

# Visual Database Modelling

## About

This project delivers a solution that is engine independent for the two most used open source databases: MySQL and PostgreSQL. The main functionalities of the final release must provide a UML-like graphical interface and also cover the incompatibilities between the two engines. It must also provide a versioning system of the schema - it must allow comparing among various schemas and various versions of the same schema .

## User Stories

A user can:

|              |   |
|--------------|---|
| Main stories | Open Existing Schema (Internal Representation)  |
|              | Import an existing database (PostgreSQL and MySQL) using credentials like: hostname, port, username, pass, dbname, etc. He will be prompted to save the connection with a given name  |
|              | He can open existing schema. A popup window to select the required file will appear on the screen. By default the path will be set to the default working directory (USER_HOME/Application/).   |
|              | By default the acceptable extensions will be set to the application's internal representation file extension. On success the schema will be displayed. On failure a message with the current error will be displayed (Ex: File format not recognized, corrupt file, etc.) |
|              | Upon closing the application, the User is prompted to save (if ANYTHING has changed).   |
|              | Can import a Liquibase file and update the database according to that   |
|              | View database properties: name, OID(object ID), owner, tablespaces, default tablespace, character encoding, schema, allow/don't allow connections, status, connection limit   |
|              | Drop/create new database  |
|              | View database statistics  |
|              | View database dependencies and dependents   |
|              | View database catalogs  |
|              | View/add/edit/delete database extensions, schema and Slony replication master or slave  |
|              | View tablespaces and their properties(name, OID, size, owner, location, ACL-access control and any other descriptions/comments) and dependencies/dependents   |
|              | View/edit group roles an login roles  |
|              | Refresh something selected in the interface   |
|              | Open a console to execute pgSQL commands  |
|              | Open an embedded text editor for PL/pgSQL   |
|              | He can import a PostgreSQL database and export it to MySQL.   |
|              | Can generate reports with all the statistics of a schema.   |
|              | Can export the schema in UML-like in PDF format   |
|              | Can have access to all the main functionalities from the CLI.   |
|              | As a programmer I would like to be able to do different actions on a schema file via CLI (useful for scripting).  |
|              | Can view system versioning  |

|                    |  |
|--------------------|--|
| Versioning stories | Can open a UML-like representation widow and edit the schema, then save all the changes with a new revision number in the versioning system.   |
|                    | Can open the versioning system tab and view all the revisions that have been made on a schema.   |
|                    | Can compare two revisions of the same schema and view the changes that were made on it. He can also revert to an older revision.   |
|                    | Can compare two different schemas and view the extra indexes, different column, extra tables, different column types, different encoding, etc.   |
|                    | After he compares two different schemas, he can merge them if the architecture allows this.  |
|                    | If he works in the console and types “commit” or just presses the “commit” button, the system will compare the modifications with those from the last revision, and if there are no conflicts, the changes will be committed with a new revision ID and the user’s comment.  |
|                    | As a User I would like to save my schema without committing my changes.  |
|                    | He can update the schema to the changes from the last revision   |
|                    | After pressing the “SYNC” button, the local schema is synchronized with the remote database.<br>When synchronizing, the Schema will be merged with the one in the DB if possible.<br>As a User when synchronizing I want to be able to choose the modification I want to keep: from the actual DB or from my schema. |
|                    | If he has DBA rights over a schema he can choose to receive an email with the daily report changes.  |

### *Similar applications on the market:*

|                             |   |
|-----------------------------|---|
| SQL Compare 10              | <a href="http://www.red-gate.com/products/sql-development/sql-compare/">http://www.red-gate.com/products/sql-development/sql-compare/</a> |
| DB Ghost                    | <a href="http://www.innovartis.co.uk/home.aspx">http://www.innovartis.co.uk/home.aspx</a>   |
| SQL comparison toolset v3.0 | <a href="http://www.idera.com/SQL-Toolbox/SQL-comparison-toolset/">http://www.idera.com/SQL-Toolbox/SQL-comparison-toolset/</a>           |
| SQL Accord                  | <a href="http://www.sqleffects.com/">http://www.sqleffects.com/</a>   |