Redmine - Feature #35073

Escape values in LIKE statements to prevent injection of placeholders (_ or %)

2021-04-12 08:50 - Jens Krämer

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Marius BĂLTEANU	**% Done:**	0%
Category:	Database	**Estimated time:**	0.00 hour
Target version:	5.0.0		
Resolution:			

Description

While not technically a security risk, LIKE queries with lots of placeholders can result in high database load, very slow query execution and therefore are a possible vector for denial of service attacks. Further, escaping the wildcard characters in actual query values now allows to actually search for values containing these characters.

The attached patches, which have been extracted from Planio are:

- The first patch removes '%' signs from query strings used in the search test case. These did not matter until now since they just resulted in queries like field LIKE '%%value%%', but now would be looking for a value with literal percent signs. Tests pass before and after that change.
- Patches 2-4 add sanitize_sql_like calls to the various places where we currrently generate sql LIKE statements. Corresponding tests are included.
- The last patch is just a cosmetic change that replaces a .send with a direct call since the called method is now public.

Related issues:

Related to Redmine - Feature #13347: Filtering by issue subject with wildcard	**New**
Related to Redmine - Defect #19786: '%' and '_' are treated as SQL wildcards ...	**Closed**
Blocks Redmine - Feature #35764: Multiple search terms in the "contains" oper...	**Closed**

Associated revisions

Revision 21229 - 2021-10-03 21:42 - Marius BĂLTEANU

Removes '%' signs from test strings in search test (#35073).

Patch by Jens Krämer.

Revision 21230 - 2021-10-03 21:43 - Marius BĂLTEANU

Use sanitize_sql_like on search tokens (#35073).

Patch by Jens Krämer.

Revision 21231 - 2021-10-03 21:44 - Marius BĂLTEANU

Use sanitize_sql_like in like scopes (#35073).

Patch Jens Krämer.

Revision 21232 - 2021-10-03 21:45 - Marius BĂLTEANU

Use sanitize_sql_like in Query#sql_contains (#35073).

Patch by Jens Krämer.

Revision 21233 - 2021-10-03 21:46 - Marius BĂLTEANU

Replace send with direct call because sanitize_sql_for_conditions is now public (#35073).

Patch by Jens Krämer.

Revision 21234 - 2021-10-03 21:56 - Marius BĂLTEANU

Fix rubocop offense (#35073).

Revision 21240 - 2021-10-09 08:35 - Marius BĂLTEANU

Explicitly specify escape character using an ESCAPE on SQLite (#35073).

Patch by Go MAEDA.

History

#1 - 2021-04-12 08:55 - Go MAEDA

- Related to Feature #13347: Filtering by issue subject with wildcard added

#2 - 2021-04-12 08:55 - Go MAEDA

- Related to Defect #19786: '%' and '_' are treated as SQL wildcards in issue filter added

#3 - 2021-05-12 06:20 - Go MAEDA

I actually find the current behavior useful. For example, when I search for photos using the "File" filter, I use the query string "dsc%.jpg".

If Redmine prohibits the inclusion of "_" or "%" in the query string, then I want some alternative.

#4 - 2021-07-07 08:46 - Jens Krämer

- File 0001-tokenize-query-strings-for-Issue.like-and-Query-sql_.patch added

In general I believe these SQL wild cards are nowhere documented in the Redmine context (please correct me if I'm wrong), and therefore more or less a 'hidden feature' for power users who are familiar with SQL. Given the potential of abuse (and the current inability to search for terms that actually contain these wildcards), I still believe it is the right way to not expose such a low level database feature.

However I get your point and in fact I am right now revisiting this because some Planio power users do miss the SQL wild cards as well :)

The alternative I am experimenting with right now is breaking up the user's query string into tokens like the global search does, and use these to build a query with multiple LIKE clauses that are combined with AND. I'm attaching a preliminary patch that implements this for the Issue.like scope (used by the autocompleter) and the Query#sql_contains method (which should cover all query filters). Do you think that's a viable approach?

#5 - 2021-07-23 09:26 - Go MAEDA

Jens Krämer wrote:

> In general I believe these SQL wild cards are nowhere documented in the Redmine context (please correct me if I'm wrong), and therefore more or less a 'hidden feature' for power users who are familiar with SQL. Given the potential of abuse (and the current inability to search for terms that actually contain these wildcards), I still believe it is the right way to not expose such a low level database feature.

I agree with you. Actually, I opened #13347 that reports about it 6 years ago.

> The alternative I am experimenting with right now is breaking up the user's query string into tokens like the global search does, and use these to build a query with multiple LIKE clauses that are combined with AND. I'm attaching a preliminary patch that implements this for the Issue.like scope (used by the autocompleter) and the Query#sql_contains method (which should cover all query filters). Do you think that's a viable approach?

I tried out the patch and found it is really useful. I think it is a great improvement to be able to do OR searches with multiple keywords, which is currently not possible with the current filters. Although it is a breaking change, it is worth more than that.

#6 - 2021-07-29 06:59 - Go MAEDA

- Target version set to Candidate for next major release

#7 - 2021-08-15 07:23 - Go MAEDA

- Blocks Feature #35764: Multiple search terms in the "contains" operator of text filters added

#8 - 2021-08-15 10:11 - Marius BĂLTEANU

- Target version changed from Candidate for next major release to 5.0.0

#9 - 2021-08-15 13:44 - Go MAEDA

Jens Krämer wrote:

> The alternative I am experimenting with right now is breaking up the user's query string into tokens like the global search does, and use these to build a query with multiple LIKE clauses that are combined with AND. I'm attaching a preliminary patch that implements this for the Issue.like scope (used by the autocompleter) and the Query#sql_contains method (which should cover all query filters). Do you think that's a viable approach?

I have extracted attachment:0001-tokenize-query-strings-for-Issue.like-and-Query-sql_.patch to a separete issue #35764, because it adds a new feature but others fix undesirable behavior as described in the subject.

#10 - 2021-10-03 21:50 - Marius BĂLTEANU

- Status changed from New to Closed

- Assignee set to Marius BĂLTEANU

All five patches committed, thanks.

#11 - 2021-10-05 09:49 - Go MAEDA

- Status changed from Closed to Reopened

After applying the patch, some tests fail when the backend database is SQLite. I will post a fix soon.

#12 - 2021-10-06 16:56 - Go MAEDA

- File 35073-sqlite-fix.patch added

Go MAEDA wrote:

> After applying the patch, some tests fail when the backend database is SQLite. I will post a fix soon.

I am attaching a patch that fixes the issue. There is no default escape character in SQLite, so it must be specified explicitly using an ESCAPE clause.

#13 - 2021-10-06 17:06 - Marius BĂLTEANU

Go MAEDA wrote:

> Go MAEDA wrote:
>
> > After applying the patch, some tests fail when the backend database is SQLite. I will post a fix soon.
>
> I am attaching a patch that fixes the issue. There is no default escape character in SQLite, so it must be specified explicitly using an ESCAPE clause.

I don't have a local environment with SQLite to test it. Is it ok to commit the patch directly?

#14 - 2021-10-07 16:20 - Go MAEDA

- File 35073-sqlite-fix.patch added

Updated the fix for SQLite. The previous patch fails when the database is MS SQL Server.

Marius BALTEANU wrote:

> I don't have a local environment with SQLite to test it. Is it ok to commit the patch directly?

Yes, it can be committed as is. The new patch was tested with all supported databases (PostgreSQL, MySQL, SQLite, and MS SQL Server).

#15 - 2021-10-09 13:12 - Marius BĂLTEANU

- Subject changed from escape values in LIKE statements to prevent injection of placeholders (_ or %) to Escape values in LIKE statements to prevent injection of placeholders (_ or %)

- Status changed from Reopened to Closed

All tests passed now.

#16 - 2022-03-27 05:00 - Go MAEDA

- Tracker changed from Patch to Feature

Files

0005-sanitize_sql_for_conditions-is-now-public.patch	1.02 KB	2021-04-12	Jens Krämer

0004-use-sanitize_sql_like-in-Query-sql_contains.patch	1.75 KB	2021-04-12	Jens Krämer
0003-use-sanitize_sql_like-in-like-scopes.patch	6.13 KB	2021-04-12	Jens Krämer
0002-use-sanitize_sql_like-on-search-tokens.patch	2.34 KB	2021-04-12	Jens Krämer
0001-removes-signs-from-test-strings-in-search-test.patch	1.16 KB	2021-04-12	Jens Krämer
0001-tokenize-query-strings-for-Issue.like-and-Query-sql_.patch	3.93 KB	2021-07-07	Jens Krämer
35073-sqlite-fix.patch	2.29 KB	2021-10-06	Go MAEDA
35073-sqlite-fix.patch	2.32 KB	2021-10-07	Go MAEDA