How to track data changes in a database table

Asked 16 years, 3 months ago Modified 7 years, 7 months ago

Viewed 103k times

▲

50

▼

What is the best way to track changes in a database table?

Imagine you got an application in which users (in the context of the application not DB users) are able to change data which are store in some database table. What's the best way to track a history of all changes, so that you can show which user at what time change which data how?

database

Share

Improve this question

Follow

edited May 3, 2013 at 10:48

drf

8,669 ● 33 ● 50

asked Sep 1, 2008 at 21:10

user3711

503 ● 1 ● 4 ● 6

2 Some good discussion here: discuss.joelonsoftware.com/default.asp?design.4.483891.10

– Simon Woodside Mar 14, 2011 at 4:32

1 See also Best design for a changelog / auditing database table? – user Sep 14, 2017 at 7:51

8 Answers

Sorted by: Highest score (default)

▲

13

▼

✓

In general, if your application is structured into layers, have the data access tier call a stored procedure on your database server to write a log of the database changes.

In languages that support such a thing aspect-oriented programming can be a good technique to use for this kind of application. Auditing database table changes is the kind of operation that you'll typically want to log for all operations, so AOP can work very nicely.

Bear in mind that logging database changes will create lots of data and will slow the system down. It may be sensible to use a message-queue solution and a separate database to perform the audit log, depending on the size of the application.

It's also perfectly feasible to use stored procedures to handle this, although there may be a bit of work involved passing user credentials through to the database itself.

Share Improve this answer Follow

answered Sep 1, 2008 at 21:19

Jeremy McGee
25.2k ● 10 ● 64 ● 96

2 It seems like if you want to allow the user to view their own history or other people's history a log file wouldn't be very accessible to them. – Simon Woodside Mar 14, 2011 at 4:15

9

You've got a few issues here that don't relate well to each other.

At the basic database level you can track changes by having a separate table that gets an entry added to it via triggers on INSERT/UPDATE/DELETE statements. Thats the general way of tracking changes to a database table.

The other thing you want is to know which *user* made the change. Generally your triggers wouldn't know this. I'm assuming that if you want to know which user changed a piece of data then its possible that multiple users could change the same data.

There is no right way to do this, you'll probably want to have a separate table that your application code will insert a record into whenever a user updates some data in the other table, including user, timestamp and id of the changed record.

Make sure to use a transaction so you don't end up with cases where update gets done without the insert, or if you do the opposite order you don't end up with insert without the update.

Share Improve this answer answered Sep 1, 2008 at 21:17

Follow

thelsdj
9,124 ● 10 ● 46 ● 58

6

One method I've seen quite often is to have audit tables. Then you can show just what's changed, what's changed and what it changed from, or whatever you heart desires :) Then you could write up a trigger to do the actual logging. Not too painful if done properly...

No matter how you do it, though, it kind of depends on how your users connect to the database. Are they using a single application user via a security context within the app, are they connecting using their own accounts on the domain, or does the app just have everyone connecting with a generic sql-account?

If you aren't able to get the user info from the database connection, it's a little more of a pain. And then you might look at doing the logging within the app, so if you have a process called "CreateOrder" or whatever, you can log to the Order_Audit table or whatever.

Doing it all within the app opens yourself up a little more to changes made from outside of the app, but if you have multiple apps all using the same data and you just wanted to see what changes were made by yours, maybe that's what you wanted... <*shrug*>

Good luck to you, though!

--Kevin

Share Improve this answer

Follow

answered Sep 1, 2008 at 21:24

Kevin Fairchild
10.9k ● 6 ● 34 ● 52

4

In researching this same question, I found a discussion here very useful. It suggests having a parallel table set for tracking changes, where each change-tracking table has the same columns as what it's tracking, plus columns for who changed it, when, and if it's been deleted. (It should be possible to generate the schema for this more-or-less automatically by using a regexed-up version of your pre-existing scripts.)

Share Improve this answer

Follow

answered Dec 8, 2012 at 23:15

TomNysetvold
1,075 ● 7 ● 9

1 As of 2021-12-31 link appears to lead to an off-topic blog post. – El- Dec 31, 2021 at 18:25

4

Suppose I have a Person Table with 10 columns which include PersonSid and UpdateDate. Now, I want to keep track of any updates in Person Table. Here is the simple technique I used:

1. Create a person_log table

   create table person_log(date datetime2, sid int);

2. Create a trigger on Person table that will insert a row into person_log table whenever Person table gets

> updated:
>
> create trigger tr on dbo.Person
> for update
> as insert into person_log(date, sid) select
> updatedDTTM, PersonSID from inserted

After any updates, query person_log table and you will be able to see personSid that got updated. Same you can do for Insert, delete.

Above example is for SQL, let me know in case of any queries or use this link : https://web.archive.org/web/20211020134839/https://www.4guysfromrolla.com/webtech/042507-1.shtml

Share Improve this answer

Follow

answered May 25, 2017 at 14:15

S'chn T'gai Spock

1,366 ● 18 ● 17

1

A trace log in a separate table (with an ID column, possibly with timestamps)?

Are you going to want to undo the changes as well - perhaps pre-create the undo statement (a DELETE for every INSERT, an (un-) UPDATE for every normal UPDATE) and save that in the trace?

Share Improve this answer

Follow

answered Sep 1, 2008 at 21:19

Unsliced

10.5k ● 8 ● 52 ● 81

1

Let's try with this open source component:

https://tabledependency.codeplex.com/

TableDependency is a generic C# component used to receive notifications when the content of a specified database table change.

Share Improve this answer Follow

answered Jul 19, 2016 at 19:27

Christian Del Bianco
1,043 ● 13 ● 14

I strongly recommend above mentioned excellent tabledependency library. – Kay Lee Apr 24, 2022 at 12:36

0

If all changes from php. You may use class to log evry INSERT/UPDATE/DELETE before query. It will be save ***action***, ***table***, ***column***, ***newValue***, ***oldValue***, ***date***, ***system***(if need), ***ip***, ***UserAgent***, ***clumnReference***, ***operatorReference***, ***valueReference***. All tables/columns/actions that need to log are configurable.

Share Improve this answer Follow

answered Feb 5, 2017 at 8:57

moledet
939 ● 11 ● 19