Seminar 4 - 921

17.04.2025, 16-18

### Multiversioning

Monitoring Locks

- SQL- Server Profiler

- sp\_lock

- sys.dm\_tran\_locks

- sys.dm\_tran\_active\_transactions

Resource types:

- RID - row in a heap (physical pointer)

- key - row in an index (logical pointer)

- page

- HorBT - heap or BTree

- Objects: table, view, function,

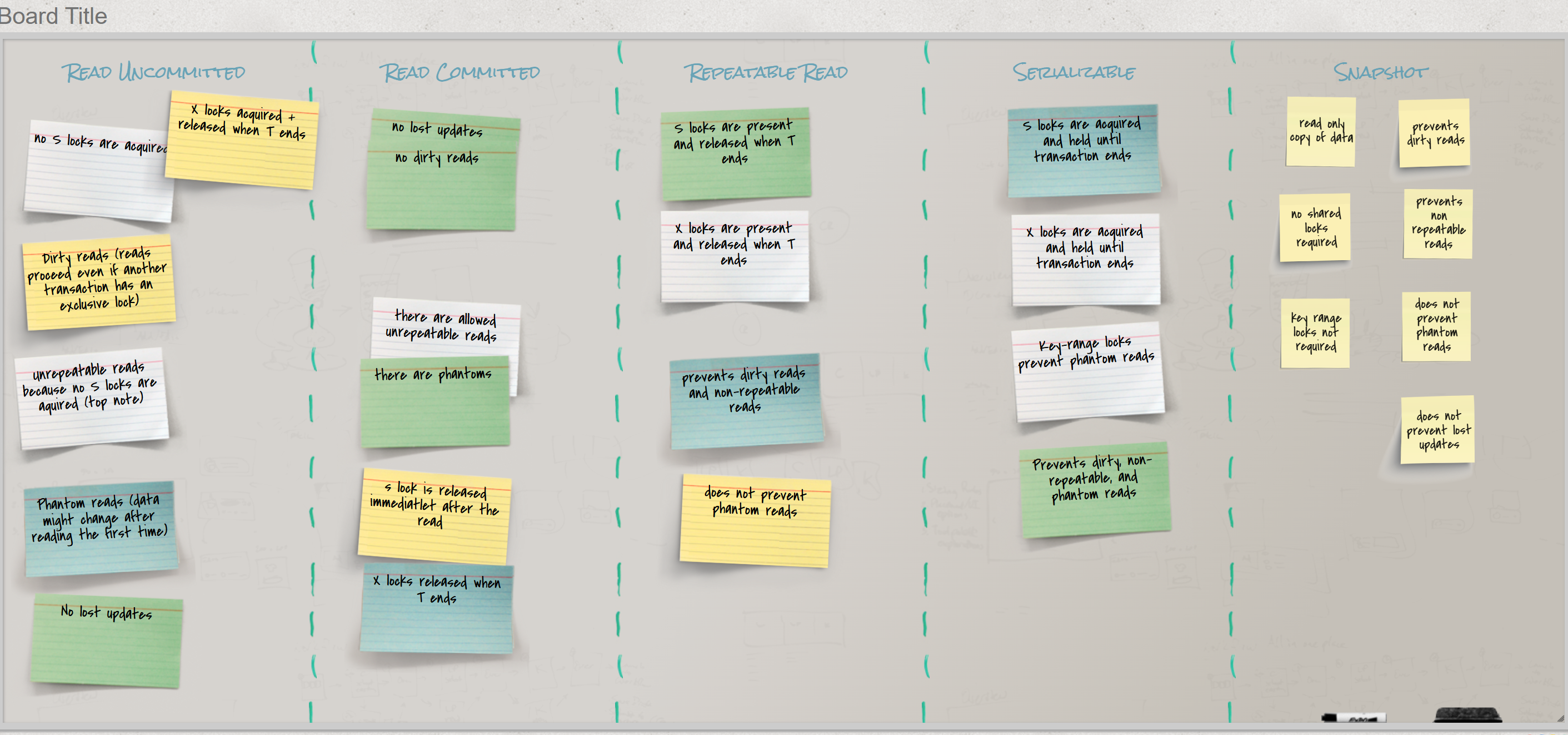
- File

- database

- metadata

- application

http://scrumblr.ca/921\_s4



Multiversioning

- update: copy the O object and modify the copy (create a new version)

- read: being able to read any version of the object O

- no overwritting

Row Level Versioning (RLV)

- useful when reading committed data but not necessarily the latest data

- two transaction isolation levels: READ COMMITTED SNAPSHOT and FULL SNAPSHOT

READ COMMITTED SNAPSHOT

- reading data committed before the current command

- update conflicts - automatically handled, transactions are automatically retried

FULL SNAPSHOT

- reading data committed before the current transaction

- update conflicts - can be replicated

- older versions of the rows are stored in tempdb in a linked list

- each element from the linked list has the data and the transaction sequence number (XSN)

Advantages:

+ triggers and indexes

+ increased concurrency level

+ prevent deadlocks (some of them)

Disadvantages:

- read operations - slower (need to iterate over the linked list)

- more storage + management costs

- update operations - slower (need to do a copy and then modify)

Triggers

- special tables: inserted and deleted

- before RLV: both tables created by parsing the transaction log (less efficient)

- after RLV: tables created with RLV (more efficient)

Indexes:

- before RLV:

Cl index: data was blocked

Ncl index: index is not available

- after RLV:

Create or rebuild the index online

CREATE INDEX …. WITH ONLINE = ON

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Concurrency issue / Transaction isolation level | Read uncommitted | Read Committed | Repeatable Read | Serializable | Read committed Snapshot | Full Snapshot |
| Dirty Reads | yes | no | no | no | no | no |
| Nonrepeatable Reads | yes | yes | no | no | yes | no |
| Phantom Reads | yes | yes | yes | no | yes | no |
| Update Conflict | no | no | no | no | no | **yes** |
| Concurrency model | pessimistic | pessimistic | pessimistic | pessimistic | optimistic | optimistic |

Pessimistic - expected to have frequent update collisions

Optimistic - does not expect to have update collisions

Query Governor

SET QUERY\_GOVERNOR\_COST\_LIMIT value (in seconds)

By default value = 0(all queries are permitted)

DBCC LOG (DBname, 0 -4 --> the details )

DBCC LOGINFO

DBCC SQLPERF (LOGSPACE)

PIVOT/ UNPIVOT

The OUTPUT Clause

The MERGE statement

QUIZ: Under which transaction isolation level are we able to replicate the update conflict?