VirtualPersistAPI Entity Relationships

by Paul Mitchum, April 23, 2012

A Markdown version of this document exists here: https://github.com/paul-m/VirtualPersistAPI/blob/master/design/EntityRelationships.md

Currently...

Currently there are three entities and a subsequent join table.

These are defined in PHP annotation using Doctrine ORM: http://docs.doctrine-project.org/projects/doctrine-orm/en/latest/index.html

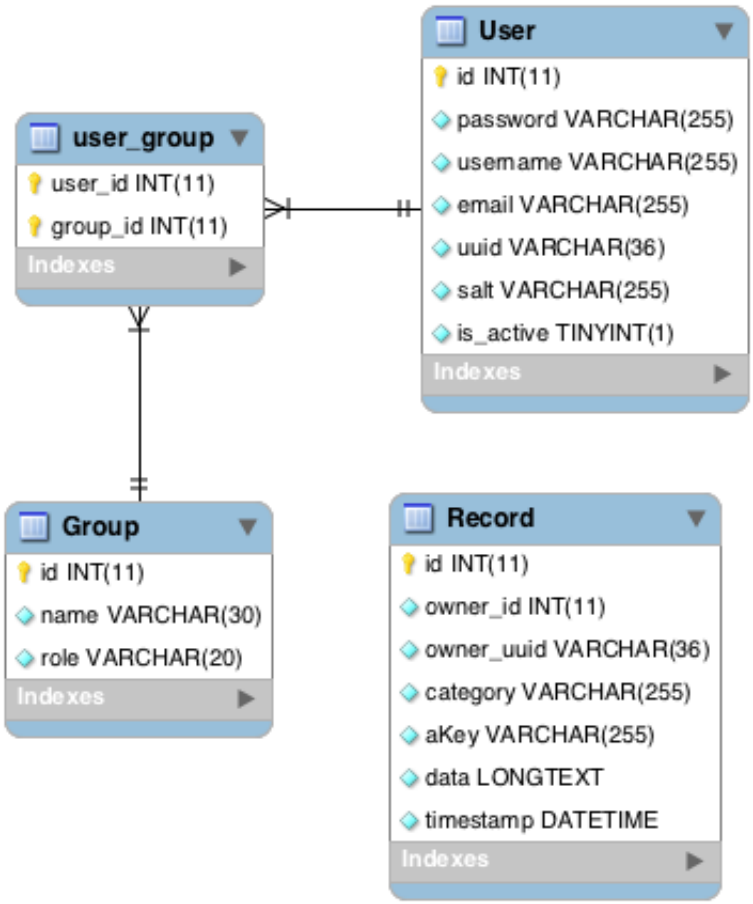
Using Symfony's console, we can generate the SQL necessary to create this schema on the database:

```
CREATE TABLE `Group` (id INT AUTO_INCREMENT NOT NULL, name VARCHAR(30) N
OT NULL, role VARCHAR(20) NOT NULL, UNIQUE INDEX UNIQ_AC016BC157698A6A (
role), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_
ci ENGINE = InnoDB;

CREATE TABLE Record (id INT AUTO_INCREMENT NOT NULL, owner_id INT NOT NU
LL, owner_uuid VARCHAR(36) NOT NULL, category VARCHAR(255) NOT NULL, aKe
y VARCHAR(255) NOT NULL, data LONGTEXT NOT NULL, timestamp DATETIME NOT
NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_c
i ENGINE = InnoDB;

CREATE TABLE User (id INT AUTO_INCREMENT NOT NULL, password VARCHAR(255)
 NOT NULL, username VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL,
uuid VARCHAR(36) NOT NULL, salt VARCHAR(255) NOT NULL, is_active TINYINT
(1) NOT NULL, UNIQUE INDEX user_uuid (uuid), PRIMARY KEY(id)) DEFAULT CH
ARACTER SET utf8 COLLATE utf8_unicode_ci ENGINE = InnoDB;

CREATE TABLE user_group (user_id INT NOT NULL, group_id INT NOT NULL, IN
DEX IDX_8F02BF9DA76ED395 (user_id), INDEX IDX_8F02BF9DFE54D947 (group_id
), PRIMARY KEY(user_id, group_id)) DEFAULT CHARACTER SET utf8 COLLATE ut
f8_unicode_ci ENGINE = InnoDB;

ALTER TABLE user_group ADD CONSTRAINT FK_8F02BF9DA76ED395 FOREIGN KEY (u
ser_id) REFERENCES User (id) ON DELETE CASCADE;

ALTER TABLE user_group ADD CONSTRAINT FK_8F02BF9DFE54D947 FOREIGN KEY (g
roup_id) REFERENCES `Group` (id) ON DELETE CASCADE;
```

The main difference between this and the previous version is the addition of the `Group` entity and table, which represents access control groups. This table is joined to the `User` table through the `user_group` table, which is helpfully generated by Doctrine thanks to `@ManyToMany` annotation.

You can see the evidence of this in the semi-random constraint names, such as `UNIQ_AC016BC157698A6A` and `FK_8F02BF9DA76ED395`.

This SQL results in an ERD such as this:


User
id INT(11)
password VARCHAR(255)
username VARCHAR(255)
email VARCHAR(255)
uuid VARCHAR(36)
salt VARCHAR(255)
is_active TINYINT(1)
Indexes
user_group
user_id INT(11)
group_id INT(11)
Indexes
Group
id INT(11)
name VARCHAR(30)
role VARCHAR(20)
Indexes
Record
id INT(11)
owner_id INT(11)
owner_uuid VARCHAR(36)
category VARCHAR(255)
aKey VARCHAR(255)
data LONGTEXT
timestamp DATETIME
Indexes