Data Sets

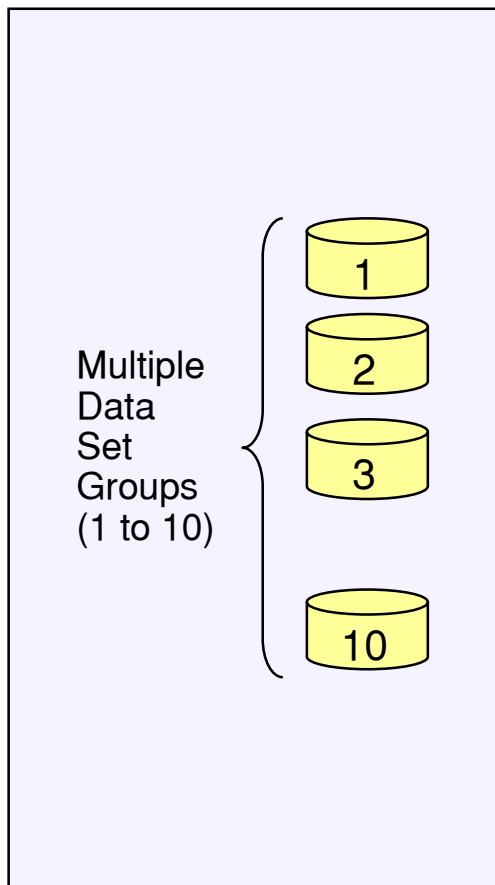


PHIDAM Database
Data Sets

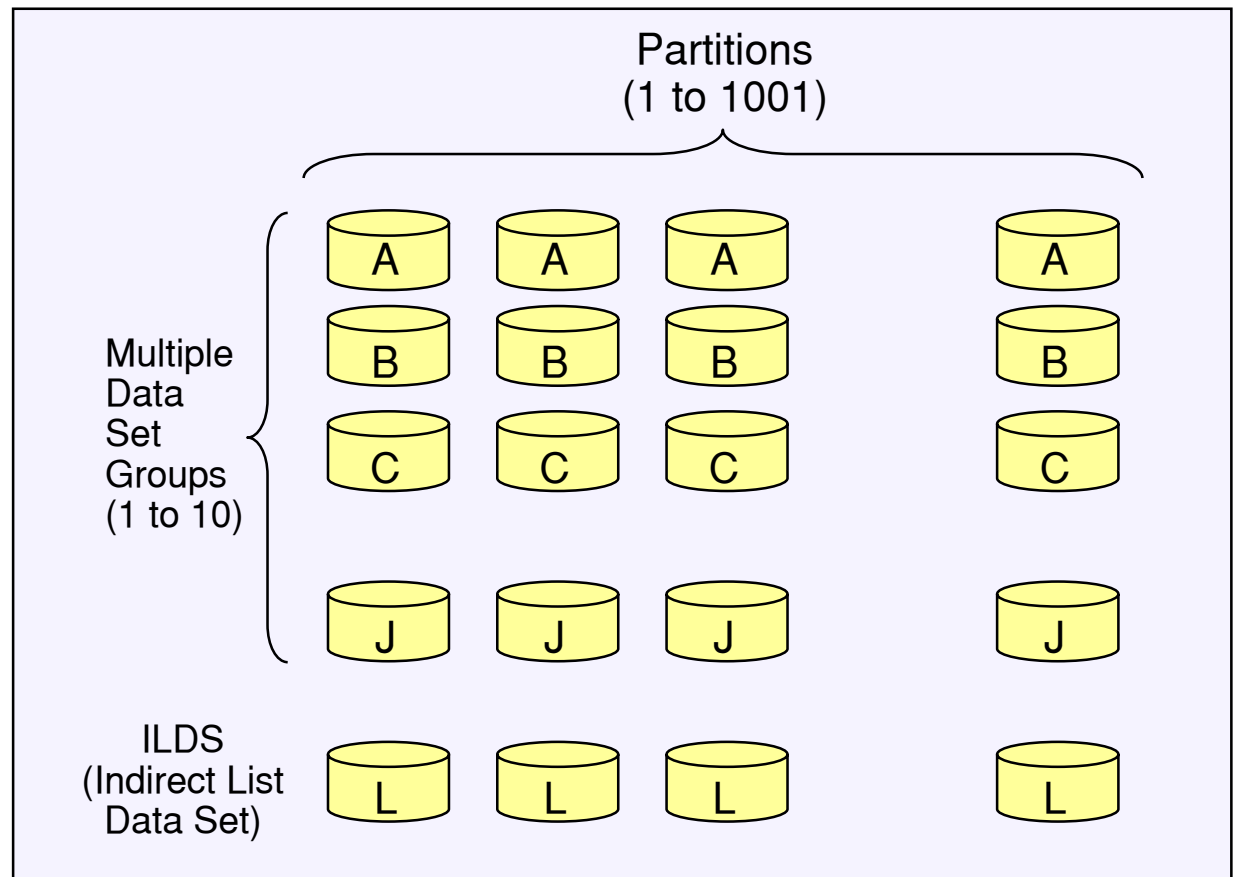


HDAM versus PHDAM Databases

HDAM Database Data Sets

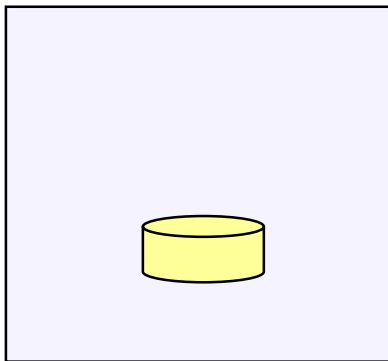


PHDAM Database Data Sets

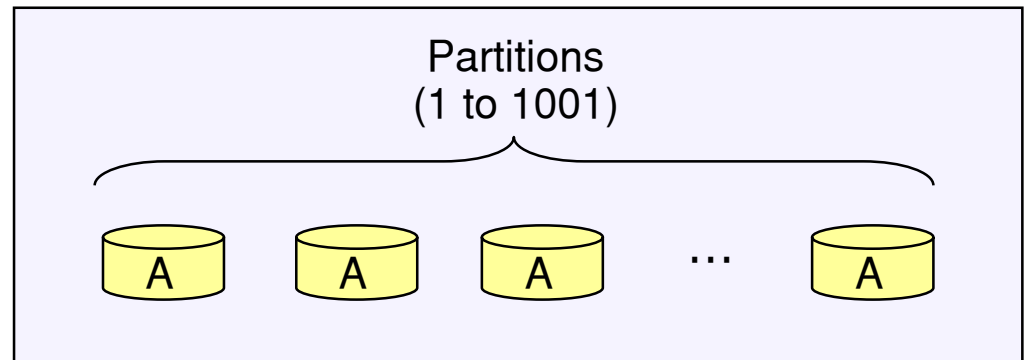


Secondary Index versus PSINDEX Databases

Secondary Index
Data Set



PSINDEX Data Sets



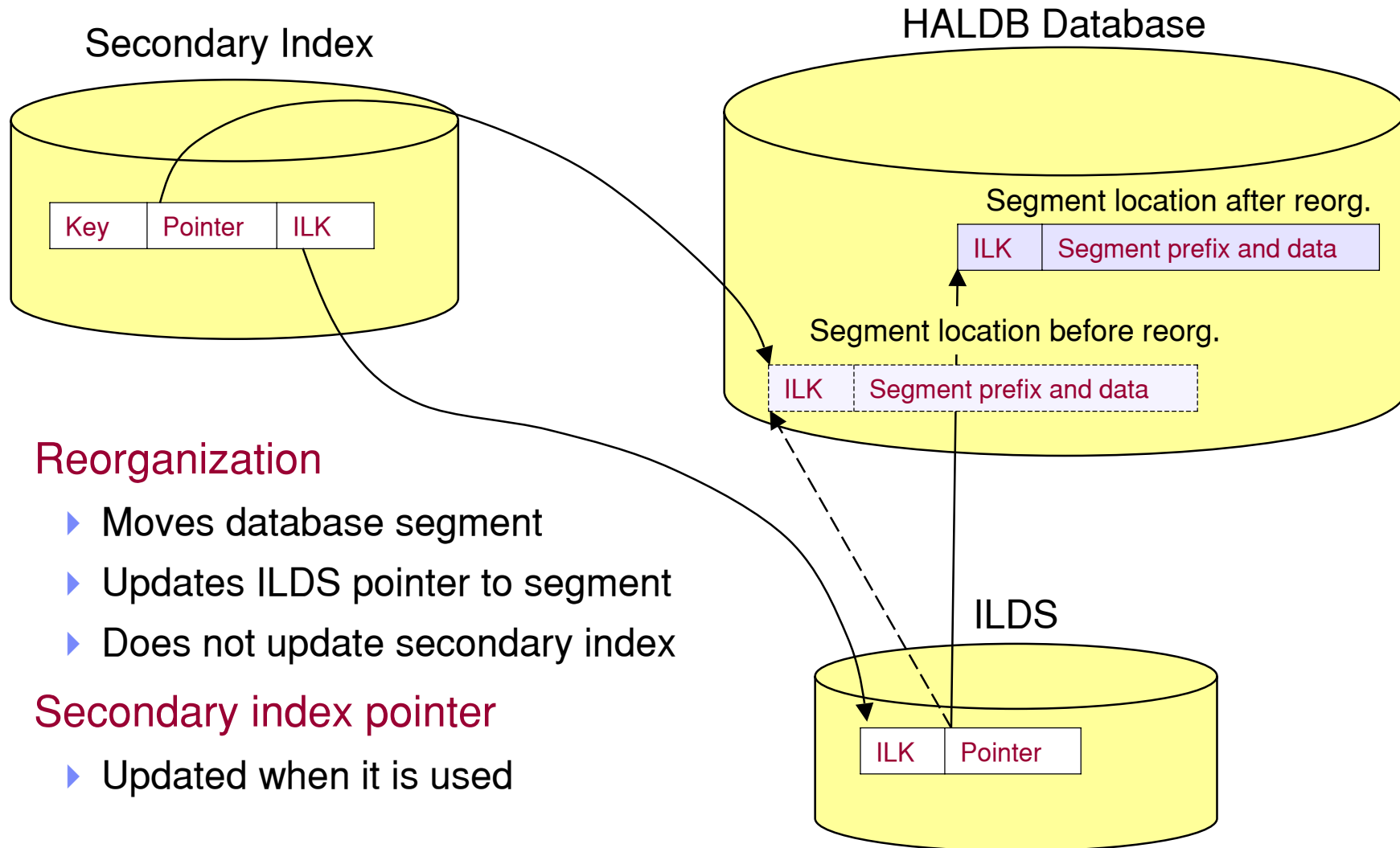
Partition Names and DDNAMEs

- Each partition has a name
 - ▶ Assigned by user
 - ▶ 1-7 characters (e.g. ABC1234)
- Partition names may be used in commands
 - ▶ /DBR DB ABC1234
 - ▶ /START DB ABC1234
- Partition names are used for DDNAMEs
 - ▶ DDNAME = partition_name concatenated with data set letter
 - ▶ DDNAMES:
 - PHIDAM index: ABC1234X
 - Primary data set group: ABC1234A
 - ILDS: ABC1234L

Partition IDs and Data Set Names

- Each partition has an ID number
 - ▶ Assigned by IMS when partition is defined
- Each partition has an 'data set name prefix'
 - ▶ Assigned by user when partition is defined
 - ▶ 1-37 character
- Data set name is formed from 'data set name prefix', data set letter, and partition ID
 - ▶ If data set name prefix: IMSP.DB.XYZ789
 - ▶ If partition ID: 00007
 - PHIDAM index: IMSP.DB.XYZ789.X00007
 - Primary data set group: IMSP.DB.XYZ789.A00007
 - ILDS: IMSP.DB.XYZ789.L00007
 - DSN prefix
 - Partition ID
 - Data set letter

Self-Healing Pointers Use of the ILDS



- **Reorganization**
 - ▶ Moves database segment
 - ▶ Updates ILDS pointer to segment
 - ▶ Does not update secondary index
- **Secondary index pointer**
 - ▶ Updated when it is used

ILK – Indirect List Key

High Availability

Availability

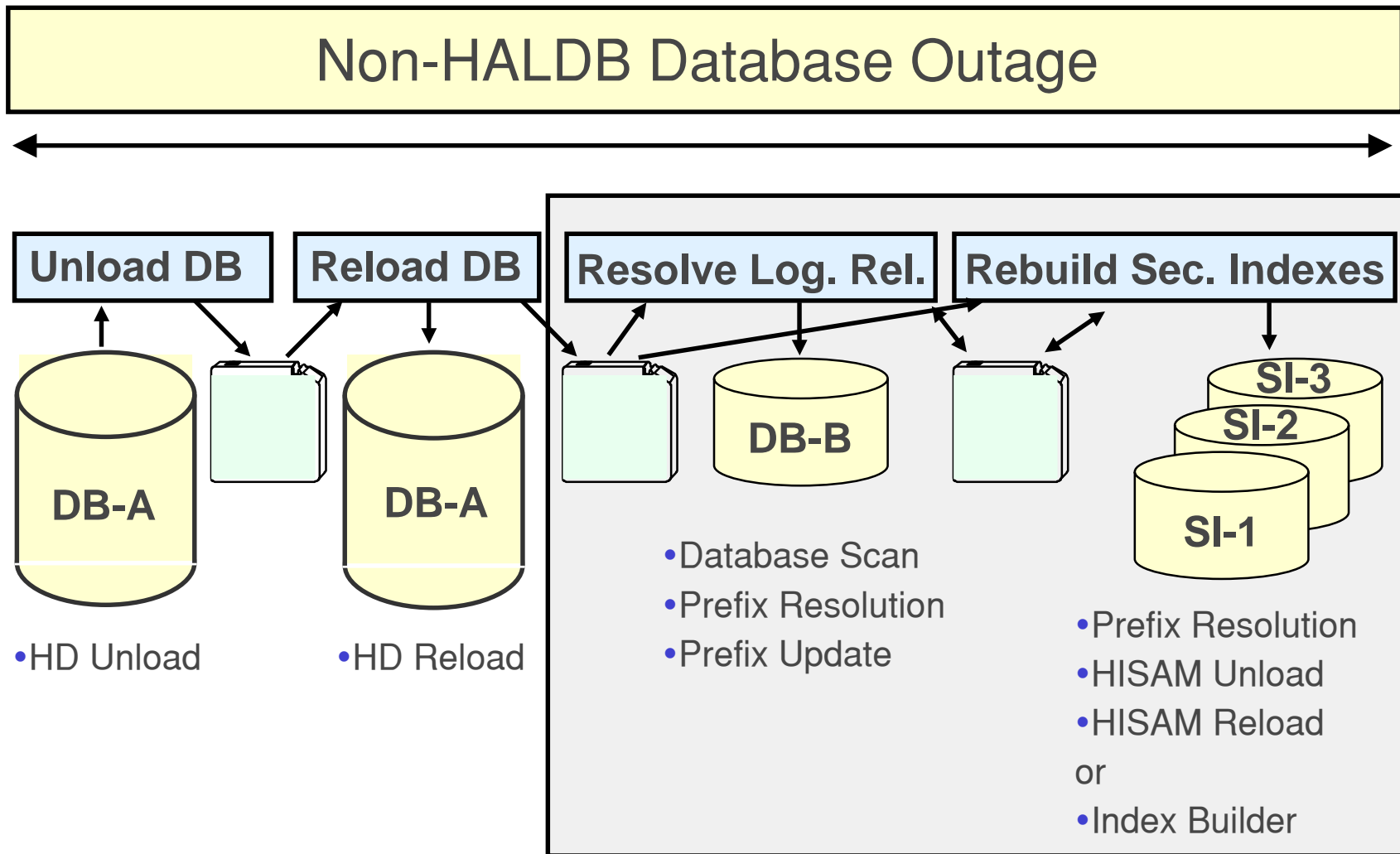
- **HALDB with IMS V7, V8, and V9**
 - ▶ Parallel processing
 - Reorganization by partition
 - Application processing
 - ▶ Smaller database data sets
 - Faster recoveries
 - Faster image copies
 - Faster reorganizations

- **HALDB with IMS V9**
 - ▶ Online reorganization
 - Absolutely no outage for reorganizations
 - Applications access all of the data during reorganizations without restrictions

Non-HALDB Reorganizations

- **Non-HALDB reorganizations**
 - ▶ Entire database is reorganized
 - Even if only part of it is disorganized
 - ▶ Entire database is unloaded
 - ▶ Entire database is reloaded
 - ▶ All secondary indexes must be rebuilt
 - ▶ All logically related databases must be updated

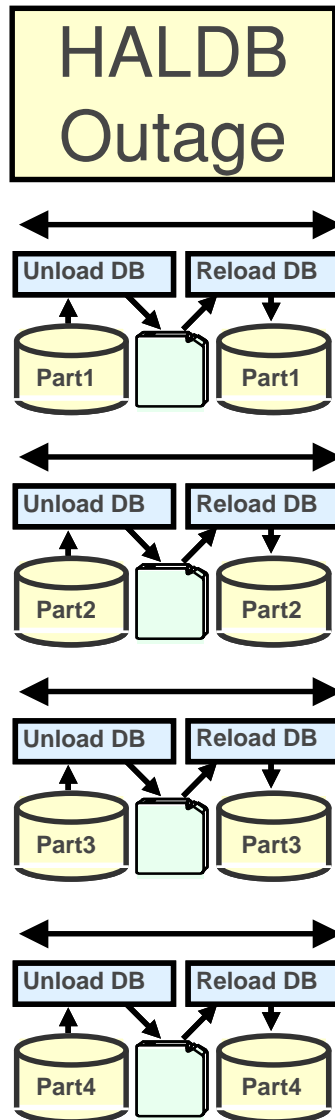
Non-HALDB Reorganizations



HALDB Reorganizations

- **HALDB reorganizations**
 - ▶ Individual partitions may be reorganized
 - Reorganizations may be staggered
 - Disorganization is often different in different partitions
 - It may not be necessary to reorganize all partitions
 - ▶ Partitions may be reorganized in parallel
 - Shortens the process
 - ▶ Secondary indexes are not rebuilt
 - Self-healing process will be used
 - ▶ Logical relationships are not resolved
 - Self-healing process will be used

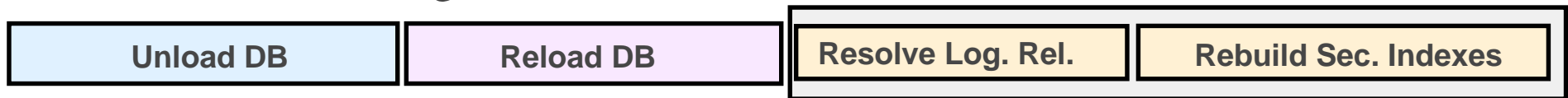
HALDB Reorganizations



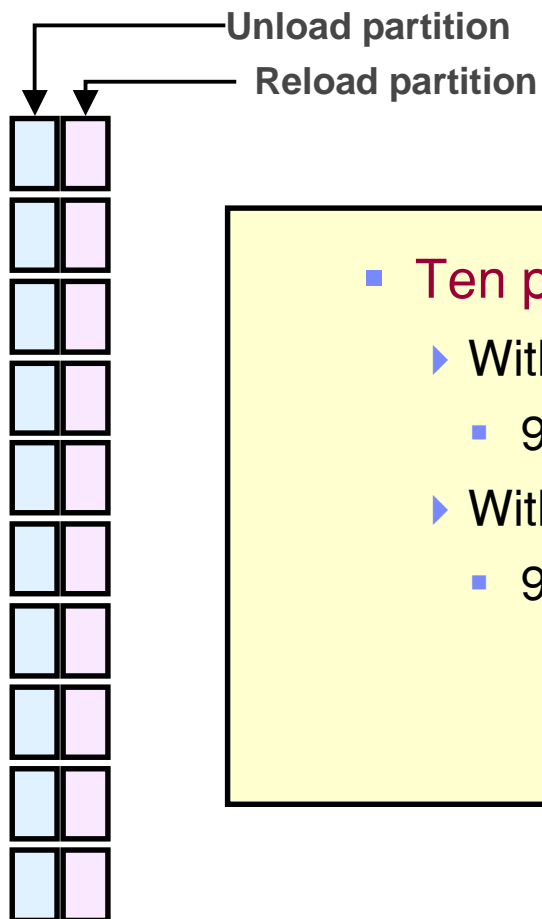
- Shorten the reorg time to your window
- Reorg partitions in parallel
 - ▶ Create enough partitions to meet your requirement
- Eliminate rebuilds of secondary indexes
 - ▶ Prefix Resolution, HISAM Unload, HISAM Reload, or Index Builder are not required
- Eliminate updates to logical relationships
 -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

Reorganization Comparison

■ Non-HALDB reorganization



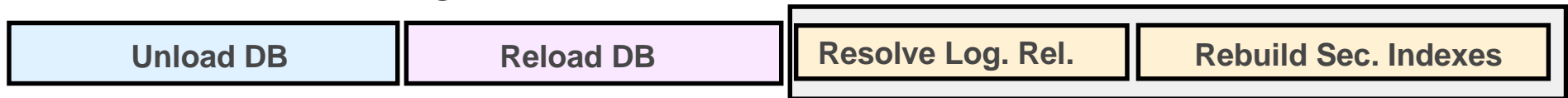
■ HALDB reorganization



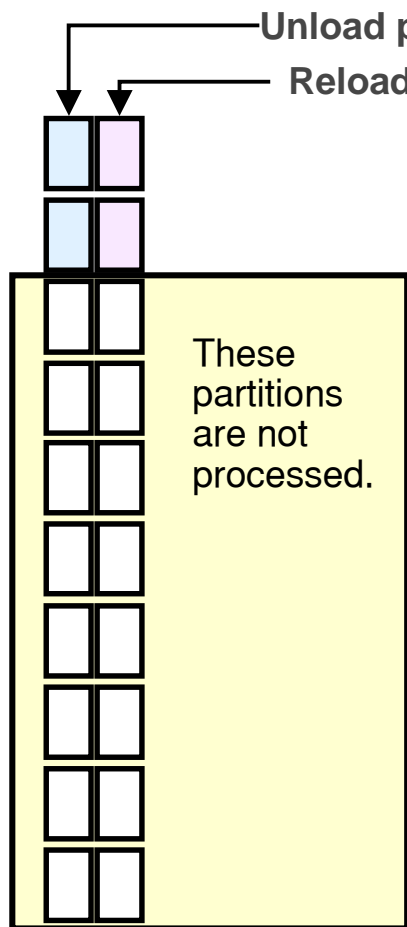
- **Ten partitions reduce reorganization time by how much?**
 - ▶ Without secondary indexes and logical relationships
 - 90% ? - From 2 hours to 12 minutes?
 - ▶ With secondary indexes or logical relationships
 - 95% ? - From 4 hours to 12 minutes?

Reorganization Comparison

- Non-HALDB reorganization



- HALDB reorganization



- **Reduced resource utilization**
 - ▶ We only reorganize partitions which need it
 - ▶ Could eliminate most of reorganization processing

IMS V9 HALDB Online Reorganization

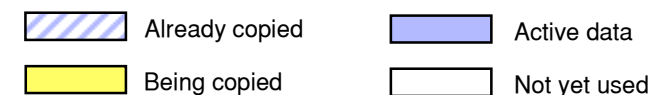
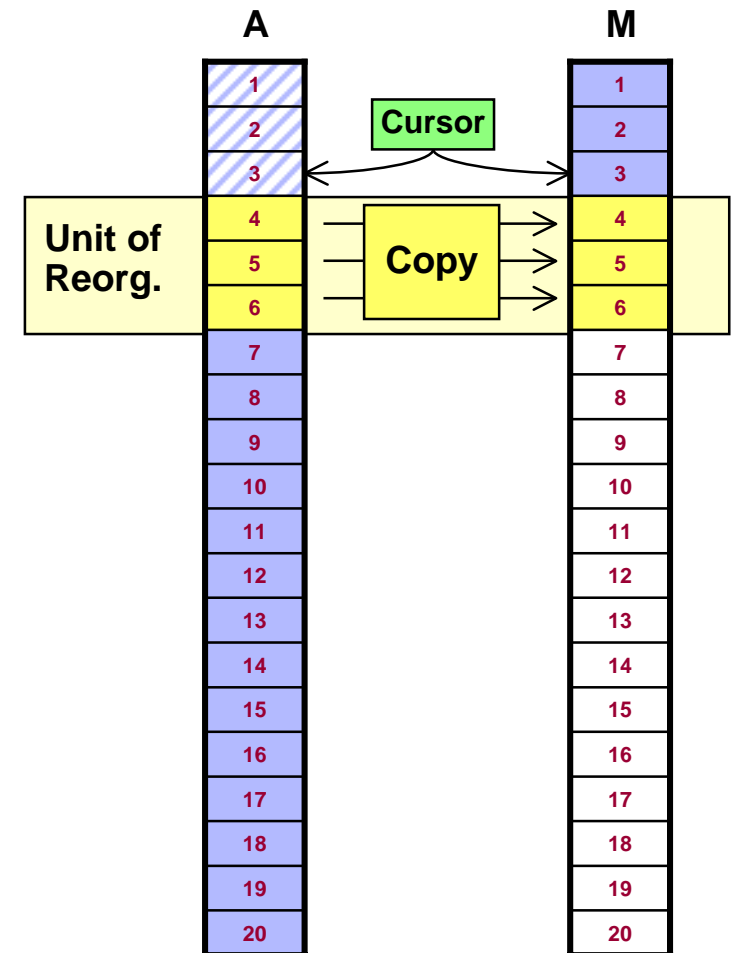
- **IMS V9 HALDB Online reorganization**
 - ▶ Absolutely no outage for reorganizations
 - ▶ Applications access all of the data during reorganizations without restrictions

- **Online reorganization**
 - ▶ Writes to new data sets
 - Dynamically allocates output data sets (optional)
 - Deletes input data sets when reorganization completes (optional)
 - Duplicate data sets
 - Only for partitions being reorganized
 - Only during the reorganization of the partitions

- **Supports data sharing**
 - ▶ Other IMS subsystems may read and update the partitions while they are reorged
 - ▶ Reorganization may be done in any data sharing IMS subsystem

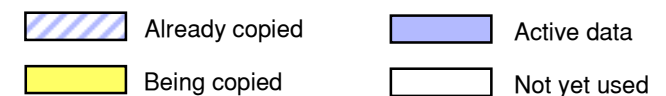
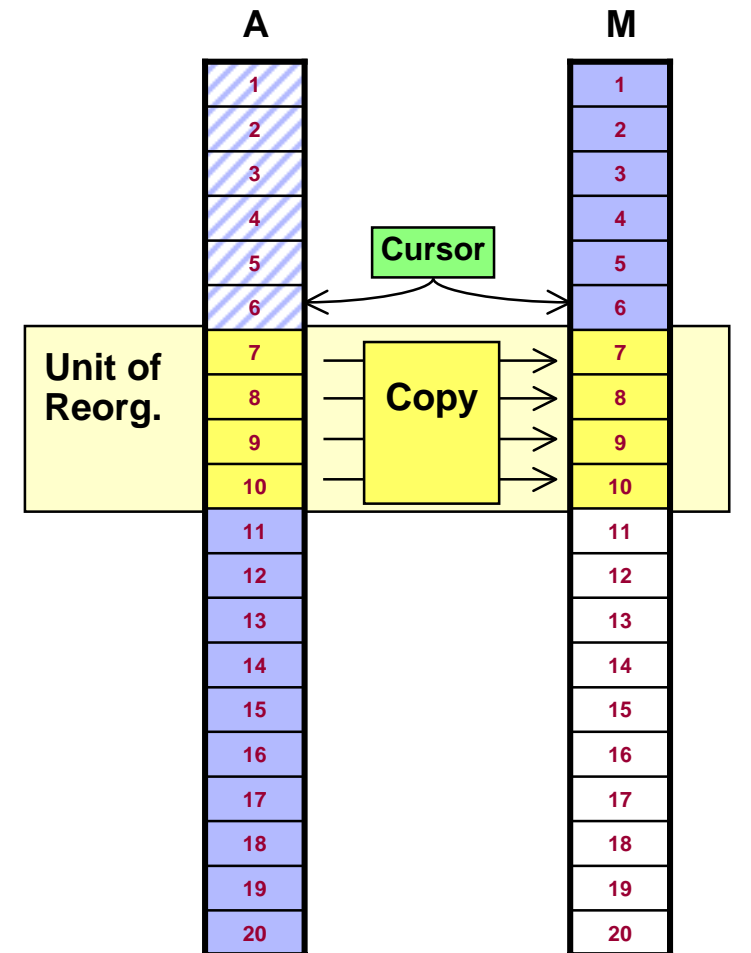
HALDB Online Reorganization

- **Two active sets of data sets**
 - ▶ Both are used during the reorg.
 - ▶ Records are copied to new data set
- **'Unit of Reorg'**
 - ▶ Set of records being copied at one time
 - ▶ Records are locked during copy
 - Number of records in UOR is dynamically adjusted
 - Algorithm limits time taken, bytes copied, and locks held during copy
 - ▶ Cursor determines which data set contains active record



HALDB Online Reorganization

- Data set used is based on cursor value
 - ▶ Cursor on record 6
 - ▶ Access Record 5:
 - Access from M data set
 - ▶ Access Record 14:
 - Access from A data set
 - ▶ Access Record 9:
 - Wait for lock,
 - then access from M data set

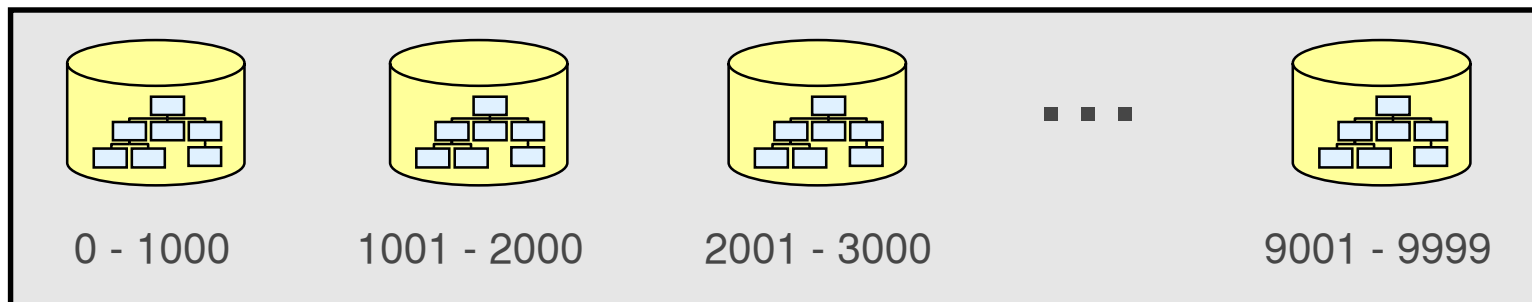


Partitioning Options

- Two methods of partitioning (both apply to PHDAM, PHIDAM, and PSINDEX)

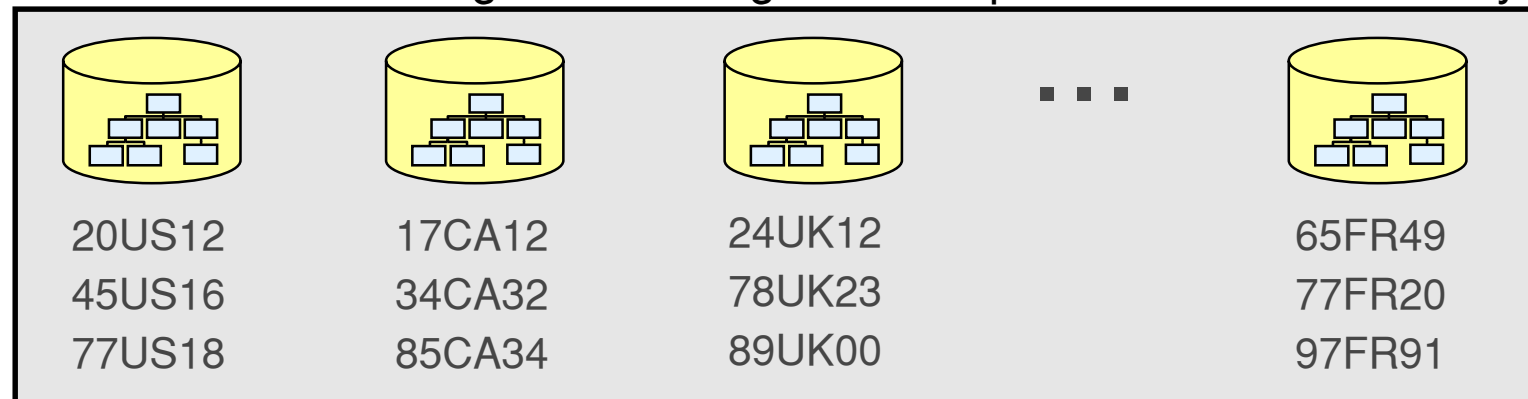
- ▶ Key range

- Each partition is assigned a range of root segment keys



- ▶ Partition Selection Exit routine

- The exit routine assigns a root segment to a partition based on its key



Modifying Partitioning of a Database

- **Partitioning may be changed for a database**

- ▶ Add partitions as database grows
- ▶ Add new partitions for new keys
- ▶ Split partitions which grow too large
- ▶ Combine partitions when data in them is deleted
- ▶ Delete partitions

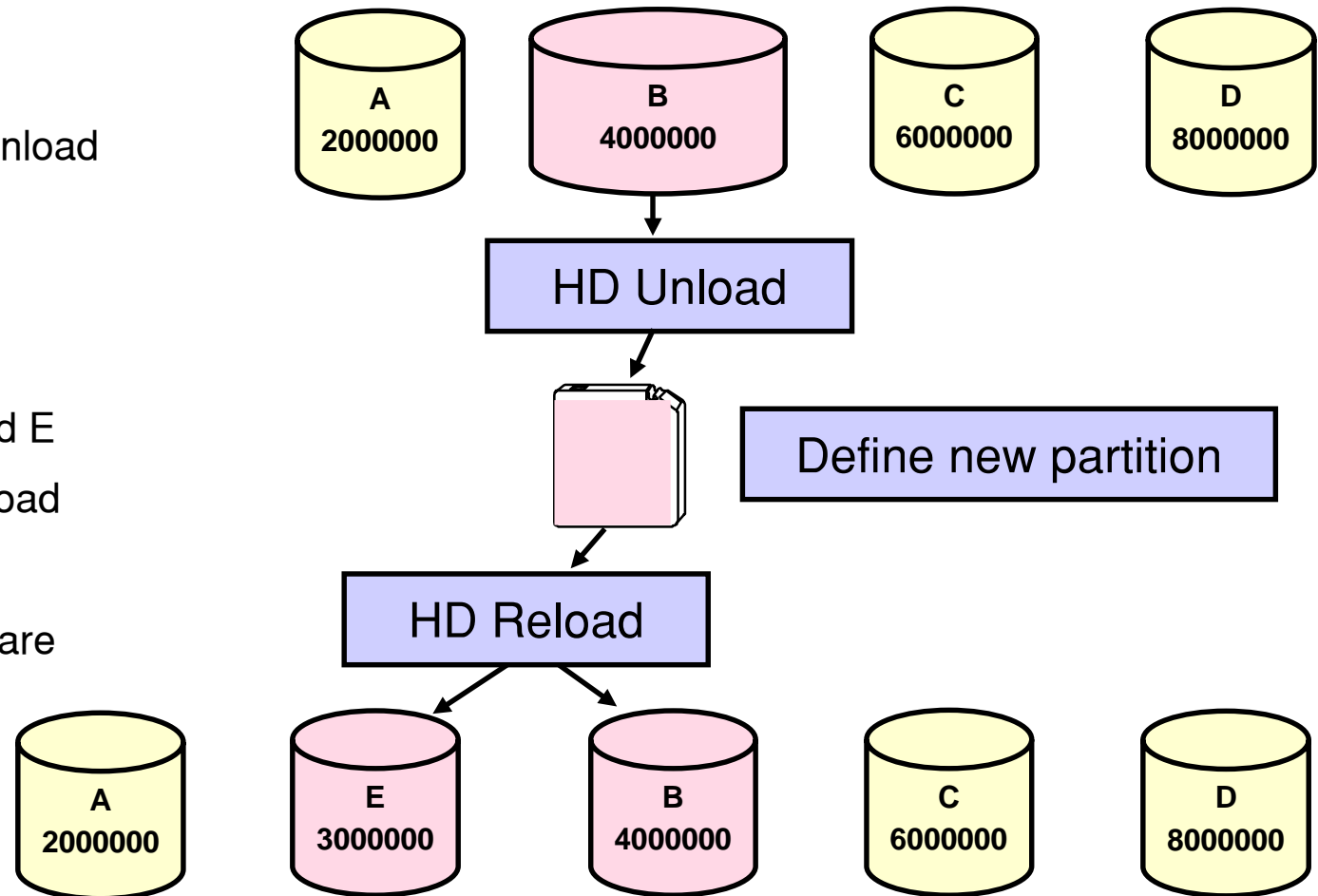
**Some examples
of these follow**

- ▶ Installations can react to changing data requirements

Splitting a Partition

- If partition B with high key 4000000 needs to be split

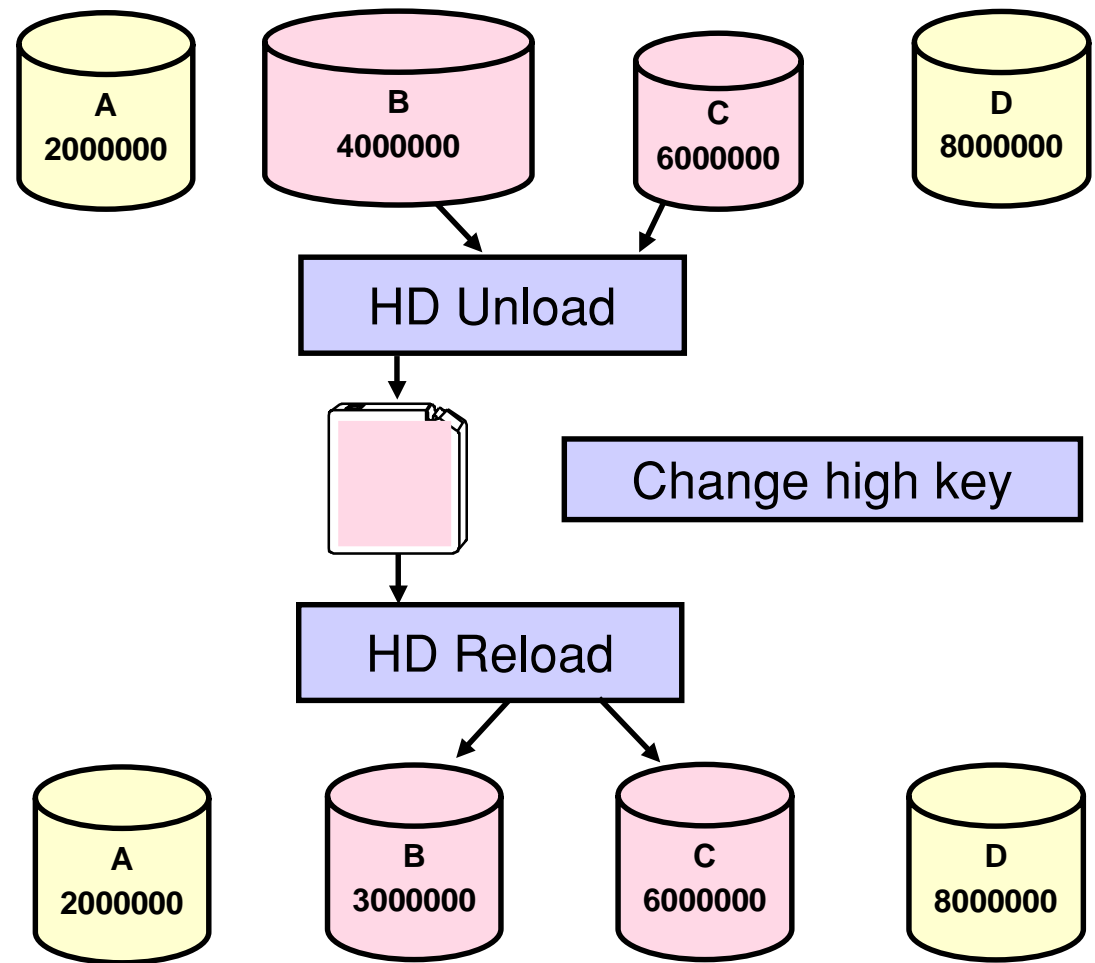
- ▶ Unload partition B
 - HD Unload or HP Unload
- ▶ Define new partition E
- ▶ Reload partitions B and E
 - HD Reload or HP Load
- ▶ Partitions A, C, and D are not affected



Modifying Partition Boundaries

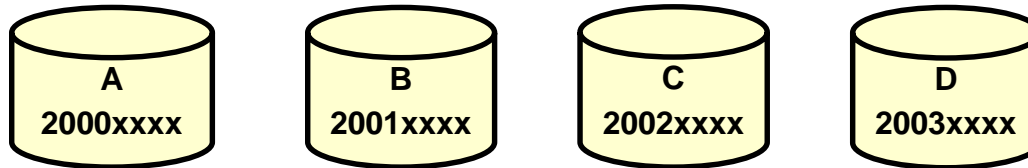
- If records need to be moved from partition B to C

- ▶ Unload partitions B and C
 - HD Unload or HP Unload
- ▶ Change high key for partition B
 - From 4000000 to 3000000
- ▶ Reload partitions B and C
 - HD Reload or HP Load
- ▶ Partitions A and D are not affected



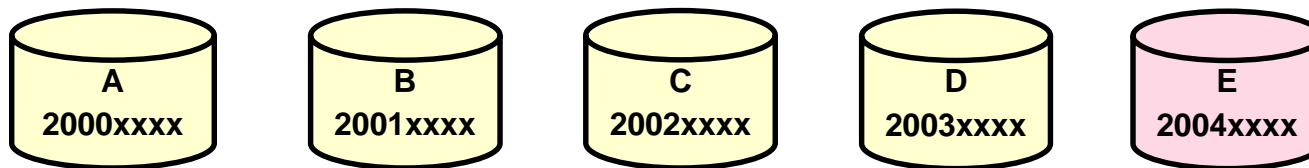
Databases with Dates for Keys

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



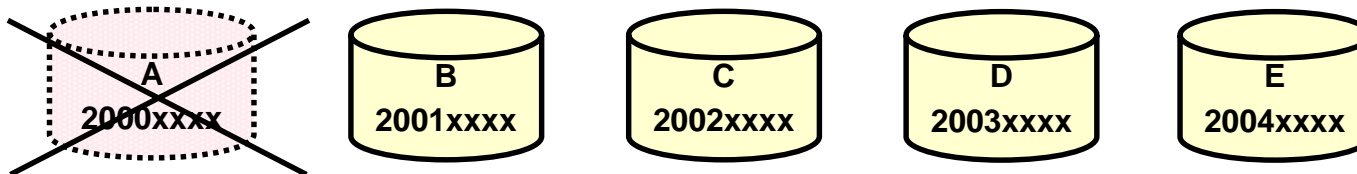
- To add a partition for a set of dates (higher keys)

- 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

Application Support

Compatibility

- Applications are compatible
 - ▶ No changes to programs when databases are converted
- Two exceptions
 - ▶ Initial loads of logically related databases
 - Cannot initially load logical children into a HALDB database
 - Do you do initial loads of databases with logical relationships?
 - ▶ Processing secondary indexes with /SX field and duplicate data
 - Only applies to programs which process the secondary index as a database
 - Does not apply to normal use of secondary indexes to access indexed database
 - Simple modification to application program
 - Increase I/O area by 4 bytes
 - Adjust offset to duplicate data by 4 bytes

Application Support

Performance

- **Application program may be limited to a partition (optional)**
 - ▶ HALDB control statement for BMP, batch (DLI and DBB), and JBP
 - Specifies a partition
 - Unqualified GU/GN call retrieves first segment in the partition
 - End of database ('GB') status code returned when end of partition is reached
 - ▶ Application without the control statement
 - Processes the entire database
 - ▶ Application with the control statement
 - **Processes only one partition**
 - **Application programming changes are NOT required for sequential processing**
 - ▶ Concurrent applications with control statements for different partitions
 - Process partitions in parallel

HALDB Benefits

- **Greater database capacity**
 - ▶ "Unlimited" capacity
 - ▶ Partitions may be modified as requirements change**Capacity**
- **Increased database availability**
 - ▶ Shortened reorganization process
 - ▶ Batch window may be shortened with concurrent processing**Availability**
- **Improved performance**
 - ▶ Concurrent processing against partitions**Performance**
- **Application program changes are not required****Compatibility**

Migrating Databases to HALDB

- Migration Overview

1. Size partitions

- Decide how many partitions you will want and their boundaries

2. Unload non-HALDB database with HD Unload or HP Unload

- Creates unload data set

3. DBDGEN for HALDB

4. Define partitions for HALDB

- Use Partition Definition Utility or DBRC commands

5. Allocate HALDB database data sets

6. Initialize HALDB database partitions

7. Load HALDB database with HD Reload or HP Load

- Uses HALDB DBD and RECONS
- Reads unload data set

Migrating Databases to HALDB

- Considerations:
 - ▶ All logically related databases must be migrated concurrently
 - No logical relationships between HALDB and non-HALDB databases
 - ▶ HALDB does not support virtual pairing
 - Migration is to physical pairing
 - ▶ All secondary indexes to a database must be migrated with the database
 - Indexes to HALDB databases must be HALDB (ACCESS=PSINDEX)

Migrating Databases to HALDB

- **The HALDB Conversion and Maintenance Aid (IMS Tool 5655-K47)**
 - ▶ A tool from IBM
 - ISPF based
 - ▶ Makes DBAs more productive
 - Automates many of the tasks
 - Allows DBAs to convert more databases more rapidly
 - Requires less skilled DBAs than manual conversions
 - Provides both conversion and maintenance capabilities
 - Provides other special functions

HALDB Conversion and Maintenance Aid

- **Simplifies conversion**
 - ▶ Determines high keys, partition sizes, etc.
 - ▶ Converts DBDs
 - ▶ Defines partitions to DBRC
 - ▶ Allocates database data sets
 - ▶ Creates unloads and reloads
 - ▶ Handles partitioned databases (PDB, PDF, and user)
 - ▶ Handles DEDB to HALDB conversions

- **Simplifies maintenance**
 - ▶ Splits and consolidates partitions
 - ▶ Clones database and partition definitions to other RECONs
 - ▶ Copies partitions to other RECONs

HALDB Information

- Redbook:
 - ▶ *The Complete IMS HALDB Guide, All You Need to Know to Manage HALDBs*
 - SG24-6945
 - Available from www.ibm.com/redbooks
- Presentations and articles on the Web
 - ▶ IMS High Availability Large Database (HALDB)
 - <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS185>
 - ▶ Migrating to IMS HALDB
 - <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS693>
 - ▶ Application Design and Programming with HALDB
 - <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS490>
 - ▶ Using GENJCL.USER to Allocate IMS HALDB Data Sets
 - <http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD100491>
 - ▶ From the IMS home page (www.ibm.com/ims) click on "presentations/papers"
 - More information about many IMS topics

HALDB Conversion and Maintenance Aid Information

- IMS tools
 - ▶ <http://www.ibm.com/software/data/db2imstools/>
 - Contains links to all of the IBM IMS tools

- HALDB Conversion and Maintenance Aid
 - ▶ <http://www.ibm.com/software/data/db2imstools/imstools/imshaldb.html>
 - Contains links to Fact Sheet and Library (User's Guide)