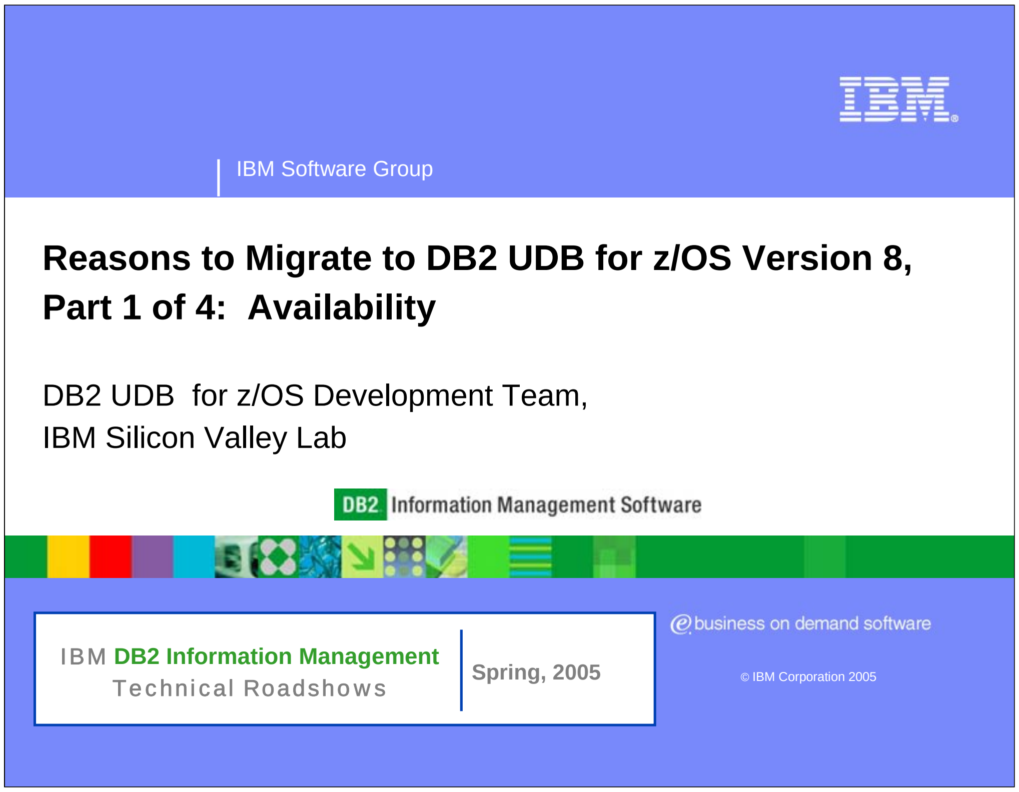

IBM Software Group

Reasons to Migrate to DB2 UDB for z/OS Version 8, Part 1 of 4: Availability

DB2 UDB for z/OS Development Team,
IBM Silicon Valley Lab

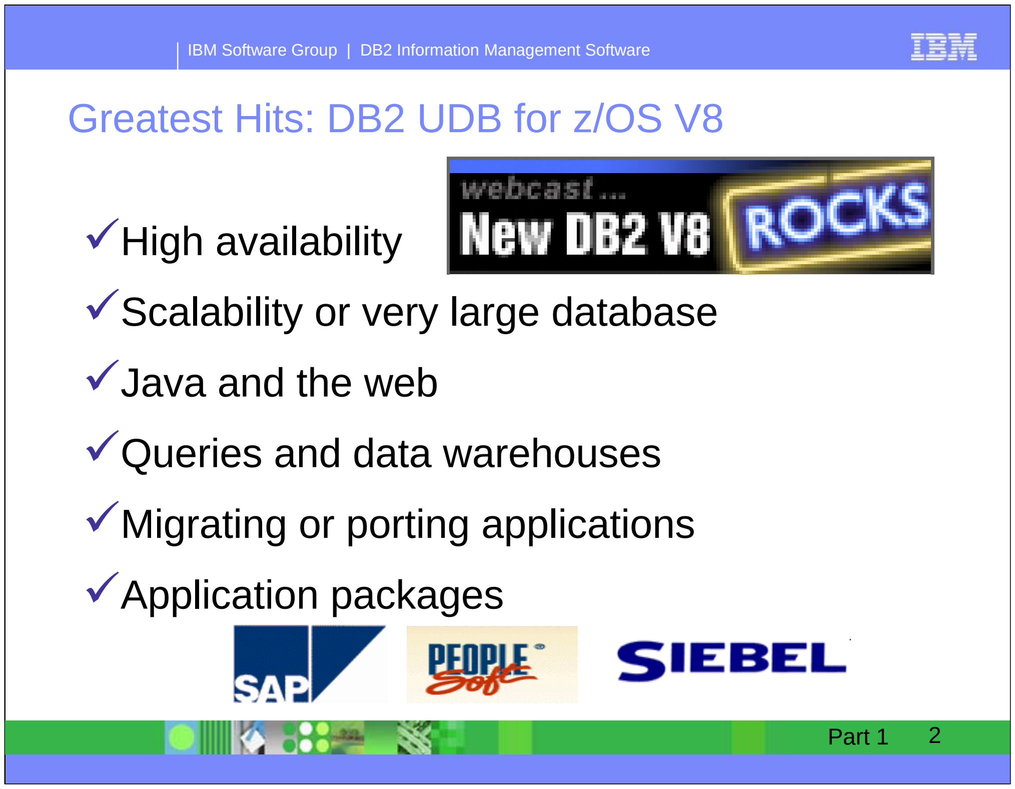
DB2 Information Management Software

IBM DB2 Information Management
Technical Roadshows

Spring, 2005

business on demand software

Find out how DB2® Universal Database for z/OS® Version 8 has been re-engineered, with fundamental changes in architecture and structure that will help you manage your very large databases more easily and cost effectively. "Reasons to Migrate to DB2 UDB Version 8" is a day long seminar suited to database technical managers, database administrators, applications developers and systems programmers. It focuses on four critical areas: availability, integration, application development productivity, and flexible growth and incremental scalability. Its information-packed sessions will familiarize you with the enhancements that are enabling organizations to streamline database management, respond more flexibly and quickly to business needs and ensure short and long-term growth capacity.



Greatest Hits: DB2 UDB for z/OS V8

webcast...
New DB2 V8 ROCKS

- ✓High availability
- ✓Scalability or very large database
- ✓Java and the web
- ✓Queries and data warehouses
- ✓Migrating or porting applications
- ✓Application packages

SAP
PEOPLE Soft
SIEBEL

The greatest hits are the situations where I would recommend looking at V8 soon.

Greatest Hit 1: High availability: One of the biggest steps for database administrators in continuous availability is online schema evolution, with the ability to add partitions and make about 20 changes with ALTER. New backup and recovery utilities are useful for disaster recovery and will be the primary backup technique for some customers.

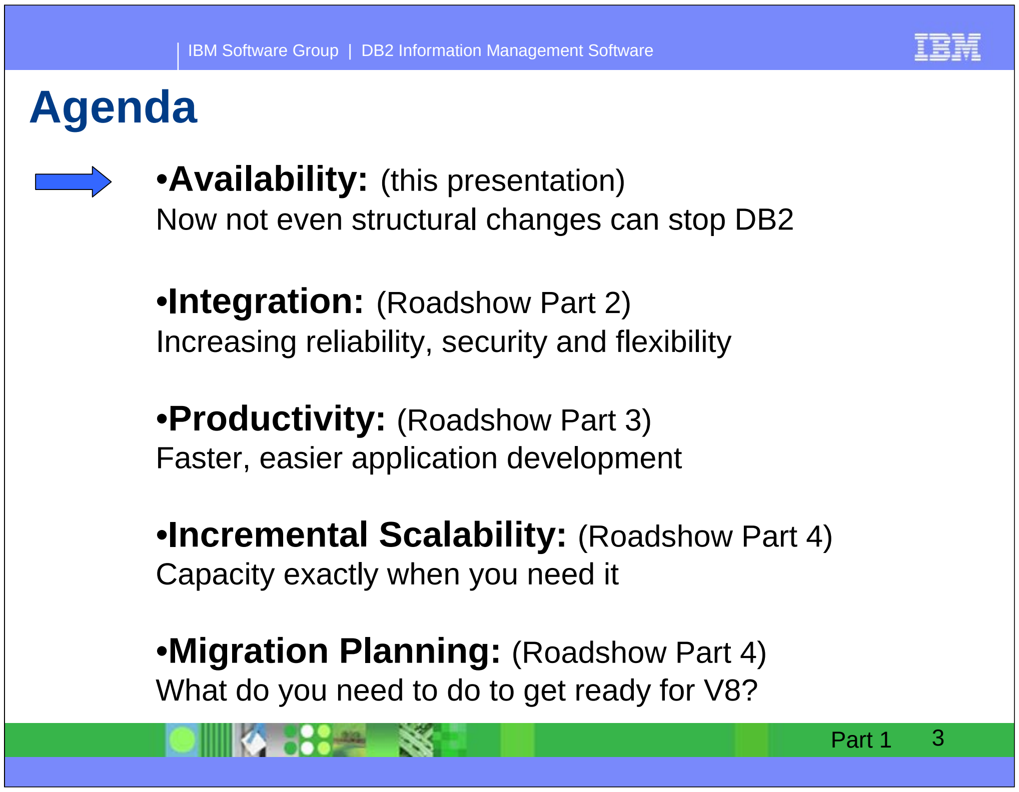
Greatest Hit 2: Scalability or very large databases: Separate partitioning and clustering allows two dimensional clustering with more effective IO. New index options and 4096 partitions provide more efficient access. Availability and optimization improvements are critical for very large databases. Use more memory, more effectively for scalability.

Greatest Hit 3: Java and the web: Improvements in the SQLJ and JDBC support, a new Java Universal Driver, enhanced Unicode support, integration with WebSphere and new XML functions make Java and web applications more robust and more productive.

Greatest Hit 4: Queries and data warehouses: Optimization changes provide the best performance improvement opportunities in V8. Faster response and reduced processing time come from improved optimization and better information for the optimizer. New database design options for indexes, clustering and materialized query tables provide more gains. Warehouses often need to have the new rotate partition capability.

Greatest Hit 5: Migrating or porting applications from other platforms: Many SQL enhancements provide better compatibility with the DB2 family and with the industry. If customers develop on Windows, Unix or Linux, and then move to z/OS, the process is much easier. Early customers reported success at porting applications.

Greatest Hit 6: Application packages: SAP, PeopleSoft, Siebel, etc. : About 50 improvements, including everything mentioned in the “Greatest Hits” section, are provided for most of the key vendor packages. SAP R/3 4.6 and PeopleSoft PeopleTools 8.45 were already certified for V8, less than four months after general availability.


IBM Software Group | DB2 Information Management Software

Agenda

- **Availability:** (this presentation)
  Now not even structural changes can stop DB2
- **Integration:** (Roadshow Part 2)
  Increasing reliability, security and flexibility
- **Productivity:** (Roadshow Part 3)
  Faster, easier application development
- **Incremental Scalability:** (Roadshow Part 4)
  Capacity exactly when you need it
- **Migration Planning:** (Roadshow Part 4)
  What do you need to do to get ready for V8?

Part 1 3

Many of the items I'm talking about today improve performance, availability, productivity and scalability. We'll try to keep the items that are closely related together, but categories don't always work when you need to choose and the real answer is both or all five.

The agenda for today is roughly

8:15 am Registration and Continental Breakfast

8:45 am Welcome and Introduction

9:00 am Availability: Now not even structural changes can stop DB2

10:30am Break

11:00 am Integration: Increasing reliability, security and flexibility

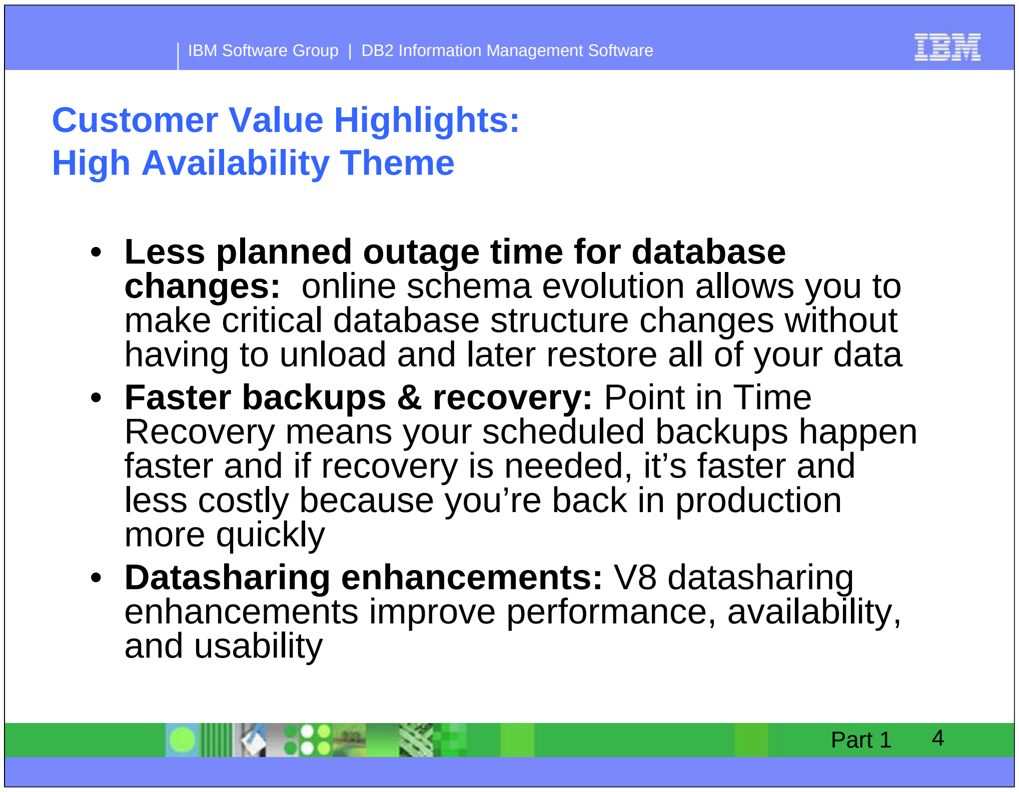
12:30 pm Lunch

1:30 pm Productivity: Faster, easier application development

3:00pm Break

3:30 pm Incremental Scalability: Capacity exactly when you need it
Migration Planning: How to get there

5:00 pm Close of Program


IBM Software Group | DB2 Information Management Software

IBM

Customer Value Highlights: High Availability Theme

- **Less planned outage time for database changes:** online schema evolution allows you to make critical database structure changes without having to unload and later restore all of your data
- **Faster backups & recovery:** Point in Time Recovery means your scheduled backups happen faster and if recovery is needed, it's faster and less costly because you're back in production more quickly
- **Datasharing enhancements:** V8 datasharing enhancements improve performance, availability, and usability

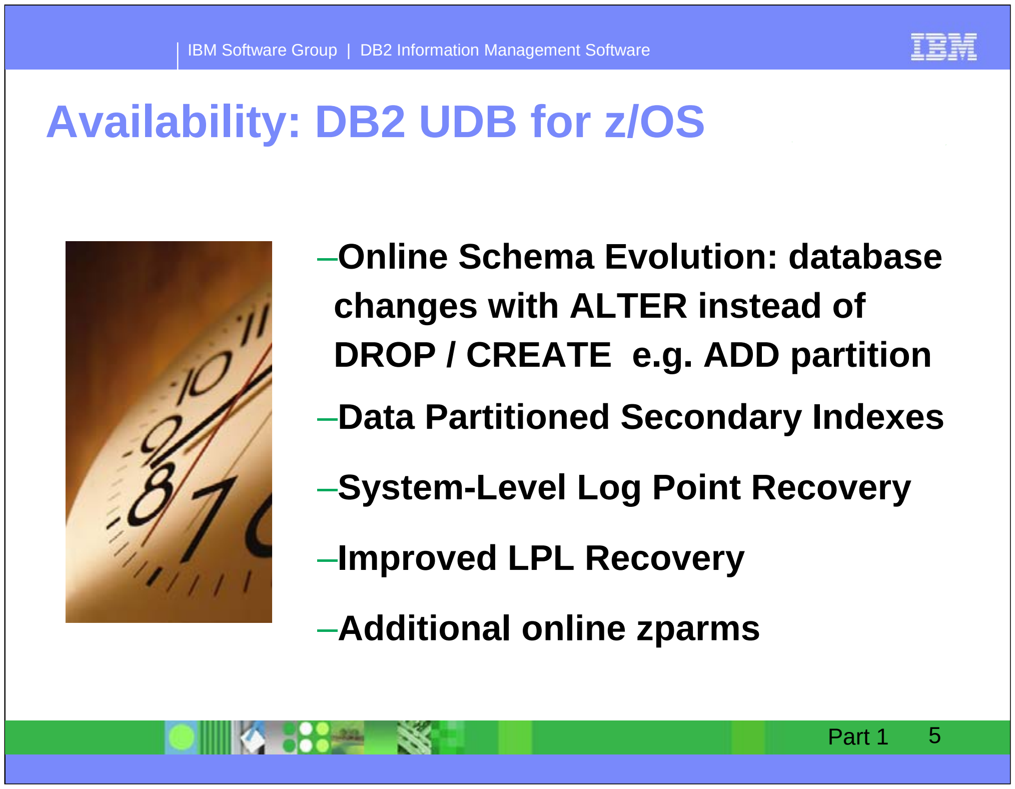
Part 1 4

The primary improvements in availability for Version 8 are

•A break through in the ability to make changes to the data definition and structure. We call it online schema evolution or database definition on demand. It may also be called the database administrator's favorite improvement, with more to come.

•A new infrastructure for backup and recovery that will become the standard technique for disaster recovery and the technique used for all backup when the number of data sets is very large.

•A number of enhancements for data sharing to improve performance and availability.



Availability: DB2 UDB for z/OS

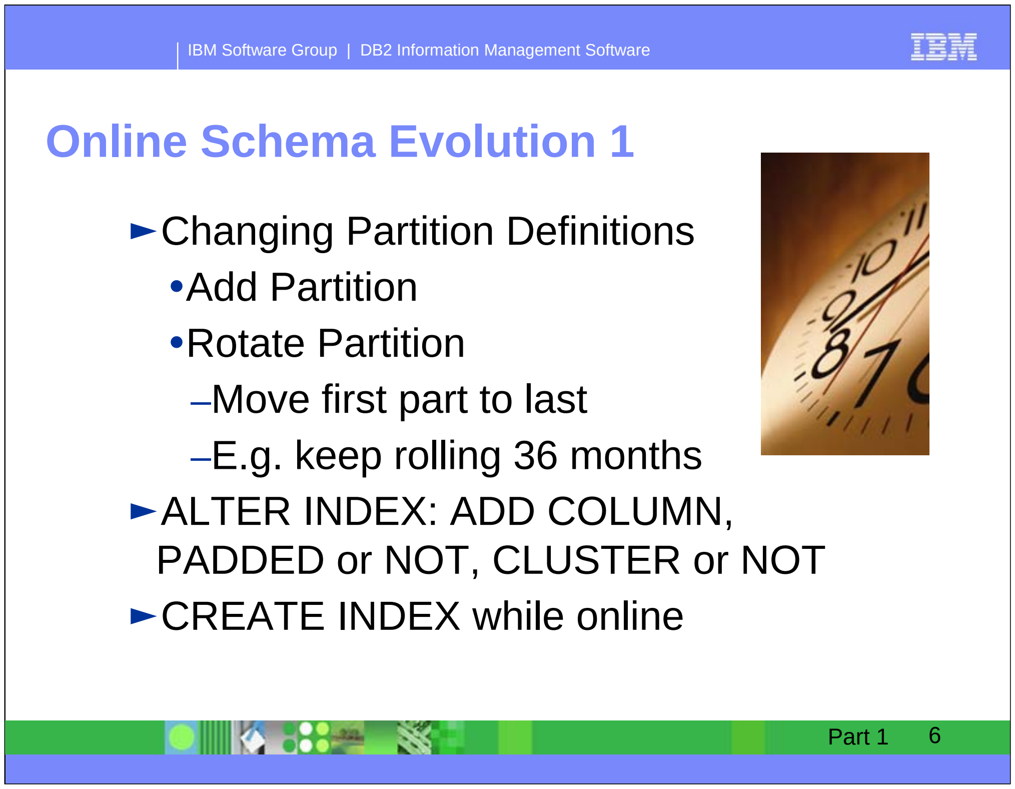
- **Online Schema Evolution: database changes with ALTER instead of DROP / CREATE e.g. ADD partition**
- **Data Partitioned Secondary Indexes**
- **System-Level Log Point Recovery**
- **Improved LPL Recovery**
- **Additional online zparms**

The most important change for many customers, especially database administrators, is the ability to use ALTER in many places instead of needing to drop and redefine. We call this schema evolution, and it can reduce outages by hours or days for a major structure change on an application.

The ability to have secondary indexes that are partitioned with the data can improve recovery times by an order of magnitude. It can also eliminate the outage for online reorganizing a single partition or BUILD2 phase.

We have some additional cases where subsystem parameters can be changed while the subsystem is running.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z03.pdf



IBM Software Group | DB2 Information Management Software

Online Schema Evolution 1

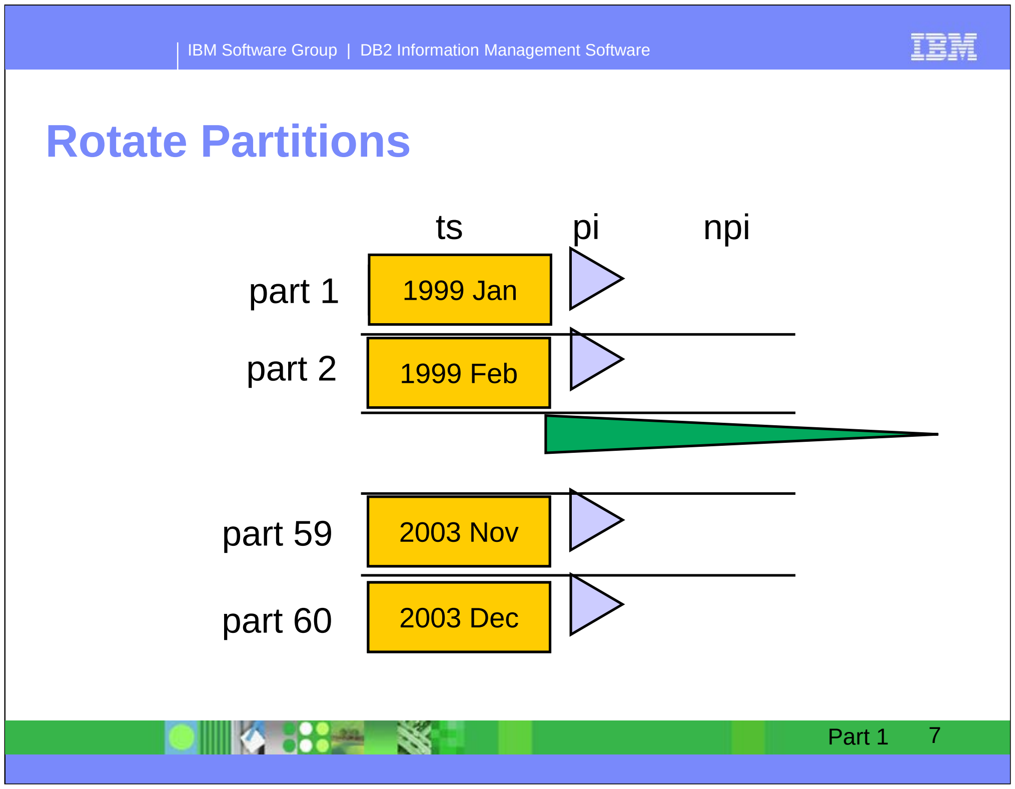
- ►Changing Partition Definitions
  - •Add Partition
  - •Rotate Partition
    - –Move first part to last
    - –E.g. keep rolling 36 months
- ►ALTER INDEX: ADD COLUMN, PADDED or NOT, CLUSTER or NOT
- ►CREATE INDEX while online

Part 1 6

Three types of changes are very high on our priority list: changing partitions, changing table attributes and unbundling partitioning and clustering. This is the first category, partition changes.
Adding a new partition to an existing partitioned tables space is very important. Rotating the partitions, such as keeping a rolling 36 months of data is also key.
Creating an index has been very disruptive, but V8 provides changes that will allow creation of an index while the work continues.

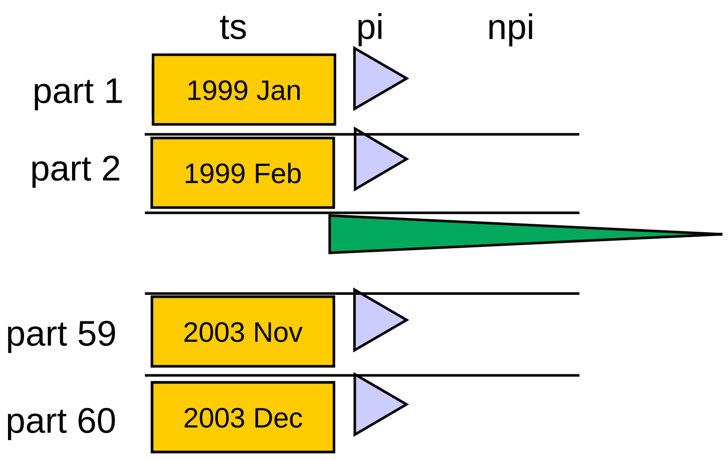
ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z21m.pdf

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z22m.pdf



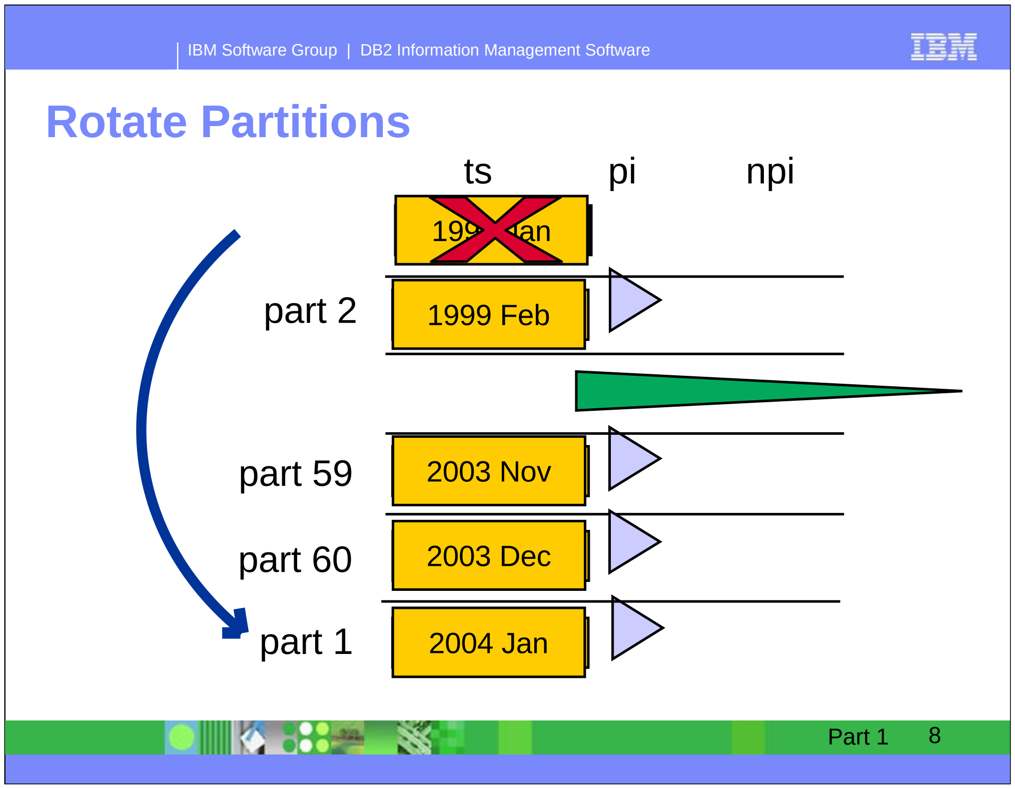
IBM Software Group | DB2 Information Management Software

Rotate Partitions



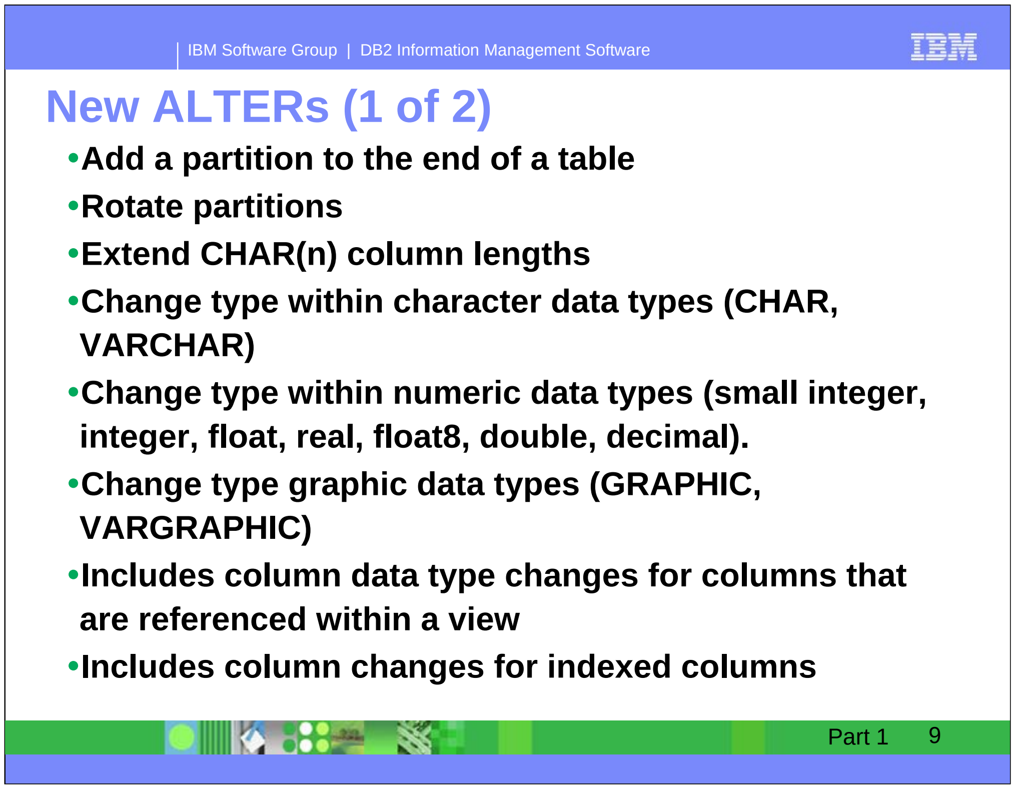
Part 1 7

This is a picture for rotating a partition, to keep the most current 60 partitions or five years by month. As we reach the end of December 2003, we need to get a new partition for 2004.



Rotate Partitions

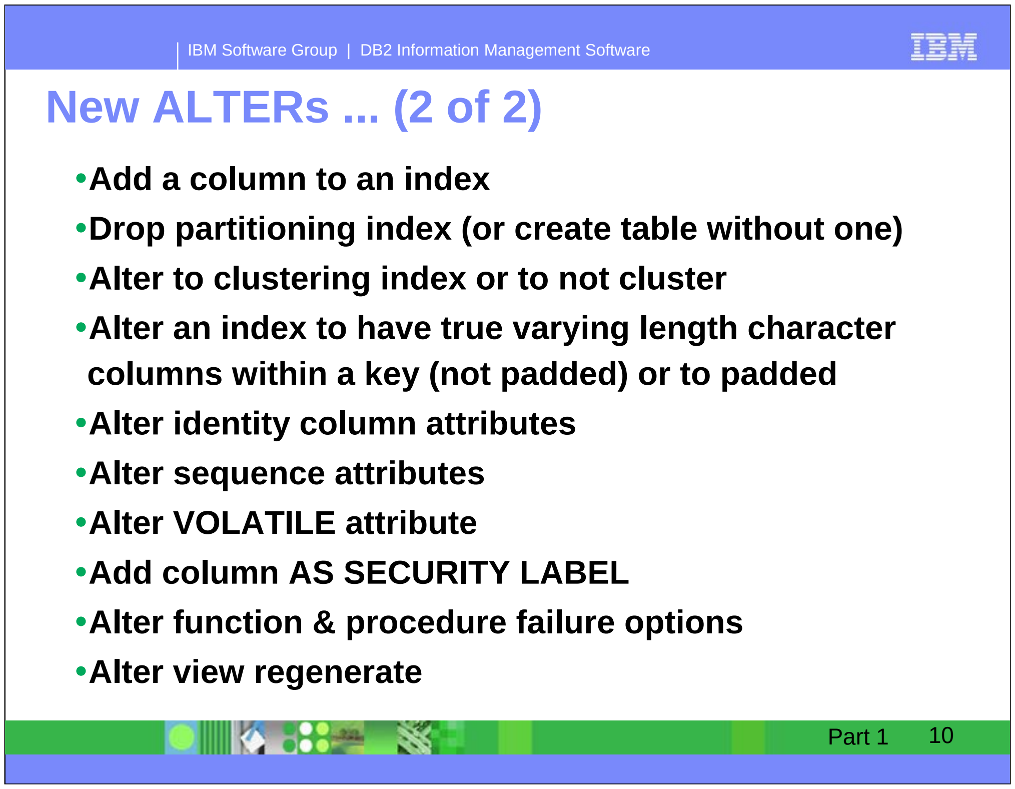
Rather than just create a new partition, we empty the first logical partition and rotate it to be the last one. In many cases, one additional partition is needed.



New ALTERs (1 of 2)

- Add a partition to the end of a table
- Rotate partitions
- Extend CHAR(n) column lengths
- Change type within character data types (CHAR, VARCHAR)
- Change type within numeric data types (small integer, integer, float, real, float8, double, decimal).
- Change type graphic data types (GRAPHIC, VARGRAPHIC)
- Includes column data type changes for columns that are referenced within a view
- Includes column changes for indexed columns

This is part one of a list of the changes in the ability to ALTER instead of needing to DROP and recreate. As we looked at the number of possible changes, it became clear that schema evolution would have to evolve. We tried to include the most important changes in this first delivery.



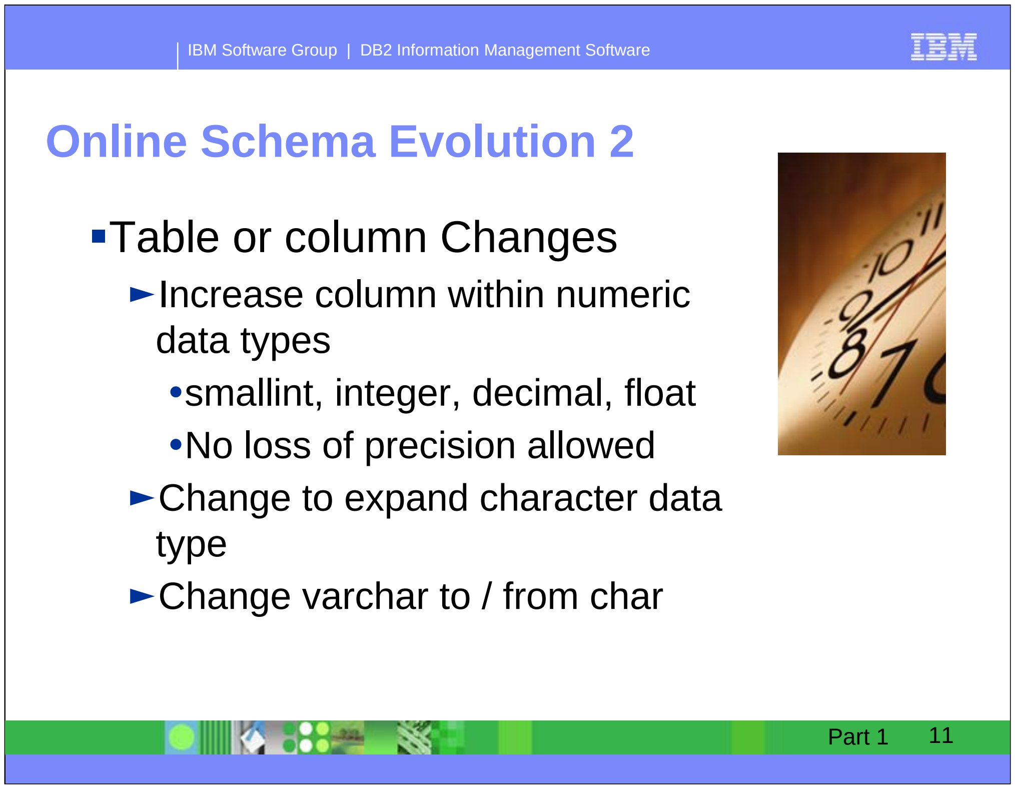
IBM Software Group | DB2 Information Management Software

New ALTERs ... (2 of 2)

- Add a column to an index
- Drop partitioning index (or create table without one)
- Alter to clustering index or to not cluster
- Alter an index to have true varying length character columns within a key (not padded) or to padded
- Alter identity column attributes
- Alter sequence attributes
- Alter VOLATILE attribute
- Add column AS SECURITY LABEL
- Alter function & procedure failure options
- Alter view regenerate

Part 1 10

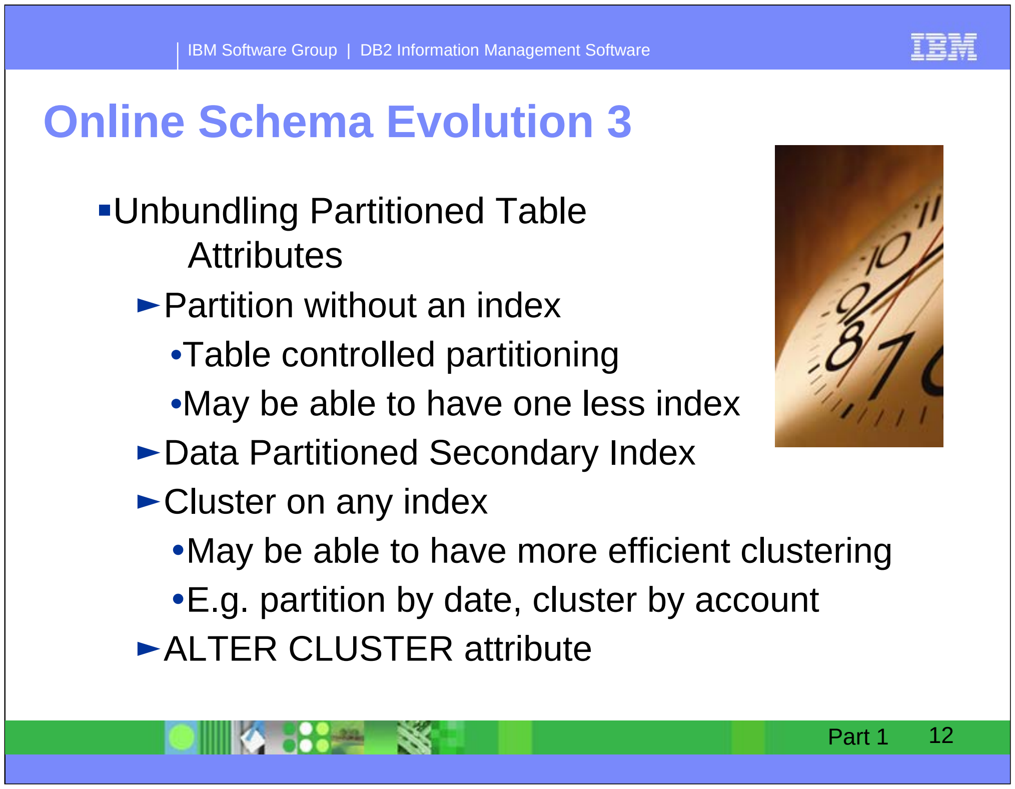
This is the second part of a list of the changes in the ability to ALTER instead of needing to DROP and recreate. On the next foils, we'll discuss the most important changes. There are many more attributes to alter. If you don’t see attributes that are important to alter, be sure to indicate your priorities to your IBMer. We expect to see online schema evolution evolve in versions of DB2 after V8.



Online Schema Evolution 2

- Table or column Changes
  - Increase column within numeric data types
    - smallint, integer, decimal, float
    - No loss of precision allowed
  - Change to expand character data type
  - Change varchar to / from char

We are able to change the data type for columns. In V5 we could increase the size of varchar columns, but this change allows us to extend numeric and character columns and to change between char and varchar.



Online Schema Evolution 3

- Unbundling Partitioned Table Attributes
  - Partition without an index
    - Table controlled partitioning
    - May be able to have one less index
  - Data Partitioned Secondary Index
  - Cluster on any index
    - May be able to have more efficient clustering
    - E.g. partition by date, cluster by account
  - ALTER CLUSTER attribute

Partitioning and clustering are bundled in current DB2. Some of the time we are required to make a difficult choice. We also want to partition without an index and be able to cluster on any index. These changes will allow us to have one less index and less random IO in some cases.

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z21m.pdf

ftp://ftp.software.ibm.com/software/db2storedprocedure/db2zos390/techdocs/Z22m.pdf

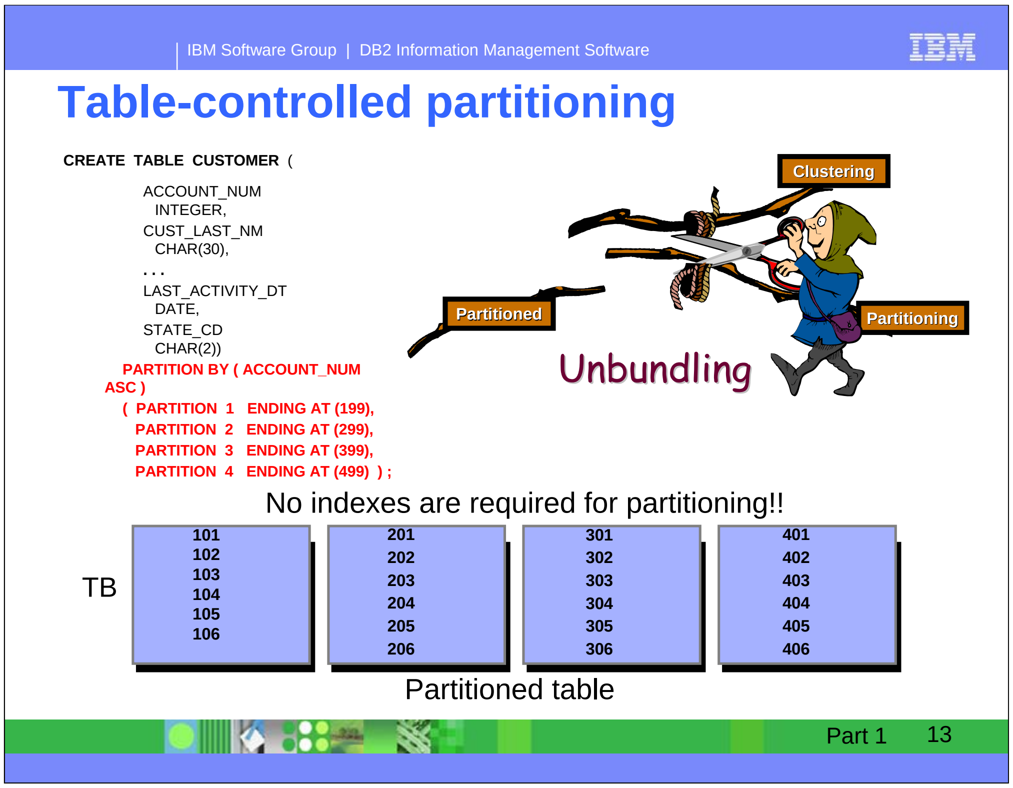
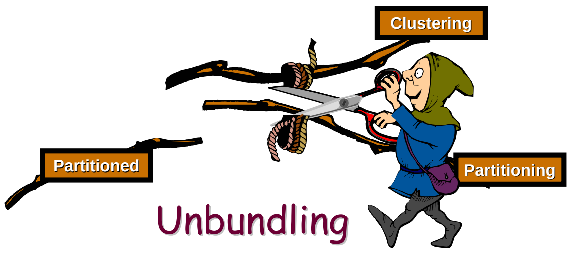


Table-controlled partitioning

```
CREATE  TABLE  CUSTOMER  (
          ACCOUNT_NUM
            INTEGER,
          CUST_LAST_NM
            CHAR(30),
          . . .
          LAST_ACTIVITY_DT
            DATE,
          STATE_CD
            CHAR(2))
  PARTITION BY ( ACCOUNT_NUM
ASC )
  (  PARTITION  1   ENDING AT (199),
     PARTITION  2   ENDING AT (299),
     PARTITION  3   ENDING AT (399),
     PARTITION  4   ENDING AT (499)  ) ;
```



No indexes are required for partitioning!!

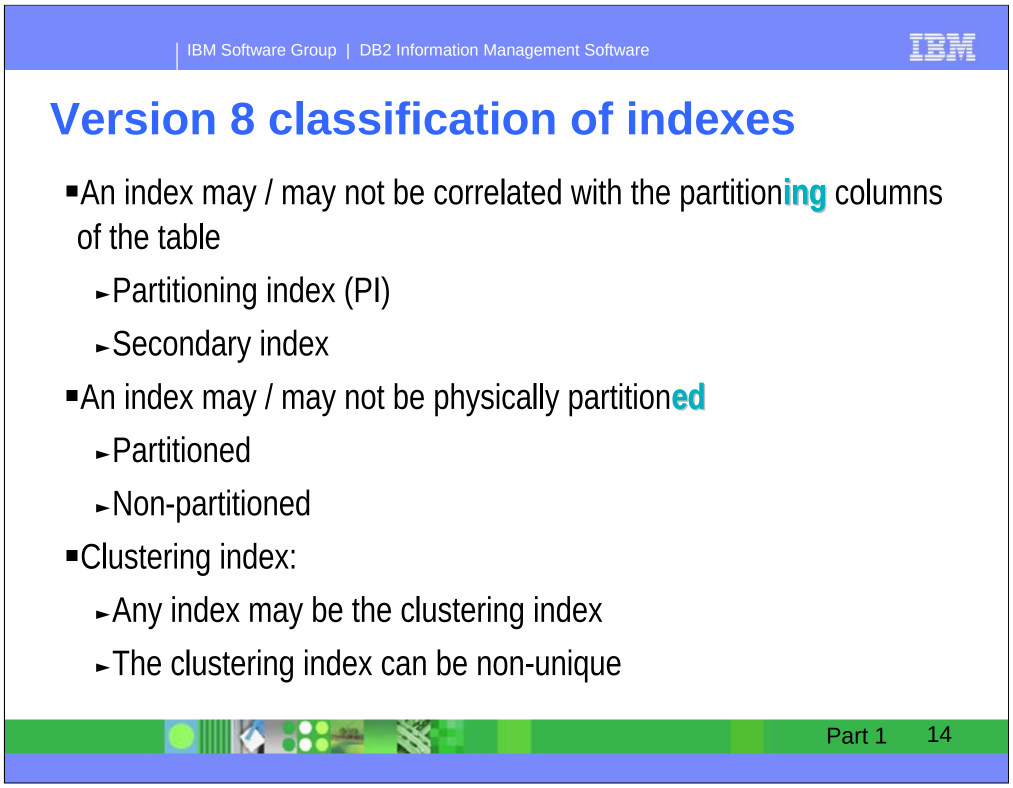
TB	101 102 103 104 105 106	201 202 203 204 205 206	301 302 303 304 305 306	401 402 403 404 405 406

Partitioned table

Before V8, only **index-controlled partitioning** was supported. With index-controlled partitioning, the partition boundaries are specified on the CREATE INDEX statement, when creating a partitioning index on the table. This results in a table that is unusable or incomplete until the partitioning index is created.
DB2 V8 introduces **table-controlled partitioning**.
That is, you specify the partition boundaries and number of partitions in your CREATE TABLE statement.

PARTITION BY is the keyword which initiates table-controlled partitioning.
Once the table definition has been completed successfully, the table is ready to use and you can insert your data. Indexes are no longer required for partitioned table spaces. If you attempt to create an index on this table using the VALUES keyword, an error is returned.

Both types of partitioning are supported with DB2 V8, but several new enhancements are only supported when using table-controlled partitioning.
You are encouraged to convert partitioned tables to use table-controlled partitioning. A non-disruptive method for doing this involves:
Creating a Data Partitioned Secondary Index (CREATE INDEX PARTITIONED) with DEFER YES. The CREATE INDEX PARTITIONED syntax triggers conversion to table-controlled partitioning, and the index is left in RBDP. There is no loss of availability.
Dropping the newly created (empty) index.
Alternatively, you can run the ALTER INDEX NOT CLUSTER statement on the partitioning index. That converts the partitoned table space to to table-controlled partitioning, and ALTER it back to CLUSTER immediately after.


IBM Software Group | DB2 Information Management Software

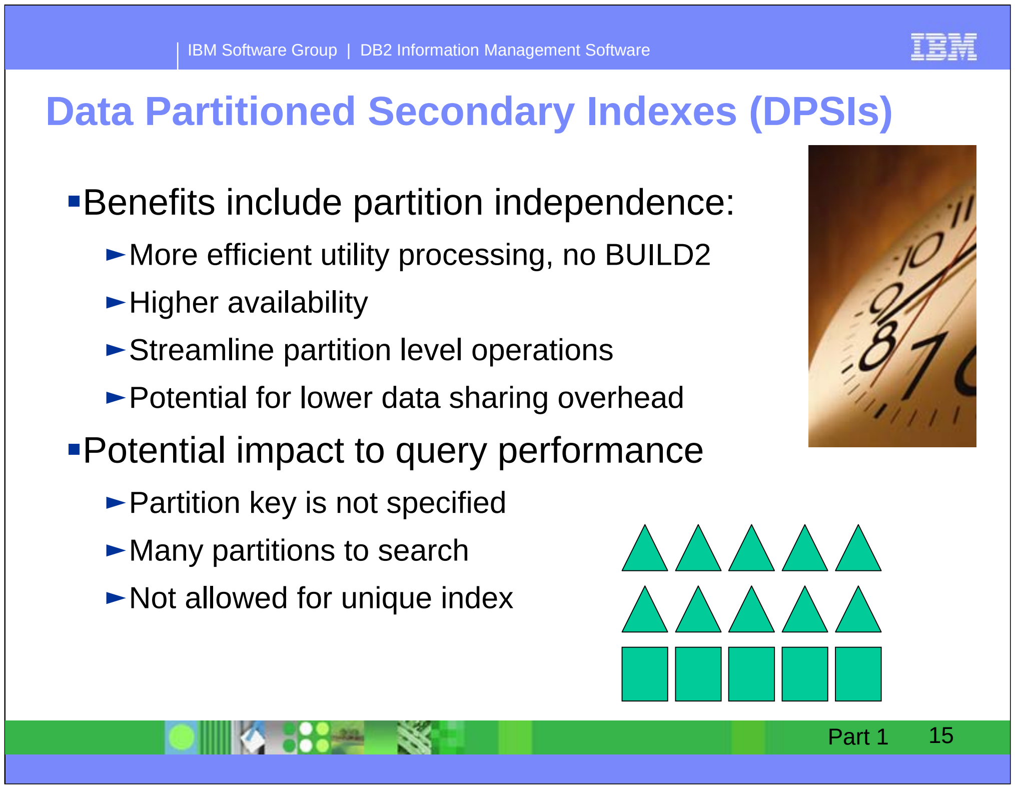
Version 8 classification of indexes

- An index may / may not be correlated with the partitioning columns of the table
  - Partitioning index (PI)
  - Secondary index
- An index may / may not be physically partitioned
  - Partitioned
  - Non-partitioned
- Clustering index:
  - Any index may be the clustering index
  - The clustering index can be non-unique

Part 1 14

In DB2 V7 you only have index-controlled partitioning to create a partitioned table. When using index-controlled partitioning, concepts such as partitioned, partitioning and clustering are intertwined. The index that defines the key ranges for the different partitions is the partitioning index, is also the clustering index.

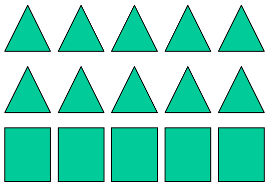
Any other index on an index-controlled partitioned table is called a secondary index, and it is non-partitioned, that is, not split up into multiple partitions (although you can define pieces for those indexes to improve I/O performance). Indexes on table-controlled partitioned tables can be classified as follows, based on whether or not an index is physically partitioned:
Partitioned index - made up of multiple physical partitions (not just index pieces)
Non-partitioned index - a single physical data set (or multiple pieces)
Based on whether or not the columns in the index correlate with the partitioning columns of the table (the partitioning columns are those specified in the PARTITION BY clause):
Partitioning index - The columns in the index are the same as (and have the same collating sequence), or start with the column(s) in the PARTITION BY clause of the CREATE TABLE statement.
Secondary index - Any index where the columns do not coincide with the partitioning columns of the table.
Based on whether or not the index determines the clustering of the data. Please note that when using table-controlled partitioning:
Clustering index - index determines data order in partitioned table.
Non-clustering index - index does not determine data order in the partitioned table.



IBM Software Group | DB2 Information Management Software

Data Partitioned Secondary Indexes (DPSIs)

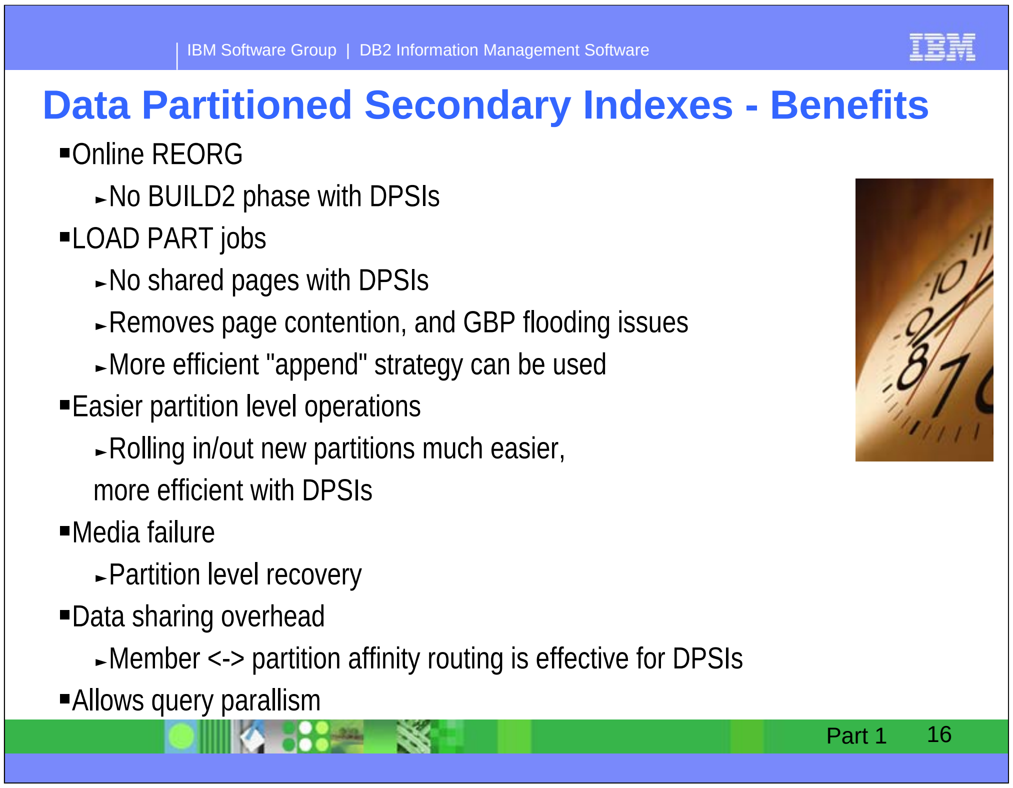
- Benefits include partition independence:
  - ►More efficient utility processing, no BUILD2
  - ►Higher availability
  - ►Streamline partition level operations
  - ►Potential for lower data sharing overhead
- Potential impact to query performance
  - ►Partition key is not specified
  - ►Many partitions to search
  - ►Not allowed for unique index



Part 1 15

DPSI benefits are substantial in terms of the ability to reduce contention on indexes in SQL and utilities that process a range of partitions. The recovery time can be improved by an order of magnitude or more. Eliminating the BUILD2 phase of online reorg is a big improvement.

This option does not fit some situations, such as when we must search many partitions or if the index must be unique.



Data Partitioned Secondary Indexes - Benefits

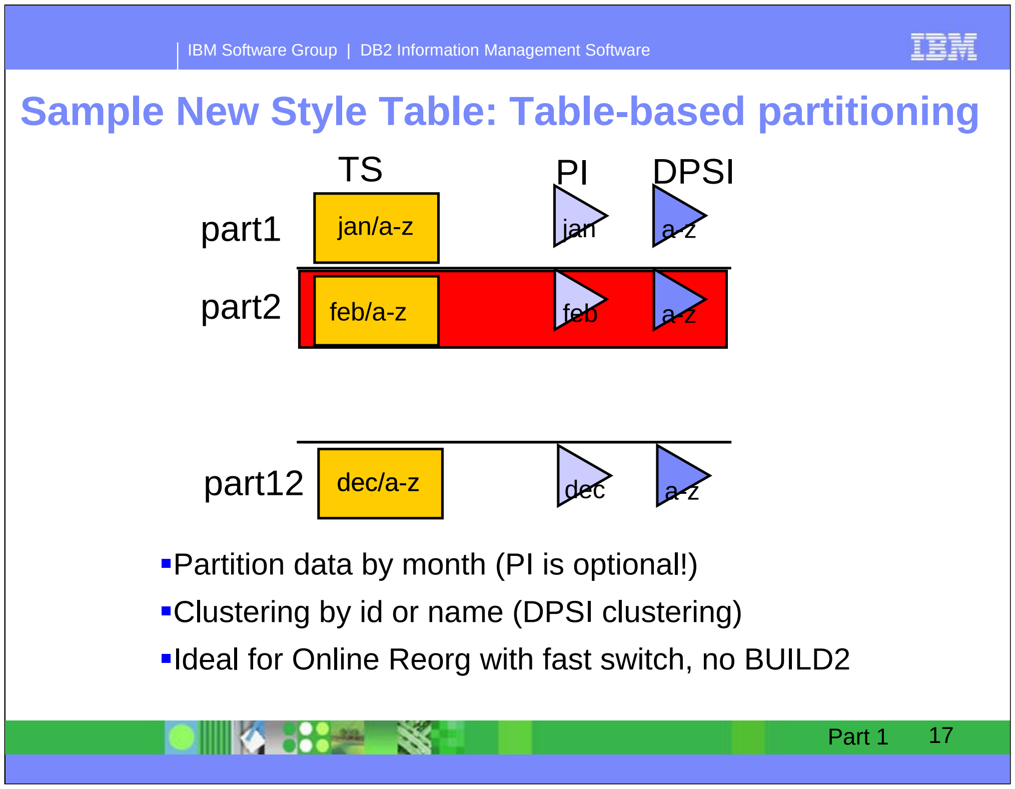
- Online REORG
  - No BUILD2 phase with DPSIs
- LOAD PART jobs
  - No shared pages with DPSIs
  - Removes page contention, and GBP flooding issues
  - More efficient "append" strategy can be used
- Easier partition level operations
  - Rolling in/out new partitions much easier, more efficient with DPSIs
- Media failure
  - Partition level recovery
- Data sharing overhead
  - Member <-> partition affinity routing is effective for DPSIs
- Allows query parallism

Data partitioned secondary indexes have been invented to be able to overcome some of the negative impacts former (non-partitioned) secondary indexes have. Therefore, if you use a DPSI instead of a NPSI (non-partitioned partitioning indexes) for your partitioned tables, you will encounter the following benefits:
Using DPSIs eliminates the need for a BUILD2 phase. There is no BUILD2 phase processing for DPSIs. Because keys for a given data partition reside in a single DPSI partition, a simple substitution of the index partition newly built by REORG for the old partition is all that is needed. If all indexes on a table are partitioned (partitioned PI or DPSIs), the BUILD2 phase of REORG is eliminated.

Using DPSIs also eliminates LOAD PART job contention.. There is no contention between LOAD PART jobs during DPSI processing. This is because there are no shared pages between partitions on which to contend. Thus if all indexes on a table are partitioned, index page contention is eliminated.

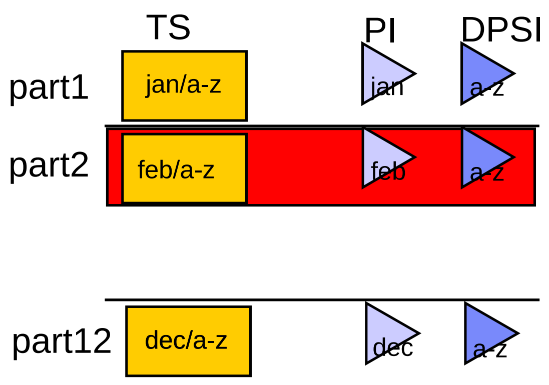
Also note that during parallel LOAD PART job execution, each LOAD job inserts (loads) DPSI keys into a separate index structure, in key order. This allows the insertion logic to follow an efficient append strategy.

DPSIs also improve the recovery characteristics of your system. DPSIs can be copied and recovered at the partition level. Individual partitions can be rebuilt in parallel to achieve a fast rebuild of the entire index.
However, beware of queries that do not have partitoning column predicates. They have to scan all parts of the DPSI



IBM Software Group | DB2 Information Management Software

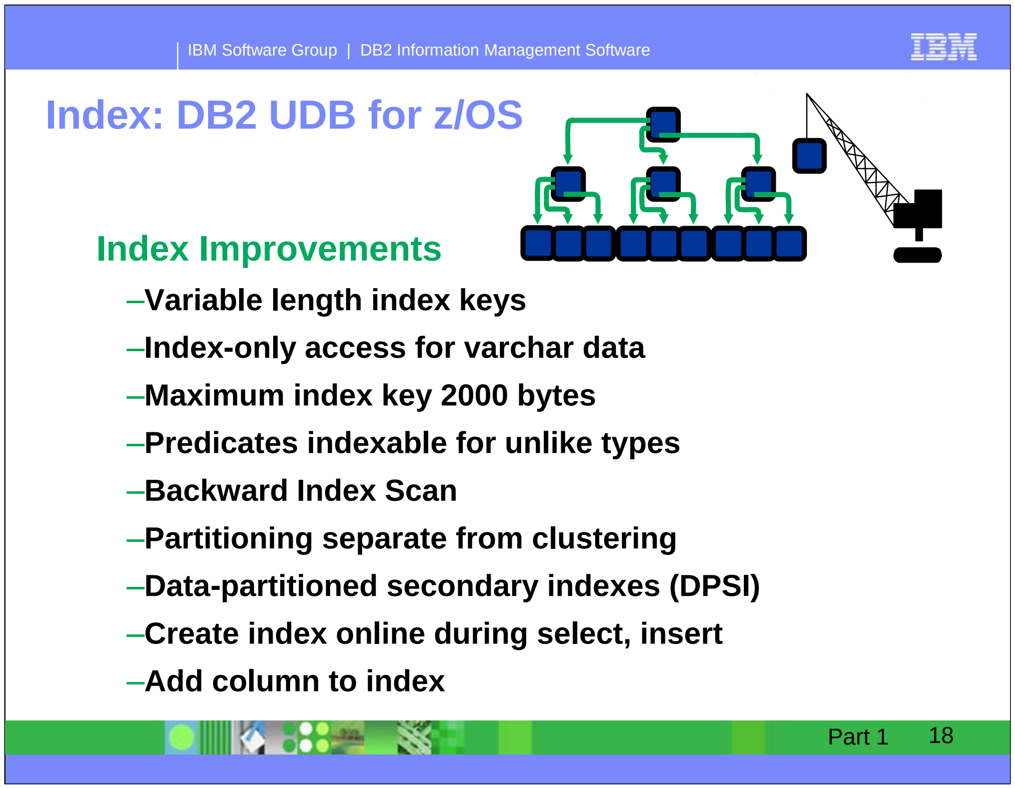
Sample New Style Table: Table-based partitioning



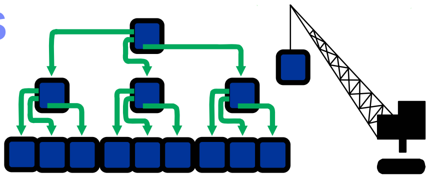
- Partition data by month (PI is optional!)
- Clustering by id or name (DPSI clustering)
- Ideal for Online Reorg with fast switch, no BUILD2

Part 1 17

This is an example of the new style table, with table-based partitioning, rather than index-based partitioning. Note that the data is partitioned by month. An index is not required for the partitioning. Clustering for the data is by the id or name within each partition of the DPSI. This an ideal organization for online reorg of a single partition. The BUILD2 phase is not required. If the month is not provided, a name search using the DPSI may need to search in every partition.



Index: DB2 UDB for z/OS



Index Improvements

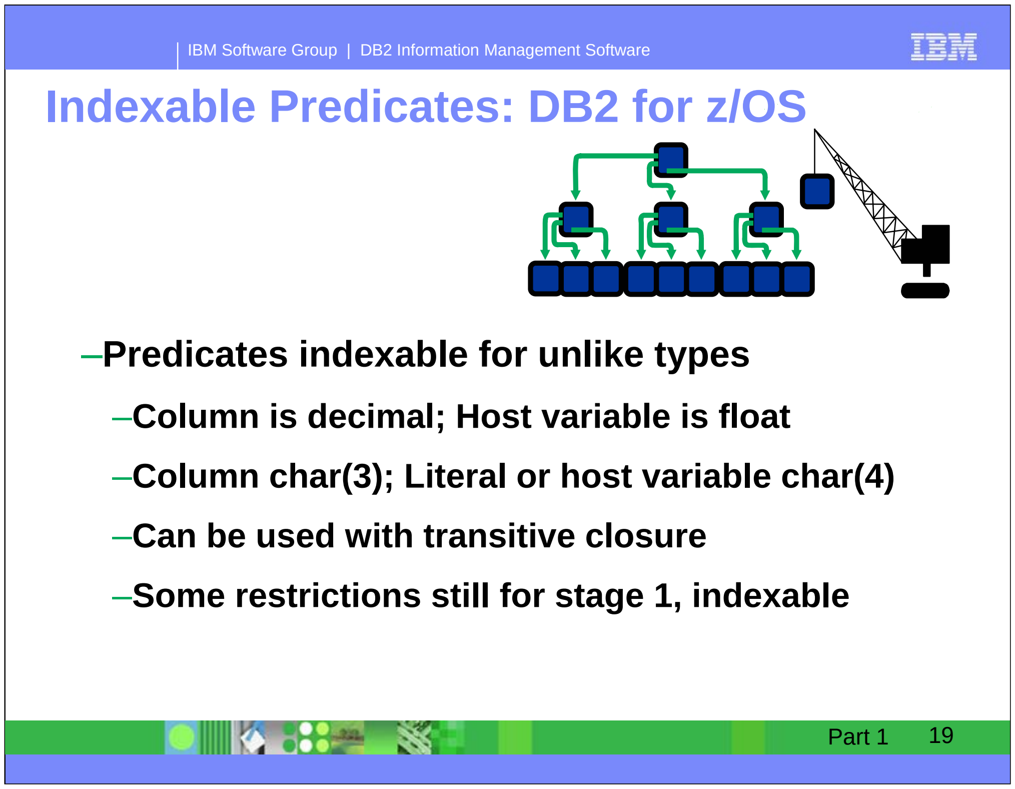
- –Variable length index keys
- –Index-only access for varchar data
- –Maximum index key 2000 bytes
- –Predicates indexable for unlike types
- –Backward Index Scan
- –Partitioning separate from clustering
- –Data-partitioned secondary indexes (DPSI)
- –Create index online during select, insert
- –Add column to index

DB2 V8 provides many new opportunities for improving index processing, rebuilding the architecture for indexes.

We are able to use indexes more effectively, reducing the space in variable-length indexes, being able to have index-only access with variable-length data and being able to use the index when the predicates do not match.

In some cases, such as backward index scans or partitioning, we will be able to work as efficiently with one less index. Being able to eliminate an index will improve the insert, delete, LOAD, REORG and update processing.

We have more flexibility in indexes, with longer index keys, the ability to partition secondary indexes and the ability to have more effective clustering.



IBM Software Group | DB2 Information Management Software

Indexable Predicates: DB2 for z/OS

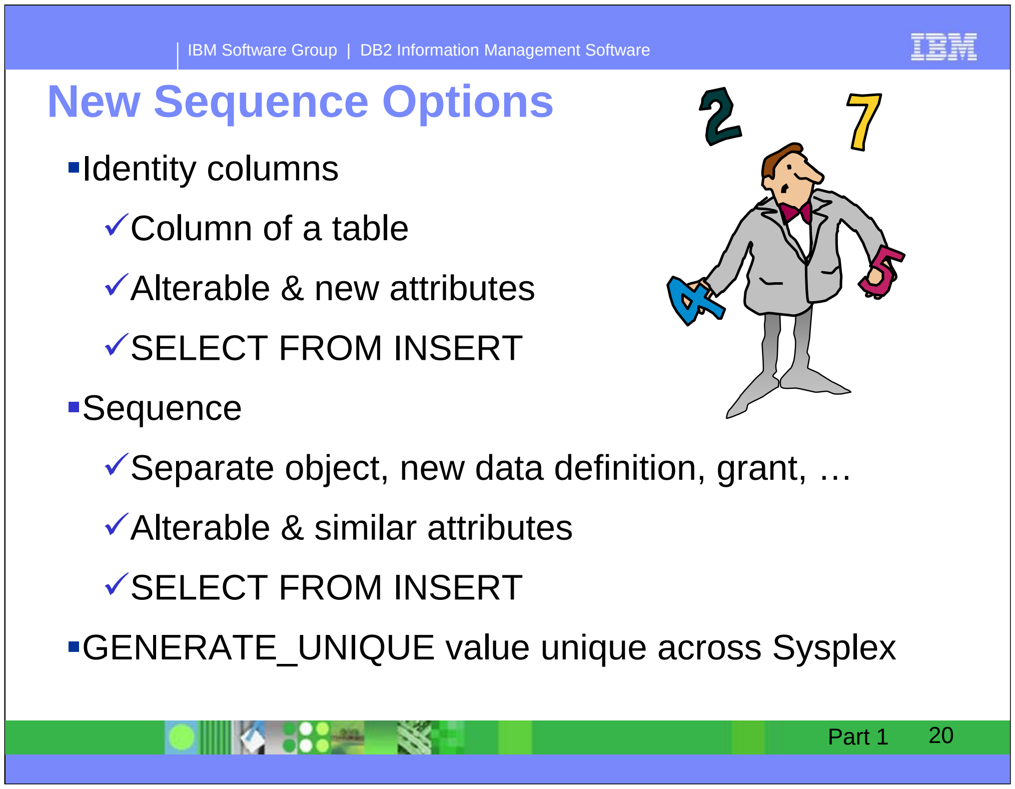
- –Predicates indexable for unlike types
  - –Column is decimal; Host variable is float
  - –Column char(3); Literal or host variable char(4)
  - –Can be used with transitive closure
  - –Some restrictions still for stage 1, indexable

Part 1 19

The most common mismatches for data types come with languages like Java, C++ and C and decimal data. Often the comparison is from a floating point host variable to a decimal column.

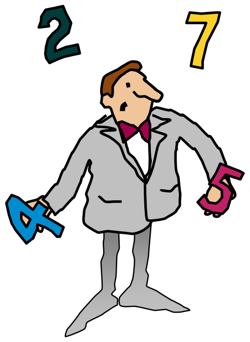
A second type of mismatch that is very common is to have a literal or host variable with a character column length greater than that of the column.

For both of these cases, the result was often poor performance because of the inability to use an index. While there are still some restrictions, performance is expected to improve substantially for many customers.



New Sequence Options

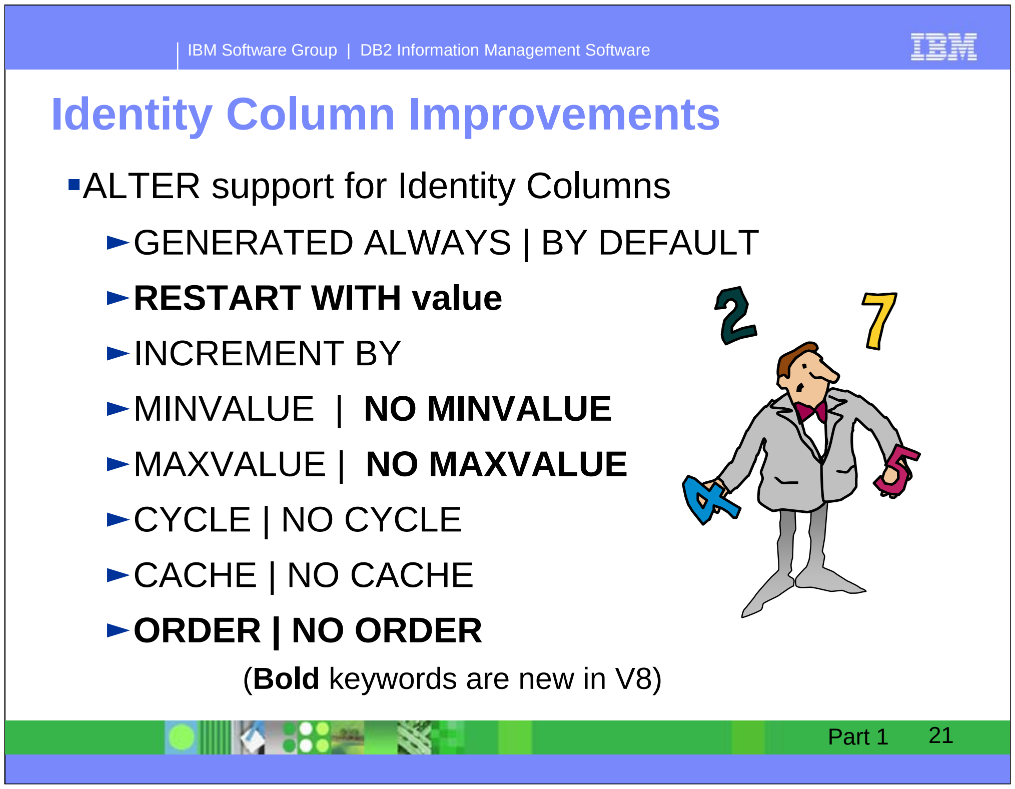
- Identity columns
  - ✓Column of a table
  - ✓Alterable & new attributes
  - ✓SELECT FROM INSERT
- Sequence
  - ✓Separate object, new data definition, grant, …
  - ✓Alterable & similar attributes
  - ✓SELECT FROM INSERT
- GENERATE_UNIQUE value unique across Sysplex



Identity columns were introduced in DB2 UDB for OS/390 Version 6, but Version 8 provides needed enhancements for usability. New attributes are added for sequences and the attributes can be altered, so that operations like LOAD are easier. A new SELECT FROM INSERT statement provides a standard way to retrieve the result of inserting an identity column.

Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences. Sequences are helpful in porting Oracle applications to DB2, while identity columns are more like the SQL Server construct.

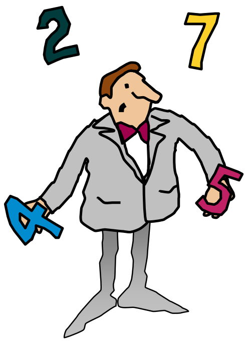
A new GENERATE_UNIQUE function is also added to generate a CHAR(13) FOR BIT DATA that is unique across a data sharing group or Parallel Sysplex.



Identity Column Improvements

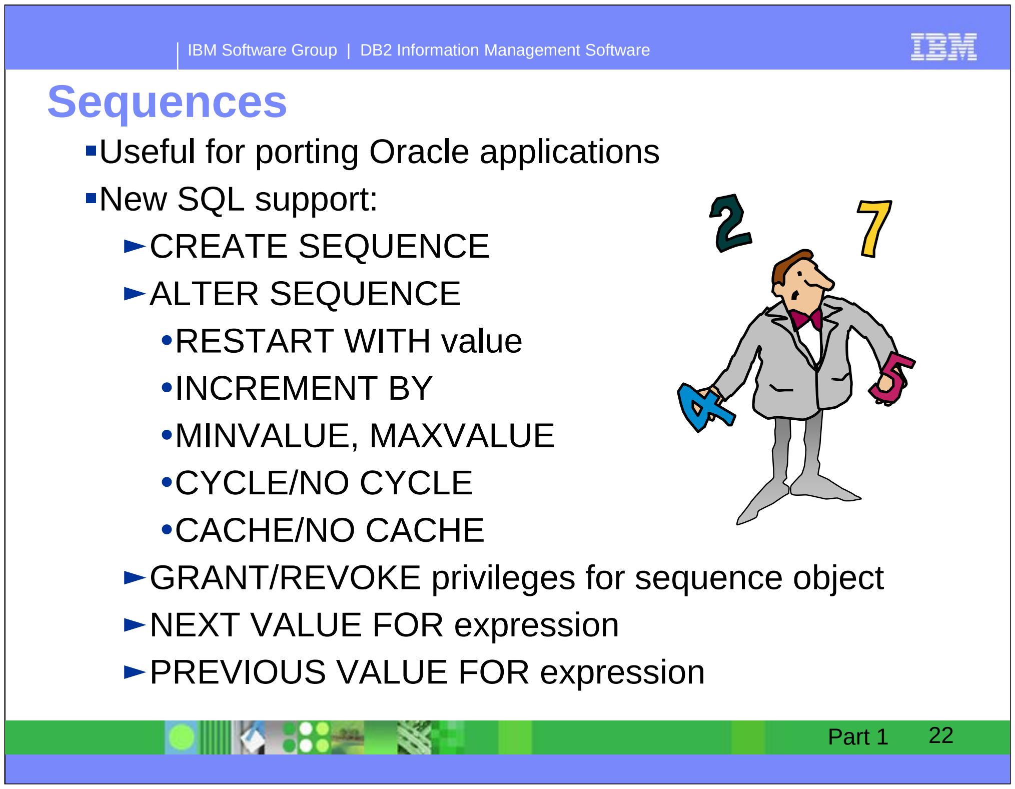
- ALTER support for Identity Columns
  - GENERATED ALWAYS | BY DEFAULT
  - **RESTART WITH value**
  - INCREMENT BY
  - MINVALUE | **NO MINVALUE**
  - MAXVALUE | **NO MAXVALUE**
  - CYCLE | NO CYCLE
  - CACHE | NO CACHE
  - **ORDER | NO ORDER**

(**Bold** keywords are new in V8)



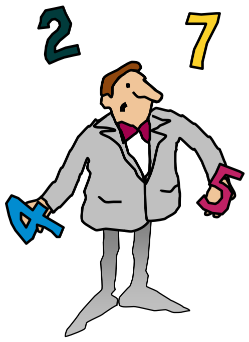
When identity columns were provided with Version 6, customers identified some important enhancements that were needed. Many customers asked for the ability to ALTER the identity column attributes, especially the ability to change the GENERATED option. This added flexibility will allow identity columns to be used in many more situations.

New identity column attributes can be specified to aid porting from other vendor implementations: RESTART WITH, NO MINVALUE, NO MAXVALUE, NO ORDER, ORDER. The identity column is a style often used by those who use Sybase.



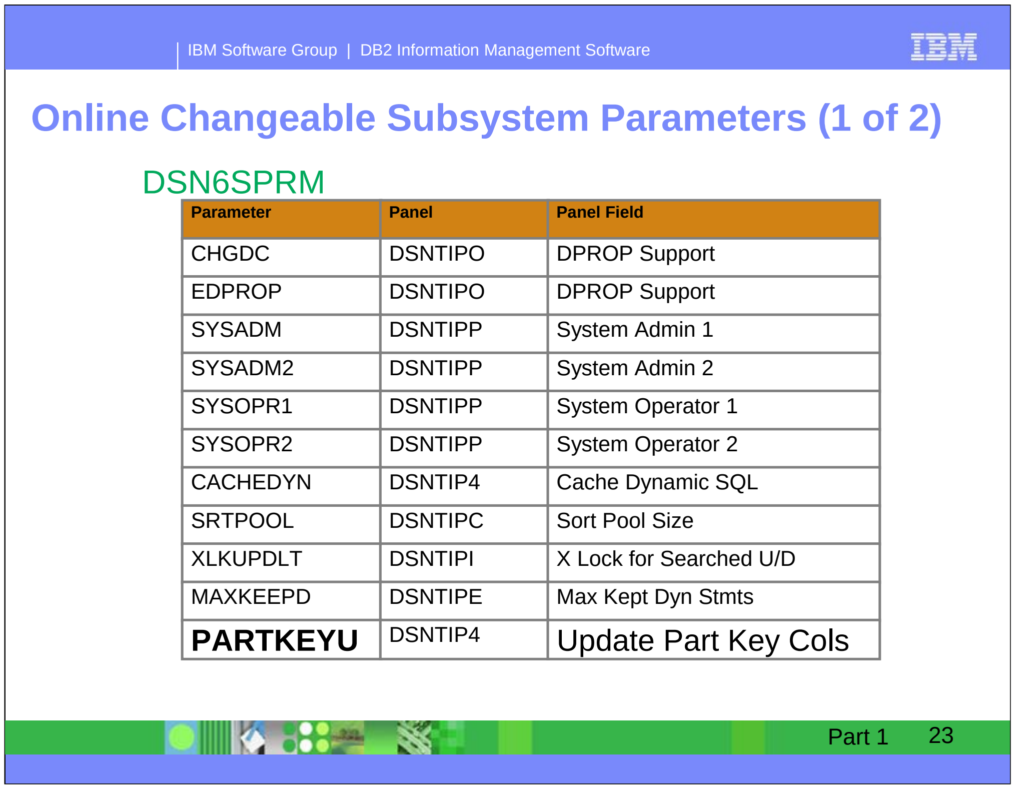
Sequences

- Useful for porting Oracle applications
- New SQL support:
  - CREATE SEQUENCE
  - ALTER SEQUENCE
    - RESTART WITH value
    - INCREMENT BY
    - MINVALUE, MAXVALUE
    - CYCLE/NO CYCLE
    - CACHE/NO CACHE
  - GRANT/REVOKE privileges for sequence object
  - NEXT VALUE FOR expression
  - PREVIOUS VALUE FOR expression



Sequences, like identity columns, provide an incremented counter within the DBMS. While the identity column is in a table, the sequence is a separate, standalone object. Most of the attributes of a sequence are very similar to those of an identity column. New data definition SQL statements are added to CREATE, ALTER and DROP sequences.

Sequences are helpful in porting Oracle applications to DB2. The language to use a sequence is to specify NEXT VALUE FOR expression or PREVIOUS VALUE FOR expression in your SQL statement.


IBM Software Group | DB2 Information Management Software

Online Changeable Subsystem Parameters (1 of 2)

DSN6SPRM

Parameter	Panel	Panel Field
CHGDC	DSNTIPO	DPROP Support
EDPROP	DSNTIPO	DPROP Support
SYSADM	DSNTIPP	System Admin 1
SYSADM2	DSNTIPP	System Admin 2
SYSOPR1	DSNTIPP	System Operator 1
SYSOPR2	DSNTIPP	System Operator 2
CACHEDYN	DSNTIP4	Cache Dynamic SQL
SRTPOOL	DSNTIPC	Sort Pool Size
XLKUPDLT	DSNTIPI	X Lock for Searched U/D
MAXKEEPD	DSNTIPE	Max Kept Dyn Stmts
PARTKEYU	DSNTIP4	Update Part Key Cols

Part 1 23

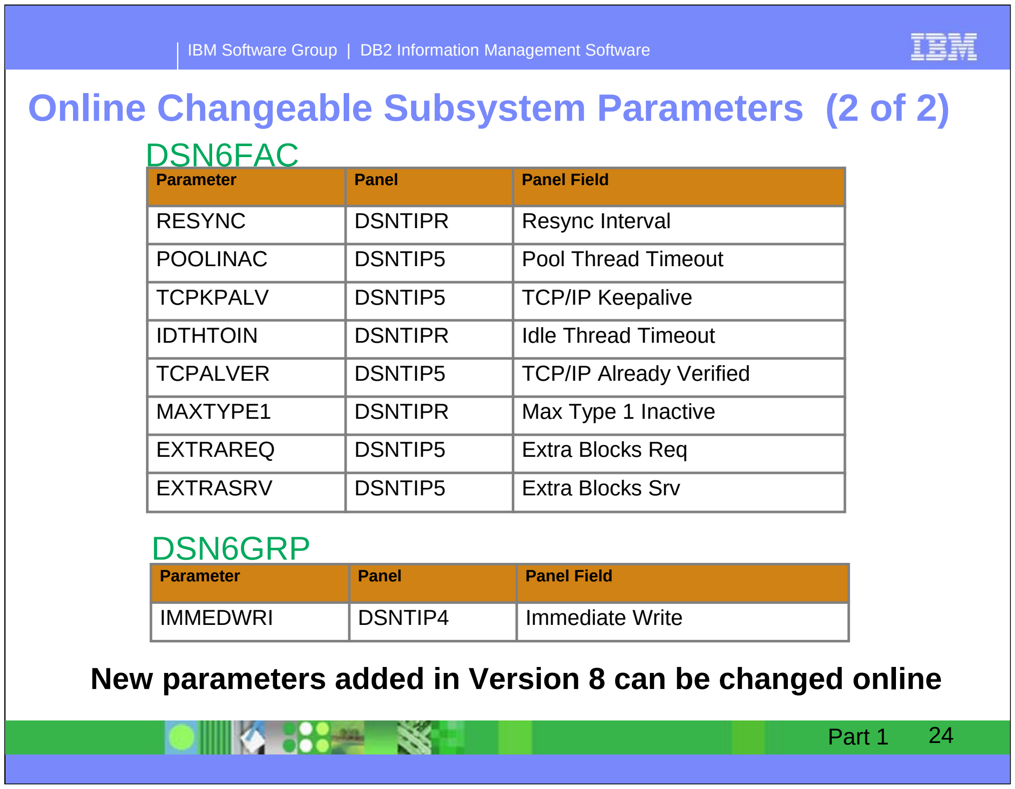
SYSADM / SYSOPR Parms: Install SYSADM auth required to change these online.

PARTKEYU : After online change, update of value in a partitioning key column may result in -904 SQL code. **PARTKEYU Avoid drain when updating values in partitioning key columns.** Currently, if the update requires moving the data row from one partition to another, DB2 tries to take exclusive control of the objects to perform the update by acquiring DRAIN locks. Because of this, no other application can access the range of partition affected by update of values in partitioning key columns.

XLKUPDLT: Changes don't affect currently running statements.

MAXKEEPD Changes take effect after next COMMIT.

CACHEDYN Changes don't affect currently running statements.



Online Changeable Subsystem Parameters (2 of 2)

DSN6FAC

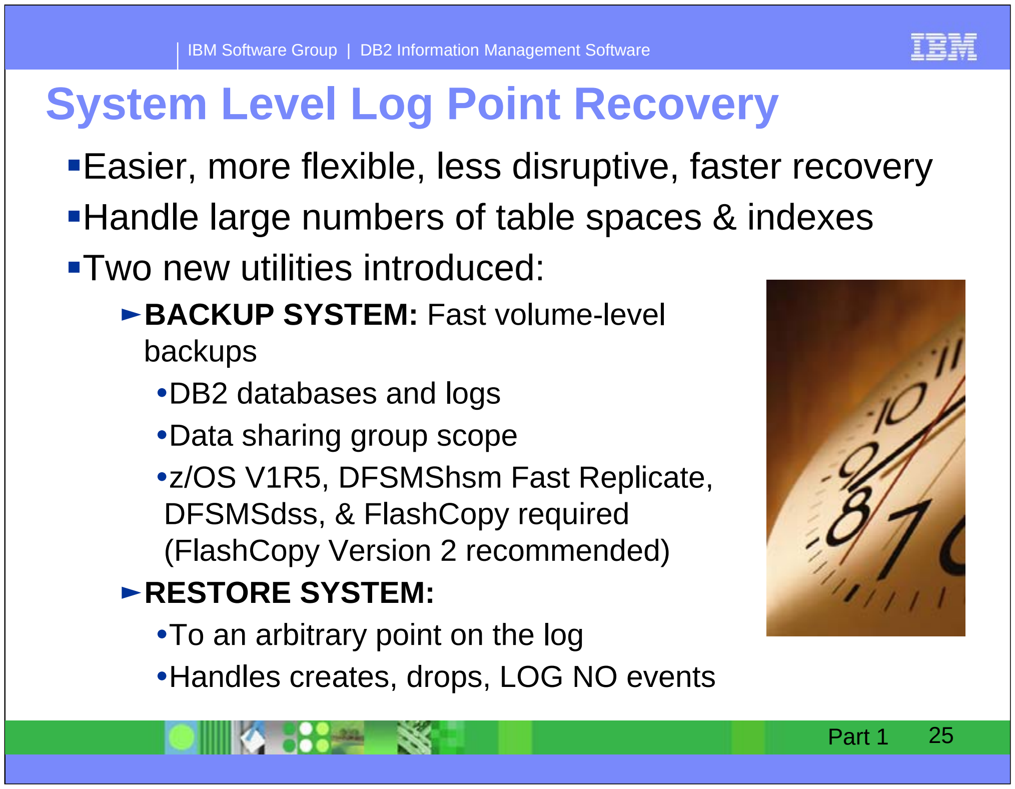
Parameter	Panel	Panel Field
RESYNC	DSNTIPR	Resync Interval
POOLINAC	DSNTIP5	Pool Thread Timeout
TCPKPALV	DSNTIP5	TCP/IP Keepalive
IDTHTOIN	DSNTIPR	Idle Thread Timeout
TCPALVER	DSNTIP5	TCP/IP Already Verified
MAXTYPE1	DSNTIPR	Max Type 1 Inactive
EXTRAREQ	DSNTIP5	Extra Blocks Req
EXTRASRV	DSNTIP5	Extra Blocks Srv

DSN6GRP

Parameter	Panel	Panel Field
IMMEDWRI	DSNTIP4	Immediate Write

New parameters added in Version 8 can be changed online

About 20 additional parameters that existed in V7 can be changed now, in addition to the roughly 60 which could be changed in Version 7 and about 25 new parameters added in Version 8 which can be changed online.
The process for changing parameters is the same as in Version 7, except that these additional parameters are changed if the new set of system parameters differ.


IBM Software Group | DB2 Information Management Software

System Level Log Point Recovery

- Easier, more flexible, less disruptive, faster recovery
- Handle large numbers of table spaces & indexes
- Two new utilities introduced:
  - **BACKUP SYSTEM:** Fast volume-level backups
    - DB2 databases and logs
    - Data sharing group scope
    - z/OS V1R5, DFSMShsm Fast Replicate, DFSMSdss, & FlashCopy required (FlashCopy Version 2 recommended)
  - **RESTORE SYSTEM:**
    - To an arbitrary point on the log
    - Handles creates, drops, LOG NO events

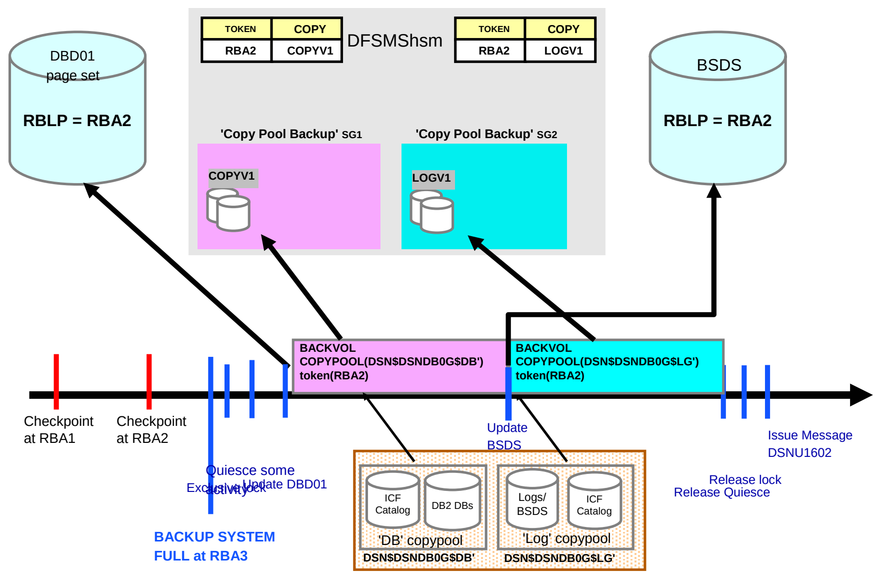
Part 1 25

Enhancements to system-level log point recovery for DB2 provide improved usability, more flexibility, and faster recovery. You can now recover your data to any point on the log, regardless of whether you have uncommitted units of work. As a result, data recovery time improves significantly for large DB2 systems that contain many thousands of objects. Two new utilities provide system-level point-in-time recovery:

The BACKUP SYSTEM utility provides fast volume-level copies of DB2 databases and logs. It relies on new DFSMShsm services in z/OS Version 1 Release 5 that automatically keep track of the volumes that need to be copied. BACKUP SYSTEM is less disruptive than using the SET LOG SUSPEND command for copy procedures. An advantage for data sharing is that BACKUP SYSTEM is group scope.

The RESTORE SYSTEM utility recovers a DB2 system to an arbitrary point in on the log. RESTORE SYSTEM automatically handles any creates, drops, and LOG NO events that might have occurred between the backup and the recovery point.

BACKUP SYSTEM operation



Backup system will do the following:
Obtain an exclusive lock (new lock type) to ensure that only one BACKUP SYSTEM process is running at a time.
Quiesce some system activities but much less disruptive than -set log suspend
Update the DBD01 header page and write the page to DASD.

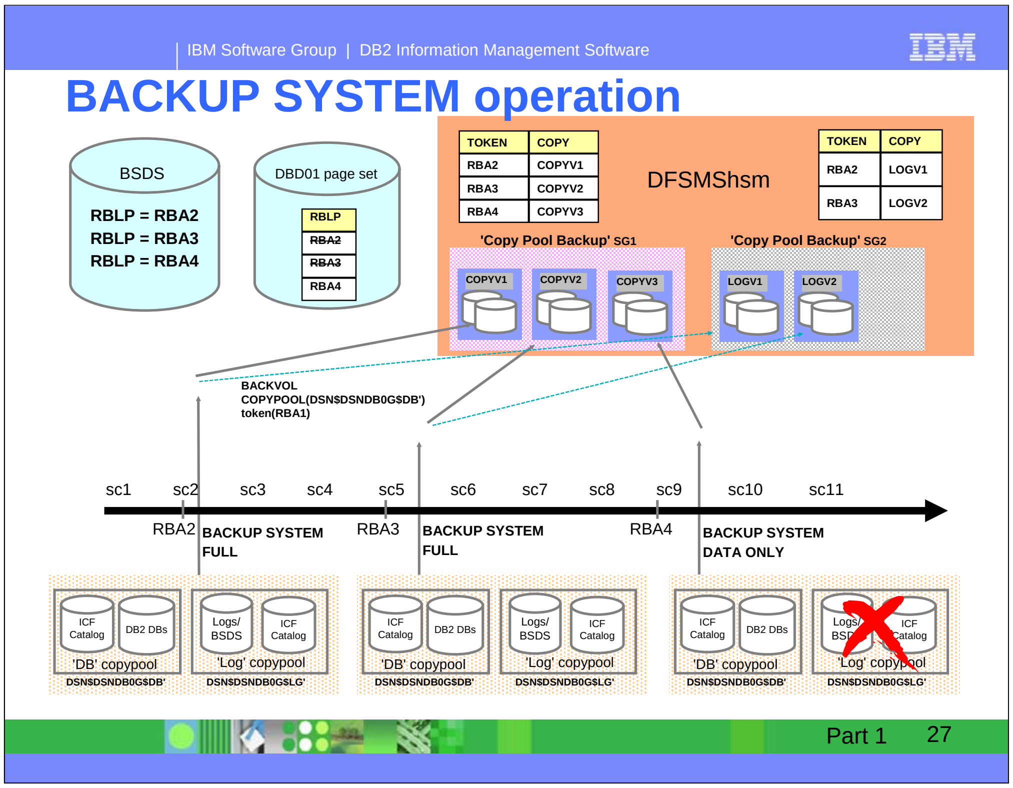
Take full volume DASD-to-DASD copies of the 'DB' COPYPOOL

Update the BSDS with the system copy information

If "full" system backups were requested, take full volume copies of the 'LG' COPYPOOL.

Resume all system activities that were quiesced above.

Release the lock.


IBM Software Group | DB2 Information Management Software

BACKUP SYSTEM operation

TOKEN	COPY
RBA2	COPYV1
RBA3	COPYV2
RBA4	COPYV3

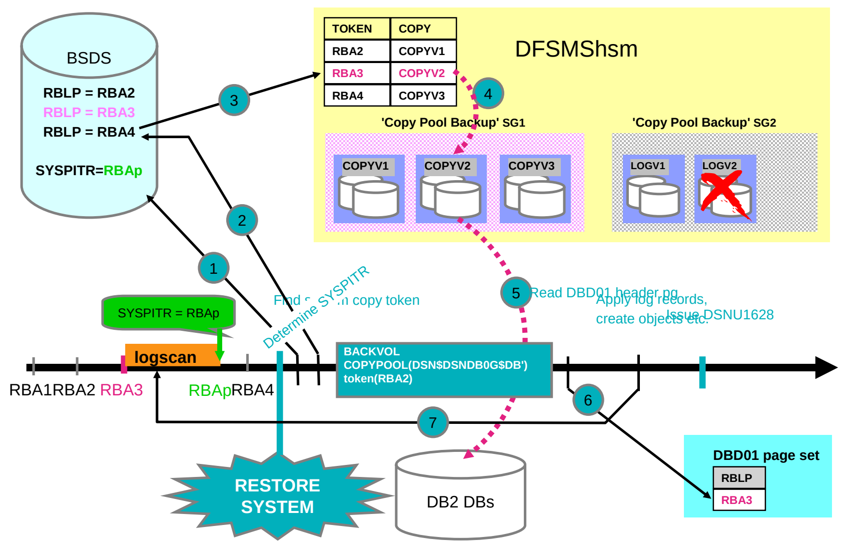
TOKEN	COPY
RBA2	LOGV1
RBA3	LOGV2

As mentioned before, you can use the new BACKUP SYSTEM utility to back up an entire DB2 subsystem or an entire DB2 data sharing group with a single command. The BACKUP SYSTEM utility has two options that you can specify:

FULL: In this case, the backup contains both the logs and databases. This is referred to as a full system backup. When taking a FULL backup, the copies of the log are always taken after the database copies.

DATA ONLY: This type of copy only contains the “database” data. Notice that a BACKUP SYSTEM DATA ONLY does NOT copy the “log” data. Such a copy is referred to as a data-only system backup.

RESTORE SYSTEM operation



Note that only the database COPYPOOL is restored

You can use RESTORE SYSTEM to restore a system to an arbitrary point in time. It is sufficient to have DATA ONLY backups (provided the logs are available to roll forward from).

To be able to run the RESTORE SYSTEM utility, the system has to be in System Recover Pending mode. A DB2 system goes into System Recover Pending mode after a conditional restart is performed, using a special type of conditional restart record called a PITR conditional restart control record (CRCR).

After the conditional restart completes the system enters into System Recover Pending mode. This means that:
Only the RESTORE SYSTEM utility is allowed to execute.

The DB2 data remains unavailable until the RESTORE SYSTEM utility has completed.

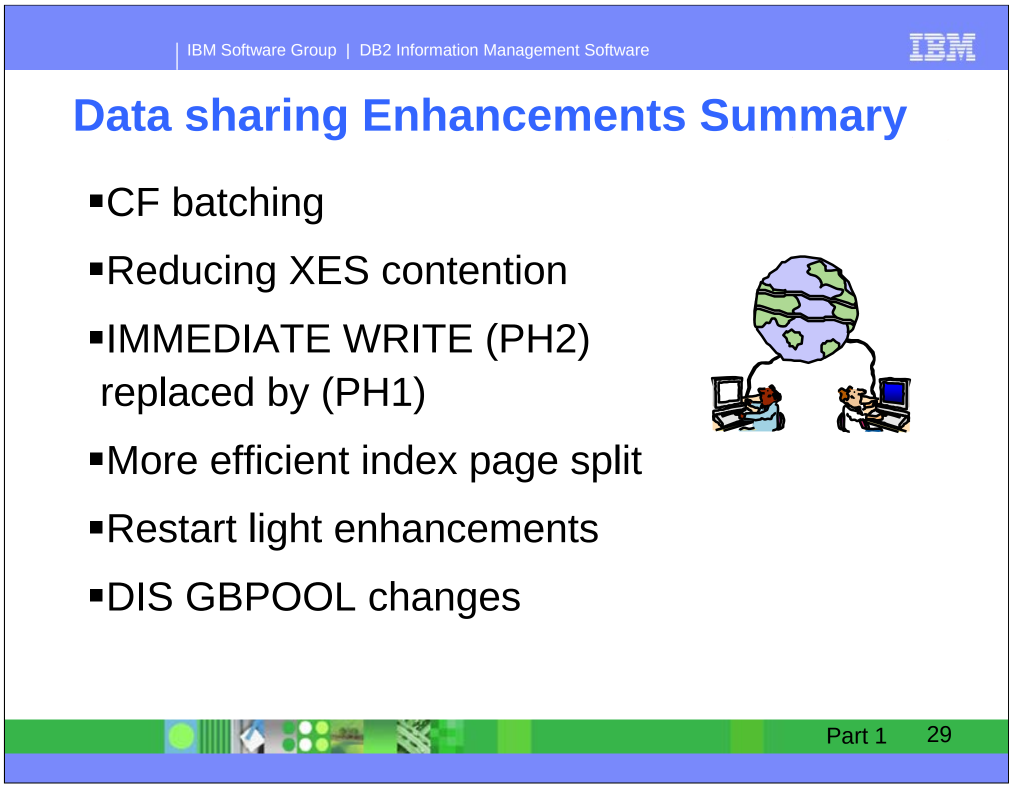
DB2 has to be recycled to reset the System Recover Pending mode.
The visual above shows roughly the steps which DB2 performs if you run the RESTORE SYSTEM utility (without the LOGONLY option):
The PITR CRCR determines the log truncation point

Find the BACKUP SYSTEM information in the BSDS that immediately prior to the log truncation point.

Restore the volume copies for the 'DB' pool

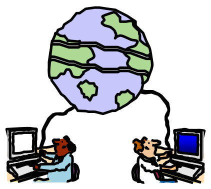
The DBD01 header page information is used to determine the starting point for the log scan.

Perform a 'modified' forward log scan until the log truncation point is reached

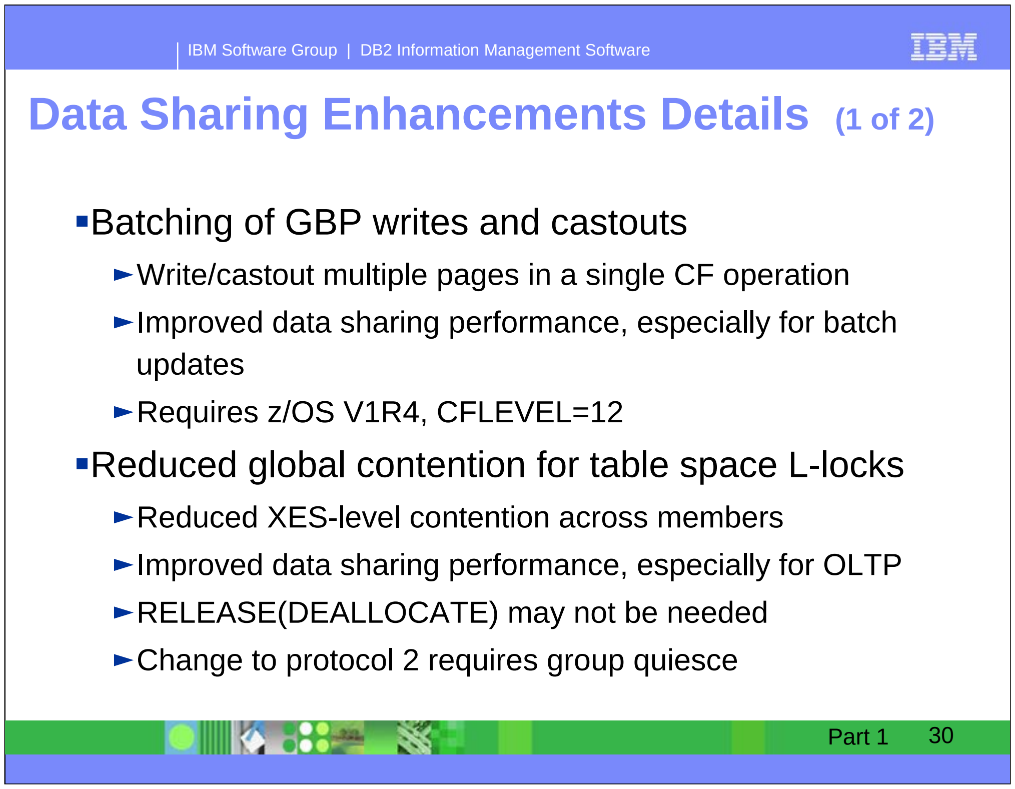


Data sharing Enhancements Summary

- CF batching
- Reducing XES contention
- IMMEDIATE WRITE (PH2) replaced by (PH1)
- More efficient index page split
- Restart light enhancements
- DIS GBPOOL changes



There are many changes for data sharing in this new version, improving performance, availability and usability. Being able to batch requests to the coupling facility improves performance. The new locking protocol reduces contention and improves performance. The more efficient page split helps in situations where there are large numbers of index page splits. The improvements in restart light allow indoubt units of work to be resolved.



Data Sharing Enhancements Details (1 of 2)

- Batching of GBP writes and castouts
  - Write/castout multiple pages in a single CF operation
  - Improved data sharing performance, especially for batch updates
  - Requires z/OS V1R4, CFLEVEL=12
- Reduced global contention for table space L-locks
  - Reduced XES-level contention across members
  - Improved data sharing performance, especially for OLTP
  - RELEASE(DEALLOCATE) may not be needed
  - Change to protocol 2 requires group quiesce

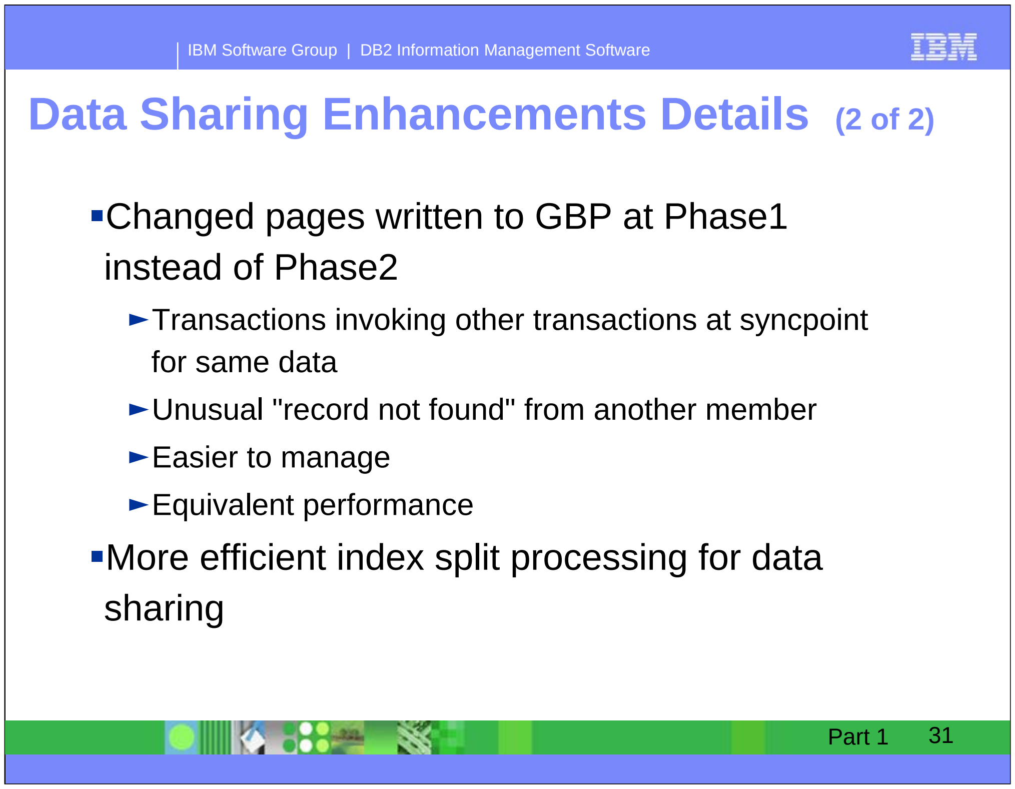
Batching of GBP writes and castouts
Write/castout multiple pages in a single CF operation

Improved data sharing performance, especially for batch updates

Requires z/OS R4, CFLEVEL=12
Reduced global contention for table space L-locks
IX/IX and IX/IS TS locks no longer hit XES-level contention across members

Improved data sharing performance, especially for OLTP

Recommendation for RELEASE(DEALLOCATE) can be softened
New locking protocol enacted only with New Function Mode, and requires quiesce of data sharing group



Data Sharing Enhancements Details (2 of 2)

- Changed pages written to GBP at Phase1 instead of Phase2
  - Transactions invoking other transactions at syncpoint for same data
  - Unusual "record not found" from another member
  - Easier to manage
  - Equivalent performance
- More efficient index split processing for data sharing

Changed pages written to GBP at Phase1 instead of Phase2

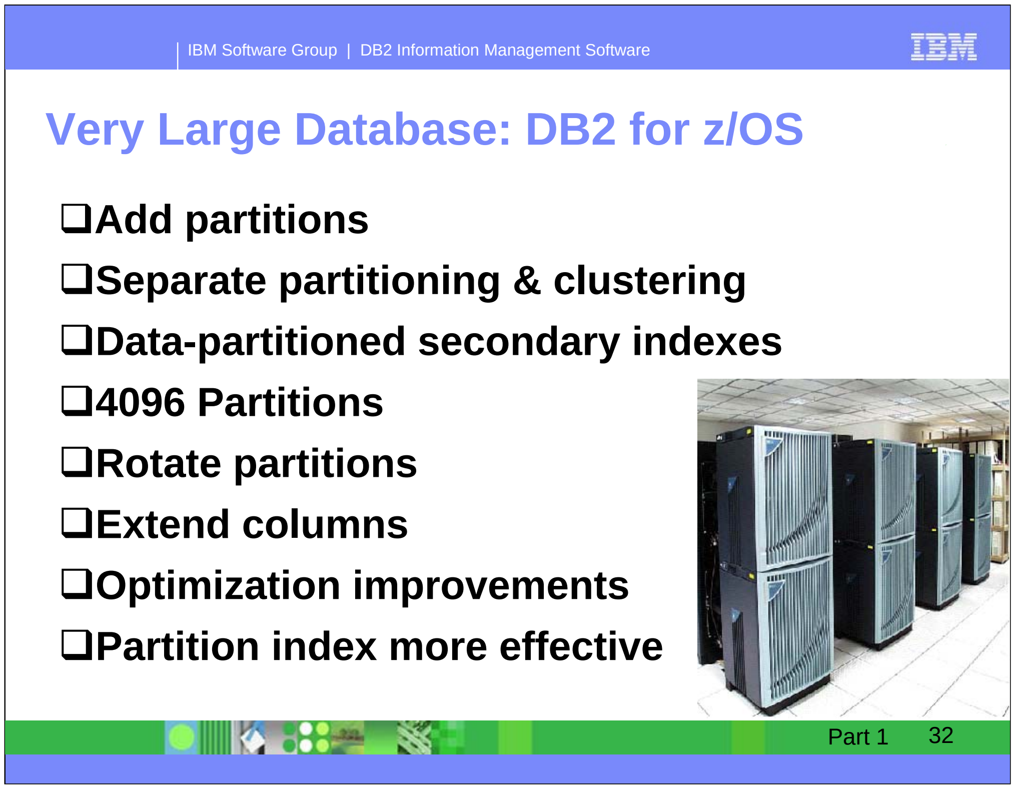
Some Tx Managers spawn other transactions at syncpoint

Spawned tx can encounter "record not found" if it tries to read originating tx's update from another member (rare, but a few customers have reported it)

Moving writes up to Phase1 by default removes need to monitor for this and to set IMMEDWRITE PH1 Zparm or Bind option if needed

Equivalent performance for Ph1 vs. Ph2 writes

More efficient index split processing for data sharing



IBM Software Group | DB2 Information Management Software

IBM

Very Large Database: DB2 for z/OS

- ❑Add partitions
- ❑Separate partitioning & clustering
- ❑Data-partitioned secondary indexes
- ❑4096 Partitions
- ❑Rotate partitions
- ❑Extend columns
- ❑Optimization improvements
- ❑Partition index more effective

Part 1 32

Very large databases face the combined challenge of very high performance needs, continuous availability and complexity. Improvements in scale and flexibility are more important in this area. Being able to have more partitions and to add them with ALTER are a big improvement.

Often it is useful to partition by date, so that we can archive or delete an entire partition, but processing will be much more efficient with another clustering order, such as by customer. Before this change, the clustering order was the same as the partitioning. This flexibility offers many opportunities for improved performance and availability.

Some customers have an index that is used only for partitioning the data or have extra columns at the beginning of the index. Being able to avoid the extra index or columns can improve our efficiency a lot.

For these very large tables, the ability to have more partitions, to add new partitions and to be able to rotate partitions is crucial.