



DB2 9.7 Highlights

Jason Chuang
Consulting I/T Specialist
IBM Taiwan SWG

Data Management

DB2 Business Value

Unparalleled reliability and scalability for the changing needs of your business

- **Application Freedom**
 - Use the database server that gives you the freedom to choose
- **Service Level Confidence**
 - Expand your critical workloads confidently and cost effectively
- **XML Insight**
 - Harness the business value of XML
- **Storage TCO Reduction**
 - Making a smart use of your storage space



Application Freedom

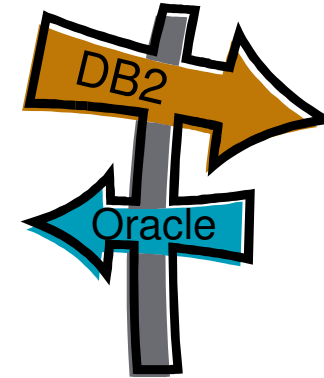
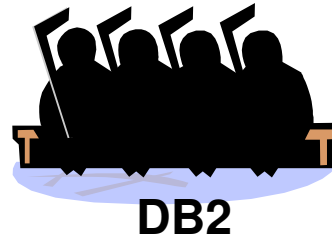
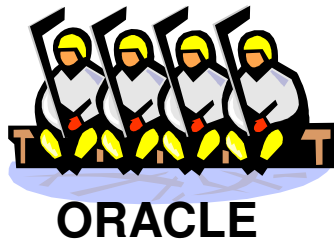
Data Management

Inhibitors to Migration

- **Impedance mismatch between source and target database features**
 - Data types, locking model, weak typing, packages, ...
- **Workarounds → Poor performance**



- **Lack of skills in development team**



THIS IS ALL HISTORY NOW!

LOOK INTO THE FUTURE:
DB2 9.7



Switching to DB2 with V97

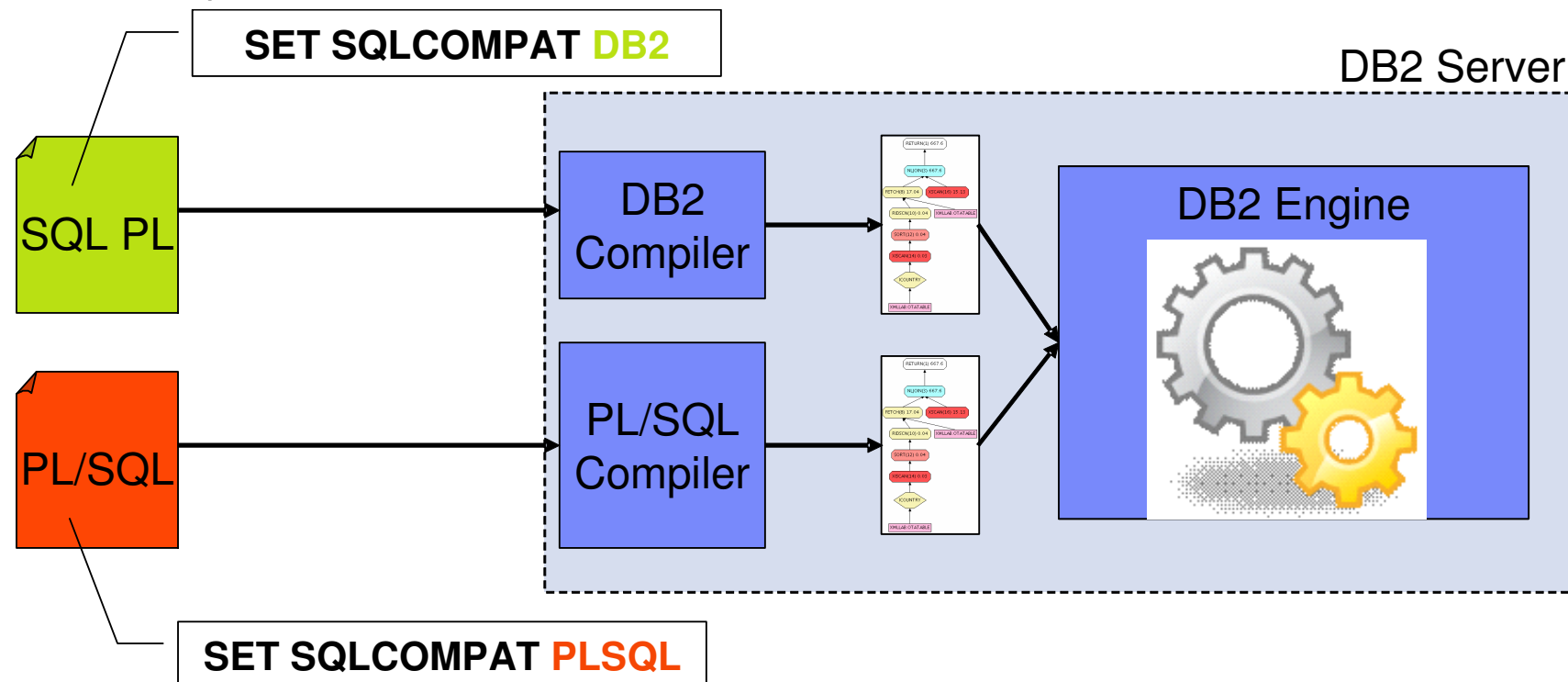
Oracle	→	DB2
Concurrency Control	→	Native support
SQL	→	Native support
PL/SQL	→	Native support
Packages	→	Native support
Built-in packages	→	Native support
OCI	→	Native support
JDBC	→	Native support
Online schema changes	→	Native support
SQL*Plus Scripts	→	Native support

Changes are the exception, not the rule.
This is why we call it “enablement” not “migration”.

Oracle application enablement made easy

▪ New DB2 Cobra's Compatibility Features

- New Registry variable: DB2_COMPATIBILITY_VECTOR
- PL/SQL language is supported by DB2 interface
 - SET SQLCOMPAT PLSQL – command to setup the CLP environment so that it can compile PL/SQL code



Oracle types in DB2 Cobra

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

TIMESTAMP WITH TIMEZONE

INTERVAL

Oracle functions in DB2 Cobra

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 Cobra

Feature	Comment
CONNECT BY	Tree walk recursion, includes helper functions
(+)-join	Old style OUTER JOIN syntax
DUAL	Equivalent to SYSDUMMY1
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

PL/SQL Features in DB2 Cobra

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 Cobra

Area	Comment
Anonymous block	New also in SQL PL dialect
Scalar function	
Procedure	
Package	Known as MODULE in DB2
Trigger	

PL/SQL Package in DB2 Cobra

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

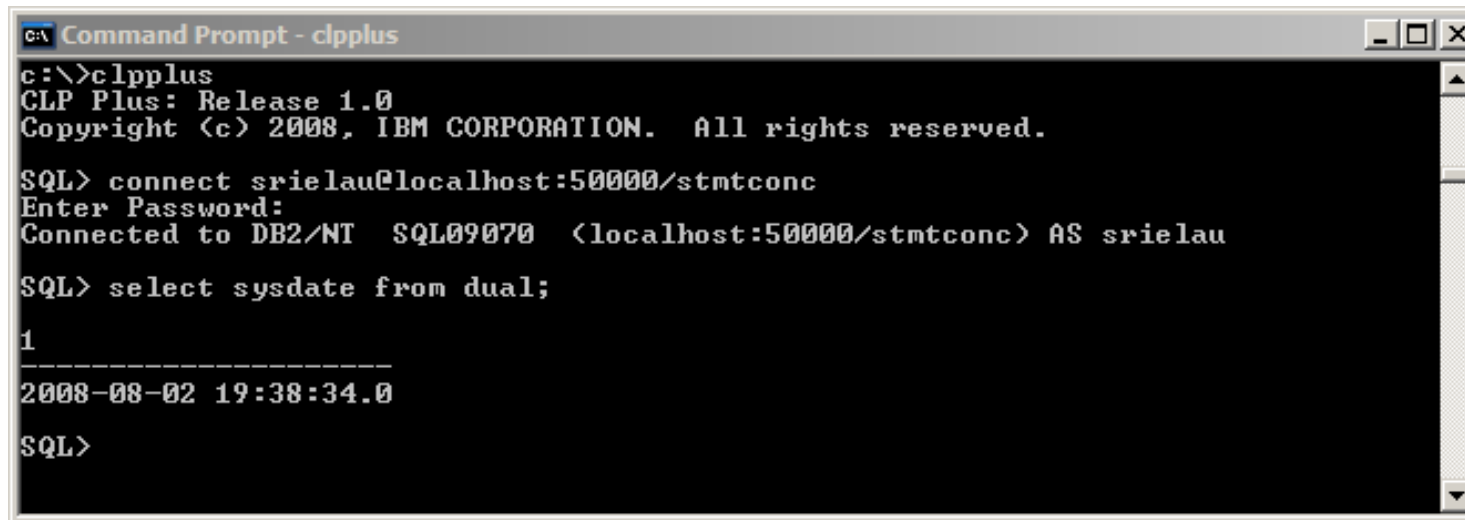
- DB2 shreds package and body into individual *module* objects
- External management view is preserved

Built-in package libraries in DB2 Cobra

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

Using SQL*Plus scripts in DB2 Cobra

- **CLPPlus**
 - SQL*Plus compatible command
 - Variable substitution
 - Column formatting
 - Simple reporting
 - Control variables



```
c:\ Command Prompt - clpplus
c:\>clpplus
CLP Plus: Release 1.0
Copyright (c) 2008, IBM CORPORATION. All rights reserved.

SQL> connect srielau@localhost:50000/stmtconc
Enter Password:
Connected to DB2/NT  SQL09070  (localhost:50000/stmtconc) AS srielau

SQL> select sysdate from dual;

1
-----
2008-08-02 19:38:34.0

SQL>
```

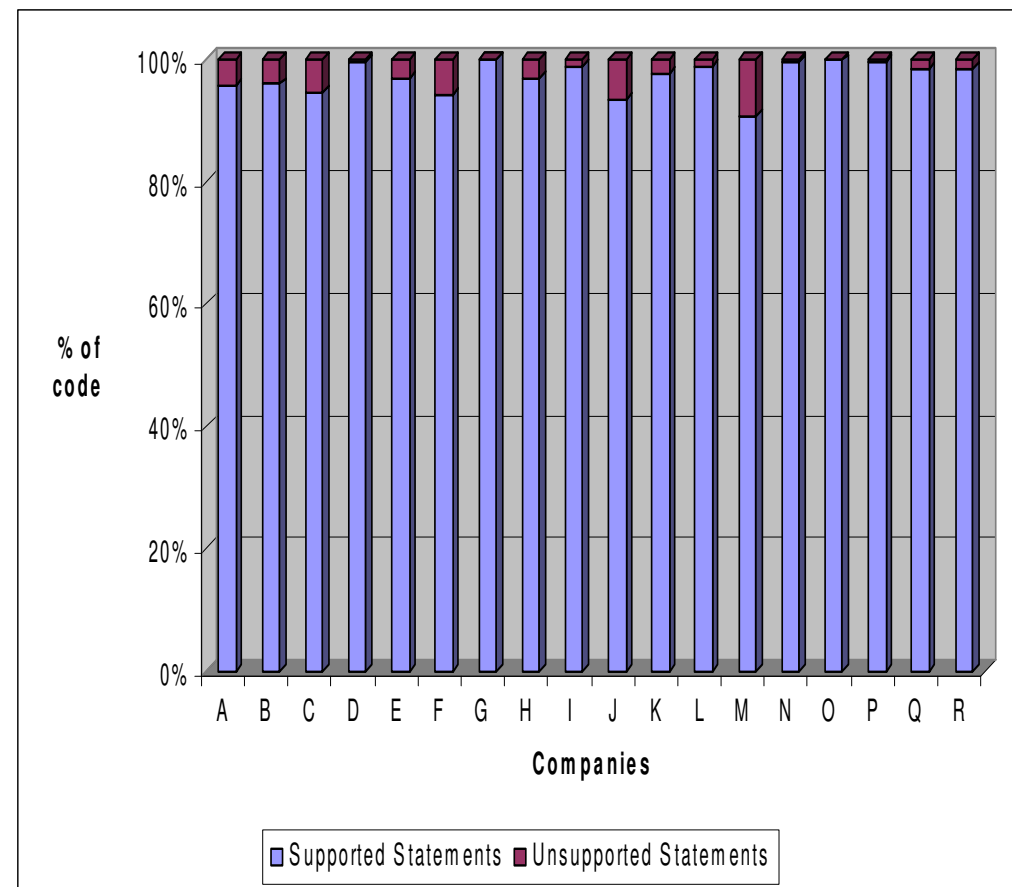
Percentage of Supported PL/SQL Statements

- **Variety of participants:**

- Different industries
- Different solutions
- Different app sizes
- Different countries

- **PL/SQL supported:**

- > 750,000 lines of code
- Average: 98.43%



Migration/Porting Before DB2 9.7

- **Map schema and data types (with DB2 MTK)**
- **Move data (with DB2 MTK)**
- **Translate (semi-automated)**
 - Triggers
 - Procedures
 - Functions
 - Anonymous blocks
- **Translate SQL in application logic (manual)**
- **Debugging**
- **Test and Tuning (including selective redesign)**

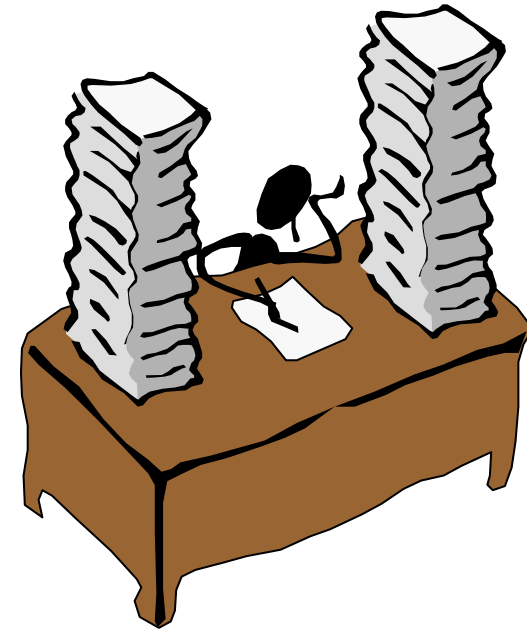
- **→ Repeat for every new release of the application**



Migration/Porting with DB2 9.7

- **Map schema and data types (with DB2 MTK)**
- **Move data (with DB2 MTK)**
- **Native Support**
 - Triggers
 - Procedures
 - Functions
 - Anonymous blocks
- **Translate SQL in application logic (manual)**
- **Debugging**
- **Test and Tuning**

- → Repeat for every new release of the application



Demo of running Oracle code right on top of DB2

```

CREATE TABLE customer(
  customer_id NUMBER(10) NOT NULL ,
  first_name VARCHAR2(30) NOT N /* Example of new built-in packages: UTL_DIR, UTL_FILE and DBMS_OUTPUT */
  last_name VARCHAR2(40) NOT NU
  email VARCHAR2(100) NOT NULL,
  phone_number char(14) NOT NULL
  birth_date date NOT NULL,
  registration_time TIMESTAMP(0)
);

PROCEDURE write_catalog( message OUT VARCHAR2) IS
  v_filehandle UTL_FILE.FILE_TYPE;
  v_filename VARCHAR2(100) DEFAULT 'catalog.out';
  v_temp_line VARCHAR2(100);
BEGIN
  message:='';
  UTL_DIR.CREATE_DIRECTORY('mydir1', 'C:\\');
  v_filehandle := UTL_FILE.FOPEN('mydir1',v_filename,'w');
  IF (UTL_FILE.IS_OPEN( v_filehandle ) <>1 ) THEN
    DBMS_OUTPUT.PUT_LINE('Cannot open file');
  END IF;
  FOR i IN
    (SELECT level, category_name FROM category
     START WITH category_name = 'Main' CONNECT BY PRIOR category_name = category_parent_name)
  LOOP
    UTL_FILE.PUT_LINE(v_filehandle,i.level || LPAD(' ', 4 * i.level - 1) ||i.category_name);
  END LOOP;
  UTL_FILE.FCLOSE(v_filehandle);

  BEGIN
    v_filehandle := UTL_FILE.FOPEN('mydir1',v_filename,'r');
    LOOP
      UTL_FILE.GET_LINE(v_filehandle, v_temp_line);
      message:=message || chr(10) || v_temp_line ;
    END LOOP;
  EXCEPTION
  WHEN no_data_found THEN
    NULL;
  END;

  UTL_DIR.DROP_DIRECTORY('mydir1');
END write_catalog;

```

```

Statement 1      SUCCEEDED
  select sysdate from dual

1
-----
2009-02-04 20:55:04

```

Concurrency and DB2 Cobra

- **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	YES
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

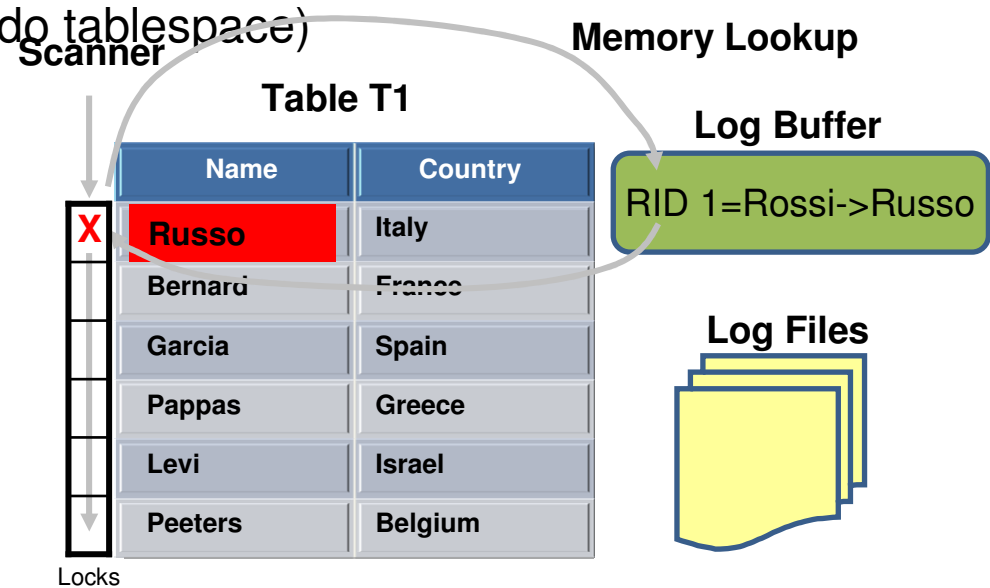
Concurrency Control in DB2 Cobra

- 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`



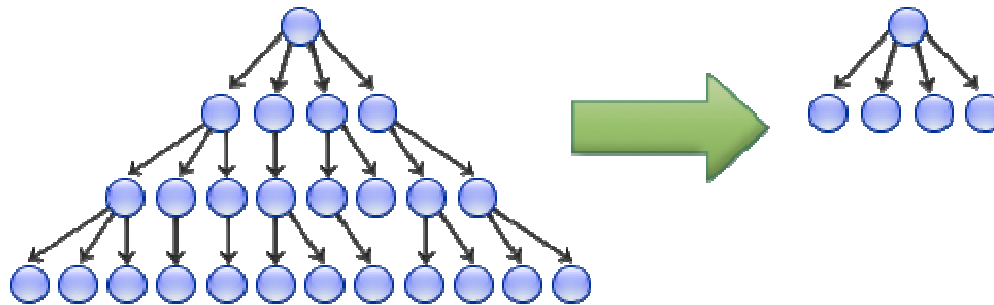


Storage Optimization Enhancement

Data Management

Improvements to Compression

- Multiple algorithms for automatic index compression



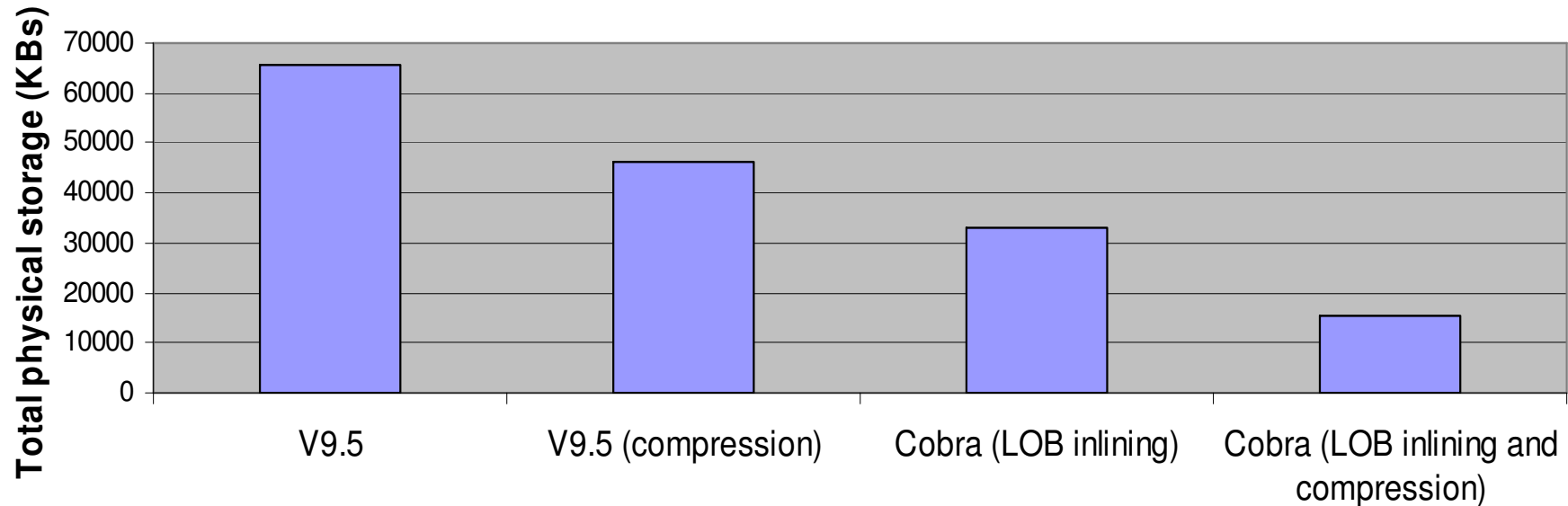
Unique in the industry



Unique in the industry



DB2 - XML and LOB In-lining: Example



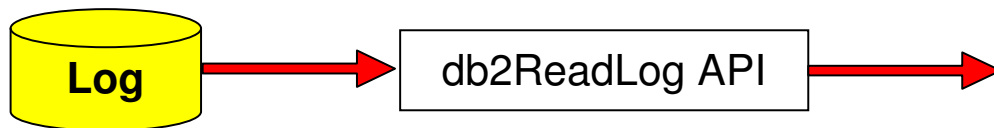
	V9.5	V9.5 (compression)	Cobra (LOB inlining)	Cobra (LOB inlining, compression)
Base table (KBs)	24320	5248	22144	4736
LOB storage (KBs)	30336	30336	128	128
Indexes (KBs)	10880	10752	10880	10752
Total (KBs)	65536	46336	33152	15616

- All LOBs in this table are smaller than the default inline length for the LOB columns
 - 100% of the LOBs for this table, inlined in Cobra!

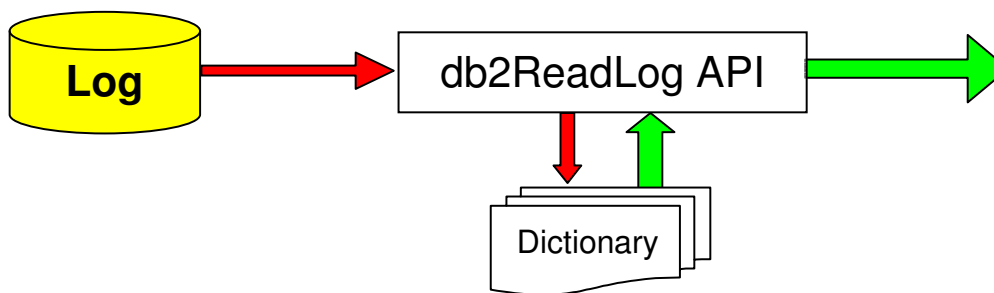
DB2 – New Replication Support for Compression

- **Replication (CAPTURE) support for compression**
- **Compression with DATA CAPTURE CHANGES (Replication)**
 - Capture requires uncompressed log data in log records
- **Log data from compressed records stored in compressed format**
 - db2ReadLog API will now decompress log data before returning log records
 - Allowing for two types of dictionaries: current data and historical compression dictionary

DB2 V9.5 with iFilterOption ON



DB2 Cobra with iFilterOption ON



- █ Compressed user data in logs
- █ Uncompressed user data in logs

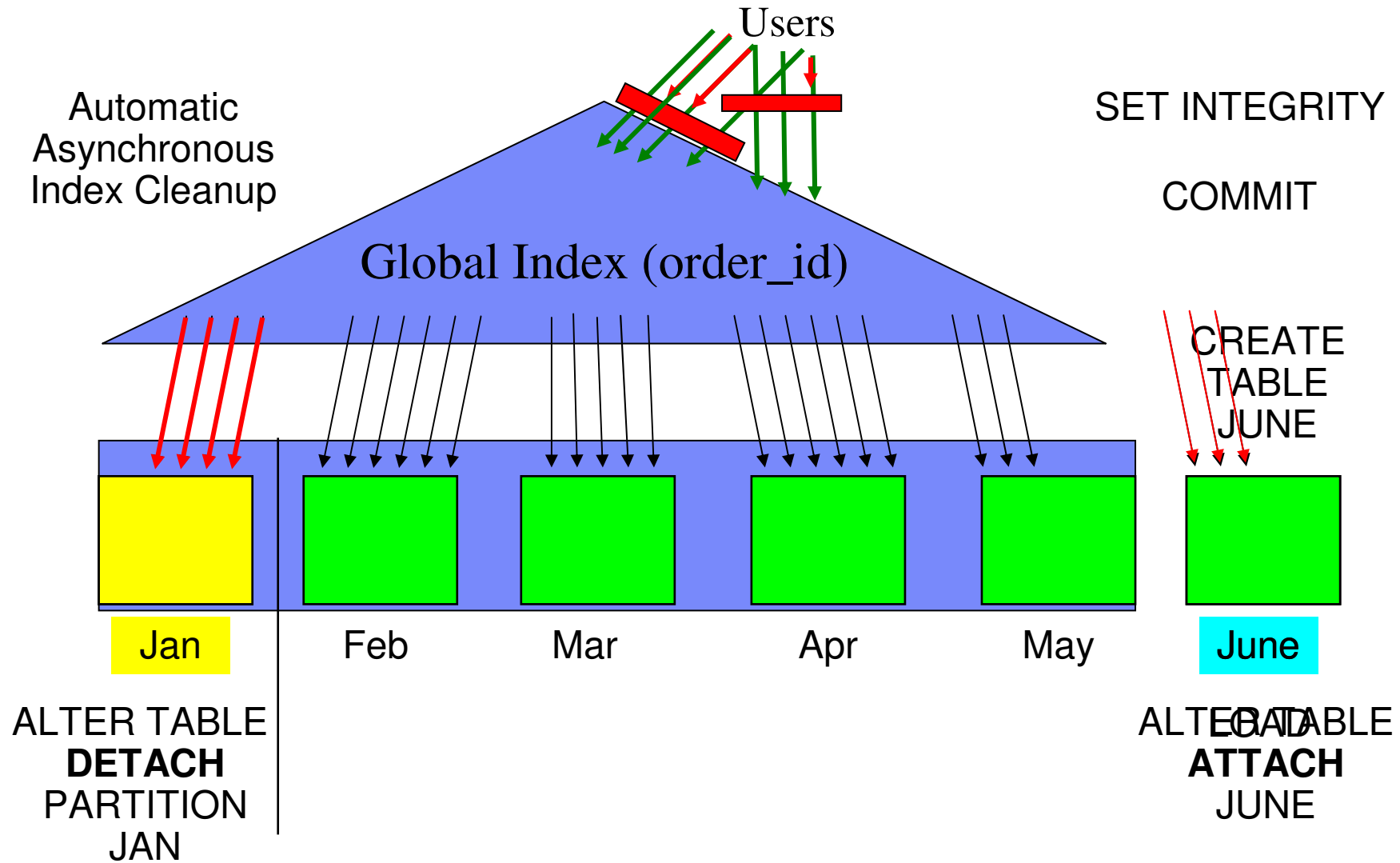
With DATA CAPTURE
We now have two
Compression dictionaries:
- **Historical**
- **Current Data**



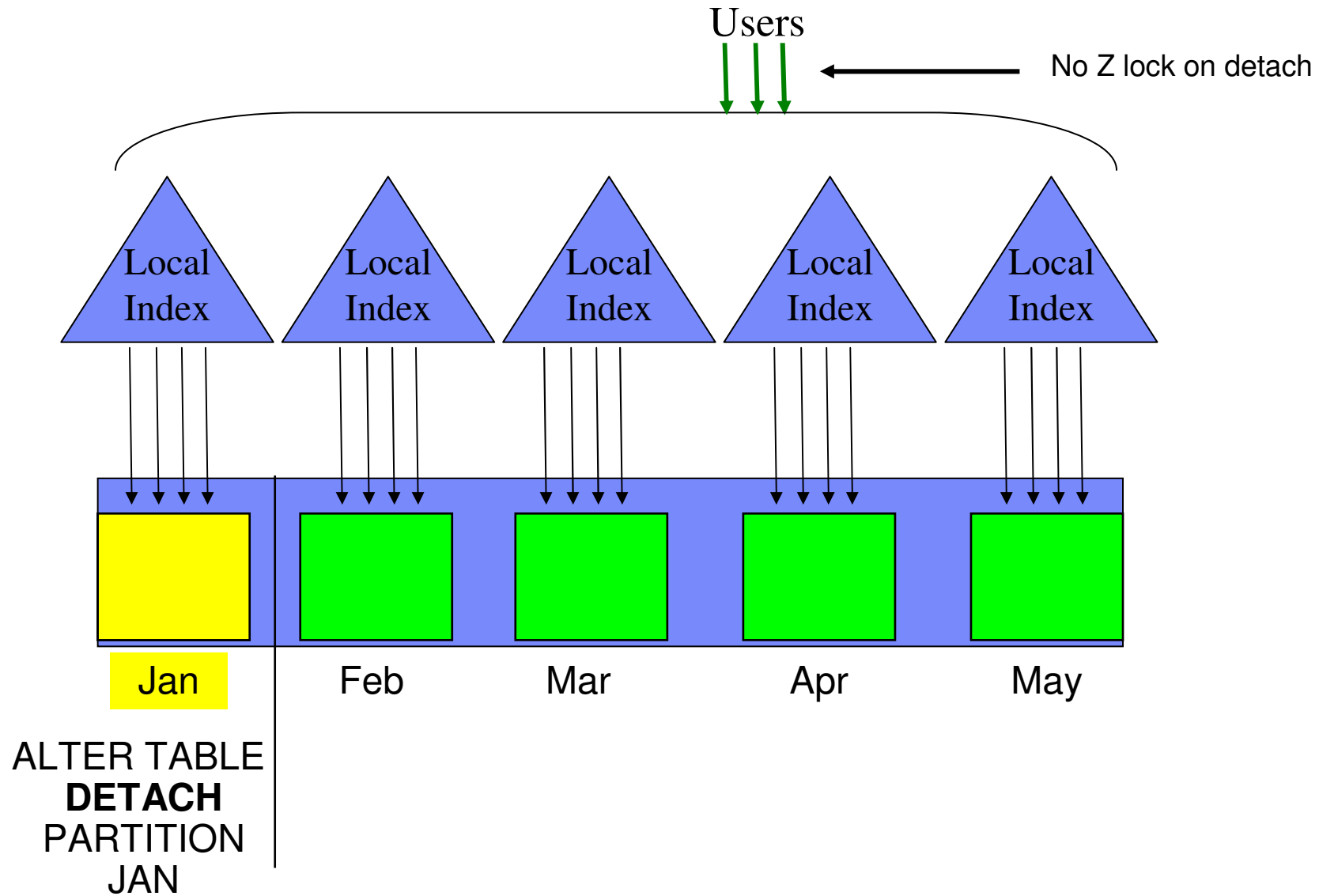
Partitioned Table Enhancement

Data Management

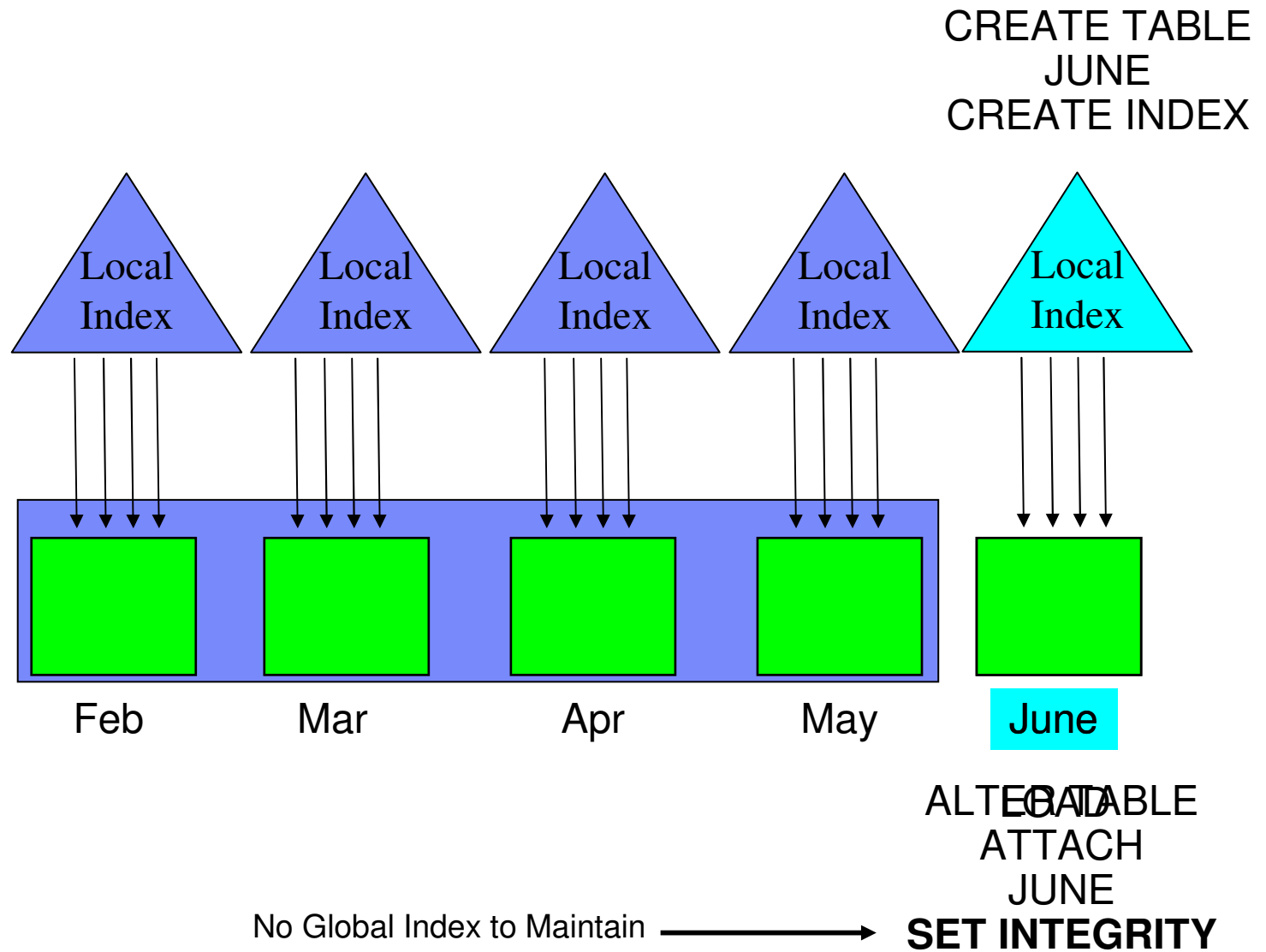
DB2 9.5 Table Partitioning – Roll-in / Roll-out



Cobra Table Partitioning – Non Intrusive Roll Out

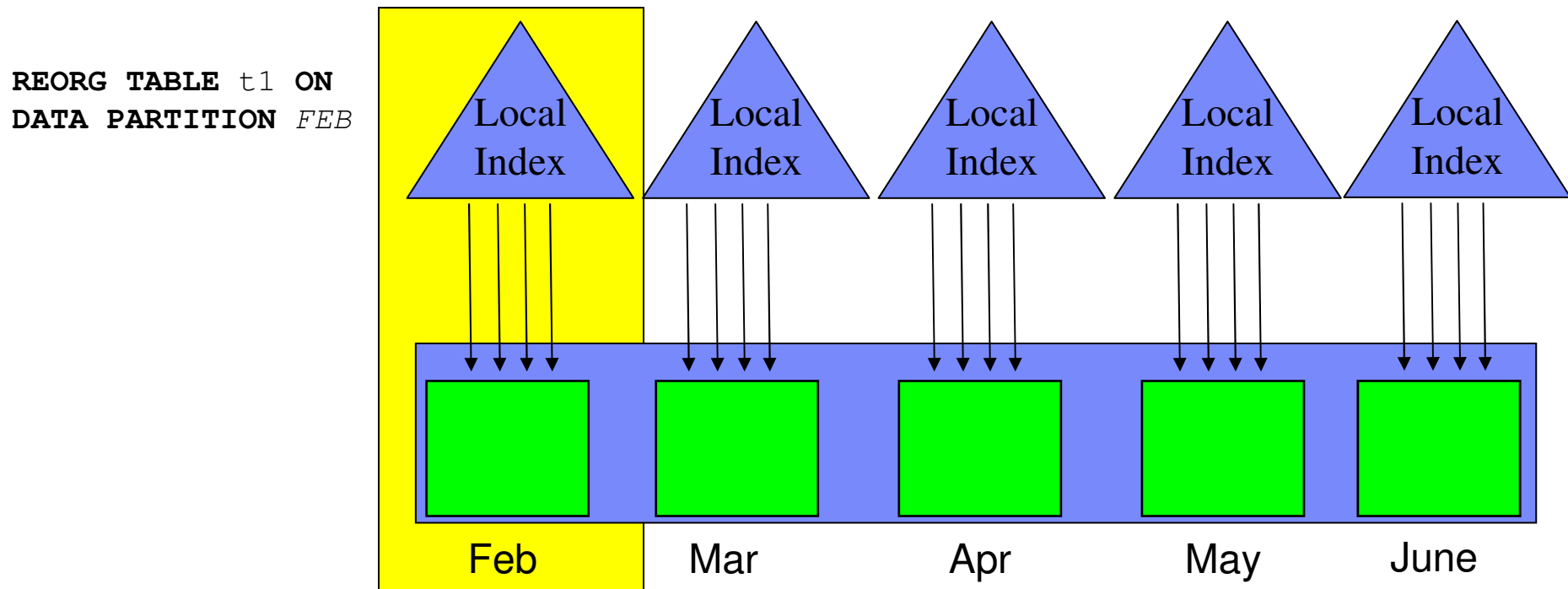


Cobra Table Partitioning – Faster Rollin



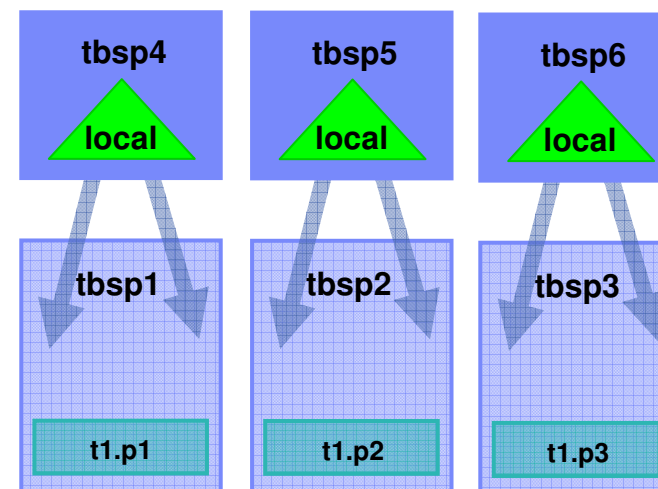
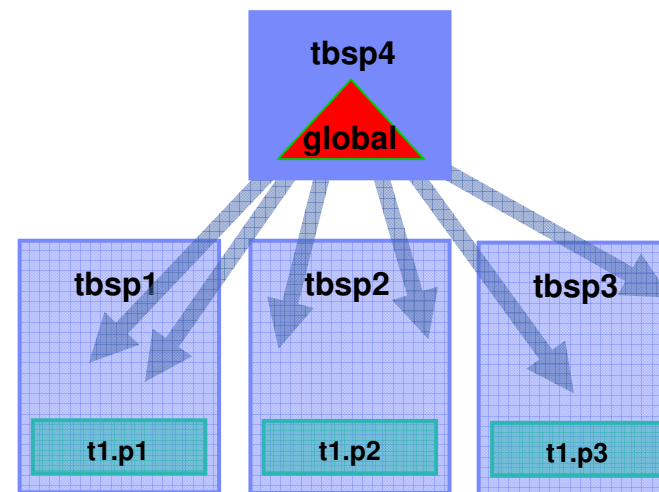
Expected Usage

- **All indexes will be partitioned as long as they can be**
 - Exception: unique indexes where the index key is not a superset of the table partitioning key
- **Increased availability via partition-level REORG**



Partition Elimination : Index Scans Improved

- `SELECT * FROM t1`
`WHERE`
`CUSTID= 23454`
- A single scan will go through the global index to find the corresponding records
- With local indexes, multiple scans can be done on the indexes and certain indexes eliminated if their partition is not part of the answer set





PureXML Enhancement

Data Management

Industry-Leading XML Storage

2. Simple

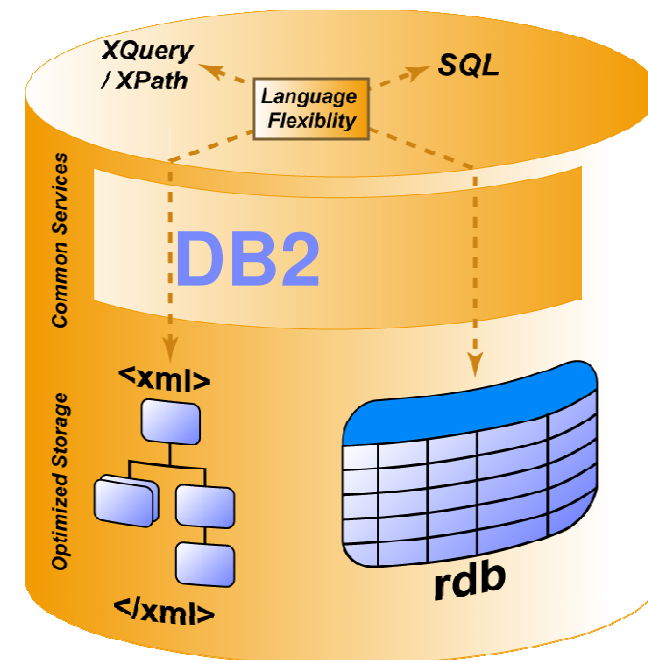


“With [DB2’s] ability to process pureXML, our customers are seeing 5 to 10 times performance improvements.” —*Keith Feingold, CEO, Skytide*



“IBM is... taking a more holistic approach than its competitors to combine XML and relational systems.” —*Forbes.com*

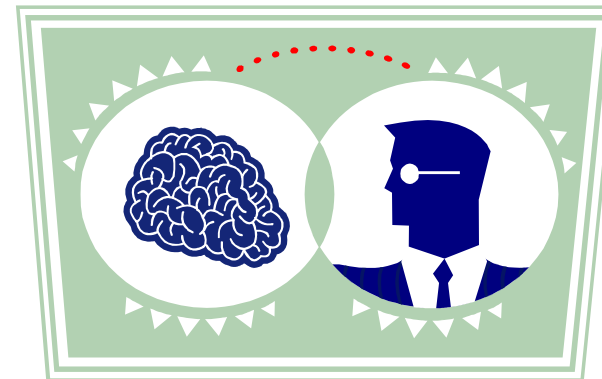
- **Easy development and integration**
 - No complex relational schemas
 - No parsing upon retrieval
- **Efficient storage**
 - Requires only 440GB of raw storage for 1TB XML benchmark
- **Untouchable performance**
 - 6,763 XML transactions per second for 1TB XML benchmark



pureXML enhancements

Unleash your XML

- **Harness the power of understanding the XML in your enterprise**
- **Business Intelligence and XML**
 - XML in data partitions (DPF)
 - XML in range partitions
 - XML in database views
 - XML in Materialized Query Tables
- **Additional Features**
 - XDA Compression
 - Online index reorg
 - Online XML index create
 - XML in UDF, MDC
 - Bulk Decomposition
 - XML DBA Functions
 - Net.Search Extender on DPF, MDC, and Range Partitioned tables

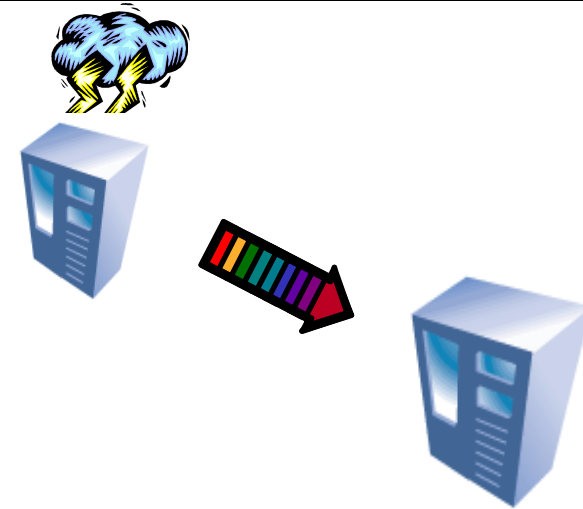




HADR Enhancement

HADR Basic Principles

(Recap of 9.5 functionality)

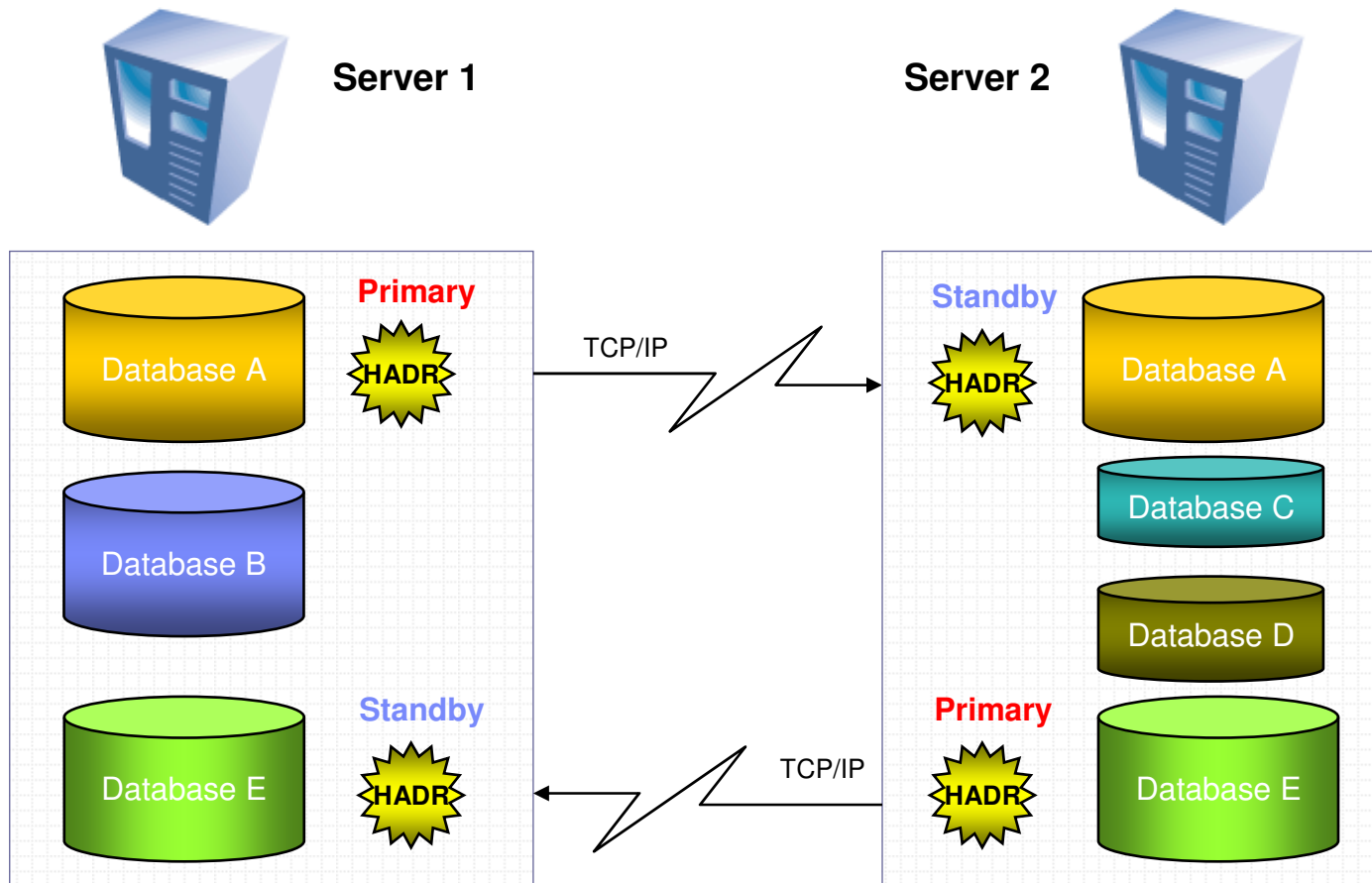


- **Two active machines**
 - Primary
 - Processes transactions using database replication
 - Ships log buffer entries to the standby machine
 - Standby
 - Cloned from the primary
 - Receives and stores log or log buffer entries from the primary
 - Re-applies the transactions
- **If the primary fails, the standby can take over the transactional workload**
 - The standby becomes the new primary
- **If the failed machine becomes available again, it can be resynchronized**
 - The old primary becomes the new standby

HADR Overview Scope of Action

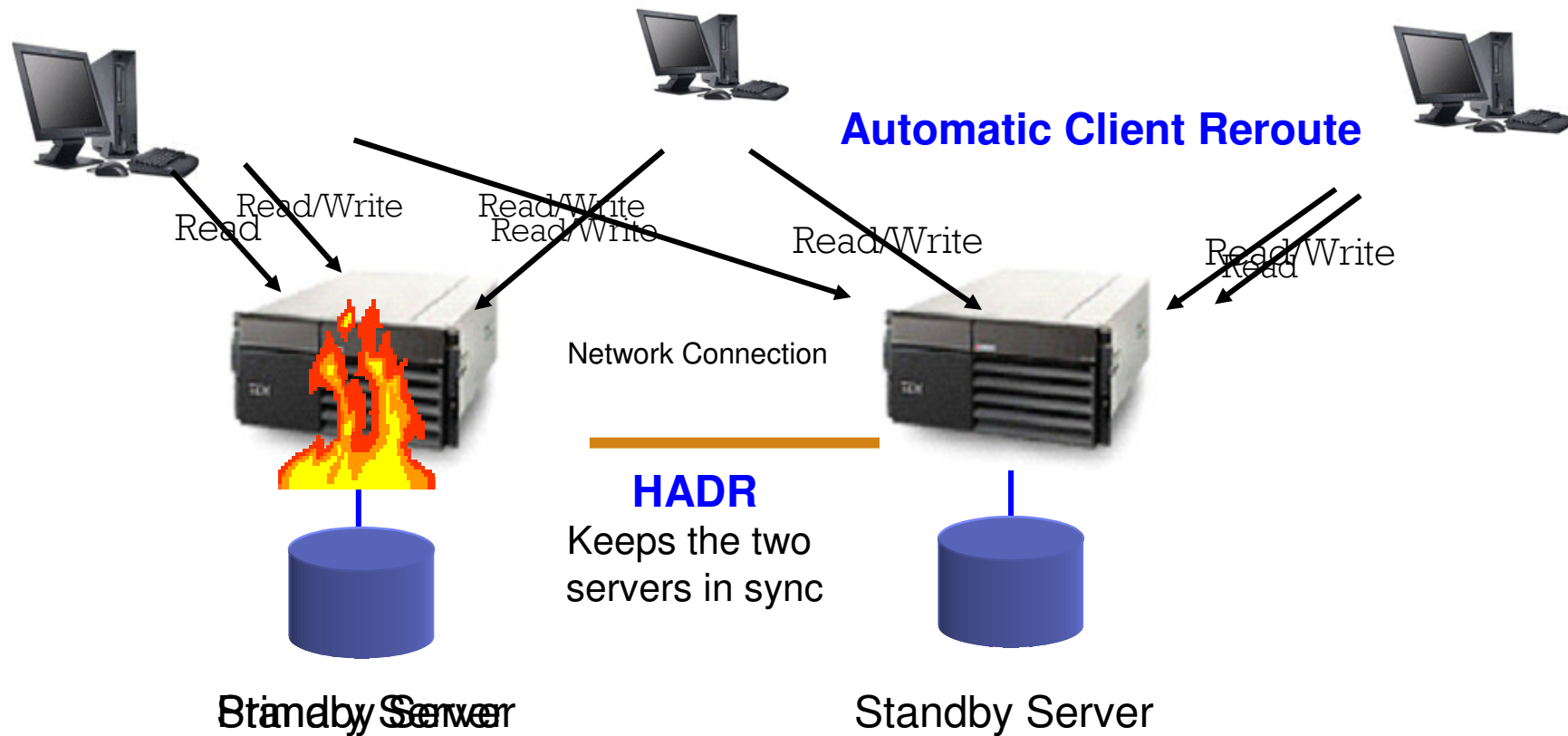
(Recap of 9.5 functionality)

HADR is at the database level



DB2 HADR with V97

- Standby DB support Read access
- Standby DB support Backup



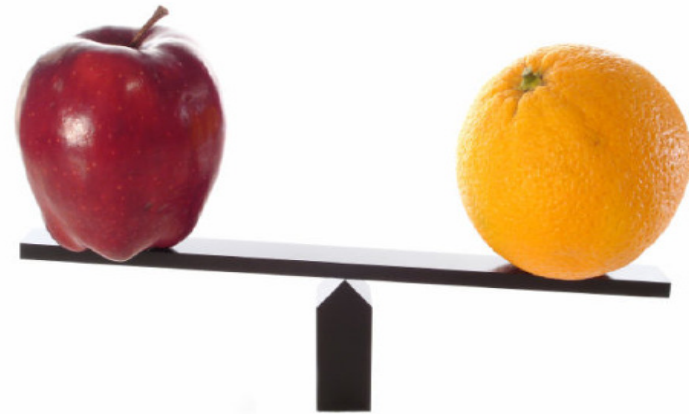


Other Enhancement

Data Management

Flexible Data Typing

- **Traditionally, DB2 has supported strict typing:**
 - In accordance with SQL standard
- **Trend towards weak typing:**
 - For PERL, RUBY, PHP, etc
- **DB2 now supports:**
 - Easy casting between data types
- **Assign or compare strings to numeric types**
 - `SELECT * FROM T WHERE INTCOL = '42'`
- **Allow untyped parameter markers and untyped NULLs in more situations**
 - `SELECT ?, NULL, MY_UDF(?, NULL)`
FROM T



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

Named and Default Parameters

- **Previous releases**

- All parameters to a procedure must be specified in the proper order on CALL

- **DB2 Cobra**

- Parameters to a procedure can be given a default value
- Can skip parameters with default values in CALL
- Named parameters allow values to be assigned to parameters by name, in any order

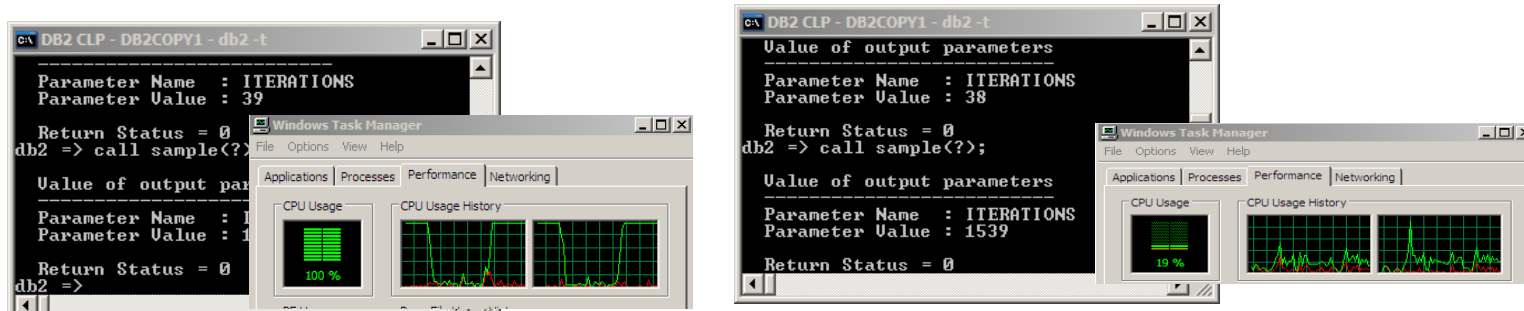
Statement Concentrator

▪ 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



Manageability Enhancements

Easy Table Movement

■ Online Table Move

- Stored procedures to allow online movement of tables
 - Move table online to a different tablespace
- New tablespace may have:
 - larger pagesize, different extent size, or other differing tablespace properties

■ Transportable Tablespaces

- Efficient schema movement between databases
- Extract DDL and other dependent objects
- Directly reference containers of tablespace in target

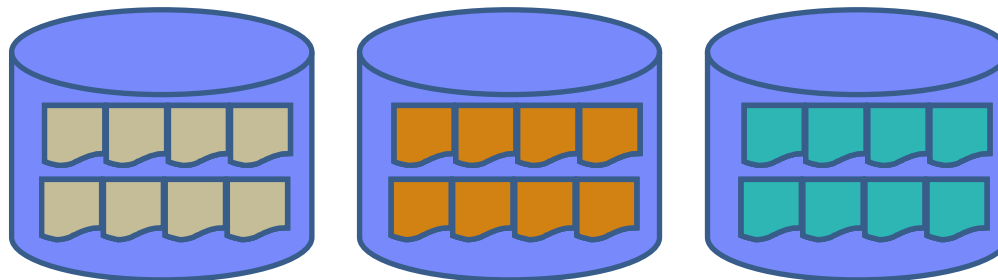


Scan Sharing

▪ 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.
- Not Supported for:
 - Index Scans
 - Utilities, catalog services, index creation
 - SMP parallelism

Multiple Scanners pre Cobra



User 1 Scans Data



User 2 Scans Data



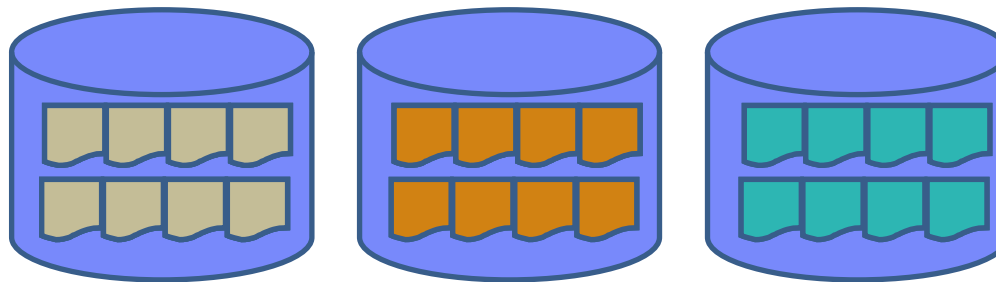
Multiple Scanners with Cobra

Reread only missing pages

Buffer Pool



Start scan 2 at current position of scan 1



User 1 Scans Data



User 2 Scans Data





DB2 9.7 Highlights

Data Management