27.3.2015 - Exercises Chapter 10
JDBC

A prerequisite for this chapter is to have a database accessible, but the goal is not to learn how to install and configure a relational database (although you can try) the most convenient method is to use one MySql dockerized image as per the following command:

$ docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=password -d mysql

This command starts up a docker container with MySql ready, listening in port 3306 with **root/password** as credentials.

The recommended JDBC (type 4) driver for the exercises is the following:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.34</version>
</dependency>
```

1. Write an app that interrogates the whole database server and provides the user useful information as:

- List of existing schemas (aka catalogs)
- List of tables inside each schema.

2. Based on the previous exercise, now create a **convention over configuration**, that is, by default the application will try to find a file in the **classpath** called **database.yaml** in order to get the connection information as:

- host
- port
- database
- user
- password

this behavior will be overridden if and only if the user is passing parameters to the command line as

$java **DatabaseInfo** host port database user password

Tip: You can use a 3rd party library for parsing the yaml file.

3. Given the class **EngineersInsert and the DDL database.sql**, write an integration Test that validates its functionality. Remember that the test must be repeatable. (same input = same output for all the automatic executions)

4. Based on the feedback provided by the test just written, explain and fix if necessary why the test passes or why the test fails.

5. Once we have engineers inserted in the table, how can you know the size of a **ResultSet** returned by a query that selects all of them?