

- Remove `ActiveRecord.legacy_connection_handling`.

Eileen M. Uchitelle

- `rails db:schema:{dump,load}` now checks `ENV["SCHEMA_FORMAT"]` before config

Since `rails db:structure:{dump,load}` was deprecated there wasn't a simple way to dump a schema to both SQL and Ruby formats. You can now do this with an environment variable. For example:

```
SCHEMA_FORMAT=sql rake db:schema:dump
```

Alex Ghiculescu

- Fixed MariaDB default function support.

Defaults would be written wrong in `db/schema.rb` and not work correctly if using `db:schema:load`. Further more the function name would be added as string content when saving new records.

kaspernj

- Add `active_record.destroy_association_async_batch_size` configuration

This allows applications to specify the maximum number of records that will be destroyed in a single background job by the `dependent::destroy_async` association option. By default, the current behavior will remain the same: when a parent record is destroyed, all dependent records will be destroyed in a single background job. If the number of dependent records is greater than this configuration, the records will be destroyed in multiple background jobs.

Nick Holden

- Fix `remove_foreign_key` with `:if_exists` option when foreign key actually exists.

fatkodima

- Remove `--no-comments` flag in structure dumps for PostgreSQL

This broke some apps that used custom schema comments. If you don't want comments in your structure dump, you can use:

```
ActiveRecord::Tasks::DatabaseTasks.structure_dump_flags = ['--no-comments']
```

Alex Ghiculescu

- Reduce the memory footprint of fixtures accessors.

Until now fixtures accessors were eagerly defined using `define_method`. So the memory usage was directly dependent of the number of fixtures and test suites.

Instead fixtures accessors are now implemented with `method_missing`, so they incur much less memory and CPU overhead.

Jean Boussier

- Fix `config.active_record.destroy_association_async_job` configuration

`config.active_record.destroy_association_async_job` should allow applications to specify the job that will be used to destroy associated records in the background for `has_many` associations with the `dependent::destroy_async` option. Previously, that was ignored, which meant the default `ActiveRecord::DestroyAssociationAsyncJob` always destroyed records in the background.

Nick Holden

- Fix `change_column_comment` to preserve column's `AUTO_INCREMENT` in the MySQL adapter

fatkodima

- Fix quoting of `ActiveSupport::Duration` and `Rational` numbers in the MySQL adapter.

Kevin McPhillips

- Allow column name with `COLLATE` (e.g., title `COLLATE "C"`) as safe SQL string

Shugo Maeda

- Permit underscores in the `VERSION` argument to database rake tasks.

Eddie Lebow

- Reversed the order of `INSERT` statements in `structure.sql` dumps

This should decrease the likelihood of merge conflicts. New migrations will now be added at the top of the list.

For existing apps, there will be a large diff the next time `structure.sql` is generated.

Alex Ghiculescu, Matt Larraz

- Fix `PG.connect` keyword arguments deprecation warning on ruby 2.7
Fixes #44307.

Nikita Vasilevsky

- Fix dropping DB connections after serialization failures and deadlocks.

Prior to 6.1.4, serialization failures and deadlocks caused rollbacks to be issued for both real transactions and savepoints. This breaks MySQL which disallows rollbacks of savepoints following a deadlock.

6.1.4 removed these rollbacks, for both transactions and savepoints, causing the DB connection to be left in an unknown state and thus discarded.

These rollbacks are now restored, except for savepoints on MySQL.

Thomas Morgan

- Make ActiveRecord::ConnectionPool Fiber-safe

When ActiveSupport::IsolatedExecutionState.isolation_level is set to :fiber, the connection pool now supports multiple Fibers from the same Thread checking out connections from the pool.

Alex Matchneer

- Add update_attribute! to ActiveRecord::Persistence

Similar to update_attribute, but raises ActiveRecord::RecordNotSaved when a before_* callback throws :abort.

```
class Topic < ActiveRecord::Base
  before_save :check_title

  def check_title
    throw(:abort) if title == "abort"
  end
end

topic = Topic.create(title: "Test Title")
# ==> #<Topic title: "Test Title">
topic.update_attribute!(:title, "Another Title")
# ==> #<Topic title: "Another Title">
topic.update_attribute!(:title, "abort")
# raises ActiveRecord::RecordNotSaved
```

Drew Tempelmeyer

- Avoid loading every record in ActiveRecord::Relation#pretty_print

```
# Before
pp Foo.all # Loads the whole table.

# After
pp Foo.all # Shows 10 items and an ellipsis.
```

Ulysse Buonomo

- Change QueryMethods#in_order_of to drop records not listed in values.

`in_order_of` now filters down to the values provided, to match the behavior of the `Enumerable` version.

Kevin Newton

- Allow named expression indexes to be revertible.

Previously, the following code would raise an error in a reversible migration executed while rolling back, due to the index name not being used in the index removal.

```
add_index(:settings, "(data->'property')", using: :gin, name: :index_settings_data_prop
```

Fixes #43331.

Oliver Günther

- Fix incorrect argument in PostgreSQL structure dump tasks.

Updating the `--no-comment` argument added in Rails 7 to the correct `--no-comments` argument.

Alex Dent

- Fix migration compatibility to create SQLite references/belongs_to column as integer when migration version is 6.0.

Reference/belongs_to in migrations with version 6.0 were creating columns as bigint instead of integer for the SQLite Adapter.

Marcelo Lauxen

- Add a deprecation warning when `prepared_statements` configuration is not set for the mysql2 adapter.

Thiago Araujo and Stefanni Brasil

- Fix `QueryMethods#in_order_of` to handle empty order list.

```
Post.in_order_of(:id, []).to_a
```

Also more explicitly set the column as secondary order, so that any other value is still ordered.

Jean Boussier

- Fix quoting of column aliases generated by calculation methods.

Since the alias is derived from the table name, we can't assume the result is a valid identifier.

```
class Test < ActiveRecord::Base
  self.table_name = '1abc'
end
Test.group(:id).count
```

```
# syntax error at or near "1" (ActiveRecord::StatementInvalid)
# LINE 1: SELECT COUNT(*) AS count_all, "1abc"."id" AS 1abc_id FROM "1..."
```

Jean Boussier

- Add `authenticate_by` when using `has_secure_password`.

`authenticate_by` is intended to replace code like the following, which returns early when a user with a matching email is not found:

```
User.find_by(email: "...")&.authenticate(...)
```

Such code is vulnerable to timing-based enumeration attacks, wherein an attacker can determine if a user account with a given email exists. After confirming that an account exists, the attacker can try passwords associated with that email address from other leaked databases, in case the user re-used a password across multiple sites (a common practice). Additionally, knowing an account email address allows the attacker to attempt a targeted phishing (“spear phishing”) attack.

`authenticate_by` addresses the vulnerability by taking the same amount of time regardless of whether a user with a matching email is found:

```
User.authenticate_by(email: "...", password: "...")
```

Jonathan Hefner

Please check 7-0-stable for previous changes.