

# ***Unlocking the Page in DB2\**** ***- the Sequel -***

**International DB2 Users Group**  
**3rd Annual Asia Pacific Conference**  
**19-21 November, 1997**  
**Sydney, Australia**

**Willie Favero**

**BMC Software, Inc.**

---

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## **Abstract:**

DB2 Version 4 is now Generally Available. And in all of the discussions about Type 2 Indexes and Data Sharing, one term keeps popping up: *Lock Avoidance*. This technique, introduced in DB2 Version 3, allows DB2 to use latches rather than locks. There has been a limited amount of information describing lock avoidance available in the DB2 publications. This presentation attempts to fill that void. An update to the top 10 IDUG '95 presentation "Unlocking the Page in DB2", new information from the most current releases of DB2 has been added to help explain lock avoidance. This includes the details behind lock avoidance, a discussion on what a DBA or Application Programmer needs to be aware of, an explanation of CURRENTDATA, and some of the pitfalls that might be encountered when implementing this new technique. The new Isolation levels Read Stability and Uncommitted Read are also discussed..

### **Key Bullets:**

- 1). Review of locking prior to DB2 Version 3
- 2). The concept of Latching as introduced in Version 3
- 3). Detail descriptions of CLSN testing
- 4). Detail descriptions of PUNC testing
- 5). Description of CURRENTDATA and its effect on locking in Version 3 & 4
- 5). Overview of Isolation levels UR and RS (as introduced in Version 4)

### **Outline:**

- I). Locking in Review
  - a). What is a lock
  - b). Lock flow prior to V3
  - c). The IRLM and why it should be avoided
  - d). Isolation levels
- II). Lock Avoidance
  - a). When is it used
  - b). Read Only Queries
  - c). Testing
    - 1). CLSN Test
    - 2 ). PUNC Test
  - d). CURRENTDATA
- III). When latches can and cannot help
- IV). Isolation Level RS and UR
  - a). How they work
  - b). When should they be used
  - c). What to expect from EXPLAIN



BMC Software, the BMC Software logo and all other product or service names are registered trademarks or trademarks of BMC Software, Inc. in the USA and in other select countries. ® and ™ indicate USA registration or USA trademark. \*DB2 is a registered trademark of International Business Machines Corp. Other logos and product/trade names are registered trademarks or trademarks of their respective companies.

**Copyright © 2001 BMC Software, Inc.  
All rights reserved.**

## NOTES

# Agenda



Introduction  
Locking - Version 2 -Version 4  
Latches  
CSLN Test / PUNC Test  
CURRENTDATA  
Ambiguous Cursors  
Application Considerations  
Isolation Level  
What's Next

Slide 4 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Additional detail about locking and avoiding locks in DB2 can be found in:

**DB2 for OS/390 Administration Guide (Volume II) - SC26-8957**

and

**Locking in DB2 for MVS/ESA Environment - SG24-4725.**

# In the Beginning

Slide 5 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

---

## NOTES

# Why DB2 Uses Locks



**Prevent Read of Un-Committed Page**

**Prevent Access to  
Physically Inconsistent Page**

**Repeatable Read**

Slide 6 of 33

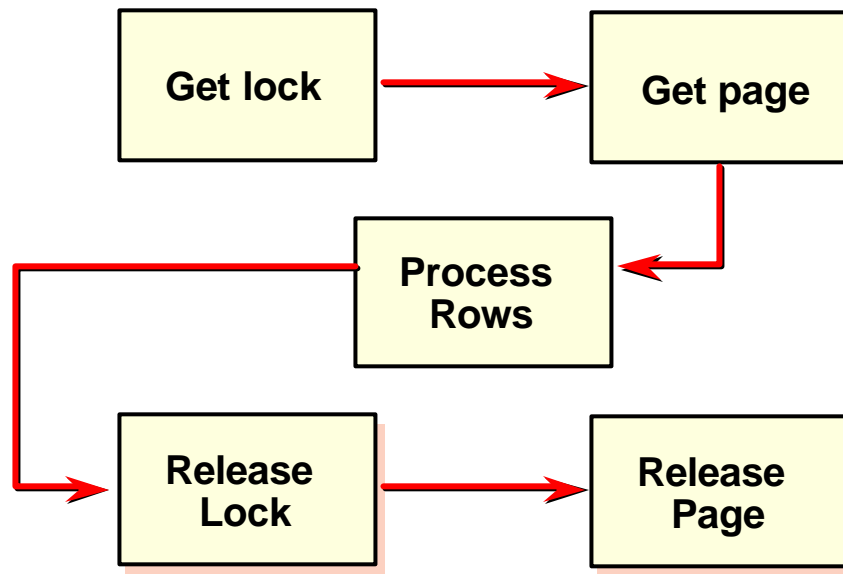
Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

There are three basic reasons why DB2 uses a lock in Version 2. First, to guarantee that the page about to be read has been committed. An S-Lock cannot be obtained on a page being accessed for change with an X-Lock or U-Lock. DB2 will not allow an SQL statement to access an uncommitted page being modified by another user. The second reason is to prevent access to a page that is physically inconsistent. When DB2 is attempting to insert a row into a page, DB2 knows that there is room for the row on that page from the free space information in the header portion of the data page. However, this free space may not be contiguous. In that case, DB2 will compact the page at the time of the insert. While the page is being compacted, rows are being moved and map-IDs are being adjusted. No one else can access this page while this is going on. Therefore, the page is locked. The third, and final, reason is in support of repeatable read.

# Locking in V2 - Overview



Slide 7 of 33

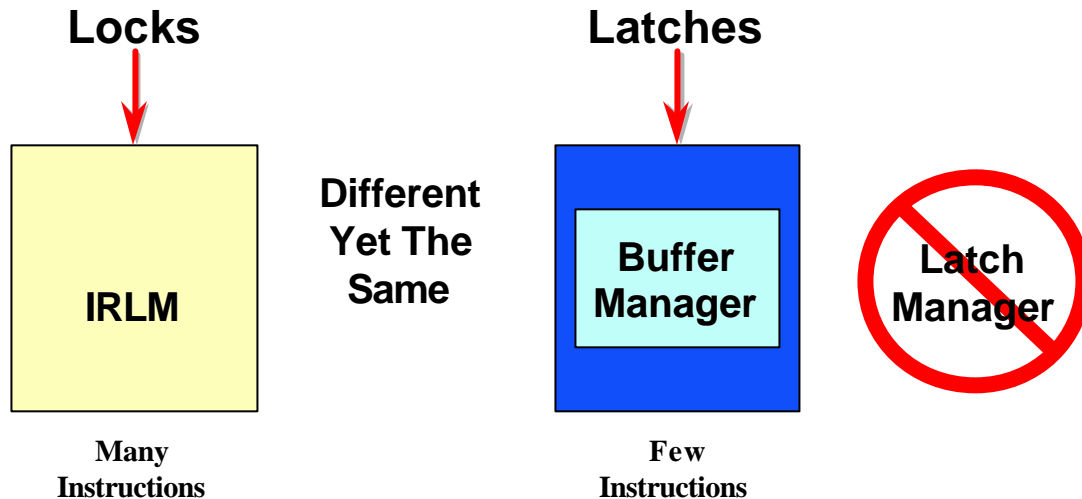
Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

In DB2 Version 2, a lock had to be obtained before a page could be accessed. This means that DB2 locks the page before it even knows if a row qualifies on that page.

# Locks vs Latches



Slide 8 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

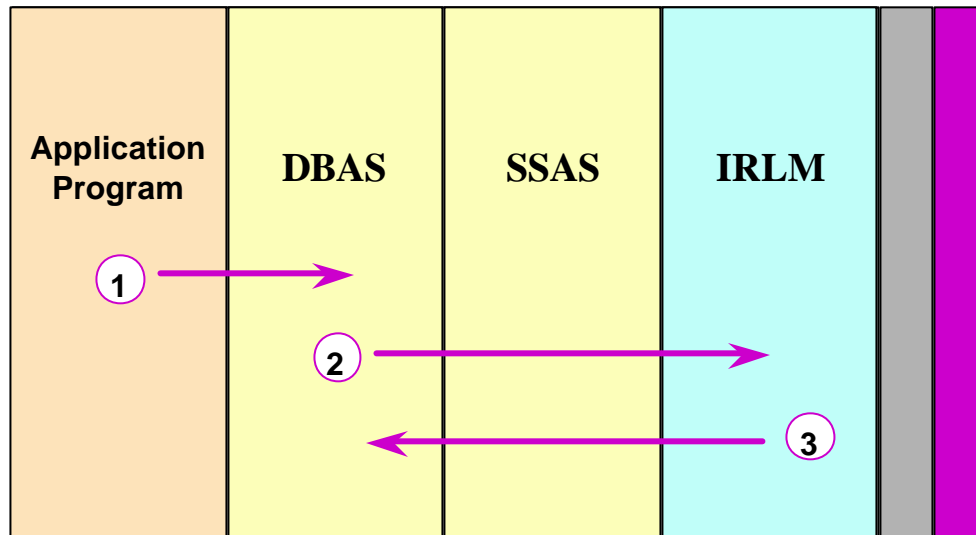
## NOTES

Page locks in Version 2 and page latches in Version 3 behave very similarly. The major difference between the two is how they are managed. A page lock is maintained by using the IRLM, requiring an MVS Cross Memory call and a search of the hash chain in the IRLM each time a page is referenced. A page latch in Version 3 is managed by the Buffer Manager at the program level. The code path for obtaining a page latch is considerably shorter than the path to access the IRLM.

Version 2.3 did use a form of latches for system objects and indexes. These latches still exist and are maintained by the Latch Manager sub component of DB2. These latches were used for a completely different purpose from the latches introduced in DB2 Version 3.



# Address Space Flow



Slide 9 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Application programs request data from DB2 by issuing a call to the DB2 language interface. The request is "passed" to the Database Services Address Space. The Database Services Address Space actually performs the I/O for the application program. However, before a page request can be completed, DB2 must check to see if a lock already exists for that page. If one does, DB2 must verify that the type of lock needed by the current page request does not conflict with any locks already held on that page. DB2 accomplishes this by checking the IMS/VS Resource Lock Manager (IRLM) to see if another application has already requested access to that page. It checks the lock requested by the second application to see if it can exist with the lock being held by the first application. Information is passed to the IRLM using MVS cross memory services and page lock data is stored in either the IRLM address space or in common service area (CSA). All of these operations are uses expensive DB2 and MVS resources.

# Latching

Slide 10 of 33

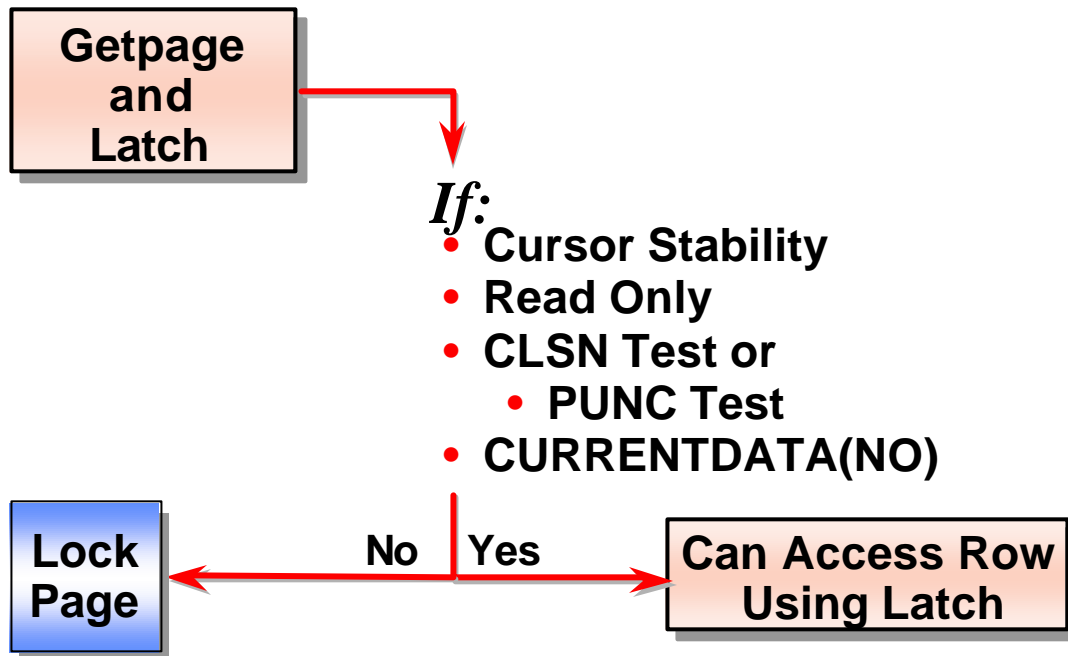
Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

---

## NOTES

# Lock Avoidance Overview



Slide 11 of 33

Unlocking the Page in DB2 – the Sequel

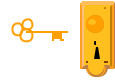
Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

To take advantage of Lock Avoidance, the SQL statement must be Read Only and the PLAN must be bound with Isolation Level Cursor Stability (CS) and CURRENTDATA(NO).

If the SQL statement and the Plan meet the above three criteria, DB2 will perform two additional checks to determine if the data row being read is committed, allowing the row to be accessed using a latch. They are the Commit Log Sequence Number (**CLSN**) and the Possibly Uncommitted (**PUNC**) checks. If either of these checks is successful, DB2 knows the data being read is physically consistent.

# When?



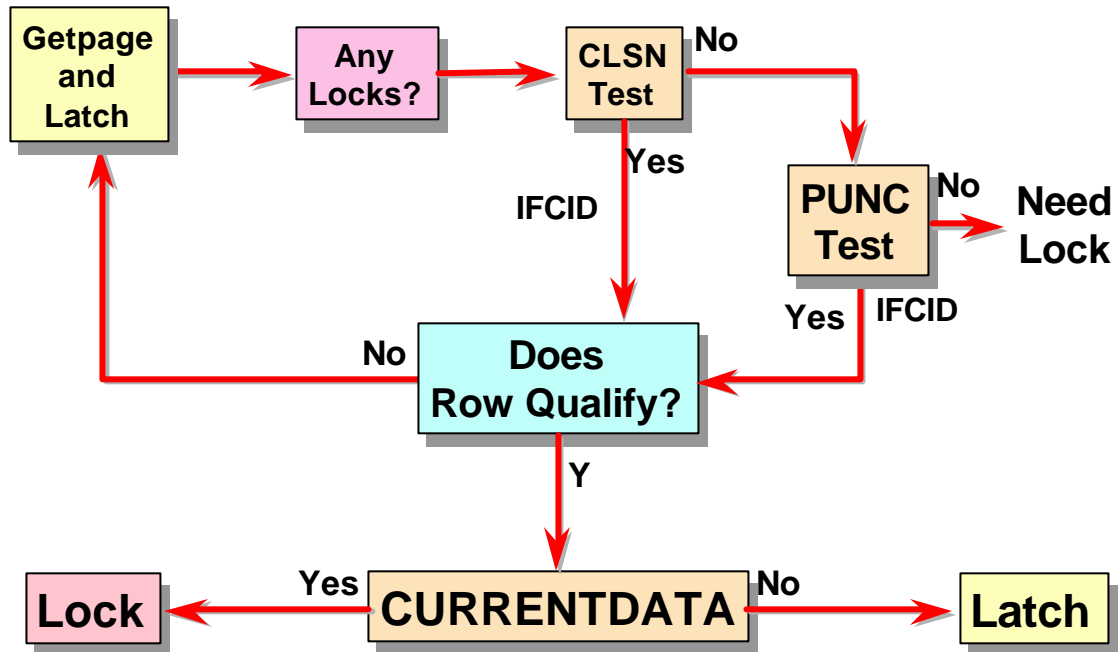
## Only If READ-ONLY

- Join
- SELECT DISTINCT
- GROUP BY
- ORDER BY
- UNION or UNION ALL
- SELECT w/Column Function
- FOR FETCH ONLY
- SELECT FROM non-updatable Catalog table

## NOTES

Uncommitted reads are only allowed for read only queries. This foils list some of the ways DB2 determines if an SQL statement is read only.

# Version 3 Lock Flow



Slide 13 of 33

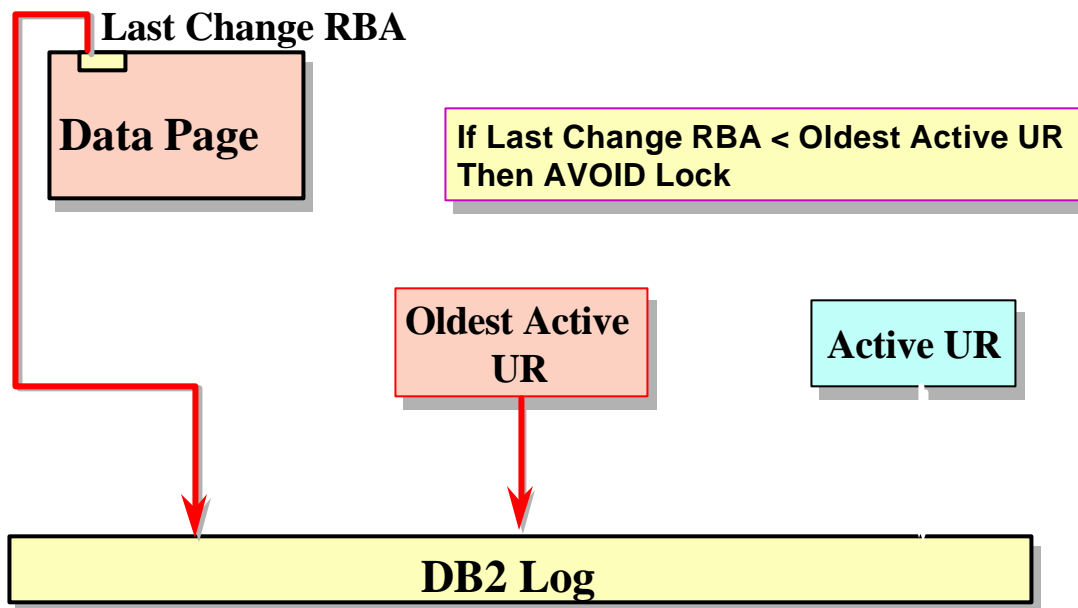
Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

This is an “artist conception” of lock avoidance’s flow. This diagram is for explaining lock avoidance only. It is not intended to show the true flow used by DB2.

# CLSN Test



Slide 14 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

The first check performed by DB2 is the CLSN check. The CLSN is sometimes called the Page LSN.

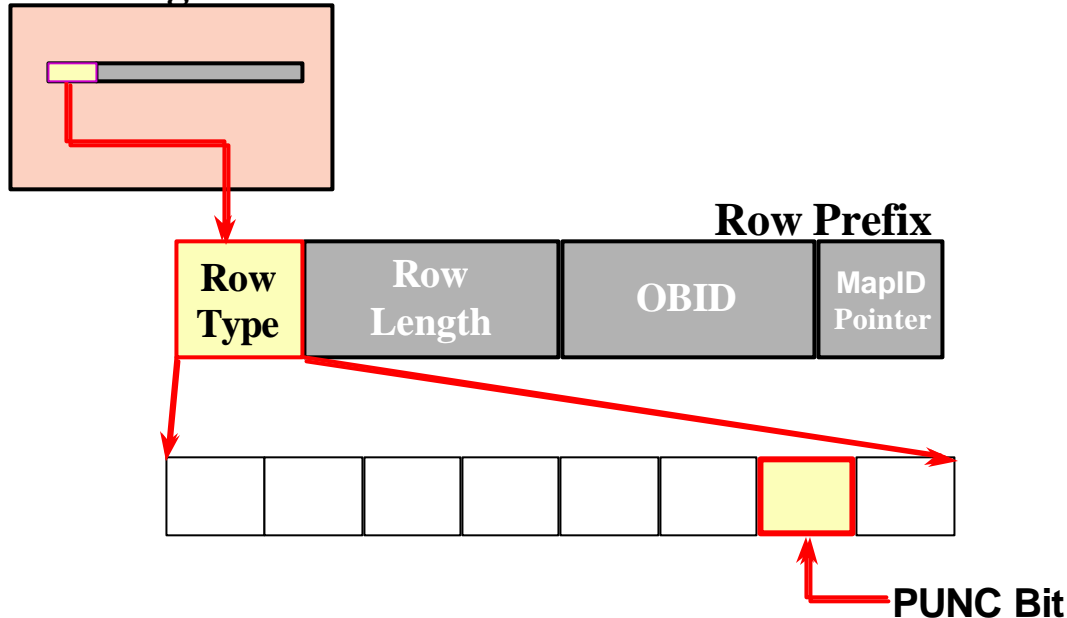
An LSN in Version 3 is the same as a relative byte address (RBA). One byte into the header portion of each data page is the log RBA of the last time this page was changed (PGLOGRBA). There is also a control block structure used by the Buffer Manager called the Page Set Piece Control Block (PB0) that contains a list of all active Units of Recovery (UR) by page set. A complete description of this control block can be found in the DB2 Diagnostics and Reference Manual (LY27-9659).

DB2 compares the last change RBA in the data page being accessed to the oldest active UR for that page set in PB0. If the last change RBA in the data page is lower than the oldest active UR, there are no active URs that can be accessing that page. If the page is not part of any active unit of recovery, it must be a committed page. Because the CLSN test is successful, DB2 does not need a lock to determine if the page is committed before reading the page. DB2 allows the application attempting the read to access the page using a page latch only.

# PUNC Test



## Data Page



Slide 15 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

If the CLSN check fails, and only if the CLSN test fails, DB2 has one additional check that can be performed to determine if the ROW that is about to be accessed is committed. This is referred to as the PUNC test.

When DB2 V3 inserts or updates a row, the sixth bit (counting from zero) in the first byte, PGSFLAGS, of the six-byte row prefix is set on (set to one). This bit is called the PUNC bit. DB2 will turn the bit off on a subsequent access of the row after the page has been committed. Thus the term possibly uncommitted. A page can be committed and still have the PUNC bit set on. If the CLSN test fails, DB2 will test the PUNC bit of the row it is about to read. If the bit is still set off (zero), DB2 knows that this row has not been changed since the last time this page was committed. Therefore, DB2 can assume this row is still committed and allow the row to be read. If the bit is set on (one), then DB2 must acquire a lock to use this page. For a table space scan or an index probe, the lock is taken immediately if the PUNC test fails.

Besides the PUNC bit in the row prefix, there is a new field defined in the page header, PGPUNCRA. This field is a counter of the number of PUNC bits in the page that are set to “on.” If this value is zero, then the entire page is committed.

# CURRENTDATA

Slide 16 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

---

## NOTES



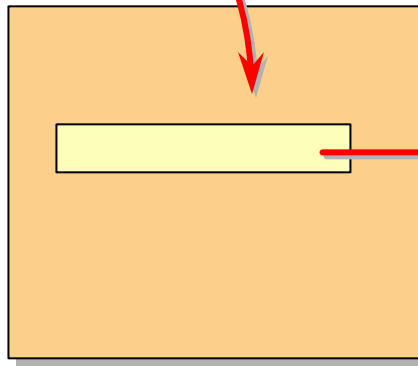
# CURRENTDATA NO



INSERT  
UPDATE  
DELETE

OK!

**DB2 Version 3 Default**



Fetch Row #1

Slide 17 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

**Persistent Data:** A data row pointed to by a cursor defined with FOR FETCH ONLY cannot be changed while the cursor continues to point to it.

If CURRENTDATA NO is specified at BIND time, DB2 will allow another process to change or delete a row after the row is read, but while the cursor is still pointing to it.

SYSIBM.SYSPLAN -

- EXPREDICATE - B - Data currency is not required (CURRENTDATA NO)
- C - Data currency is required (CURRENTDATA YES)
- N - Bound before CURRENTDATA value existed

SYSIBM.SYSPACKAGE

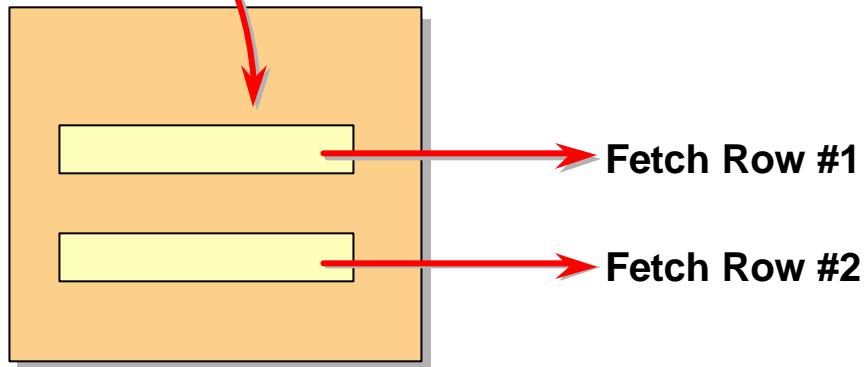
- DEFERPREP - A - CURRENTDATA YES
- B - CURRENTDATA NO
- C - CURRENT DATA YES

# CURRENTDATA YES



**INSERT  
UPDATE  
DELETE**

**DB2 Version 4 Default**



Slide 18 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

If CURRENTDATA YES is specified at BIND time, DB2 will prevent another process from changing or deleting a row while the cursor is still pointing to it.

Specify CURRENTDATA YES (or Repeatable Read) will essentially turn off Lock Avoidance.

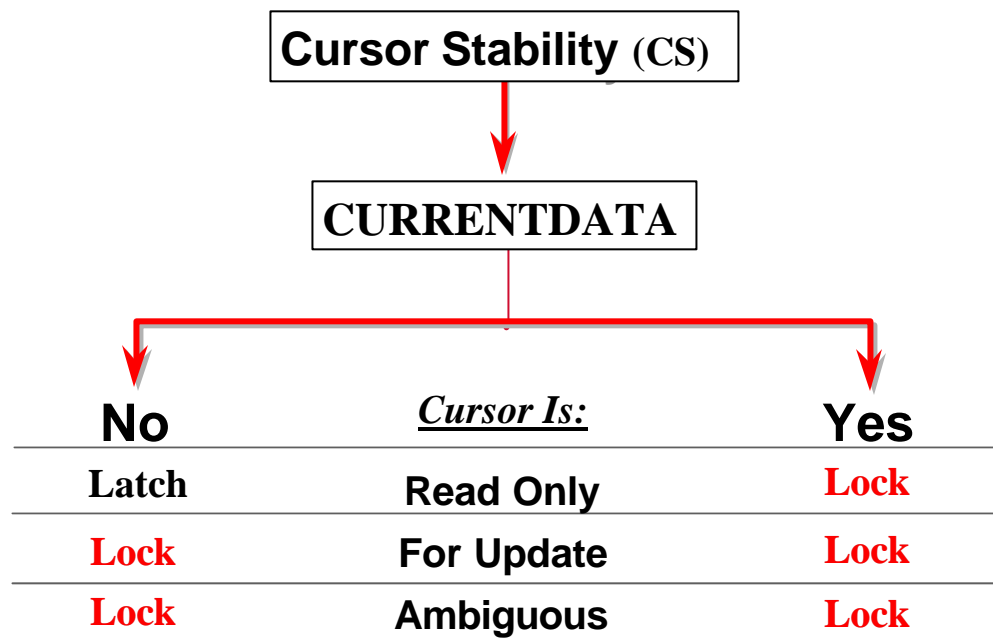
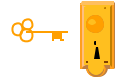
SYSIBM.SYSPLAN -

- EXPREDICATE - B - Data currency is not required (CURRENTDATA NO)
- C - Data currency is required (CURRENTDATA YES)
- N - Bound before CURRENTDATA value existed

SYSIBM.SYSPACKAGE

- DEFERPREP - A - CURRENTDATA YES
- B - CURRENTDATA NO
- C - CURRENT DATA YES

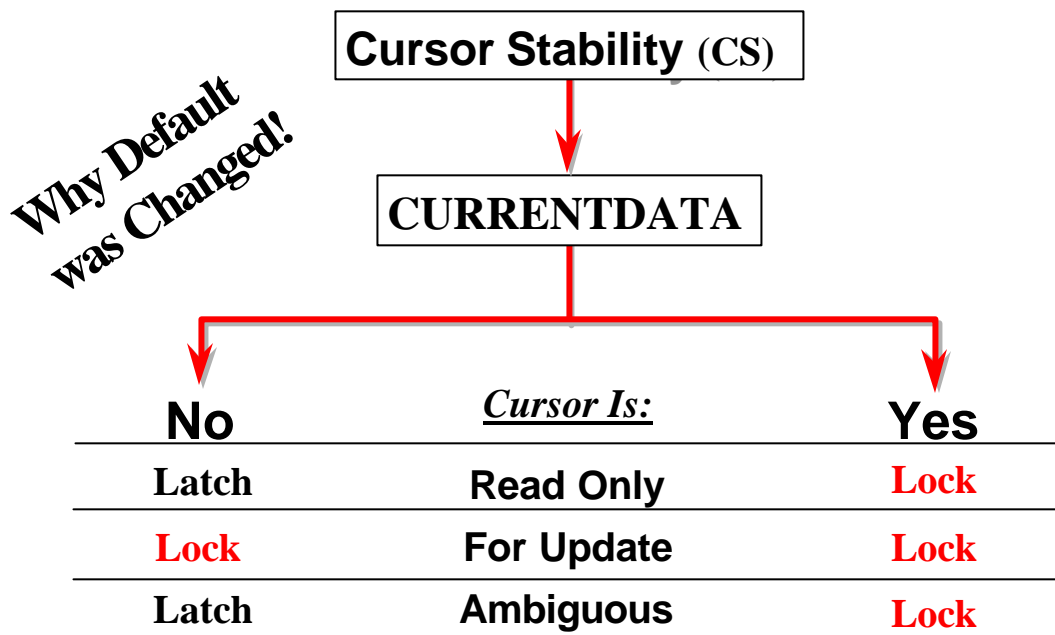
# Ambiguous Cursors



## NOTES

In DB2 Version 3, an ambiguous cursor would force a lock for both CURRENTDATA YES and NO.

# Ambiguous Cursors



Slide 20 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

In Version 4, a latch is used even with an ambiguous cursor. This is the primary reason the CURRENTDATA default was changed from NO to YES when Version 4 was released.

# Application Considerations



Define Read-Only Cursor	Can Not Update	
Issue Fetches	S-Lock	Latch
Find Row to be Modified		
Issue Stand-alone UPDATE	Promote to X-lock	Acquire X-lock
	V2	V3-and on

## NOTES

This is an example of an application using a cursor defined as read only. Because this SQL statement contains an ORDER BY clause, an UPDATE or DELETE WHERE CURRENT OF cursor clause is not allowed. Instead, an UPDATE or DELETE SQL statement independent of the cursor would need to be used. This is a fairly common practice. In Version 2, the cursor would be fetching rows under an S-Lock. When it needs to perform the update, the S-Lock is promoted to an X-Lock. Because this all happens within the same application, that application would have first shot at acquiring the X-Lock.

In Version 3 however, this would not be true. The fetch would be performed under a latch. When the application had to do the actual UPDATE, it would have to acquire an X or U-Lock. It would then be possible for another application to acquire the exclusive lock before this application.

*If time permits.....*

## Other Lock Reducing Methods

---

**NOTES**

# A Rose By Any Other Name...



**DB2's  
Choice**



**Uncommitted Read**

**Dirty Read**

**Read Thru Locks**

**GO Processing**

**Bind  
or  
SQL**

**DB2 Version 4.1**

**Requires Type 2 Index  
For Index Access path**

**Watch out for this  
It can catch you!**

Slide 23 of 33

Unlocking the Page in DB2 – the Sequel

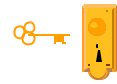
Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Read through locks, dirty reads, IMS GO processing, uncommitted read. All terms describing the ability of one application being able to read data that has NOT yet been committed (or rolled back) by another application.

# Do We Have a Dirty Read Now?

---



**Lock Avoidance is NOT a  
dirty read!**

***But.....***

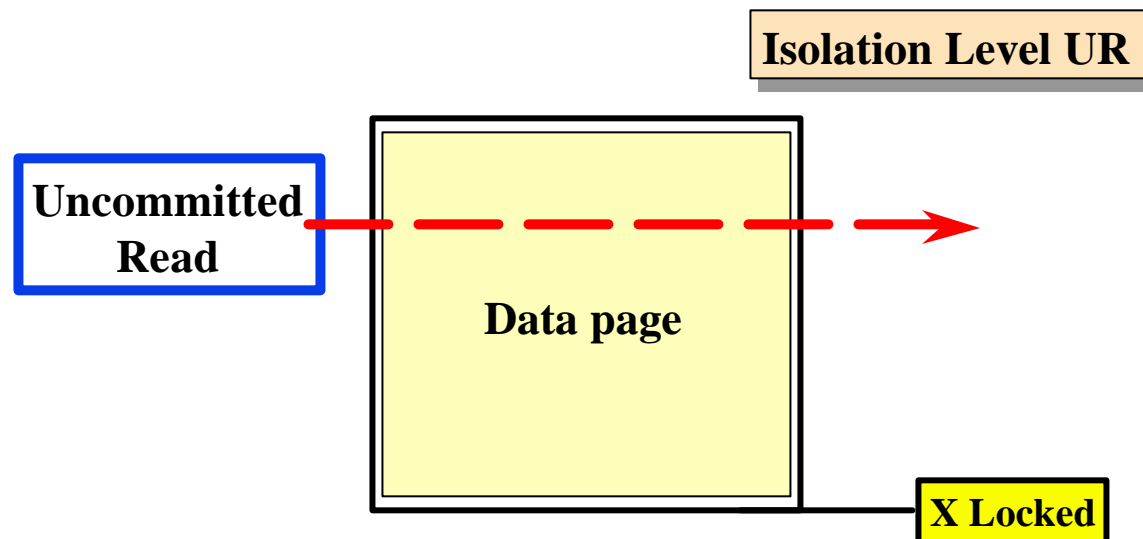
**What about Version 4?**

**NOTES**

---



# Uncommitted Read



Slide 25 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

UR or **Uncommitted Read** is a new Isolation level on the BIND PLAN and BIND PACKAGE command.

It allows a read to complete even though there may still be an exclusive lock on the page.

# Repeatable Read



INSERT  
UPDATE  
DELETE

Isolation Level RR

Locks held on each page allowing re-read of same data

Commit

Slide 26 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Repeatable read has been around as long as there has been a DB2. When this isolation level is chosen, all locks (both X, S, and U) are held until commit. If you are reading pages using RR, no updates are permitted until a commit releases the share locks.

# Read Stability



**UPDATE  
DELETE**

**Isolation Level RS**

Locks held on each page allowing re-read of same data

**INSERTs Allowed**

**Commit**

## NOTES

Read Stability is similar to Repeatable Read. The difference is that RS allow INSERTs to complete and only prevents UPDATES and DELETES.

# BIND Plan / Package



**BIND PLAN/PACKAGE - New ISOLATION Levels**

**SYSPLAN/SYSPACKAGE Catalog Tables have  
New Value for Isolation - "U" - For UR  
"T" - For RS**

**SYSSTMT/SYSPACKSTMT now Has an  
ISOLATION Column for Every SQL Statement**

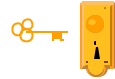
**SQL WITH Clause:  
SQL Statement → Package → Plan**

## NOTES

SYSIBM.SYSSTMT and SYSIBM.SYSPACKSTMT has a new column for the ISOLATION level by SQL statement. SYSIBM.SYSPLAN and SYSIBM.SYSPACKAGE has a new value (U) for the ISOLATION level column.

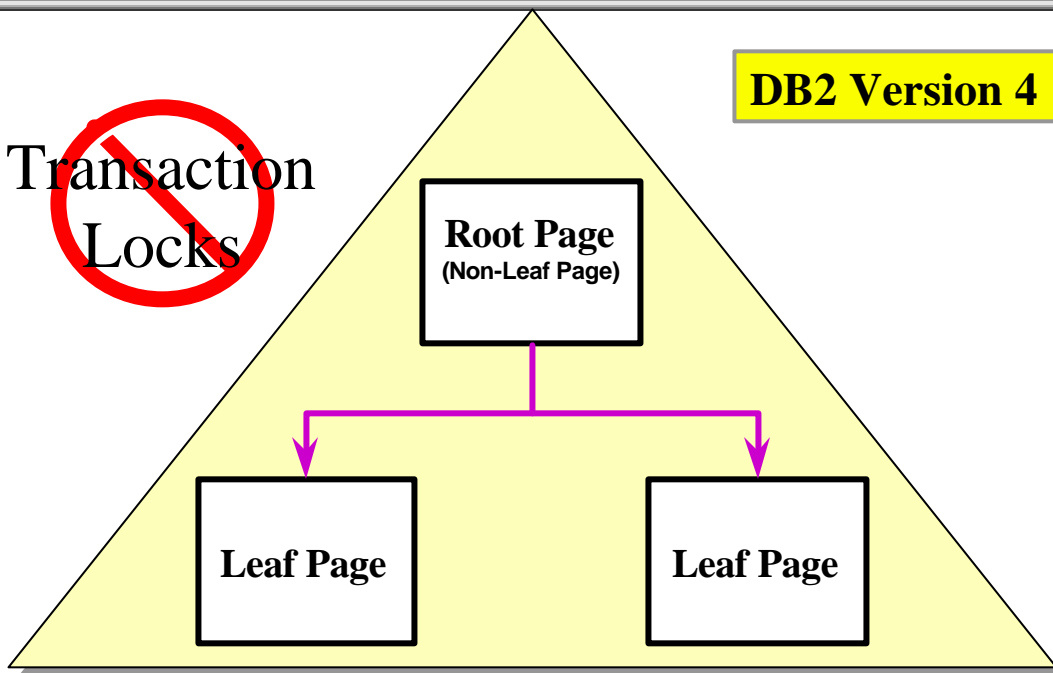
DB2 will first check the isolation level at the SQL level. An SQL statement has NO default isolation level. If an isolation level is NOT specified at the statement level, DB2 will then check the PACKAGE. Again, a PACKAGE has NO default. If an isolation level is not specified at the PACKAGE level, DB2 will check the PLAN. The PLAN does default to Repeatable Read.

# Type 2 Indexes



**DB2 Version 4**

**Transaction  
Locks**



Slide 29 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Two significant design enhancements are the elimination of locks and subpages. Both changes will effect both concurrency and availability. By no longer locking at any level of the index structure, the chances for deadlocks and time-outs, caused by the index, are greatly decreased. DB2 will use a technique called lock avoidance introduced in DB2 Version 3. This technique uses latches for index access, thus avoiding the expense and problems associated with locks.

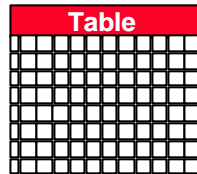
Type 2 index no longer has the concept of subpages. SUBPAGE parameter has been removed. This will end all of the discussion of whether SUBPAGE 1 or SUBPAGE > 1 should be coded. This will also end all of the excessive locking the type 1 index manager had to go through when a SUBPAGE > 1 value was specified.

# SQL Temporary Tables



## CREATE GLOBAL TEMPORARY TABLE ...

No Logging  
No Locking  
No Recovery



SQL OPEN, SELECT, INSERT, DELETE



Logical Workfiles

Slide 30 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

Requirements have been around for some time for tables that exist only for the life of the task using them, non-recoverable table spaces, and table spaces that do not require logging when changes are made to them. DB2 for OS/390 Version 5.1 satisfies these requirements with the introduction of Temporary Tables, a feature of the SQL92 full level standard. This new form of a table, although defined in the DB2 Catalog, does not have persistent rows of data. Neither logging nor **locking** is performed on the tables' behalf. Because there is no logging, the tables' data is not recoverable. A temporary table can be referenced by any program in the same DB2 application process at the same DB2 server. External table spaces or databases are not associated with a temporary table. In addition, there are new COMMIT and ROLLBACK semantics introduced in DB2 V5.1 to support temporary tables.

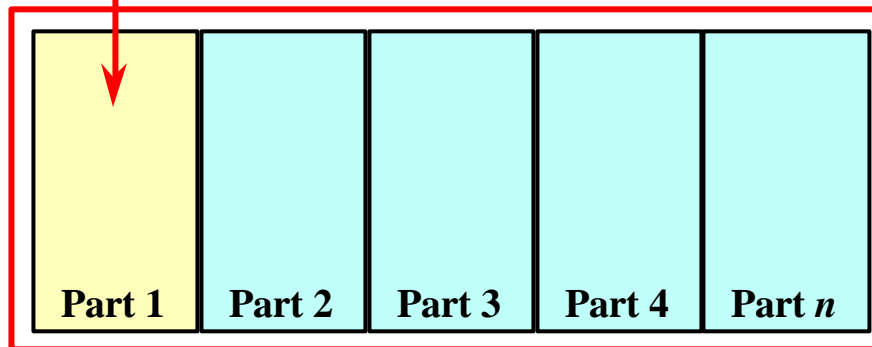
Temporary tables are implemented using DB2 logical workfiles. Because these logical workfiles are in the same physical workfile space used for sort activity, there are performance considerations when accessing temporary tables. Creation of a temporary table could be slowed because of lack of workfile space due to large or long running sorts. I/O contention could also be experienced for the same reasons. Consider increasing the size and number of physical workfiles if temporary tables are going to be used. Finally, scans of large temporary tables could be slower than using an index scan for a traditional table because indexes cannot be defined on temporary tables.

# Partition Locking in the Past



SELECT col1 FROM XYZ WHERE...

IS or IX lock on Entire  
Partitioned Table Space



Partition Table Space XYZ

**DB2 Version 4.1**

**Locked**

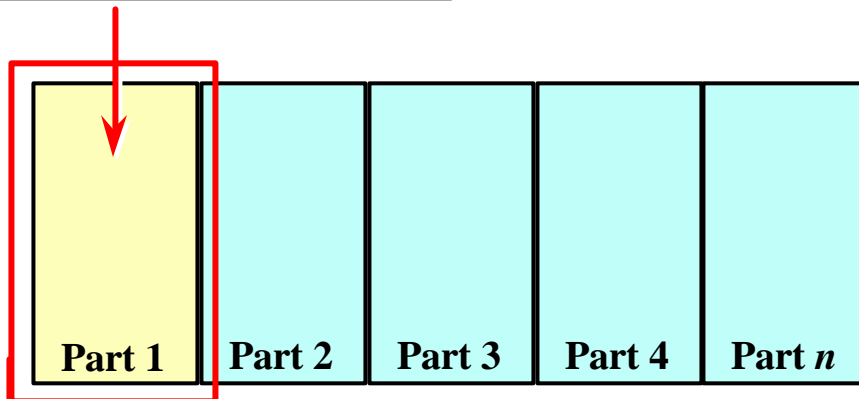
## NOTES

In DB2 Version 4.1 and prior releases, if any one partition was accessed, all partitions in the partitioned table space were locked.

# Selective Partition Locking



```
SELECT col1 FROM XYZ WHERE...
```



Partition Table Space XYZ

**Locked  
Only**

**DB2 Version 5.1**

Slide 32 of 33

Unlocking the Page in DB2 – the Sequel

Copyright © 2001 BMC Software, Inc.  
All rights reserved.

## NOTES

With DB2 Version 5.1, only the partition being accessed is locked. This feature does have restrictions. If the access path uses an index, the index must be a Type 2 index. The plan must be bound with ACQUIRE(USE). It is not supported for LOCKSIZE TABLESPACE or for the SQL statement LOCK TABLE IN EXCLUSIVE MODE. It may not be supported for certain SQL DDL operations. This feature was implemented in support of Data Sharing and Large Table Spaces.

Syntax example:

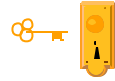
```
CREATE TABLESPACE  
  NUMPARTS nn  
  LOCKPART YES | NO
```

```
ALTER TABLESPACE  
  LOCKPART YES | NO
```

### What else?

- Lock Escalation
- STOP DB SPACE PART
- Data Sharing
- Fallback
- Coexistence
- New SYSTABLESPACE Column  
LOCKPART



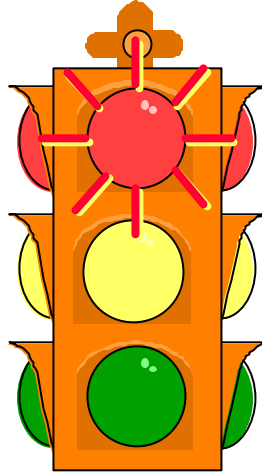


**Thank You  
for Attending!  
Willie**

## **NOTES**

# The End

---



## NOTES

---

# Contact Information

---



**Willie Favero**  
**Principal Consultant**  
**[willie\\_favero@bmc.com](mailto:willie_favero@bmc.com)**

**BMC Software, Inc.**  
**<http://www.bmc.com>**  
**2101 CityWest Blvd.**  
**Houston, TX 77042**  
**USA**

**713 918 5047 (Direct)**  
**800 841 2031 (Toll Free)**  
**713 918 1837 (Fax)**

## NOTES

---