

Entrance:
A Data Analysis Tool for MySQL Databases
Written in Java

Tod Landis

dbEntrance Software
PO Box 2383
Boulder Creek California 95006
(831) 338 6967
tod@dbentrance.com

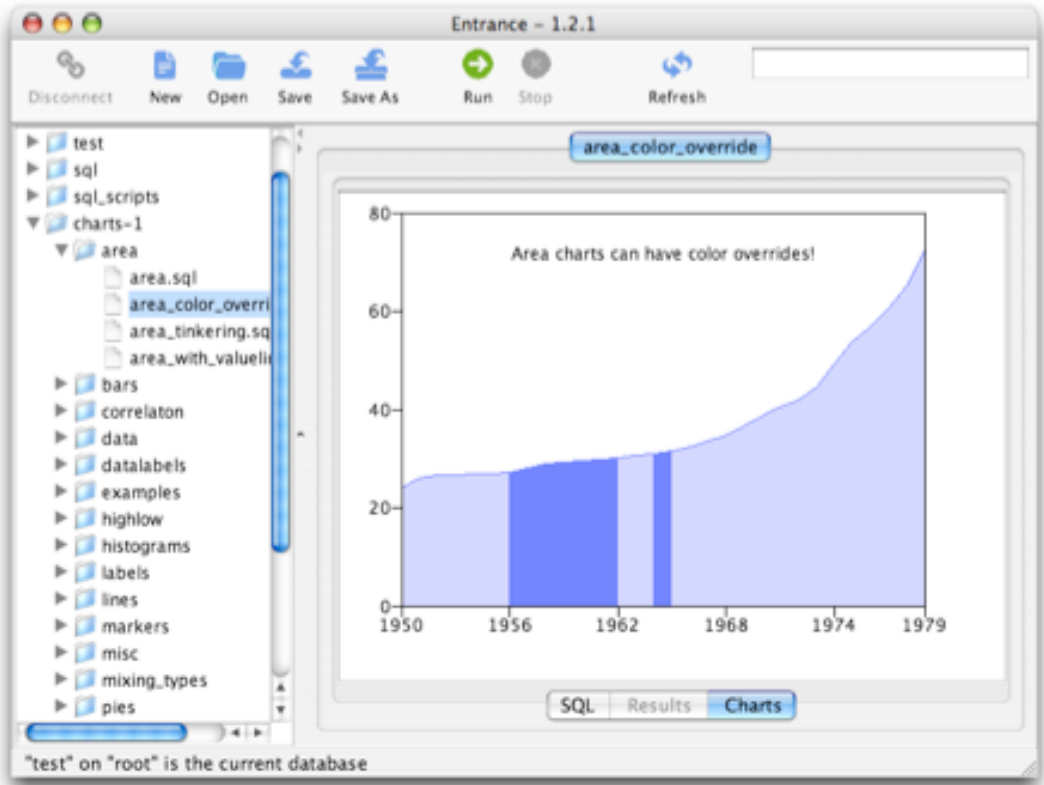
Introduction

Entrance is a desktop application combining a browser for MySQL databases with tools for charting and data exploration. The community version of Entrance is open source, released under the GPL. [dbEntrance Software](#) also sells a version with a syntax highlighting SQL editor and additional features.

Entrance is written in Java. This paper briefly introduces the program, then describes techniques that Java developers can use to customize it or “remix” its features with features from other applications.

A Brief Overview

Entrance looks and feels a lot like programming IDEs, with an explorer tree on the left and tabs on the right that may contain database tables or scripts:

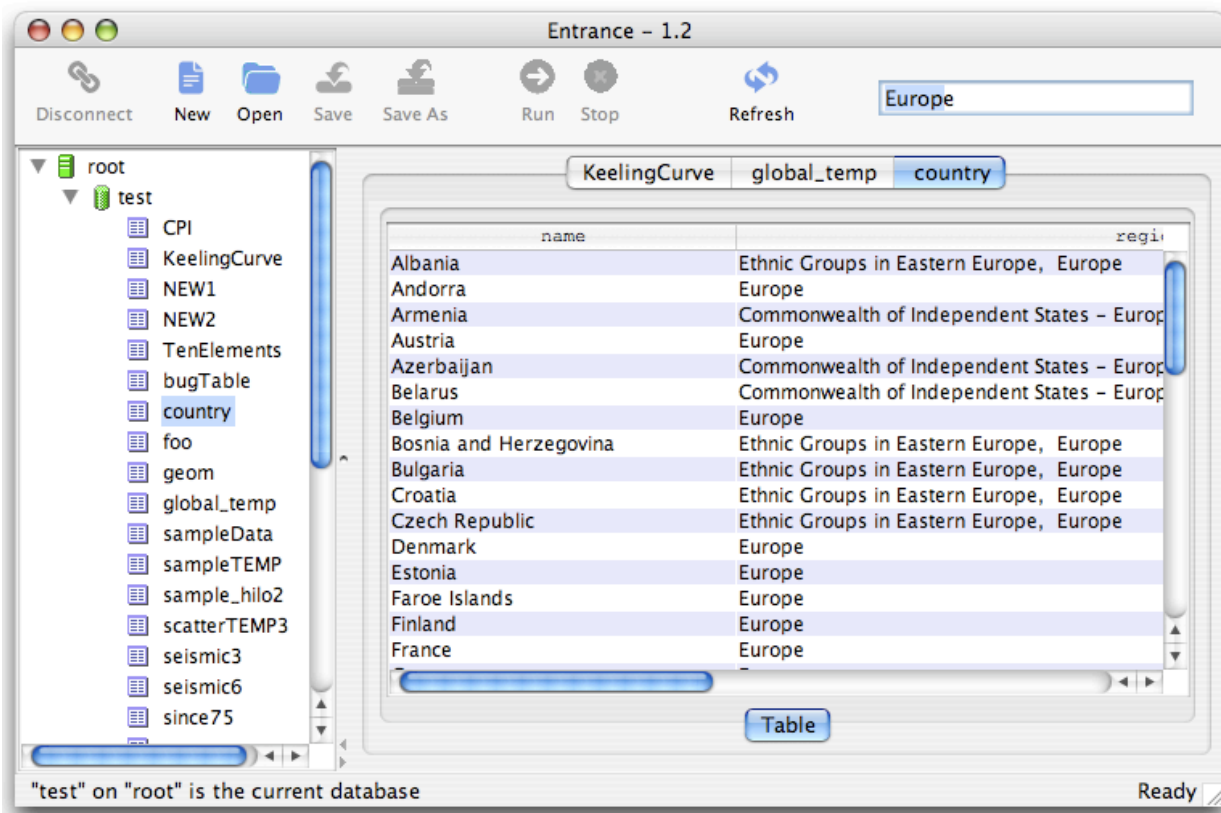


However, Entrance works differently from other IDEs. Each SQL script tab has output tabs attached to it that may contain text or graphic output, and the charts are usually large, as in the image above.

One reason Entrance charts are as large as they are, is that most are interactive. They can be zoomed and panned and data can be selected from them. Entrance also supports something called “data painting”, which is based on a technique developed by William Cleveland.

Multiple script tabs may be open at the same time, and scripts may be run concurrently. You switch between them the usual way: by clicking tabs at the top of the window..

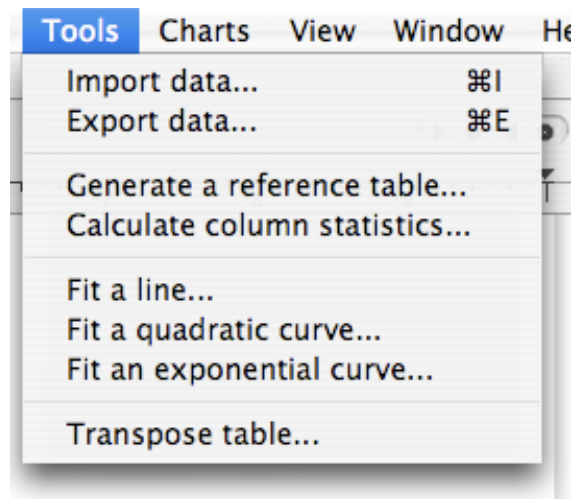
Table tabs may be opened as well. The edit box on the upper right hand corner of the Entrance window is the “Match Box”:



When the current tab is a table, you can filter for matching rows by entering text in the box. If you enter something that begins with “WHERE”, the text you enter is treated as a WHERE clause and filters the view.

Tools

The Entrance Tools menu provides tools for making acting on the data in a MySQL database. What appears in the Tools menu varies from version to version. In version 1.2.22 it looked like this:



Java is “behind” all these tools, as you would expect, but sometimes the Java code invokes scripts written in other languages. “Send to iPhoto...” is an example where AppleScript code is invoked to complete the action.

Most Entrance tools are implemented using a dialog with a constructor that takes a single `JavaClient` argument. Using that `JavaClient` instance, the dialog code can find out most other things it needs to know.

The `JavaClient` method `getCurrentPane()` returns the pane tab that is on top, which will be either a `QueryPane` or a `TablePane`. When the current pane is a `TablePane` there will be a current MySQL host, current database, current table and so on. Look at the “Fit a quadratic curve...” tool, implemented by `FitQuadraticDlg` to see how this works.

Tools can also generate scripts “on the fly”, and create new tab panes for them using `JavaClient.newQueryPane(String s)`. “Make a line chart...” works this way for example (see `MakeALineChartDlg`).

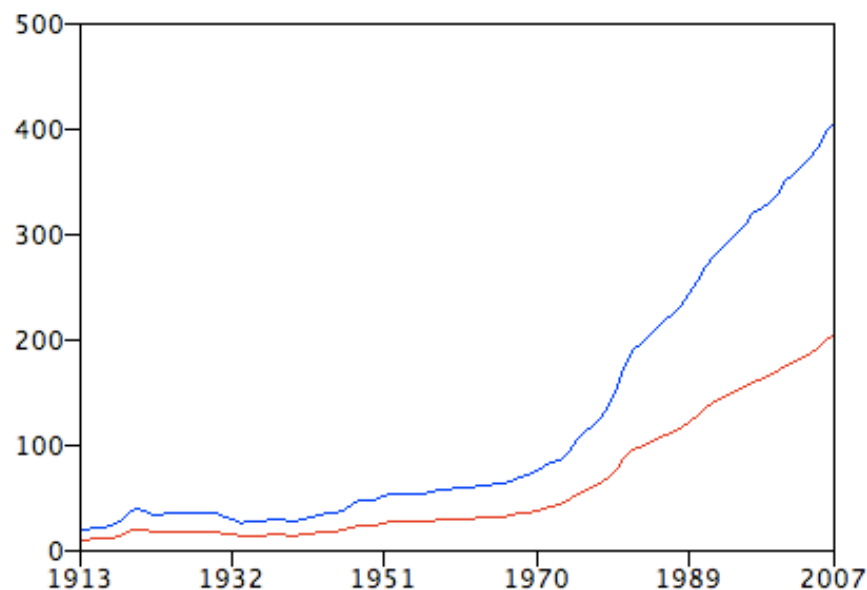
Charts

Users define charts using the PLOT keyword in SQL scripts, and there are wizard-like dialogs to help them do this.

The top part of a PLOT command describes the chart, and the bottom part is an arbitrary MySQL SELECT statement. This approach means SQL developers can quickly produce charts based on SELECT queries they are familiar with.

Chart series in the top part should match up 1-to-1 with columns in the result set produced by the SELECT statement. As an example, this script:

```
PLOT
  XLABELS, RED LINE, BLUE LINE
SELECT
  years, col1, col2 FROM sampleData;
produces this chart:
```



Users can add elements like titles, legends, and so on using the syntax described in the PLOT syntax document. See <http://dbentrance.com/docs/plotsyntax.html> for more details.

Depending on the chart type, Entrance charts may be used interactively. Many support data selection and "data painting". See <http://dbentrance.com/howto/ExploreData.html> for more details.

They can also be printed, save in various image file formats, or included in web pages.

Defining new charts

The Entrance chart making system is easily extended, and it provides elements that can be reused by new chart types. These capabilities largely "fell out" from the choice of Java as an implementation language and the fact that Entrance is open source.

The base class for all Entrance charts is Chart. Chart subclasses are expected to override draw(Graphics2D g) and draw something within the page bounds they obtain by calling getPageX(), getPageY(), getPageWidth(), and getPageHeight().

Output may be on its way to the screen, to a printer, or to an image file, but it won't usually matter to the chart code.

The Chart base class provides methods which can place a chart "frame" on the page, draw titles, gridlines, a legend and other chart elements.

There are also Chart methods which may be called by the Entrance ChartPanel class for panning, zooming, and data selection. By default, Chart is set up to inform ChartPanel that none of these features are supported, so you won't need to worry about them to write a simple extension. See the javadoc comments for Chart for more details.

These are the Chart subclasses, so far:

```
Chart
  LineChart
    Histogram
    BarChart
    AreaChart
    HighLowChart
  RotatedBarChart
  ScatterChart
  PieChart
  WhiskersChart
  CorrelationChart
  NopChart
```

The easiest way to add a new Entrance chart class is by extending an old one, so next we will describe how to do that.

Modifying LineChart

LineChart is the workhorse chart. It draws bars, markers, areas, “swaths”, ie. bands of colors, or a mix of these types. It also draws datalabels and “whenlines”, which are vertical lines drawn across the chart.

Subclass LineChart and override drawSeries() when you want the axes and other chart elements handled the way LineChart does, but the series elements, eg. lines or bars, to be drawn differently.

Here is an override that could do something special columns marked with the HIGH series type, while drawing everything else the same:

```
package entrancex.charts
import com.dbentrance.charts.LineChart;

public class MyLineChart extends LineChart {

    public void drawSeries(Graphics2D g, int jColor, int jSize) {

        while(i < data.getColumnCount()) {
            switch(getColumnType(i)) {
                case HIGH:
                    myHighSeries(g, jColor, jSize);
                    break;
                default:
                    super.drawLineSeries(g, i, jColor, jSize);
            }
        }
    }

    public void myDrawSeries(Graphics2D g, int i, int jColor,
        int jSize) {
        // ... put your new stuff here ...
    }
}
```

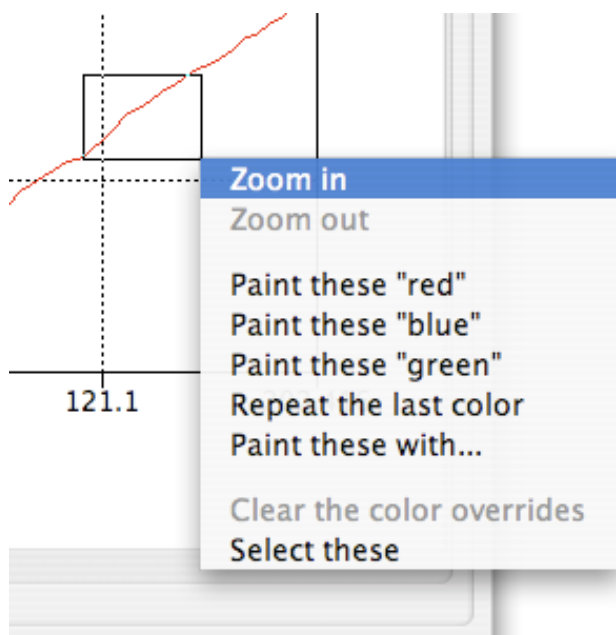
Once you have made MyLineChart available on the Entrance classpath, users can invoke it from a SQL script like this:

```
PLOT entrancex.charts.MyLineChart
    XLABELS, RED LINE
SELECT thelabels, theline FROM sample;
```

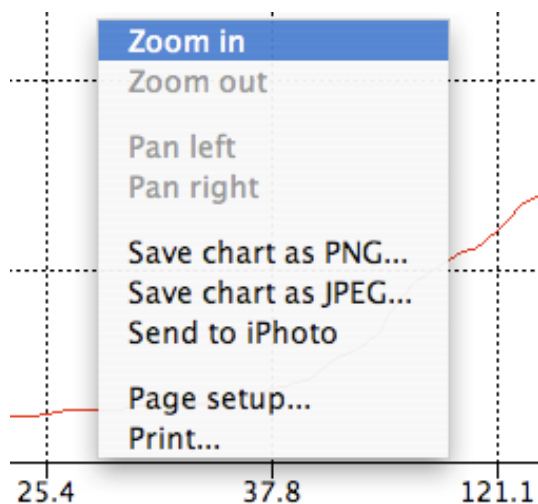
One way make it available is by including it in entrancex.jar, which is kept in the Entrance lib directory and is automatically on the Entrance classpath.

Data Exploration and “Data Painting”

Entrance supports tools for graphical data exploration, including one called “data painting”. You can see some of these tools by holding down the mouse and dragging out a box over a LineChart:



and more by right clicking it (or control-clicking on a Macintosh):

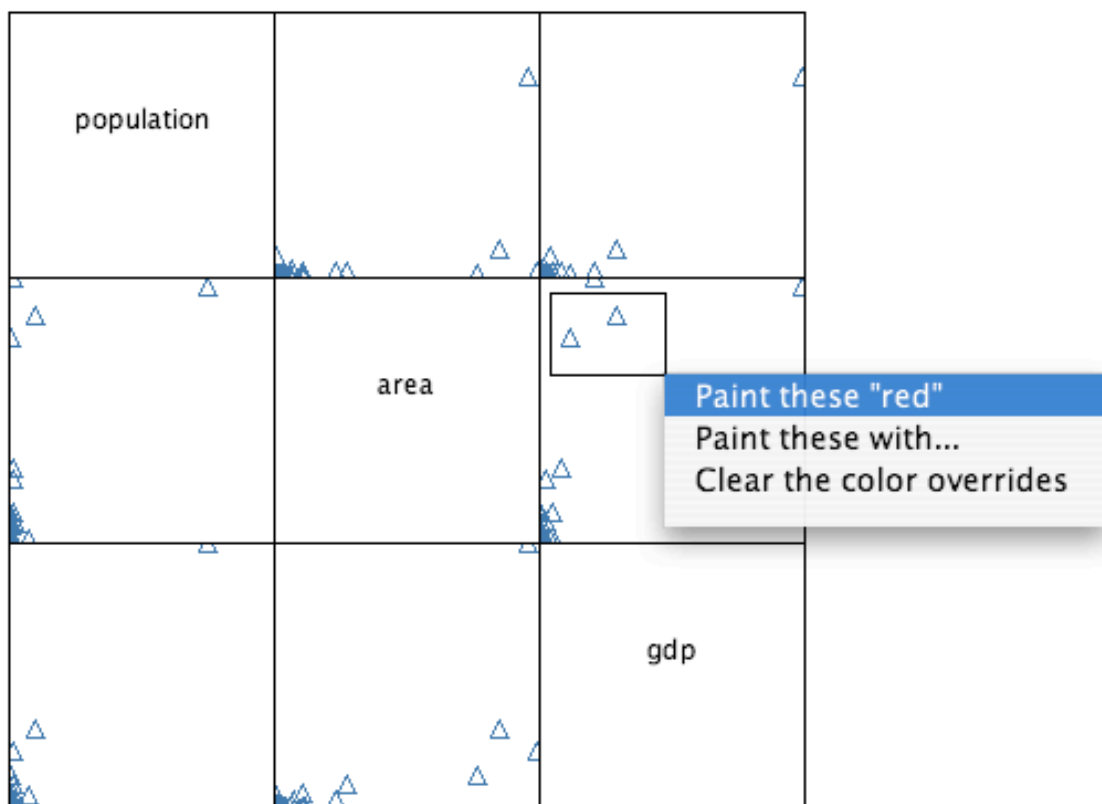


“Painting data” requires a little preparation. Your script needs a “color override”:

```
PLOT CORRELATION
  TRIANGLE, TRIANGLE, TRIANGLE, COLOR OVERRIDE
WITH
  FRAME 20 20 400 400
SELECT population, area, gdp, color from country;
```

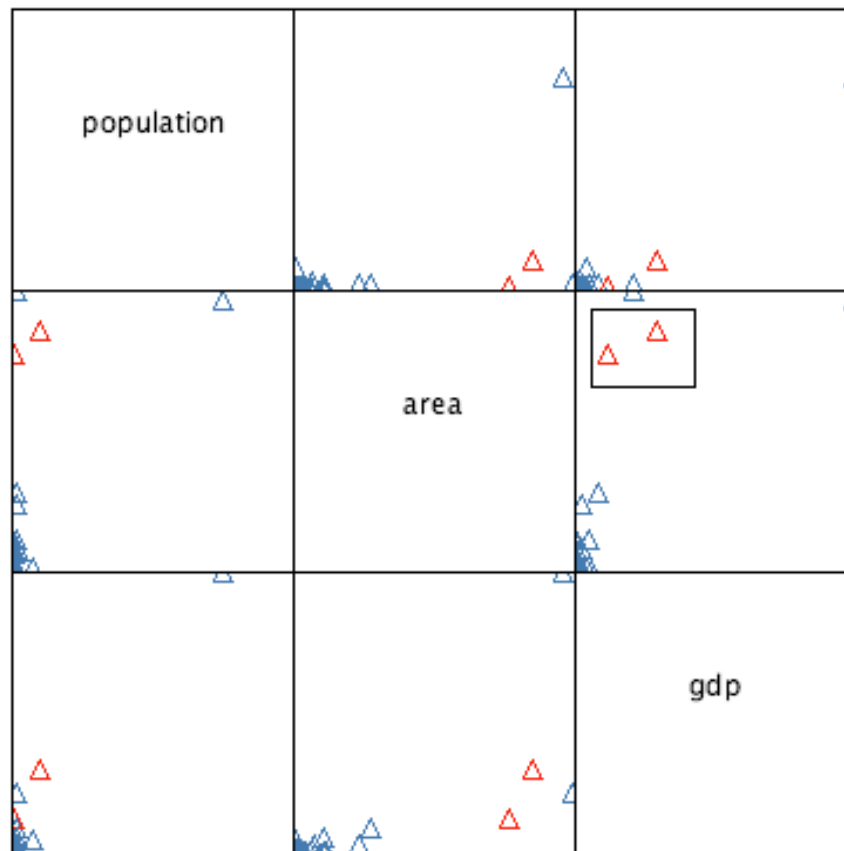
and the underlying table it references must have a primary key (so that SELECT produces an updatable ResultSet). If these conditions are met, you are ready to paint.

Drag out a box over some the points you see in the chart:



and paint them "red" or another color. You are actually changing the color value of rows corresponding to points in the table.

When the chart is automatically redrawn, you see the changes:



A Chart subclass can also choose to turn off any of the data exploration features by overriding returning false from these methods:

```
public boolean supportsZooming() {
    return false;
}
public boolean supportsPanning() {
    return false;
}
public boolean supportsReverseMapping() {
    return false;
}
```

See “Exploring Data”: <http://dbentrance.com/howto/ExploreData.html> for more.

Using Entrance charts and tools in other applications

Most dialogs in the Entrance UI require a JavaClient window, because they use JavaClient methods to learn about the current “context”: eg. the current MySQL table, current script, and so on. Entrance must be running and a JavaClient window must be open for one of these dialogs to work properly.

However, Entrance also contains classes which are Entrance-independent, meaning that they can be run when Entrance is not. They can be used by placing the jar file, entrance.jar, on another application’s class path.

All of the Chart classes are Entrance-independent, The example, EntrancePlot, shows how to use them to produce image files from a command line utility. A similar approach could be used to build chart generating servlets.

The Entrance Chart classes can actually be used to plot data in any JDBC ResultSet, not just those generated by the MySQL driver. They can even plot data in two dimensional arrays of doubles or integers. To learn more, read the comments for the Data and Chart classes.

Classes implementing line and curve fitting are also Entrance-independent. These classes are: ExponentialFit, LineFit, and QuadraticFit, in the package com.dbentrance.entrance.tools. More Entrance-independent tools will be joining them there soon.

Using tools from other applications in Entrance

We are also very interested in “remixing” capabilities from other applications and Java class libraries in Entrance. Two packages we already use in this way are the [JAMA matrix package](#) and the [Cotta file management tools](#). Their jars can be found in the Entrance library folder.

Because Entrance is open source, it is a simple thing to add new menu items to the Tools menu or other menus by rebuilding the jar. Entrance makes it easy to include web page bookmarks in the menu, the Help menu contains examples, or to invoke script languages like AppleScript , “Send to iPhoto...” is implemented this way.

What's coming?

Extension frameworks for tools and report generation: We're working on new frameworks that will work similarly to the chart framework. These frameworks will make it easier to "remix" functionality from other applications.

HTML Export: Entrance HTML export will be template driven

JavaScript as an automation language: JavaScript will be to Entrance, as Basic is to Microsoft Access.

A place to exchange Entrance scripts and extensions: We plan to provide a web page where Entrance developers can meet each other and exchange ideas. More to come on that.

Learning more

If you are interested in working on Entrance scripts or extensions, send me email at: tod@dbentrance.com, and I will let you know the latest developments. (This will work best if you include Entrance somewhere in the subject)

Entrance can be downloaded from <http://dbentrance.com/>. The web site also has links to documentation and support pages.

There are extended SQL script examples in "charts.zip". The best way to try them is download the file, unzip it, and add the resulting folder to Entrance as a file folder node. Right click (control-click on the Mac) on a script to bring up its popup menu and run it. The "data" folder contains SQL scripts for creating the tables you need for the examples.

Shane Duan and I will be at the [MySQL Conference in Santa Clara, April 14th-17th](#). If you attend the conference, we hope you will stop by the Entrance booth in the Dot Org Pavillion to say hello. Shane can be reached at shane@dbentrance.com.