SQLEditor 3.0

User Guide

MalcolmHardie Solutions Limited

About This Document

The SQLEditor user guide was created by Angus Hardie

Icons

In this document the following icons are used to highlight special sections

Warning Icon
This icon is used where an important warning is given.

Information Icon
This icon is used to provide additional information about a topic

Tip Icon
This icon is used to give a short cut or hint on performing specific actions

Suggestions, Comments and Errata

We welcome suggestions, comments and errata for this and all of our products.

Please send email to **support@malcolmhardie.com**

Visit our website at **http://www.malcolmhardie.com**

Or write to us at

MalcolmHardie Solutions Ltd
Office 22
196 Rose Street
Edinburgh
EH2 4AT
Scotland

Revision History

June 2009	Start of revisions for 1.6 Release
November 2009	First public release
January 2012	First Revision for v2.0
June 2012	Further revisions for v2.0
July, August 2012	Final release for v2.0
April,May,June 2013	Updates for 2.1 and 2.2
October 2015	Updates for v3.0

Introduction

SQLEditor For Mac OS X is a program to help you work with databases. Relational databases are the main type of database that SQLEditor supports, however other types can also be used.

SQLEditor allows you to create, import, view, edit and export database structures. You can use it to create diagrams of databases showing the logical layout of objects within the database. You can save images of databases to use on webpages or in documents. You can also import and view the structure of live databases via JDBC

This guide describes SQLEditor 3.0 which is the third major release version for SQLEditor. It differs in some ways from version 2.0, but if you're familar with earlier versions most things should be fairly similar.

SQLEditor is a powerful tool that allows you to do lots of things. However like any tool it is possible to do bad things as well as good. Please take care when using SQLEditor, because incorrect operation may cause serious damage to your database and the data contained in it.

Make sure you have lots of backups of your database and its data; and please don't ignore the warnings in this manual and in SQLEditor itself.

What's new in SQLEditor 3

Single Window Design

SQLEditor now uses a single window design. We've merged most of panels and windows together, which reduces clutter, and streamlines your work.

We've also added a new status panel, so you can see what actions have been performed and view the related log information.

Javascript Plugin Support

SQLEditor now supports writing export plugins using 100% javascript, no native coding required

Object styling

SQLEditor now allows you to change the fonts used for drawing the diagrams

Expanded scripting support

You can now script editing of most types of object using Applescript or Javascript (With Javascript For Automation).

Rewritten engine

SQLEditor 3 has been significantly rewritten to improve performance and stability.

This includes support for larger documents and more complicated structures

One of our test cases has 10,000 fields and 9,900 foreign keys across 100 tables.

We've also made internal improvements which should allow further performance improvements in future releases

Report generation

SQLEditor can now export a HTML report on the document. It lists each table, column, foreign key, index and view along with all comments.

Choose Export File, then HTML Report

Improved tree view

You can now drag objects within the tree view, giving better control over large documents

Contents

Contents

Contents

Indices and Appendices   111

Getting Started

System Requirements

Minimum System Version

To use SQLEditor at all you will need:

- An Apple Mac Computer running Mac OS X 10.9 or later
- Mac OS X 10.10 or later recommended
- Some optional components or plugins may have higher system requirements, you should read the documentation to find out what these requirements are.
- SQLEditor requires 64 bit mode

Database Drivers

SQLEditor connects to a particular database system using a small piece of software called a driver. If you want to use the database import and export functions in SQLEditor then you will need the correct driver for your database.

> ⓘ SQLEditor includes built-in drivers for some databases and can use **JDBC** drivers if they are installed

Please see the chapter on *database drivers* (page 48) for more details.

Installing SQLEditor

SQLEditor is distributed either as a compressed ZIP format archive, as a downloadable DMG format disk image or on physical media like an optical disk or a USB stick.

Installing from a compressed ZIP archive

If you have downloaded the SQLEditor ZIP archive from our website you should follow these instructions to install the application.

1. Locate the downloaded ZIP archive on your computer
2. double click the archive zip file to open it
3. If you see a security message asking you if you want to open the zip file confirm that the disk image is the correct one that was downloaded, then click continue
4. The archive should decompress and the SQLEditor.app application will appear next to the archive
5. Drag the SQLEditor application to the Applications folder on your Mac.

Installing from a downloadable disk image

If you have downloaded the SQLEditor disk image from our website you should follow these instructions to install the application.

1. Locate the disk image on your computer
2. double click the disk image to open it
3. If you see a security message asking you if you want to open the disk image
4. confirm that the disk image is the correct one that was downloaded
5. click continue
6. Drag the SQLEditor application from the install disk window to the Applications folder on your Mac.

Installing from a physical CD-ROM

If you have a physical CD-ROM containing SQLEditor you should follow these instructions to install the application.

1. Insert the CD-ROM into your computer
2. Drag the SQLEditor application from the install disk window to the Applications folder on your Mac.

Where to Install SQLEditor

We recommend placing SQLEditor in the **/Applications** folder. But you can use any other folder if you prefer. You can also move it after installation if you want to.

When you use SQLEditor it will create preference and settings files in the following locations.

- **~/Library/Application/Support/SQLEditor/**
- **~/Library/Preferences/**

Uninstalling

To uninstall SQLEditor you should drag the application icon to the trash.

This is normally sufficient, however the following files and directories are used by SQLEditor, so you may wish to delete them too

- **/Applications/SQLEditor.app** (The application itself)

- **~/Library/Application/Support/SQLEditor/**
- **~/Library/Preferences/com.malcolmhardie.sqleditor.cocoa.plist**

There may also be entries in your keychain for the databases that you access. You may wish to keep these if other applications are using them too.

Automatic updating

When you first start SQLEditor you will be offered the **option** to use the automatic update feature. This uses the Sparkle update framework[1].

Automatic update can also be enabled or disabled from the **Preferences Window**.

If you enable automatic updates then SQLEditor will check our website to see if a new version is available.

If an update is available then a window will appear on your screen offering the option to upgrade to the new version. The updated version will be downloaded and installed.

If you don't wish to upgrade to the new version at all then you should select the **Skip This Version** button. You will not be prompted to upgrade to the specified version again unless you choose **SQLEditor→Check for Upgrades...**.

If you want to upgrade at a later time then you can click the **Remind Me Later** button and the update notice will appear on the next restart of SQLEditor.

There is also an option to send some profile information about your system when you check for updates. This is used for market research to identify which versions are in use and what types of environment SQLEditor is running in. You can disable this option in the preferences window.

> (i) Other than the system profile information (if enabled) we don't deliberately collect personal data about you or your mac, however just sending a request will identify you by IP address and other data may, in theory, be sufficient to identify you if you have a unique configuration.
>
> **If you don't want this to happen then please don't use automatic updates.**

---

[1]http://sparkle.andymatuschak.org/

Tutorial: Creating a simple database design

In this tutorial we will introduce SQLEditor, explain how to create a simple database design and then export it to a SQL text file for later use.

In our simple example we are going to create a simple database representing a library. The library will keep records of books and borrowers.

Creating a new Document

1. The first step is to open the SQLEditor application. You should see a new untitled document window appear on your screen.

2. If you don't see a new document window use **File→New**

The document window is where you view and edit your design.

The big section in the middle is the canvas, where the diagram is actually drawn.

To the right is the **Inspector Panel**, which allows you to edit the objects you create

The left side has the source view, which displays a live view of the SQL source for the diagram

> If you don't see the inspector panel visible use **Window→Object Sidebar→Show Inspector**
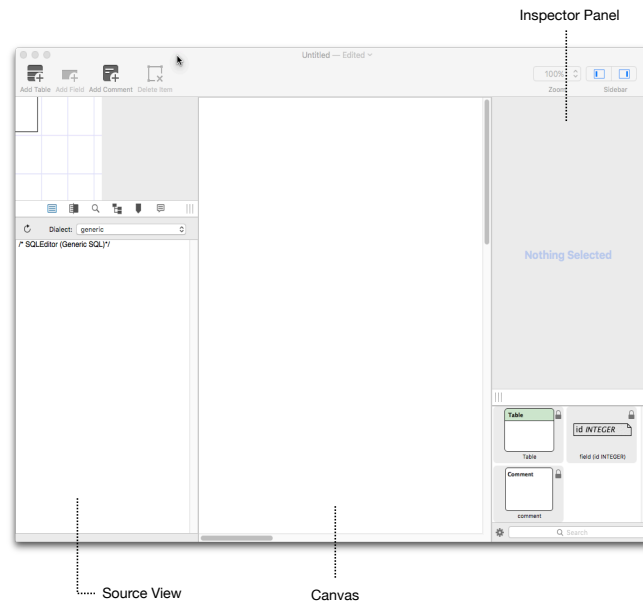
Figure 1: The main window

Adding a new Table

1 Click the **Add Table** button on the **toolbar**

A new Table object will appear in the window and the inspector will change to show information about this new table.
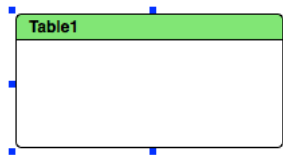
Figure 2: The first table

Most likely the new table will be called **Table1**. But we're going to change it to be **books**.

You can move the table around by dragging the colored title bar

Renaming the Table

1 Make sure the table is selected by clicking once on the colored title bar at the top of the table.
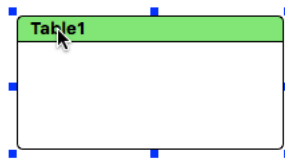
2 Then move to the inspector and click once in the **Name:** field where it says **Table1**.
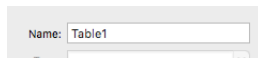
3 Edit the name of the table to read **books**
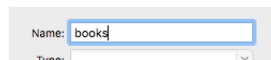
4 When you've finished editing the name hit the **return** key.

5 In the main window the table's name will change.
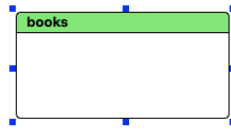
Adding a field to the table

Next we're going to add some fields to this table. Fields normally have the name **id** when created. Fields have to have unique names so SQLEditor will automatically rename fields by adding a number (id_1, id_2, id_3, etc)

1 Select the **books** table by clicking on the colored title bar at the top

2 Click the **Add Field** button in the **toolbar**

3 A new field will appear in the table

4 Click on the field to select it

5 Then move to the inspector on click once in the **Name:** field where it says **id**.

6 rename the field to **book_id**

7 The look at the **Type:** popup box. Select the type **INTEGER** from the popup. (It may already be selected)

Make one of the fields into a primary key

The **book_id** field is going to used as the primary key for the books table.

1 Click on the **book_id** field in the books table to select it.

2 Look at the inspector, at the bottom is a tabbed panel. Make sure the tab **Options** is visible. Click the tab if it isn't visible

3 Look for the checkbox **Primary Key** and check it

In the main window a **P** icon will appear on the field. This indicates that it is used as a primary key.

Create more fields

1 Add fields to the table in the same way until it looks like this

books	
book_id INTEGER	
title VARCHAR(255)	
author VARCHAR(255)	

Enter the numerical size of the field type into the **size/members** field in the inspector window.

Create more tables

1 Now create another table for the borrowers

borrower	
borrower_id INTEGER	
surname VARCHAR(255)	
firstname VARCHAR(255)	

2 Then create a third table for loans, this table will link the borrowers to the books

loans	
loan_id INTEGER	
book_id INTEGER	
borrower_id INTEGER	

3 Your diagram should look something like this

Don't worry if your tables aren't laid out exactly like this, the exact layout isn't important, as long as you can see all of the tables easily. Remember that you can drag the tables around by the colored titlebar if you want to rearrange them.

books

book_id INTEGER		Ⓟ
title VARCHAR(255)		
author VARCHAR(255)		

loans

loan_id INTEGER		Ⓟ
book_id INTEGER		
borrower_id INTEGER		

borrower

borrower_id INTEGER		Ⓟ
surname VARCHAR(255)		
firstname VARCHAR(255)		

Making foreign key relations

The next step is to link the tables together with foreign key relationships.

1 Look for the tables **books** and **loans**. Move them until they are both visible and do not overlap.

2 Using the mouse drag from the middle of the **book_id** field in the loans table to the **book_id** field in the **books** table.

Do not start the mouse drag inside the gray control handle on the the left side of the field, because that will move the whole field instead
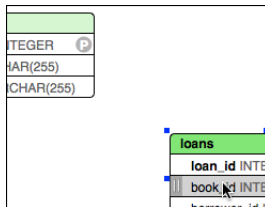
3 As soon as you start dragging a line will appear. Move the mouse pointer up towards the the **books_id** field in the **books** table.
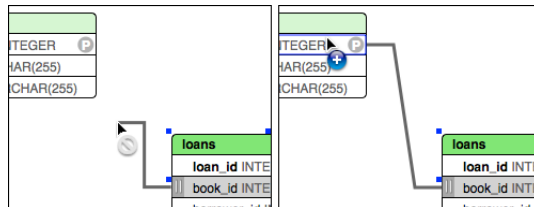
4 When you reach the **books_id** field release the mouse. The line should remain visible between the two fields

5 If the line disappears when you release the mouse, try again and make sure you don't release the mouse until you pointer is directly over the target field

6 You should see a icon **F** appear on the **loans(book_id)** field. This indicates that it is a foreign key and references another field elsewhere.

7 Next make a foreign key relationship between **loans(borrow_id)** and **borrower(borrower_id)**

Your diagram should now look like this:
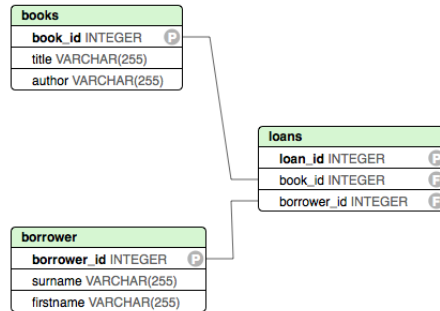
Exporting to SQL

1 Choose **File→Export to File ...**

2 In the save panel that appears choose a file name (for example **database.sql**).

3 Click Save

The file extension will normally be **.sql**

> SQLEditor associates itself with the .sql file extension, so you may need to open the saved file from within your text editor.

If you open the file using Text Edit (or your favourite text editor) you can then see the contents should look something like this:

```
/* SQLEditor (Generic SQL)*/

CREATE TABLE books
(
book_id INTEGER,
title VARCHAR(255),
author VARCHAR(255),
PRIMARY KEY (book_id)
);

CREATE TABLE borrower
(
borrower_id INTEGER,
surname VARCHAR(255),
firstname VARCHAR(255),
PRIMARY KEY (borrower_id)
);

CREATE TABLE loans
(
loan_id INTEGER,
book_id INTEGER,
borrower_id INTEGER,
PRIMARY KEY (loan_id)
);
```

```
ALTER TABLE loans ADD FOREIGN KEY (book_id) REFERENCES
    books (book_id);

ALTER TABLE loans ADD FOREIGN KEY (borrower_id)
    REFERENCES borrower (borrower_id);
```

Reference

Common Tasks

This section gives step-by-step instructions to perform some of the more frequently performed tasks in SQLEditor.

You may also wish to see the sections *Creating and Editing Objects* on page 40 and *Types of Object* on page 26

Create a new document

1 Choose **File→New**

If you want to select the document dialect you can choose **File→Document with Dialect** instead

Add a table

1 Choose **Object→Add Table**

The new table will be added to the **Canvas** and the **Inspector** will display information about it.

2 Click on the **Name field** in the **Inspector** and give the new table a name.

You can also set other options at this stage

You can drag a table object from the palette instead of using the menu.

Add a field to a table

1 Click on the table you want.

2 Choose **Object→Add Field**

A new field will be added to the table representing the SQL table column.

The Inspector will display information about the new column.

3 In the Inspector click on the name field and give the new column a name.

You can also drag a field object from the palette to the table or use the add field button in the table inspector

Create a foreign key relation

1 Find the two fields that you want to create the relationship between.

2 Drag from the field you want to make a foreign key to the field that you want to reference.

3 The F icon will appear on the field set to be a foreign key

Create a multi-column Foreign Key

1 Click on the table you want to add the foreign key to

2 Choose **Object→Add Foreign Key**. A new foreign key entry will appear in the table

3 Drag from the new foreign key entry to the target (referenced) table

4 Move to the inspector and look at the list of column pairs

5 Click the **+** (plus) button to add a new column pair

6 Choose the columns in the source and target tables that you wish to pair

7 Add further column pairs if required

8 Specify a constraint name for the foreign key (if required)

You can drag the columns to reorder the way they appear in the index

Create a multi-column Index

1 Click on the table you want to add the index to

2 Choose **Object→Add Index**. A new index entry will appear in the table

3 Move to the inspector and look at the list of columns

4 Tick the checkbox for each column that you wish to include

You can drag the columns to reorder the way they appear in the index

Create a View

1 Choose **Object→Add View**

The new view will be added to the **Canvas** and the **Inspector** will display information about it.

2 Click on the **name field** in the **Inspector** and give the new view a name.

3 click on the **view sql** in the **Inspector** then enter the SQL defining the view

Add a document comment to the document

1 Choose **Object→Add comment**

2 In the inspector click on the comment input field to enter the text of the comment.

As well as document comments you can also have comments for specific tables or fields You add these using the inspector

Export to SQL DDL Schema Text File

SQLEditor can export your database design to SQL Data description language (Schema Text) in a number of different formats. You should select the export dialect of SQL that most closely matches the database system that you are using.

1 Identify the database system that you are using. E.g MySQL, Postgres

2 Choose **File→Export As SQL**

3 Using the popup menu select the export dialect that you wish to use

4 Choose a name for the export file

5 Click the Save button

If you have previously selected a dialect for your document then SQLEditor should select this automatically in the popup box.

Export selected objects

If you just want to export a few objects at a time you can use copy and paste.

1 Select the objects you want to export

2 Choose **Edit→Copy**

3 Paste the objects into another application

What will I get when I use paste in the other application?

- If the other application accepts text then you should get the SQL representation of the object as text.
- If the other application supports SQLEditor object format then you will get a representation of the objects.

⚠️ The SQLEditor document's dialect is used when exporting objects in this way, you cannot change the dialect during the copying process.

Change export SQL dialect

The document dialect is the default dialect of SQL that is used when SQLEditor exports SQL code. The document dialect is specific to each document and can be changed at any time using the following procedure.

1 Click the **Export Tab**

2 Choose **File→Document Options**

3 Use the Target For SQL popup menu to choose the dialect

4 Click the OK button to close the window. You can also change other SQL Export settings in this window.

Import SQL database description from a text file

To import SQL from a file use the following procedure.

1 Choose **File→Import from File…**

2 Choose the file you want to import

3 Click the Import button

SQLEditor will attempt to import the file that you select.

The Import Status window will appear giving details of any errors that may occur during import.

> **Please report any errors!**
> Reporting parser errors is one of the main ways that the SQL parser is improved.
> You can send them to support@malcolmhardie.com or use the **Help→Send Suggestion** menu option

Set the default SQL dialect for new documents

When SQLEditor creates a new document the dialect for the document is set to the **Default Dialect**

You can change this **Default Dialect** using the following procedure:

1 Choose **SQLEditor→Preferences...**

2 Select the **Export** tab in the **Preferences** window

3 Change the popup menu to the dialect that you want to use

All future documents will now be created with this dialect

> You can specify the dialect of a document by using the **File→New Document with Dialect** command

Types of Object

SQLEditor supports a number of different types of objects. Additional object types may appear in future versions.

It is also possible for plugins to add new object types. You will need to refer the documentation that came with the plugin to find out how such objects might work.

Some objects in SQLEditor can only be used when contained within another object. Fields, Indexes and Constraints can only be added to Table objects for example.

The diagrams in this section are taken from the current release of SQLEditor and use the default appearance.

If you are using have altered the display or appearance settings, or if you are using a different version of SQLEditor, then the appearance may differ.

Tables
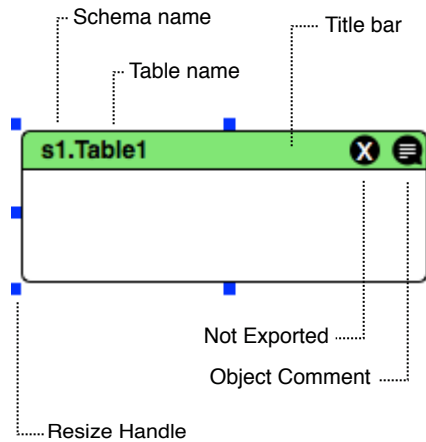
Figure 3: An example table

The table object represents a table in a database. Tables can have various properties that can be edited using the inspector or by direct editing Tables can be assigned to a category, which may be identified by a color. To edit a table (or any object) using direct editing hit the return key and the table name will become editable.

You can add different types of objects to tables. These include fields, indexes and constraints. Other types of object may be added in future versions of SQLEditor.

Figure 3 shows a sample table without any fields.

There is an option to make tables resize automatically to fit the objects that they contained. This option is **View**→**Auto Size Tables**

Fields

Fields in SQLEditor represent columns within database tables. Fields have properties that can be edited using the inspector. They can also be moved between tables by dragging. Fields are always contained within tables except when being moved.

> You can also change the data type and some other settings of a field by using the context menu (Right Mouse Button/Control Click)

Fields support direct editing of the field name.

Columns in databases can have various options and SQLEditor supports most of the common options. Icons appear on fields for each option that is set.

Figure 4 on page 29 shows the example table from the previous section with a single field added to it.

The field shows all of the possible icons that fields can currently have in SQLEditor.

Most fields will not display all of these icons.

You can also add comments for tables and columns

Field Options

Option	Icon	Description
Foreign Key	F	Field is a foreign key and references another field
Primary Key	P	The field is part of a primary key
Default Value	D	The field has a default value set
Not Null	N	The field cannot be Null
Auto Increment	A	Field has an auto incrementing value (in supported databases)
Unsigned	+/-	The field is has the unsigned option set (for MySQL)
Zero Fill	Z	The field is has the Zero Fill option set (for MySQL)
Unique	U	The field has a unique constraint set on it
Indexed	I	The field is indexed
Binary	b	The field type is binary
Time Zone	tz	The field is of type timestamp and has a time zone set.
Field Comment	?	The field has an attached comment
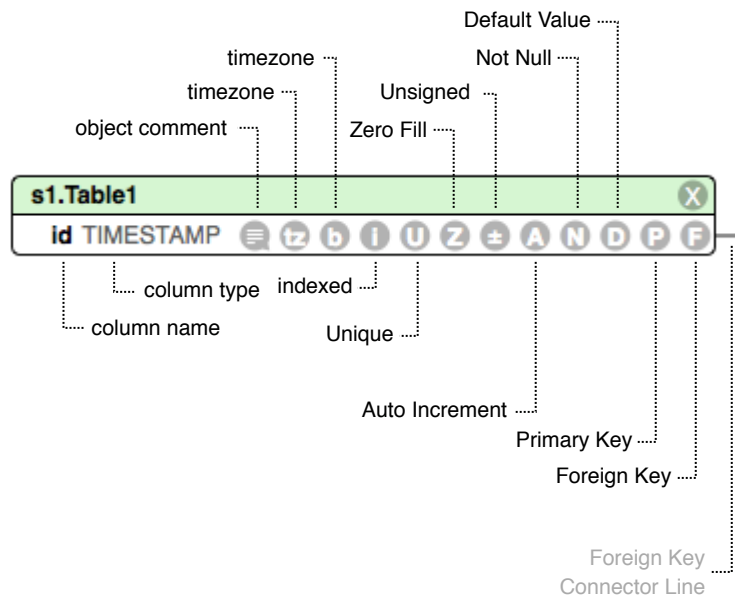
Figure 4: Table and Field layout and icons

Field Inspector

The field inspector allows editing of fields, which in SQLEditor represent table columns

Name   This is the name of the field. SQLEditor may modify names you enter to ensure that the eventual SQL output is valid. In particular, you can't normally have two columns with the same name. If you try, then SQLEditor will add a number suffix to the name to ensure that each object is uniquely named

Type   This popup menu contains a list of the data types that are supported for the current dialect. You can also enter your own types here if you wish. If you enter a type with brackets (e.g VARCHAR(255) then the section within the brackets will be copied to the size/members field

Size/Members   This field is used to specify any parameters of the type. This usually means length, but might also be precision or decimal places displayed. If the type is
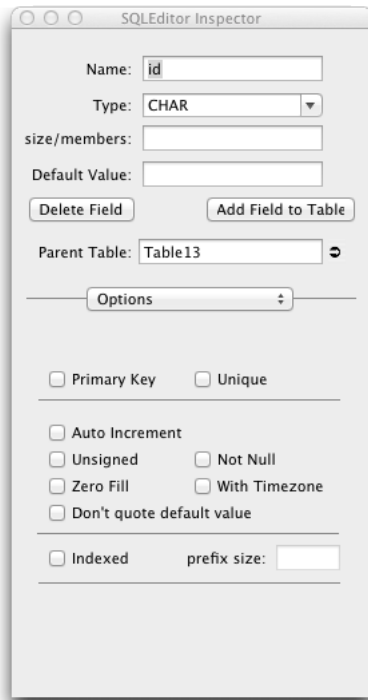
Figure 5: The Field inspector (base view)

a set type it might include the entries in the set. It's normally database specific

Default Value   The default value for the field. SQLEditor will normally try to quote values if appropriate, so you don't need to add quotes. If you want to manually specify the exact value and any quotes yourself, check the **Don't quote default value** option

Delete Field   Delete this field from its parent table

Add Field to Table   Adds another field to this field's parent table. Older versions of SQLEditor call this **Add Sibling**

Parent Table   The name of the parent table for this field. There is a small arrow to the right of the name which, if you click it, will select the parent table and show the inspector for it

Sub-panel menu   The field inspector has more controls than can fit in the window, so they are divided into sub-panels. The subpanel selector menu allows you to change which subpanel is displayed

The lower section of the field inspector has several sub-panels which can be switched between using the popup menu

Options sub-panel

By selecting **Options** from the field inspector sub-panel menu, the **Options sub-panel** is displayed. This allows editing of a field's foreign key settings. This panel allows editing basic settings on fields and is the default sub-panel visible.

Primary Key   Checked if this field should be part of a primary key. Checking this adds the current field to the table primary key, unchecking it will remove this field from the primary key. It won't affect whether other fields are part of the primary key. (It isn't an exclusive operation)

Unique   Checked if this field should have be UNIQUE. The result of this varies between databases, in MySQL it will normally export with the UNIQUE keyword, in other databases it may generate a UNIQUE index

If a field is the referenced target of a foreign key, then it is normally required to be either UNIQUE or be part of a primary key. SQLEditor will normally enforce this using UNIQUE, so this option will be checked and grayed out

Auto Increment   Checked if this field should have an auto-incremented value when a row is added. This varys between databases, in MySQL it adds the AutoIncrement keyword to the export, if the type is appropriate. SQLEditor will normally try not to add this option to the exported SQL unless it makes sense

Unsigned   Checked if this field has a datatype that should be UNSIGNED. This won't export anything unless the base data type supports UNSIGNED

Not Null   Checked if this field should be declared NOT NULL. If it should be nullable then it should be unchecked

Zero Fill   Checked if the field should be specified Zero Fill. (Mainly for MySQL use)

With Timezone   Checked if this field should be a time or date type and should be specified with a time zone. This will have an effect only if the base data type supports time zones

Don't Quote Default Value   If checked, the default type will be exported without adding any quote marks. This allows you to control how the default value is quoted without SQLEditor making changes. If it is unchecked then SQLEditor will quote the default value as appropriate and escape any quote marks that you include.

Indexed   Checked if this field should be indexed. This normally causes a new index to be added to the table at export. (Note that this index is only seen at export, it won't add an index to the parent table in SQLEditor)

prefix size   Allows you specify the size of the index prefix. An index prefix limits the size of an index to improve performance

Foreign key sub-panel

By selecting **Foreign Key** from the inspector panel popup menu, the **foreign key sub-panel** is displayed. This allows editing of a field's foreign key settings. Not all fields have foreign keys set, so the controls may be disabled. Also if you have a set a foreign key using a **table level foreign key object,** this section will also be disabled. (To edit a table level foreign key, click on the foreign key object itself within the table, not the fields that are referenced by it)

Foreign Key specification   The foreign key specification appears in a text field. It is not editable.

Remove   The **Remove** button removes the foreign key from the field
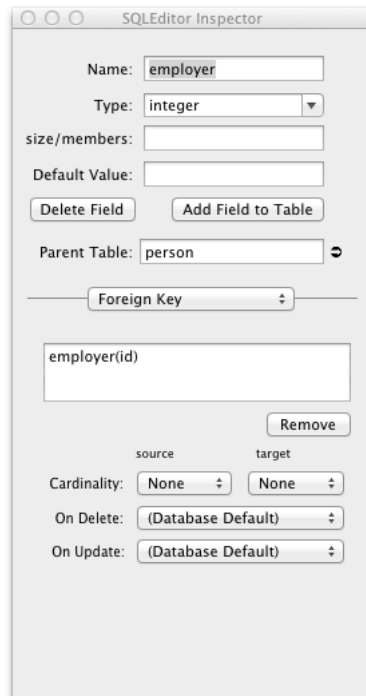
Figure 6: The Field inspector (foreign key view)

Cardinality The two **Cardinality** popup menus allow editing of the realtion cardinality, the source end on the left, the target or destination end on the right. Options are "1", "0 or 1","1+","0,1+" and "Many". Changing the option here changes the diagram, but doesn't affect the SQL output in any current exporter

On Delete The **On Delete** action popup allows you to specify the action that the database takes when values in a foreign key relationship are deleted. Options are currently "Database Default","Cascade","Restrict","Set Null","No Action" and "Set Default". Different exporters will support different variations of this.

On Update The **On Update** action popup allows you to specify the action that the database takes when values in a foreign key relationship are updated. Options are

currently "Database Default","Cascade","Restrict","Set Null","No Action" and "Set Default". Different exporters will support different variations of this.

Note sub-panel

Figure 7: The Field inspector (note view)

By selecting **Note** from the field inspector sub-panel menu, the **Note sub-panel** is displayed. This allows editing of the field object's informational note or comment

The main section of this sub panel is a text field into which you can type. The text is then available to the SQL exporter as the comment text (in MYSQL for example). Some exporters may export this as part of the object SQL, others may include it as a block comment in the exported SQL.
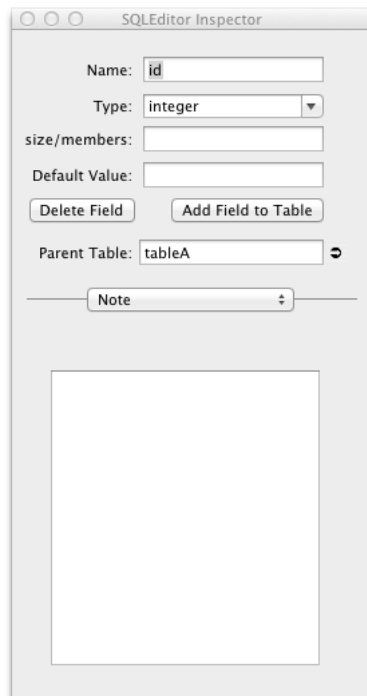
Label sub-panel
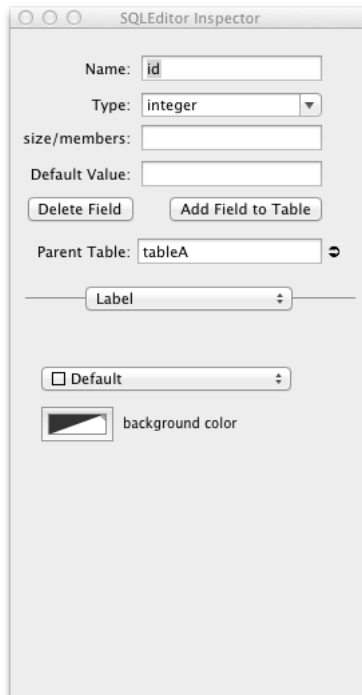
Figure 8: The Field inspector (label view)

By selecting **Label** from the field inspector sub-panel menu, the **Label sub-panel** is displayed. This allows editing of the field object's label

Label popup   Use the popup menu to choose a label to assign to this field, or to create and assign a new label

Background Color   The **background color** well indicates the color that will be used to draw the field. If a label is specified then the color will be the label's color, otherwise you can specify any color you like

Indexes

Figure 9: An Index in a table

An index object represents a table index in a database.

Indexes allow for specifying the type of index (normally a database specific value) and the columns that are included in the index.

The Index Inspector

Name   Using the index inspector you can edit the name of the index using the **Name** field, this is then used in the final SQL. SQLEditor may modify the name, particularly to prevent duplicate index names, but also to remove spaces or invalid characters.

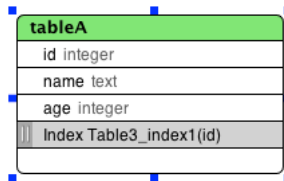**Parent Table**   The **Parent Table** section indicates the table containing the index, clicking the small arrow to the right will select the parent table and deselect this index

Index Type   The **Index Type** field allows you to select the type (e.g UNIQUE, FULLTEXT, SPATIAL) of the index. Note that you can also enter your own types in this field. The effect of the index type field depends on the SQL dialect currently in use.

Indexed Columns   The **Indexed Columns** section of the index inspector allows you to control which columns appear in the index. The table contains all of the columns found in the parent table. To include a particular column in this index, click
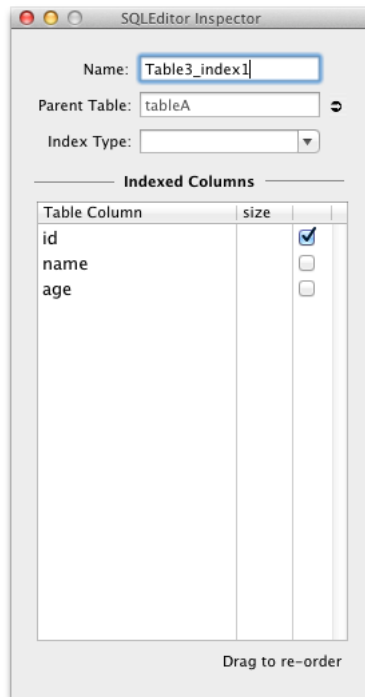
Figure 10: The Index inspector

checkbox next to the name. The entries may rearrange so that columns included in the index appear at the beginning of the list in the correct order.

You can rearrange the order of columns in this index by dragging the entries. Note that this doesn't change the order of the columns in the table itself, this only affects the index column order

Some dialects allow you to specify prefix sizes per column, you can enter these in the **size** column, however this may be ignored by the exporter if the dialect doesn't support this option

Constraints

A constraint object represents a table or column constraint on the database. Currently only **UNIQUE** constraints are supported.

⚠ | Support for constraints is database specific and somewhat variable

Connectors

Connector objects represent foreign key links between fields and tables. Clicking on a connector will cause the Inspector window to show the connector inspector panel which offers options for cardinality, information about source and destination fields and a button to delete the connector.

Comments

A comment object allows you to include extra information about the diagram. Comments are exported to SQL along with tables, although they will normally have no effect on the actual database.

Comments may have different background colors and can have both labels and body text.

Canvas Areas

A canvas area object is used to visually group together other objects. Canvas area objects appear behind all other objects. Canvas objects can have labels and extra information attached to them. The extra information is not displayed (unlike with comment objects which display both pieces of information)

Older versions of SQLEditor used special types of comment object instead of a dedicated canvas area object. To improve compatibility you can convert Comment objects into Canvas Area objects or vice-versa.

Views

A View object is a SQL database construct that combines different data from multiple tables together into a single table-like object. Typically a view is specified using a select statement. In SQLEditor view objects appear much like tables. As you edit the select statement, the tables and fields that the view references will be identified and listed in the view object

Compound Foreign Key

The compound foreign key object is primarily used to create compound foreign key relationships between sets of columns. It can also be used to create simple foreign key relationships that link only two columns together (however it usually easier to use the foreign key option on fields directly in this simple case)

Creating and Editing Objects

This section explains how to create and edit objects in SQLEditor. SQLEditor tries to behave as other Mac OS X programs.

Adding objects

There are several methods of adding objects to a database structure:

- The objects menu contains commands to create all types of object
- The objects can dragged from the objects palette
- The context (right click) menu can be used to create some kinds of object

You can also add extra fields using the **Add Sibling** in the field inspector, and the **add field** button in the table inspector

It is also possible to use a popup menu or copy and paste objects or sql script into the document.

See also: *Adding a table* (page 20) and *Adding a field* (page 21)

Editing Objects

Objects can be selected by clicking on them. Note that selecting a field will also select the parent table.

When an table or comment object is selected then control handles will appear on all sides of the object. These control handles can be used to resize the object. To move the object the title bar can be dragged or the cursor keys used. Holding down shift causes the cursor key movement to be faster.
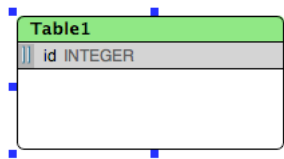

Figure 11: A Selected Field

Field objects do not have control handles, but a drag handle appears when a field is selected. This drag handle can be used to move or copy a field from one table to another.
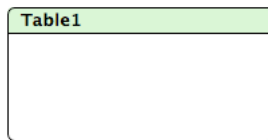
You can copy a field from one table to another by holding down the **Option key** while dragging.

Automatic Renaming

Objects in SQL should normally have unique names. SQLEditor will attempt to prevent you from creating two objects that have the same name.
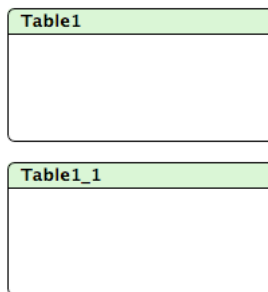
Typically this happens as you add new objects or edit the names of existing objects. The name will change as soon as the object is redrawn.

For example, if there are one table in the document, **Table1**

A new table is added. It will automatically be given the name **Table1_1**

Deleting objects

To delete an object either:

- select it and then choose **edit→delete**
- hit the **Delete** key on the keyboard.
- right click and choose delete object

Copy and paste

You can use copy and paste on table and comment objects. If you copy an object and switch to another application then SQL code will be placed on the clipboard. This is a quick way to export table structures.

If you have sql code in another application you can paste it into SQLEditor and it will be imported into the current structure using the SQL importer

If the SQL contains an error, or if SQLEditor doesn't understand it, then the status window will appear

Dragging and Drop

Most objects in SQLEditor support dragging. Inside the canvas you can move objects by dragging.

If you hold down the **Option key** (alt) then you can create a copy, or drag an object into another application or a different window.
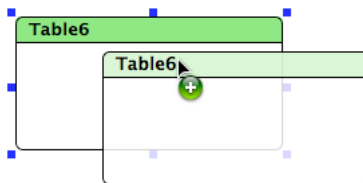

Figure 12: Duplicating a table by option dragging

You can use the objects palette to create new objects on the canvas by dragging the icon representing the object.

Foreign keys

Currently there are two ways of creating foreign keys in SQLEditor. For simple column to column relationships it is easiest to link the fields together directly.

For more complicated configurations including more than one pair of columns you should look at adding a foreign key object.

Using Field level foreign keys

With this method foreign keys are created by dragging from one field to another field. You must ensure that you do not start the drag within the drag handle area of the field. When the drag is complete release the mouse button and a foreign key link will be added.

⚠️ Not all databases support foreign keys and some only offer limited support. For example: MySQL requires that you use the INNODB table type if you want foreign key support.

Foreign key links can be configured to draw in several different styles.

See the *Appearance Preferences Section* on page 93 for more details

ⓘ Note that if you create a foreign key relation, the referenced field must be a unique or primary key.
SQLEditor will automatically set the field to be unique if necessary.

Example

1 In this example we want to create a foreign key on the widget table that references the primary key of the sprocket table.
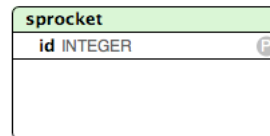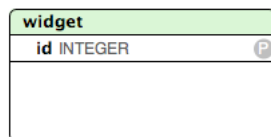
2 Create a new field to be the foreign key
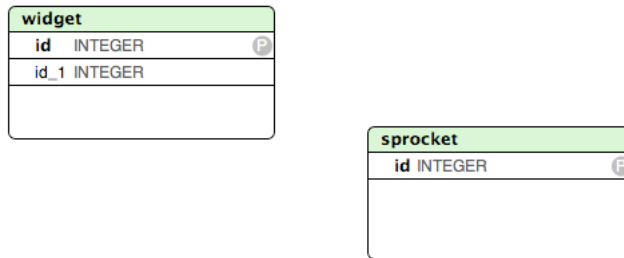
3 Drag from a field on the widget table to the primary key of the sprocket table.
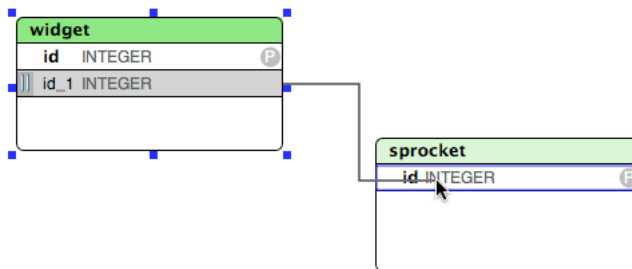
4 The new foreign key relation:
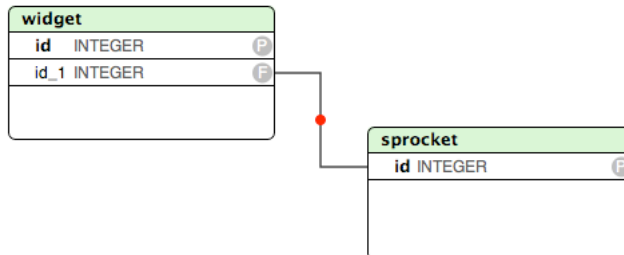
Using a Foreign Key Object

For more complicated foreign key relationships you should use a foreign key object.

1 Select the table you want to link from

2 choose **Object**→**Add Foreign key**

3 Drag from the newly created foreign key object to the target table to create a connection.

4 Click on the foreign key to update the inspector window

5 Add a column pair by clicking the **[+]** button at the bottom of the foreign key inspector.

6 choose a column for source and target in the column pair.

(i) Entries in foreign key objects need both source and target set or they will be ignored.

Databases: Import and Exporting

SQLEditor can import existing database structures into a new document and it can directly create a database structure from a document.

About Database Drivers

SQLEditor needs a database driver specific to the database that you wish to use.

You may also need to build the database with special options for jdbc compatibility or set specific options in the configuration of the database.

> (i) SQLEditor includes built-in drivers for Postgres, MySQL, DrizzleDB and SQLite.
>
> Other databases may require installing a specific driver.

Built-in Database Drivers

SQLEditor includes built-in drivers and support for the following databases. You do not need to install anything for SQLEditor to connect to these databases

- MySQL
- PostgreSQL
- DrizzleDB
- SQLite

If you prefer, you can manually install a JDBC driver for these databases and use that instead.

SQLEditor will auto-detect JDBC drivers for these databases.

Supported Database Drivers

Other drivers must be installed manually.

When SQLEditor is next started it should find the following drivers automatically:

- Oracle
- Filemaker
- OpenBase
- Microsoft SQL Server

Drivers that are not on this list may need be added manually using the Database pane of the Preferences window. See also *Database Preferences* on page 94

The supported list may increase in the future. If the driver you want to use isn't on this list it is worth seeing if SQLEditor has auto-detected it before adding it manually.

Installing Database Drivers

General Instructions

1 Download the correct driver

2 Find the .jar driver file in the download package

3 Place the driver file in **/Library/Java/Extensions**

4 Restart SQLEditor

   You only need to perform the following step if the driver is not visible in the Database Driver List in the Database Preferences pane:

5 Add the database driver to the driver list.

If you prefer you can install drivers in

/Library/Application Support/SQLEditor/JDBC Drivers/

Drivers installed in this location will be available only to SQLEditor. This may be useful if you intend to run other JDBC applications which require incompatible drivers.

MySQL

You can download this from
MySQL Connector J Driver Download Page[2]

How to install MySQL driver

Follow these instructions to install the MySQL JDBC driver.
The file names may vary slightly if you use a different version of the driver.

1 Go to MySQL Connector J Driver Page[3]

2 Download the file **mysql-connector-java-5.1.13.tar.gz**

(Platform Independent Architecture Independent, Compressed TAR Archive)

3 Decompress the file by double clicking

4 Locate the JDBC driver jar file **mysql-connector-java-5.1.13-bin.jar** (inside the decompressed archive)

5 Copy the jar file **mysql-connector-java-5.1.13-bin.jar** to **/Library/Java/Extensions**

6 Restart SQLEditor - SQLEditor should then be able to use the driver correctly

SQLite

Our SQLEditor SQLite Driver Page[4] offers downloads of our recommended driver

- Go to SQLEditor SQLite Driver Page[5]
- Download the binary release
- Copy the downloaded .jar file to **/Library/Java/Extensions**

SQLEditor should find this driver automatically

---

[2]http://dev.mysql.com/downloads/connector/j/5.0.html
[3]http://dev.mysql.com/downloads/connector/j/
[4]http://www.malcolmhardie.com/support/sqleditor/sqlite/index.html
[5]http://www.malcolmhardie.com/support/sqleditor/sqlite/index.html

Oracle

The Oracle JDBC drivers can be downloaded from the Oracle JDBC Driver Page[6].
The download contains several driver jar files. You will need at least a **ojdbc*.jar** file
and a **orai*.jar**. Different configurations of Oracle systems may need different files

We have done much more testing of the 10g driver. The 11i drivers may work, but
only with SQLEditor 1.6 or later.

- Install the JDBC driver in **/Library/Java/Extensions**

Postgresql

First obtain a suitable driver:

- You may need to download the JDBC driver from the Postgresql JDBC site[7]

Then install the driver:

1. Copy the jdbc driver to **/Library/Java/Extensions**
2. Edit **/usr/local/pgsql/data/postgresql.conf** so that **tcpip_socket = true** is set.
   It may be set to false or not be present at all in the default settings

SQLEditor should find this driver automatically

OpenBase

OpenBase uses the URL **com.openbase.jdbc.ObDriver**.

- Install the JDBC driver in **/Library/Java/Extensions**

SQLEditor should find this driver automatically

---

[6]http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/index.html
[7]http://jdbc.postgresql.org/

Unsupported Databases

If a database system has a JDBC driver it should work, although databases that use a java swing or AWT user interface in their JDBC driver will probably not.

Making a JDBC Driver available only to SQLEditor

You can place JDBC database drivers in either:

- **/Library/Application Support/SQLEditor/JDBC Drivers/** - for use only by **SQLEditor only**
- **/Library/Java/Extensions/** - for use by all programs on your machine

The instructions on this page assume that you will use the **/Library/Java/Extensions/** directory.

If you use the SQLEditor specific directory option you will need to subsititute the correct directory path.

The *Database Preferences Pane* (page 94) is used

Importing from a database

The **File→Import From Database...** function will connect to an existing database and extract the structure, it will then create a new document containing that structure.

See also *The Database Import Panel* on page 74

SQLEditor doesn't keep a link between the newly created document and the original database.

To create a new document from an existing database

1 Choose **File→Import From Database...**

2 Enter the connection details into the Import Database Panel

3 Click the Import Button

Exporting to a database

The **File→Export To Database...** function will connect to an existing database and attempt to modify that database's structure to match the current document.

The modification of existing databases is somewhat imperfect. Best results are achieved with an freshly created and empty database as the target.

Exporting to an existing database is likely to cause loss of data.

See also *The Database Export Panel* on page 77

To export a document design to a database

1. Choose **File→Export To Database...**
2. Enter the connection details into the Import Database Panel
3. Click the Export Button

Files: Importing and Exporting

As well as importing and exporting designs to a database, SQLEditor can also import and export to certain file formats. The main format is SQL data description language (often called a database dump), but SQLEditor also support Ruby On Rails migration format.

Future versions of SQLEditor may support additional export formats.

Importing from a SQL File

SQLEditor contains a fairly complete SQL parser that can read in text representations of SQL Databases. Typically you can create these using a database dump utility.

1 Choose **File→Import From File...**

2 Select the file you want to import

3 Click **Import**

A new document will appear containing all of the objects that SQLEditor was able to read in the file.

If SQLEditor had difficulties reading the file then the Status Panel will appear with details of the errors.

See also *The Status Panel* on page 73

Exporting to a SQL text File

SQLEditor can export a database design to a text file, which is can then be imported into a database using a suitable tool.

This is useful if you have a database running on a remote server and you cannot use JDBC to connect to it.

1 Choose **File→Export To File...**

2 Choose the export dialect that matches the database system that you are using

3 Enter a suitable name for the new file

4 Click **Export**

A new document will appear containing all of the objects that SQLEditor was able to read in the file.

If SQLEditor had difficulties reading the file then the Status Panel will appear with details of the errors.

See also *The Status Panel* on page 73

The dialect in the export window is preset to match the document dialect

Exporting to a SQL Diff

Sometimes you may have an existing database, but need to make some changes. For example you might wish to add a column to a table. SQLEditor can help you with by generating a **diff**. A diff is a series of SQL instructions that alter the existing database to a new state.

To generate a diff you will normally need two documents, the first representing the initial state, the second representing the required state. Both need to be open in SQLEditor. You cannot currently compare against a document unless it is open, nor can time-machine or versions documents be used.

SQLEditor also allows you to generate a diff between the current state of a document and a snapshot of the current document when it was most recently loaded. SQLEditor automatically creates a snapshot of each document when loaded for comparison. Note however that if you close a document, any snapshots will be lost. So if you need to do more than a simple one-off comparison, you should save a copy of the document first and compare against that.

1 Make sure that both current and initial state documents are open in SQLEditor

2 Choose **File→Export diff...**

3 Choose the export dialect that matches the database system that you are using

4 Choose the document or snapshot representing the initial state that you want to generate the diff against.

5 Click **Export**

Ruby On Rails Export

The Ruby On Rails exporter is not a database export. Instead it is considered a dialect of SQL, so use the **File→Export to File...** menu option. (This is because the Rails exporter exports a text file rather than interacting with the Rails setup directly.

1 Choose **File→Export to File...**

2 Make sure **Rails** is selected as the dialect

3 select a location to save the model file

4 Enter a suitable name for the new file

5 Click the **Save** Button

Ruby On Rails Import

You can import Ruby On Rails migration files using the Import from File option

1 Choose **File→Import from File...**

2 Choose a file to load

3 Click the **Import** Button

⚠ The Ruby On Rails importer will execute any file that it imports within a Ruby interpreter.

It applies security limitations, but it is possible that these limitations may be overcome by malicious code.

Saving as an Image

You can save an image of the current document in a variety of formats

1 Choose **File→Save as Image...**

2 Select a format to save the image

3 Click the **Save** button

See also *The Image Export Panel* on page 79

Import and Export with the clipboard

SQLEditor also supports importing and exporting SQL via the clipboard.

If you paste SQL code into an SQLEditor document window then SQLEditor will attempt to parse and import any objects it finds. Errors will appear in the status panel.

If you copy objects out of SQLEditor and then paste into a text application then you will get the code for the objects, normally in the current SQL dialect. If you are using Rails as the default dialect then you will get Rails migration format as the pasted text.

Note that you cannot paste Rails Migrations into SQLEditor.
You should use **File→Import From File...** instead

SQLEditor Windows and Panels

The document window

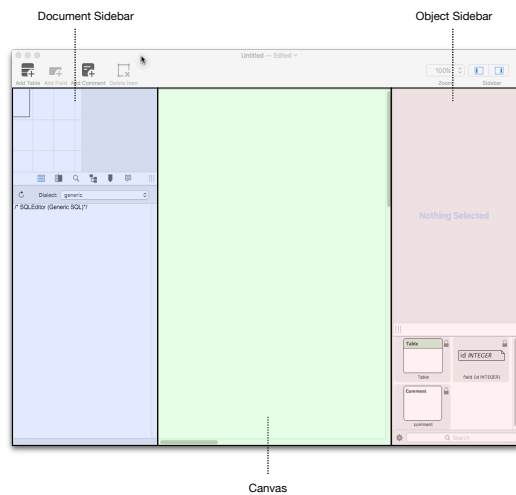The document window is the main window which contains the database structure.

Figure 13: The three parts of the main window

There are three main parts to the document window.

- The **Document Sidebar** (page 67) (blue) on the left, with the **Overview Panel** at the top and the various document source panels below.
- The **Canvas** (green) is in the center of the main window. It shows the main diagram view.
- The **Object Sidebar** (red) on the right. This displays the **Inspector Panel** is in the upper section and the **Object Palette** in the lower section.

The Object Sidebar

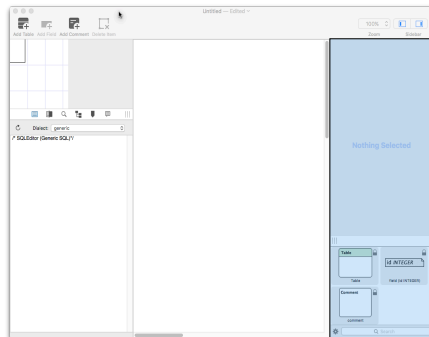The Object Sidebar is on the right side of the main window.

Figure 14: The Object Sidebar is on the right side of the main window

Panels

- The upper section is the *Inspector* (page 63) which lets you edit the settings of the currently select objects
- The lower section is the *Object Palette* (page 64) which manages template objects that can be used in your document.

Figure 15: The Object Sidebar resize handle

Resize the Object Sidebar horizontally

1 Locate the resize handle on the left side of the middle divider

2 Drag the control handle left or right

Change the vertical split between the panels

1 Locate the the middle divider between the two panels

2 Drag the control handle up or down

Do not drag the resize handle on the left edge of the divider, this is for horizontal resizing only

Hide the Object Sidebar

1 Choose **View→Object Sidebar→Hide Object Sidebar**

Show the Object Sidebar

1 Choose **View→Object Sidebar→Show Inspector**

The Inspector

The **Inspector** is the upper section of the *Object Sidebar* (page 61).
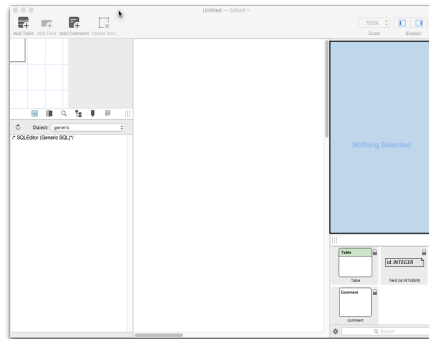
Figure 16: The Inspector is in the upper right corner of the main window

Use the **Inspector** to edit objects.

The **Inspector** will change to reflect the object or objects that are currently selected. This inspector panel is different for each kind of object and offers options and setting specific to that object.

With tables, fields, comments, canvas areas and views, SQLEditor allows you to edit multiple objects at once. Note that you cannot rename groups of objects and that some types of objects don't allow multiple editing.

Show the Inspector

1 Choose **View→Object Sidebar→Show Inspector**

The Object Palette

The object palette contains template objects that can be dragged to the document window.

It is located in the lower right corner of the main document window in the *Object Sidebar* (page 61)
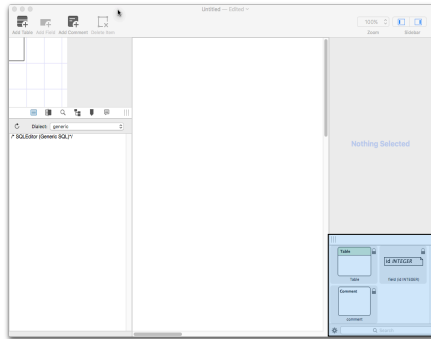
Figure 17: The Object Palette is in the lower right corner of the main window

It can be hidden by dragging the vertical resize bar to the bottom of the window

or revealed by dragging this bar vertically up from the bottom of the window.

When you first start SQLEditor there are object templates for the three basic object types (tables, comments and fields/columns).

You can then add custom templates for objects and groups of objects.

The object palette is shared between documents and is an easy way to keep reusable parts to hand.

Template Action Menu

There is a popup menu on the bottom left of the panel

New from Clipboard   This will create a new template from the current contents of the clipboard (if possible)
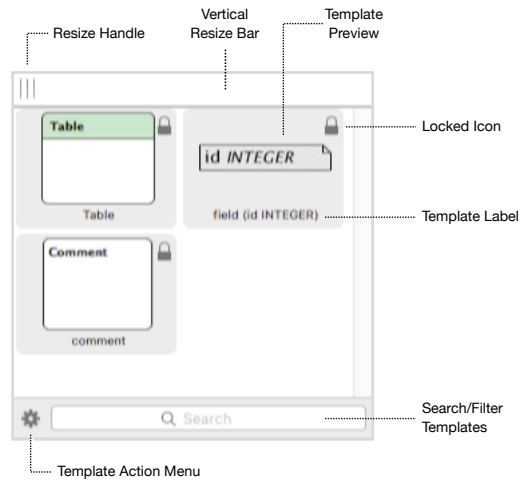
Figure 18: The Object Palette

Duplicate   This will duplicate the currently selected template

Delete Selected   This will delete the currently selected template

Using a palette template

1 Scroll the document to the required location

2 Drag the template to the document window

   A new object will appear in the document window.

Note that the new object name may change slightly depending on the objects currently in the document

Adding templates to the palette

1 Create the new template in a SQLEditor document.

2 Select all of the objects that you want in the new template

| 65

3 Hold option and then drag the objects to the palette

The new palette entry should appear

Alternatively you can use the clipboard instead

1 Create the new template in a SQLEditor document.

2 Select all of the objects that you want in the new template

3 Use **Edit→Copy**

4 Click on the gear icon in the object palette

5 Choose **New from clipboard**

The new palette entry should appear

Removing a template

⚠️ The object palette is shared between all documents and the data is stored in your user preferences.
Deleting a template will remove it for all documents.

You can remove templates that you have added. However the built in templates (currently tables, fields and comments) cannot be removed

1 Click the template that you wish to delete

2 Click on the **Palette Action Button** (the gear icon)

3 In the menu choose **Delete Selected**

The entry should disappear

The Document Sidebar

The document sidebar is on the left side of the main window.

- The **Overview Panel** (page 68) shows a miniature view of the whole of the current document
- The **SQL Source View Panel** displays the SQL source text of the database design.

  Normally it appears on the **left** although it can be hidden if you prefer.
- The **Diff View Panel** displays SQL instructions to change from one version of a document to another
- The **Find Panel** allows you to search the document for specific objects
- The **Tree View Panel** displays a hierarchical tree view of the document
- The **Labels Panel** (page 69) allows you to edit and assign labels to objects
- The **Status Log** shows a list of recent actions for the document and allows you to view output files or action logs

> The **Sidebar** may be hidden by dragging or double clicking the **Sidebar Divider Control.**
> SQLEditor can be configured to hide the **Sidebar** when creating a new document. If the sidebar is hidden just double click on the divider control to open the sidebar.

The Overview Panel

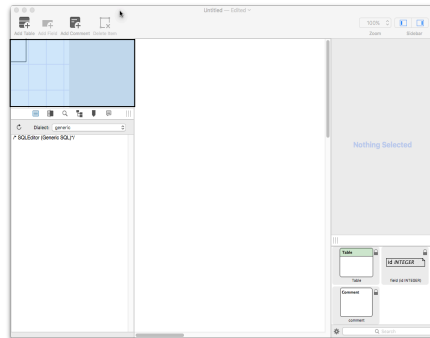The **Overview Panel** is the upper section of the *Document Sidebar* (page 67).

Figure 19: The Overview Panel is in the upper left corner of the main window

The **Overview Panel** shows a miniature view of the whole of the current document, with a rectangle overlayed to show the portion that is currently visible in the main document window.

Objects are drawn using a diagramatic style, rather than being a scaled copy of the full diagram. This panel also shows a grid representing page breaks using the current print settings.

You can drag the visible rectangle using the mouse and the main document scroll position will change to match

If you click on the overview panel the visible area will centered on the point where you clicked.
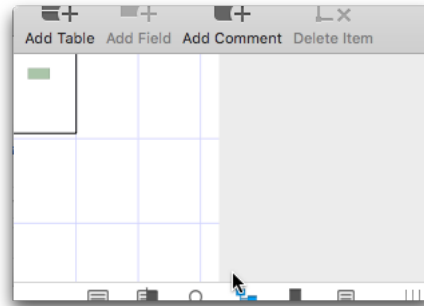
Figure 20: The Overview Panel

The Labels Panel

Labels are used to group together sets of related objects. All of the objects with a specific label will color coded with the label's color. The **Default Label** appears at the top of the list and contains any object not associated with another label. The default label can be shown and hidden, and you can change the color, however you cannot rename it, nor can it be deleted.

In SQLEditor, labels are a way to logically group tables together in a document. They don't affect the output, so you're free to arrange them for your convenience

The labels panel allows you to rename labels, change the label color, move tables from one label to another (as an alternative to using the inspector).

You can also change the visibility of all of the objects in a label group in one action using the visibility control

The label panel has four columns: visibility, name, table count and color:

- The **visibility column** shows whether the label (and all of its tables) is visible or not. Clicking the icon changes the label between visible (open eye) and hidden (closed eye), to change it back click the icon again
- The **name column** displays the name of the label, this can be edited by double clicking it.
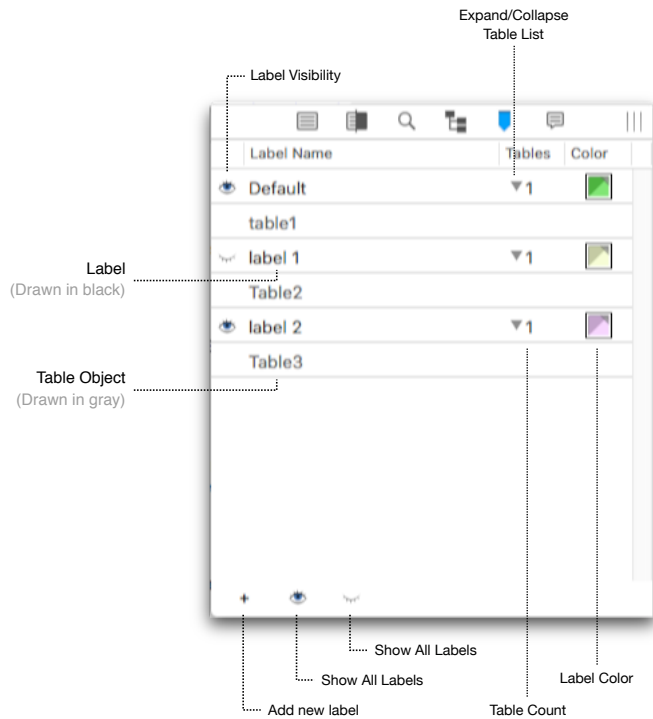
Figure 21: The Labels Panel

- The **table count column** displays the number of tables that are associated with the label, there is also a disclosure triangle button to expand the label and display the associated tables
- The **label color column** allows you to change the tint color for the label. Clicking on the color will display a standard color chooser

At the bottom of the window are controls to to add a new label and to hide and show all the labels in the document. To remove a label from the document, use the context/right click menu and choose **Remove Label**. If you remove a label then all of the tables associated with it are transfered back to the default label.

Editing label name

To edit the name of a label, double click on it.

(Note that you cannot rename the 'Default' label that appears at the top of the list)

Changing label color

Click the label color box in the right column to change the color of an existing label. You may choose any color that you like.

Hide or show all objects with label

If you want to hide all objects with a particular label then you can click the eye icon that appears in the left column. If the eye is open then label is visible, click it to hide all objects with this label. If the eye is closed then the label is hidden, click it to show all lablled objects.

If any individual tables have been hidden, making them visible like this will show them. (It overrides any previous hidden state)

Hide or show all labels

Use the eye buttons at the bottom of the panel to hide or show all labels (and so all objects)

The open eye will make all labels visible.

The close eye will make all labels hidden.

Add new Label

Use the **Add new label** button at the bottom of the panel to add a new label.

If any objects are selected in the current document then the new label will be set on them.

Set object labels

You can set a label for a table by dragging the table onto the label name.

Tables start with the **Default** label until you assign one.

If the table you're looking for isn't visible, try using the arrow in the tables column to expand the label to show the list of tables

Remember that you can also set the object label using the inspector

Currently fields and other objects that appear in tables can be labeled, but don't appear in the labels panel

See also *The Inspector* on page 63

The Status Panel

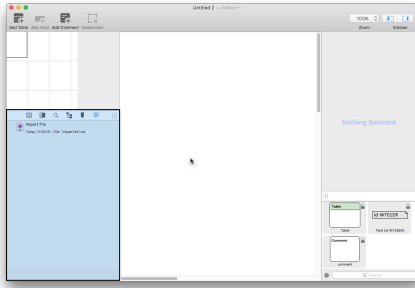The **Overview Panel** is the lower section of the *Document Sidebar* (page 67).

Figure 22: The Status Panel is in the lower left corner of the main window

The Status Panel appears when SQLEditor wants to report something.

The Status Panel displays messages for each action. Actions can also include buttons to allow you to see more information.

Most actions have an icon. For example the image export message shows the icon of the exported image. If you click on it, the newly exported file will be found in the Finder

The actions that can cause the Status Panel to appear include

- **Import From File ...**
- **Import From Database ...**
- **Export To Database ...**
- **Paste** (When pasting SQL directly into a document)
- **Pre-flight Document**

Moving around the canvas

The canvas can be scrolled using the scroll bars at the right and bottom. These scroll bars respond to mouse wheel movements too. You can also scroll by holding down the **Option** or **Alt** key and dragging the canvas using the mouse.

If you customize the document toolbar and add the **Tool** toolbar button you can use a special hand tool to move the canvas.
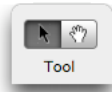
Zooming

You can zoom out the canvas to display more objects simultaneously by using the zoom control in the toolbar. If you have a multi-touch trackpad you can use the standard pinch zoom gestures. SQLEditor can zoom down to 5%.

> You can also optionally zoom using the mouse wheel (although this needs to be enabled first in the Preferences window)

> Print scaling is independent of document zooming. If you set a print scale factor in the **File→Page Setup** panel then this will always be used for printing, regardless of the document zoom level (which is for display only).

You can have more than one document window open at once. To switch between different windows either click on the window you want or use the window menu to choose a different window.

To learn about how to create and edit objects see *Creating and Editing Objects* (page 40).

The Database Import Panel

This panel is used to import database designs from a database.

The import panel is fairly similar to the *Database Export Panel* (page 77).

Presets

The presets menu is used to store groups of connection settings for future use. To do this you should enter all of the details for a connection and then choose **Add New**
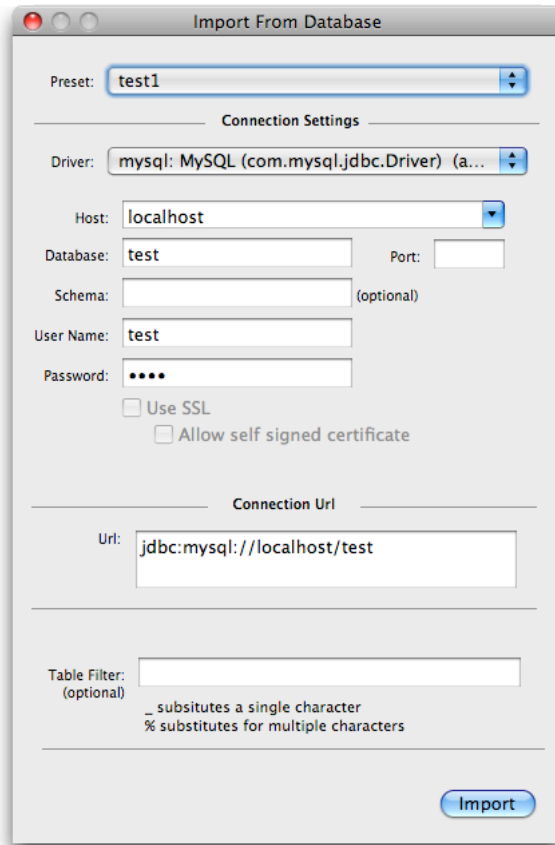
Figure 23: The Database Import Panel

Connection... from the Preset menu. A new entry will then be created in the **Preset Menu**

Driver

The driver popup allows you to choose which database driver you want to use to connect to the database. These need to be installed in a specific location.

Host

The host field specifes the computer that the database software is running on. use **localhost** if the database is running on the same machine as SQLEditor.

Database

Enter the name of the database into this field.

Port

If your database is running on a non-standard port (or you think the number that SQLEditor provides is wrong) then you can provide a different port number in this box.

Schema

If you are using schemas in your database then you can limit the import by entering a particular schema's name into this field

User Name

Enter the username into this field

Password

Enter the password into this field
SQLEditor stores usernames and passwords in the keychain for better security, so you may be asked to enter your keychain password

Use SSL

Some databases support secure connections using SSL, if this is a supported option for the driver then this box will be enabled.

Use Self Signed Certificates

If you enable secure connections but are using a self signed certificate this option relaxes some security checking to allow the connection to work.

Url

The connection string that SQLEditor will use appears here. If you wish you can alter the connection string and SQLEditor will use your string instead.

Table Filter

If you want to import only some tables then you can enter a search string into this box. A simple substitution will occur and any tables which have names matching the string will be imported. Tables that don't match will not be imported.

Import

The import button starts the import process. SQLEditor will create a new document and display a progress bar. When it has received the necessary information from the database it will display the new objects in the document window.

The Database Export Panel

This panel is used to export database designs to supported databases.

The export panel is fairly similar to the *Database Import Panel* (page 74)

The Export panel has most of the options that the Import Panel has, except that the Table Filter field is replaced with the **Remove tables not in current Document**

Remove tables not in current Document

When selected this option causes SQLEditor to remove any tables from the database that are not in the document being exported.
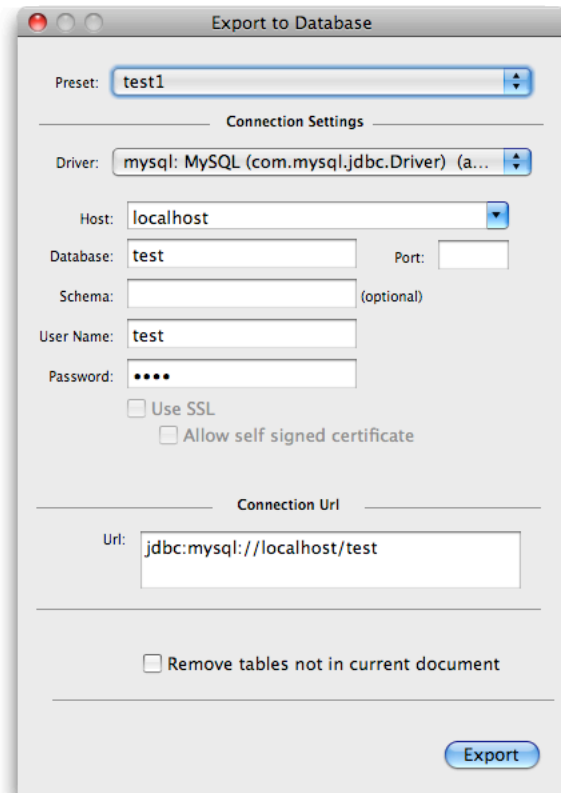
Figure 24: The Database Export Panel

Normally SQLEditor does not delete tables from the database. However any database operation that modifies the structure is potentially harmful and care should be taken.

Simulate Export

Selecting this option will make SQLEditor inspect the existing database and generate instructions to update it. However no updating instructions will actually be sent to the database. You can view the instructions that would have been sent in the Status Panel.

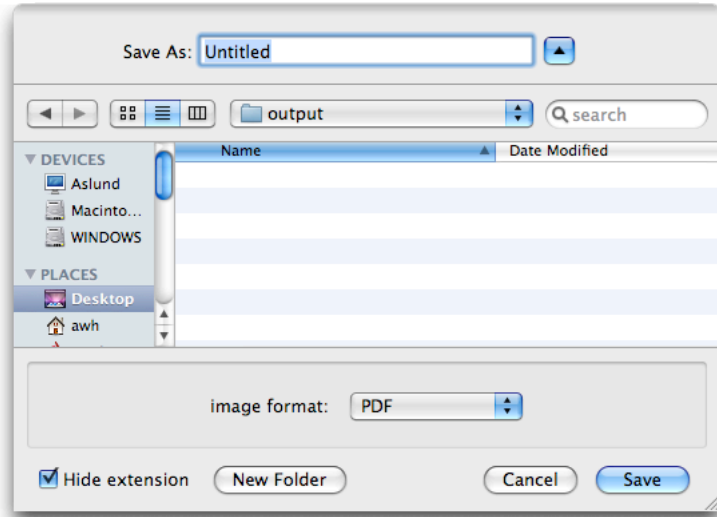See also *The Status Panel* on page 73

The Image Export Panel

Figure 25: The Image Export Panel

This panel is used to export documents as image files.
See also *Saving as an Image* on page 59

Image Format

Choose the image format you want from the popup list.

The Document Options Panel

This panel is used to edit document level settings
It currently has two tabs, **Export** and **Printing**

Export

This tab includes some options relating to SQL export for the document

Target for SQL

This is the dialect that is used to generate SQL in the source sidebar and the default dialect for exporting

It is not used when exporting to JDBC (the correct dialect for the database will normally be used instead)

> You can also change this setting using the menu at the top of the source sidebar

Drop table before create

This option causes SQLEditor to drop all tables at the beginning of an export. It may make table creation more reliable

> If you check this box then existing tables will be deleted

Add SQLEditor settings table

This option (if available) causes SQLEditor to create an extra table to store document settings and object location iwthin the exported SQL. It is normally unavailable and unchecked.

Printing

This tab includes some options relating to document printing

Header

This box allows you to set the header text that appears on the page when printing a document

Footer

This box allows you to set the footer text that appears on the page when printing a document

Preferences

The preferences window allows you modify the way that SQLEditor behaves It preferences are arranged into sections and each section is displayed on a separate tab.

Choose **Preferences**→**SQLEditor** to display the **Preferences Window**

The preferences window contains multiple tabs, change tabs by clicking the icons at the top of the window

General Preferences

The General preferences Tab displays preferences that don't fit one of the other categories.

Open New Document at launch

Normally SQLEditor will create a new untitled document when it starts up. If you uncheck this option then this doesn't happen.

For example if you normally always start by importing a file or if you only working on existing documents then you may wish to change this option.

Show Data Loss warning for database export

This option allows you to disable the warning that appears when SQLEditor thinks a database export operation may cause tables to be deleted

Figure 26: The General Preferences Tab

Disabling this warning means that SQLEditor will delete tables in your database without telling you first. Make sure you have backups.

Use Scroll wheel for Zoom

This option allows you to use the scroll wheel on your mouse to zoom in and out on the document window.

Use command+scroll zoom

This option allows you to zoom the canvas by holding the command key and moving the mouse wheel.

Check for updates automatically

If enabled then SQLEditor will check at intervals to see if a newer version is available.

Check for beta versions   Allow the automatic updater to download pre-release beta versions as well as stable releases.

If you are running a beta version then you cannot uncheck the **Check For beta versions** option (because you are already on a beta version)

Send configuration information   If enabled, SQLEditor will send a snapshot of your current Mac configuration for statistical purposes. This will not contain any identifiable information. The information will contain information such as the current Mac OS X version, the amount of memory you have and the current version of SQLEditor in use. We use this information to determine where to focus our development efforts

Force foreign key type matching

This option makes SQLEditor change the data type of a field with a foreign key relation so that it matches the type of the referenced field.

It is disabled by default

Save connector position

Normally SQLEditor will save the position of orthagonal connectors if they have been manually moved. If you would prefer that this didn't happen disable this option.

Create simple foreign key links when possible

SQLEditor offers two types of foreign keys, either simple connections from one field to another or foreign key field objects which allow multiple fields to be linked together.

Enabling this option causes SQLEditor to generate the simple style references whenever possible, only using the foreign key object when specifically required.

This option only applies when SQLEditor creates a foreign key object, you can always create both types of foreign key by hand if you wish

Use OS X Autosave and Versioning

If this option is enabled then SQLEditor will autosave documents and supporting versioning in OS X 10.7/Lion and later.

Documents may be automatically saved at any time and all changes are tracked.

Disabling this option makes SQLEditor behave in the traditional OS X style, documents are saved only when you choose

You must restart to see changes to this setting.

Auto save documents

Auto save a backup of the current document at intervals. This option is incompatible with the Autosave and Versions support.

Defaults Preferences

The options on the **Defaults** preference tab change the settings that are used by new documents. The options match those available in the **View** menu

See also *The View Menu* on page 105

Show Page Grid

Draws a grid behind the objects for alignment purposes

Snap to Grid

The position of objects will be aligned (or snapped) to the intersections of the grid

Show Page Boundaries

Display lines showing approximate edges of printed pages if the document were printed using current settings

Show Source Sidebar

This option causes SQLEditor to make the source sidebar visible when a new document is opened.

If this option is disabled, the source sidebar can still be made visible after the document has been created.
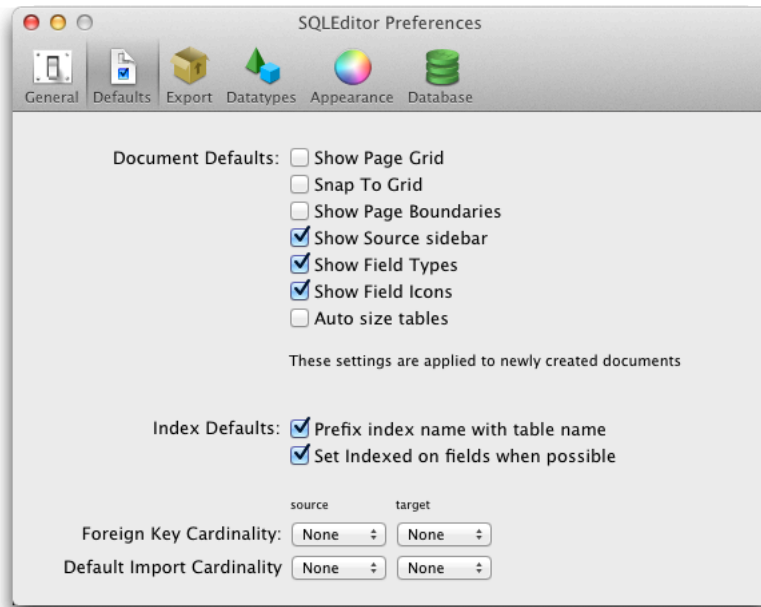
Figure 27: The Defaults Preferences Tab

Show Field Types

Display field types (e.g. VARCHAR, INT) in tables when displayed. This only affects display of the fields, the underlying types are still specified.

This option is mainly to simply display of complicated diagrams where the simplicity of not displaying types is traded off against the total amount of information displayed

Show Field Icons

Display icons for field information (e.g. P for PRIMARY KEY, U for UNSIGNED, etc) This only affects display, the underlying data is unaffected

This option is mainly to simply display of complicated diagrams where the simplicity of not displaying field icons is traded off against the total amount of information displayed

Auto size tables

SQLEditor will resize tables so that the whole width of the name and all of the content in the fields contained in the table are visible

Prefix index name with table name

When creating a new index, add the name of the table to the new index name
This helps to create indexes with unique names

Set indexed on fields where possible

If enabled, when importing, SQLEditor will try to set the **indexed** property on fields rather than creating a new index object in the table.

Foreign Key Cardinality

This option specifies the default cardinality settings for newly created foreign keys that are manually created by you

Default Import Cardinality

This option specifies the default cardinality settings for newly created foreign keys that are created by SQLEditor when it imports a foreign key from a database or via the SQL parser

SQL Export Panel

Default SQL Dialect   This popup menu allows you change the SQL dialect that is used for new untitled documents.

Use Go for sybase Export   This option allows you to choose whether you want to include "GO" statements when using the Sybase Dialect.

Quote Identifiers on Export   This option specifies whether object identifers (field names, table names, etc) are quoted in exported SQL.
This is normally a good idea because it means you have more freedom in the names that you use for objects.
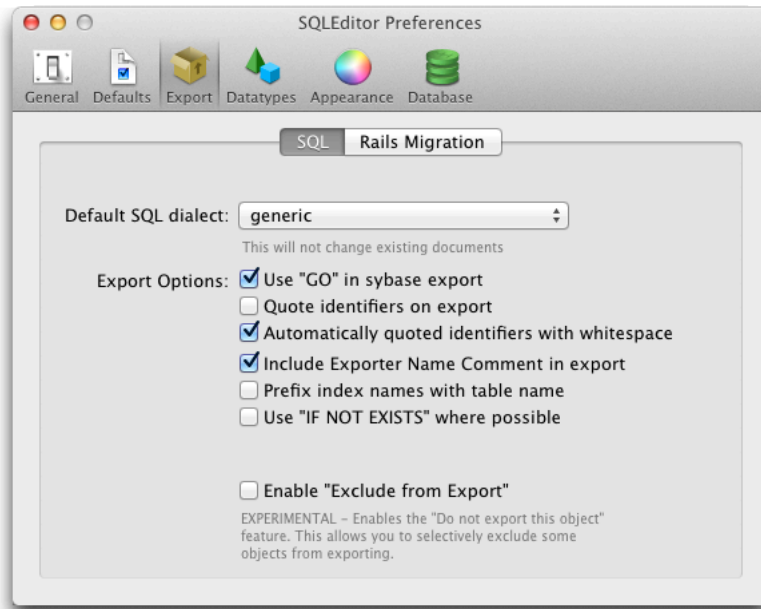
Figure 28: SQL Export Preferences

Automatically quoted identifiers with whitespace   This option specifies whether objects with whitespace in them (e.g "Hello World") are automatically quoted.

This is normally a good idea because it means you have more freedom in the names that you use for objects.

Include Exporter Name Comment in export   This option specifies whether you want the exported SQL to contain a short comment at the top of the file indicating which dialect was used to export it. This is normally useful to readers of the file and has no side effects, but disabling it has no effect either.

Prefix index names with table name   This causes index names to modified on export so that they are prefixed with the name of the table they refer to. This reduces name conflicts, however if you prefer to handle this yourself then disable the option.

Use "IF NOT EXISTS" where Possible   Use the IF NOT EXISTS form of the create table instruction if this is possible

Some database dialects offer the option to create tables only if they do not currently exist. This prevents errors if the table already exists. However it also means that if the table does already exist then you may end up with a database that contains an old version of the table that isn't replaced.

The default export settings are chosen to reduce the likelihood of export errors. Changing these settings may cause exporting to fail.
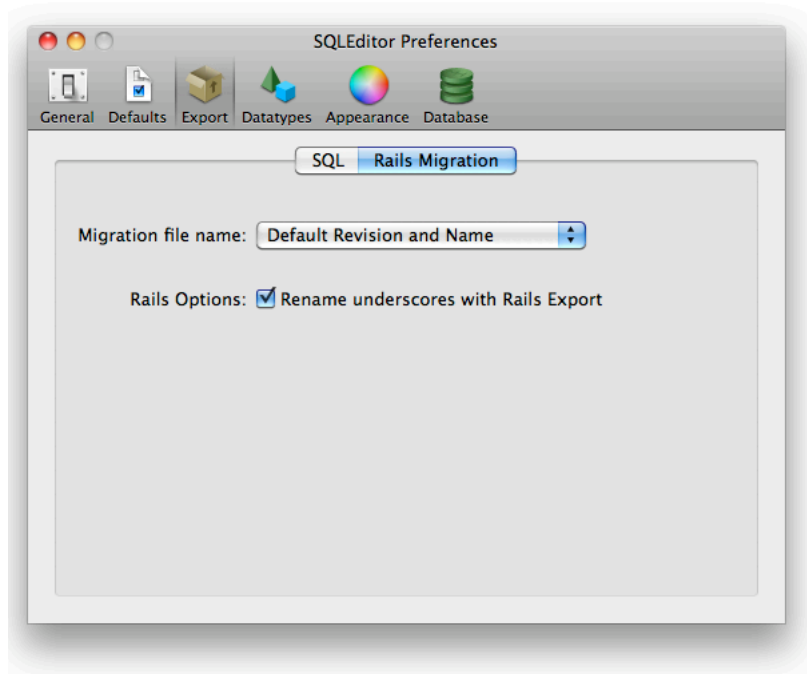
Rails Migration Export Panel

Figure 29: Ruby Migration Export Preferences

Migration file name   This option allows you to choose the default file name when exporting rails migrations. This is the file name that appears in the save sheet before

you edit it. You always have final control over the file name of the migration.

- **Name only** This option gives you a default file name that consists of the document file name modified to meet Ruby class naming requirements
- **Timestamp and Name** This option gives you a Rails 2.x style timestamp and file name default. e.g 20100101_file.rb
- **Default Revision and Name** This option gives you the default revision (initally 1) and a valid file name. e.g 1_file.rb

Rename underscores with Rails Export This option causes underscores in table names to be removed during export.
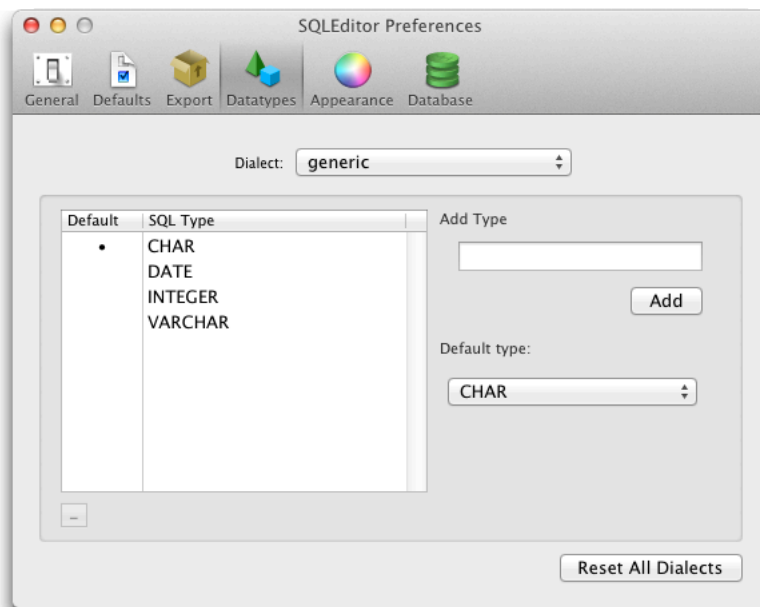
Datatypes

Figure 30: The Data Types Preferences Tab

The DataTypes pane is used to manage data types for different dialects.

The DataTypes pane displays one dialect at a time. To choose which dialect to display use the popup menu **Dialect** at the top of the pane.

When you select a dialect the datatypes for that dialect appear in the left hand list.

You can delete types by clicking the - button at the bottom of the list.

To add a new type enter the type into the **Add Type** box and click the **Add** button.

To set the default type for the dialect choose it from the popup menu. A mark will appear in the list to the left of the default type indicating this.

Automatically add new types as used

If this is enabled then SQLEditor will automatically add new types to the document dialect when you use them for fields.

Reset All Dialects

This button resets all of the dialects back to the configuration that was supplied with SQLEditor.

Reset All Dialects will permanently remove all of your modifications to the dialect types.

Please be careful!

Appearance Preferences

The appearance pane is used to set options for object appearance.

Draw Field Borders

This option sets whether SQLEditor will draw lines between fields in tables.

Always show Drag Handle

This option specifies whether the drag handle on field objects should remain visible at all times. If it is disabled then the handle will be visible only when the field is selected.
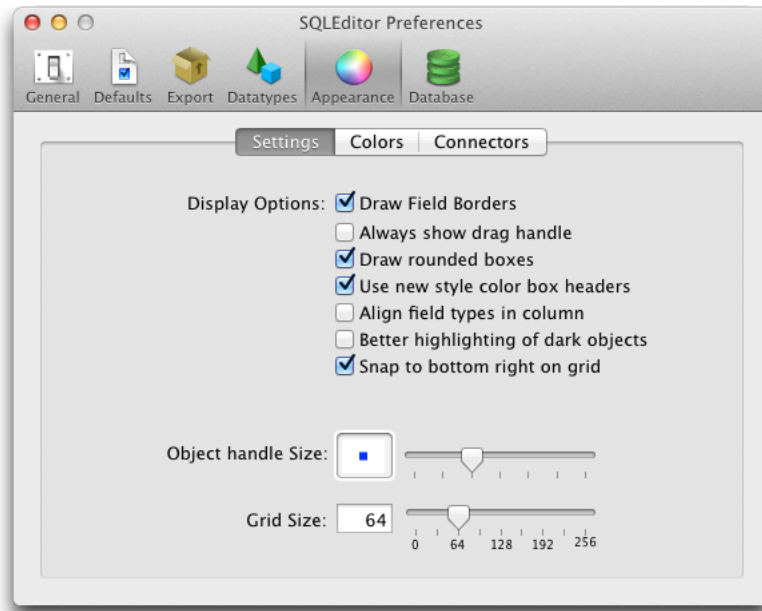
Figure 31: The Appearance Preferences Tab

Draw Rounded Boxes

This option switches between the newer drawing style that uses round boxes and the original SQLEidotr drawing style that uses square boxes.

Use new style color box headers

By default SQLEditor uses changes the color of the table header to represent labels.

If you deselect this option then SQLEditor will color the background of the table instead.

Align Field Types in Column

This option indicates whether SQLEditor should align field type labels in a vertical column or draw the labels immediately after the field name.

Better highlighting of dark objects

Uses an asthetically better, but slower algorithm when drawing highlighted objects. This improves the appearance of objects that have darker backgrounds.

Snap to bottom right on grid

If snap to grid is enabled and this option is also enabled then objects will be snapped to both top left and bottom right on the grid, increasing size if necessary. This means that all objects will be sized to a multiple of the grid size.

Object Handle Size

This allows you to change the size of the draw handles that appear when seleecting, moving and reszing objects.

Grid Size

This allows you to change the size of the background grid that appears when you select **View→Show Grid** and which is used for alignment purposes with **View→Snap to grid**

Object Color Preferences

This section allows you to change the colors of the various display objects. Click the color well to change the color of the specified object

Reset Colors   This button resets the colors to the original defaults, clearing any changes you have made

Connector Style Preferences

On this pane you can choose which style you want SQLEditor to use to draw connectors.

The orthogonal style is the only one that allows you to manually change the position of the connectors.

The vertical connector style will draw a connector attached to the top or bottom of a table (as appropriate)

Figure 32: Appearance Color Options Tab

Database Preferences

The database pane is used to manage database drivers for JDBC Export.

The main section of this pane is taken up with a list of installed database drivers.

All of the database drivers listed will appear when you choose **File→Import From Database...** or **File→Export To Database....**

Adding a new Database Driver

If the database you wish to use has a JDBC driver, but that driver is not currently listed in the window then it may be necessary to add it manually.

1 Click the Button marked with a **+** icon.

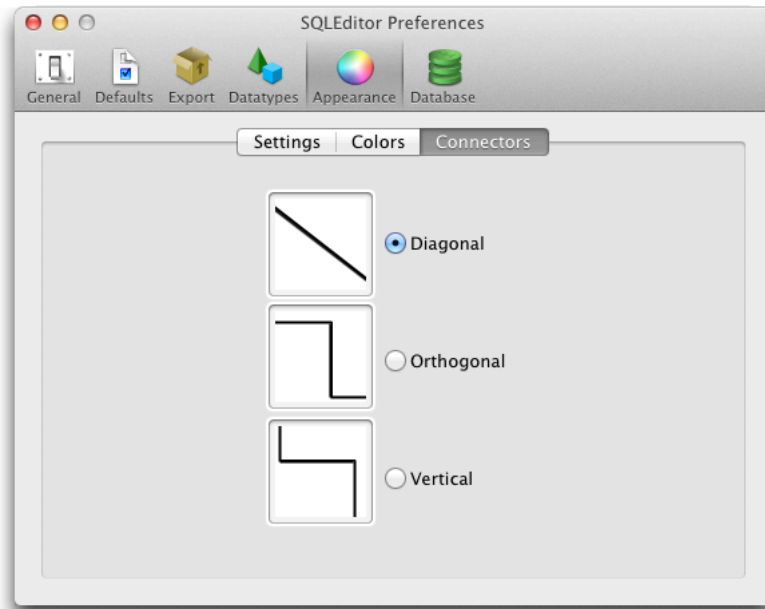2 Enter the details of the new driver in the sheet that appears

3 Click OK

Figure 33: Appearance Connector Options Tab

Open Driver Directory

This will open the JDBC Driver installation directory in the Finder

JDBC Driver Preferences

Use login timeout

Ask the JDBC driver to timeout if it fails to connect within a time limit

Only some databases support this option, but if the server is down or the connection fails, this will display the error faster. In exceptional circumstances this may cause the connection to be cancelled before the server responds, but if the connection is that slow it probably isn't usable anyway
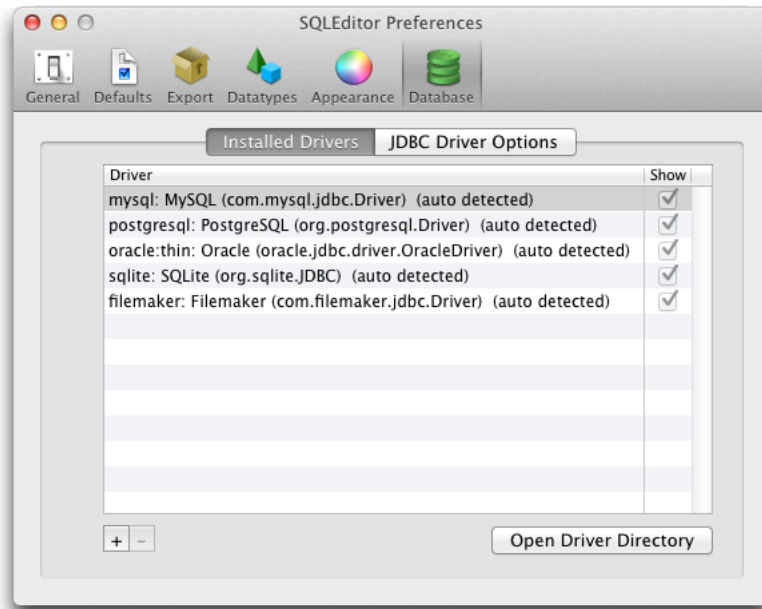
Figure 34: The Database Preferences Tab

Save non-preset connection passwords

This option makes SQLEditor save passwords that you enter into the database connection window even if you don't specifically save them to a preset connection.

Connections saved to a preset will normally always save the password

Figure 35: The JDBC Options Preferences Tab

Menus

SQLEditor Menu

The SQLEditor Menu contains commands to control the application with respect to the rest of the system.

About SQLEditor

Shows the about box which contains the application version number, the serial number and username of the registered user and the expiry date (if applicable)

Check For Updates

Checks the SQLEditor website to see if an updated version is available and can automatically upgrade SQLEditor if you choose

Registration

Allows you to enter your username and serial code to register SQLEditor and disable the nag screen at startup

File Menu

New

Creates a new untitled document using the default dialect

New with Dialect...

Creates a new untitled document, offering a choice of dialect

Open...

Open an existing SQLEditor file

Open recent

A submenu offering recently used documents

Import From File

Create a new document by importing from an existing SQL text file
See also: *Importing from a SQL file* on page 56

Import From Database

See also: *Importing from a Database* on page 53

Close Document

Close the current document. If there are unsaved changed you will be asked if you want to save

The following commands apply with versions support turned off

Save

Save the current document, to the existing file (if one exists) or SQLEditor will ask you to choose a location to save

Save As...

Save the current document to a new location

Revert to Saved

Abandon any changes in the current document and revert the version that was last saved

The following commands apply on 10.7 and later, with versions support enabled

Save a version

Save the current state of the document as a snapshot in the timeline

Duplicate

Create a copy of the current document, to allow saving as a separate document

Revert Document

Revert to a previous revision of the document

Export to File

Export the current document to a SQL text file (or database migration)
See also: *Importing from a SQL file* on page 57

Export to Database

Export the current document to a database, possibly updating an existing database
See also: *Exporting to a database* on page 53

Export Diff

Export a SQL diff to a file. The current document is compared against another document or snapshot. You can choose which in the save panel

Export

If there are any export plugins installed, then they may appear here.

Save as Image

Save the entire current document as an image.
See also: *Save as Image* on page 59

Document Options...

Displays the **Document Options Window** for the current document

See also: *The Document Options Window* on page 79

Page Setup...

Displays the standard Mac OS X page setup window.

Print

Displays the standard Mac OS X print sheet. SQLEditor adds a panel which offers control over how the document is fitted to the paper

Edit Menu

Undo

Undo the last operation

Redo

Redo the last operation that was undone

Cut

Cut the current object from the document and place it on the clipboard

Copy

Copy the current object and place it on the clipboard

Paste

Insert the contents of the clipboard into the document

(i) Not everything can be inserted, watch the status window for errors

Delete

Remove the currently selected object(s) from the document

Select All

Select all objects in the document

Duplicate

Duplicate the currently select object(s)

Find

A submenu with various find commands, only **Find Object** is valid for objects, it opens the Find Object Panel

Spelling

The spelling submenu offers standard commands for spell checking, which are available when editing text

Object Menu

Add Table

Add a new table to the document

Add Comment

Add a new comment to the document

Add Field

Add a new field to the currently selected table.

> (i) If no table is selected, or if more than one table is selected, this option is disabled

Add Index

Add a new Index to the currently selected table.

> (*i*) If no table is selected, or if more than one table is selected, this option is disabled

Add Constraint

Add a new Constraint to the currently selected table.

> (*i*) If no table is selected, or if more than one table is selected, this option is disabled

Add Foreign Key

Add a new Foreign Key object to the currently selected table.
Foreign key objects are distinct from field level foreign keys

> (*i*) If no table is selected, or if more than one table is selected, this option is disabled

Add Canvas Area

Add a new Canvas Area to the document

Add View

Add a new View object the document

Previous Object

Select the previous object in the document.

Next Object

Select the next object in the document.

Inspect Object

Opens the inspector and inspects the current object.

Arrange Objects

Arrange the objects in the document, attempting to remove overlaps and minimize the space used

> *(i)* This may cause arrangements that aren't optimal

Align Objects

The commands in this submenu align objects on the canvas to each other

Align Left Edges Aligns the selected objects so that the left edges are all aligned to the left edge of the leftmost object.

Align Right Edges Aligns the selected objects so that the right edges are all aligned to the right edge of the rightmost object

Align Top Edges Aligns the selected objects so that the top edges are all aligned to the top edge of the highest object

Align Bottom Edges Aligns the selected objects so that the bottom edges are all aligned to the bottom edge of the lowest object

Align Vertical Centers This moves objects so that they are all aligned through their center points in the vertical axis. The alignment point is chosen by averaging the vertical center points.

Align Horizontal Centers This moves objects so that they are all aligned through their center points in the horizontal axis. The alignment point is chosen by averaging the horizontal center points.

Label

This submenu allows you to change the label of an object.

New Label with Selection   Creates a new label, asking for a name and color, then adds all of the currently selected objects to the new label

View Menu

The view menu contains commands to control the visual appearance of the current document.

The settings used in new documents can be changed in the *Defaults Preferences Panel* (page 85)

Zoom

The Zoom menu allows you to increase or decrease the visible size of the objects. It doesn't make any changes to the objects themselves, just your view of them.

Zoom 100%, Zoom 75%, Zoom 50%, Zoom 25%   These commands zoom the canvas to the specified zoom factor

Zoom Area   This command displays a cross hair cusor which you use to drag out an area on the canvas. The canvas zoom factor will then be set so that the entire zoom area fits within the window

Zoom to fit   Change the zoom factor so that all of the objects on the canvas fit in the window without scrolling

Zoom In   Increase the zoom factor by 5%

Zoom Out   Decrease the zoom factor by 5%

Show Page Boundaries

SQLEditor can draw a blue line where the approximate edge of the page will be if the document is printed using the current printer settings

Show Field Types

Set whether the SQL type of fields is drawn.

Show Field Icons

Set whether the informational icons are drawn on fields.

Snap To Grid

Set whether table locations are constrained to the page grid.
This applies when dragging objects

Auto Size Tables

Set whether tables are automatically resized to accomodate their contents.

Resizing may occur when table or field options are changed in order to display all required information and icons

Show All Objects

Set any objects previously hidden to be visible

Toggle Label Visibility

Make hidden labels (and their objects) to be visible, while making visible labels (and their objects) hidden.

Window Menu

Minimize

This action temporarily removes the current window from the screen and places a icon representing it in **the Dock**. You can also perform this action by clicking the orange **Minimize** button on any document window titlebar

Zoom

This exapnds the window to fill the available area.

Enter Full Screen

This action makes the current document a full screen window and hides the menu bar.

Document Sidebar (submenu)

This menu controls the *Document Sidebar* (page 67) on the left of the main window

Source View   This displays the **Source View Panel**

Diff View   This displays the **Diff View Panel**

Find Panel   This displays the **Find Panel**

Object Tree View   This displays the **Object Tree Panel**

Status Log   This displays the **Status Log Panel**

Show/Hide Document Sidebar   This action will show or hide *Document Sidebar* (page 67).

Show/Hide Preflight Panel

This action will show or hide The Preflight Panel

Object Sidebar (submenu)

This menu controls the *Object Sidebar* (page 61) on the right of the main window

Show Inspector   This displays the *The Inspector Panel* (page 63)

Show Object Palette   This displays the *The Object Palette* (page 64)

Bring All To Front

This action moves all open **SQLEditor** document windows in front of any other application's windows.

Bring Current To Front

This action moves the current document window to the front and selects it.

This allows you to switch focus between the inspector and the document window using the keyboard

Show/Hide Toolbar

This action shows or hides the document window toolbar. You can also perform this action by clicking the oval toolbar control button at the top right corner of any document window.

Customize Toolbar

This action will display the standard Mac OS X toolbar customization panel which allows you to add, remove and rearrange the document toolbar.

Open Documents List

The **Window Menu** will also list all currently open documents. You can choose an entry to switch to that document.

Help Menu

SQLEditor Help

Display online help

Read me file

Display the read me file that ships with SQLEditor

Release Notes

Display the SQLEditor change log

Send Suggestion

A shortcut for sending suggestions to the developers. This will either open a new email message in your email program or display a web form. All suggestions are gratefully received

SQLEditor Support (web)

Opens the SQLEditor support page in your web browser

JDBC Driver Support (web)

Opens the SQLEditor JDBC driver web page in your web browser

Online Store (web)

Opens the SQLEditor online store in your web browser

Indices and Appendices

Index

List of Figures