# Table of Contents

Acknowledgements

1.0 Introduction

1.1 Purpose

1.2 Product Scope

1.3 Existing System

1.4 Proposed System

2.0 Review of Literature

2.1 HTML

2.2 CSS

2.3 PHP

2.4 jQuery

2.5 MySQL

2.6 AJAX

3.0 Requirements

3.1 Functional Requirements

3.2 Non Functional Requirements

4.0 System Analysis and Design

4.1 Data Flow Diagrams

4.2 Behavioural Modelling

Use case Diagrams

Interaction Diagrams

Activity Diagrams

Database Diagrams

4.3 User Interface Design

4.4 Architecture of the System

4.1.1 Two Tier Architecture

4.1.2 Three Tier Architecture

5.0 Implementation

5.1 Development Tools

5.1.1 NetBeans

5.1.2 XAMPP

5.1.3 Google Chrome Dev Tools

5.2 Repository

5.2.1 SVN

5.2.2 Assembla SVN

5.2.3 Tortoise SVN

5.2.4 Checkout and Commit

5.2.5 Other Operations

5.3 Productivity Tools

5.3.1 Google Hangout

5.3.2 Google Drive

5.4 Modules and Functionalities

5.4.1 Work Area

5.4.2 Data Structures

5.4.2.1 queryString

5.4.2.2 metaArray

5.4.2.3 metaArray2

5.4.3 Green-Red Concept

5.4.4 Generating Joins

5.4.5 Eliminating Extra Join Column

5.4.6 Methods Library

5.4.7 Slick Grid

5.4.8 ColorBox

5.4.9 jQuery mCustomScrollbar

6.0 Testing and Deployment

6.1 Test Database

6.1.1 Classic Models

6.1.2 Sakila

6.1.3 Sakila Tables

6.2 Test Queries

6.3 GUI Testing for different Screen Resolutions

6.4 Production Environment

6.5 Minimum System Requirements

6.5.1 Client Requirements

Hardware

Sofware

6.5.2 Server Requirements

Hardware

Software

6.5.3 Database Server Requirements

5.3.1 Hardware

5.3.2 Software

6.6 Installation and Configuration

7.0 Conclusion / Future Enhancements / Recommendations

8.0 References / Bibiliography

# ACKNOWLEDGMENTS

We would like thank the following sources for their help, guidance and timely Motivation:

1. MVSR Engineering College
2. Mr. Prasanna Kumar sir, Head of the Computer Science and Engineering Department
3. Mr. K V Phaneendhra sir
4. Dr. Sandhya Madam

The Project Development Team includes:

MVSR Students:

1. Rahul Jujarey, CSE, 2010-2013
2. Goutami Kondakindi 2009-2013
3. Janaki Purnima 2009-201
4. Santhosh Kolloju 2010-2014

# Introduction



## 1.1 Purpose

Develop a highly interactive and presentable web interface or simply a great UI to communicate with backend databases and fetch requested data.

## 1.2 Product Scope

The purpose of this WA (Web App ) is to enable Web users to fetch their data needs from processed data stores in the datacentres.

## Existing System

There are many Open Source Database Servers, and small database Servers such as MySQL that are feature effective in comparison with the commercial products, but they lack an user interface for accessing and executing queries.

There are some applications where we may need access to database in a Web Context, and some applications require to access huge database that may contain millions of rows. Such accesses should be handled neatly and effectively.

## 1.4 Proposed System

We propose a Web Application that can run in Web Context (Client Browsers), can access remote Database, effectively handle million rows, Dynamic enough, to be able to accommodate in the changes in the database.

# 2.0 Review of Literature

## 2.1 HTML

HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

HTML5 is the latest verion in HTML. HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. It is the fifth revision of the HTML standard. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but XHTML 1 and DOM Level 2 HTML as well.

## 2.2 CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, and spacing) to Web documents. It is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. It’s most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS style sheet, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

CSS specifies a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities or weights are calculated and assigned to rules, so that the results are predictable.

CSS3 is the latest standard for CSS.

## 2.3 PHP



PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language.

PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP acts primarily as a filter, taking input from a file or stream containing text and/or PHP instructions and outputting another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

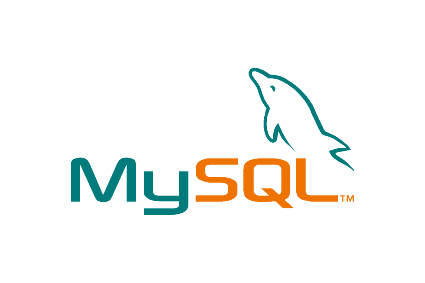
## 2.4 jQuery



jQuery is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML. It was released in January 2006 at BarCamp NYC by John Resig. It is currently developed by a team of developers led by Dave Methvin. Used by over 55% of the 10,000 most visited websites, jQuery is the most popular JavaScript library in use today.

jQuery is free, open source software, licensed under the MIT License. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

## 2.5 MySQL



MySQL (My Sequel) is the world's most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, Nokia.com, and YouTube.

MySQL is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

## 2.6 AJAX

AJAX = Asynchronous JavaScript and XML.

AJAX is not a new programming language, but a new way to use existing standards.

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

It is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. Data can be retrieved using the XMLHttpRequest object. Despite the name, the use of XML is not required (JSON is often used instead), and the requests do not need to be asynchronous.

The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with the current state of the page.

The following Technologies can be incorporated into AJAX:

* HTML (or XHTML) and CSS for presentation
* The Document Object Model (DOM) for dynamic display of and interaction with data
* XML for the interchange of dataThe XMLHttpRequest object for asynchronous communication
* JavaScript to bring these technologies together

XML is not the only format for data interchange. JavaScript Object Notation (JSON) is often used as an alternative format for data interchange, although other formats such as preformatted HTML or plain text can also be used.

# 4.0 Modules and Functionalities

# 5.0 Requirements

## 5.1 Client Requirements

### Hardware

* RAM : 128 MB
* Harddisk : 40GB

### Software

* Operating System: Windows XP, 7, 8
* Browser: Anyone of Firefox , Chrome 16.0 or higher versions

## 5.2 Server Requirements

### Hardware

* RAM:2 GB
* Harddisk:500 GB

### Software

* Operating System: Windows Server 2003 or Linux
* Server: Apache 2.0 or greater with Php Module

## 5.3 Database Server Requirements

### Hardware

* RAM:2 GB
* Harddisk:500 GB

### Software

* Operating Sytem: Windows Server 2003 or Linux
* Server: MySQL 5.5 or greater

## 

## 5.4 Functional Requirements

1. Enable the user in comminicating with any back end databases, by seeking user requests , passing these requests to the backend, retrieving the data sets , displaying the dataset in the form of a grid in the front end UI.
2. Abilty to store the data results for the stored queries ( mentioned in # 3)
3. Abilty to display the time taken to fetch the data from the backend data base, # of records ( it might be possible that the data fetch may yield millions of rows, but due to the limitation of display , we might end up showing the first 100 rows , while a facility to be offered to display other records upon user request).
4. Ability to add Dynamic Filters to the data query.
5. Be able to export the final dataset in the form of Excel Sheet

## Non-Functional Requirements

1. A great UI , highly interactive & pleasant experience.
2. Extensible to add more Features
3. Pagination Controls for navigating through the database.
4. Be able to display the Final Query to the user.
5. SubVersioning the development process, so that intermediate rip-offs can be produced tailored for particular business function
6. **Maintainability :A**ll the modules must be clearly separate to allow different user interfaces to be developed in future. Through thoughtful and effective software engineering, all steps of the software development process will be well documented to ensure maintainability of the product throughout its life time. All development will be provided with good documentation
7. **Performance :** The response time, utilization and throughput behavior of the system. Care is taken so as to ensure a system with comparatively high performance.
8. **Usability:** The ease of use and training the end users of the system is usability. System should have qualities like- learning ability, efficiency, affect, control. The main aim of the project is to reduce server failure and gets high performance of server and reduce the rework of the programmer.
9. **Modifiability:** The ease with which a software system can accommodate changes to its software is modifiability. Our project is easily adaptable for changes that is useful for the application to withstand the needs of the users.
10. **Portability:** The ability of the system to run under different computing environments. The environment types can be either hardware or software, but is usually a combination of two.

# 6.0 System Design

## 6.2 Sequence Diagram

# 7.0 System Analysis

## 7.1 Architecture

We have Database, Server Side, ClientSide

### 7.1.1 Two Tier Architecture

In a Two Tier Architecture we have Client and Client Side, Our PHP Application and Database on a Single Server



### 7.1.2 Three Tier Architecture

In a Three Tier Application, we have Client on Client Side, Our PHP Application and Config Database (UI-Data) on a Server Side System. This Application establishes a remote connection with the Remote Database, located at a Customer Site.

# 

# 8.0 Development

## 8.1 Development Tools

### 8.1.1 NetBeans



NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others.

The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM.

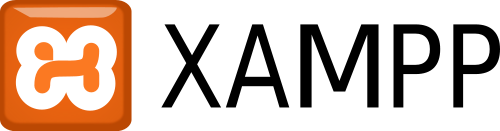
The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers.

#### NetBeans IDE Bundle for PHP

NetBeans supports PHP since version 6.5. The bundle for PHP includes:

* Syntax highlighting, code completion, occurrence highlighting, error highlighting, CVS version control
* Semantic analysis with highlighting of parameters and unused local variables
* PHP code debugging with xdebug
* PHP Unit testing with PHPUnit and Selenium
* Code coverage
* Symfony framework support
* Zend Framework support
* PHP 5.3 namespace and closure support
* Code Folding for Control Structures

### 8.1.2 XAMPP



XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages

The philosophy behind XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers XAMPP is configured with all features turned on.

The default configuration is not good from a security point of view and it's not secure enough for a production environment - please don't use XAMPP in such environment.

XAMPP's name is an acronym for:

* X (to be read as "cross", meaning cross-platform)
* Apache HTTP Server
* MySQL
* PHP
* Perl

XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.[2] In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla. It is also possible to connect to localhost via FTP with a HTML editor.

XAMPP 1.8.1 for Windows, including:

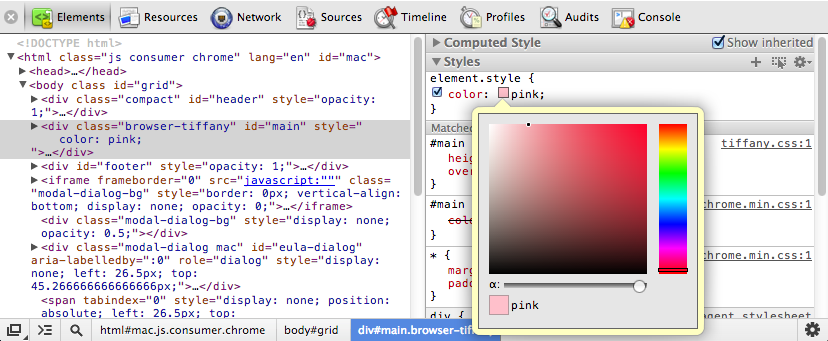
* Apache 2.4.3
* MySQL 5.5.27
* PHP 5.4.7
* phpMyAdmin 3.5.2.2
* FileZilla FTP Server 0.9.41
* Tomcat 7.0.30 (with mod\_proxy\_ajp as connector)
* Strawberry Perl 5.16.1.1 Portable
* XAMPP Control Panel 3.1.0 (from hackattack142)

### 8.1.3 Google Chrome Dev Tools

The DevTools, bundled in Chrome, provide web developers deep access into the internals of the browser and their web application.

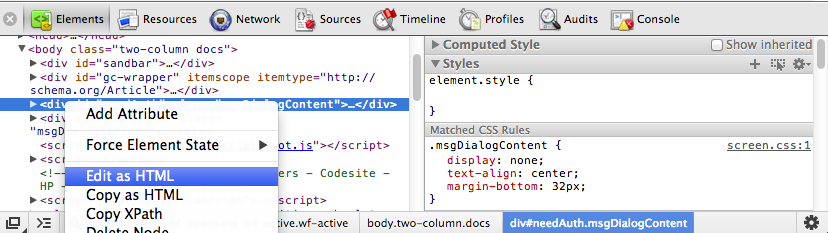
#### Dev Tools Window

The DevTools are organised into task-oriented groups in the toolbar at the top of the window. Each toolbar item and corresponding panel let you work with a specific type of page or app information, including DOM elements, resources, and sources.



#### Inspecting DOM and Tools

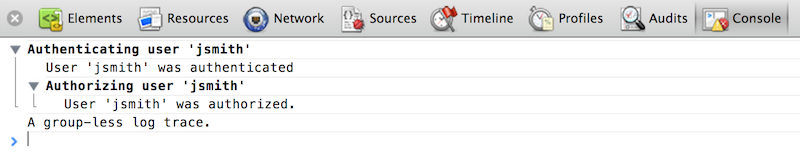
The Elements panel lets you see everything in one DOM tree, and allows inspection and on-the-fly editing of DOM elements. You will often visit the Elements tabs when you need to identify the HTML snippet for some aspect of the page. For example, you may be curious if an image has an HTML id attribute, and what that attribute's value is.



#### Working with The console

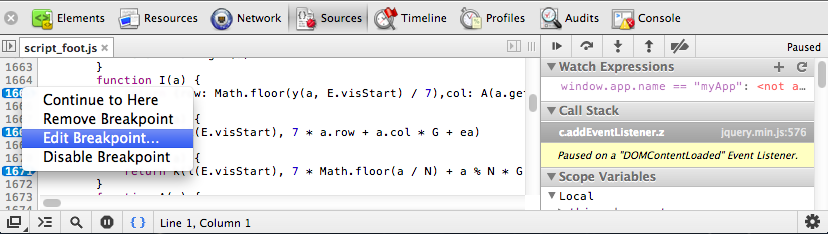
The JavaScript Console provides two primary functions for developers testing web pages and applications:

* A place to log diagnostic information using methods provided by the Console API, such as console.log(), or console.profile().
* A shell prompt where you can enter commands and interact with the document and the Chrome DevTools. You can evaluate expressions directly in the Console, and can also use the methods provided by the Command Line API, such as $() command for selecting elements, or profile() to start the CPU profiler.



#### Debugging JavaScript

As the complexity of JavaScript applications increase, developers need powerful debugging tools to help quickly discover the cause of an issue and fix it efficiently. The Chrome DevTools include a number of useful tools to help make debugging JavaScript less painful.



## 8.2 Repository

### 8.2.1 SVN



Apache Subversion (often abbreviated SVN, after the command name svn) is a software versioning and revision control system distributed under an open source license. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System (CVS).

The open source community has used Subversion widely: for example in projects such as Apache Software Foundation, Free Pascal, FreeBSD, GCC, Mono and SourceForge. Google Code also provides Subversion hosting for their open source projects. BountySource systems use it exclusively. CodePlex offers access to Subversion as well as to other types of clients.

#### Features

* Commits as true atomic operations (interrupted commit operations in CVS would cause repository inconsistency or corruption).
* Renamed/copied/moved/removed files retain full revision history.
* The system maintains versioning for directories, renames, and file metadata (but not for timestamps). Users can move and/or copy entire directory-trees very quickly, while retaining full revision history.
* Versioning of symbolic links.
* Native support for binary files, with space-efficient binary-diff storage.
* Apache HTTP Server as network server, WebDAV/Delta-V for protocol. There is also an independent server process called svnserve that uses a custom protocol over TCP/IP.
* Branching as a cheap operation, independent of file size (though Subversion itself does not distinguish between a branch and a directory)
* Natively client–server, layered library design.
* Client/server protocol sends diffs in both directions.
* Costs proportional to change size, not to data size.
* Parsable output, including XML log output.
* Open source licensed — Apache License in the projected 1.7 release; prior versions use a derivative of the Apache Software License, v1.1
* Internationalized program messages.
* File locking for unmergeable files ("reserved checkouts").
* Path-based authorization.
* Language bindings for C#, PHP, Python, Perl, Ruby, and Java.
* Full MIME support — users can view or change the MIME type of each file, with the software knowing which MIME types can have their differences from previous versions shown.
* Merge tracking - Merges between branches will be tracked, this allows automatically merging between branches without telling Subversion what (doesn't) need to be merged.
* Changelists to organise commits into commit groups

#### Repository types

Subversion offers two types of repository storage.

#### Berkeley DB

The original development of Subversion used the Berkeley DB package. Subversion has some limitations with Berkeley DB usage when a program that accesses the database crashes or terminates forcibly. No data loss or corruption occurs, but the repository remains offline while Berkeley DB replays the journal and cleans up any outstanding locks. The safest way to use Subversion with a Berkeley DB repository involves a single server-process running as one user (instead of through a shared filesystem).

#### FSFS

In 2004, a new storage subsystem was developed and named FSFS. It works faster (than the Berkeley DB backend) on directories with a large number of files and takes less disk space, due to less logging.

Beginning with Subversion 1.2, FSFS became the default data store for new repositories.

The etymology of "FSFS" is based on Subversion's use of the term "filesystem" for its repository storage system. FSFS stores its contents directly within the operating system's filesystem, rather than a structured system like Berkeley DB. Thus, it is a "[Subversion] FileSystem atop the FileSystem".

#### Repository access

Access to Subversion repositories can take place by:

1. Local filesystem or network filesystem,[6] accessed by client directly. This mode uses the file:///path access scheme.
2. WebDAV/Delta-V (over http or https) using the mod\_dav\_svn module for Apache 2. This mode uses the http://host/path access scheme or https://host/path for secure connections using ssl.
3. Custom "svn" protocol (default port 3690), using plain text or over TCP/IP. This mode uses either the svn://host/path access scheme for unencrypted transport or svn+ssh://host/path scheme for tunneling over ssh.

All three means can access both FSFS and Berkeley DB repositories.

### 8.2.2 Assembla SVN

Assembla is a company that provides cloud-based tools to organize and manage open-source and commercial software development. Assembla is used by over 500,000 users in 100 countries.

Project Workspaces, hosted through Assembla, feature project management software elements such as task management, issue tracking, cardwall (Kanban), agile project management, subversion repositories and scrum meetings.[8] The service caters mostly to software developers, featuring Subversion, Git, Perforce and Mercurial source control,[9] but one third of its tools serve general collaboration tools such as wikis and message boards.[10] It hosts its toolset on dedicated servers

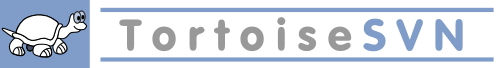
#### Free Subversion Repository:

Assembla’s Subversion hosting leads the industry in reliability, accessibility, features, and innovation. Our repositories are hosted on triple-redundant server clusters with off-site backups, so you can sleep well at night knowing your data is safe and always available.

Key features of Assembla’s Subversion hosting include:

* On-demand repositories, so you can be up and running within seconds.
* An integrated code browser with full syntax highlighting.
* Inline code commenting for lines of code and entire changesets.
* Tools to import and export repositories with just a few clicks.
* A tool to publish static websites in seconds.
* Communication protected by 128-bit SSL encryption.
* Optional daily backups to your private Amazon S3 account
* Full integration with Assembla’s ticket tool, activity stream, email notifications, and collaboration tools.

### 8.2.3 Tortise SVN



TortoiseSVN is a Subversion client, implemented as a Microsoft Windows shell extension. It is free software released under the GNU General Public License.

In Windows Explorer, besides showing context menu items for Subversion commands, it also provides icon overlay that indicates the status of Subversion working copies.

TortoiseSVN won the SourceForge.net 2007 Community Choice Award for Best Tool or Utility for Developers. The TortoiseMerge utility comes with the TortoiseSVN distribution to visually compare differences between two files. It is available from the Tigris.org website.

TortoiseSVN can be integrated into Microsoft Visual Studio by using a third-party plugin such as VisualSVN. VsTortoise, and AnkhSVN. A third-party repository monitoring application using TortoiseSVN was named SVN-Monitor, and then evolved into Vercue in 2011.

### 8.2.4 Checkouts and Commit

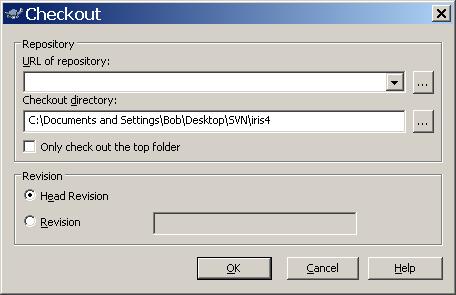
#### Checkouts and Commits

When we want to work with SVN version-controlled source code, we must first 'check out' the current version of the code (or possibly an older version, if necessary). 'Check out' describes the process of the TortoiseSVN client connecting to the SVN server, and downloading a version of the code in a repository. Once the code is checked out, it can be worked with just like un-versioned code. After some milestone has been reached (or the workday has ended), the updated code can then be 'committed' back to the SVN repository as a new version of the source code, and subsequent attempts to check out the latest version of the code will acquire this newer, updated version.

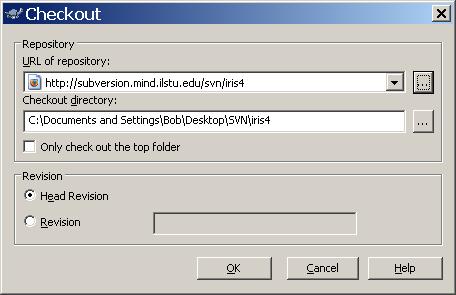
#### The Checkout Operation

In the following example, we will be 'checking out' the latest version of the iris4 source code (just replace the name iris4 with the repository you wish to access).

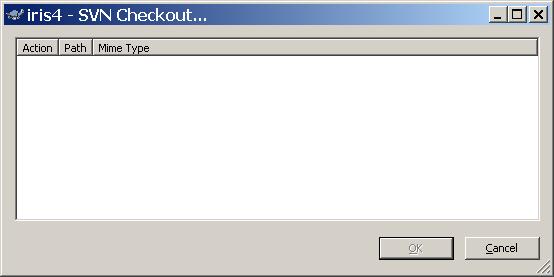
1. Download and install the latest version of **[TortoiseSVN](http://tortoisesvn.net/downloads" \t "_blank)**.
2. Restart your computer.
3. Now that TortoiseSVN is installed and ready to go, we'll use it to checkout the latest version of the Iris.4 Mobile Robot source code. Create a new folder called *SVN* somewhere on your system (e.g. in your My Documents folder, or on the Desktop). Within the SVN folder, create a second new folder, this one named *iris4*.
4. Now, right-click on the iris4 folder. In the shell context menu that pops up, select *SVN Checkout...*.
5. A window should pop up that looks like this:



1. For the *URL of repository:* field, enter:http://subversion.mind.ilstu.edu/svn/iris4
2. For the *Checkout directory:* field, click on the ellipses button Elipses Fieldand surf to the iris4 folder you just right-clicked on, and select it.
3. Your SVN client should now look something like this:

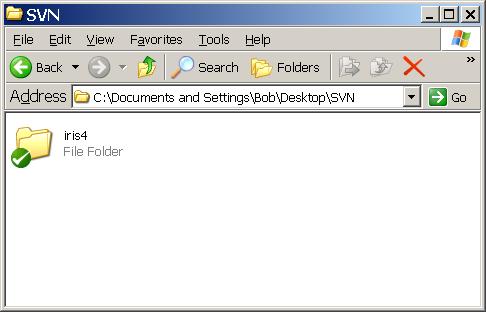


1. Click OK on the client GUI. If you have not previously saved your UliD and password for this repository with TortoiseSVN, you will be prompted for it now. Enter your UliD and password, check the Check box to save them, and click OK. You should see a new screen like this:



1. Assuming that your credentials checked out, and you entered in all of the connection information correctly, a progress list of files being checked out will stream down the Checkout Results screen.
2. Once TortoiseSVN is finished processing your request, the OKbutton will become active; click it.

You now have a whole bunch of robot source code on your hard drive. Go to the SVN folder. The *iris4* folder should now have a special SVN icon:



The green check means that the folder contains SVN files. Go into the folder and browse its contents; you should see a whole bunch of C++ and Java code in a variety of folders.

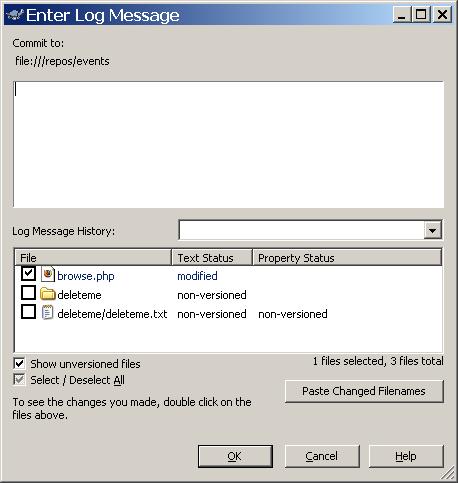
Please note that you could have just check out a sub-folder of the repository, or even a single file. To do this, just modify the *URL of repository...* field, and run the checkout operation as usual. For example, to get just the Java source code from the iris4 repository, enter *http://subversion.mind.ilstu.edu/svn/iris4/java* in the *URL of repository...* field and click **OK**. Go ahead, try it out yourself!

#### The Commit Operation

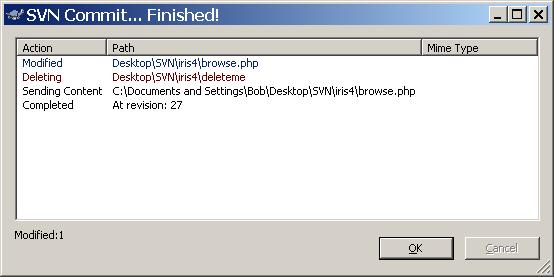
In the following example, we will be committing a change to the repository. Note: please don't actually make a commit to the repository for this tutorial - just read along! (We don't want the iris4 repo to get messed up with a bunch of 'SVN practice' commits.)

If you have modified any of the files you have checked out, added new files to the folder (or a sub-folder) where you have versioned files, or if you have deleted versioned files, you will have to commit these changes to the SVN repository to try and make them stick. I say *try* here because it is possible that the commit operation will fail if your changes conflict with someone else's changes (more on this below, under [**Conflicts**](http://www.mind.ilstu.edu/research/complete_inactive/iris4/iris40/developers/svntutorial/#conflicts).

1. You may have made changes to just one file, or just a set of files within a subfolder of the local snapshot of the repository, or you may have made changes all over the place, added files, deleted folders, etc. You can commit these changes by right-clicking on the file, sub-folder, or repository folder (whichever will cover the entire set of files/folders you have changed), and select *SVN Commit...*. You should see a screen pop up like the following:



1. The above screen will show you a list of all the modified files, and will include other information depending on whether you've been deleting files/folders, renamed them, or added them to the repository. In the upper text box, write a comment that is concise, yet fully describes the important changes you have made to the repository (for example, you might comment, "Adding interface.h and fixed the I/O bug in main() in file main.cpp"). Only use accurate, descriptive comments so others can understand how the new version of code you are creating differs from the previous version!
2. You may need to interact with the commit interface further; more info on deleting, renaming, and adding files and folders is available[**below**](http://www.mind.ilstu.edu/research/complete_inactive/iris4/iris40/developers/svntutorial/#otherops). Once you are done, (or, if you just updated a versioned file and did nothing else), click **OK**. Once the operation completes, you should see something like the following:

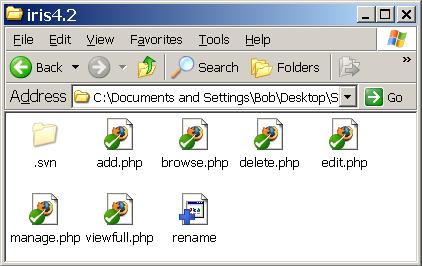


1. Click **OK**, you have successfully committed a change to the repository. To see your change, go ahead and delete the entire iris4 folder. Create a new iris4 folder in the same location as the old one (the new folder will not be checked). Now, checkout the iris4 repository again. Go to the file(s) you updated, and you will be able to see your changes.

### 8.2.5 Other Operations

#### Renaming Files and Folders

To rename a file or a folder you must first checkout the file or folder to your machine. Once it's on your machine, right-click on the file or folder, and select the menu option *SVN Rename...*. Type in the new name, and the icon for the file or folder will change to:



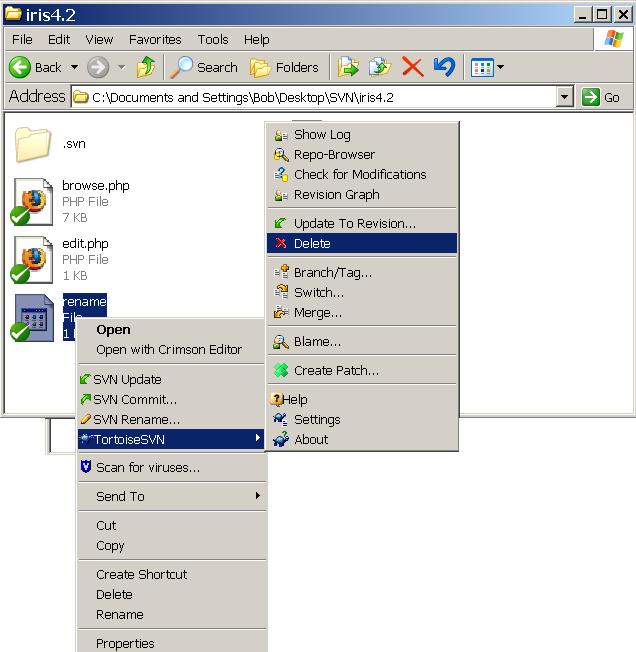
Now, just run a Commit and the repository will be updated with the new name, and any other changes you've made. If you just want to commit the name change (and not changes to other files/folders), right-click on the file you renamed and select the *SVN Commit...* menu option.

##### Please Note:

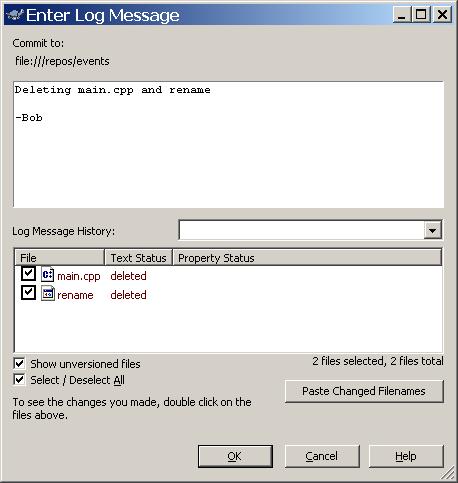
SVN handles renaming in a strange way; when you rename a file or folder, what actually happens is a new file/folder is created and added to the repository with the new name. However, the original file/folder remains in the repository. Thus if you renamed the file main\_c.cpp to main.cpp, ran a Commit, then ran a Checkout of main\_c.cpp and main.cpp's containing folder, both files would be downloaded from the repository. So... once you've renamed the file/folder, you should **[delete](http://www.mind.ilstu.edu/research/complete_inactive/iris4/iris40/developers/svntutorial/" \l "deleting)**the original from the repository.

#### Deleting Files and Folders

To delete a file or folder, simply right-click on it, and select the *Delete...*option from the *TortoiseSVN* menu:



Once the operation completes, the folder will be deleted from your hard drive. To make the delete stick, run a Commit, and make sure the check boxes are checked for deleting the items you want deleted:



When the operation completes, the repository will be updated to a new version and the file(s) and folder(s) you deleted will not be contained in the new version.

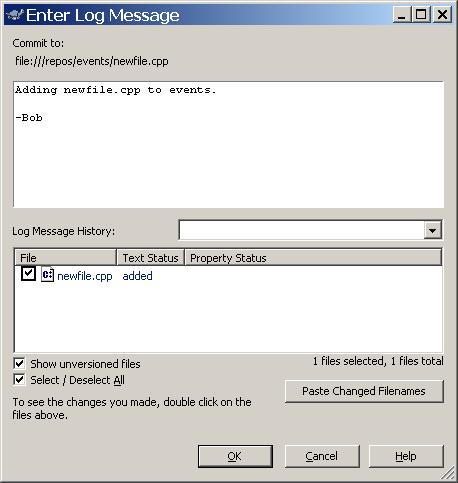
#### Adding Files and Folders

To add a file or folder, check out the repo (if you haven't already done so). move the new file(s) and folder(s) to the location you want them in the repository (for e.g. to add the file newfile.cpp to the iris4/java/newclient/ folder, move newfile.cpp to that folder). Now, with everything in its place, right-click on the file(s) and folder(s) you want to add to the repository, and select the *SVN Add...* menu option. You will be prompted with a confirmation dialog box:



If everything is as you want it, click **OK**. You will then see the TortoiseSVN Add dialog box which shows the progress of the operation. When it is complete, hit **OK** to continue.

To make the change stick, run a Commit and make sure the check boxes are checked for adding the items you want to add:



When everything is as you want it, click **OK**. Once the Commit operation completes, the file(s) and folder(s) will have been added to the repository, and the version will have increased.

#### Updates

At any time you can update your local snapshot of the repository to the latest version available by running an Update Operation. To do this, just right-click on a folder containing versioned files and folders in it, and select the *SVN Update...* menu option.

##### Please Note:

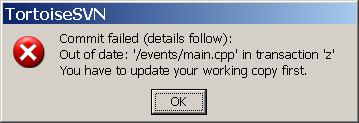
When you run an update, only the files already checked out will be updated (or deleted, if they were deleted in the repository since you last updated) - if new files have been added to the repository, or you if you have deleted a file (or renamed a file), you will not get all of the files in the current version. Again, only the files already on your hard drive will be touched, and they can only be deleted or overwritten with the latest version of the file.

##### Warning...

Be very careful when you update you local version of the repository - any changes you have made to versioned files being updated will be**completely** wiped out. If you want to update a folder with modified versioned files in it, first run a Commit to commit your changes, then run the update. Also note that entire files/folders can be deleted when running an update (if they were deleted in the repository since you last updated).

#### Conflicts

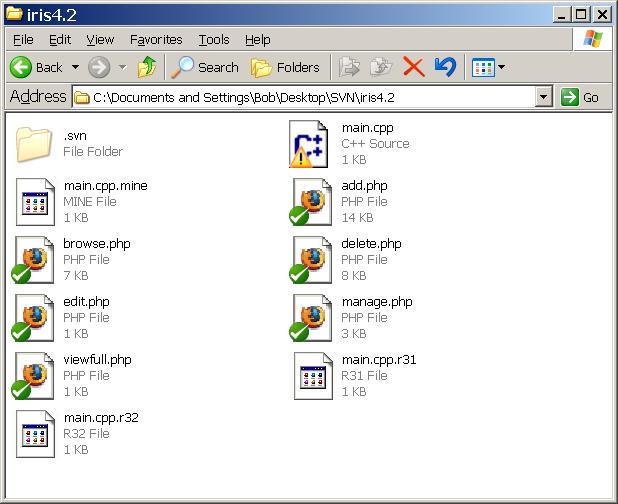
Conflicts primarily occur in a situation like the following. Suppose you have been working on a file, main.cpp, in the irsi4 repository using revision 31. While you are busily programming away, someone else commits a modified version main.cpp to the same repository (thus updating the code to revision 32). Now main.cpp v32 in the SVN repository is different from the version you have been working on (v31). When you try and commit your modified file, main.cpp, you will get an error like this:



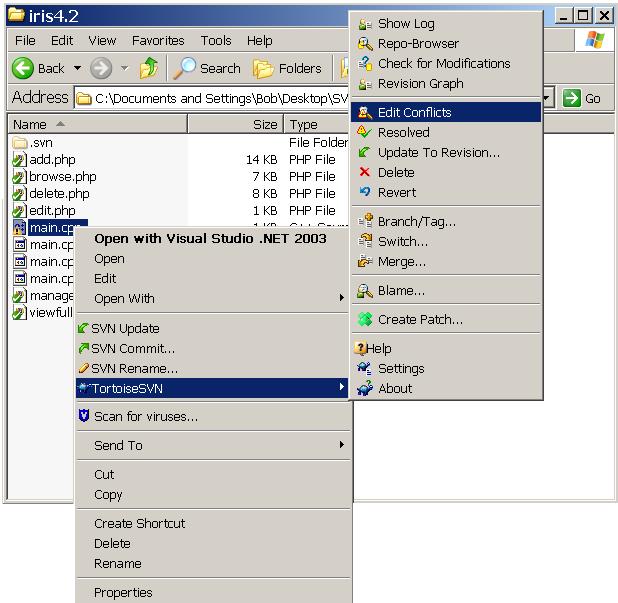
You then run an Update operation, and you will see the following:



After you click **OK**, the folder containing main.cpp would now have several new, non-versioned files in it:



You can now right-click on the file main.cpp, and under the *TortoiseSVN*option, select *Edit Conflicts*:



For details on using this Merge program, see the help file (just hit **F1**). Basically, you need to decide which parts of the file should go in the new 'merged' version of the file. Once you've done this, you can mark the file as merged (either from within the merge program, or from the TortoiseSVN context menu (right-click on the file in question)). You can now commit the changes to the repository; the conflict is resolved!

## 8.3 Producticity Tools

### 8.3.1 Google Hangouts

If your team is scattered all over the world, a Hangout can provide a simple way to come together to discuss ideas, collaborate, make decisions, and share documents.

If you’ve got an accountability partner for your business, and the two of you don’t live in the same city or town, Google Hangouts can be a great way to meet, discuss progress and problems, and brainstorm solutions. You can collaborate using Google Docs during your Hangout, or just share your screen to show your partner your latest project.

#### Chat face-to-face

Seeing someone's face can make all the difference, so catch up with the whole gang from your computer, phone or tablet device.

#### Host virtual meetings

Get work done faster in virtual meetings with powerful tools like screen sharing and Google Drive collaboration.

#### Broadcast to the world

Go live in front of a global audience, whether you're an aspiring artist, a global celebrity or a concerned citizen.

Brainstorm

Get Accountability

Demonstration

### 8.3.2 Google Drive

Google Drive is a file storage and synchronization service provided by Google, which enables user cloud storage, file sharing and collaborative editing. Google Drive is now the home of Google Docs, a suite of productivity applications, that offer collaborative editing on documents, spreadsheets, presentations, and more.

Google Drive gives all users 5 GB of cloud storage to start with.[5] A user can get additional storage, which is shared between Picasa and Google Drive.

Google Docs is a free web-based office suite offered by Google within its Google Drive service. It also was a storage service but has since been replaced by Google Drive [4] It allows users to create and edit documents online while collaborating in real-time with other users. Google Docs combines the features of Writely and Spreadsheets with a presentation program incorporating technology designed by Tonic Systems.

Google Docs (now housed in Google Drive) is a free, Web-based office suite and data storage service offered by Google. It allows users to create and edit documents online while collaborating in real-time with other users. Google Docs combines the features of Writely and Spreadsheets with a presentation program incorporating technology designed by Tonic Systems. This extension or replacement of Google Docs called Google Drive was opened to the public on April 24, 2012.

# 9.0 Results Discussions

## 9.1 Work Area

## 9.2 Data Structures

### 9.2.1 queryString

In the Work Area, when the user makes selection, Then we need to get the list of selected tables and selected columns. This information is then passed through AJAX to backend, where it is stored into database.

Whenever the Query is retrieved or displayd, this queryString is processed to obtain a query and executed on the database.

Structure of queryString, storing the list of Selected Columns List

queryString{

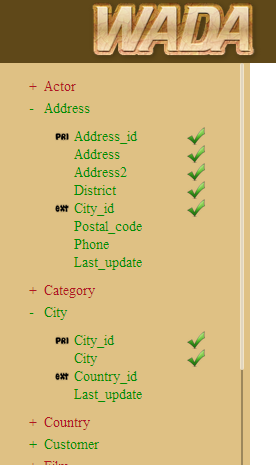
[0] : { [0] : <tableName>, [1] : <columnName> },

.

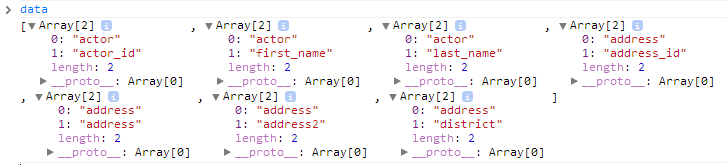
.

}

#### Example



The Resultatnt queryString for the following selection is



The JSON Array is compressed into a Strong using javascript function JSON.stringify() and passed to the backend.

### [["address","address\_id"],["address","address"],["address","address2"],["address","district"],["address","city\_id"],["city","city\_id"],["city","city"]]

### 9.2.2 metaArray

This Array stores the Primary Key and Foreign Key Information, that is used to mark columns as primary and external in the work area.

#### Structure

The Structure consists as metaArray is an Array of (Array for Primary Key and Foreign Key).

Primary Key is represente as 3 Column Array

array (size=3)

0 => string <PRI>

1 => string <tableName>

2 => string <columnName>

Foreign Keys are represented as a 5 Column Array

array (size=5)

0 => string <FOR>

1 => string <table>

2 => string <key>

3 => string <referingTable>

4 => string <referingKey>

#### Sample MetArray for Sakila Database

array (size=40)

0 =>

array (size=3)

0 => string 'PRI' (length=3)

1 => string 'actor' (length=5)

2 => string 'actor\_id' (length=8)

1 =>

array (size=3)

0 => string 'PRI' (length=3)

1 => string 'address' (length=7)

2 => string 'address\_id' (length=10)

2 =>

array (size=5)

0 => string 'FOR' (length=3)

1 => string 'address' (length=7)

2 => string 'city\_id' (length=7)

3 => string 'city' (length=4)

4 => string 'city\_id' (length=7)

3 =>

array (size=3)

0 => string 'PRI' (length=3)

1 => string 'category' (length=8)

2 => string 'category\_id' (length=11)

4 =>

array (size=3)

0 => string 'PRI' (length=3)

1 => string 'city' (length=4)

2 => string 'city\_id' (length=7)

.

.

.

38 =>

array (size=5)

0 => string 'FOR' (length=3)

1 => string 'store' (length=5)

2 => string 'address\_id' (length=10)

3 => string 'address' (length=7)

4 => string 'address\_id' (length=10)

39 =>

array (size=5)

0 => string 'FOR' (length=3)

1 => string 'store' (length=5)

2 => string 'manager\_staff\_id' (length=16)

3 => string 'staff' (length=5)

4 => string 'staff\_id' (length=8)

### 9.2.3 metaArray2

To make immediate search of a key whether it is primary or not and if it is a foreign key then getting its foreign key references where very costly to perform on metaArray. So we had to develop a new data structure, that can speed up the above operations.

#### Structure

It is an associative Array. Where

The key represented as <tableName>.<columnName>

And its value represents

<PRI> if it is a primary key or

<tableName>.<columnName> to the table this foreign key is referring to

#### Sample metaArray2 for Sakila Database

array (size=36)

'actor.actor\_id' => string 'PRI' (length=3)

'address.address\_id' => string 'PRI' (length=3)

'address.city\_id' => string 'city.city\_id' (length=12)

'category.category\_id' => string 'PRI' (length=3)

'city.city\_id' => string 'PRI' (length=3)

'city.country\_id' => string 'country.country\_id' (length=18)

'country.country\_id' => string 'PRI' (length=3)

'customer.customer\_id' => string 'PRI' (length=3)

'customer.address\_id' => string 'address.address\_id' (length=18)

'customer.store\_id' => string 'store.store\_id' (length=14)

'film.film\_id' => string 'PRI' (length=3)

'film.language\_id' => string 'language.language\_id' (length=20)

'film.original\_language\_id' => string 'language.language\_id' (length=20)

'film\_actor.actor\_id' => string 'actor.actor\_id' (length=14)

'film\_actor.film\_id' => string 'film.film\_id' (length=12)

'film\_category.film\_id' => string 'film.film\_id' (length=12)

'film\_category.category\_id' => string 'category.category\_id' (length=20)

'film\_text.film\_id' => string 'PRI' (length=3)

'inventory.inventory\_id' => string 'PRI' (length=3)

'inventory.film\_id' => string 'film.film\_id' (length=12)

'inventory.store\_id' => string 'store.store\_id' (length=14)

'language.language\_id' => string 'PRI' (length=3)

'payment.payment\_id' => string 'PRI' (length=3)

'payment.customer\_id' => string 'customer.customer\_id' (length=20)

'payment.rental\_id' => string 'rental.rental\_id' (length=16)

'payment.staff\_id' => string 'staff.staff\_id' (length=14)

'rental.rental\_id' => string 'PRI' (length=3)

'rental.customer\_id' => string 'customer.customer\_id' (length=20)

'rental.inventory\_id' => string 'inventory.inventory\_id' (length=22)

'rental.staff\_id' => string 'staff.staff\_id' (length=14)

'staff.staff\_id' => string 'PRI' (length=3)

'staff.address\_id' => string 'address.address\_id' (length=18)

'staff.store\_id' => string 'store.store\_id' (length=14)

'store.store\_id' => string 'PRI' (length=3)

'store.address\_id' => string 'address.address\_id' (length=18)

'store.manager\_staff\_id' => string 'staff.staff\_id' (length=14)

## 9.3 Green-Red Concept

The User can execute simple queries where he selects columns from some table and execute them. He can also execute join queries between two or more tables, not all tables can be joined, only some can be joined.

When the user makes a selection of columns, the other tables with which he can perform join(there is a primary key and foreign key relation between them) becomes green, and the other tables with which he cannot perform join become red(there is no primary key and foreign key relation between them).

We have written a reflectState() javascript function, that is excuted everytime a selection or deselection of a column is done.

The Function scans all the selections made. Then it generates table array, which contains all the tables that are green.

arrTables contain

* The Table Name of Columns Selected
* For each foreign key selected, its referencing table is included in the arrtables
* For every Primary Key selected, any external tables referring to it are included.

This A

## 9.4 Generating Joins

## 9.4 Eliminating Extra Join Column

## 9.6 Methods Library

## 9.7 Slick Grid

## 9.8 ColorBox

## 9.9 jQuery mCustomScrollbar

# 10.0 Deployment

## 8.1 Production Environment

# 11.0 Testing

## 9.1 Database

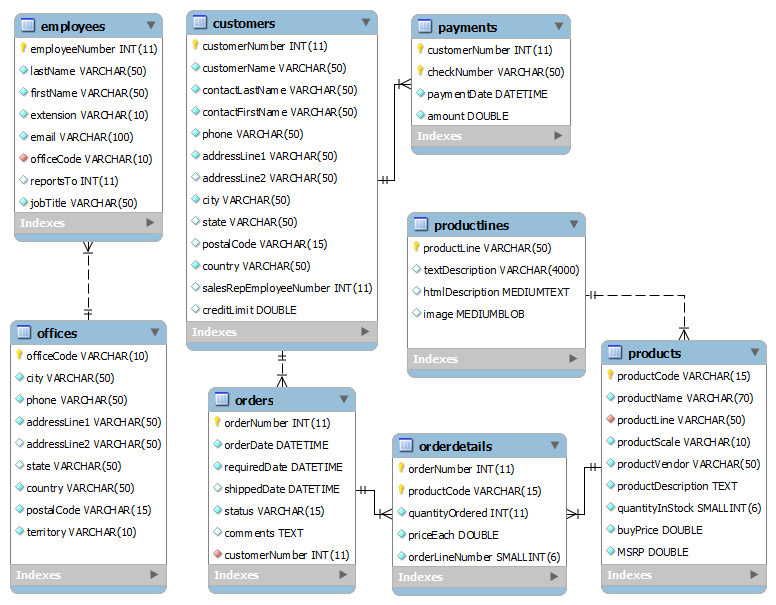
### 9.1.1 Classic Models

The classic models database is a retailer of scale models of classic cars database. It contains typical business data such as customers, products, sale orders, sale order line items and etc. It is used in our MySQL tutorials to demonstrate many features of MySQL.

The sample database schema consists of the following tables:

* **Customers**: stores customer’s data
* **Products**: stores a list of scale model cars.
* **ProductLines**: stores a list of product line category.
* **Orders**: stores orders placed by customers.
* **OrderDetails**: stores order line items in each order.
* **Payments**: stores payments made by customers based on their account.
* **Employees**: stores all employee information include organization unit structure such as who reports to whom.
* **Offices**: stores sale office data.

#### Schema



### 9.1.2 Sakila

The Sakila sample database was developed by Mike Hillyer, a former member of the MySQL AB documentation team, and is intended to provide a standard schema that can be used for examples in books, tutorials, articles, samples, and so forth. Sakila sample database also serves to highlight the latest features of MySQL such as Views, Stored Procedures, and Triggers.

### 9.1.3 Sakila Tables

#### The actor Table

The actor table lists information for all actors.

The actor table is joined to the [film](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-film.html) table by means of the [film\_actor](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-film_actor.html" \o "5.1.8. The film_actor Table) table.

##### Columns

* actor\_id: A surrogate primary key used to uniquely identify each actor in the table.
* first\_name: The actor's first name.
* last\_name: The actor's last name.
* last\_update: The time that the row was created or most recently updated.

#### The address Table

The address table contains address information for customers, staff, and stores.

The address table primary key appears as a foreign key in the [customer](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-customer.html), [staff](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-staff.html), and [store](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-store.html) tables.

##### Columns

* address\_id: A surrogate primary key used to uniquely identify each address in the table.
* address: The first line of an address.
* address2: An optional second line of an address.
* district: The region of an address, this may be a state, province, prefecture, etc.
* city\_id: A foreign key pointing to the [city](http://dev.mysql.com/doc/sakila/en/sakila-structure-tables-city.html) table.
* postal\_code: The postal code or ZIP code of the address (where applicable).
* phone: The telephone number for the address.
* last\_update: The time that the row was created or most recently updated.

#### The category Table

The category table lists the categories that can be assigned to a film.

The category table is joined to the film table by means of the film\_category table.

##### Columns

* category\_id: A surrogate primary key used to uniquely identify each category in the table.
* name: The name of the category.
* last\_update: The time that the row was created or most recently updated.

#### The city Table

The city table contains a list of cities.

The city table is referred to by a foreign key in the address table and refers to the country table using a foreign key.

##### Columns

* city\_id: A surrogate primary key used to uniquely identify each city in the table.
* city: The name of the city.
* country\_id: A foreign key identifying the country that the city belongs to.
* last\_update: The time that the row was created or most recently updated.

#### The country Table

The country table contains a list of countries.

The country table is referred to by a foreign key in the city table.

##### Columns

* country\_id: A surrogate primary key used to uniquely identify each country in the table.
* country: The name of the country.
* last\_update: The time that the row was created or most recently updated.

#### The customer Table

The customer table contains a list of all customers.

The customer table is referred to in the payment and rental tables and refers to the address and store tables using foreign keys.

##### Columns

* customer\_id: A surrogate primary key used to uniquely identify each customer in the table.
* store\_id: A foreign key identifying the customer's “home store.” Customers are not limited to renting only from this store, but this is the store they generally shop at.
* first\_name: The customer's first name.
* last\_name: The customer's last name.
* email: The customer's email address.
* address\_id: A foreign key identifying the customer's address in the address table.
* active: Indicates whether the customer is an active customer. Setting this to FALSE serves as an alternative to deleting a customer outright. Most queries should have a WHERE active = TRUE clause.
* create\_date: The date the customer was added to the system. This date is automatically set using a trigger during an INSERT.
* last\_update: The time that the row was created or most recently updated.

#### The film Table

The film table is a list of all films potentially in stock in the stores. The actual in-stock copies of each film are represented in the inventory table.

The film table refers to the language table and is referred to by the film\_category, film\_actor, andinventory tables.

##### Columns

* film\_id: A surrogate primary key used to uniquely identify each film in the table.
* title: The title of the film.
* description: A short description or plot summary of the film.
* release\_year: The year in which the movie was released.
* language\_id: A foreign key pointing at the language table; identifies the language of the film.
* original\_language\_id: A foreign key pointing at the language table; identifies the original language of the film. Used when a film has been dubbed into a new language.
* rental\_duration: The length of the rental period, in days.
* rental\_rate: The cost to rent the film for the period specified in the rental\_duration column.
* length: The duration of the film, in minutes.
* replacement\_cost: The amount charged to the customer if the film is not returned or is returned in a damaged state.
* rating: The rating assigned to the film. Can be one of: G, PG, PG-13, R, or NC-17.
* special\_features: Lists which common special features are included on the DVD. Can be zero or more of:Trailers, Commentaries, Deleted Scenes, Behind the Scenes.
* last\_update: The time that the row was created or most recently updated.

#### The film\_actor Table

The film\_actor table is used to support a many-to-many relationship between films and actors. For each actor in a given film, there will be one row in the film\_actor table listing the actor and film.

The film\_actor table refers to the film and actor tables using foreign keys.

##### Columns:

* actor\_id: A foreign key identifying the actor.
* film\_id: A foreign key identifying the film.
* last\_update: The time that the row was created or most recently updated.

#### The film\_category Table

The film\_category table is used to support a many-to-many relationship between films and categories. For each category applied to a film, there will be one row in the film\_category table listing the category and film.

The film\_category table refers to the film and category tables using foreign keys.

##### Columns:

* film\_id: A foreign key identifying the film.
* category\_id: A foreign key identifying the category.
* last\_update: The time that the row was created or most recently updated.

#### The film\_text Table

The film\_text table is the only table in the Sakila sample database that uses the MyISAM storage engine. This table is provided to allow for full-text searching of the titles and descriptions of the films listed in the film table.

The film\_text table contains the film\_id, title and description columns of the film table, with the contents of the table kept in synchrony with the film table by means of triggers on the film table's INSERT,UPDATE and DELETE operations (see Section 5.5, “Triggers”).

##### Columns

* film\_id: A surrogate primary key used to uniquely identify each film in the table.
* title: The title of the film.
* description: A short description or plot summary of the film.

The contents of the film\_text table should never be modified directly. All changes should be made to the filmtable instead.

#### The inventory Table

The inventory table contains one row for each copy of a given film in a given store.

The inventory table refers to the film and store tables using foreign keys and is referred to by the rentaltable.

##### Columns

* inventory\_id: A surrogate primary key used to uniquely identify each item in inventory.
* film\_id: A foreign key pointing to the film this item represents.
* store\_id: A foreign key pointing to the store stocking this item.
* last\_update: The time that the row was created or most recently updated.

#### The language Table

The language table is a lookup table listing the possible languages that films can have for their language and original language values.

The language table is referred to by the film table.

##### Columns

* language\_id: A surrogate primary key used to uniquely identify each language.
* name: The English name of the language.
* last\_update: The time that the row was created or most recently updated.

#### The payment Table

The payment table records each payment made by a customer, with information such as the amount and the rental being paid for (when applicable).

The payment table refers to the customer, rental, and staff tables.

##### Columns

* payment\_id: A surrogate primary key used to uniquely identify each payment.
* customer\_id: The customer whose balance the payment is being applied to. This is a foreign key reference to the customer table.
* staff\_id: The staff member who processed the payment. This is a foreign key reference to the staff table.
* rental\_id: The rental that the payment is being applied to. This is optional because some payments are for outstanding fees and may not be directly related to a rental.
* amount: The amount of the payment.
* payment\_date: The date the payment was processed.
* last\_update: The time that the row was created or most recently updated.

#### The rental Table

The rental table contains one row for each rental of each inventory item with information about who rented what item, when it was rented, and when it was returned.

The rental table refers to the inventory, customer, and staff tables and is referred to by the payment table.

##### Columns

* rental\_id: A surrogate primary key that uniquely identifies the rental.
* rental\_date: The date and time that the item was rented.
* inventory\_id: The item being rented.
* customer\_id: The customer renting the item.
* return\_date: The date and time the item was returned.
* staff\_id: The staff member who processed the rental.
* last\_update: The time that the row was created or most recently updated.

#### The staff Table

The staff table lists all staff members, including information on email address, login information, and picture.

The staff table refers to the store and address tables using foreign keys, and is referred to by the rental,payment, and store tables.

##### Columns

* staff\_id: A surrogate primary key that uniquely identifies the staff member.
* first\_name: The first name of the staff member.
* last\_name: The last name of the staff member.
* address\_id: A foreign key to the staff member's address in the address table.
* picture: A BLOB containing a photograph of the employee.
* email: The staff member's email address.
* store\_id: The staff member's “home store”. The employee can work at other stores but is generally assigned to the store listed.
* active: Whether this is an active employee. If employees leave their rows are not deleted from this table, instead this column is set to FALSE.
* username: The user name used by the staff member to access the rental system.
* password: The password used by the staff member to access the rental system. The password should be stored as a hash using the SHA1() function.
* last\_update: The time that the row was created or most recently updated.

#### The store Table

The store table lists all stores in the system. All inventory is assigned to specific stores, and staff and customers are assigned a “home store”.

The store table refers to the staff and address tables using foreign keys and is referred to by the staff,customer, and inventory tables.

##### Columns

* store\_id: A surrogate primary key that uniquely identifies the store.
* manager\_staff\_id: A foreign key identifying the manager of this store.
* address\_id: A foreign key identifying the address of this store.
* last\_update: The time that the row was created or most recently updated.

## 9.2 Test Queries

## 9.3 Testing for different Screen Resolutions

# 12.0 Conclusion / Future Enhancements / Recommendations

# 13.0 References / Bibiliography

[] dev.mysql.com

sample databases, mysql documentation syntax

[] SakilaDB Reference - <http://dev.mysql.com/doc/sakila/en/index.html>

[]Classic Models Reernce - <http://www.mysqltutorial.org/mysql-sample-database.aspx>

[] Chrome Developer Tools - <https://developers.google.com/chrome-developer-tools/>

# 14.0 Appendices