

[How To Export Redmine Issues Using SQL](#)

By [Roger Keays](#), 18 June 2015

I stopped using Redmine a few years ago and left all that data in the database just sitting there rotting away. Some of my notes were pretty good so I decided to see if I could get all those issues out of Redmine without reinstalling the whole damn stack again.

Here is the query I came up with to export all the issues and issue comments into a readable text format. Once exported you can load it up in a text editor that supports code folding like VIM.

The query is for postgresql and has some non-standard SQL functions so it is not portable, but you might be able to find some replacements for those functions for your database system.



```
$ psql -A -F ' ' -c "
  select '{{{ ' || project_id || '/' || id || ' ' || subject || E'\n\n',
         replace(regexp_replace(description, '\s+$', ''), '}}}', '}}}') || E
         (select array_to_string(array_agg('Comment posted on ' || created_on
         replace(regexp_replace(notes, '\s+$', ''), '}}}', '}} } }')), :
         (select * from journals where notes <> '' and journalized_type = 'issue'
         journalized_id=issues.id order by created_on) as or
from issues order by project_id desc, created_on desc" -Upostgres -hl
```

```
$ vim issues.txt
:set foldmethod=marker
```

It's surprising how useful simple text format is. No database, no network, no ruby, no browser. Just good ol' text.

[No comments yet, be the first to comment!](#)

[JPA CascadeType.REMOVE vs Hibernate @OnDelete](#)

By [Roger Keays](#), 13 June 2014

Somehow database models and ORM always end up being more difficult than you expect. Here is a common source of confusion between JPA cascade operations and database cascade operations. Basically they do the opposite thing. For example:

```
public class House {  
  
    @OneToOne  
    Object door;  
}
```

If you use **CascadeType.REMOVE** then deleting the house will also delete the door (using an extraSQL statement).

```
@OneToOne(cascade=CascadeType.REMOVE)  
Object door;
```

If you use **@OnDelete** then deleting the door will also delete the house (using an ON DELETE CASCADE database foreign key).

```
@OneToOne  
@OnDelete(action = OnDeleteAction.CASCADE)  
Object door;
```

JPA has not standardized the ON DELETE and ON UPDATE foreign key actions possibly because they are SQL-specific and JPA is supposed to be storage-agnostic. I think this is unfortunate -what I'm looking for is ON DELETE SET NULL which would mean that when I delete the door, House.door gets set to null automatically. It's a fairly common requirement and is implemented in OpenJPA like this:

```
@OneToOne  
@ForeignKey(deleteAction=ForeignKeyAction.NULL)  
Object door;
```

For the moment it looks like I'll have to stick to OpenJPA. Not sure why this isn't an option in Hibernate.

[1 comment, post a comment!](#)



[java.lang.ClassFormatError Exception With EclipseLink Static Weaving](#) [\[SOLVED\]](#)

By [Roger Keays](#), 19 March 2012

Here's an interesting exception that slapped me when I tried out using the EclipseLink static weaver on against a fresh Maven pom.xml for Java EE Web projects. The same exception occurs with the OpenJPA enhancer.



Executing tasks

```
[java] Exception in thread "main" java.lang.ClassFormatError: Absent Code
[java]         at java.lang.ClassLoader.defineClass1(Native Method)
[java]         at java.lang.ClassLoader.defineClassCond(ClassLoader.java:
[java]         at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
[java]         at java.security.SecureClassLoader.defineClass(SecureClass:
[java]         at java.net.URLClassLoader.defineClass(URLClassLoader.java
[java]         at java.net.URLClassLoader.access$000(URLClassLoader.java:
[java]         at java.net.URLClassLoader$1.run(URLClassLoader.java:197)
[java]         at java.security.AccessController.doPrivileged(Native Meth
[java]         at java.net.URLClassLoader.findClass(URLClassLoader.java:1
[java]         at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
[java]         at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.jav
[java]         at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
[java]         at org.apache.openjpa.persistence.PersistenceProductDeriva
[java]         at org.apache.openjpa.persistence.PersistenceProductDeriva
```

It turns out this obscure error is because the default Java EE jars do not contain any actual code. To fix it, change your Java EE dependencies from

```
<!-- Java EE libraries -->
<dependency>
  <groupId>javax</groupId>
  <artifactId>javaee-web-api</artifactId>
  <version>6.0</version>
  <scope>provided</scope>
</dependency>
```

to

```
<!-- default Java EE jars don't include code necessary for  
    bytecode enhancement so we use these instead -->  
<dependency>  
    <groupId>org.jboss.spec</groupId>  
    <artifactId>jboss-javaee-6.0</artifactId>  
    <version>1.0.0.Final</version>  
    <type>pom</type>  
    <scope>provided</scope>  
</dependency>
```

You will also need to add the JBoss repository to your pom.xml if you aren't already using it:

```
<repositories>  
    <repository>  
        <id>repository.jboss.org-public</id>  
        <name>JBoss repository</name>  
        <url>https://repository.jboss.org/nexus/content/groups/public</url>  
    </repository>  
</repositories>
```

[4 comments, post a comment!](#)

[How To Move A Node In Nested Sets With SQL](#)

By [Roger Keays](#), 8 January 2012

Moving nodes in nested sets is a complex operation. There were a few solutions on [Stack Overflow](#) for moving a node under a given parent, however this doesn't let you select the position of the node amongst its siblings.

This solution lets you move a node to any position in the tree, with just a single input parameter - the new left position (newpos) of the node.

Fundamentally there are three steps:

1. Create new space for the subtree.
2. Move the subtree into this space.
3. Remove the old space vacated by the subtree.



In psuedo-sql, it looks like this:

```
/**
 * -- create new space for subtree
 * UPDATEtags SETlpos = lpos + :width WHERE lpos >= :newpos
 * UPDATEtags SETrpos = rpos + :width WHERE rpos >= :newpos
 *
 * -- move subtree into new space
 * UPDATE tags SET lpos = lpos + :distance, rpos = rpos + :distance
 *         WHERE lpos >= :tmppos AND rpos < :tmppos + :width
 *
 * -- remove old space vacated by subtree
 * UPDATEtags SETlpos = lpos - :width WHERE lpos > :oldrpos
 * UPDATEtags SET rpos = rpos - :width WHERE rpos > :oldrpos
 */
```

[Read more...](#)

[21 comments, post a comment!](#)

Deleting Duplicate Database Rows

By [Roger Keays](#), 13 July 2008

Coming up with a query to find duplicate rows in a database is not as easy as it should be and always turns out dog slow. Here's a simple alternative which I used when I had to do just that:

- Dump the database.
- Delete all the records.
- Add a UNIQUE constraint on the natural key.
- Reload the records from the database dump.

If you have duplicates in the dump, the INSERT will fail and they won't make it back into the database.

Easy or WHAT?!

[No comments yet, be the first to comment!](#)

Converting Unsigned Bigints to Signed in PostgreSQL

By [Roger Keays](#), 24 May 2008

Just upgrading Dspam and discovered that the new version uses (or recommends) a BIGINT column for tokens instead of NUMERIC(20) on postgresql for better performance. Dspam tokens are unsigned 64 bit values, but postgresql's BIGINTs are signed 64 bit values. The new dspam just

marshals back and forth between signed and unsigned values which avoids the costly NUMERIC data type.

The problem is, of course that all our data is stored as unsigned NUMERIC values. Dspam has a program to convert the database, `dspam_pg2int8`, but it gave me a segmentation fault :(. Anyway, I figured out that it's not too hard to do yourself in SQL and came up with this script, which presumes postgres is using two's complement for negative values:

```
begin;
alter table dspam_token_data rename token to token2;
alter table dspam_token_data add column token bigint;
update dspam_token_data set token =
    case when token2 < (2 ^ 63)
        then token2
        else token2 - (2 ^ 64)
    end;
alter table dspam_token_data drop column token2;
commit;
```

Hope you find it helpful.

[1 comment, post a comment!](#)

[javax.naming.NameNotFoundException: Name jdbc is not bound in this Context](#)

By [Roger Keays](#), 15 April 2008

It seems that Tomcat 5.5 and 6.0 DataSourceRealms don't like referring to JNDI resources declared in the same context. The following configuration gives me a `javax.naming`.

`NameNotFoundException`:

```
<Context path="/">
```

```
<!-- connection pool for database -->
<Resource name="jdbc/data" auth="Container" type="javax.sql.DataSource"
    driverClassName="org.postgresql.Driver"
    url="jdbc:postgresql://localhost/test"
    username="postgres" password=""
    maxActive="5" maxIdle="2" maxWait="-1"/>
```

```
<!-- authentication realm -->
<Realm className="org.apache.catalina.realm.DataSourceRealm"
```

```
        dataSourceName="jdbc/data" digest="MD5"
        userTable="web_users" userRoleTable="web_user_roles"
        userNameCol="id" userCredCol="password" roleNameCol="role"/>
</Context>
```

However, if I move the JNDI resource up it works as expected:

```
<GlobalNamingResources>

    <!-- connection pool for database -->
    <Resource name="jdbc/data" auth="Container" type="javax.sql.DataSource"
        driverClassName="org.postgresql.Driver"
        url="jdbc:postgresql://localhost/test"
        username="postgres" password=""
        maxActive="5" maxIdle="2" maxWait="-1"/>
</GlobalNamingResources>

...
<Context path="/">

    <!-- authentication realm -->
    <Realm className="org.apache.catalina.realm.DataSourceRealm"
        dataSourceName="jdbc/data" digest="MD5"
        userTable="web_users" userRoleTable="web_user_roles"
        userNameCol="id" userCredCol="password" roleNameCol="role"/>
</Context>
```

Odd...

[2 comments, post a comment!](#)

Drop-and-create-tables in OpenJPA

By [Roger Keays](#), 6 March 2008

Toplink had a useful drop-and-create-tables options which made unit testing a lot easier because you always new you were testing a fresh database. Well, I've been trying to do the same thing in OpenJPA and couldn't figure it out until I found this JIRA issue: [OPENJPA-94](#).

The nearest equivalent in OpenJPA is the following obscure setting:

```
openjpa.jdbc.SynchronizeMappings=buildSchema(SchemaAction='add,deleteTableCont
```

It is actually faster than the Toplink method because the tables aren't recreated, they are just updated and emptied. It'd just be nice if it had a sensible name, and appeared in the docs somewhere!

[No comments yet, be the first to comment!](#)

JPA 2.0 Feature Requests

By [Roger Keays](#), 7 August 2007

I've just come across a [blog by David Van Couvering enumerating the plans for JPA 2.0](#) and was pretty disappointed that it didn't really include any of the features I've been hanging out for in JPA. Fortunately there was a feedback address: persistence-feature-requests@sun.com, so I hastily collated my ideas and sent them there. Will it make any difference? Good question.

Here is my list of feature requests which I sent to that address.

[Read more...](#)

[No comments yet, be the first to comment!](#)

The Only DAO You'll Ever Need

By [Roger Keays](#), 29 May 2007

The Data Access Object (DAO) is a pattern which, I believe, is way overused. Basically the idea is to delegate all your persistence operations to a wrapper object to make it easier to change persistence technologies. In a webapp the DAO serves the dual purpose of being a container for objects and persistence operations which can be invoked on them. For this reason, people often extend the pattern to have one DAO per entity, so `Customer` has a `CustomerDAO`, `Product` a `ProductDAO` etc etc etc.

This seemed like overkill to me. What I really wanted is to be able to write a webapp which consisted (in it's entirety) only of entity classes and CRUD templates for those entities.

[Read more...](#)

[No comments yet, be the first to comment!](#)

< Prev1 [2](#) [Next](#) >