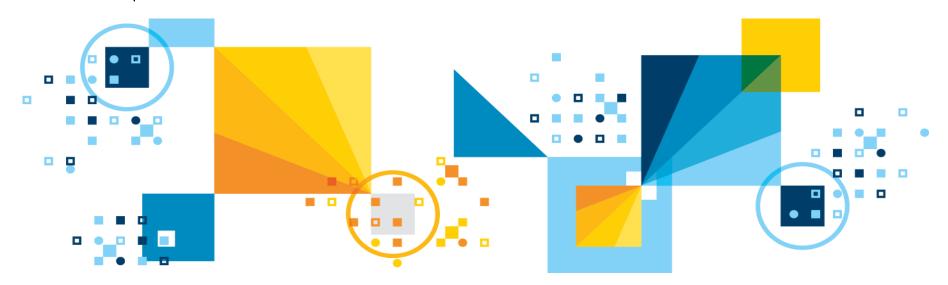


DB2 Partitioning Features

Module ID 10109

Length 1 hour + 1 hour Hands on Lab



For questions about this presentation contact askdata@ca.ibm.com



Disclaimer

© Copyright IBM Corporation 2015. All rights reserved.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM, the IBM logo, ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.



Module Information

- You should have completed or acquired the necessary knowledge for the following modules in order to complete this module:
 - DB2 Fundamentals
- After completing this module, you should be able to:
 - Explain the concepts of:
 - Database Partitioning
 - Table Partitioning
 - Multi-Dimensional Clustering
 - Insert Time Clustering



Module Content

DPF – Database Partitioning Feature

- Supported Configurations
- Partition Groups
- Data Distribution
- Single System View Management

Table Partitioning (aka Range Partitioning)

- Setup
- Roll-In and Roll-Out Scenarios
- Partitioned Indexes
- Data Partition Elimination in Queries
- Table Partitioning REORG
- Backup and Restore at Table Partition Level

Multi-Dimensional Clustering

- Concepts and Definitions
- Differences Between Regular Indexes and Block Indexes
- Combining All Features
- Insert Time Clustering



The Need for Partitioning

Divide and Conquer

- Smaller subsets of data to analyze and return
 - Less I/O
 - Focus on particular subsets
- Smaller portions for more cores
 - Parallelism to tackle problems

Faster Data Availability

- Less Overhead
- Move legacy data from a table
- Place relevant data into a table

Improved Manageability

 Easily maintain specific portions of data



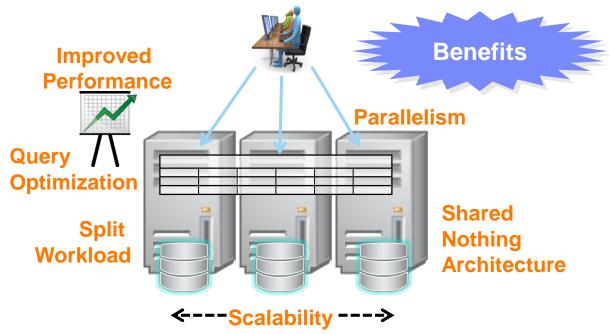
Data, while physically split, is considered as a logical whole

6



DB2 Database Partition Feature (DPF)

- A partitioned database environment is a database installation that supports the distribution of data across database partitions.
 - A database partition is a part of a database that consists of its own data, indexes, configuration files, and transaction logs
 - Data access across partitions is transparent to applications
- Leverages the power of multiple CPUs on multiple physical machines to efficiently process complex queries like those find in OLAP or data mining applications





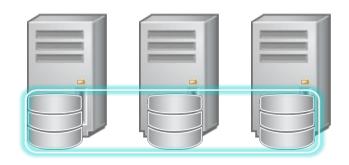
DPF Configurations



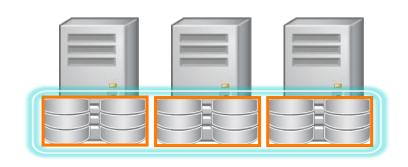
Single Logical Partition on Single Physical Partition



Multiple Logical Partitions on Single Physical Partition



Single Logical Partition on Multiple Physical Partitions



Multiple Logical Partitions on Multiple Physical Partitions



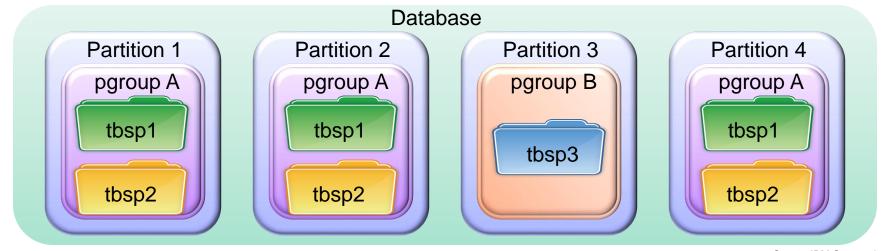
DPF – Partition Groups

- Defines the table space distribution across all partitions or a subset of partitions.
- Default Partition groups:
 - IBMDEFAULTGROUP: user tables ← All partitions
 - IBMTEMPGROUP: temporary tables
 - IBMCATGROUP: catalog tables

CREATE DATABASE PARTITION GROUP pgroupA ON DBPARTITIONNUMS (1,2,4)

CREATE DATABASE PARTITION GROUP pgroupB ON DBPARTITIONNUMS (3)

CREATE TABLESPACE tbsp3 IN pgroupB MANAGED BY DATABASE USING
FILE'd:\db2data\tbsp3' 5000)





DPF – Node Configuration (db2nodes.cfg)

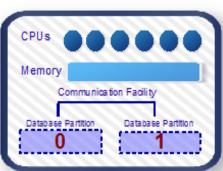
- When a database is created, all database partitions that are specified in the database partition configuration file (db2nodes.cfg) are created as well
- DB partition number
 - Unique database partition ID
- Server hostname
 - Machine's name or IP address
- Server logical-port

9

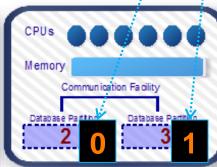
Logical partition ID within a machine

Partition	Server Name	Logical Port
0	Server A	0
1	Server A	1
2	Server B	0
3	Server B	1
4	Server C	0
5	Server C	1

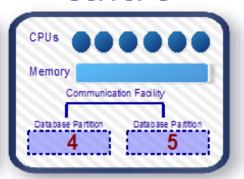
Server A



Server B



Server C





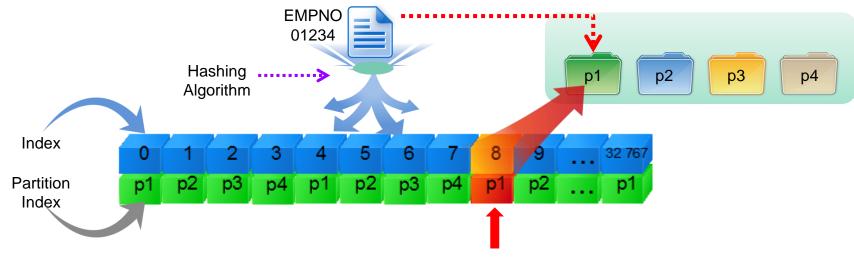
DPF - How Data is Distributed

Distribution Map

- Internally generated array containing 32 768 (32KB) entries used for deciding where data will be stored within the partitions
- Partition numbers are specified in a round-robin fashion in the array

Distribution Key

- Column(s) that determines the partition, physically storing, a particular row of data
- Define key using CREATE TABLE statement with the **DISTRIBUTE BY** clause
- Design Advisor can be used to suggest an optimal distribution key
- Hashing Algorithm generates a value between 0 and 32 767 based on the distribution key



The partition key EMPNO (value 01234) is hashed to a value 8, which is used to index to the partition 1.



DPF - Single System View Management

DB administration commands are propagated to all partitions



Backup and restore

- Use Data Studio or CLP to backup a partitioned database simultaneously
- Specify partition to backup by using **DBPARTITIONNUM(S)** attribute

```
BACKUP DATABASE SAMPLE ON DBPARTITIONNUM 1;

BACKUP DATABASE SAMPLE ON DBPARTITIONNUMS (1,2);
```

Use the EXCEPT clause to single out a partition

```
BACKUP DATABASE SAMPLE ON ALL DBPARTITIONNUMS EXCEPT DBPARTITIONNUM (1);
```

- Database configuration updates
 - Perform update normally and it will propagate to all partitions

```
UPDATE DB CFG FOR SAMPLE USING MAXAPPLS 50;
```

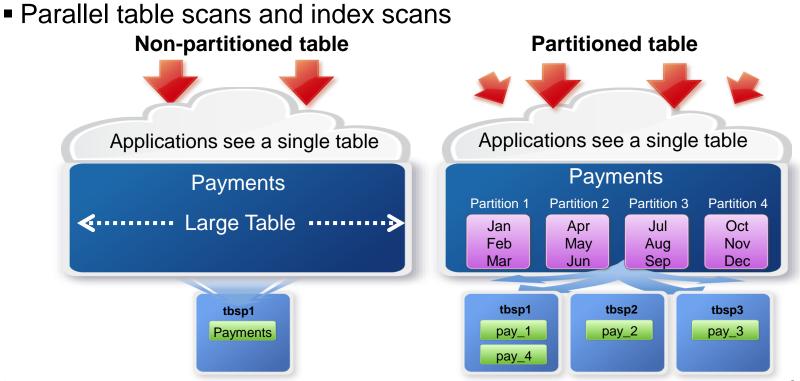
- Specify partition by using **DBPARTITIONNUM(S)** attribute

```
UPDATE DB CFG FOR SAMPLE DBPARTITIONNUM 1 USING MAXAPPLS 40;
```



Table Partitioning

- Allows a single logical table to be broken up into multiple separate physical storage objects (a.k.a. data partitions)
 - Up to 32K data partitions
 - Each partition defines a range of values
 - A partition will only contain rows that match its range of values





Benefits of Table Partitioning

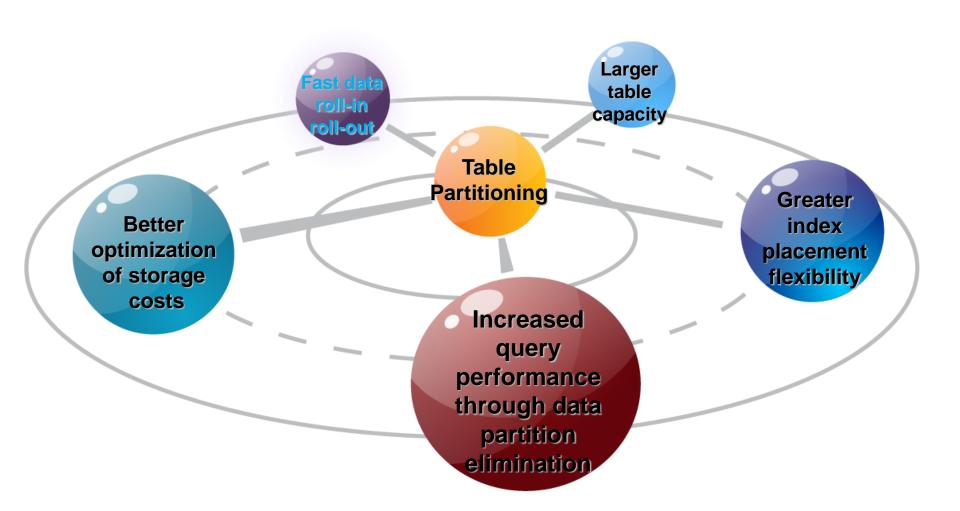




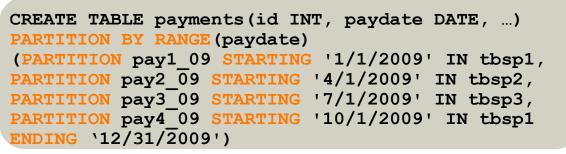
Table Partitioning - Syntax

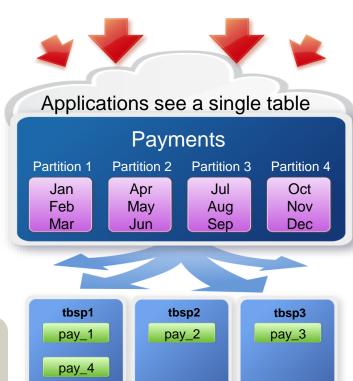
- Partitioning Columns
 - Must be base types (No LOBS, LONG VARCHAR)
 - Accepts multiple columns and generated columns
 - MINVALUE and MAXVALUE can be used to specify open boundaries
- It only accepts values for the defined ranges
 - SQL0327N is raised if no range matches the data being inserted

Short Form

```
CREATE TABLE payments (id INT, paydate DATE, ...)
IN tbsp1, tbsp2, tbsp3 PARTITION BY RANGE (paydate)
(STARTING '1/1/2009' ENDING '12/31/2009' EVERY 3
MONTHS)
```

Long Form







Data Partition Elimination

 Ability to determine that only a subset of the data partitions in a table are necessary to answer a query



- DB2 EXPLAIN
 - Provides detailed information about which data partition are used when a query is run
 - db2exfmt provides details from EXPLAIN statement



Roll-in – Attaching Data Partitions

- Roll-in: ALTER TABLE ... ATTACH
 - Incorporates an existing table as a new range
- Use SET INTEGRITY to validate data and maintain indexes after ATTACH operation
 - If data integrity can be checked before attachment, newly attached data can be made available sooner with SET INTEGRITY IMMEDIATE UNCHECKED
 - If **SET INTEGRITY IMMEDIATE CHECKED**, commit is required before data is visible

```
ALTER TABLE PAYMENTS ATTACH PARTITION pay1_10
STARTING '01/01/2010' ENDING '03/31/2010'
FROM TABLE NEWPAY2010;
COMMIT;
SET INTEGRITY ... ;
COMMIT;
```

 Minimal interruption to other queries that access the table

No data movement



NEWPAY2010

table

<< stand alone >>

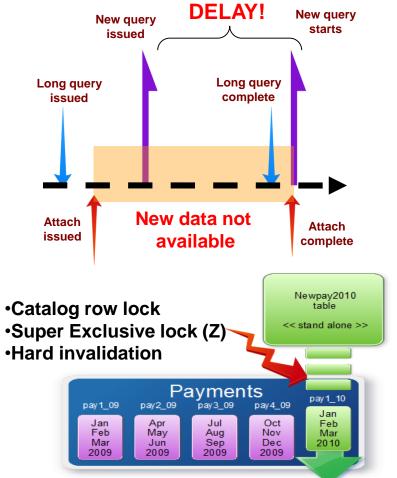


Faster Data Availability

Previous DB2 versions

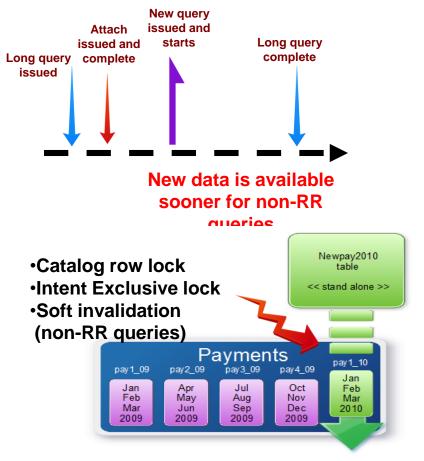
17

 Command waits for existing queries to complete and then delays new queries



In DB2 10

 ATTACH or ADD completes immediately, allowing new partition to be accessible sooner!



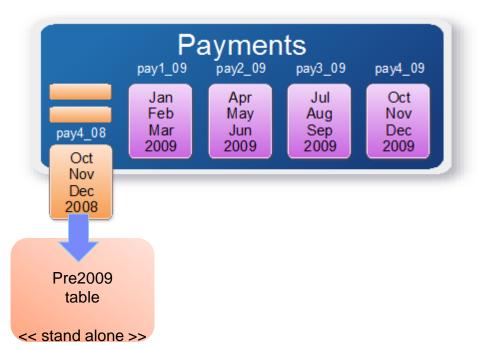


Roll-out – Detaching Data Partitions

- Roll-out: ALTER TABLE ... DETACH
 - Detaches the data partition and renames it to SQLyymmddhhssxxx
 - Data instantly becomes invisible
 - An asynchronous task converts the detached partition into a stand-alone table

ALTER TABLE payments DETACH PARTITION pay4 08 INTO PRE2009; COMMIT;

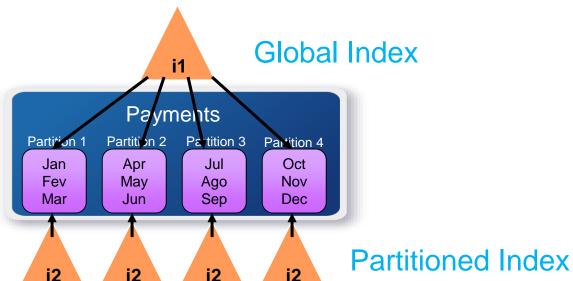
- Queries running against the table continue to run
- Minimal interruption to other queries accessing table
- No data movement





Indexes on Partitioned Tables

- Without table partitioning, all indexes for a particular table are stored in the same storage object by default
- With table partitioning, each index can be placed on its own table space
 - Including MDC (aka block) indexes
- Benefits
 - Improved data roll-in and roll-out performance
 - More granular control of index placement
 - Improved performance on dropping and creating online indexes
 - Space is freed the moment an index is dropped
 - More efficient concurrent access to index data for the table due to reduced I/O contention





REORG, Backup and Restore

REORG

- Can run concurrently in several partitions
 - ALLOW WRITE ACCESS

Read & write access on the specified partition until the index is updated Read & write access allowed on other partitions

ALLOW READ ACCESS

Read access on the specified partition until index is updated Read & write access allowed on other partitions

- Specify a different partition with the ON DATA PARTITION clause
- Enhanced availability
 - REORG certain partitions and allow other partitions to remain available

REORG INDEXES ALL ALLOW NO ACCESS ON DATA PARTITION PAY1_09
REORG TABLE ALLOW NO ACCESS ON DATA PARTITION PAY2 09

 Skips data partitions that are in a restricted state due to an ATTACH or DETACH operation.

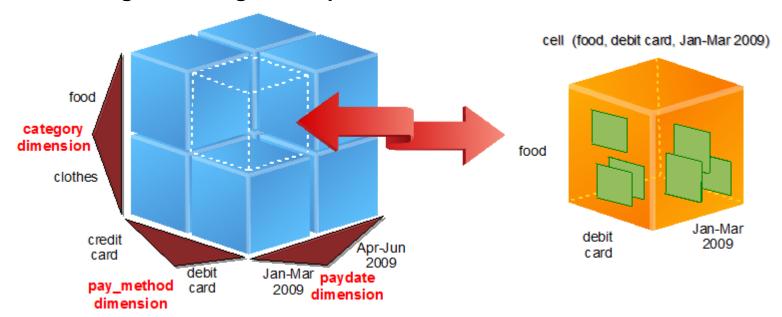
Backup and restore

 Backup time reduced, because the individual table partitions can be backed up if they are allocated to different table spaces



Multi-Dimension Clustering

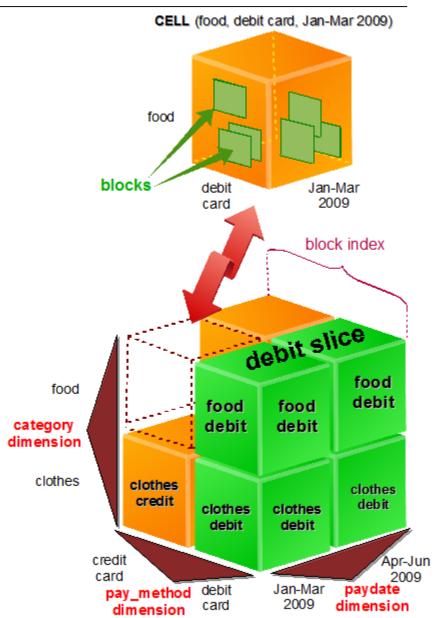
- Enables a table to be physically clustered on several dimensions simultaneously
- Primarily intended for data warehousing and large database environments
- DB2 places records that have the same column values in physical locations that are close together
 - Block indexes are smaller than RID indexes
 - Faster lookup
 - Scan only required blocks
 - Index ANDing and ORing can be performed at block level





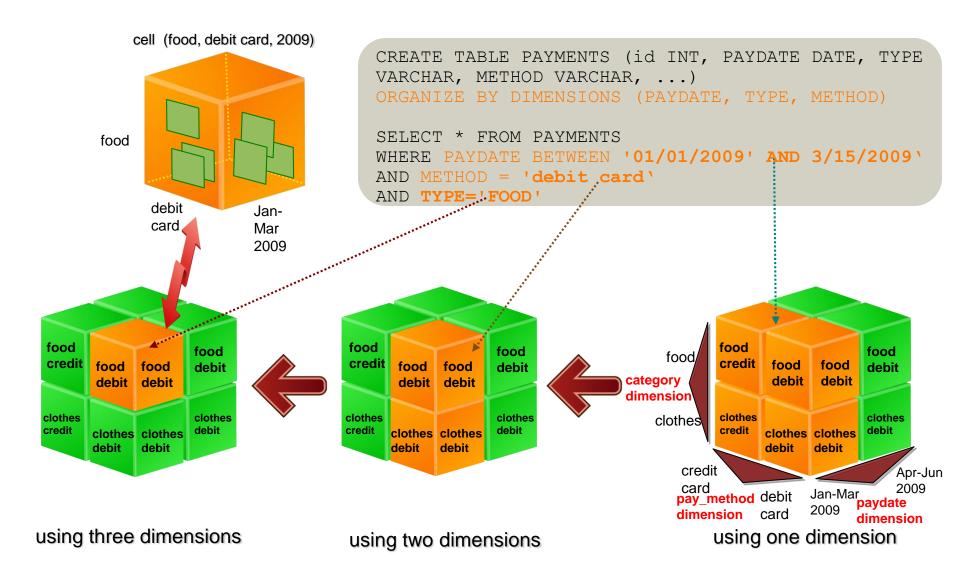
Multi-Dimensional Clustering

- Blocks: consecutive set of pages on the disk.
- Block Indexes: indexes that point to an entire block of pages
- Dimension: axis along which data is organized in an MDC table
 - Good candidates:
 - Columns referenced by ORDER BY and GROUP BY clauses
 - Columns used for range, equality, and IN predicates
 - Roll-in or roll-out of data
- Slice: portion of the table that contains all the rows that have a specific value for one of the dimensions.
- Cells: portion of the table that contains rows having the same unique set of dimension values





MDC Table Example

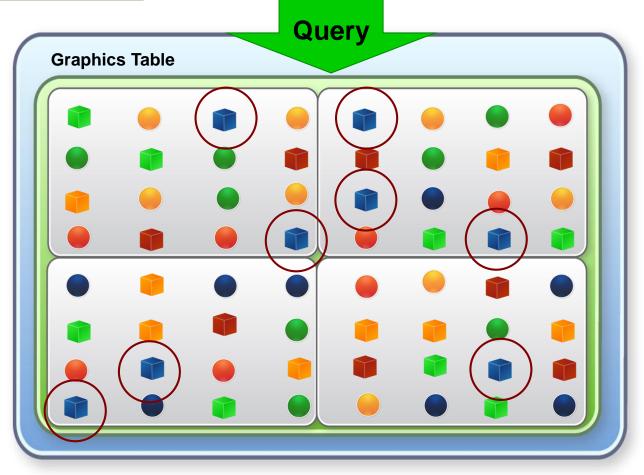




SELECT * FROM GRAPHICS where

Single large table without partitioning

Each I/O can pick up many unrelated records that happen to reside on the same page.

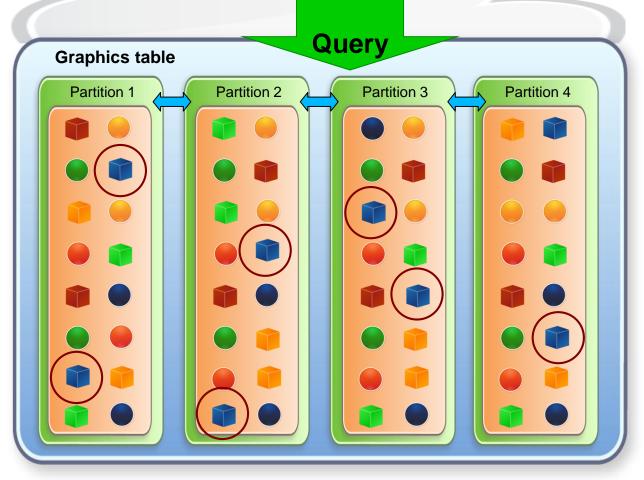




Using database partitioning

SELECT * FROM GRAPHICS where Applications see a single table

Take the advantage of the query parallelism provided by DB2 DPF attacking each partition in parallel.





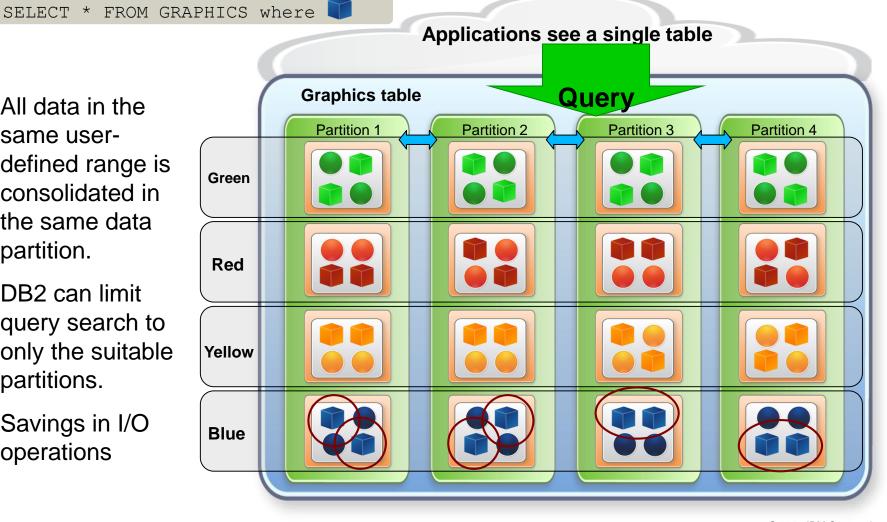
Using database partitioning and table partitioning

All data in the same userdefined range is consolidated in the same data

 DB2 can limit query search to only the suitable partitions.

partition.

Savings in I/O operations

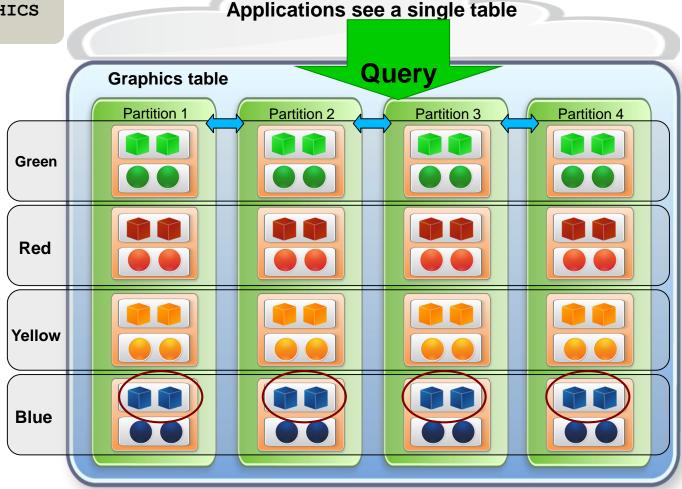




Using database partitioning, table partitioning, and MDC

SELECT * FROM GRAPHICS where

- Even less I/O is performed to retrieve the records of interest.
- DB2 can retrieve more matching pages as data is clustered according to the dimensions





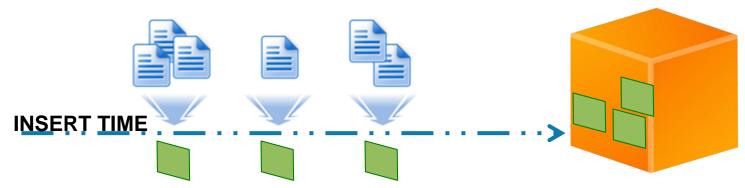
Insert Time Clustering (ITC) Tables



- Clusters data inserted at a similar time using an implicitly created virtual dimension
 - Virtual dimension cannot be manipulated
 - Cannot be used if table is a typed table
- Manage database size more effectively without manual intervention
 - Very similar to MDC tables. It uses block based allocation and block indexes.
 - Enhanced reorganization algorithm to reclaim more disk space within a short time

```
DB2 CREATE TABLE T1(c1 int, c2 char(100), ...) IN TABLESPACE1 ORGANIZE BY INSERT TIME;

DB2 CREATE INDEX INX1 ON T1(C1);
```





Summary

Database Partitioning Feature

- Many different configurations
- Single System View

Table Partitioning (aka Range Partitioning)

- Roll-in and Roll-out data with unchecked option
- Partitioned indexes are broken up and have a faster lookup
- Data partition elimination allows for the ability to scan only relevant data

Multi-Dimensional Clustering

- Allow for even better clustering to include only relevant data
- Block indexes are smaller and can result in a faster lookup



The next steps...





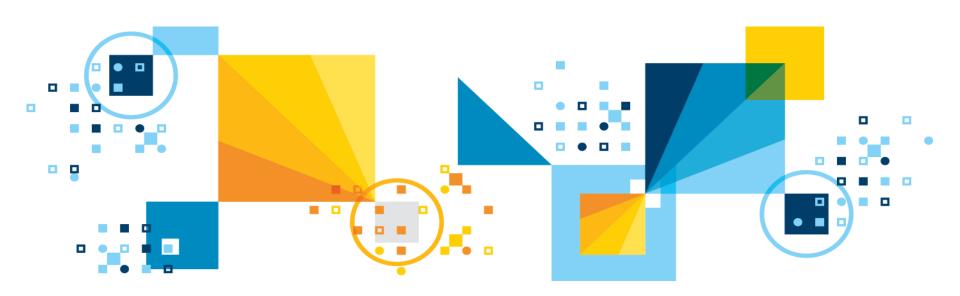
The Next Steps...

- Complete the Hands on Lab for this module
 - Log onto SKI, go to "My Learning" page, and select the "In Progress" tab.
 - Find the module
 - Download the workbook and the virtual machine image
 - Follow the instructions in the workbook to complete the lab
- Complete the online quiz for this module
 - Log onto SKI, go to "My Learning" page, and select the "In Progress" tab.
 - Find the module and select the quiz
- Provide feedback on the module
 - Log onto SKI, go to "My Learning" page
 - Find the module and select the "Leave Feedback" button to leave your comments





Questions? askdata@ca.ibm.com



January 30, 2015