



DB2 9.7: What's New

***Chris Eaton
IBM DB2 Technical Evangelist
ceaton@ca.ibm.com***

Agenda

- **Reducing Storage Costs and Storage Administration**
 - Index Compression
 - Temporary Table Compression
 - XML Compression
 - LOB Inlining
 - New Tablespace Type to lower high water mark
 - Partitioned Indexes
- **Break Free From Oracle**
 - New Concurrency Model
 - New data types
 - New SQL, PL/SQL, Functions, Packages, etc.
- **Improved Performance**
 - New SQL
 - Faster scan performance and better multi user query performance
- **And much, much more ...**

DB2 Reduces Storage Costs And Storage Administration

The Current IT Storage Climate



“Storage hardware is steadily increasing as a percentage of IT budgets. Backup and recovery was the second most important driver of spending...and is the No. 1 storage issue keeping CIOs awake at night”



*Global spending on the data center **power and cooling** in 2007 was roughly **equivalent to spending on servers***



*Half of the world's data centers will face an **acute power shortage** by the end of 2008*

How Compression Works

- **Compression looks for repeating patterns across the entire table**
 - When a pattern is found, string is replaced with 12-bit symbol
 - Symbols are stored in a dictionary for fast lookup
- **Compression is *not* just about storage savings**
 - Less pages means **less time for backup, smaller backup size, better I/O, better performance, less power and cooling costs**

| Name | Dept | Salary | City | Province | Postal_Code |
|-------------|------|--------|--------|----------|-------------|
| Zikopoulos | 510 | 56105 | Whitby | ONT | L4N5R4 |
| Katsopoulos | 500 | 82475 | Whitby | ONT | L4N5R4 |

How Compression Works

- **Compression looks for repeating patterns across the entire table**
 - When a pattern is found, string is replaced with 12-bit symbol
 - Symbols are stored in a dictionary for fast lookup
- **Compression is *not* just about storage savings**
 - Less pages means **less time for backup, smaller backup size, better I/O, better performance, less power and cooling costs**

| Name | Dept | Salary | City | Province | Postal_Code |
|-------------|------|--------|--------|----------|-------------|
| Zikopoulos | 510 | 56105 | Whitby | ONT | L4N5R4 |
| Katsopoulos | 500 | 82475 | Whitby | ONT | L4N5R4 |

| Dictionary | |
|------------|-----------------|
| 01 | opoulos |
| 02 | WhitbyONTL4N5R4 |
| ... | ... |

Zikopoulos 510 56105 Whitby ONT L4N5R4 Katsopoulos 500 82475 Whitby ONT L4N5R4 ...

Unique
to DB2

Zik (01) 510 56105 (02) Kats (01) 500 82475 (02) ...

The DB2 Compression Advantage

- **Superior compression rates due to DB2 algorithm**
 - DB2 compresses data by looking at all values in the table
 - Other vendors only remove duplicates at the page/block level
 - Disadvantages of page level approach
 - Consistent repeating values throughout the entire table will be stored multiple times in each page header
 - There may be repeating patterns in the table but not on each page

| Table | Compression Ratio | |
|----------|-------------------|-------------------|
| LINEITEM | 38% | 58% (1.5x better) |
| ORDERS | 18% | 60% (3x better) |
| | Oracle | DB2 |

 InfoWorld

“Row-level compression is a revolutionary development that will leave Oracle and Microsoft green with envy”.

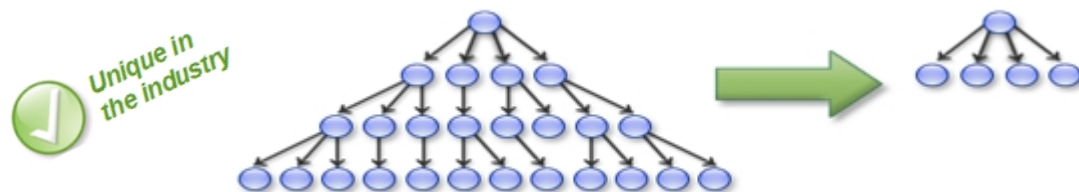
A close-up photograph of a gold coin. The coin is highly reflective and shows intricate embossed details. In the upper left, a circular seal is partially visible with the word "TRICENTENARIUM" embossed around its edge. The main body of the coin features large, bold, embossed letters, likely "1000", which are slightly out of focus. The background is a textured, golden surface.

**"...cost savings are more than \$2M initially,
with ongoing savings of \$500k a year"**

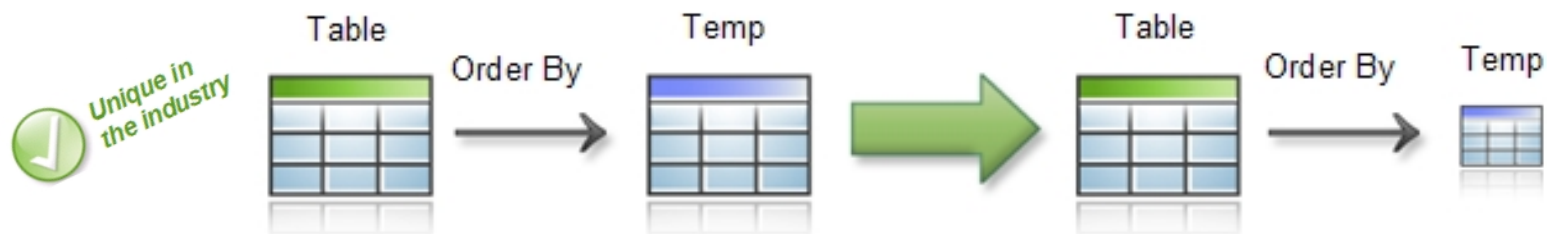
—Michael Henson, SunTrust Bank, Inc.

DB2 9.7 Compression Enhancements

- Multiple algorithms for **automatic** index compression



- **Automatic** compression for temporary tables



- Intelligent and **automatic** LOB and XML compression



RID List Compression - Example

10 rows stored on page 4 for a table in a LARGE tablespace

In an uncompressed index, DB2 stores the following RIDs:

<00 00 00 04, 00 00>, <00 00 00 04, 00 01>, <00 00 00 04, 00 02>,
<00 00 00 04, 00 03>, <00 00 00 04, 00 04>, <00 00 00 04, 00 05>
<00 00 00 04, 00 06>, <00 00 00 04, 00 07>, <00 00 00 04, 00 08>
<00 00 00 04, 00 09>

RID List Compression - Example

10 rows stored on page 4 for a table in a LARGE tablespace

In an uncompressed index, DB2 stores the following RIDs:

<00 00 00 04, 00 00>, <00 00 00 04, 00 01>, <00 00 00 04, 00 02>,
<00 00 00 04, 00 03>, <00 00 00 04, 00 04>, <00 00 00 04, 00 05>
<00 00 00 04, 00 06>, <00 00 00 04, 00 07>, <00 00 00 04, 00 08>
<00 00 00 04, 00 09>

In a compressed index, DB2 will compress the RID list

<00 00 00 04, 00 00>, <1>, <1>, <1>, <1>, <1>, <1>, <1>, <1>, <1>

Delta's are stored where appropriate

- can use one byte instead of 6

Prefix Compression - Example

If we have the following keys in an uncompressed index

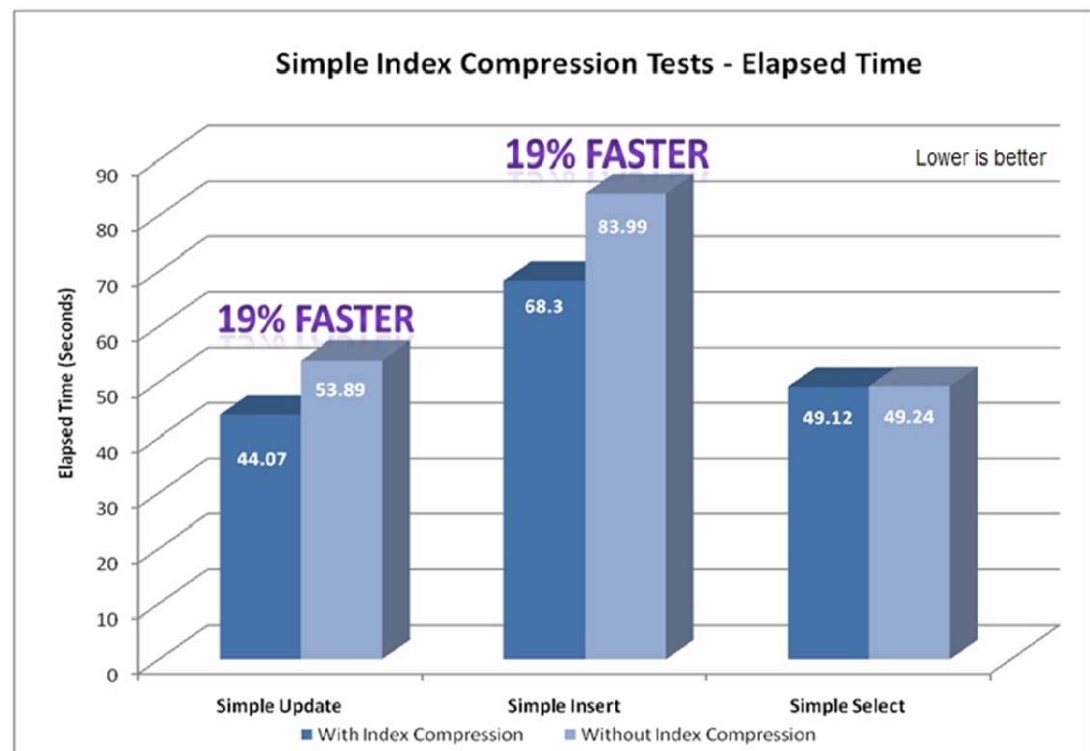
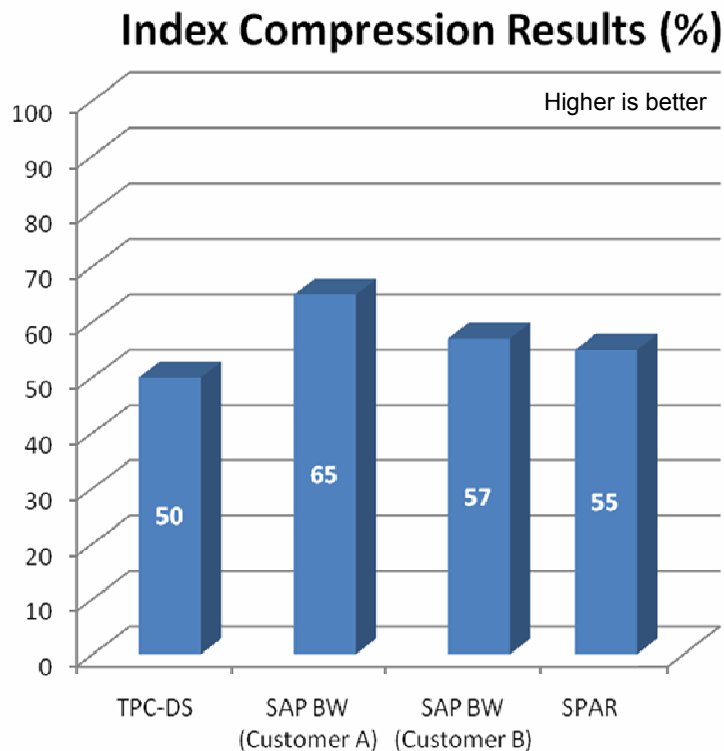
(**'Albert'**, **'Adams'**, **10502**),
(**'Albert'**, **'Adams'**, **10952**),
(**'Albert'**, **'Adams'**, **10999**),
(**'Bob'**, **'Backer'**, **10010**),
(**'Bob'**, **'Backer'**, **20111**)

DB2 Cobra we will keep only

(**'Albert'**, **'Adams'**, **10502**),
(**952**),
(**999**),
(**'Bob'**, **'Backer'**, **10010**)
(**20111**)

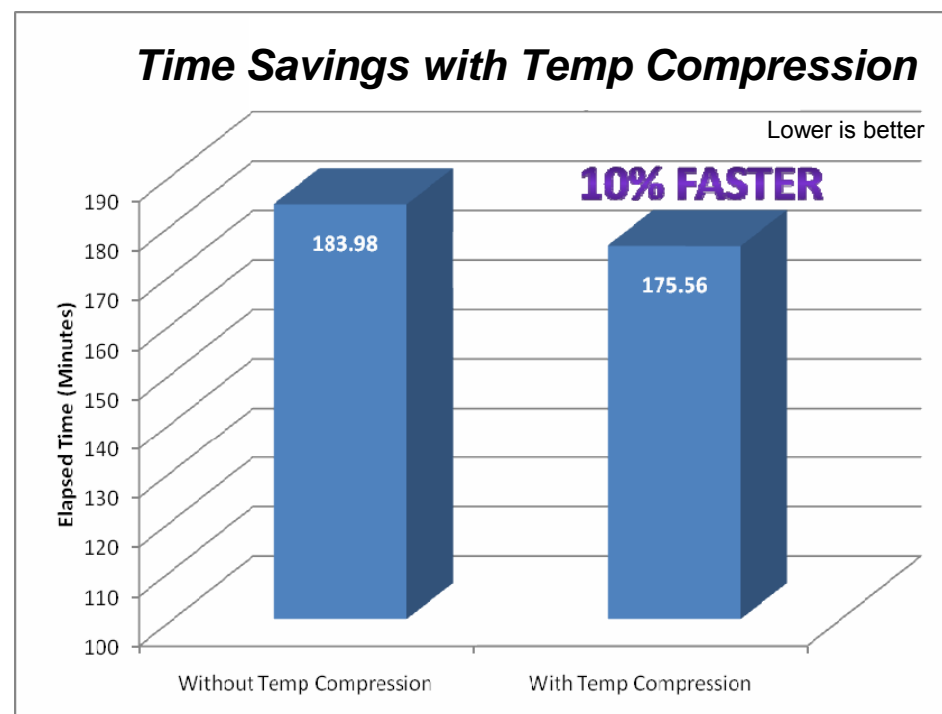
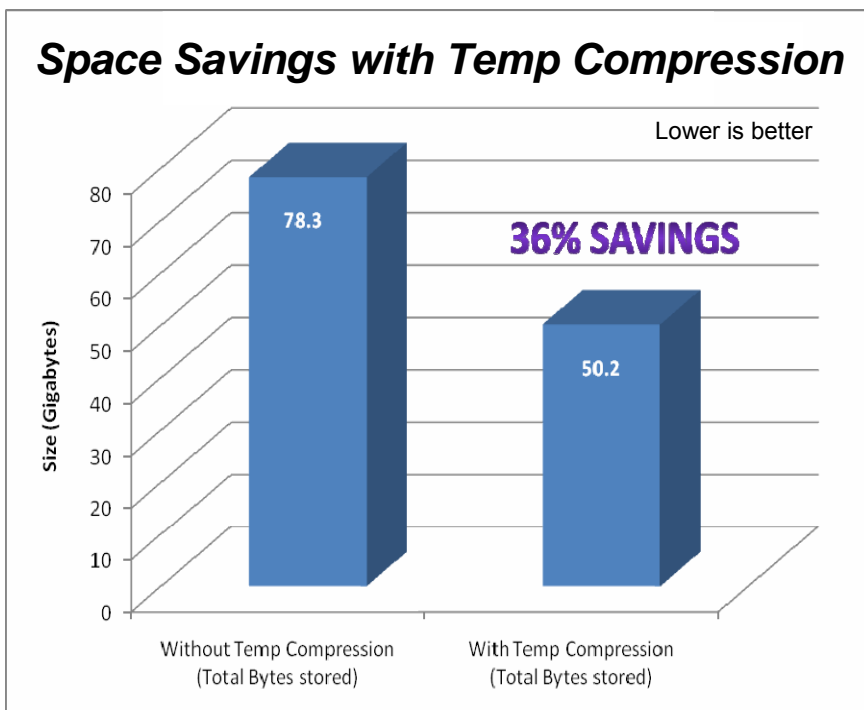
Index Compression: Sample Data

- **Significant index compression ratios added to table compression should yield database reductions up to 70%**
- **These savings require no DBA intervention**

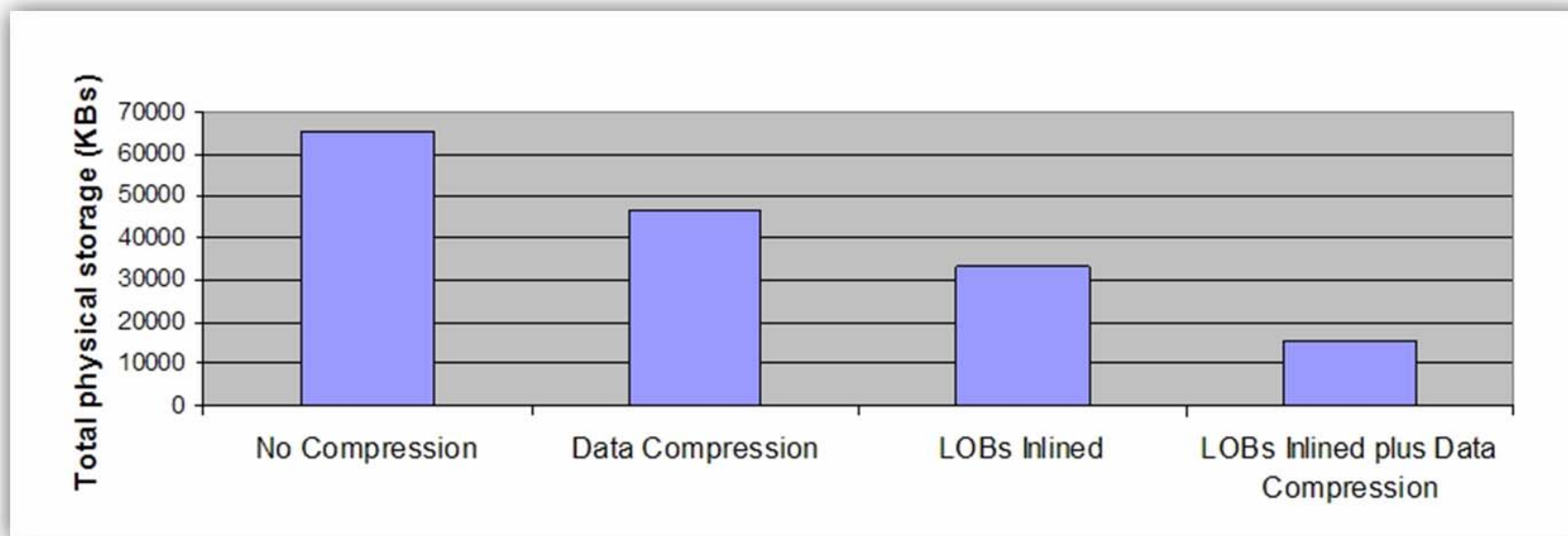


Temporary Table Compression Results

- **Only vendor in industry to compress temp tables**
 - Internal (DB2 utilities, SORT) and external (DGTTs) compressed
- **Savings and performance benefits require no DBA intervention**



LOB Inlining: Example Savings



"We have achieved a very significant performance boost of 20 percent in execution time on the database server while exploiting the new LOB inlining feature. The efficient DB2 processing of LOB objects reduced the system time requirements on the server by as much as 50 percent..."

– François Miginiac, GraphTalk A.I.A Product Director

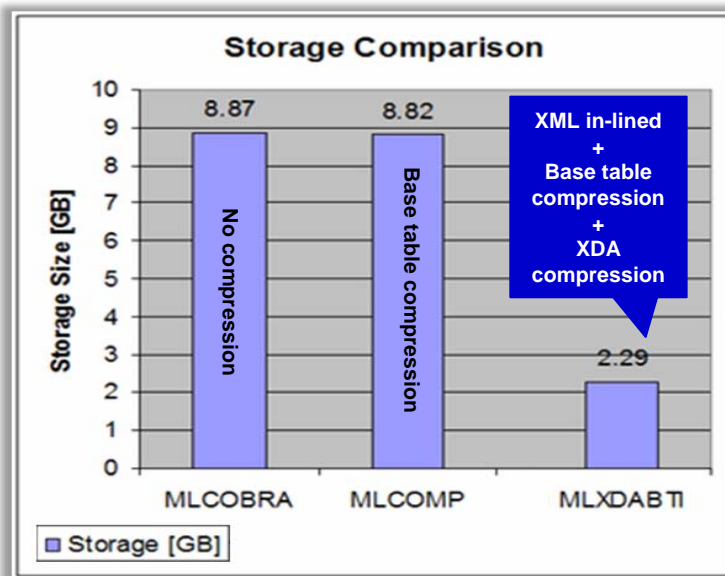


"Compression with data capture changes and LOB inlining will enable us to lower costs and simplify administration. These enhancements are very valuable to our business." – Marija Ljubic, IT Specialist

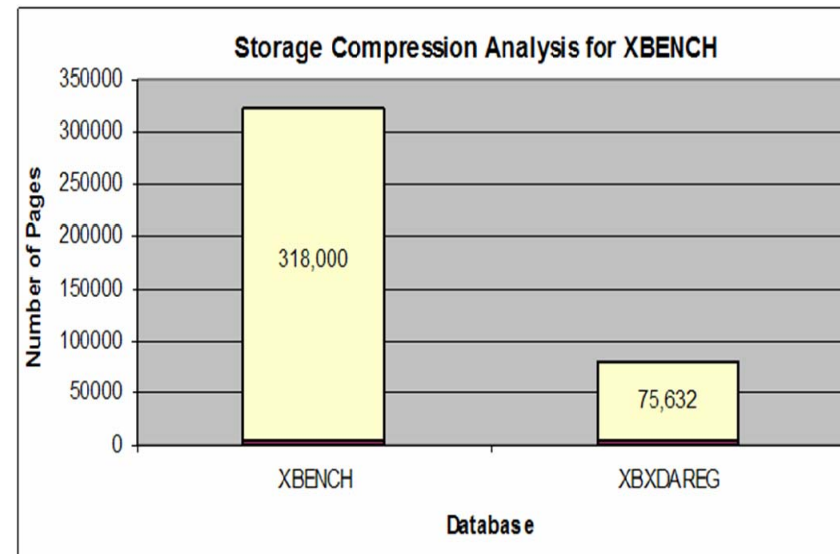
pureXML Compression of the XDA Area

- DB2 9.7 introduces the ability to **automatically** compress XML documents that aren't in-lined and stored in the XDA

Small Document Compression Ratio: **70%**
 (Storage size reduces from 8.87GB to 2.67GB)



Large Document Compression Ratio: **75%**
 (Storage size reduces from 2.46GB to 0.61GB)



UCLA Health System

"The hospital is experiencing the compression benefit reducing the amount of storage space for patient's medical records in XML by 50 percent today."

- Charles Wang, PhD, UCLA Health System

ScotDB Limited
 Database Consulting

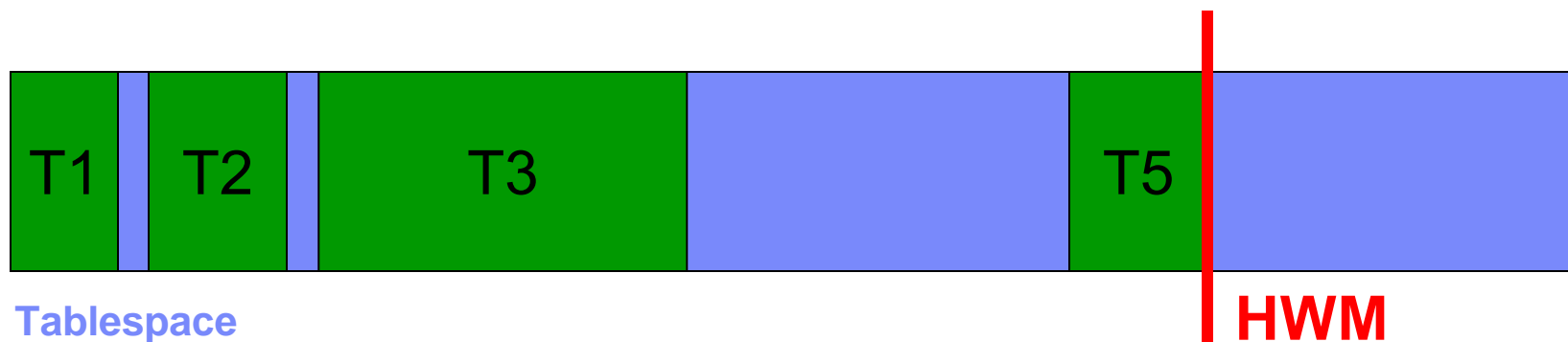
"XML compression reduces disk space consumption by more than 60%, irrespective of the size of the XML documents..." - Phil Nelson, ScotDB

Tablespace Extent Remapping:

- **Often want to lower the high water mark to reclaim free space embedded within a tablespace**
- **9.7 Solution:**
 - **ALTER TABLESPACE... LOWER HIGH WATERMARK**
 - All new DMS tablespaces will be created with this 'new' underlying infrastructure
- **Tablespace snapshot and new tablespace procedure to determine whether the tablespace is type1 (old version) or type2 (new version)**

Example of Tablespace with embedded space

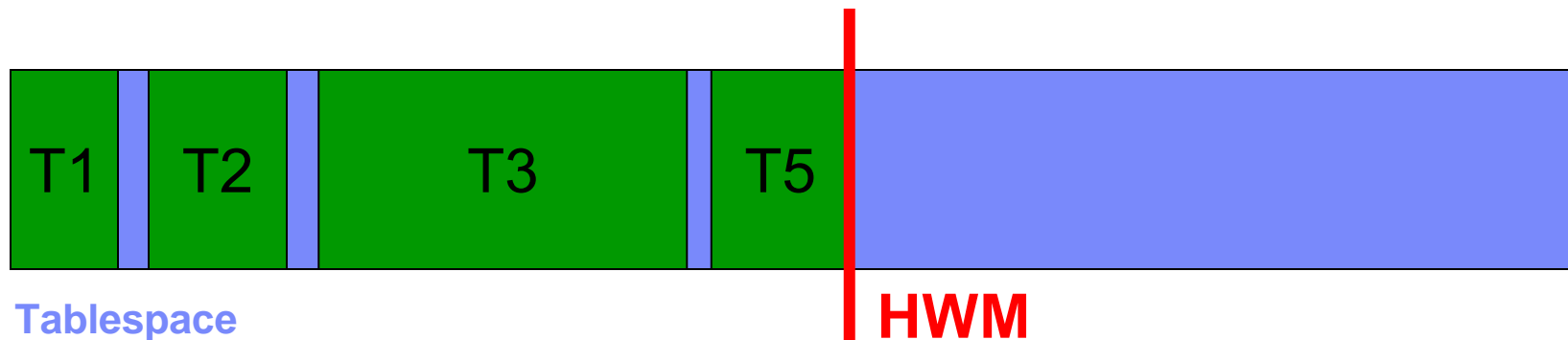
- **DB2 will back up all pages up to the high watermark**
- **New tables can consume embedded free space**
- **But you cannot reduce the tablespace beyond the high watermark to release that freespace back to the operating system.**



New Tablespace Type in 9.7 Can Move Tables

ALTER TABLESPACE LOWER HIGH WATERMARK

- **Moves T5 down the tablespace by moving extents**
 - No change needed to the index
 - Can monitor progress with new snapshot table function



Partitioned (Local) Indexes

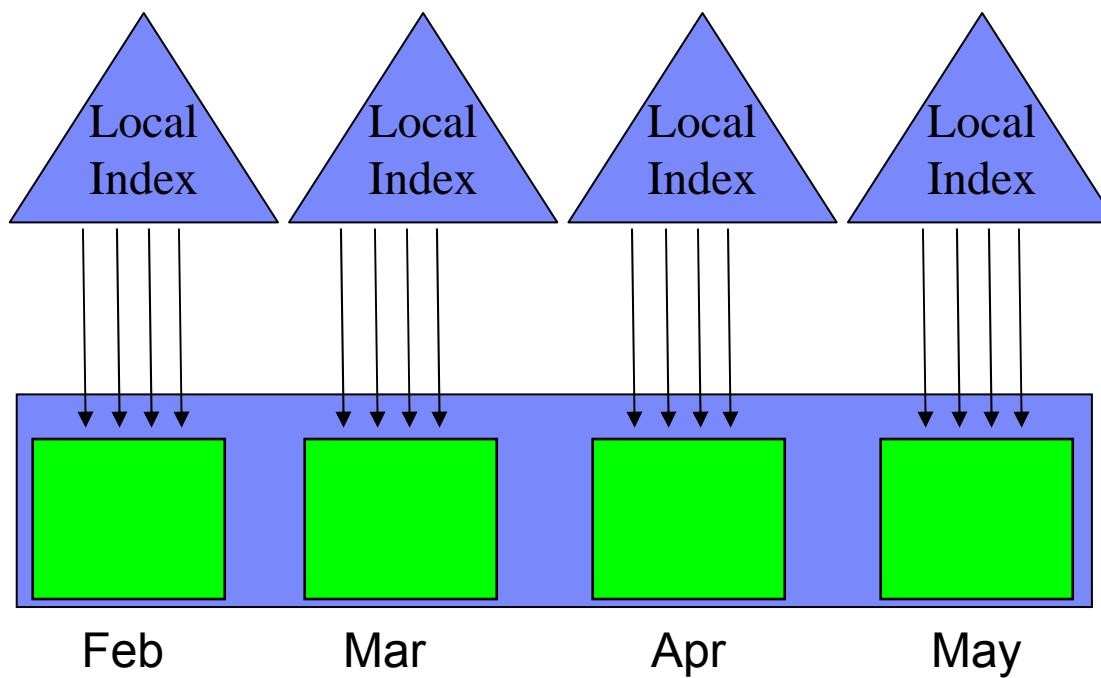
- **Will support the ability to create local (partitioned) indexes**
- **This will relieve current issues associated with Roll-In processing, mainly the global index maintenance and associated logging**
- **Allows for reorg table at the range partition level**
 - Improved ease of use with respect to range level compression

Example:

CREATE INDEX pINX1 on SALES (sales_date, partID) PARTITIONED

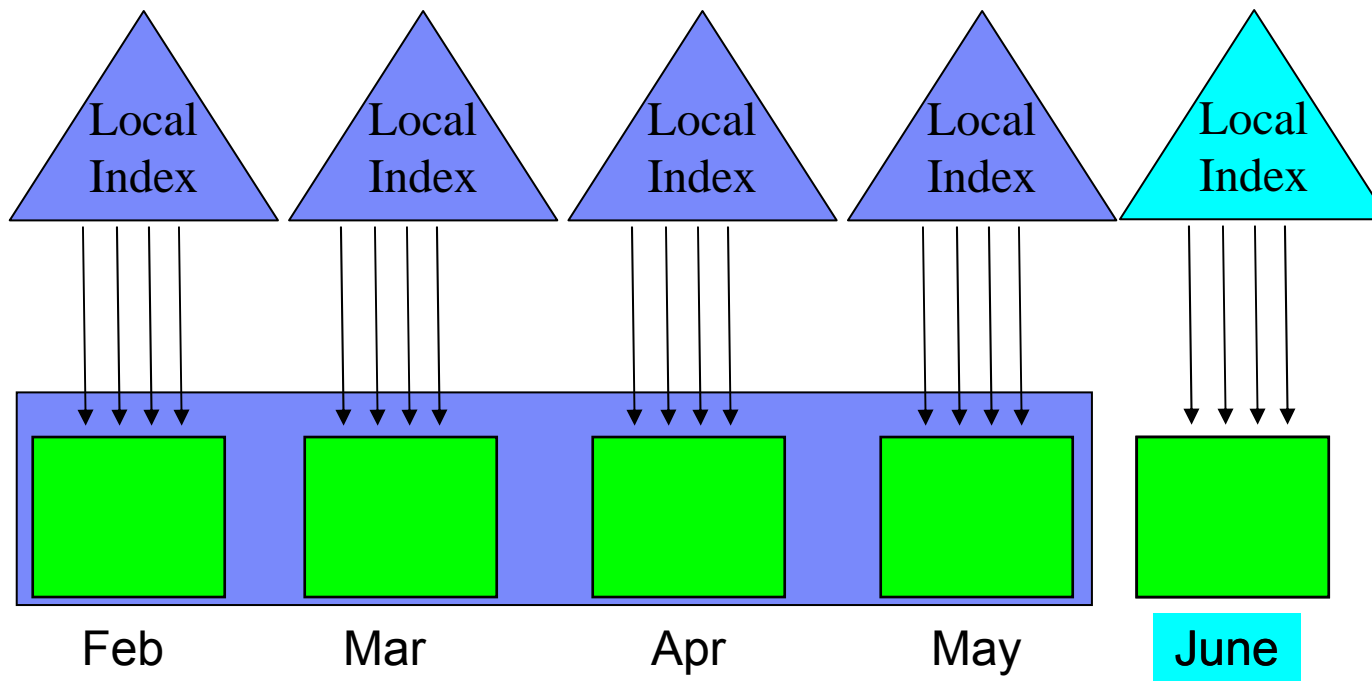
- **Partitioned index is the default**
- **Statistics –**
 - RUNSTATS will collect at the table level as well as at partition data and index level
 - SYSIBM.SYSDATAPARTITIONS is extended to include:
 - CARD, OVERFLOW, NPAGES, FPAGES, AVGROWSIZE, PCTROWSCOMPRESSED, ...

DB2 9.7 Index Partitioning – Faster Roll-in

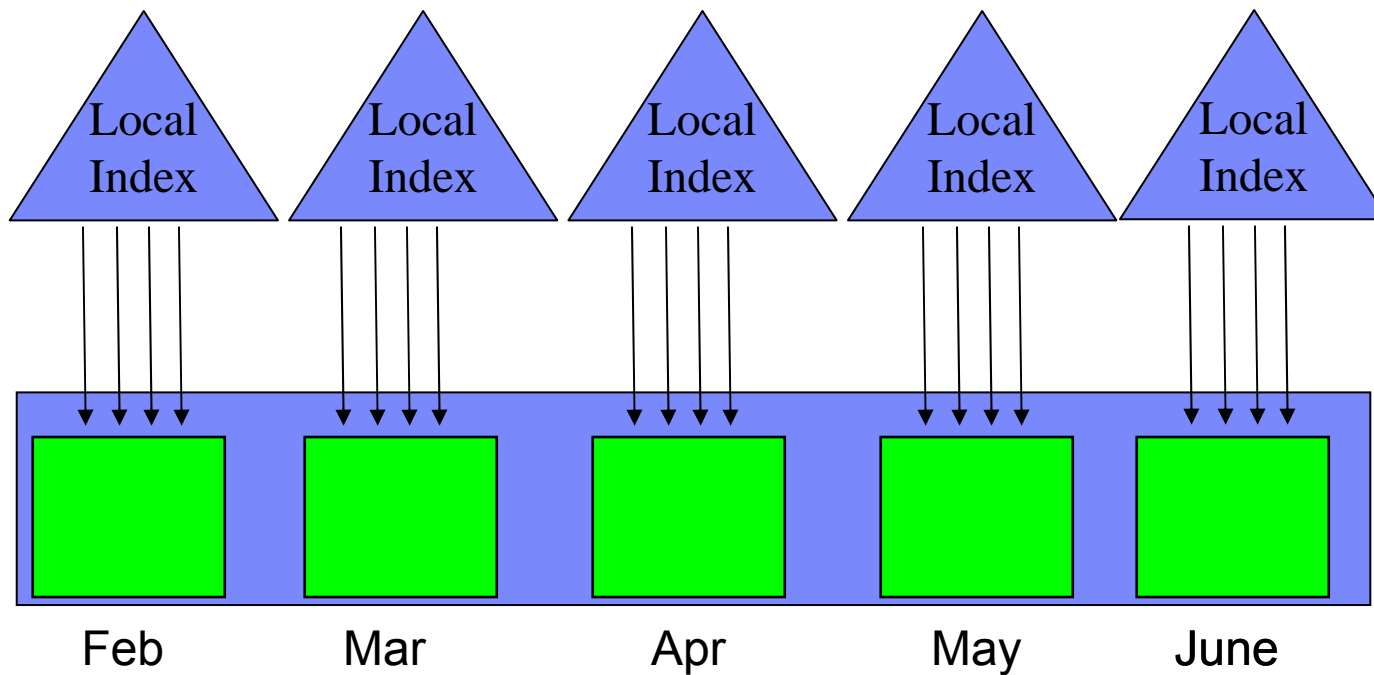


DB2 9.7 Index Partitioning – Faster Roll-in

```
CREATE TABLE  
JUNE  
CREATE INDEX
```



DB2 9.7 Index Partitioning – Faster Roll-in



No Global Index to Maintain



ALTER TABLE
ATTACH
JUNE
SET INTEGRITY

Move From Oracle to DB2 the Easy Way

Oracle Database Features Supported by DB2 9.7

| Oracle Database | → | DB2 | Not Emulation |
|-----------------------|---|----------------|---------------|
| Concurrency Control | → | Native support | |
| SQL | → | Native support | |
| PL/SQL | → | Native support | |
| Packages | → | Native support | |
| Built-in packages | → | Native support | |
| OCI | → | Native support | |
| JDBC | → | Native support | |
| Online schema changes | → | Native support | |
| SQL*Plus Scripts | → | Native support | |

Differences are the exception, rather than the rule.
This is why we call it enablement **not migration.**

Concurrency and DB2 9.7

- **Oracle default**

- Statement level snapshot

| blocks | Reader | Writer |
|--------|--------|--------|
| Reader | No | No |
| Writer | No | Yes |

- **DB2 default prior to Cobra**

- Cursor stability

| blocks | Reader | Writer |
|--------|--------|--------|
| Reader | No | Maybe |
| Writer | Yes | Yes |

Enabling Oracle application to DB2 required significant effort to re-order table access to avoid deadlocks

- **DB2 default with Cobra**

- Currently Committed

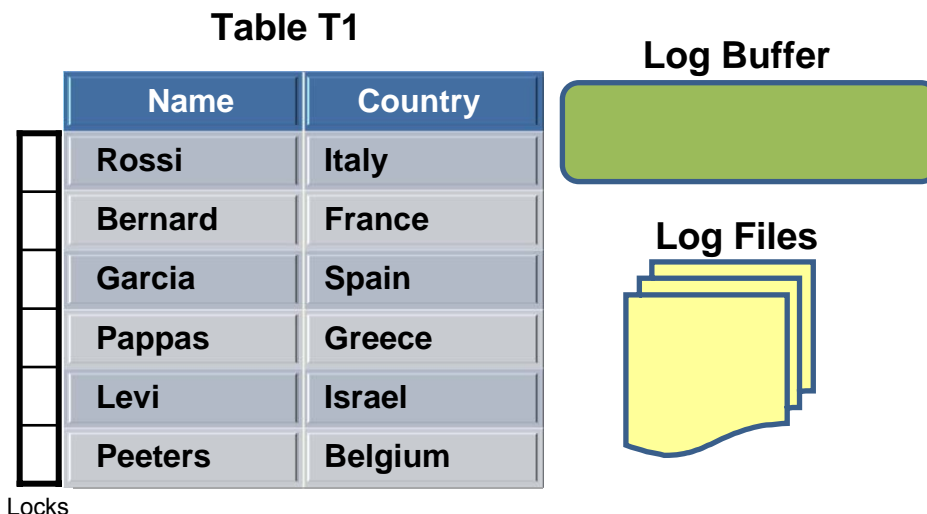
| blocks | Reader | Writer |
|--------|--------|--------|
| Reader | No | No |
| Writer | No | Yes |

Concurrency Control in DB2 9.7

- Reads the currently committed version of a row
 - If uncommitted row-change found use currently committed version
- Log based
 - No management overhead
 - No performance overhead
 - No wasted memory/storage (no undo tablespace)

User 1:

```
update T1 set name = 'Russo'  
where country='Italy'
```

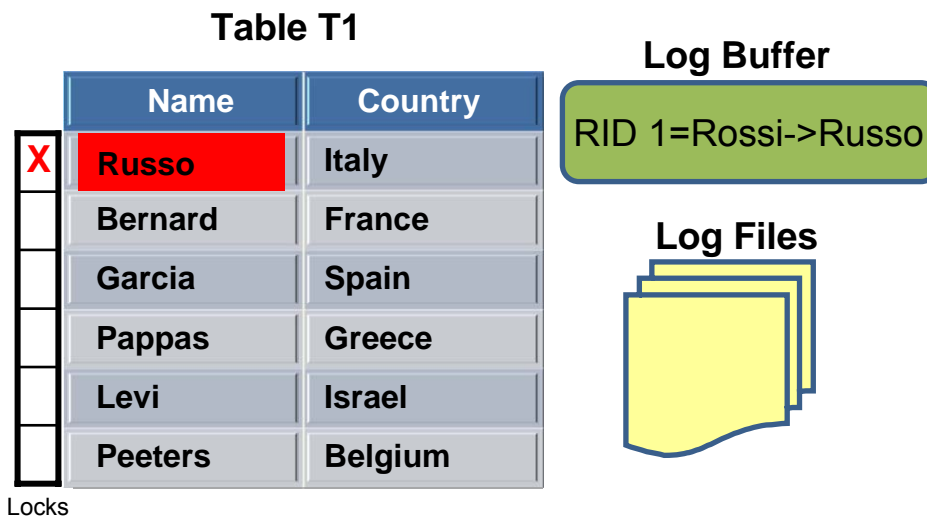


Concurrency Control in DB2 9.7

- Reads the currently committed version of a row
 - If uncommitted row-change found use currently committed version
- Log based
 - No management overhead
 - No performance overhead
 - No wasted memory/storage (no undo tablespace)

User 1:

```
update T1 set name = 'Russo'  
where country='Italy'
```

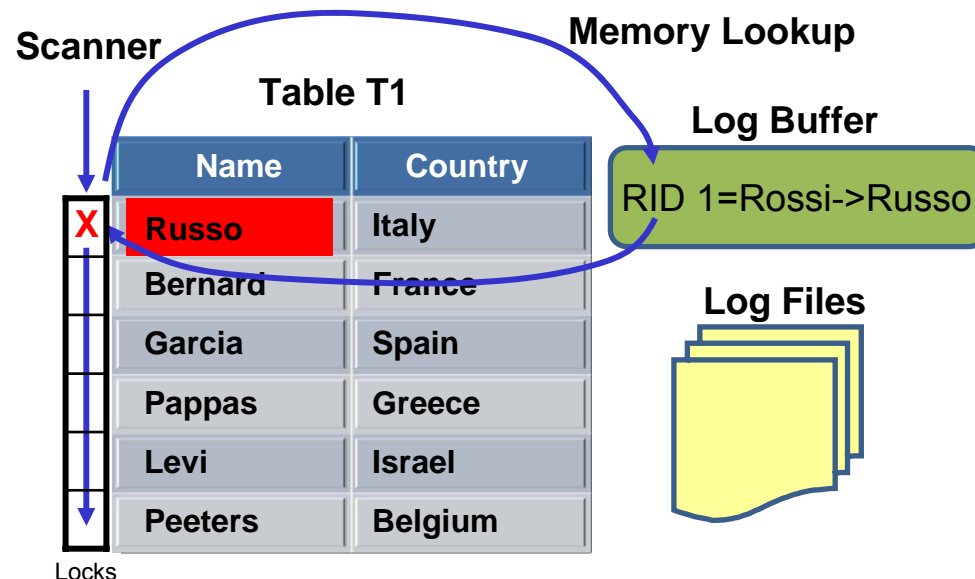


Concurrency Control in DB2 9.7

- Reads the currently committed version of a row
 - If uncommitted row-change found use currently committed version
- Log based
 - No management overhead
 - No performance overhead
 - No wasted memory/storage (no undo tablespace)

User 1:
`update T1 set name = 'Russo'`
`where country='Italy'`

User 2:
`select * from T1`



Currently Committed Competitive Advantage

- **Only incur added processing when a reader and writer are working on the same row**
- **No added overhead for a “just in case” collision**
 - With Oracle past images are stored in the undo tablespace just in case there is a collision
- **DB2 uses existing log infrastructure to retrieve currently committed data in flight**
 - Better performance
 - Lower overhead
 - Simplified management

Oracle types in DB2 9.7

| Type | Comment |
|---------------------|--|
| NUMBER | Exploits P6 hardware accelerated DECFLOAT |
| VARCHAR2 | NULL = '', trailing blank sensitive collation |
| TIMESTAMP(n) | 0 (date + time) <= N <= 12 (date + time + picoseconds) |
| "DATE" | Year to seconds, SYSDATE |
| BOOLEAN | In procedural code |
| INDEX BY | Associative arrays in procedural code |
| VARRAY | Regular arrays in procedural code |
| Row Type | In procedural code, VARRAY, INDEX BY |
| Ref Cursor | Allows passing, and predefining of cursors |

Oracle functions in DB2 9.7

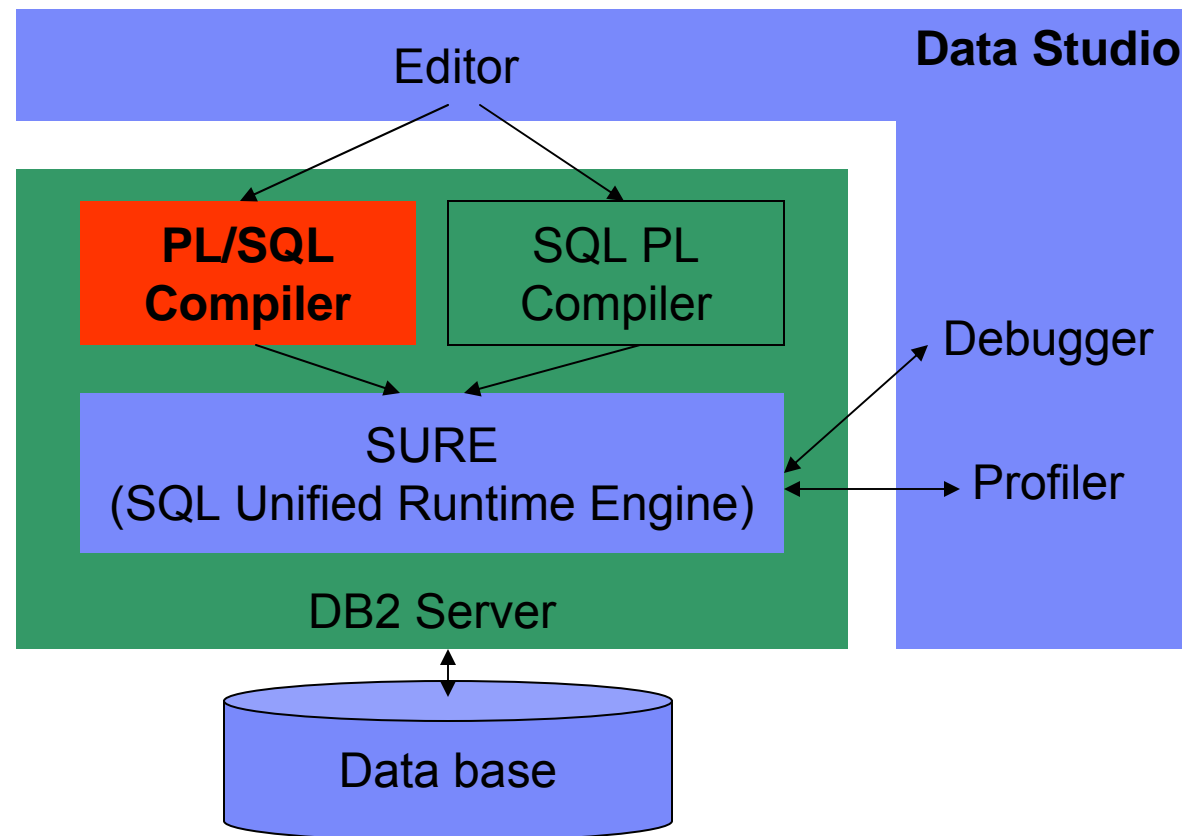
| Function | Comment |
|----------------------------------|---|
| Conversion and Formatting | TO_CHAR, TO_DATE, TO_TIMESTAMP, TO_NUMBER, TO_CLOB |
| Datetime arithmetic | EXTRACT, ADD_MONTHS, ... |
| String manipulation | INITCAP, RPAD, LPAD, INSTR, REVERSE, ... |
| Misc | DECODE, NVL, LEAST, GREATEST, BITAND |

Oracle SQL in DB2 9.7

| Feature | Comment |
|-----------------------------|--|
| CONNECT BY | Tree walk recursion, includes helper functions |
| (+)-join | Old style OUTER JOIN syntax |
| DUAL | Equivalent to SYSDDUMMY1 |
| ROWNUM | Pseudo column syntax for ROW_NUMBER() |
| NEXTVAL/CURRVAL | Pseudo column syntax for sequences |
| MINUS | A synonym for EXCEPT |
| Unnamed inline views | Optional correlation names for subqueries |
| TRUNCATE table | |
| Public synonym | For table, sequence, module/package |
| CREATED temp table | Temp table with persistent definition |
| Much, much more | |

PL/SQL in DB2 9.7

- **Built in PL/SQL native compiler**
- **Source level debugging and profiling**



PL/SQL Features in DB2 9.7

| Function | Comment |
|-------------------------|--|
| All logic | IF, WHILE, :=, etc.. |
| EXCEPTION | Try/catch handling |
| User Defined Exceptions | Define conditions with or without SQLCODEs |
| Constant variables | Variables that cannot be set |
| FOR over range | Step through numbers |
| over SELECT | Step through result set of query |
| over cursor | Step through result set of cursor |
| %TYPE | Anchored scalar data types |
| %ROWTYPE | Anchored row types |
| BULK COLLECT/FETCH | Aggregate result set into array |
| FORALL | Pipe array into SQL statement |
| AUTONOMOUS transaction | Executes a procedure in an independent TX |

PL/SQL in DB2 9.7

| PL/SQL now in: | Comment |
|------------------------|-----------------------------------|
| Anonymous block | New also in SQL PL dialect |
| Scalar function | |
| Procedure | |
| Package | Known as MODULE in DB2 |
| Trigger | |

PL/SQL Package in DB2 9.7

| Feature | Comment |
|-----------------------------|--|
| CREATE PACKAGE | Defines prototypes and public objects |
| CREATE PACKAGE BODY | Defines content and private objects |
| Replace package body | Replace body without losing prototypes or public objects |
| PKG [BODY] VARIABLE | Public/private variables |
| CURSOR | Public/private cursors |
| TYPE | Public/private types |
| EXCEPTION | User defined exceptions |
| SYNONYM ON PACKAGE | Public synonyms |

- DB2 shreds package and body into individual *module* objects
- External management view is preserved

Built-in package libraries in DB2 9.7

| Feature | Comment |
|---------------------|---|
| DBMS_OUTPUT | “print debugging” and simple reporting |
| UTL_FILE | Server side I/O API |
| DBMS_ALERT | Cross session semaphoring |
| DBMS_PIPE | Cross session data pipe |
| DBMS_JOB | Job scheduler |
| DBMS_LOB | Alternate API to DB2 native LOB functions |
| DBMS_SQL | Alternate API to PREPARE/EXECUTE |
| DBMS_UTILITY | Misc functions and procedures |
| UTL_MAIL | Server API to email |
| UTL_SMTP | Server API to SMTP |

Option to add more libraries as needed.

The Praise for SQL Compatibility

Gartner.

“The Oracle compatibility feature will enable Oracle applications to run natively on DB2. In discussions with Gartner, reference customers tell us that **DB2 runs 95% or more of Oracle specific functionality found in SQL statements and natively runs PL/SQL**, Oracle’s stored procedure language. This **native functionality is not an emulator**, nor does it require changes to the application code (other than the 5%, which is mostly minor functionality, not found in many applications). Any provider offering applications that run with the Oracle DBMS can **easily port its application to IBM’s DB2...**”

— Donald Feinberg, Gartner VP Distinguished Analyst

openbravo[®]
opening ERP's future!

“To move our application to a previous version of DB2 would have taken an estimated two-year effort. **We were thrilled to see it took only one week to move it to the new version of DB2..**” — Paolo Juvara, CTO of Openbravo

New SQL and Improved Performance

Overview of SQL Enhancements

- **New Scalar Functions**
- **Create Global Temp Tables**
 - Real objects can be accessed by multiple connections
- **Truncate Table**
 - Simple way to clean out a table without logging rows
- **Statement Concentrator**
 - Promotes sharing of access plans for better performance
- **Scan sharing**
 - Better workload concurrency and performance without requiring any expensive hardware upgrades

New Scalar Functions

- **ADD_MONTHS**
- **DAYNAME**
- **DECFLOAT_FORMAT** or **TO_NUMBER**
- **EXTRACT**
- **INITCAP**
- **LAST_DAY**
- **LOCATE_IN_STRING** or **INSTR**
- **LPAD**
- **MONTHS_BETWEEN**
- **MONTHNAME**
- **NEXT_DAY**
- **ROUND_TIMESTAMP**
- **RPAD**
- **TO_CLOB**
- **TRUNC_TIMESTAMP**
- **more to come...**

Created Global Temporary Tables

- **Previous releases**
 - Global temporary tables could only be declared (DGTT)
 - DGTT must be declared in every session
 - Cannot build other database objects on top of DGTT
- **DB2 Cobra**
 - Global temporary tables can be created (CGTT)
 - CGTT defined in the catalogs – users simply reference them
 - CGTT are under authorization control
 - Views, indexes, and triggers can be defined on CGTT
 - CGTT (and DGTT) can contain LOBs

Truncate Table

- **Previous releases**
 - Delete all the rows – slow and uses up the log
 - Load an empty file – ugly
- **DB2 Cobra**
 - New statement is fast and tidy

Statement Concentrator

- **Previous releases**

- Every unique statement is compiled – can be a significant overhead for some workloads
 - Statements generated with literals instead of parameter markers
 - Typically seen in PERL, PHP, RUBY

- **DB2 Cobra**

- Optionally replace literals with parameter markers
 - Increases section sharing and reduces compilation
- Reduces number of statements to be compiled
- Must be explicitly requested

Even Faster OLTP with Statement Concentrator

DB2 9.7

- Optionally replace literals with parameter markers
 - Increases section sharing and reduces compilation
- Reduces number of statements to be compiled

SELECT BALANCE WHERE ACCOUNT_ID = 12345

SELECT BALANCE WHERE ACCOUNT_ID = 11111

SELECT BALANCE WHERE ACCOUNT_ID = 54321

SELECT BALANCE WHERE ACCOUNT_ID = 12121

Compile

Execute

Even Faster OLTP with Statement Concentrator

DB2 9.7

- Optionally replace literals with parameter markers
 - Increases section sharing and reduces compilation
- Reduces number of statements to be compiled

SELECT BALANCE WHERE ACCOUNT_ID = 12345

SELECT BALANCE WHERE ACCOUNT_ID = 11111

SELECT BALANCE WHERE ACCOUNT_ID = 54321

SELECT BALANCE WHERE ACCOUNT_ID = 12121

SELECT BALANCE WHERE ACCOUNT_ID = ?

Compile

Execute

Scan Sharing

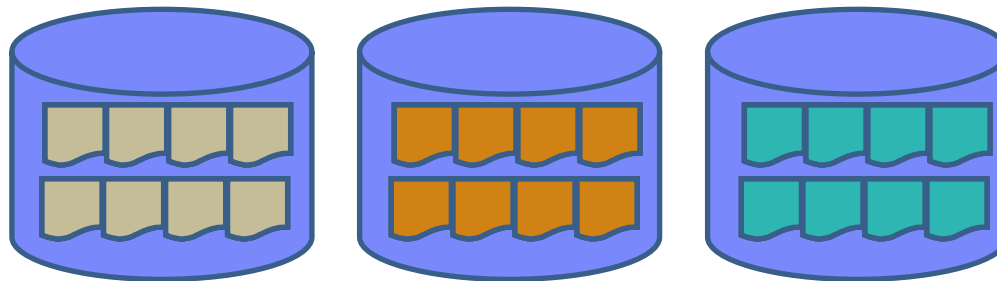
- **Scan Sharing**

- Focus on concurrent table, range and block scan sharing
- New scan will start based on current scan position
- When it reaches end of file it will wrap and finish when it reaches the starting point
- Runtime decision on whether scan will or will not participate
- Shared scans are run in “share groups”
 - faster scanner may be throttled so it does not get far ahead of group.

Scan Sharing Example

- **User 1 starts scanning a table**

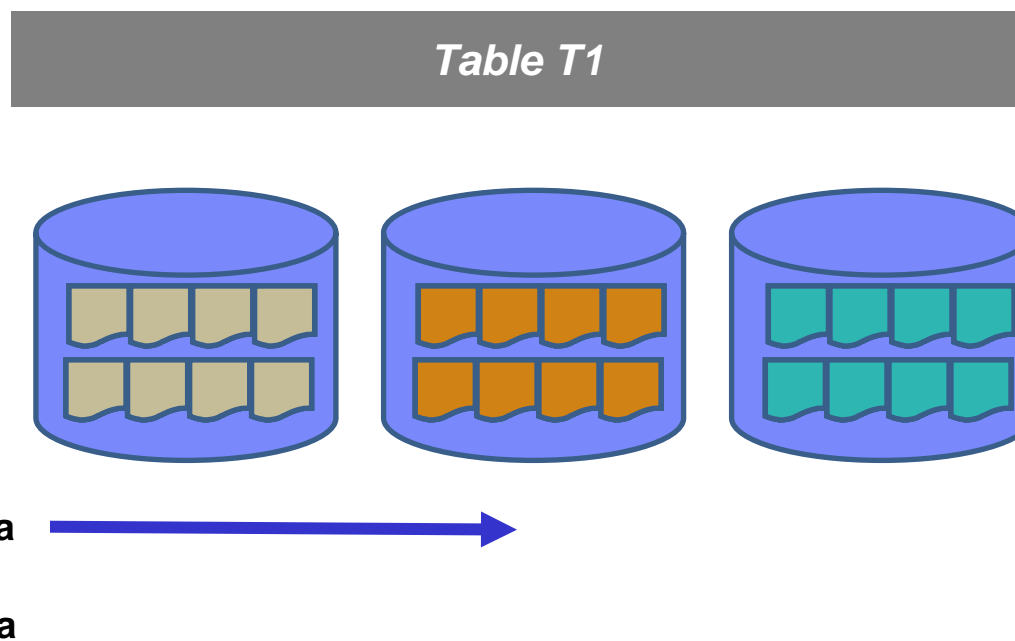
Table T1



User 1 Scans Data →

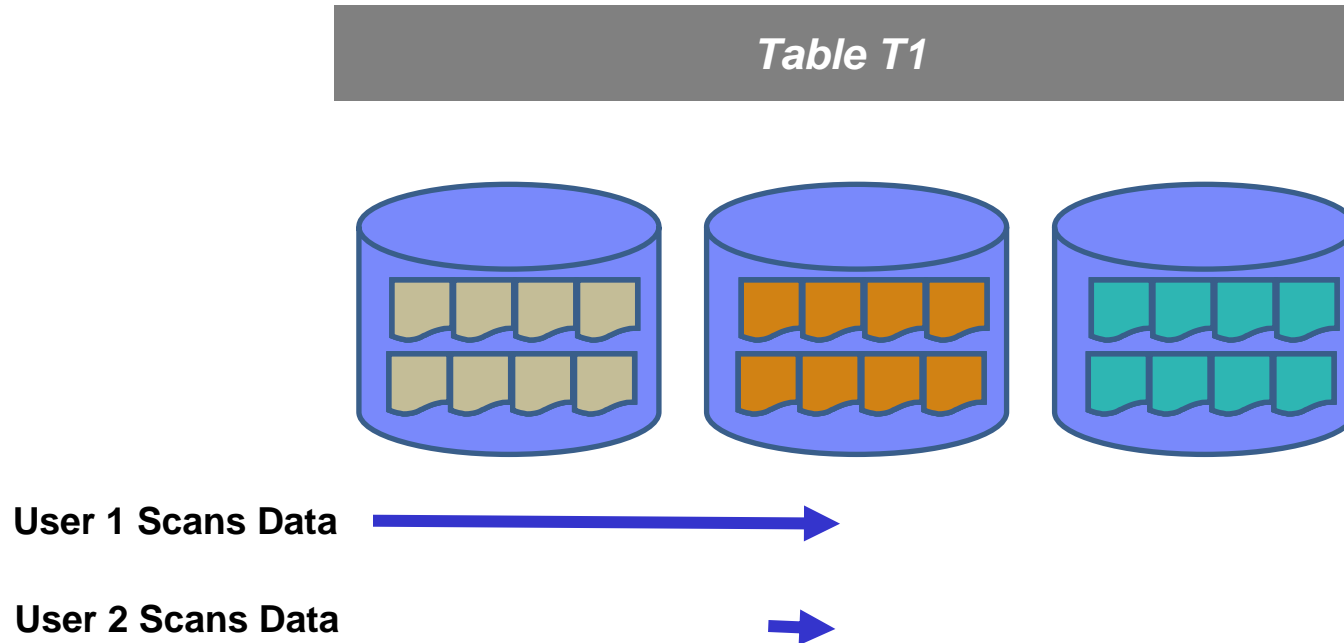
Scan Sharing Example

- **User 1 is halfway through the table scan when user 2 starts their scan**



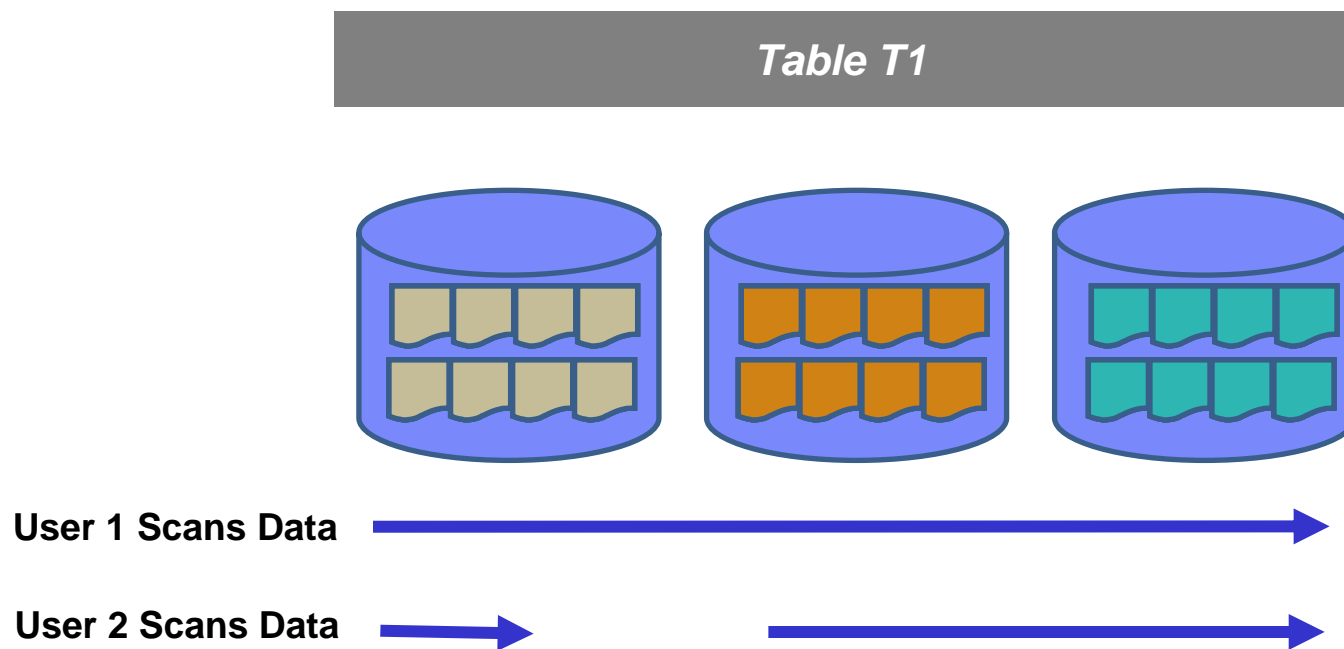
Scan Sharing Example

- Instead of starting at the beginning of the table, User2 starts where user1 is currently scanning



Scan Sharing Example

- **When User 1 is finished, user2 will go back and read the pages it missed**



And That's Just the Tip of the 9.7 Iceberg

- **New Monitoring Capabilities**
- **Enhanced Workload Management**
- **New High Availability features**
- **Online Table Move Capability**
- **More Online Schema Change Ability**
- **Enhancements to pureXML**
- **And much, much, more**

