Pessimistic versus Optimistic Concurrency (Locking versus Feedback)

Asked 15 years, 9 months ago Modified 15 years, 9 months ago

Viewed 3k times

I'm building an application with the following criteria:

- work items: items that need to be manually worked via the web by users (short one page form)
- Multiple users working 'work items'
- Each user has a queue of 'work items'
- There is a search that allows users to view 'work items' and assign 'work items' to their queues
- Users can take 'work items' out of other people's queues by assigning them to themselves

Note: 'work items' are worked only once. This is not a wiki page, it's more of a matching exercise that should only be performed once by one user. Once the 'work item' is worked, it is gone from the system (aside from some auditing/reporting), Somewhat like a bug tracking system

Which option do you believe is better? Can you cite any mainstream applications that support your opinion?

Option 1:

- When user A goes to view or work a 'work item', the 'work item' becomes locked.
- When other users go to the 'work item' after User A opens the 'work item', they will only be able to see the 'work item'. They can not write.
- The lock expires after n minutes at which point another user can lock the 'work item'.

Option 2:

- Any user can pull up a 'work item' without locking it.
- If User A works the 'work item' by submitting the form and User B works the same 'work item', then user A's work will take affect in the database, and User B will be informed that their changes did not take affect because another user has modified the 'work item'.

I personally like option 2. Thoughts please?

concurrency **locking** **editing** **multi-user**

Share

Improve this question

Follow

edited Mar 18, 2009 at 20:39

asked Mar 18, 2009 at 18:13

ckarbass

3,681 ● 7 ● 36 ● 43

3 Answers

Sorted by: Highest score (default)

7

Sounds like you're talking about pessimistic verses optimistic concurrency control.

Both are widely used, personally I find optimistic concurrency easier to deal with, but it will depend upon your own requirements and usage. If edits (and potential conflicts) are common then pessimistic concurrency control may be appropriate, if not, then optimistic concurrency will be faster and simpler to work with.

Let me know if you want to see code examples using RowVersion datatype in SQL Server (that's what I'm currently using), it's pretty simple though:

- All tables include RowVersion column
- All SELECT queries include this column (for data which can be modified)
- All UPDATE or DELETE queries include a WHERE RowVersion = @RowVersion. This is the optimistic part, if 0 rows returned then someone else has touched the row, no update takes place, so tell user about it. NOTE: If row was updated then new value for RowVersion should also be returned. This also applies to INSERT queries, much like you would return the Id of an identity column after an insert.

Share Improve this answer edited Mar 18, 2009 at 18:32

Follow

answered Mar 18, 2009 at 18:23

si618

16.8k 12 69 84

thanks, do you know what most bug tracking systems use? pessimistic or optimistic, i would assume the latter – ckarbass Mar 18, 2009 at 18:40

Sorry, I don't know, but as a guess since most bug tracking systems are web based, and the web is (mostly:) stateless, then optimistic locking is probably used. – si618 Mar 18, 2009 at 18:53

0

> Not sure how to describe the form in simple terms, but its not a community page, its a one time thing. Hypothetically, let's say the user had to match the name John DOEE to one of the following John Doe Jon Do Once it's worked, the edit is complete. In our case, we would not need to mere

Considering that comment, I would go with option 1. Considering that these changes are one time changes, there is no benefit to allowing multiple people to work on the same change. You're only wasting the 2nd person's time.

Share Improve this answer

answered Mar 18, 2009 at 18:26

Follow

Brian
25.8k ● 18 ● 86 ● 178

not sure if this is clear in the question, but what if I'm a manager and I pull up a work item (formally called edit) and now its locked, then you go to work your queue and you can't work that item because your manager has it open in a browser – ckarbass Mar 18, 2009 at 18:37

I differentiate between pulling the item up for editting and pulling it up for viewing. It's simple enough to have an edit button which uses JSON when you click EDIT to decide if it should make the fields be editable or not. – Brian Mar 18, 2009 at 18:53

That button can go a step further and, if they aren't editable, show the changes via black magic (without reloading the page). Though this sort of thing adds to the complexity. – Brian Mar 18, 2009 at 18:55

-1

I personally would go with option 2 - plus, if there applicable (for example those edits are on a longer text) make it the responsibility of User B to merge the edits. Give B a tool to do so.

Share Improve this answer

Follow

answered Mar 18, 2009 at 18:17

Tobias Hertkorn
2,750 ● 1 ● 21 ● 28

updated question, merging is not needed in this scenario, although valid feedback – ckarbass Mar 18, 2009 at 18:22