

Redmine - Defect #23586

Create index on mysql exceed limits

2016-08-15 09:53 - Chen Lei

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Database	Estimated time:	0.00 hour
Target version:		Affected version:	3.2.3
Resolution:	Fixed		
Description			
When I create database:			
<pre>RAILS_ENV=production bundle exec rake db:migrate</pre>			
Got these errors, Redmine version: (3.3.0, 3.2.3 git master)			
<pre>-- add_index(:wiki_pages, [:wiki_id, :title], {:name=>:wiki_pages_wiki_id_title}) rake aborted! StandardError: An error has occurred, all later migrations canceled: Mysql2::Error: Specified key was too long; max key length is 767 bytes: CREATE INDEX `wiki_pages_wiki_id_title` ON `wiki_pages` (`wiki_id`, `title`) /opt/redmine/db/migrate/028_create_wiki_pages.rb:8:in `up' /usr/local/bin/bundle:22:in `load' /usr/local/bin/bundle:22:in `<main>' ActiveRecord::StatementInvalid: Mysql2::Error: Specified key was too long; max key length is 767 bytes: CREATE INDEX `wiki_pages_wiki_id_title` ON `wiki_pages` (`wiki_id`, `title`) /opt/redmine/db/migrate/028_create_wiki_pages.rb:8:in `up' /usr/local/bin/bundle:22:in `load' /usr/local/bin/bundle:22:in `<main>' Mysql2::Error: Specified key was too long; max key length is 767 bytes /opt/redmine/db/migrate/028_create_wiki_pages.rb:8:in `up' /usr/local/bin/bundle:22:in `load' /usr/local/bin/bundle:22:in `<main>' Tasks: TOP => db:migrate (See full trace by running task with --trace)</main></main></main></pre>			
Related issues:			
Related to Redmine - Feature #31921: Changes to properly support 4 byte chara...		Closed	

History

#1 - 2016-08-15 10:55 - Toshi MARUYAMA

- Status changed from New to Needs feedback

Try innodb_large_prefix.

<https://github.com/rails/rails/issues/9855#issuecomment-35618750>

#2 - 2017-04-11 19:48 - She Bytes

I had this problem too, and it turned out I needed to update mariaDB to beta v10.2.5 from (10.0.?) to get the additional setting for innodb_default_row_format

In your mysql cnf file for mariaDB, add these lines:

```
innodb_file_format=Barracuda #default for >= 10.2.2
innodb_file_per_table=ON #default for >= 5.5
innodb_large_prefix=1 #default for >= 10.2.2
innodb_default_row_format=DYNAMIC #default for >= 10.2.2
```

These lines should all be default at this point, but since it started working, I just left it.

My system:

Environment:
Redmine version 3.3.3.stable.16539
Ruby version 2.2.5-p319 (2016-04-26) [x86_64-linux]
Rails version 4.2.7.1
Environment production
Database adapter MySQL2

#3 - 2017-12-20 11:09 - Stanislav Tilsh

Sorry for broken English.

I install the version of Redmine 3.4.3 on Debian 9.

The redmine version was downloaded here: <https://www.redmine.org/releases/redmine-3.4.3.tar.gz>

An error during the import of the database to the redmine database being installed is repeated exactly as described above:

```
== 28 CreateWikiPages: migrating =====  
- create_table (:wiki_pages)  
  -> 0.0206s  
- add_index (:wiki_pages, [:wiki_id, :title], {:name=>:wiki_pages_wiki_id_title})  
rake aborted!  
StandardError: An error has occurred, all later migrations canceled:
```

```
MySQL2 :: Error: Index column size too large. The maximum column size is 767 bytes .: CREATE INDEX `wiki_pages_wiki_id_title` ON `wiki_pages` (`wiki_id`, `title`)
```

It is not possible to overcome this on my system:

```
Debian 9.  
redmine 3.4.3.  
ruby 2.3.3  
rails 4.2.7.1  
Environment production  
Database adapter MySQL2
```

#4 - 2017-12-20 11:58 - Stanislav Tilsh

Modifying the /home/sat/Redmine/redmine-3.4.3/db/migrate/028_create_wiki_pages.rb file to the following:

```
class CreateWikiPages < ActiveRecord::Migration  
  def self.up  
    create_table :wiki_pages , options: ' ROW_FORMAT=DYNAMIC ' do |t|  
      t.column :wiki_id, :integer, :null => false  
      t.column :title, :string, :limit => 255, :null => false  
      t.column :created_on, :datetime, :null => false  
    end  
    add_index :wiki_pages, [:wiki_id, :title], :name => :wiki_pages_wiki_id_title  
  end  
  def self.down  
    drop_table :wiki_pages  
  end  
end
```

I skipped migration to a similar problem:

```
67 CreateWikiRedirects: migrating =====  
-- create_table(:wiki_redirects)  
  -> 0.0205s  
-- add_index(:wiki_redirects, [:wiki_id, :title], {:name=>:wiki_redirects_wiki_id_title})  
rake aborted!  
StandardError: An error has occurred, all later migrations canceled:  
MySQL2::Error: Index column size too large. The maximum column size is 767 bytes.: CREATE INDEX `wiki_redirects_wiki_id_title` ON `wiki_redirects` (`wiki_id`, `title`)
```

#5 - 2017-12-20 12:15 - Stanislav Tilsh

Changed the file /home/sat/Redmine/redmine-3.4.3/db/migrate/067_create_wiki_redirects.rb adding to it:

```
, options: ' ROW_FORMAT=DYNAMIC '
```

having resulted a line to a kind:

```
create_table :wiki_redirects , options: ' ROW_FORMAT=DYNAMIC ' do |t|
```

The database structure was imported, but there was an error importing data into this structure:

```
-- add_index(:changesets, [:repository_id, :revision], {:unique=>true, :name=>:changesets_repos_rev})
rake aborted!
StandardError: An error has occurred, all later migrations canceled:
Mysql2::Error: Index column size too large. The maximum column size is 767 bytes.: CREATE UNIQUE INDEX `change
sets_repos_rev` ON `changesets` (`repository_id`, `revision`)
```

#6 - 2017-12-20 13:17 - Stanislav Tilsh

ups...

#7 - 2017-12-20 13:22 - Stanislav Tilsh

In the course of studying this problem, I found the following solution on the materials of this page:

<https://github.com/rails/rails/issues/9855#issuecomment-35618750>

Created the file:

```
/home/sat/Redmine/redmine-3.4.3/config/initializers/ar_innodb_row_format.rb
```

With the following content:

```
ActiveSupport.on_load :active_record do
  module ActiveRecord::ConnectionAdapters
    class AbstractMysqlAdapter
      def create_table_with_innodb_row_format(table_name, options = {})
        table_options = options.reverse_merge(:options => 'ENGINE=InnoDB ROW_FORMAT=DYNAMIC')
        create_table_without_innodb_row_format(table_name, table_options) do |td|
          yield td if block_given?
        end
      end
      alias_method_chain :create_table, :innodb_row_format
    end
  end
end
```

As a result, migration is performed completely:

- Invoke db:_dump (first_time)
- Execute db:_dump
- Invoke db:schema:dump (first_time)
- Invoke environment
- Invoke db:load_config
- Execute db:schema:dump

#8 - 2017-12-20 13:40 - Stanislav Tilsh

In the course of studying this problem, I found the following solution on the materials of this page:

<https://github.com/rails/rails/issues/9855#issuecomment-35618750>

Created the file:

```
/home/sat/Redmine/redmine-3.4.3/config/initializers/ar_innodb_row_format.rb
```

With the following content:

```
ActiveSupport.on_load :active_record do
  module ActiveRecord::ConnectionAdapters
    class AbstractMysqlAdapter
      def create_table_with_innodb_row_format(table_name, options = {})
        table_options = options.reverse_merge(:options => 'ENGINE=InnoDB ROW_FORMAT=DYNAMIC')
        create_table_without_innodb_row_format(table_name, table_options) do |td|
          yield td if block_given?
        end
      end
      alias_method_chain :create_table, :innodb_row_format
    end
  end
end
```

Changed my.conf:

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

adding to it:

```
innodb_large_prefix = 1
innodb_file_format = barracuda
innodb_file_per_table = 1
```

As a result, migration is performed completely:

- Invoke db:_dump (first_time)
- Execute db:_dump
- Invoke db:schema:dump (first_time)
- Invoke environment
- Invoke db:load_config
- Execute db:schema:dump

#9 - 2017-12-21 04:51 - Toshi MARUYAMA

Stanislav Tilsh wrote:

Sorry for broken English.
I install the version of Redmine 3.4.3 on Debian 9.

What is version of MySQL or MariaDB?
Which do you use utf8 or utf8mb4?

#10 - 2017-12-21 12:49 - Stanislav Tilsh

Toshi MARUYAMA wrote:

What is version of MySQL or MariaDB?
Which do you use utf8 or utf8mb4?

mysql Ver 15.1 Distrib 10.1.26-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2

```
MariaDB [(none)]> SHOW VARIABLES WHERE Variable_name LIKE 'character\_set\_%' OR Variable_name LIKE 'collation%';
```

```
character_set_client    utf8mb4
character_set_connection utf8mb4
character_set_database  utf8mb4
character_set_filesystem binary
character_set_results   utf8mb4
character_set_server    utf8mb4
character_set_system    utf8
collation_connection    utf8mb4_general_ci
collation_database     utf8mb4_general_ci
collation_server        utf8mb4_general_ci
```

```
MariaDB [(none)]>
```

#11 - 2018-01-11 17:58 - Andrey Luzgin

- *File key_too_long_fix.patch added*

Inserted , options: ' ROW_FORMAT=DYNAMIC ' after all create_table
For apply patch run: patch -p1 < ../key_too_long_fix.patch

You should see:

```
patching file db/migrate/001_setup.rb
patching file db/migrate/007_create_journals.rb
patching file db/migrate/008_create_user_preferences.rb
patching file db/migrate/010_create_comments.rb
patching file db/migrate/013_create_queries.rb
patching file db/migrate/015_create_repositories.rb
patching file db/migrate/017_create_settings.rb
patching file db/migrate/027_create_wikis.rb
patching file db/migrate/028_create_wiki_pages.rb
patching file db/migrate/029_create_wiki_contents.rb
patching file db/migrate/032_create_time_entries.rb
patching file db/migrate/034_create_changesets.rb
patching file db/migrate/035_create_changes.rb
patching file db/migrate/039_create_watchers.rb
patching file db/migrate/040_create_changesets_issues.rb
```

patching file db/migrate/042_create_issue_relations.rb
patching file db/migrate/045_create_boards.rb
patching file db/migrate/046_create_messages.rb
patching file db/migrate/067_create_wiki_redirects.rb
patching file db/migrate/068_create_enabled_modules.rb
patching file db/migrate/081_create_projects_trackers.rb
patching file db/migrate/107_add_open_id_authentication_tables.rb
patching file db/migrate/20090503121501_create_member_roles.rb
patching file db/migrate/20090704172355_create_groups_users.rb
patching file db/migrate/20110902000000_create_changeset_parents.rb
patching file db/migrate/20130602092539_create_queries_roles.rb
patching file db/migrate/20130713104233_create_custom_fields_roles.rb
patching file db/migrate/20150113194759_create_email_addresses.rb
patching file db/migrate/20150528092912_create_roles_managed_roles.rb
patching file db/migrate/20150730122707_create_imports.rb
patching file db/migrate/20150730122735_create_import_items.rb
patching file db/migrate/20151025072118_create_custom_field_enumerations.rb

#12 - 2018-05-06 23:22 - Philippe Ferrucci

This patch did nothing here (Debian 9 - 4.9.30-2+deb9u5 (2017-09-19), Ruby 2.5.1, MySQL 10.1.26-MariaDB-0+deb9u1).

I had to recreate the database with the correct charset, as explained by Leslie in [issue 54308](#)

#13 - 2018-09-14 17:03 - Luc Lalonde

This solved the problem... Add this script redmine/config/initializers/mysqlpls.rb:

```
require 'active_record/connection_adapters/abstract_mysql_adapter'

module ActiveRecord
  module ConnectionAdapters
    class AbstractMysqlAdapter
      NATIVE_DATABASE_TYPES[:string] = { :name => "varchar", :limit => 191 }
    end
  end
end
```

I would like to give credit to the person who came up with this solution... But I can't remember where I got it from.

#14 - 2019-01-11 14:17 - mr gosh

Stanislav Tilsh wrote:

Created the file:

```
/home/sat/Redmine/redmine-3.4.3/config/initializers/ar_innodb_row_format.rb
```

With the following content:

```
ActiveSupport.on_load :active_record do
  module ActiveRecord::ConnectionAdapters
    class AbstractMysqlAdapter
      def create_table_with_innodb_row_format(table_name, options = {})
        table_options = options.reverse_merge(:options => 'ENGINE=InnoDB ROW_FORMAT=DYNAMIC')
        create_table_without_innodb_row_format(table_name, table_options) do |td|
          yield td if block_given?
        end
      end
      alias_method_chain :create_table, :innodb_row_format
    end
  end
end
```

As a result, migration is performed completely:

- Invoke db:_dump (first_time)
- Execute db:_dump
- Invoke db:schema:dump (first_time)
- Invoke environment
- Invoke db:load_config
- Execute db:schema:dump

THX! this is still needed on my ubuntu 18.04 installation with mariadb from the repos...!

#15 - 2019-08-17 18:14 - Marius BĂLTEANU

- *Related to Feature #31921: Changes to properly support 4 byte characters (emoji) when database is MySQL added*

#16 - 2020-03-08 09:13 - Go MAEDA

- *Status changed from Needs feedback to Closed*

- *Resolution set to Fixed*

The error does not occur even when you set the encoding to utf8mb4 if you use MySQL 5.7.7 or later.

Files

key_too_long_fix.patch	31.5 KB	2018-01-11	Andrey Luzgin
------------------------	---------	------------	---------------