**10. FILE DESIGN**

A ﬁle system provides the machinery to support the project tasks. At the highest level a ﬁle system is a way to organize, store, retrieve, and manage information on a permanent storage medium such as a disk. File systems manage permanent storage and form an integral part of all operating systems. There are many different approaches to the task of managing permanent storage. At one end of the spectrum are simple ﬁle systems that impose enough restrictions to inconvenience users and make using the ﬁle system difficult. In deciding what type of ﬁling system is appropriate for a particular operating system, we must weigh the needs of the problem with the other constraints of the project. The two basic abstractions of files and directories form the basis of what a ﬁle system can operate on. There are many operations that a file system can perform on ﬁles and directories. All ﬁle systems must provide some basic level of support. Beyond the most basic ﬁle system primitives lay other features, extensions, and more sophisticated operations.

The Structure of a File is given the concept of a ﬁle, a ﬁle system may impose no structure on the ﬁle, or it may enforce a considerable amount of structure on the contents of the ﬁle. An unstructured, “raw” ﬁle, often referred to as a “stream of bytes,” literally has no structure. The ﬁle system simply records the size of the ﬁle and allows programs to read the bytes in any order or fashion that they desire. If a ﬁle system chooses to enforce a formal structure on ﬁles, it usually does so in the form of records. With the concept of records, a programmer specifies the size and format of the record, and then all I/O to that ﬁle must happen on record boundaries and be a multiple of the record length.

**11. PHP PROGRAMMING**

**PHP** is a [scripting](http://en.wikipedia.org/wiki/Scripting_programming_language) language designed to fill the gap between [SSI](http://en.wikipedia.org/wiki/Server_Side_Includes) (Server Side Includes) and [Perl](http://en.wikibooks.org/wiki/Perl), intended for the Web environment. Its principal application is the implementation of Web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. The newest version of PHP is 5.5 and it is heavily recommended to always use the newest version for better security, performance and of course features.

If you've been to a website that prompts you to login, you've probably encountered a server-side scripting language. Due to its market saturation, this means you've probably come across PHP. [PHP](http://en.wikipedia.org/wiki/PHP) was designed by [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) to display his resume online and to collect data from his visitors.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "**P**HP: **H**ypertext **P**reprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

Before you embark on the wonderful journey of Server Side Processing, it is recommended that you have a basic understanding of the [HyperText Markup Language (HTML)](http://en.wikibooks.org/wiki/HTML). But [PHP](http://en.wikipedia.org/wiki/PHP) can also be used to build [GUI](http://en.wikipedia.org/wiki/GUI)-driven applications for example by using [PHP-GTK](http://en.wikipedia.org/wiki/PHP-GTK).

PHP development began in 1994 when [Rasmus Lerdorf](http://en.wikipedia.org/wiki/Rasmus_Lerdorf) wrote a series of [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) binaries in C, which he used to maintain his [personal homepage](http://en.wikipedia.org/wiki/Personal_homepage). He extended them to add the ability to work with [web forms](http://en.wikipedia.org/wiki/Web_form) and to communicate with [databases](http://en.wikipedia.org/wiki/Database), and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

**11.1. PHP 6 AND UNICODE**

PHP received mixed reviews due to lacking native [Unicode](http://en.wikipedia.org/wiki/Unicode) support at the core language level. In 2005, a project headed by Andrei Zmievski was initiated to bring native Unicode support throughout PHP, by embedding the [International Components for Unicode](http://en.wikipedia.org/wiki/International_Components_for_Unicode) (ICU) library, and representing text strings as [UTF-16](http://en.wikipedia.org/wiki/UTF-16) internally. Since this would cause major changes both to the internals of the language and to user code, it was planned to release this as version 6.0 of the language, along with other major features then in development.

However, a shortage of developers who understood the necessary changes, and performance problems arising from conversion to and from UTF-16, which is rarely used in a web context, led to delays in the project. As a result, a PHP 5.3 release was created in 2009, with many non-Unicode features back-ported from PHP 6, notably namespaces. In March 2010, the project in its current form was officially abandoned, and a PHP 5.4 release was prepared containing most remaining non-Unicode features from PHP 6, such as traits and closure re-binding. Initial hopes were that a new plan would be formed for Unicode integration, but as of 2014 none has been adopted.

**11.2. PHP 7**

As of 2014, work is underway on a new major PHP version named PHP 7. There was some dispute as to whether the next major version of PHP was to be called PHP 6 or PHP 7. While the PHP 6 unicode experiment had never been released, a number of articles and book titles referenced the old PHP 6 name, which might have caused confusion if a new release were to reuse the PHP 6 name. After a vote, the name PHP 7 was chosen.

In terms of new language features, PHP 7 will add features such as return type declarations, which will complement its existing parameter type declarations. PHP 7 will also contain an improved variable syntax which is internally consistent and complete, resolving a long-standing issue in PHP, what will allow use of ->, [], (), {}, and :: operators with arbitrary meaningful left-hand-side expressions.

## 11.3. SYNTAX

The following [Hello world program](http://en.wikipedia.org/wiki/Hello_world_program) is written in PHP code embedded in an [HTML](http://en.wikipedia.org/wiki/HTML) document:

<!DOCTYPE html>

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php echo '<p>Hello World</p>'; ?>

</body>

</html>

However, as PHP does not need to be embedded in HTML or used with a web server, the simplest version of a Hello World program can be written like this, with the closing tag omitted as preferred in files containing pure PHP code (prior to PHP 5.4.0, this short syntax for echo() only works with the short\_open\_tag configuration setting enabled, while for PHP 5.4.0 and later it is always available):

<?= 'Hello world';

### 11.4. DATA TYPES

PHP stores whole numbers in a platform-dependent range, either a 64-bit or 32-bit [signed](http://en.wikipedia.org/wiki/Signed_number_representations) [integer](http://en.wikipedia.org/wiki/Integer_%28computer_science%29) equivalent to the [C-language long type](http://en.wikipedia.org/wiki/C_variable_types_and_declarations). Unsigned integers are converted to signed values in certain situations; this behavior is different from other programming languages. Integer variables can be assigned using decimal (positive and negative), [octal](http://en.wikipedia.org/wiki/Octal), [hexadecimal](http://en.wikipedia.org/wiki/Hexadecimal), and [binary](http://en.wikipedia.org/wiki/Binary_code) notations.

[Floating point](http://en.wikipedia.org/wiki/Floating_point) numbers are also stored in a platform-specific range. They can be specified using floating point notation, or two forms of [scientific notation](http://en.wikipedia.org/wiki/Scientific_notation). PHP has a native [Boolean](http://en.wikipedia.org/wiki/Boolean_datatype) type that is similar to the native Boolean types in [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Using the Boolean type conversion rules, non-zero values are interpreted as true and zero as false, as in [Perl](http://en.wikipedia.org/wiki/Perl) and C++.

**11.5. FUNCTION**

PHP has hundreds of functions provided by the core language functionality and thousands more available via various extensions; these functions are well documented in the online PHP documentation. However, the built-in library has a wide variety of naming conventions and associated inconsistencies, as described under [history](http://en.wikipedia.org/wiki/PHP#ORGANIC) above.

Additional functions can be defined by the developer:

function myAge($birthYear) // defines a function, this one is named "myAge"

{

$yearsOld = date('Y') - $birthYear; // calculates the age

return $yearsOld . ' year' . ($yearsOld != 1 ? 's' : ''); // returns the age in a descriptive form

}

echo 'I am currently ' . myAge(1981) . ' old.'; // outputs the text concatenated

// with the return value of myAge()

// As the result of this syntax, myAge() is called.

// In 2014, the output of this sample program will be 'I am currently 33 years old.'

## 11.6. IMPLEMNTATIONS

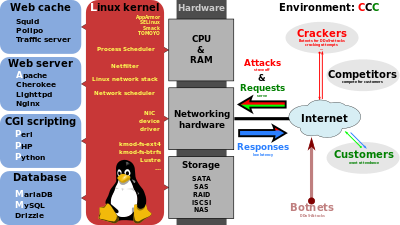
The original, only complete and most widely used PHP implementation is powered by the [Zend Engine](http://en.wikipedia.org/wiki/Zend_Engine) and known simply as PHP. To disambiguate it from other implementations, it is sometimes unofficially referred to as "Zend PHP". The Zend Engine [compiles](http://en.wikipedia.org/wiki/Compiler) PHP [source code](http://en.wikipedia.org/wiki/Source_code) on-the-fly into an internal format that it can execute, thus it works as an [interpreter](http://en.wikipedia.org/wiki/Interpreter_%28computing%29). It is also the "reference implementation" of PHP, as PHP has no formal specification, and so the semantics of Zend PHP define the semantics of PHP itself. Due to the complex and nuanced semantics of PHP, defined by how Zend works, it is difficult for competing implementations to offer complete compatibility.

## 11.7. INSTALLATION AND CONFIGURATION

There are two primary ways for adding support for PHP to a web server – as a native web server module, or as a CGI executable. PHP has a direct module interface called [Server Application Programming Interface](http://en.wikipedia.org/wiki/Server_Application_Programming_Interface) (SAPI), which is supported by many web servers including [Apache HTTP Server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), [Microsoft IIS](http://en.wikipedia.org/wiki/Microsoft_Internet_Information_Server), [Netscape](http://en.wikipedia.org/wiki/Netscape) (now defunct) and [iPlanet](http://en.wikipedia.org/wiki/IPlanet). Some other web servers, such as OmniHTTPd, support the [Internet Server Application Programming Interface](http://en.wikipedia.org/wiki/Internet_Server_Application_Programming_Interface) (ISAPI), which is a [Microsoft](http://en.wikipedia.org/wiki/Microsoft)'s web server module interface. If PHP has no module support for a web server, it can always be used as a [Common Gateway Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) or [FastCGI](http://en.wikipedia.org/wiki/FastCGI) processor; in that case, the web server is configured to use PHP's CGI executable to process all requests to PHP files.

When PHP is installed and used in [cloud](http://en.wikipedia.org/wiki/Cloud_computing) environments, [software development kits](http://en.wikipedia.org/wiki/Software_development_kit) (SDKs) are provided for using cloud-specific features. For example:

* [Amazon Web Services](http://en.wikipedia.org/wiki/Amazon_Web_Services) provides the AWS SDK for PHP
* [Windows Azure](http://en.wikipedia.org/wiki/Windows_Azure) can be used with the Windows Azure SDK for PHP.

**[](http://en.wikipedia.org/wiki/File:LAMP_software_bundle.svg)**

**12. MYSQL**

**MySQL** is (as of July 2013) the world's second most widely used [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) and most widely used open-source RDBMS. It is named after co-founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, My. The [SQL](http://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query Language](http://en.wikipedia.org/wiki/Structured_Query_Language).

The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB), now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29)." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: [TYPO3](http://en.wikipedia.org/wiki/TYPO3), [MODx](http://en.wikipedia.org/wiki/MODx), [Joomla](http://en.wikipedia.org/wiki/Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal) and other software. MySQL is also used in many high-profile, large-scale [websites](http://en.wikipedia.org/wiki/Website), including [Google](http://en.wikipedia.org/wiki/Google)[[13]](http://en.wikipedia.org/wiki/MySQL#cite_note-mysqlatgoogle-14)[[14]](http://en.wikipedia.org/wiki/MySQL#cite_note-15) (though not for searches), [Facebook](http://en.wikipedia.org/wiki/Facebook), [Twitter](http://en.wikipedia.org/wiki/Twitter), [Flickr](http://en.wikipedia.org/wiki/Flickr), and [YouTube](http://en.wikipedia.org/wiki/YouTube).

## 12.1. INTERFACE

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/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](http://en.wikipedia.org/wiki/MySQL_Workbench) is actively developed by Oracle, and is freely available for use.

### 12.1.1. GRAPHICAL

The official [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, [MySQL GUI Tools](http://en.wikipedia.org/wiki/MySQL_GUI_Tools). Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular [free and open source](http://en.wikipedia.org/wiki/Free_and_open_source_software) CommunityEdition which may be downloaded from the MySQL website, and the proprietary StandardEdition which extends and improves the feature set of the Community Edition.

### 12.2. COMMAND LINE

MySQL ships with many [command line](http://en.wikipedia.org/wiki/Command_line) tools, from which the main interface is 'mysql' client. Third parties have also developed tools to manage MySQL servers.

* MySQL Utilities – a set of utilities designed to perform common maintenance and administrative tasks. Originally included as part of the MySQL Workbench, the utilities are now a stand-alone download available from Oracle.
* Percona Toolkit – a cross-platform toolkit for MySQL, developed in [Perl](http://en.wikipedia.org/wiki/Perl).[[31]](http://en.wikipedia.org/wiki/MySQL#cite_note-32) Percona Toolkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Percona Toolkit is included with several [Linux](http://en.wikipedia.org/wiki/Linux) distributions such as [CentOS](http://en.wikipedia.org/wiki/CentOS) and [Debian](http://en.wikipedia.org/wiki/Debian), and packages are available for [Fedora](http://en.wikipedia.org/wiki/Fedora_%28operating_system%29) and [Ubuntu](http://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29) as well. Percona Toolkit was originally developed as Maatkit, but as of late 2011, Maatkit is no longer developed.

### 

### 12.3. PROGRAMMING

MySQL works on many [system platforms](http://en.wikipedia.org/wiki/System_platform), including [AIX](http://en.wikipedia.org/wiki/AIX_operating_system), [BSDi](http://en.wikipedia.org/wiki/BSD/OS), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), [HP-UX](http://en.wikipedia.org/wiki/HP-UX), [eComStation](http://en.wikipedia.org/wiki/EComStation), [i5/OS](http://en.wikipedia.org/wiki/IBM_i5/OS), [IRIX](http://en.wikipedia.org/wiki/IRIX), [Linux](http://en.wikipedia.org/wiki/Linux), [OS X](http://en.wikipedia.org/wiki/OS_X), [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [NetBSD](http://en.wikipedia.org/wiki/NetBSD), [Novell NetWare](http://en.wikipedia.org/wiki/Novell_NetWare), [OpenBSD](http://en.wikipedia.org/wiki/OpenBSD), [OpenSolaris](http://en.wikipedia.org/wiki/OpenSolaris), [OS/2](http://en.wikipedia.org/wiki/OS/2) Warp, [QNX](http://en.wikipedia.org/wiki/QNX), [Oracle Solaris](http://en.wikipedia.org/wiki/Solaris_%28operating_system%29), [Symbian](http://en.wikipedia.org/wiki/Symbian), [SunOS](http://en.wikipedia.org/wiki/SunOS), [SCO OpenServer](http://en.wikipedia.org/wiki/SCO_OpenServer), SCO [UnixWare](http://en.wikipedia.org/wiki/UnixWare), [Sanos](http://en.wikipedia.org/wiki/Sanos) and [Tru64](http://en.wikipedia.org/wiki/Tru64). A port of MySQL to [OpenVMS](http://en.wikipedia.org/wiki/OpenVMS) also exists.

MySQL is written in [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) and [C++](http://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [yacc](http://en.wikipedia.org/wiki/Yacc), but it uses a home-brewed [lexical analyzer](http://en.wikipedia.org/wiki/Lexical_analysis). Many [programming languages](http://en.wikipedia.org/wiki/Programming_language) with language-specific [APIs](http://en.wikipedia.org/wiki/Application_programming_interface) include [libraries](http://en.wikipedia.org/wiki/Library_%28computing%29) for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's [Visual Studio](http://en.wikipedia.org/wiki/Visual_Studio) (languages such as [C#](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29) and [VB](http://en.wikipedia.org/wiki/Visual_Basic) are most commonly used) and the JDBC driver for Java. In addition, an [ODBC](http://en.wikipedia.org/wiki/ODBC) interface called [MyODBC](http://en.wikipedia.org/wiki/MyODBC) allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as [ASP](http://en.wikipedia.org/wiki/Active_Server_Pages) or [ColdFusion](http://en.wikipedia.org/wiki/Adobe_ColdFusion). The [HTSQL](http://en.wikipedia.org/wiki/HTSQL) – [URL](http://en.wikipedia.org/wiki/Uniform_resource_locator)-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

## 12.4. FEATURES

MySQL is offered under two different editions: the open source MySQL Community Server and the proprietary [Enterprise Server](http://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

* A broad subset of [ANSI SQL 99](http://en.wikipedia.org/wiki/SQL:1999), as well as extensions
* Cross-platform support
* [Stored procedures](http://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](http://en.wikipedia.org/wiki/SQL/PSM)
* [Triggers](http://en.wikipedia.org/wiki/Database_trigger)
* [Cursors](http://en.wikipedia.org/wiki/Cursor_%28databases%29)
* Updatable [views](http://en.wikipedia.org/wiki/View_%28SQL%29)
* [Online DDL](http://en.wikipedia.org/wiki/Data_Definition_Language) when using the InnoDB Storage Engine.
* [Information schema](http://en.wikipedia.org/wiki/Information_schema)
* Performance Schema
* A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
* [X/Open XA](http://en.wikipedia.org/wiki/X/Open_XA) [distributed transaction processing](http://en.wikipedia.org/wiki/Distributed_transaction_processing) (DTP) support; [two phase commit](http://en.wikipedia.org/wiki/Two-phase-commit_protocol) as part of this, using the default [InnoDB](http://en.wikipedia.org/wiki/InnoDB) storage engine
* Transactions with [savepoints](http://en.wikipedia.org/wiki/Savepoint) when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
* [ACID](http://en.wikipedia.org/wiki/Atomicity,_consistency,_isolation,_durability) compliance when using InnoDB and NDB Cluster Storage Engines
* [SSL](http://en.wikipedia.org/wiki/Secure_Sockets_Layer) support
* Query [caching](http://en.wikipedia.org/wiki/Cache_%28computing%29)
* Sub-[SELECTs](http://en.wikipedia.org/wiki/Select_%28SQL%29) (i.e. nested SELECTs)
* Built-in [Replication](http://en.wikipedia.org/wiki/Database_replication) support (i.e. Master-Master Replication & Master-Slave Replication) with one master per slave, many slaves per master. [Multi-master replication](http://en.wikipedia.org/wiki/Multi-master_replication) is provided in [MySQL Cluster](http://en.wikipedia.org/wiki/MySQL_Cluster), and multi-master support can be added to unclustered configurations using Galera Cluster.
* Full-text [indexing](http://en.wikipedia.org/wiki/Index_%28database%29) and searching
* Embedded database library
* [Unicode](http://en.wikipedia.org/wiki/Unicode) support
* Partitioned tables with pruning of partitions in optimizer
* [Shared-nothing](http://en.wikipedia.org/wiki/Shared-nothing) clustering through [MySQL Cluster](http://en.wikipedia.org/wiki/MySQL_Cluster)
* Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.
* Native storage engines InnoDB, MyISAM, Merge, Memory (heap), [Federated](http://en.wikipedia.org/wiki/MySQL_Federated), Archive, [CSV](http://en.wikipedia.org/wiki/Comma-separated_values), Blackhole, NDB Cluster.
* Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

### 12.5. LIMITATIONS

Like other [SQL databases](http://en.wikipedia.org/wiki/SQL#Cross-vendor_portability), MySQL does not currently comply with the full SQL standard for some of the implemented functionality, including foreign key references when using some storage engines other than the default of InnoDB.

Up until MySQL 5.7, triggers are limited to one per action / timing, meaning that at most one trigger can be defined to be executed after an INSERT operation, and one before INSERT on the same table. No triggers can be defined on views.

MySQL, like most other [transactional](http://en.wikipedia.org/wiki/Database_transaction) relational databases, is strongly limited by hard disk performance. This is especially true in terms of write latency. Given the recent appearance of very affordable consumer grade SATA interface [solid-state drives](http://en.wikipedia.org/wiki/Solid-state_drive) that offer zero [mechanical latency](http://en.wikipedia.org/wiki/Latency_%28engineering%29#Mechanical_latency), a fivefold speedup over even an eight drive RAID array can be had for a smaller investment.

### 12.6. DEPLOYMENT

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the [package management system](http://en.wikipedia.org/wiki/Package_management_system) can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29)-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as [phpMyAdmin](http://en.wikipedia.org/wiki/PhpMyAdmin). In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is [replicated](http://en.wikipedia.org/wiki/Database_replication) to multiple slaves that handle all read operations. The master server synchronizes continually with its slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using [memcached](http://en.wikipedia.org/wiki/Memcached), or breaking down a database into smaller chunks called [shards](http://en.wikipedia.org/wiki/Shard_%28database_architecture%29) which can be spread across a number of distributed server clusters.

### 12.7. BACKUP

* [Filesystem snapshot](http://en.wikipedia.org/wiki/Filesystem_snapshot) or [volume manager snapshot](http://en.wikipedia.org/wiki/Volume_manager_snapshot) – backups are performed by using an external tool provided by the operating system (such as [LVM](http://en.wikipedia.org/wiki/Logical_Volume_Manager_%28Linux%29)) or storage device, with additional support from MySQL for ensuring consistency of such snapshots.
* *mysqldump* – a logical backup tool included with both community and enterprise editions of MySQL. Supports backup from all storage engines.
* *MySQL Enterprise Backup* – a hot backup utility included as part of the MySQL Enterprise subscription from Oracle. Offers native InnoDB hot backup, as well as backup for other storage engines.
* *XtraBackup* – open source MySQL hot backup software. Some notable features include hot, non-locking backups for InnoDB storage, incremental backups, streaming, parallel-compressed backups, throttling based on the number of I/O operations per second, etc.

### 12.8. HIGH AVAILABILITY

Ensuring [high availability](http://en.wikipedia.org/wiki/High_availability) requires a certain amount of redundancy in the system. For database systems, the redundancy traditionally takes the form of having a primary server acting as a master, and using [replication](http://en.wikipedia.org/wiki/Replication_%28computing%29) to keep secondaries available to take over in case the primary fails. This means that the "server" that the application connects to is in reality a collection of servers, not a single server. In a similar manner, if the application is using a [sharded](http://en.wikipedia.org/wiki/Shard_%28database_architecture%29) database, it is in reality working with a collection of servers, not a single server. In this case, a collection of servers is usually referred to as a *farm.*

One of the projects aiming to provide high availability for MySQL is *MySQL Fabric*, an integrated system for managing a collection of MySQL servers, and a [framework](http://en.wikipedia.org/wiki/Software_framework) on top of which high availability and database sharding is built. MySQL Fabric is open-source and is intended to be extensible, easy to use, and to support procedure execution even in the presence of failure, providing an execution model usually called *resilient execution.* MySQL client libraries are extended so they are hiding the complexities of handling [failover](http://en.wikipedia.org/wiki/Failover) in the event of a server failure, as well as correctly dispatching transactions to the shards. As of September 2013, there is currently support for Fabric-aware versions of Connector/[J](http://en.wikipedia.org/wiki/J_Sharp), Connector/[PHP](http://en.wikipedia.org/wiki/PHP), Connector/[Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29),

### 13. RELATED PROJECTS

* [Drizzle](http://en.wikipedia.org/wiki/Drizzle_%28database_server%29) – a fork targeted at the web-infrastructure and cloud computing markets. The developers of the product describe it as a "smaller, slimmer and (hopefully) faster version of MySQL". As a result, many common MySQL features will be stripped out, including stored procedures, query cache, prepared statements, views, and triggers. This is a partial rewrite of the server that does not maintain compatibility with MySQL. However, Drizzle made no new releases since May 2012.
* [MariaDB](http://en.wikipedia.org/wiki/MariaDB) – a community-developed fork of the MySQL database source code. The intent also being to maintain high fidelity with MySQL, ensuring a "drop-in" replacement capability with library binary equivalency and exact matching with MySQL APIs and commands. It includes the [XtraDB](http://en.wikipedia.org/wiki/XtraDB) storage engine as a replacement for InnoDB.
* [Percona Server](http://en.wikipedia.org/wiki/Percona_Server) – a fork of MySQL that includes the XtraDB storage engine. Its policy is to deviate as little as possible from MySQL and remain fully compatible, while providing new features, better performance, and additional instrumentation for analysis of performance and usage.
* [WebScaleSQL](http://en.wikipedia.org/wiki/WebScaleSQL) – A branch of MySQL 5.6 maintained by Facebook, LinkedIn, Google and Twitter.

## 14. HISTORY

MySQL was created by a Swedish company, MySQL AB, founded by [David Axmark](http://en.wikipedia.org/wiki/David_Axmark), [Allan Larsson](http://en.wikipedia.org/wiki/Allan_Larsson) and [Michael "Monty" Widenius](http://en.wikipedia.org/wiki/Michael_%28Monty%29_Widenius). The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from [mSQL](http://en.wikipedia.org/wiki/MSQL) based on the low-level language [ISAM](http://en.wikipedia.org/wiki/ISAM), which the creators considered too slow and inflexible. They created a new [SQL](http://en.wikipedia.org/wiki/Structured_Query_Language) interface, while keeping the same [API](http://en.wikipedia.org/wiki/Application_programming_interface) as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use MySQL instead of the (proprietarily licensed) mSQL antecedent.

### 15. MILESTONES

Notable milestones in MySQL development include:

* Original development of MySQL by [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius) and [David Axmark](http://en.wikipedia.org/wiki/David_Axmark) beginning in 1994
* First internal release on 23 May 1995
* Version 3.19: End of 1996, from www.tcx.se
* Version 3.20: January 1997
* Windows version was released on 8 January 1998 for Windows 95 and NT
* Version 3.21: production release 1998, from www.mysql.com
* Version 3.22: alpha, beta from 1998
* Version 3.23: beta from June 2000, production release 22 January 2001
* Version 4.0: beta from August 2002, production release March 2003 ([unions](http://en.wikipedia.org/wiki/Set_operations_%28SQL%29))
* Version 4.01: beta from August 2003, Jyoti adopts MySQL for database tracking
* Version 4.1: beta from June 2004, production release October 2004 ([R-trees](http://en.wikipedia.org/wiki/R-tree) and [B-trees](http://en.wikipedia.org/wiki/B-tree), subqueries, prepared statements)
* Version 5.0: beta from March 2005, production release October 2005 (cursors, stored procedures, triggers, views, [XA transactions](http://en.wikipedia.org/wiki/Database_transaction))

The developer of the Federated Storage Engine states that "The Federated Storage Engine is a [proof-of-concept](http://en.wikipedia.org/wiki/Proof_of_concept) storage engine", but the main distributions of MySQL version 5.0 included it and turned it on by default. Documentation of some of the short-comings appears in "MySQL Federated Tables: The Missing Manual". [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) acquired [MySQL AB](http://en.wikipedia.org/wiki/MySQL_AB) in 2008.

* Version 5.1: production release 27 November 2008 (event scheduler, [partitioning](http://en.wikipedia.org/wiki/Partition_%28database%29), plugin API, row-based replication, [server log](http://en.wikipedia.org/wiki/Server_log) tables)

Version 5.1 contained 20 known crashing and wrong result bugs in addition to the 35 present in version 5.0 *(almost all fixed as of release 5.1.51)*. MySQL 5.1 and 6.0-alpha showed poor performance when used for [data warehousing](http://en.wikipedia.org/wiki/Data_warehousing) – partly due to its inability to utilize multiple CPU cores for processing a single query.

* Oracle acquired Sun Microsystems on 27 January 2010.
* The day Oracle announced the purchase of Sun, Michael "Monty" Widenius forked MySQL, launching MariaDB, and took a swathe of MySQL developers with him.
* MySQL Server 5.5 was generally available (as of December 2010). Enhancements and features include:
  + The default storage engine is InnoDB, which supports transactions and referential integrity constraints.
  + Improved InnoDB I/O subsystem
  + Improved [SMP](http://en.wikipedia.org/wiki/Symmetric_multiprocessing) support
  + Semisynchronous replication.
  + SIGNAL and RESIGNAL statement in compliance with the SQL standard.
  + Support for supplementary Unicode character sets utf16, utf32, and utf8mb4.
  + New options for user-defined partitioning.
* MySQL Server 6.0.11-alpha was announced on 22 May 2009 as the last release of the 6.0 line. Future MySQL Server development uses a New Release Model. Features developed for 6.0 are being incorporated into future releases.
* MySQL 5.7 Development Milestone 3 was released December 2013.

**16. CODING**

<?php

include\_once("config.php");

include\_once("Userheader.php");

?>

<?php

error\_reporting(0);

$query2 = "select mobile from tab\_user where U\_name='".$\_SESSION['loggedIn']."'";

//echo $query2;

$result1 = mysql\_query($query2);

if(mysql\_num\_rows($result1))

{

$row1 = mysql\_fetch\_assoc($result1);

}

?>

<style type="text/css">

<!--

.ed{

border-style:solid;

border-width:thin;

border-color:#00CCFF;

padding:5px;

margin-bottom: 4px;

}

#button1{

text-align:center;

font-family:Arial, Helvetica, sans-serif;

border-style:solid;

border-width:thin;

border-color:#00CCFF;

padding:5px;

background-color:#00CCFF;

height: 34px;

}

#imagelist{

border: thin solid silver;

float:left;

padding:5px;

width:auto;

margin: 0 5px 0 0;

}

p{

margin:0;

padding:0;

text-align: center;

font-style: italic;

font-size: smaller;

text-indent: 0;

}

#caption{

margin-top: 5px;

}

img{

height: 225px;

}

-->

</style>

<form action="addexec.php" method="post" enctype="multipart/form-data" name="addroom">

<br>

<center>

<br>

<center><font color="red" size="14">Add Disaster Details</font></center>

<br>

<br>

User Id<br />

<input name="uname" type="text" class="ed" id="unameid" value="<?php echo $\_SESSION['login\_user']; ?>"/>

<br />

Disaster Heading<br />

<input name="DH" type="text" class="ed" id="DH" />

<br />

Description<br />

<input name="DES" type="text" class="ed" id="DES" />

<br />

Date<br />

<input name="DAT" type="text" class="ed" id="DAT" />

<br>

Select Image: <br />

<input type="file" name="image" class="ed"><br />

<input type="submit" name="Submit" value="Upload" id="button1" />

&nbsp;

<a href="Viewdisaster.php">My Update</a>

&nbsp;

<a href="Viewalldisaster.php">View All Disaster</a>

</center>

</form>

<br />

Photo Archieve

<br />

<?php

//include('config.php');

$result = mysql\_query("SELECT \* FROM adddisaster where Uid='".$\_SESSION['login\_user']."'");

while($row = mysql\_fetch\_array($result))

{

echo '<div id="imagelist">';

echo '<p><img src="'.$row['Ipath'].'"></p>';

echo '<p id="caption">'.$row['Des'].' </p>';

echo '</div>';

}

?>

<br>

<br>

<br>

<br>

<br>

<?php

include('Adminheader.php');

include('config.php');

error\_reporting(0);

if(isset($\_POST['submit']))

{

if($\_POST['area']=="" || $\_POST['city']==""|| $\_POST['emer']==""|| $\_POST['help']=="")

{

echo '<script> alert("PLEASE FILL THE DETAILS");</script>';

}

else

{

$query = "INSERT INTO `addemergency` (`id`,`Area`, `City`, `Emergency`,`Help`) VALUES ('null','".$\_POST['area']."','".$\_POST['city']."','".$\_POST['emer']."','".$\_POST['help']."')";

//echo $query;exit;

if(mysql\_query($query)){

echo 'ADD SUCCESSFULLY';

}

else{

echo 'NOT REGISTERED';

}

header("location:AddEmergency.php");

exit;

}

}

?>

<style>

h1 {

color: blue;

font-family: verdana;

font-size: 300%;

}

p {

color: gray;

font-family: Georgia, serif;

font-size: 140%;

font-weight: bold;

}

</style>

<style>

input[type=button], input[type=submit], input[type=reset] {

background-color: #FFC912;

border: none;

color: white;

padding: 15px 42px;

text-decoration: none;

margin: 4px 2px;

cursor: pointer;

text-align: center;

display: inline-block;

border-radius: 16px;

}

</style>

<style>

.textbox {

border: 1px solid #848484;

-webkit-border-radius: 30px;

-moz-border-radius: 30px;

border-radius: 30px;

outline:0;

height:25px;

width: 275px;

padding-left:10px;

padding-right:10px;

}

</style>

<div id="container">

<div background="">

<br>

<div id="error\_msg"></div>

<form action="" method="post" >

<center>

<table border="0" cellpadding="5" cellspacing="0" width="350" Height="120" align="center" class="details\_single">

<tr>

<td colspan="2"><h1><b>Add Emergency</b></h1> </td>

</tr>

</table>

<table border="0" cellpadding="5" cellspacing="0" width="310" Height="100" align="center" class="details\_single">

<tr>

<td ><p>Area</p> </td>

<td ><input type="text" name="area" id="area" class="textbox"/>

</td>

</tr>

<tr>

<td ><p>City </p></td>

<td ><input type="text" name="city" id="city" class="textbox"/>

</td>

</tr>

<tr>

<td ><p>Emergency </p></td>

<td ><input type="text" name="emer" id="emer" class="textbox"/>

</td>

</tr>

<tr>

<td ><p>Help </p></td>

<td ><input type="text" name="help" id="help" class="textbox"/>

</td>

</tr>

<tr>

<td colspan="2" align="center">

<input type="submit" name="submit" value="Add" />

</td>

</tr>

</table>

<h1>Emergency Details</h1>

<table border="2" cellspacing="6" class="displaycontent" width="1200" height="120" style="border:10px solid #800000;" align="center">

<tr>

<th bgcolor=Black><font color=white size=2>id</font></th>

<th bgcolor=Black><font color=white size=2>Area</font></th>

<th bgcolor=Black><font color=white size=2>City</font></th>

<th bgcolor=Black><font color=white size=2>Emergency</font></th>

<th bgcolor=Black><font color=white size=2>Help</font></th>

<th bgcolor=Black><font color=white size=2>Delete</font></td>

</tr>

<?php

$query = "select \* from addemergency";

$result = mysql\_query($query) or die(mysql\_error());

while($row = mysql\_fetch\_assoc($result))

{

?>

<tr>

<td bgcolor=white><font color=#000000 size=2><?php echo $row['id']; ?></font></td>

<td bgcolor=white><font color=#000000 size=2><?php echo $row['Area']; ?></font></td>

<td bgcolor=white><font color=#000000 size=2><?php echo $row['City']; ?></font></td>

<td bgcolor=white><font color=#000000 size=2><?php echo $row['Emergency']; ?></font></td>

<td bgcolor=white><font color=#000000 size=2><?php echo $row['Help']; ?></font></td>

<td bgcolor=white><font color=#000000 size=2><a href="?delete=<?php echo $row['id'];?>">Delete</a></font></td>

</tr>

<?php } ?>

</table>

</center>

</form>

</div>

<br>

<br>

<?php

if(isset($\_GET['delete']))

{

$query = "delete from addemergency where id='".$\_GET['delete']."'";

mysql\_query($query);

echo '<script>alert("Deleted");</script>';

header("location:AddEmergency.php");

exit;

}

?>