MySQL Essentials - Part 1

Part 1: What’s MySQL?

Part 2: Building, Installing & Configuring MySQL

Part 3: Application / Architecture Considerations

Part 4: Developing MySQL Applications
What’s MySQL?
Overview of MySQL

• 12 million product installations

• 65,000 downloads each day

• Part of the rapidly growing open source LAMP stack

• MySQL GPL & Commercial Editions Available
LAMP

- Operating System
- Application Server
- Database
- Scripting
MySQL is Everywhere

Multiple Platforms

Multiple Languages

C
C++
C#
MySQL: #3 Most Deployed Database

63% Are Deploying MySQL or Are Planning To Deploy
Oracle’s Investment in MySQL

- **Make MySQL a Better MySQL**
  - #1 Open Source Database for Web Applications

- **MySQL Focus Areas**
  - Web, Embedded & Telecom
  - LAMP
  - Windows

- **Develop, Promote and Support MySQL**
  - Improve engineering, consulting and support
  - Leverage 24x7, World-Class Oracle Support

- **MySQL Community Edition**
  - Source and binary releases
  - GPL license
MySQL Customers

MySQL is Powering the Web
Web: facebook

Application
Facebook is a social networking site that connects people with friends and others who work, study and live around them.

Key Business Benefit
MySQL has enabled facebook to grow to 500 million users.

Why MySQL?
“We are one of the largest MySQL web sites in production. MySQL has been a revolution for young entrepreneurs.”

Owen Van Natta
Chief Operating Officer
Facebook
Web: eBay

**Application**
Real-time personalization platform to display advertising to more relevant buyers and conduct more effective merchandizing.

**Key Business Benefits**
Highly scalable and cost-effective system that handles all of eBay’s personalization and session data needs. Manages 4 billion requests per day of 50/50 read/write operations.

**Why MySQL Enterprise Edition?**
- Cost-effective
- Performance: 13,000 TPS on Sun Fire x4100
- Scalability: Designed for 10x future growth
- Monitoring: MySQL Enterprise Monitor

*Chris Kasten, Kernel Framework Group, eBay*
OEM/ISV: Adobe

Application
Adobe embeds MySQL into several Adobe Creative Suite 3 components, including Adobe Acrobat CS3, Adobe Bridge CS3 and Adobe Version Cue CS3.

Key Business Benefit
MySQL allows Adobe to implement the workgroup productivity features in Adobe Creative Suite CS3 so that our users can be more productive.

Why MySQL?
“We chose MySQL for its reliability, ease of use, low administration and high performance.”

Mike Wallen
Version Cue Product Manager
Telco: Alcatel-Lucent

Application
Alcatel-Lucent’s next generation subscriber database applications

Key Business Benefit
The subscriber database at the heart of the application needed to provide more flexibility and to deliver higher performance, scalability, and reliability at a lower cost.

Why MySQL? Performance & Lower Cost
“MySQL Cluster won the performance tests hands down, and met our needs perfectly.”

Alain Chastagner,
Alcatel-Lucent
Before We Get Started: Basics
Hardware: The Perfect MySQL Server

- The more cores the better (especially for 5.5 and later)
- x86_64 - 64 bit for more memory is important
  - Data/Memory ratio 1/3 to 1/10 is good rule of thumb
  - The more the better
- Linux or Solaris best, Windows and Unix also fine.
- RAID 10 for most, RAID 5 OK if very read intensive
- Hardware RAID battery backed up cache critical!
  - More disks are always better!
    - 4+ recommended, 8-16 can increase IO performance if needed
- At least 2 x NICs for redundancy
- Slaves should be as powerful as the Master
- Oracle Sun X4170 for example
MySQL Essentials - Part 2

Part 1: Why MySQL?

Part 2: Building, Installing & Configuring MySQL

Part 3: Application / Architecture Considerations

Part 4: Developing MySQL Applications
Software: Where to Download?

- **Oracle E-Delivery**
  - MySQL Commercial Binaries
  - Commercial-licensed add-ons
    - MySQL Enterprise Monitor
    - MySQL Enterprise Backup
- **My Oracle Support**
  - Latest patch levels
- **Dev.mysql.com**
  - GPL Binaries and Source Code
- **Labs.mysql.com**
  - Experimental code
  - E.g. InnoDB + Memcached
MySQL Windows Installer
MySQL Server Support for Windows

- All current Windows versions…
  - Windows XP
  - Windows Vista
  - Windows 7
  - Windows Server 2003
  - Windows Server 2008

- Minimal additional requirements

- 32 and 64 bit x86 architectures supported
Installation Packages

- **Complete (MSI Installer or unzip)**
- **Essentials**
  - *Excludes Instance Manager*
  - *Documentation*
  - *Developer Components*
Begin Installation

- Run `mysql-5.x.xx-winxx.msi`

- **Select type of install...**
  - Developers select **Complete**
  - DBAs select **Typical**
Complete Installation

- Several **Next** clicks
- Then **Finish**
- Leave **Configure** checked
To re-launch configuration wizard:
Start->MySQL->MySQL Server 5.5-> MySQL Server Instance Config Wizard
Selecting Configuration Options

- **Instance**
  - a running MySQL Server process
  - or a MySQL Windows Service

- **Instance features and properties are determined by**
  - Parameters
  - Startup options
  - Active (enabled) components

- **Configuration file is** *my.ini*

- **Located in the MySQL installation directory**
Selecting Configuration Options (cont.)

- Select appropriate server type
- For a trial use **Developer Machine**
Selecting Configuration Options (cont.)

- MySQL supports many database **storage engines**
- Perhaps a new concept for some DBAs
- Can all run at the same time on the same MySQL Instance
- **InnoDB** (transactional) is the one to care about most
- Select **Multifunction**
Selecting Configuration Options (cont.)

- Select location for InnoDB files

✅ Performance Tips

- High performance storage?
  - Specify drive letter
- Select a disk that has:
  - Space available
  - Where OS is not installed
Selecting Configuration Options (cont.)

- **Set expected connections**

- **Default Port** – 3306
- **Use Strict Mode**

Strict mode controls how MySQL handles invalid or missing input values.
Selecting Configuration Options (cont.)

MySQL Server Instance Configuration Wizard
Configure the MySQL Server 5.1 server instance.

Please set the Windows options.

- **Install As Windows Service**
  This is the recommended way to run the MySQL server on Windows.
  Service Name: MySQL

- **Include Bin Directory in Windows PATH**
  Check this option to include the directory containing the server/client executables in the Windows PATH variable so they can be called from the command line.

MySQL Server Instance Configuration Wizard
Configure the MySQL Server 5.1 server instance.

Please select the default character set.

- **Standard Character Set**
  Makes Latin1 the default charset. This character set is suited for English and other West European languages.

- **Best Support For Multilingualism**
  Makes UTF8 the default character set. This is the recommended character set for storing text in many different languages.

- **Manual Selected Default Character Set / Collation**
  Please specify the character set to use.
  Character Set: latin1

< Back | Next > | Cancel

ORACLE
Selecting Configuration Options (cont.)

- **Password for MySQL “root” account.**
  - Equiv to “sa” in MSSQL
  - NOT an OS account
  - Don’t forget

- **Enable access from remote if you plan on non-local access.**

![MySQL Server Instance Configuration Wizard](image)

- Please set the security options.
  - **Modify Security Settings**
    - New root password:
    - Confirm:
    - Enter the root password.
    - Retype the password.
    - Enable root access from remote machines

- **Create An Anonymous Account**
  - This option will create an anonymous account on this server. Please note that this can lead to an insecure system.
Selecting Configuration Options (cont.)

- **From the command line**
  
  ```
  >"C:\Program Files\MySQL\MySQL Server 5.5\bin\mysqld"
  --defaults-file="C:\Program Files\MySQL\MySQL Server 5.5\my.ini"
  ```

- **Or installed and Started as a service**

- **Hint:** if you are having any issues append
  
  `--console`
Building from Source
Why/When to Build from Source

• Binary packages available for all supported platforms
• Install and start coding in SQL, Java, C, C++, C#, PHP, Perl, Python, VB…
• Build-from-source appropriate for:
  • Deploying on unsupported platforms
  • Compiling with more aggressive optimization options, or more extensive debugging options
  • Developing add-ons such as storage engines
  • Packaging MySQL as part of custom hardware or software solutions
  • Developing performance patches
Post Installation Tasks & Security
Post Installation Tasks To Secure Installation

- Delete the test database
- Secure the root account
- Ensure root can’t login from an anonymous host
- Delete anonymous accounts
- MSI installer or mysql_secure_installation script (in scripts directory) can be used to automate some of these steps
Delete the test Database

```sql
mysql> use mysql;
Database changed
mysql> show databases;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| mysql                    |
| test                     |
+--------------------------+
3 rows in set (0.00 sec)
mysql> drop database test;
Query OK, 0 rows affected (0.00 sec)
mysql> show databases;
+--------------------------+
| Database                 |
+--------------------------+
| information_schema       |
| mysql                    |
+--------------------------+
2 rows in set (0.00 sec)
```
One (or two) accounts are created named root
Super user accounts that can do anything
The initial root account passwords are blank
One root account for connecting from local host
Other allows connections from any host
Rename the root account with a different (more difficult name) to prevent brute-force attacks
Secure root Account

```
C:\Documents and Settings\jguerrero>mysql -uroot
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 5.1.43-community MySQL Community Server <GPL>

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mysql;
Database changed
mysql> SELECT user, host, password FROM user;
+----------+-----------+-----------+
| user     | host      | password  |
+----------+-----------+-----------+
| root     | localhost | password  |
| root     | 127.0.0.1 | localhost |
+----------+-----------+-----------+
3 rows in set (0.00 sec)

mysql> SET PASSWORD FOR root@localhost=PASSWORD('secure_password');
Query OK, 0 rows affected (0.03 sec)

mysql> SELECT user, host, password FROM user;
+----------+-----------+-----------+
| user     | host      | password  |
| root     | localhost | *F31445443BB93ED07F5FAB7744A3FCE47021238F |
| root     | 127.0.0.1 | localhost |
+----------+-----------+-----------+
3 rows in set (0.00 sec)
```
Change the Name of root User

C:\Documents and Settings\jguerrero>mysql
Welcome to the MySQL monitor. Commands end with ; or \\.
Your MySQL connection id is 25
Server version: 5.1.41-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> USE mysql;
Database changed
mysql> UPDATE user SET user='superadmin' WHERE user='root';
Query OK, 2 rows affected (0.02 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.03 sec)
Anonymous Accounts

- Two anonymous user accounts are created by default
  - Empty user names and no passwords
  - One anonymous account for connections from the local host
  - Prior to 5.1.16 had global privileges, just like the root accounts
  - Other is for connections from any host
  - Has all privileges for the test database or other databases with names that start with test
- Delete these accounts!
Delete Anonymous Accounts

```sql
mysql> DELETE FROM user WHERE user = '';
Query OK, 1 row affected (<0.01 sec)

mysql> SELECT user, host, password FROM user;
+----------------+-------+---------------------+
| user | host | password             |
+----------------+-------+---------------------+
| root | localhost | *F31445443BB93ED07F5FAB7744A3FCE47021238F |
| root | 127.0.0.1 | *F31445443BB93ED07F5FAB7744A3FCE47021238F |
+----------------+-------+---------------------+
2 rows in set (<0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (<0.00 sec)
```
MySQL Essentials - Part 3

Part 1: Why MySQL?

Part 2: Building, Installing & Configuring MySQL

Part 3: Application / Architecture Considerations

Part 4: Developing MySQL Applications
InnoDB as default storage engine

- ACID Transactions, FKs, Crash Recovery
Storage Engines

- MySQL cliches are all about MyISAM:
  - Fast for simple operations, but not scalable due to table locks
  - Vulnerable to crashes; repair tables on restart
  - No transactions; can’t roll back after mistakes or errors

- InnoDB is the most Oracle-like part of MySQL:
  - Concurrent readers and writers; high scalability
  - Transactional - commit changes, roll back mistakes, queries see snapshot data; a.k.a. ACID model
  - Fast and reliable crash recovery

- InnoDB = default storage engine in 5.5 and up
- Still on 5.1? Install InnoDB Plugin
Storage Engines (cont.)

• What does it mean for InnoDB to be default?
  • Define a primary key for all tables, based on most important queries.
  • Secondary indexes also important to avoid full-table scans.
  • Turn auto-commit off.
  • Set innodb_buffer_pool_size as high as practical.
  • Investigate file-per-table mode, Barracuda file format, compression.
  • Lots more documentation in 5.5: SE chapter, Optimization, Glossary.

• MEMORY storage engine:
  • InnoDB with big buffer pool can give similar benefits.
  • For queries with >, <, BETWEEN operators, explore USING BTREE.
MySQL Replication
Read Scalability

- Used by leading web properties for scale-out
- Reads are directed to slaves, writes to master
- Delivers higher performance & scale with efficient resource utilization
# Medium: Web Reference Architecture

<table>
<thead>
<tr>
<th></th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
<th>Extra Large</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Queries/Second</strong></td>
<td>&lt;500</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td><strong>Transactions/Second</strong></td>
<td>&lt;100</td>
<td>&lt;1,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td><strong>Concurrent Read Users</strong></td>
<td>&lt;100</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td><strong>Concurrent Write Users</strong></td>
<td>&lt;10</td>
<td>&lt;100</td>
<td>1,000+</td>
<td>2,500+</td>
</tr>
</tbody>
</table>

## Database Size

<table>
<thead>
<tr>
<th></th>
<th>Sessions</th>
<th>eCommerce</th>
<th>Analytics</th>
<th>Content Management</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sessions</strong></td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>20+ GB</td>
<td>40+ GB</td>
</tr>
<tr>
<td><strong>eCommerce</strong></td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>20+ GB</td>
<td>40+ GB</td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>1+ TB</td>
<td>2+ TB</td>
</tr>
<tr>
<td><strong>Content Management</strong></td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>1+ TB</td>
<td>2+ TB</td>
</tr>
</tbody>
</table>
Medium: Web Reference Architecture

Session Management

Memcache / Application Servers

Heartbeat Mechanism

MySQL Enterprise Monitor

MySQL Enterprise Backup

eCommerce

MySQL Master

Analytics

MySQL Master

Content Management

Memcache / Application Servers

MySQL Master

Slave 1  Slave 2  Slave 3  Slave N

SAN

XOR
Best Practices (1)
Medium Web Reference Architecture

- Each component deployed onto dedicated server & storage infrastructure
  - Deployed, managed and scaled independently
- Server ratio: 8 application servers to each MySQL Server
  - More for PHP applications, less for Java
- Memcached deployed in session & content management components
  - Distributed memory caching layer
  - Reads fulfilled from cache, relieving load on the source database servers
Best Practices (2)
Medium Web Reference Architecture

• Content Management
  • Each slave can handle around 3,000 concurrent users
  • Each master can handle up to 30 slaves
  • MySQL Replication for high availability
    • Can include Heartbeat, depending on application failover requirements
  • Meta data of content assets managed by MySQL
  • Distributed File System (i.e. MogileFS for indexing content assets)
• Physical storage
  • High quality SAN (redundancy for HA)
  • Distributed across local storage with DRBD for HA of indexing and meta data
**Best Practices (3)**

Medium Web Reference Architecture

- **Session Management & eCommerce**
  - Deployed onto InnoDB storage engine
  - Session data maintained for up to 1 hour in a dedicated partition, rolling partitions used to delete aged data
  - Data is captured in Analytics Datababase
  - MySQL Replication with Heartbeat for HA
    - Optionally add DRBD for eCommerce
  - If web traffic grows, move Session Management to MySQL Cluster
    - Persist session data for real-time personalization of user experience
    - 99.999% availability and in-memory data management can reduce need for DRBD & memcached
MySQL Essentials - Part 4

Part 1: Why MySQL?

Part 2: Building, Installing & Configuring MySQL

Part 3: Application / Architecture Considerations

Part 4: Developing MySQL Applications
Developing MySQL Applications
MySQL User Survey: Top Languages

Which languages do you use to develop your MySQL Applications?

- Java
- PHP
- JavaScript/AJAX
- Perl
- C++
- C
- C#
- Other
- Python
- Visual Basic
- Ruby
- Delphi

Legend:
- Enterprise
- Community
Developing Java Applications with MySQL
Connecting Java to MySQL

• *The JDBC driver is called MySQL Connector/J*
• *Type IV (all-java)*
• *Available from the following sources*
  • **Software**
    • [http://dev.mysql.com/downloads/connector/j/5.5.html](http://dev.mysql.com/downloads/connector/j/5.5.html)
  • **Maven, Ivy**
  • **Many Linux and BSD distributions**
  • **Documentation**
Connector/J Wisdom

• rewriteBatchedStatements=true
• useServerPrepStmts=true
• cachePrepStmts=true; also set prepStmtCacheSize and prepStmtCacheSqlLimit
• maintainTimeStats=false
• useUnbufferedIO=false, useReadAheadInput=false
  • Always benchmark I/O settings like these
• useConfigs=... for bundled sets of config options:
  • maxPerformance, solarisMaxPerformance, fullDebug, 3-0-Compat, 5-0-Compat
Understanding the Classpath

• The classpath is where Java looks for compiled code
• There are some standard classpath locations:
  • WEB-INF/lib, WEB-INF/classes in webapps
  • $JRE_HOME/lib/ext
• When developing or deploying avoid these:
  • $JRE_HOME/lib/ext
  • Application-server-wide
    • e.g. $CATALINA_HOME/lib
• If possible let your IDE or tools manage it
Architecting for Success - Start Clean

Transactions

Presentation

Business Logic

Persistence

Caching

} Test this

} Test this

} Test this
Points to take away

• Start with clean architecture
  • Easier to start with than refactor to
  • Enable reusability
  • Enable quality through testability
  • Easier to add caching, partitioning later

• Consider frameworks instead of “raw” JDBC:
  • Avoid repetitive boilerplate code
  • Avoid resource-wasting mistakes
  • Enable re-use of persistence components
  • Business logic-level transactions become simpler
Developing PHP Applications with MySQL
Introduction to PHP

- PHP Hypertext Preprocessor
- Web-Centric Scripting Language
  - Processed by a Web-Server module
  - Can be embedded in HTML
  - Built-in functionality for dealing with Web-Things
- PHP consists out of a relatively small core and a large collection of function libraries (“extensions”)
- http://php.net
PHP Installation

• You need:
  – A Web Server (Apache HTTPd, Oracle Web Server, Microsoft IIS, nginx, …)
  – MySQL Server
  – PHP runtime

• For getting started there are bundles installing all of them
  – XAMPP

• It is also useful to install an IDE like NetBeans for editing PHP files
XAMPP

- Press Install button to start extraction.
- Use Browse button to select the destination folder from the folders tree. It can be also entered manually.
- If the destination folder does not exist, it will be created automatically before extraction.
- After extraction, the setup script will be started.
- To install services or start/stop the servers, please use the XAMPP Control Panel.

www.apachefriends.org/xampp
Verifying the PHP Installation

c:\xampp\htdocs\test.php:

```php
<?php
phpinfo();
?>
```

http://localhost/test.php

---

**MySQL Support**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>MySQL Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client API library version</td>
<td>enabled</td>
</tr>
<tr>
<td>Active Persistent Links</td>
<td>0</td>
</tr>
<tr>
<td>Inactive Persistent Links</td>
<td>0</td>
</tr>
<tr>
<td>Active Links</td>
<td>9</td>
</tr>
</tbody>
</table>
A First PHP Example

```php
<?php
$connection = mysqli_connect('localhost', 'root', '', 'test');
if (!$connection) {
    die('Error: ' . mysqli_connect_error());
}
$result = mysqli_query($connection,
    'SELECT first_name, last_name FROM employees LIMIT 5');
if (!$result) {
    die('Error: ' . mysqli_error());
}

echo "<table>
";
while ($row = mysqli_fetch_assoc($result)) {
    printf("<tr><td>%s</td><td>%s</td></tr>
",
        htmlentities($row['first_name']),
        htmlentities($row['last_name'])),
}

echo "</table>
";
mysqli_free_result($result);
mysqli_close($connection);
?>
```
ext/mysql

- One of the first PHP extensions
- Actively maintained with PHP 4
  - No new features in PHP 5
    - Exception: Added mysqli support with PHP 5.3
  - Bug fixing only
- Best documented database extension
  - Tons of books, tutorials, ...
- Missing support for many MySQL features
  - Prepared statements, Queries with multiple result sets (stored procedures), compression, encryption, full charset support, ...
mysqli - The Improved MySQL Extension

- Full support for all MySQL features
  - Stored Procedures
  - Prepared Statements
  - Encryption (SSL)
  - Compression
  - Charsets
  - ...

- Actively developed, maintained and supported by Oracle
PDO_mysql

- “The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP.” [http://php.net/intro.pdo](http://php.net/intro.pdo)
- Lowest common denominator
- PHPish API
- PDO emulates prepared statements by default
  - *We recommend turning on “real” prepared statements:*
    
    ```php
    $pdo->setOption(PDO::MYSQL_ATTR_DIRECT_QUERY, true);
    ```
PDO Example

• <?php
  $pdo = new PDO("mysql:host=localhost;dbname=test", "user", "password");

  $query = $pdo->prepare("SELECT id FROM table LIMIT ?, ?");
  $query->bindValue(1, $_GET["offset"], PDO::PARAM_INT);
  $query->bindValue(2, (int)$_GET["limit"], PDO::PARAM_INT);

  $query->execute();
Reasons for using different APIs

- **mysqli**
  - Support for all MySQL features
  - Best support / stability
  - Integration with existing applications / environments

- **PDO**
  - Simple applications supporting multiple databases (for instance Oracle DB and MySQL)
  - Integration with existing applications / environments
Validation

- Validate all input values.
- Prevent mistakes by users
  - The sooner a wrong input is detected the better it can be handled
- Might prevent some attacks
  - No full security
Validation Examples

- *In some countries last names contain spaces*
  - García Gonzalez
- *Some countries have their own letters*
  - Schlüter
- *Some countries use characters with special meanings to databases*
  - O'Harra
- *Non-Latin alphabets present other challenges, e.g. multi-byte characters*

> Validation can't do everything we need!
  - A validation might check the min. and max. length at least
  - Validate your input, escape your output. Always.
Validation can be relatively easy ...
Escaping for mysqli

- mysqli_real_escape_string()
  - Escapes special characters for usage in SQL statements
  - Takes current encoding into account

```
$sql = sprintf("INSERT INTO employees
  (birth_date, first_name, last_name, gender)
VALUES ('%s', '%s', '%s', '%s')",
  mysqli_real_escape_string($conn, $_POST['birth_date']),
  mysqli_real_escape_string($conn, $_POST['first_name']),
  mysqli_real_escape_string($conn, $_POST['last_name']),
  mysqli_real_escape_string($conn, $_POST['gender'])
);
if ( ! mysqli_query($conn, $sql) ) {
  // ERROR
}
```
Prepared Statements

Client

SELECT foo FROM bar
WHERE id = ?

Server

prepare()

• Create Execution plan

Handle

Handle

execute()

Param 1: 42

• Query database

Resultset(s)
$query = "INSERT INTO employees (first_name, last_name, gender) VALUES (? , ? , ?)";
$stmt = mysqli_prepare($conn, $query);

mysqli_stmt_bind_param($stmt, "sss", $val1, $val2, $val3,$val4);

$val1 = 'Johannes';
$val2 = 'Schlüter';
$val3 = 'M';
mysqli_stmt_execute($stmt);

$val1 = 'Andrey';
$val2 = 'Hristov';
$val3 = 'M';
mysqli_stmt_execute($stmt);
mysqli_stmt_close($stmt);
$connectionParams = array(
    'dbname' => 'mydb',
    'user' => 'user',
    'password' => 'secret',
    'host' => 'localhost',
    'driver' => 'pdo_mysql',
);
$conn = DriverManager::getConnection($connectionParams);

$conn->insert('user', array('username' => 'johannes'));
// INSERT INTO user (username) VALUES (?) (johannes)

$conn->update('user', array('username' => 'johannes'), array('id' => 1));
// UPDATE user (username) VALUES (?) WHERE id = ? (johannes, 1)
PHP Frameworks

- PHP applications often have to do the same things over and over again
  - Handling navigation
  - Handling form data
- There are proven concepts for application architectures
  - Model-View-Controller

- Frameworks usually save time and enforce clean structures
- Also make choice of PHP extension less of a factor
Examples of Frameworks

- There are two major general purpose frameworks
  - Zend Framework
    - Zend Technologies, Ltd.
    - [http://framework.zend.com](http://framework.zend.com)
  - Symfony
    - Sensio Labs
    - [http://www.symfony-project.org](http://www.symfony-project.org)
  - Others include: CakePHP, Agavi, Zeta Components
- Many applications provide their own framework
  - Typo3, Drupal, Joomla
Learn More: PHP Resources

- MySQL DevZone (technical articles, developer interview …)
  http://dev.mysql.com/

- MySQL Documentation
  http://dev.mysql.com/doc/

- Download Free MySQL White Papers
  http://dev.mysql.com/why-mysql/white-papers/

- View MySQL Training Courses
Developing .NET Applications with MySQL
What You Need for Connector/NET Development

• *MySQL Database Installed and Running*

• *Connector/NET Driver (v6.3.2 or later for VS 2010)*


• *MySQL Workbench 5.2 (optional)*
Connectors & Visual Studio

- ODBC, JDBC, C++, C....
- Connector/.NET
  - C#
  - ASP.NET
  - VB.net
- Connector/Net
  - Implements the ADO.NET interfaces
  - Integrates into ADO.NET aware tools
  - Fully managed ADO.NET driver
  - 100% pure C#
- Integrates with Visual Studio

http://www.mysql.com/products/connector
Classes in Connector/.NET connect to the database, execute queries and statements, and manage query results:

- **MySqlCommand**: Represents an SQL statement to execute against a MySQL database.
- **MySqlCommandBuilder**: Automatically generates single-table commands used to reconcile changes made to a DataSet with the associated MySQL database.
- **MySqlConnection**: Represents an open connection to a MySQL Server database.
- **MySqlDataAdapter**: Represents a set of data commands and a database connection that are used to fill a data set and update a MySQL database.
- **MySqlDataReader**: Provides a means of reading a forward-only stream of rows from a MySQL database.
- **MySqlException**: The exception that is thrown when MySQL returns an error.
- **MySqlHelper**: Helper class that makes it easier to work with the provider.
- **MySqlTransaction**: Represents an SQL transaction to be made in a MySQL database.
Creating a Connection String

- The `MySqlConnection` object is configured using a connection string. A connection string contains several key/value pairs, separated by semicolons:

  ```
  Server=127.0.0.1;Uid=root;Pwd=12345;Database=test;
  ```

- In this example, the `MySqlConnection` object is configured to connect to a MySQL server at 127.0.0.1, with a user name of root and a password of 12345, and a default database test for all statements.
Connection Example

The following **Visual Basic** code creates a **MySqlConnection** object, assigns the connection string, and opens the connection.

```vbnet
Dim conn As New MySql.Data.MySqlClient.MySqlConnection
Dim myConnectionString as String
myConnectionString = "server=127.0.0.1;" & "& "uid=root;" & "& "pwd=12345;" & "& "database=test;"
Try
    conn.ConnectionString = myConnectionString
    conn.Open()
Catch ex As MySql.Data.MySqlClient.MySqlException
    MessageBox.Show(ex.Message)
End Try
```

**Equivalent C# example:**

```csharp
using MySql.Data.MySqlClient;

string myConnectionString = "server=127.0.0.1;uid=root;pwd=12345;database=test;"
try
{
    MySqlConnection conn = new MySqlConnection();
    conn.ConnectionString = myConnectionString;
    conn.Open();
}
catch (MySql.Data.MySqlClient.MySqlException ex)
{
    MessageBox.Show(ex.Message);
}
```
Connection Errors

Because connecting to an external server is unpredictable, it is important to add error handling to your .NET application. When there is an error connecting, the MySqlConnection class returns a MySqlException object.

This object has two properties for handling errors:

- **Message**: A message that describes the current exception.
- **Number**: The MySQL error number.

When handling errors, your application responds based on the error number. The two most common error numbers when connecting are:

- **0**: Cannot connect to server.
- **1045**: Invalid user name and/or password.
Using MySqlCommand

- A MySqlCommand has the CommandText and CommandType properties associated with it. The CommandText is handled differently depending on the setting of CommandType.

CommandType can be one of:

1. Text - A SQL text command (default)
2. StoredProcedure - The name of a Stored Procedure
3. TableDirect - The name of a table (new in Connector.NET 6.2)

The default CommandType, Text, is used for executing queries and other SQL commands.

If CommandType is set to StoredProcedure, CommandText should be set to the name of the Stored Procedure to access.
Using MySqlCommand (cont.)

• If `CommandType` is set to `TableDirect`, all rows and columns of the named table will be returned when you call one of the Execute methods. In effect, this command performs a `SELECT *` on the table specified.

• The `CommandText` property is set to the name of the table you wish to query.

• The following illustrates this:

```csharp
MySQLCommand cmd = new MySqlCommand();
cmd.CommandText = "mytable";
cmd.Connection = someConnection;
cmd.CommandType = CommandType.TableDirect;
SqlDataReader reader = cmd.ExecuteReader();
While (reader.Read())
{
    Console.WriteLine(reader[0], reader[1]...);
}
```
Integrating Connector/NET with Visual Studio


- Visual Studio 2010 support was introduced with MySQL Connector/NET 6.3.2. From version 6.3.2 the connector ships with both NET 2.x and .NET 4.x versions of the Entity Framework support files, `mysql.data.ef.dll` and `mysql.visualstudio.dll`.

- When MySQL Connector/NET is installed on Microsoft Windows, Visual Studio integration components are also installed and initialized. This enables the developer to work seamlessly with MySQL Connector/NET in the familiar Visual Studio environment.
Integrating Connector/.NET with Visual Studio (cont.)

The .NET 4.x versions need to be shipped to enable new integration features supported in Visual Studio 2010, including:

- New DDL T4 template for the Entity Framework (EF)

- Enables developers to design an EF model from scratch and use the native Visual Studio 2010 facility to generate MySQL DDL from that model. This is done by creating the model and choosing the SSDLToMySQL template in the properties window.

- The correct DDL is then generated and the developer can then save this code as a .mysql file in their project and execute it against the MySQL server.

- New SQL Editor - A new SQL editor has been included that enables connections to servers to execute SQL. This is activated by creating a new file with a .mysql extension. A new template is also included to allow creation of this file type using the Visual Studio 2010 main menu item FILE, NEW.

- Note: the MySQL SQL Editor is also available in 2005 and 2008.
## Visual Studio Integration: Editing Database Objects

<table>
<thead>
<tr>
<th>Tables</th>
<th>Views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>Stored Procs &amp; Functions</td>
</tr>
<tr>
<td>Indexes</td>
<td>Triggers</td>
</tr>
<tr>
<td>Foreign Keys</td>
<td>User-Defined Functions (UDF)</td>
</tr>
<tr>
<td>Column &amp; Table Properties</td>
<td>Cloning Database Objects</td>
</tr>
<tr>
<td>Using ADO.NET Entity Framework</td>
<td>MySQL SQL Editor</td>
</tr>
<tr>
<td>DDL T4 Template Macro</td>
<td>MySQL Website Configuration Tool</td>
</tr>
</tbody>
</table>
ALTER DEFINER='root'@'localhost' FUNCTION `get_customer_balance` (p_customer_id INT, p_effective_date DATE)
    READS SQL DATA
    DETERMINISTIC
    BEGIN

    # OK, WE NEED TO CALCULATE THE CURRENT BALANCE GIVEN A CUSTOMER_ID AND A DATE
    # THAT WE WANT THE BALANCE TO BE EFFECTIVE FOR. THE BALANCE IS:
    # 1) RENTAL FEES FOR ALL PREVIOUS RENTALS
    # 2) ONE DOLLAR FOR EVERY DAY THE PREVIOUS RENTALS ARE OVERDUE
    # 3) IF A FILM IS MORE THAN RENTAL_DURATION * 2 OVERDUE, CHARGE THE REPLACEMENT_COST
    # 4) SUBTRACT ALL PAYMENTS MADE BEFORE THE DATE SPECIFIED

    DECLARE v_rentfees DECIMAL(5,2);  # FEES PAID TO RENT THE VIDEOS INITIALLY
    DECLARE v_overfees INTEGER;       # LATE FEES FOR PRIOR RENTALS
    DECLARE v_payments DECIMAL(5,2);  # SUM OF PAYMENTS MADE PREVIOUSLY

    SELECT IFNULL(SUM(film.rental_rate),0) INTO v_rentfees
        FROM film, inventory, rental
        WHERE film.film_id = inventory.film_id
            AND inventory.inventory_id = rental.inventory_id
            AND rental.rental_date <= p_effective_date
            AND rental.customer_id = p_customer_id;

    SELECT IFNULL(SUM(IF((TO_DAYS(rental.return_date) - TO_DAYS(rental.rental_date)) > film.rental_duration,0)),0) INTO v_overfees
        FROM rental, inventory, film
        WHERE film.film_id = inventory.film_id
            AND inventory.inventory_id = rental.inventory_id
            AND rental.rental_date <= p_effective_date
            AND rental.customer_id = p_customer_id;

    SELECT IFNULL(SUM(payment.amount),0) INTO v_payments
        FROM payment
        WHERE payment.payment_date <= p_effective_date
            AND payment.customer_id = p_customer_id;

    # FINISH THE BALANCE AS CALCULATED ABOVE.
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace EF_To_WindowsFormData
{
    public partial class Form1 : Form
    {
        worldEntities we;

        public Form1()
        {
            InitializeComponent();
        }

        private void cityDataGridView_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            we = new worldEntities();
            cityBindingSource.DataSource = we.cities;
        }
    }
}
The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.
Installation Overview

- Download source file
- Add MySQL user and group
- Select directory to uncompressed
- Navigate to top-level directory
- Configure and compile
- Install the distribution
- Navigate to installation directory
- Set permissions
- Create data directory and initialize GRANT tables
- Start Server
### Download Source

**dev.mysql.com/downloads/mysql/#downloads**

#### MySQL Community Server 5.1.47

<table>
<thead>
<tr>
<th>Source Code</th>
<th>Development Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SuSE Linux Enterprise Server var. 11</strong> (Architecture Independent), RPM Package</td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.11-0.rpm)</td>
<td>MD5: 0a1f5e7b7b0e235d5280b1c0704989a</td>
</tr>
<tr>
<td><strong>Red Hat Enterprise Linux 5 (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.11-0.rpm)</td>
<td>MD5: a089277b240d49a51f3ed580f91ebd</td>
</tr>
<tr>
<td><strong>SuSE Linux Enterprise Server 10 (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.10-0.rpm)</td>
<td>MD5: af4d7a80d4960d670f7e703d1e0a</td>
</tr>
<tr>
<td><strong>Generic Linux (glibc 2.3) (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.0-0.rpm)</td>
<td>MD5: 10be287e3e55d6b9f7f00d64c77</td>
</tr>
<tr>
<td><strong>SuSE Linux Enterprise Server 9 (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.9-0.rpm)</td>
<td>MD5: 10be287e3e55d6b9f7f00d64c77</td>
</tr>
<tr>
<td><strong>Red Hat Enterprise Linux 4 (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.4-0.rpm)</td>
<td>MD5: b60fadb920e55a95f9e0e6e677</td>
</tr>
<tr>
<td><strong>Red Hat Enterprise Linux 3 (Architecture Independent), RPM Package</strong></td>
<td>5.1.47</td>
</tr>
<tr>
<td>(MySQL Community 5.1.47-1.3-0.rpm)</td>
<td>MD5: 618e6b12b231a27f7c83d547b</td>
</tr>
</tbody>
</table>

---

**Oracle**

102
Steps to Build MySQL on Linux / UNIX

- **Create MySQL User and Group**
  > `groupadd mysql`
  > `useradd -g mysql mysql`
- **Uncompress Source tar**
  > `gunzip < mysql-VERSION.tar.gz | tar –xvf –`
  > `cd mysql-VERSION`
- **Configure, Compile and Install**
  > `./configure --prefix=/usr/local/mysql`
  > `make`
  > `make install`
- **Set Up Option File**
  > `cp support-files/my-medium.cnf /etc/my.cnf`
- **Set Permissions**
  > `cd usr/local/mysql`
  > `chown -R mysql .`
  > `chgrp -R mysql .`
Steps to Build MySQL on Linux / UNIX (cont.)

Data Directory and GRANT tables
> bin/mysql_install_db –user=mysql

Set Up Option File
> chown –R root .
> chown –R mysql var

Start MySQL Server
> bin/mysqlld_safe –user=mysql &
Installing from Development Tree (Linux)
Configure the Build Machine

- Bazaar
- GNU make
- autoconf 2.58 or greater
- automake 1.81
- libtool 1.5
- m4
- bison
Building MySQL From Dev Tree on Linux

- **Bazaar Repo and Initialize Directory**
  - mkdir mysql-server
  - bzr init-repo --trees mysql-server

- **Create Local Tree**
  - cd mysql-server
  - bzr branch lp:mysql-server/5.5 mysql-5.5

- **Copy Active Branch**
  - bzr branch mysql-5.5 mysql-5.5-build

- **Prepare Source Tree for Configuration**
  - cd mysql-5.5
  - autoreconf --force --install

- **Configure and Compile**
  - ./configure [OPTIONS]

- **Make**
  - make install

- **Make Install**
  - ./configure [OPTIONS]
  - make
Installing from Source

(Windows)
Configure the Build Machine

- Bazaar
- Visual Studio Express
- Windows Platform SDK
- CMake
- Bison for Windows
- Cygwin
- WiX
Building MySQL on Windows

- Configure the Build Machine
- Obtain MySQL Source
- Build the Source – Configure CMake
- Build the Source – Create VS Solution
- Build the Source – Build the Solution
- Test the Build – Run the Test Suite
### Download MySQL Source Files

![MySQL Download Page]

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Size</th>
<th>Download</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic Linux (glibc 2.3) (Architecture Independent), RPM Package</td>
<td>5.1.47</td>
<td>22.1MB</td>
<td><img src="https://example.com" alt="Download" /></td>
</tr>
<tr>
<td>SubE Linux Enterprise Server 9 (Architecture Independent), RPM Package</td>
<td>5.1.47</td>
<td>22.1MB</td>
<td><img src="https://example.com" alt="Download" /></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 4 (Architecture Independent), RPM Package</td>
<td>5.1.47</td>
<td>22.1MB</td>
<td><img src="https://example.com" alt="Download" /></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 5 (Architecture Independent), RPM Package</td>
<td>5.1.47</td>
<td>22.1MB</td>
<td><img src="https://example.com" alt="Download" /></td>
</tr>
<tr>
<td>Generic Linux (Architecture Independent), Compressed TAR Archive</td>
<td>5.1.47</td>
<td>22.7MB</td>
<td><img src="https://example.com" alt="Download" /></td>
</tr>
</tbody>
</table>

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

[MySQL Download Page]

Oracle 2010. Oracle Corporation and/or its affiliates.
Uncompress Download

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
<th>Date Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql-5.1.47</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:37 PM</td>
</tr>
<tr>
<td>BUILD</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>client</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>cmd-line-utils</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>config</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>dbug</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>Docs</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>extra</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>include</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>libmysql</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>libmysql-r</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>libmysqld</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>man</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:47 PM</td>
</tr>
<tr>
<td>mysql-test</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>msys</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>netware</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>plugin</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:49 PM</td>
</tr>
<tr>
<td>pstack</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>regex</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>scripts</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:49 PM</td>
</tr>
<tr>
<td>server-tools</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>std</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>std-bench</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>std-common</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>storage</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>strings</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>support-files</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
<tr>
<td>tests</td>
<td></td>
<td>File Folder</td>
<td>5/25/2010 2:48 PM</td>
</tr>
</tbody>
</table>
Build the Source – Configure CMake

- `win/configure.js` used to create the configuration file (`win/configure.data`) used by CMake

- Configuration options are defined in `win/README`

- Example:

  ```
  'cscript win/configure.js WITH_INNODB_BASE_STORAGE_ENGINE 
      WITH_PARTITION_STORAGE_ENGINE 
      WITH_ARCHIVE_STORAGE_ENGINE 
      WITH_BLACKHOLE_STORAGE_ENGINE 
      WITH EXAMPLE_STORAGE_ENGINE 
      WITH_FEDERATED_STORAGE_ENGINE __NT__'
  ```
Build the Source – Create VS Solution

- CMake uses files named CMakeLists.txt in each directory to create the necessary solution/project files
- Three batch files are provided to create Visual Studio solutions
  - win/build_vs71.bat
  - win/build_vs8.bat
  - win/build_vs8_x64.bat
Create Visual Studio Solution

Microsoft Windows XP (Version 5.1.2600)
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\jguerrero> cd c:\workdir

C:\workdir>win\build-us9.bat
build CSU as static library
build HEAP as static library
build MYISAM as static library
build MYISAMMRG as static library
build ARCHIVE as DLL
build BLACKHOLE as DLL
build EXAMPLE as DLL
build FEDERATED as DLL
build INNODB as DLL
build INNODB_PLUGIN as DLL
-- Configuring done
-- Generating done
-- Build files have been written to: C:\workdir
C:\workdir>
Build the Solution

- **Open the solution ‘MySQL.sln’ with Visual Studio**
- **Choose the configuration**
  - Official MySQL builds use Debug and RelWithDebInfo
- **Click Build > Build Solution (F7) to start the build**
Test the Build – Run the Test Suite

- Microsoft allows building different configurations of the source in the same tree
- Useful for mix-match, release/debug, client/server testing
- Mysql-test-run(mtr) will try to guess which configuration but better to be specific with –vsconfig dir option
Test the Build – Debugging

• Debugging MySQL in Visual Studio can be awkward but well worth the effort

• If you haven’t already done so, build the solution’s Debug or RelWithDebInfo configuration

• MySQL Test suite has a ‘-manual-debug’ option which displays configuration options needed to run the server in the debugger
# Small: Web Reference Architecture

<table>
<thead>
<tr>
<th></th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
<th>Extra Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries/Second</td>
<td>&lt;500</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Transactions/Second</td>
<td>&lt;100</td>
<td>&lt;1,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Read Users</td>
<td>&lt;100</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Write Users</td>
<td>&lt;10</td>
<td>&lt;100</td>
<td>1,000+</td>
<td>2,500+</td>
</tr>
</tbody>
</table>

### Database Size

<table>
<thead>
<tr>
<th></th>
<th>Sessions</th>
<th>eCommerce</th>
<th>Analytics</th>
<th>Content Management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;2 GB</td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>&lt;10 GB</td>
</tr>
<tr>
<td></td>
<td>&lt;10 GB</td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>&lt;500 GB</td>
</tr>
<tr>
<td></td>
<td>20+ GB</td>
<td>20+ GB</td>
<td>1+ TB</td>
<td>1+ TB</td>
</tr>
<tr>
<td></td>
<td>40+ GB</td>
<td>40+ GB</td>
<td>2+ TB</td>
<td>2+ TB</td>
</tr>
</tbody>
</table>
Small: Web Reference Architecture

- **Single server supporting all workloads**
- **Data replicated to slaves for back-up & analysis**

**Applications**
- **Members/Authentication**
  - eCommerce
- **Content Management**
  - Search

Only deploy when future traffic growth is very limited
Best Practices

Small Web Reference Architecture

• If future scalability is required, start with the Medium Reference Architecture
  • Complex to tune multiple applications on shared hardware

• Use default InnoDB storage engine for all workloads
  • Default MySQL storage engine
  • ACID Compliant, Transactional
  • MVCC & Row-Level Locking
  • Foreign Keys & constraints

• If traffic volumes increase, scale session management first
  • Migrate Session Management to a dedicated MySQL server
# Large: Web Reference Architecture

<table>
<thead>
<tr>
<th></th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
<th>Extra Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries/Second</td>
<td>&lt;500</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Transactions/Second</td>
<td>&lt;100</td>
<td>&lt;1,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Read Users</td>
<td>&lt;100</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Write Users</td>
<td>&lt;10</td>
<td>&lt;100</td>
<td>1,000+</td>
<td>2,500+</td>
</tr>
<tr>
<td>Database Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sessions</td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>20+ GB</td>
<td>40+ GB</td>
</tr>
<tr>
<td>eCommerce</td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>20+ GB</td>
<td>40+ GB</td>
</tr>
<tr>
<td>Analytics</td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>1+ TB</td>
<td>2+ TB</td>
</tr>
<tr>
<td>Content Management</td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>1+ TB</td>
<td>2+ TB</td>
</tr>
</tbody>
</table>
Large: Web Reference Architecture
Conceptual View

East Coast Data Center
- Session Mgmt
- eCommerce
- Data Refinery
- Content Mgmt
- Analytics

West Coast Data Center
- Session Mgmt
- eCommerce
- Data Refinery

Geographic Replication
Large: Web Reference Architecture

Session Management

MySQL Servers

Node Group 1
Node 3
Node 4
Node Group 2
Node 3
Node 4
MySQL Cluster Data Nodes

eCommerce

MySQL Servers

Node Group 1
Node 3
Node 4
Node Group 2
Node 3
Node 4
MySQL Cluster Data Nodes

Data Refinery

MySQL Master

Content Management

Memcache / Application Servers

Slave N

Slave 1
Slave 2
Slave 3
Slave 4
Slave 5
Slave 6
Slave 7
Slave 8
Slave 9
Slave 10

Analytics

MySQL Master

MySQL Enterprise Monitor

MySQL Enterprise Backup

Analytics

MySQL Master

MySQL Enterprise Monitor

MySQL Enterprise Backup

SAN

Distributed Storage

XOR
Best Practices
Large Web Reference Architecture

• Builds on best practices of Medium Web Ref Arch
  • Dedicated infrastructure for each workload, MySQL Replication, Memcached, etc.

• Introduces Data Refinery
  • Aggregate data across the web components
  • Data cleansing
  • Builds Data Warehouse Dimensions
  • Supports higher volume content management and analytics

• Introduces MySQL Cluster
  • Session Management and eCommerce
## Extra Large: Social Network Reference Architecture

<table>
<thead>
<tr>
<th></th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
<th>Extra Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries/Second</td>
<td>&lt;500</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Transactions/Second</td>
<td>&lt;100</td>
<td>&lt;1,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Read Users</td>
<td>&lt;100</td>
<td>&lt;5,000</td>
<td>10,000+</td>
<td>25,000+</td>
</tr>
<tr>
<td>Concurrent Write Users</td>
<td>&lt;10</td>
<td>&lt;100</td>
<td>1,000+</td>
<td>2,500+</td>
</tr>
</tbody>
</table>

### Database Size

<table>
<thead>
<tr>
<th></th>
<th>Sessions</th>
<th>eCommerce</th>
<th>Analytics</th>
<th>Content Management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;2 GB</td>
<td>&lt;10 GB</td>
<td>&lt;10 GB</td>
<td>&lt;10 GB</td>
</tr>
<tr>
<td></td>
<td>&lt;10 GB</td>
<td>&lt;500 GB</td>
<td>&lt;500 GB</td>
<td>&lt;500 GB</td>
</tr>
<tr>
<td></td>
<td>20+ GB</td>
<td>20+ GB</td>
<td>1+ TB</td>
<td>1+ TB</td>
</tr>
<tr>
<td></td>
<td>40+ GB</td>
<td>40+ GB</td>
<td>2+ TB</td>
<td>2+ TB</td>
</tr>
</tbody>
</table>
Best Practices
Social Networking Reference Architecture

• **Builds on best practices of Web Ref Archs**
• **MySQL Cluster used for authentication & Look-Up table (Shard Catalog)**
• **Introduces Sharding**
  • Implemented at the application layer for scaling very high volume of writes
  • Data divided into smaller sets, distributed across low-cost hardware
  • Shards based on Hash of a single column – ie. User ID
• **Sharding is complex**
  • Recommend the Architecture and Design Consulting Engagement
• **Sharding only used in a small percentage of workloads**
  • Most Web 2.0 workloads are still read-intensive, ie record is read before updates applied