

Redmine - Defect #27283

Failed to run "rake db:migrate" command against clean database on MySQL 5.7

2017-10-24 18:08 - Akiko Takano

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Jean-Philippe Lang	% Done:	0%
Category:	Database	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Fixed		

Description

I'm not sure if this is a defect or not, but I hope to feedback this situation.

Summary

Failed to run "rake db:migrate" command against clean database on MySQL5.7 under Redmine4.0 / Rails5.

Description

I always fail to migrate against clean (blank) database on MySQL 5.7 under Redmine4.0 (trunk).

Environment

- MySQL5.7 / Default setup, without any custom configurations
- Redmine trunk (SVN: trunk, Mercurial: default, GitHub: master branch)
- Without plugins
- Database and Redmine under Docker container and VirtualBox VM CentOS7

Visual Proof

Here is an excerpt from the log.

```
== 20150113194759 CreateEmailAddresses: migrating =====
-- adapter_name()
--> 0.0005s
-- adapter_name()
--> 0.0003s
-- create_table(:email_addresses, {:id=>:integer})
rake aborted!
StandardError: An error has occurred, all later migrations canceled:

Mysql2::Error: Invalid default value for 'updated_on':

.....

ActiveRecord::StatementInvalid: Mysql2::Error: Invalid default value for 'updated_on'
```

Steps to reproduce

1. Install Redmine and MySQL like above. (Especially, run MySQL5.7 default with settings)
2. Create database.yml for mysql.
3. Run migration task: rake db:create && rake db:migrate

At first, migration process proceeds normally, but an error occurs in the middle, at "20150113194759 CreateEmailAddresses".

Expected Results & Actual Results

- Expected: Migration completed successfully
- Result: Failed

Workaround

As error message says, this may be caused, "Changed the default null value for timestamps to false".

Ref. http://guides.rubyonrails.org/5_0_release_notes.html

A. Change migration file

```
diff --git a/db/migrate/20150113194759_create_email_addresses.rb b/db/migrate/20150113194759_create_email_addresses.rb
index 22ad19e..fd49722 100644
--- a/db/migrate/20150113194759_create_email_addresses.rb
+++ b/db/migrate/20150113194759_create_email_addresses.rb
@@ -6,7 +6,7 @@ class CreateEmailAddresses < ActiveRecord::Migration[4.2]
   t.column :is_default, :boolean, :null => false, :default => false
   t.column :notify, :boolean, :null => false, :default => true
   t.column :created_on, :timestamp, :null => false
-  t.column :updated_on, :timestamp, :null => false
+  t.column :updated_on, :timestamp
   end
 end
end
```

If applied above, the result of "show create table email_addresses;" is following:

```
| email_addresses | CREATE TABLE `email_addresses` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `address` varchar(255) NOT NULL,
  `is_default` tinyint(1) NOT NULL DEFAULT '0',
  `notify` tinyint(1) NOT NULL DEFAULT '1',
  `created_on` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `updated_on` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `index_email_addresses_on_user_id` (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 |
```

After migration finished, I confirmed EmailAddress model's behavior.

(Exp. e = EmailAddress.create(..), e.update(address: xxxxx))

It seems to work correctly.

B. Change MySQL's sql-mode

Change MySQL sql-mode to "", or remove "STRICT_ALL_TABLES" statement from sql-mode.

If applied above, migration completed without any change to migration files.

But I'm not sure which is the best way because I have not tried some other database engines...

My report is as above.

I hope this would be any help.

Related issues:

Related to Redmine - Feature #23630: Migrate to Rails 5.2

Closed

Associated revisions

Revision 17515 - 2018-09-23 16:43 - Jean-Philippe Lang

Failed to run "rake db:migrate" command against clean database on MySQL5.7 (#27283).

Patch by Marius BALTEANU.

History

#1 - 2017-10-24 18:09 - Akiko Takano

- File migration-failed.jpg added

- File migration-error-trace.txt added

Upload capture and error log.

Screenshot of migration error

migration-failed.jpg

Current Env

```
sh-4.3# rails about
About your application's environment
Rails version      5.1.2
Ruby version       2.4.1-p111 (x86_64-linux)
RubyGems version   2.6.11
Rack version       2.0.3
Middleware         Rack::Sendfile, Rack::ContentLength, ActionDispatch::Static, ActionDispatch::Executo
r, ActiveSupport::Cache::Strategy::LocalCache::Middleware, Rack::Runtime, Rack::MethodOverride, ActionDispatch
::RequestId, ActionDispatch::RemoteIp, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::De
bugExceptions, ActionDispatch::Reloader, ActionDispatch::Callbacks, ActionDispatch::Cookies, ActionDispatch::S
ession::CookieStore, ActionDispatch::Flash, Rack::Head, Rack::ConditionalGet, Rack::ETag, RequestStore::Middle
ware, Module
Application root   /tmp/redmine
Environment        development
Database adapter   mysql2
Database schema version 20170723112801
```

```
sh-4.3# git branch
```

```
* master
```

```
sh-4.3# git log
```

```
commit fc1912442fed541c8d3a207ce71bbeb9569a32d1
```

```
Author: Jean-Philippe Lang <jp_lang@.....>
```

```
Date: Sun Oct 15 19:50:46 2017 +0000
```

```
    Adds changes from 3.3.5.
```

```
    git-svn-id: http://svn.redmine.org/redmine/trunk@17004 e93f8b46-1217-0410-a6f0-8f06a7374b81
```

#2 - 2017-10-24 18:30 - Toshi MARUYAMA

- Description updated

#3 - 2017-11-15 13:06 - Nishant Karki

- File Screen Shot 2017-11-15 at 5.50.06 PM.png added

I'm seeing this issue with both master branch and 3.4-stable on github

Current Env: OSX

```
$ rails about
```

```
About your application's environment
```

```
Rails version      5.1.2
```

```
Ruby version       2.3.0-p0 (x86_64-darwin16)
```

```
RubyGems version   2.5.1
```

```
Rack version       2.0.3
```

```
Middleware         Rack::Sendfile, Rack::ContentLength, ActionDispatch::Static, ActionDispatch::Executor,
```

```
ActiveSupport::Cache::Strategy::LocalCache::Middleware, Rack::Runtime, Rack::MethodOverride, ActionDispatch::RequestId,
```

```
ActionDispatch::RemoteIp, Rails::Rack::Logger, ActionDispatch::ShowExceptions, ActionDispatch::DebugExceptions, ActionDispatch::Reloader,
```

```
ActionDispatch::Callbacks, ActionDispatch::Cookies, ActionDispatch::Session::CookieStore, ActionDispatch::Flash, Rack::Head,
```

```
Rack::ConditionalGet, Rack::ETag, RequestStore::Middleware, Module
```

```
Application root   /Users/nishant/projects/ruby/redmine
```

```
Environment        development
```

```
Database adapter   mysql2
```

```
Database schema version 0
```

#4 - 2017-11-23 07:36 - Toshi MARUYAMA

- Category set to Database

- Target version set to 4.0.0

I tried this.

```
diff --git a/db/migrate/20150113194759_create_email_addresses.rb b/db/migrate/20150113194759_create_email_addresses.rb
```

```
--- a/db/migrate/20150113194759_create_email_addresses.rb
```

```
+++ b/db/migrate/20150113194759_create_email_addresses.rb
```

```
@@ -6,7 +6,7 @@ class CreateEmailAddresses < ActiveRecord
```

```

    t.column :is_default, :boolean, :null => false, :default => false
    t.column :notify, :boolean, :null => false, :default => true
    t.column :created_on, :timestamp, :null => false
-   t.column :updated_on, :timestamp, :null => false
+   t.column :updated_on, :timestamp, :null => false, default: -> { 'NOW()' }
  end
end
end
end

```

```
mysql> desc email_addresses;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
user_id	int(11)	NO	MUL	NULL	
address	varchar(255)	NO		NULL	
is_default	tinyint(1)	NO		0	
notify	tinyint(1)	NO		1	
created_on	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
updated_on	timestamp	NO		CURRENT_TIMESTAMP	

```
7 rows in set (0.00 sec)
```

#5 - 2017-11-23 07:40 - Toshi MARUYAMA

Nishant Karki wrote:

I'm seeing this issue with both master branch and 3.4-stable on github

I cannot reproduce on 3.4-stable MySQL 5.7.

#6 - 2017-12-10 05:42 - Akiko Takano

MARUYAMA-San,

Thank you for handling this report.

I've tried to run migration task against MySQL 5.7, and completed successfully.

```

-   t.column :updated_on, :timestamp, :null => false
+   t.column :updated_on, :timestamp, :null => false, default: -> { 'NOW()' }

```

But, it seems there are some difference from above table definition.

Here is my result, as FYI.

Please note: No customize against SQL mode. (I used MySQL official Docker image)

```
mysql> SELECT @@GLOBAL.sql_mode;
```

@@GLOBAL.sql_mode
ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGINE_SUBSTITUTION

```
1 row in set (0.00 sec)
```

```
mysql> SELECT @@VERSION;
```

@@VERSION
5.7.20

```
1 row in set (0.00 sec)
```

```
mysql> desc email_addresses;
```

Field	Type	Null	Key	Default	Extra
id	bigint(20)	NO	PRI	NULL	auto_increment
user_id	int(11)	NO	MUL	NULL	
address	varchar(255)	NO		NULL	

```

| is_default | tinyint(1) | NO | | 0 | |
| notify     | tinyint(1) | NO | | 1 | |
| created_on | datetime   | NO | | NULL | |
| updated_on | datetime   | NO | | NULL | |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

On the other hand, I also tried to run migrate task against new SQLite database, and unfortunately, migration was failed. This is because SQLite3 does not support 'NOW()' function. (It seems both PostgreSQL and MySQL support this function.)

Maybe MSSQL also does not support NOW() function. I hope this will be of some help.

#7 - 2017-12-28 20:55 - Toshi MARUYAMA

Ref: <https://github.com/rails/rails/issues/23418>

```

diff --git a/db/migrate/20150113194759_create_email_addresses.rb b/db/migrate/20150113194759_create_email_addresses.rb
--- a/db/migrate/20150113194759_create_email_addresses.rb
+++ b/db/migrate/20150113194759_create_email_addresses.rb
@@ -1,12 +1,13 @@
-class CreateEmailAddresses < ActiveRecord::Migration[4.2]
+class CreateEmailAddresses < ActiveRecord::Migration[5.0]
  def change
    create_table :email_addresses do |t|
      t.column :user_id, :integer, :null => false
      t.column :address, :string, :null => false
      t.column :is_default, :boolean, :null => false, :default => false
      t.column :notify, :boolean, :null => false, :default => true
-     t.column :created_on, :timestamp, :null => false
-     t.column :updated_on, :timestamp, :null => false
+     t.timestamps
    end
+    rename_column :email_addresses, :created_at, :created_on
+    rename_column :email_addresses, :updated_at, :updated_on
  end
end

```

```

mysql> SELECT @@GLOBAL.sql_mode;
+-----+
| @@GLOBAL.sql_mode |
+-----+
| ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+-----+
1 row in set (0.00 sec)

```

```

mysql> SELECT @@VERSION;
+-----+
| @@VERSION |
+-----+
| 5.7.20 |
+-----+
1 row in set (0.00 sec)

```

```

mysql> desc redmine.email_addresses;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| user_id | int(11) | NO | | NULL | |
| address | varchar(255) | NO | | NULL | |
| is_default | tinyint(1) | NO | | 0 | |
| notify | tinyint(1) | NO | | 1 | |
| created_on | datetime | NO | | NULL | |
| updated_on | datetime | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

```

#8 - 2017-12-29 05:09 - Toshi MARUYAMA

- Target version deleted (4.0.0)

#9 - 2018-01-26 20:10 - Toshi MARUYAMA

<https://stackoverflow.com/questions/42020422/mysql-server-5-7-17-multiple-timestamp-not-null-fields-on-table>

#10 - 2018-01-26 20:37 - Toshi MARUYAMA

I have created issue.

<https://github.com/rails/rails/issues/31804>

#11 - 2018-04-30 23:21 - Marius BĂLTEANU

- Related to Feature #23630: Migrate to Rails 5.2 added

#12 - 2018-05-14 23:45 - Marius BĂLTEANU

- Tracker changed from Patch to Defect

- Status changed from New to Confirmed

- Target version set to 4.0.0

I'm reassigning this defect to [4.0.0](#) because the Github issue was closed without any solution and this problem still blocks the migrations on a default Mysql 5.7. I'm considering this ticket a real issue.

#13 - 2018-06-17 15:40 - Jean-Philippe Lang

Migration is working fine for me with mysql 5.7.22 and default config using current trunk. Please provide your my.conf in order to reproduce.

#14 - 2018-06-17 16:53 - Go MAEDA

Jean-Philippe Lang wrote:

Migration is working fine for me with mysql 5.7.22 and default config using current trunk. Please provide your my.conf in order to reproduce.

I can reproduce the problem.

- Ubuntu 16.04.2 LTS
- MySQL 5.7.22-0ubuntu0.16.04.1

/etc/mysql/my.cnf:

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
character-set-server=utf8

[mysql]
default-character-set=utf8
```

Error message:

```
== 20150113194759 CreateEmailAddresses: migrating =====
-- adapter_name()
--> 0.0000s
-- adapter_name()
--> 0.0000s
-- create_table(:email_addresses, {:id=>:integer})
rake aborted!
StandardError: An error has occurred, all later migrations canceled:

Mysql2::Error: Invalid default value for 'updated_on': CREATE TABLE `email_addresses` (`id` int NOT NULL AUTO_INCREMENT PRIMARY KEY, `user_id` int NOT NULL, `address` varchar(255) NOT NULL, `is_default` tinyint(1) DEFAULT 0 NOT NULL, `notify` tinyint(1) DEFAULT 1 NOT NULL, `created_on` timestamp NOT NULL, `updated_on` timestamp NOT NULL) ENGINE=InnoDB
...
```

Official Docker image:

```
mysql Ver 14.14 Distrib 5.7.19, for Linux (x86_64) using EditLine wrapper
```

sql_mode

```
mysql> SELECT @@GLOBAL.sql_mode;
+-----+
| @@GLOBAL.sql_mode |
+-----+
| ONLY_FULL_GROUP_BY, STRICT_TRANS_TABLES, NO_ZERO_IN_DATE, NO_ZERO_DATE, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGINE_SUBSTITUTION |
+-----+
1 row in set (0.00 sec)
```

explicit_defaults_for_timestamp

```
mysql> select @@GLOBAL.explicit_defaults_for_timestamp;
+-----+
| @@GLOBAL.explicit_defaults_for_timestamp |
+-----+
| 0 |
+-----+
1 row in set (0.00 sec)
```

MySQL configs:

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
```

```
root@2a83a66372bd:/# cat /etc/mysql/conf.d/docker.cnf
[mysqld]
skip-host-cache
skip-name-resolve
```

```
root@2a83a66372bd:/# cat /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
#
# The MySQL Server configuration file.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

[mysqld]
pid-file      = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
datadir       = /var/lib/mysql
#log-error    = /var/log/mysql/error.log
# By default we only accept connections from localhost
#bind-address = 127.0.0.1
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
```

db:migrate

```
notroot@42f1f3063342:/work$ rake db:drop db:create db:migrate
Dropped database 'redmine_test'
Created database 'redmine_test'
== 1 Setup: migrating =====
-- adapter_name()
--> 0.0000s
-- adapter_name()
...
== 20150113194759 CreateEmailAddresses: migrating =====
-- adapter_name()
```

```

-> 0.0000s
-- adapter_name()
-> 0.0000s
-- create_table(:email_addresses, {:id=>:integer})
rake aborted!
StandardError: An error has occurred, all later migrations canceled:

Mysql2::Error: Invalid default value for 'updated_on': CREATE TABLE `email_addresses` (`id` int NOT NULL AUTO_INCREMENT PRIMARY KEY, `user_id` int NOT NULL, `address` varchar(255) NOT NULL, `is_default` tinyint(1) DEFAULT 0 NOT NULL, `notify` tinyint(1) DEFAULT 1 NOT NULL, `created_on` timestamp NOT NULL, `updated_on` timestamp NOT NULL) ENGINE=InnoDB

```

Setting `explicit_defaults_for_timestamp` to 1 fixes the issue:

```
mysql> set global explicit_defaults_for_timestamp =1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select @@GLOBAL.explicit_defaults_for_timestamp;
+-----+
| @@GLOBAL.explicit_defaults_for_timestamp |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

```

notroot@42f1f3063342:/work$ rake db:migrate
== 20150113194759 CreateEmailAddresses: migrating =====
-- adapter_name()
-> 0.0000s
-- adapter_name()
-> 0.0000s
-- create_table(:email_addresses, {:id=>:integer})
-> 0.0994s
== 20150113194759 CreateEmailAddresses: migrated (0.0998s) =====

== 20150113211532 PopulateEmailAddresses: migrating =====
== 20150113211532 PopulateEmailAddresses: migrated (0.1006s) =====

```

but I don't think that this should be the solution.

#16 - 2018-06-23 07:15 - Jean-Philippe Lang

Could someone provide a patch?

#17 - 2018-06-26 09:34 - Marius BĂLTEANU

- File `0001-fix-migration.patch` added

I made some checks and it seems that the changes between `rake db:migrate` on Rails 4.2 and `rake db:migrate` on Rails 5.2 are quite big.

First of all, it seems that the `:timestamp` method from migrations generates different column types:

Redmine 3.4.6 with Rails 4.2

```
mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| login          | varchar(255)  | NO   |     |         |                |
| hashed_password | varchar(40)   | NO   |     |         |                |
| firstname      | varchar(30)   | NO   |     |         |                |
| lastname       | varchar(255)  | NO   |     |         |                |
| admin          | tinyint(1)    | NO   |     | 0       |                |
| status        | int(11)       | NO   |     | 1       |                |
| last_login_on  | datetime      | YES  |     | NULL    |                |
| language       | varchar(5)    | YES  |     |         |                |
| auth_source_id | int(11)       | YES  | MUL | NULL    |                |
| created_on     | datetime      | YES  |     | NULL    |                |
| updated_on     | datetime      | YES  |     | NULL    |                |
| type          | varchar(255)  | YES  | MUL | NULL    |                |
| identity_url   | varchar(255)  | YES  |     | NULL    |                |
| mail_notification | varchar(255) | NO   |     |         |                |

```



```
| salt          | varchar(64) | YES | | NULL |
| must_change_passwd | tinyint(1) | NO  | | 0    |
| passwd_changed_on | datetime   | YES | | NULL |
+-----+-----+-----+-----+-----+
18 rows in set (0.00 sec)
```

Redmine 4.0.0 with Rails 5.2

```
mysql> desc redmine_test.users;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| login          | varchar(255)  | NO   |     |         |                |
| hashed_password | varchar(40)   | NO   |     |         |                |
| firstname      | varchar(30)   | NO   |     |         |                |
| lastname       | varchar(255)  | NO   |     |         |                |
| admin          | tinyint(1)    | NO   |     | 0       |                |
| status         | int(11)       | NO   |     | 1       |                |
| last_login_on  | datetime      | YES  |     | NULL    |                |
| language       | varchar(5)    | YES  |     |         |                |
| auth_source_id | int(11)       | YES  | MUL | NULL    |                |
| created_on     | timestamp     | YES  |     | NULL    |                |
| updated_on     | timestamp     | YES  |     | NULL    |                |
| type           | varchar(255)  | YES  | MUL | NULL    |                |
| identity_url   | varchar(255)  | YES  |     | NULL    |                |
| mail_notification | varchar(255) | NO   |     |         |                |
| salt           | varchar(64)   | YES  |     | NULL    |                |
| must_change_passwd | tinyint(1) | NO   |     | 0       |                |
| passwd_changed_on | datetime   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
18 rows in set (0.01 sec)
```

As can be observed, the columns `created_on` and `updated_on` on Rails 4.2 are of type `:datetime`, but in Rails 5.2, the type changed to `:timestamp`. This behaviour can be observed in all tables that have columns of type `:timestamp` in the migrations.

For me, the attached patch fixes the migration on Rails 5.2 and generates the same table as Rails 4.2. All tests pass.

Redmine 3.4.6 with Rails 4.2

```
mysql> desc email_addresses;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| user_id        | int(11)       | NO   | MUL | NULL    |                |
| address        | varchar(255)  | NO   |     | NULL    |                |
| is_default     | tinyint(1)    | NO   |     | 0       |                |
| notify         | tinyint(1)    | NO   |     | 1       |                |
| created_on     | datetime      | NO   |     | NULL    |                |
| updated_on     | datetime      | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Redmine 4.0.0 with Rails 5.2

```
mysql> desc redmine_test.email_addresses;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       | NO   | PRI | NULL    | auto_increment |
| user_id        | int(11)       | NO   | MUL | NULL    |                |
| address        | varchar(255)  | NO   |     | NULL    |                |
| is_default     | tinyint(1)    | NO   |     | 0       |                |
| notify         | tinyint(1)    | NO   |     | 1       |                |
| created_on     | datetime      | NO   |     | NULL    |                |
| updated_on     | datetime      | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Even if the versioned migration from Rails 5.2 says that the behaviour for old migrations will not be changed when you specify the version, in my tests, it doesn't work.

I'm not sure if what I said is 100% correct, but these are the results of my findings and understandings.

Also, if I'm right, I think that we should change ":timestamp" with ":datetime" in all migration files (this is Rails 5.2 default type, see Toshi's note 6).

#18 - 2018-08-08 21:21 - Marius BĂLTEANU

Unfortunately, the issue still reproduces on Rails 5.2.1.

#19 - 2018-09-23 16:43 - Jean-Philippe Lang

- Subject changed from Failed to run "rake db:migrate" command against clean database on MySQL5.7 under Redmine4.0 / Rails5 to Failed to run "rake db:migrate" command against clean database on MySQL5.7

- Status changed from Confirmed to Closed

- Assignee set to Jean-Philippe Lang

- Resolution set to Fixed

I was able to reproduce with the details about your Mysql configuration.
Committed, thanks.

#20 - 2018-09-25 03:17 - Go MAEDA

- Subject changed from Failed to run "rake db:migrate" command against clean database on MySQL5.7 to Failed to run "rake db:migrate" command against clean database on MySQL 5.7

#21 - 2018-12-08 15:41 - Go MAEDA

- Target version deleted (4.0.0)

Removing the target version because I think this issue is a part of [#23630](#).

The bug only existed in the trunk while migrating to Rails 5 and was not included in past releases, therefore it is not necessary to add this issue to the changelog.

Files

migration-failed.jpg	125 KB	2017-10-24	Akiko Takano
migration-error-trace.txt	14.6 KB	2017-10-24	Akiko Takano
Screen Shot 2017-11-15 at 5.50.06 PM.png	350 KB	2017-11-15	Nishant Karki
0001-fix-migration.patch	1.07 KB	2018-06-26	Marius BĂLTEANU