CNT 4714: Enterprise Computing
Spring 2010

Introduction To MySQL

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cnt4714/spr2010

School of Electrical Engineering and Computer Science
University of Central Florida

MySQL RDBMS

- MySQL is a database server (although it does come with a set of simple client programs). The current stable version is 5.1.43 and can be downloaded from www.mysql.com. (Any of the versions of MySQL 5.1.32 or greater will be fine for our purposes.)

- It is typically used in thin client environments. In other words, it is used in client-server systems where the bulk of the processing and storage takes place on the server, and the client is little more than a dumb terminal.

- MySQL performs multithreaded processing, which means that multiple clients are allowed to connect to it and run queries simultaneously. This makes MySQL extremely fast and well suited to client-server environments such as Web sites and other environments that process numerous transactions for multiple users.

MySQL :: The world's most popular open source database - Windows Internet Explorer

http://www.mysql.com/

File Edit View Favorites Tools Help

Google Search Share Sidewiki Check Sign In

Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid- Download and s...

MySQL :: The world's most popular open source ...

Page Safety Tools

Recommended Servers for MySQL

Contact a MySQL Representative

Search

Login | Register

MySQL

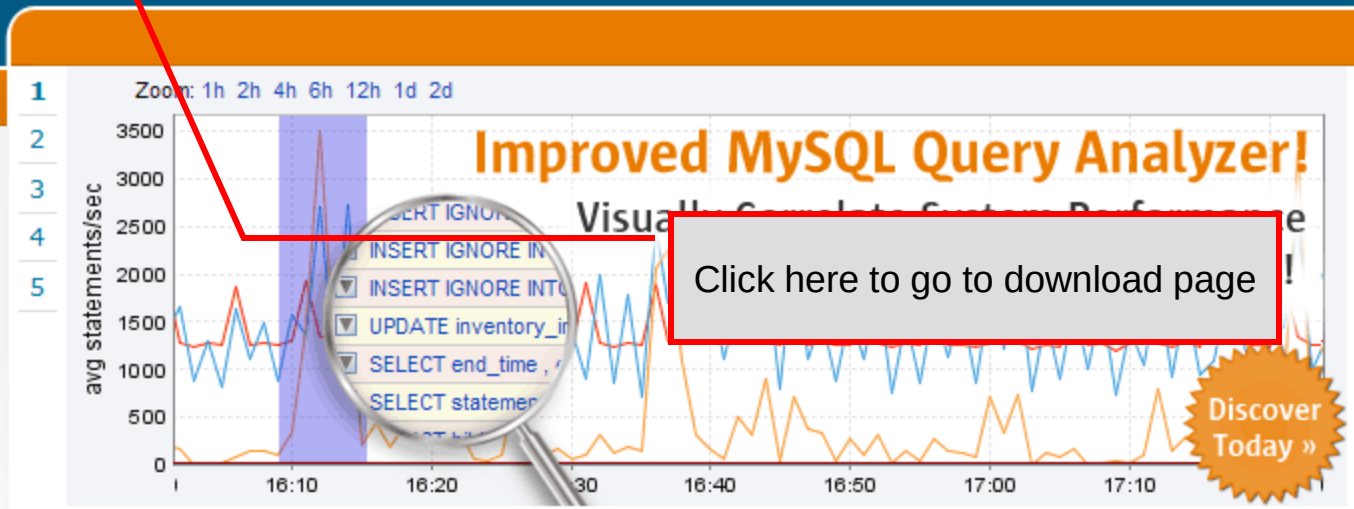
The world's most popular open source database

MySQL.com | Downloads (GA) | Partners & Solutions | Customer Login

Home | Products | Services | Customers | Why MySQL? | News & Events | About MySQL | How to Buy

GET STARTED

- MySQL Enterprise
- Free Web Seminars
- ISVs and OEMs
- Test Drive MySQL Query Analyzer
- MySQL Enterprise 30-day Trial
- MySQL Training
- Buy Now



Click here to go to download page

Free Web Seminars

Internet | Protected Mode: Off 100%

This should be the next page you see. Click this option to go to MySQL Community Server page.

MySQL :: MySQL Downloads (Generally Available) - Windows Internet Explorer

http://www.mysql.com/downloads/

File Edit View Favorites Tools Help

Google Search Share

Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid- Download and s...

MySQL :: MySQL Downloads (Generally Available)

Page Safety Tools

Recommended Servers for MySQL

Contact a MySQL Representative

Search

Login | Register

MySQL

The world's most popular open source database

MySQL.com | Downloads (GA) | Partners & Solutions | Customer Login

Overview | MySQL Server | MySQL Cluster | MySQL Workbench (GUI Tool) | MySQL Connectors

Downloads (GA) Overview

MySQL Downloads (Generally Available)

MySQL Community Server

(Current Generally Available Release: 5.1.43)

MySQL Community Server is a freely downloadable version of the world's most popular open source database that is supported by an active community of open source developers and enthusiasts.

New Releases (GA)

- MySQL Community Server 5.1 (5.1.43 GA)
- Connector/J 5.1 (5.1.11 GA)
- Connector/Net 6.2 (6.2.2 GA)
- MySQL Cluster 7.0

Internet | Protected Mode: Off 100%

The MySQL Community Server page.

MySQL :: Download MySQL Community Server - Windows Internet Explorer

http://www.mysql.com/downloads/mysql/

File Edit View Favorites Tools Help

Google Search Share

Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid- Download and s...

MySQL :: Download MySQL Community Server

Page Safety Tools

Recommended Servers for MySQL

Contact a MySQL Representative

Search

Login | Register

MySQL

The world's most popular open source database

MySQL.com | Downloads (GA) | Partners & Solutions | Customer Login

Overview | MySQL Server | MySQL Cluster | MySQL Workbench (GUI Tool) | MySQL Connectors

MySQL Community Server

MySQL Enterprise

MySQL Embedded Server

Download MySQL Community Server

MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services and support to ensure your business achieves the highest levels of reliability, security, and uptime. MySQL Enterprise includes the MySQL Enterprise Server software, which is the most reliable, secure and up-to-date version of the world's most popular open source database. Users also receive monthly rapid updates and quarterly service packs with the latest bug fixes of MySQL Enterprise Server.

MySQL Software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

Done

Internet | Protected Mode: Off

100%

MySQL :: Download MySQL Community Server - Windows Internet Explorer

http://www.mysql.com/downloads/mysql/

File Edit View Favorites Tools Help

Scroll down this page and select the proper platform for your system.

Germany:
+49 89 420 95 98 95
France: +33 1 70 61 48 95
Sweden: +46 730 207 871
Benelux: +31 6 25003558
Italy: +39 06-99268193
Israel: +31 6 25003558
Spain & Portugal:
+ 34 933905461
Other EMEA countries:
+353 1 6919191

Australia/NZ: +61 2 42314328
Asia Pacific: +81 3 5843 1140

Contact Us Online »
Renew Subscriptions »

Select Platform:

Microsoft Windows | Select

Windows (x86, 32-bit), ZIP Archive (mysql-5.1.43.zip)	5.1.43	25.5M	Download
	MD5: 2aedd2d80e5cc755134f4be85e5d1505		
Windows (x86, 32-bit), MSI Installer (mysql-essential-5.1.43-win32.msi)	5.1.43	39.0M	Download
	MD5: 9d5fb5144de1d89780531ccad7d8dbc2		
Windows (x86, 32-bit), MSI Installer (mysql-5.1.43-win32.msi)	5.1.43	104.9M	Download
	MD5: 1774e59012916cf835749741e396d1f7		
Windows (x86, 64-bit), MSI Installer (mysql-essential-5.1.43-winx64.msi)	5.1.43	31.7M	Download
	MD5: 4625c3449dae04583198929536 1ec5af		
Windows (x86, 32-bit), ZIP	5.1.43	113.4M	Download

Internet | Protected Mode: Off 100%

MySQL :: MySQL 5.1 GA - Windows Internet Explorer

http://dev.mysql.com/downloads/mysql/5.1.html#win32

File Edit View Favorites Tools Help

Google Search

Favorites Suggested Sites Free Hotmail Web Slice Gallery

MySQL :: MySQL 5.1 GA [Valid] Markup Validation...

Scroll down this page and select the proper version for your system and a site to begin download. There will be a registration type form at the top of the page…you can ignore this if you wish and go straight to the download site.

Windows downloads (platform notes)

Windows Essentials (x86)	5.1.38	38.3M	Pick a mirror
MD5: 5a077abefee447cbb271e2aa7f6d5a47 | Signature			
Windows MSI Installer (x86)	5.1.38	103.5M	Pick a mirror
MD5: 4011f11770fd4cbe2f96ed7b61b5d3ba | Signature			
Without installer (unzip in C:\)	5.1.38	126.9M	Pick a mirror
MD5: 78328ae74e464b0f333e6b0825c6fd6b | Signature			

Windows x64 downloads (platform notes)

Windows Essentials (AMD64 / Intel EM64T)	5.1.38	29.4M	Pick a mirror
MD5: 40b3eb7055f296c88f66262602d1645f | Signature			
Windows MSI Installer (AMD64 / Intel EM64T)	5.1.38	93.5M	Pick a mirror
MD5: 27ba306cde7df2711450cea983123084 | Signature			

Internet | Protected Mode: Off 100%

MySQL :: Download MySQL Workbench - Windows Internet Explorer

http://www.mysql.com/downloads/workbench/

File Edit View Favorites Tools Help

Google Search Share Sign In

Favorites Suggested Sites Free Hotmail Web Slice Gallery KeepVid

MySQL :: Download MySQL Workbench

Tools

Go back to the main download page and also download MySQL Administrator and MySQL Query Browser available from the GUI Tools option.

Recommended Servers for MySQL

Login | Register

MySQL

The world's most popular open source database

MySQL.com | Downloads (GA) | Partners & Solutions | Cu

Overview | MySQL Server | MySQL Cluster | MySQL Workbench (GUI Too

The Query Browser and Administrator come in a bundle with some other tools. Scroll down and select the correct option for your machine.

MySQL Workbench (GUI Tool)

Download MySQL Workbenc

MySQL Workbench provides DBAs and developers an integrated tools environment for:

- Database Design & Modeling
- SQL Development (replacing MySQL Query Browser)
- Database Administration (replacing MySQL Administrator)

MySQL Software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

The Community (OSS) Edition is available from this page under the GPL.

Done | Internet | Protected Mode: Off | 100%

MySQL :: MySQL Connectors - Windows Internet Explorer

http://www.mysql.com/downloads/connector/

File Edit View Favorites Tools Help

Once again, go back to the main download page and select Connectors.

MySQL :: MySQL Connectors

The world's most popular open source database

MySQL.com | Downloads (GA) | Partners & Solutions | Customer Login

Overview | MySQL Server | MySQL Cluster | MySQL Workbench (GUI Tool) | MySQL Connectors

Login | Register

Search

MySQL Connectors

- Connector Overview
- Connector/ODBC
- Connector/J
- Connector/Net
- Connector/MXJ
- Connector/C++
- Connector/C

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. Any system that works with ODBC or JDBC can use MySQL.

MySQL Software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

Connector/ODBC

Standardized database driver Windows, Linux, Mac OS X, and Unix platforms.

Done | Internet | Protected Mode: Off | 100%

MySQL :: MySQL Connectors - Windows Internet Explorer

http://www.mysql.com/downloads/connector/

File Edit View Favorites Tools Help

Download the Connector/J for use with Java applications.

MySQL Connectors

- Connector Overview
- Connector/ODBC
- Connector/J
- Connector/Net
- Connector/MXJ
- Connector/C++
- Connector/C

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. Any system that works with ODBC or JDBC can use MySQL.

MySQL Software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

Connector/ODBC

Standardized database driver Windows, Linux, Mac OS X, and Unix platforms.

Connector/J

Standardized database driver for Java platforms and development.

Connector/Net

Standardized database driver for .NET platforms and development.

Connector/MXJ

MBean for embedding the MySQL server in Java applications.

Contact Sales

USA/Canada - Toll Free:
+1-866-221-0634
USA - From abroad:

Installing MySQL 5.1.43

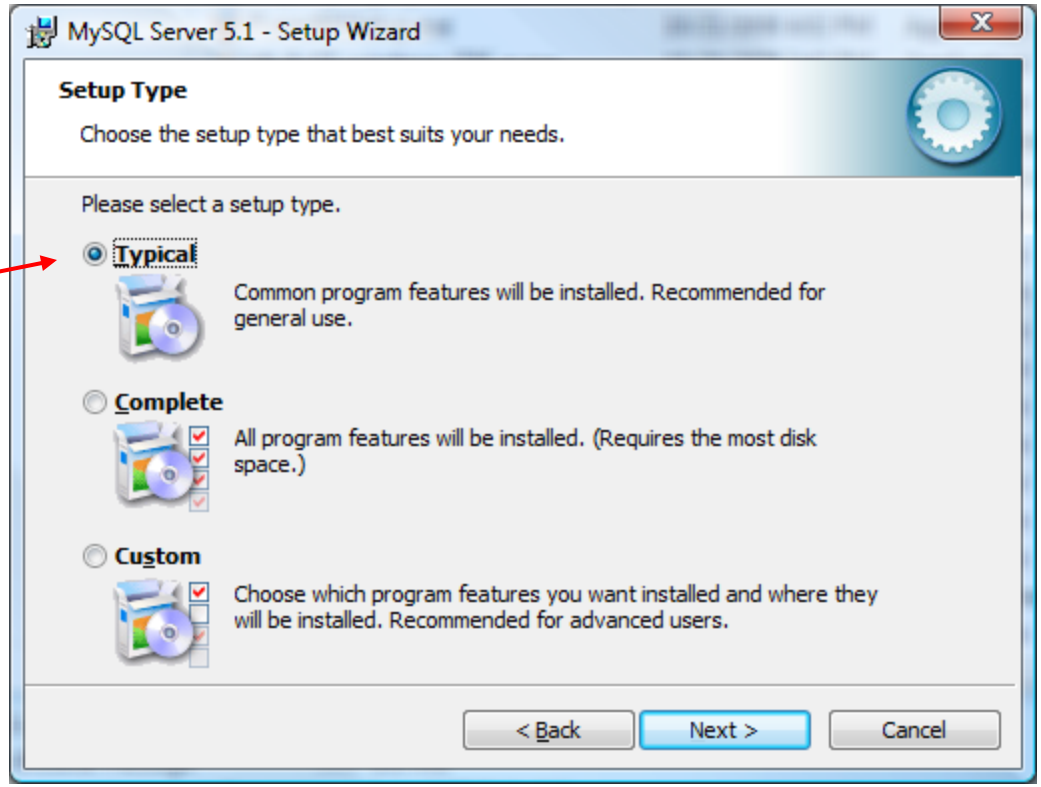
- Once you've got MySQL downloaded, go through the installation process. It may vary somewhat depending on platform.
- I've illustrated the basic install on Windows XP over the next few pages, just to give you an idea of what you should be seeing.

Installing MySQL 5.1.43

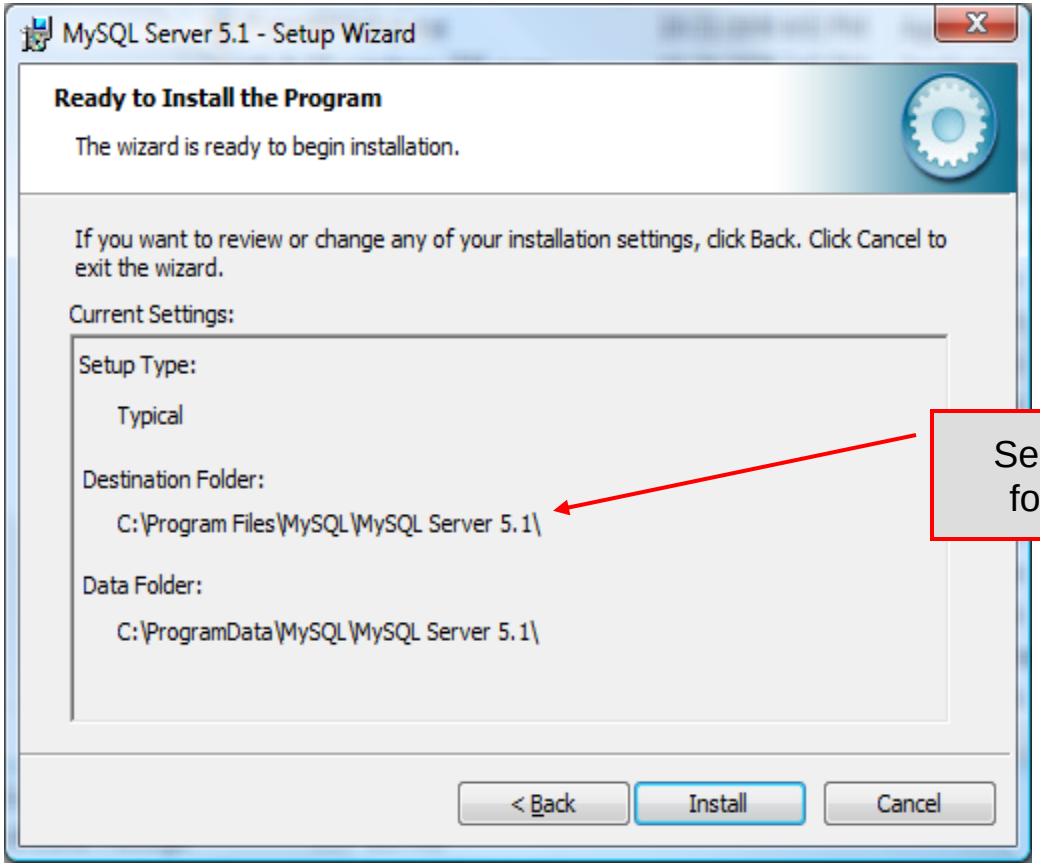
- Once you've got MySQL downloaded, go through the installation process. It may vary somewhat depending on platform.
- I've illustrated the basic install on Windows XP over the next few pages, just to give you an idea of what you should be seeing.
- Once the Window installer is running you should see the following window appear:

Installing MySQL 5.1.43 (cont.)

Your choice here. For this course, a typical set-up will be fine.



Installing MySQL 5.1.43 (cont.)



Select the destination folder for the install.

Installing MySQL 5.1.43 (cont.)

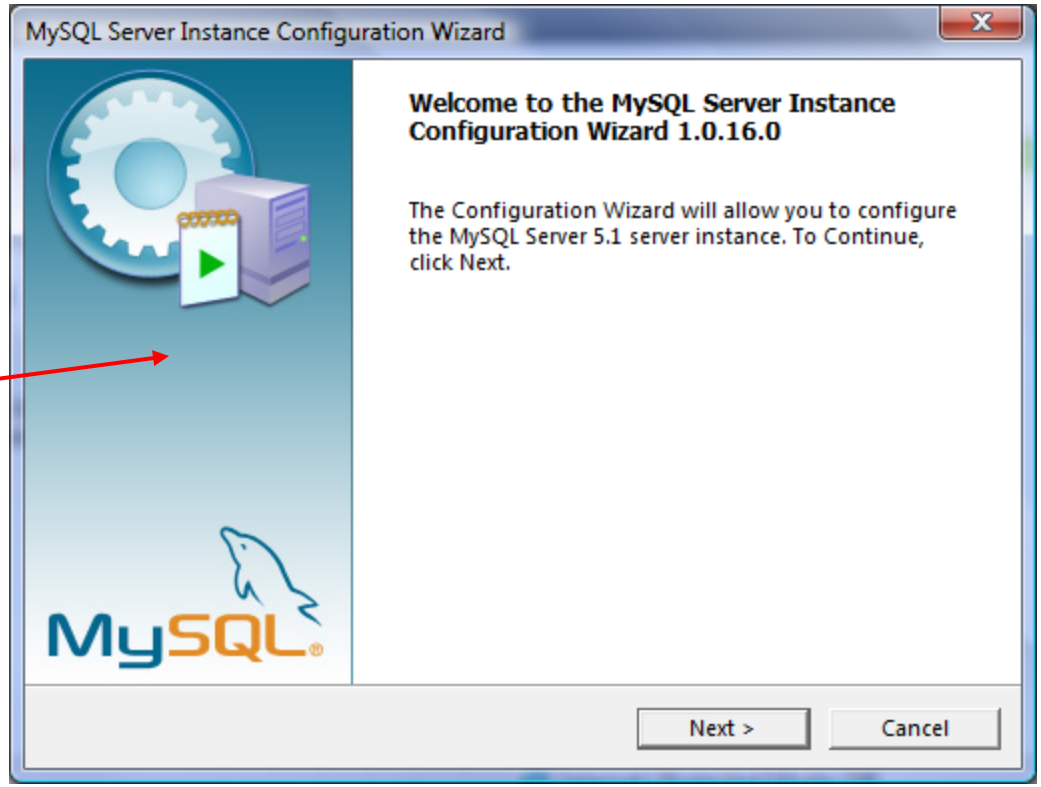
Again, your choice here. If you want to skip the sign-up that's fine.

Installing MySQL 5.1.43 (cont.)

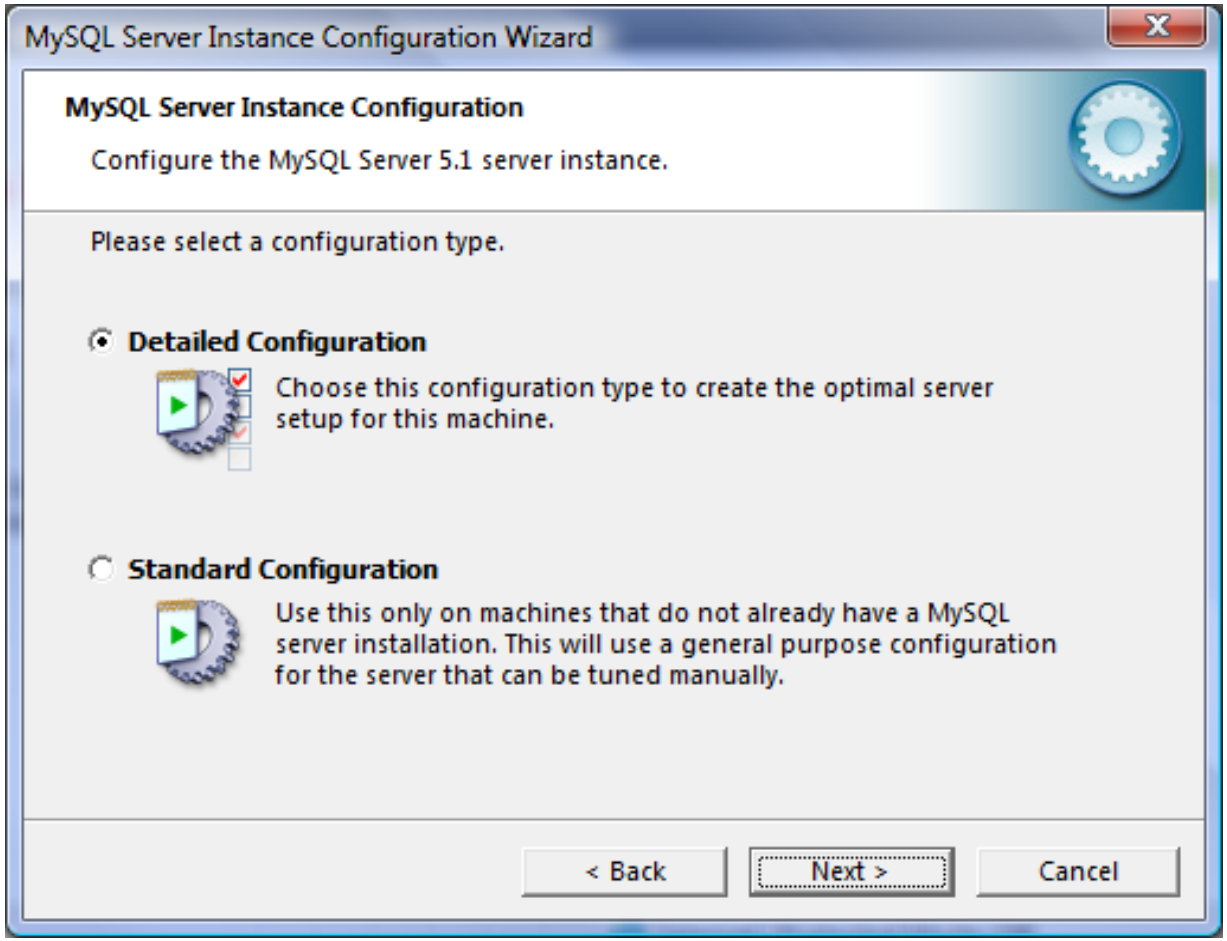
If everything has good well up to this point, you should see a window similar to this one. Click the Finish button, cross your fingers, and hang-on while the installer configures your system and gets MySQL up and running as a service.

Installing MySQL 5.1.43 (cont.)

Initial server configuration window



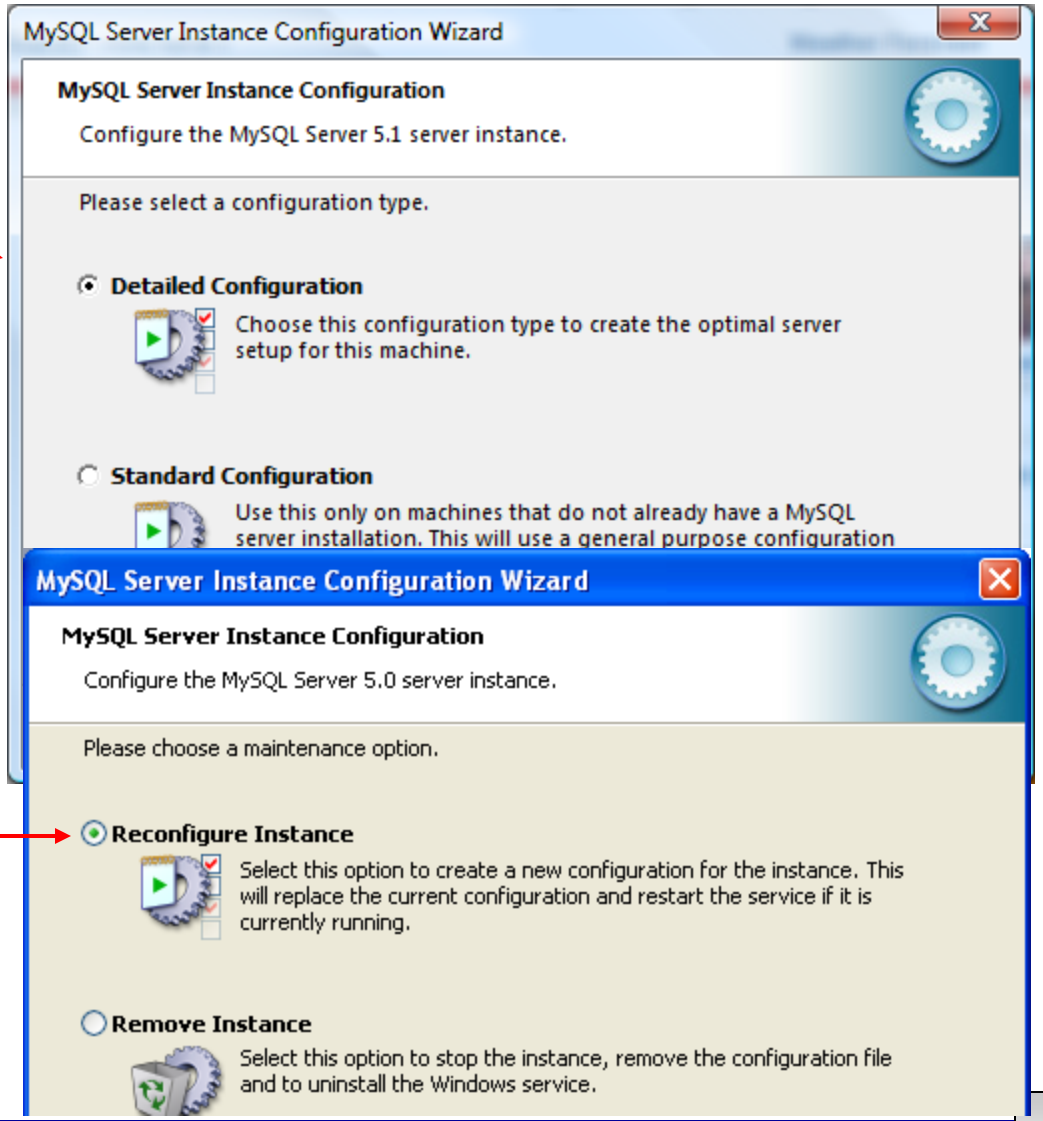
Installing MySQL 5.1.43 (cont.)



Installing MySQL 5.1.43 (cont.)

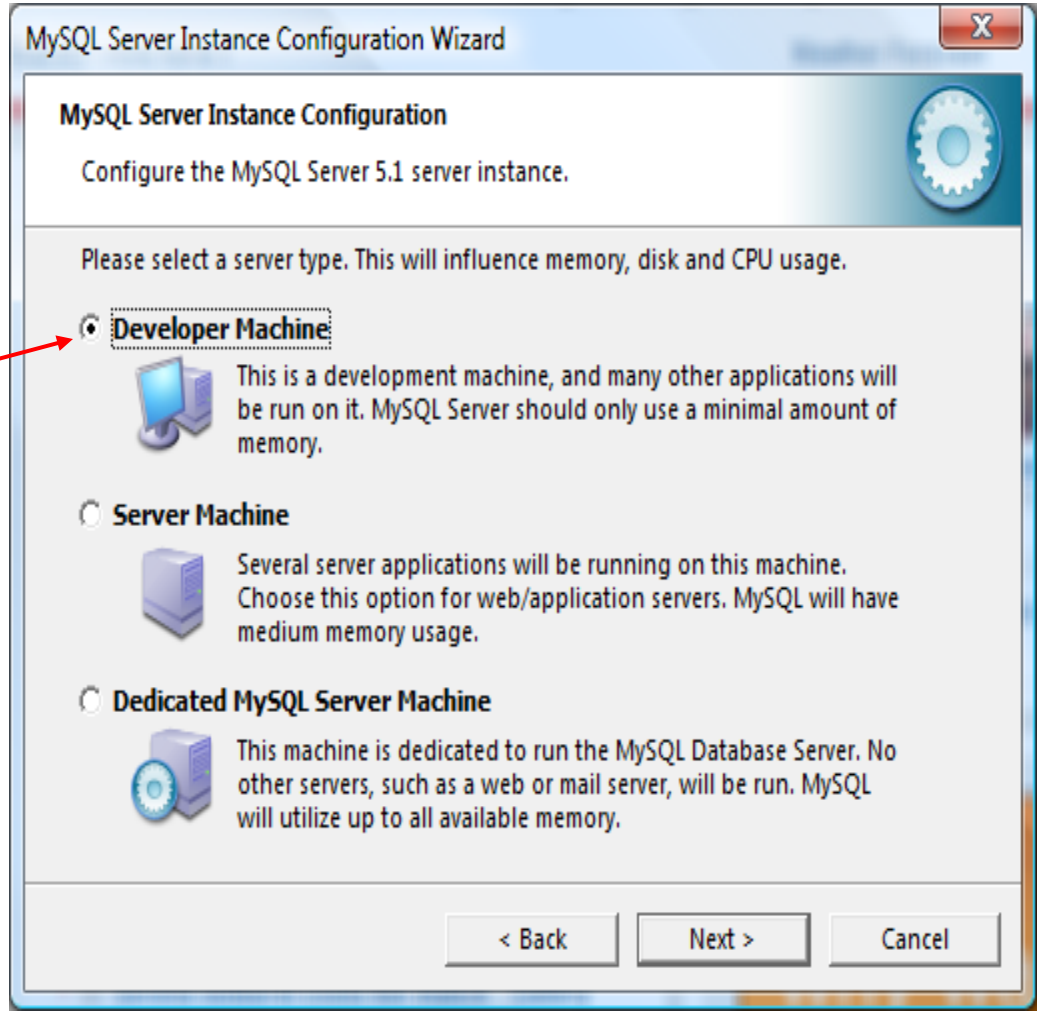
Your choice here. If you are not sure if there is already a MySQL server on your machine, choose the detailed configuration setting.

If you already have an instance of a MySQL server on your machine, you'll see this screen first, followed by the one above. Select reconfigure instance.



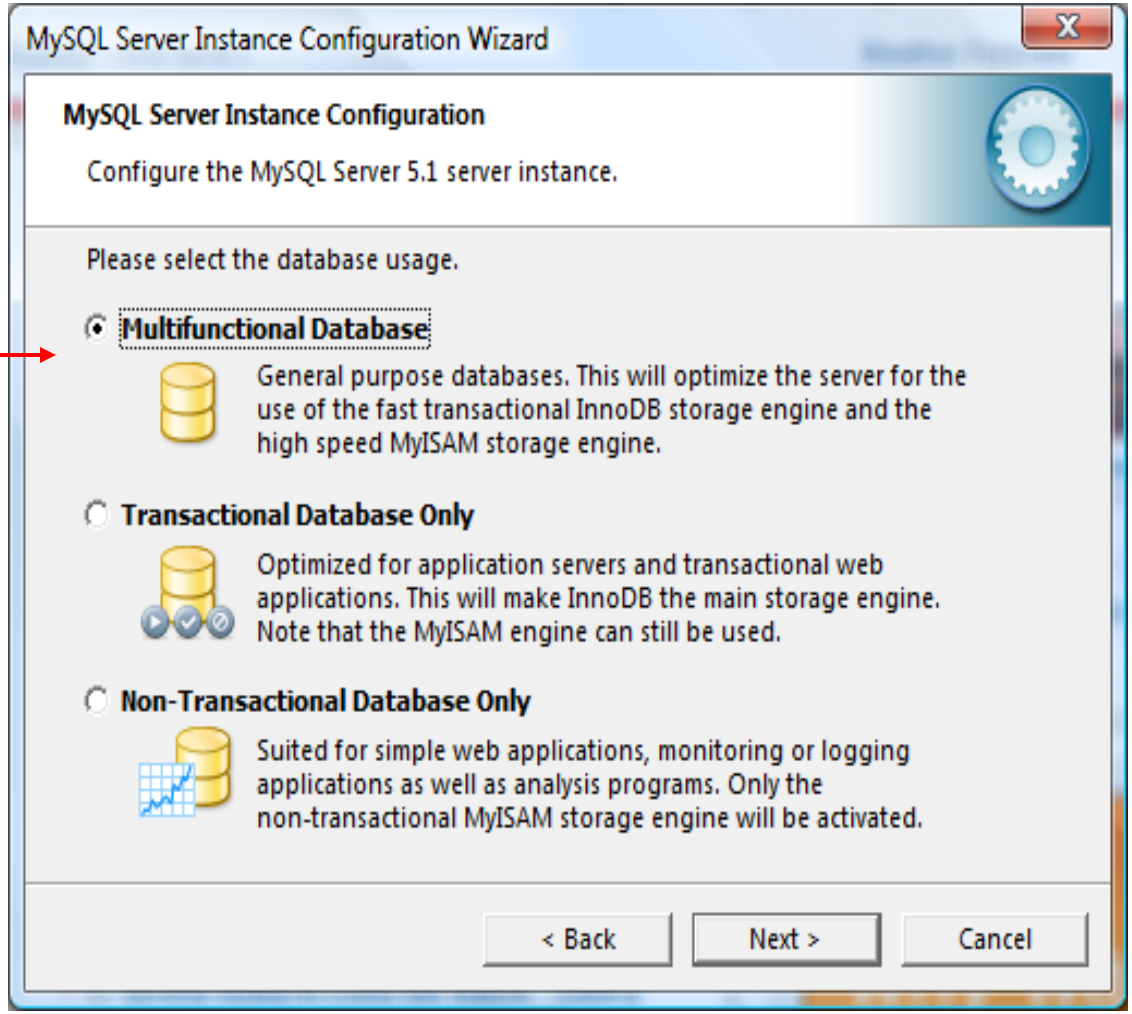
Installing MySQL 5.1.43 (cont.)

Choose the developer machine option



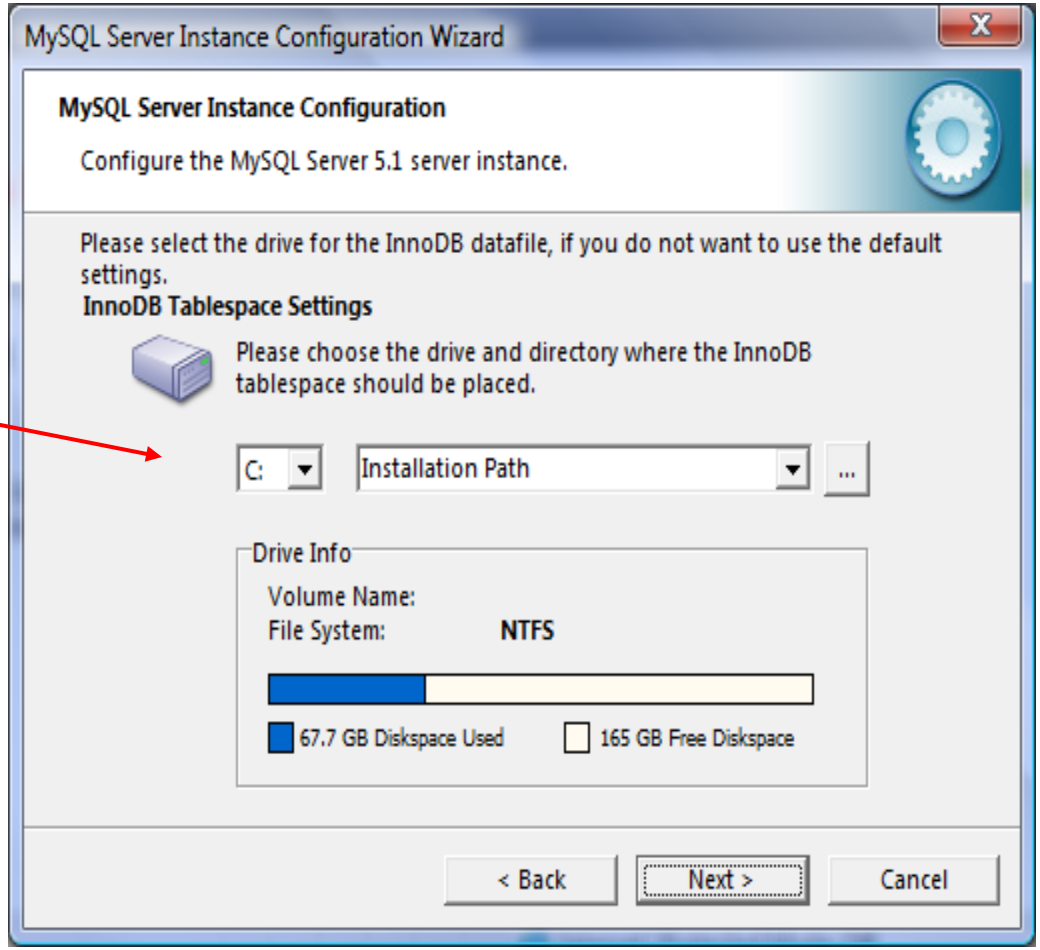
Installing MySQL 5.1.43 (cont.)

Choose the multifunctional database option



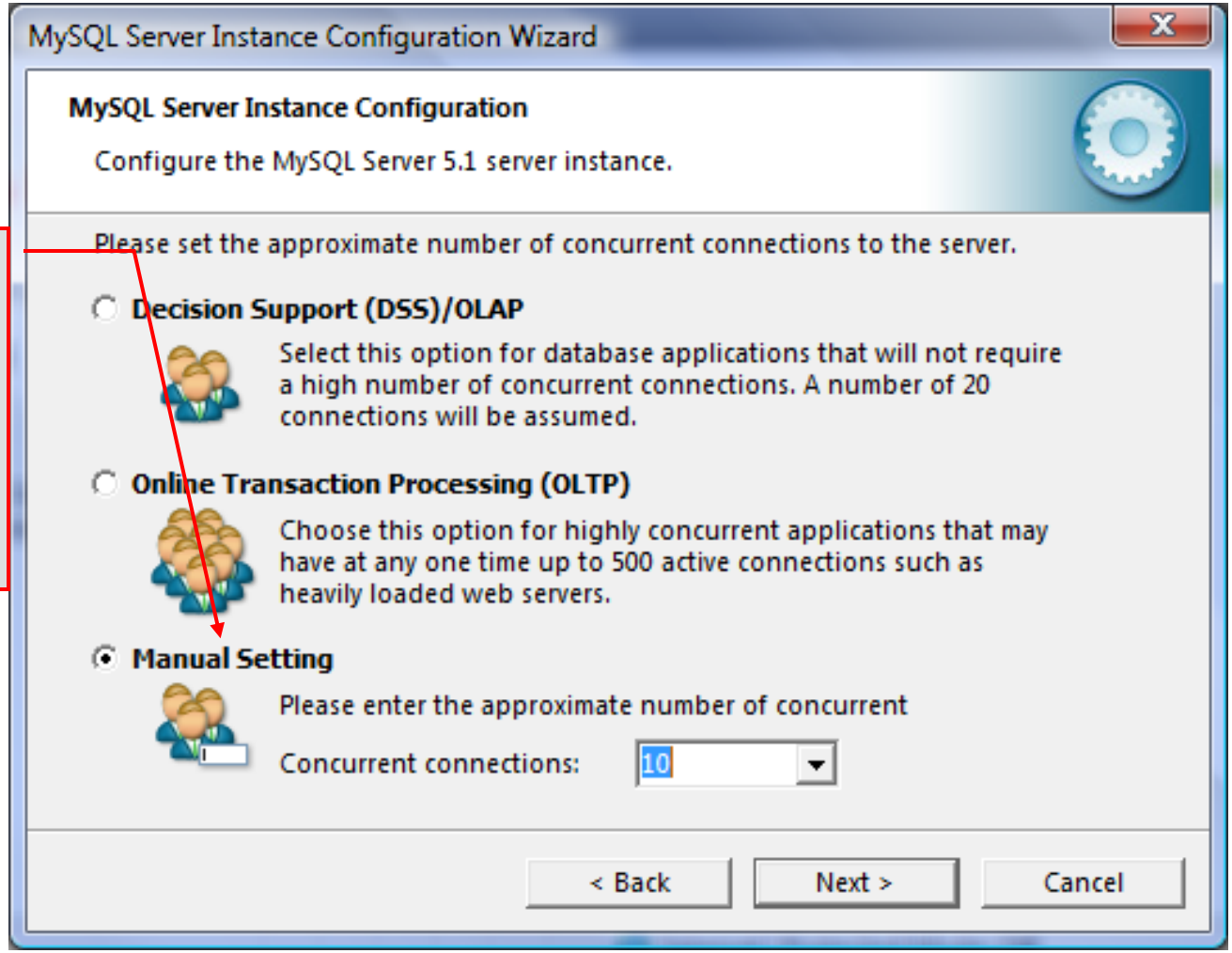
Installing MySQL 5.1.43 (cont.)

Choose the installation path to keep InnoDB tables in same area as other MySQL files



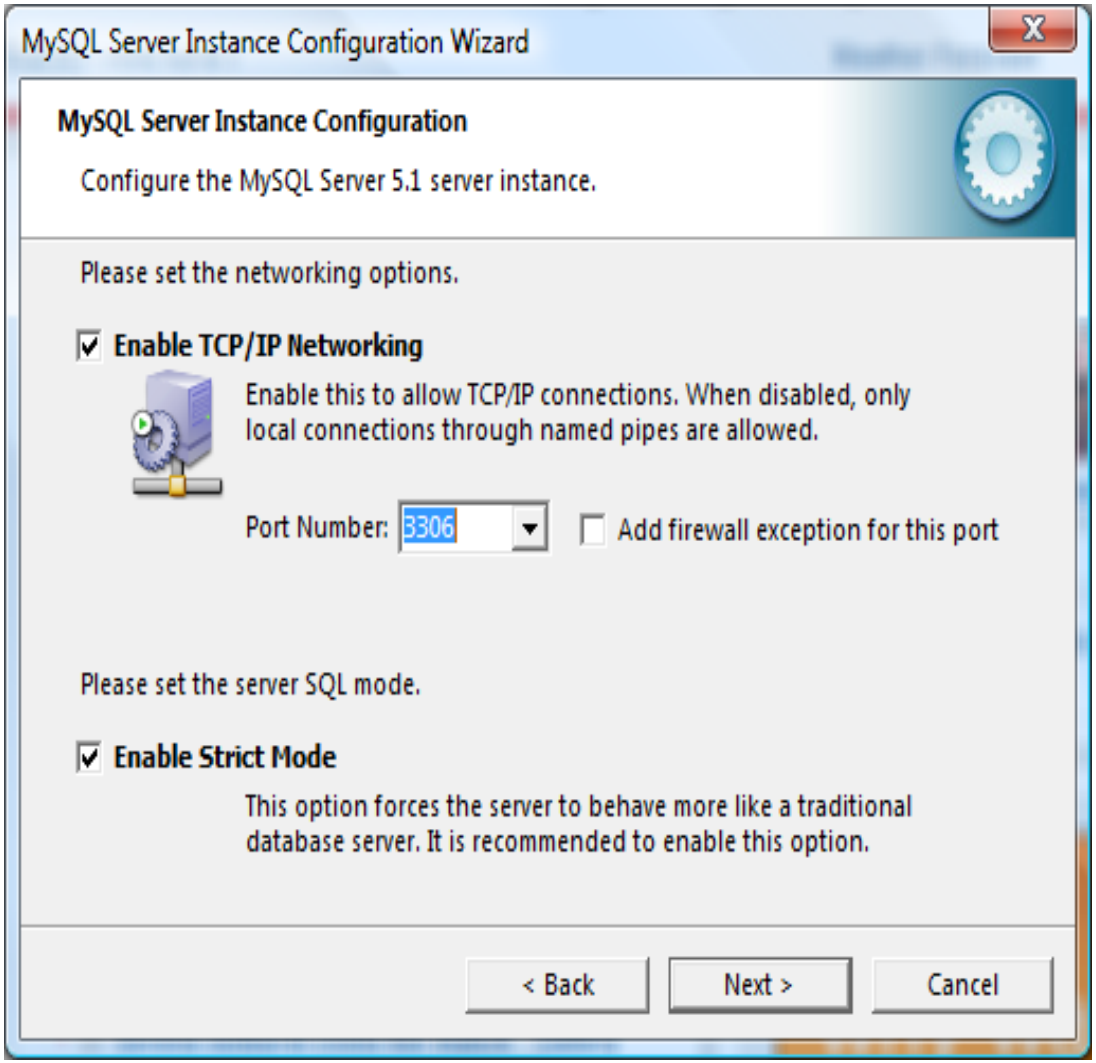
Installing MySQL 5.1.43 (cont.)

Select manual setting for this option. The default is 15, I set mine to 10, but you can use any number you would like, but pick something greater than 3 or 4.



Installing MySQL 5.1.43 (cont.)

Accept all defaults in this window



Installing MySQL 5.1.43 (cont.)

Your choice again

MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.1 server instance.

Please select the default character set.

Standard Character Set

Hello!

Makes Latin1 the default charset. This character set is suited for English and other West European languages.

Best Support For Multilingualism

日本語

Make 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.

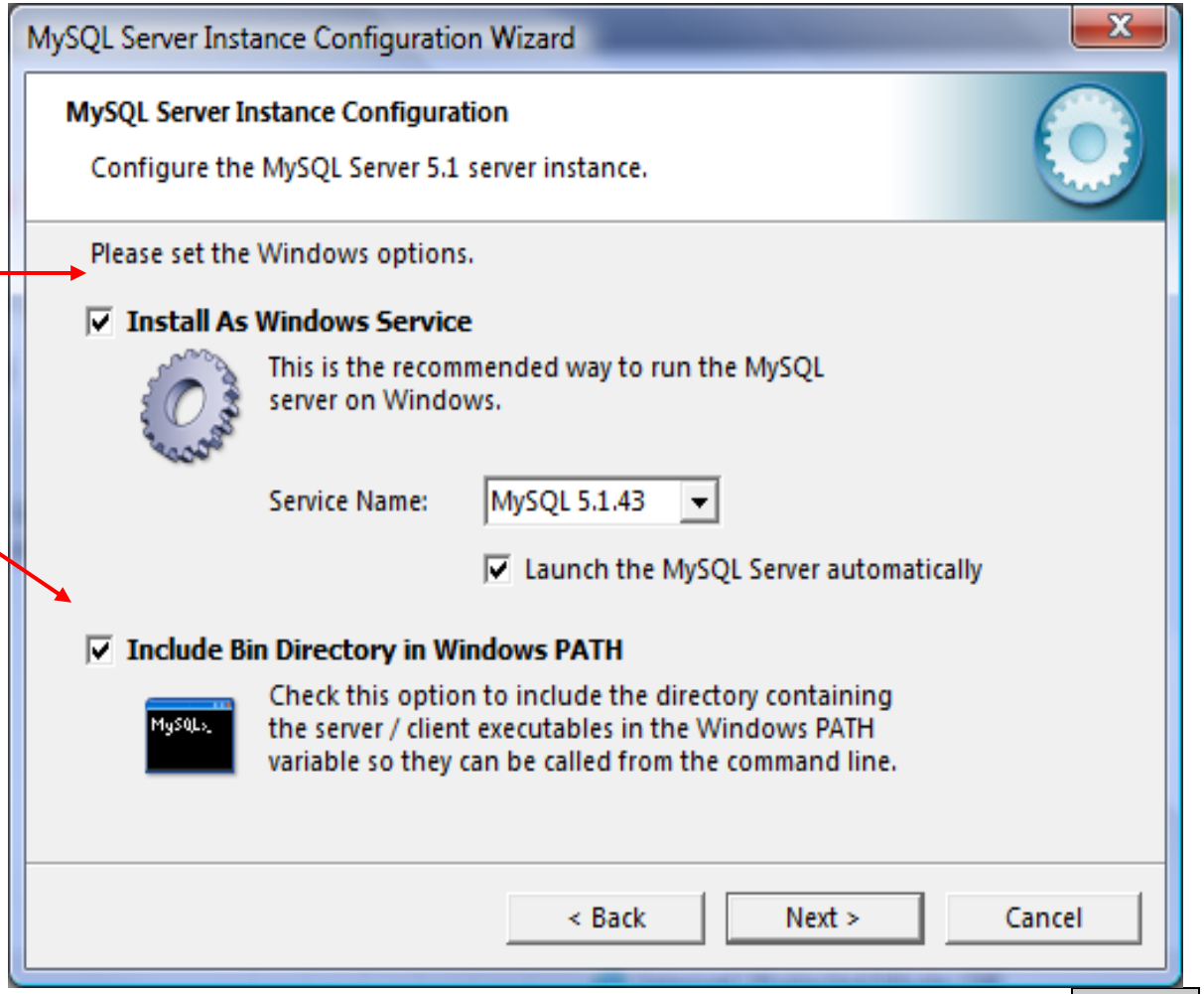
Character Set: latin1

< Back | Next > | Cancel

Installing MySQL 5.1.43 (cont.)

Accept default options

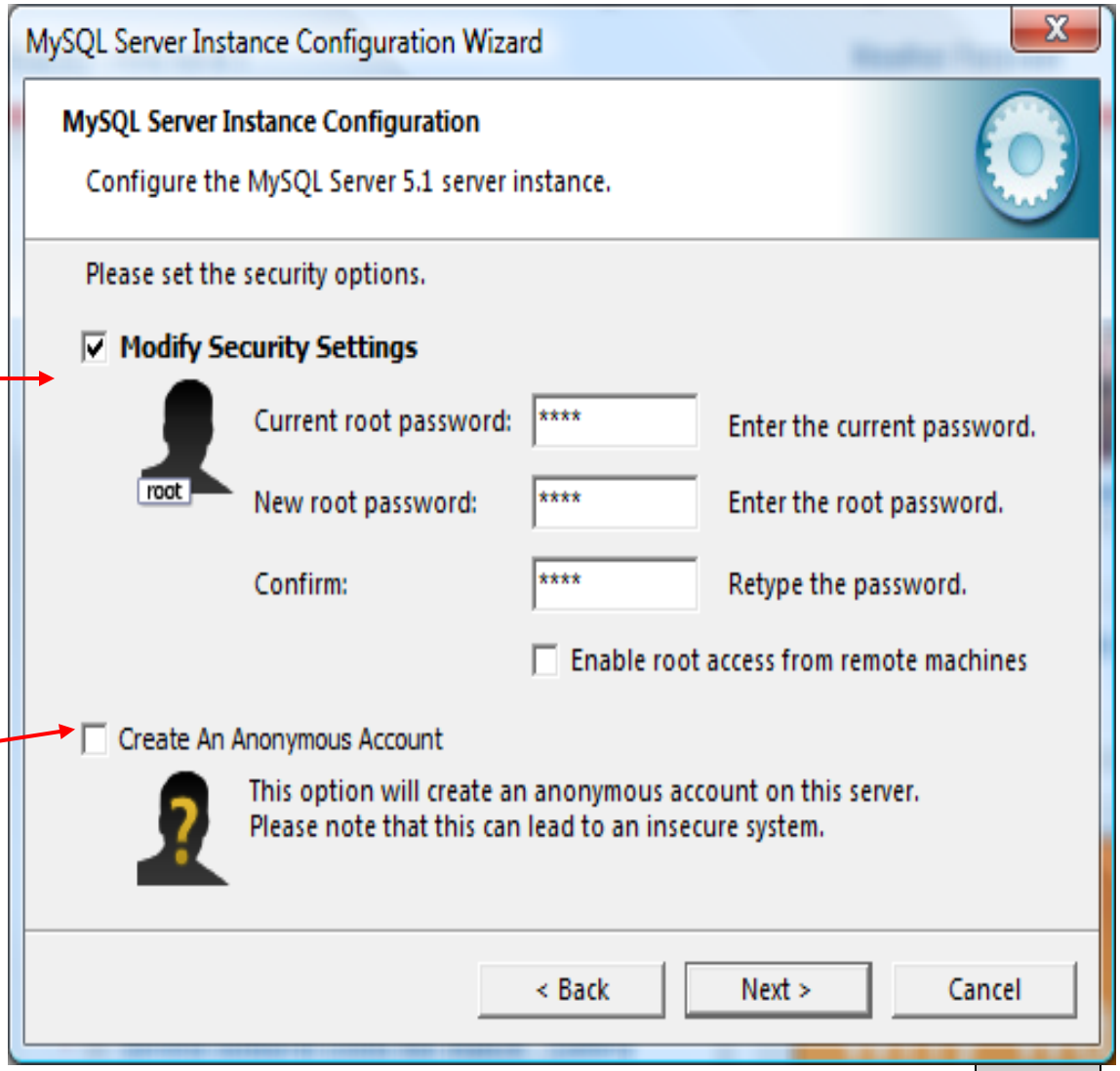
This option is not marked by default, but you can mark and accept it if you want to include MySQL file locations in your PATH statement.



Installing MySQL 5.1.43 (cont.)

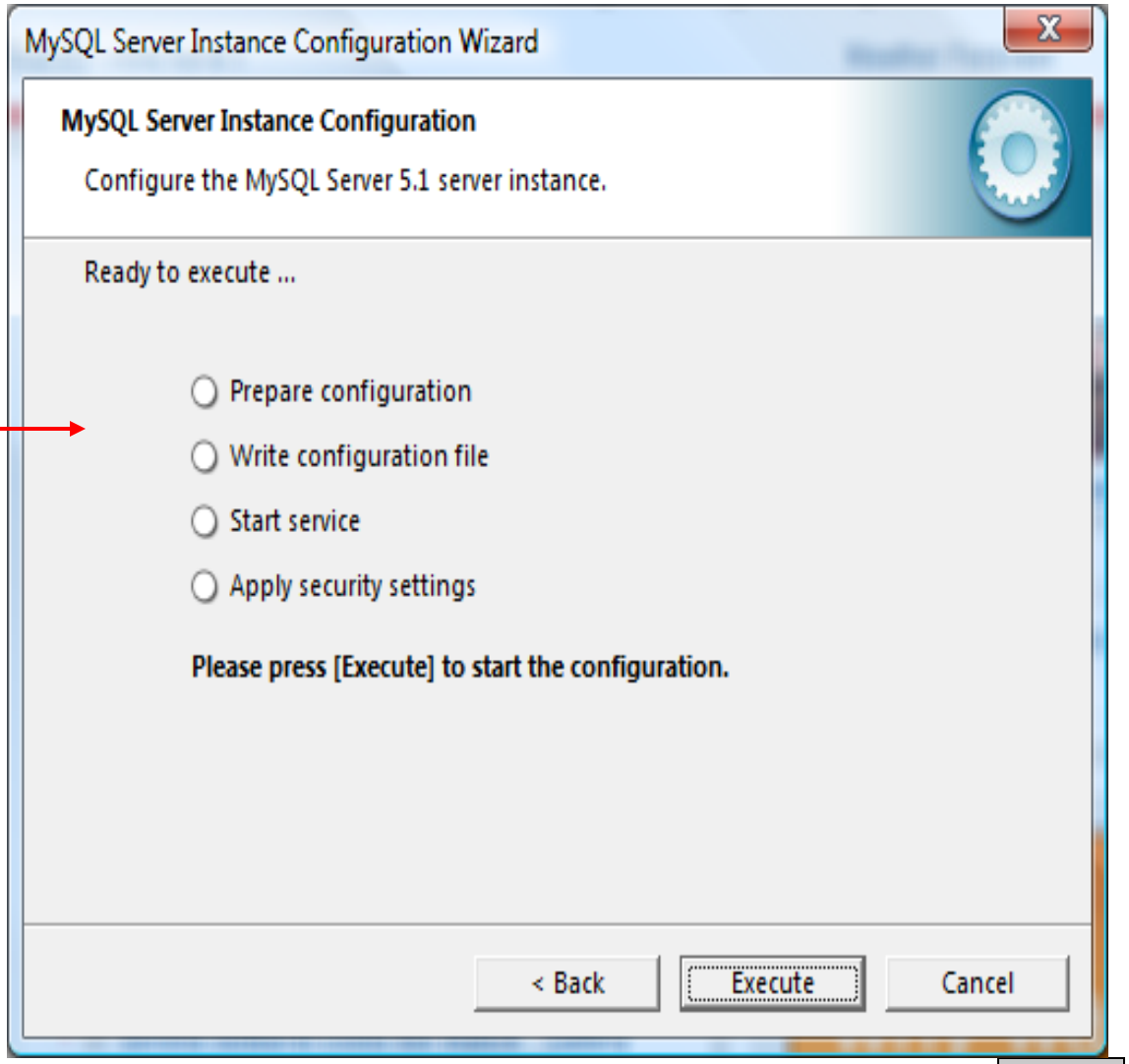
Accept default setting and enter a password for the root (superuser with all privileges by default). Enabling root access from remote machines is only necessary if you will be accessing the DB as the root user from a remote machine – we will not be doing this in this course.

Do not enable this option



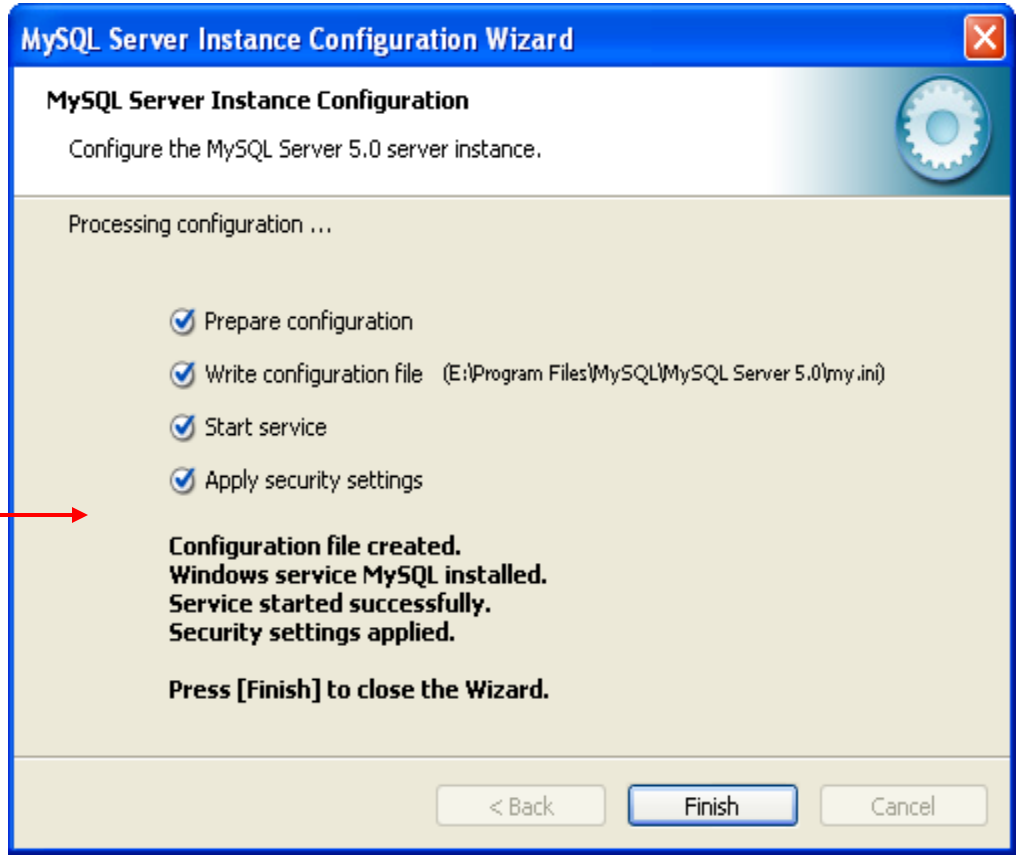
Installing MySQL 5.1.43 (cont.)

Configuration is about to begin. Now cross your fingers, toes, and anything else you have, take a deep breath, click the Execute button and close your eyes for a few seconds.



Installing MySQL 5.1.43 (cont.)

You've successfully installed MySQL!!



Running MySQL 5.1.43

- If you've successfully installed MySQL, it should now be running as a service on your machine. It will start automatically when your machine boots.
- Go into your listing of programs (from the start menu at the bottom: All Programs) and you should see MySQL appear. Since you will be running MySQL clients a lot, it will be easier if you pin the MySQL Client to the start menu.
- To verify that MySQL is running properly as a service you can either check the process window or run a MySQL client.

Running MySQL 5.1.43 (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
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> status
--------------
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe  Ver 14.14 Distrib 5.1.43,
(ia32)

Connection id:          1
Current database:
Current user:           ODBC@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.1.43-community MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    latin1
Conn.  characterset:    latin1
TCP port:               3306
Uptime:                 13 hours 22 min 0 sec

Threads: 1  Questions: 4  Slow queries: 0  Opens: 15  Flush tables: 1  Open tabl
ies per second avg: 0.0
--------------

mysql> _
```

Hopefully, you see this output from MySQL. The MySQL server is now awaiting a command from this client.

Running MySQL 5.1.43 (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
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> status;
--------------
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe  Ver 14.14 Distrib 5.1.43, for Win32
(ia32)

Connection id:          20
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.1.43-community MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Server characterset:    latin1
Db     characterset:    latin1
Client characterset:    latin1
Conn.  characterset:    latin1
TCP port:               3306
Uptime:                 14 hours 43 min 49 sec

Threads: 3  Questions: 204  Slow queries: 0  Opens: 23  Flush tables: 1  Open tables: 8  Qu
eries per second avg: 0.3
--------------

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| sample             |
| test               |
| testdb             |
+--------------------+
10 rows in set (0.00 sec)

mysql>
```

List all databases managed by this MySQL server which are accessible to this client.

Note: new installations will contain only 3 databases: `information_schema`, `mysql`, and `test`.

Running MySQL 5.1.43 (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
Server version: 5.1.43-community MySQL

Type 'help;' or '\h' for help. Type '\                                  ent.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| sample             |
| test               |
| testdb             |
+--------------------+
10 rows in set (0.00 sec)

mysql> exit;
```

List all databases managed by this MySQL server which are accessible to this client.

Terminate client connection.

Specifying A Database Within MySQL

- Unless, it is specifically stated, in the following slides we'll assume that the user has root-level privileges.
- To select a database for use in MySQL the `use` command must be issued. In the example below, we'll select the `bikedb` database.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| sample             |
| test               |
| testdb             |
+--------------------+
9 rows in set (0.00 sec)

mysql> create database bikedb;
Query OK, 1 row affected (0.00 sec)

mysql> use bikedb;
Database changed
mysql> _
```

MySQL acknowledges selection of bikedb database.

Viewing the Schema of a Relation

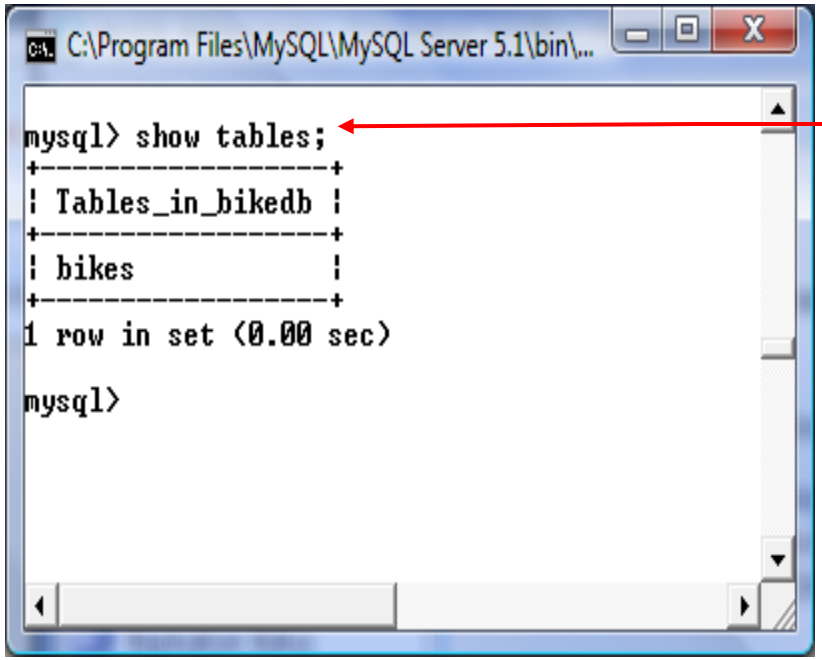
- To see the schema of a relation within a database, use the `describe <tablename>` command as illustrated below.

```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
mysql> use bikedb;
Database changed
mysql> create table bikes (
    -> bikename varchar(30) not null,
    -> size int (2),
    -> color varchar(15),
    -> cost int(5),
    -> purchased date,
    -> mileage int(6),
    -> primary key (bikename)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> describe bikes;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| bikename  | varchar(30) | NO   | PRI | NULL    |       |
| size      | int(2)      | YES  |     | NULL    |       |
| color     | varchar(15) | YES  |     | NULL    |       |
| cost      | int(5)      | YES  |     | NULL    |       |
| purchased | date        | YES  |     | NULL    |       |
| mileage   | int(6)      | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

Specify which table's schema to describe. All information regarding the schema visible to the user is displayed.

Viewing the Relations of a Database

- Once a database has been selected you can see the relations (tables) within that database with the `show tables` command as illustrated below.



Show tables command lists all the relations within a database visible to the user. There are two tables in this database.

Running a Simple Select Query in MySQL

- Within the MySQL monitor, running an SQL query is straight forward. The example below illustrates a simple selection query on the `bikes` table of the `bikedb` database.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from bikes;
+-------------------------+------+-------------+------+------------+---------+
| bikename                | size | color       | cost | purchased  | mileage |
+-------------------------+------+-------------+------+------------+---------+
| Colnago Dream Rabobank  |   60 | blue/orange | 5500 | 2002-07-07 |    4300 |
| Bianchi Evolution 3     |   58 | celeste     | 4800 | 2003-11-12 |    2000 |
| Eddy Merckx Molteni     |   58 | orange      | 5100 | 2004-08-12 |       0 |
| Eddy Merckx Domo        |   58 | blue/black  | 5300 | 2004-02-02 |       0 |
| Battaglin Carrera       |   60 | red/white   | 4000 | 2001-03-10 |   11200 |
| Gianni Motta Personal   |   59 | red/green   | 4400 | 2000-05-01 |    8700 |
| Gios Torino Super       |   60 | blue        | 2000 | 1998-11-08 |    9000 |
| Schwinn Paramount P14   |   60 | blue        | 1800 | 1992-03-01 |     200 |
| Bianchi Corse Evo 4     |   58 | celeste     | 5700 | 2004-12-02 |     300 |
| Colnago Superissimo     |   59 | red         | 3800 | 1996-03-01 |   13000 |
+-------------------------+------+-------------+------+------------+---------+
10 rows in set (0.00 sec)

mysql>
```

The tuples within the bikes table are displayed as the result of the query.

Creating a Database in MySQL

- From the MySQL monitor enter `create database <db name>`

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| test               |
| testdb             |
+--------------------+
9 rows in set (0.00 sec)

mysql> create database sample;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| sample             |
| test               |
| testdb             |
+--------------------+
10 rows in set (0.00 sec)

mysql> _
```

Create new database from within MySQL monitor.

Subsequent listing shows newly created database

Dropping a Database in MySQL

- From the MySQL monitor execute the `drop database <db name>` command.

```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| sample             |
| test               |
| testdb             |
+--------------------+
10 rows in set (0.00 sec)

mysql> drop database sample;
Query OK, 0 rows affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| colorsurvey        |
| emps               |
| mailinglist        |
| mysql              |
| prog2              |
| test               |
| testdb             |
+--------------------+
9 rows in set (0.00 sec)

mysql> _
```

From within the MySQL monitor, no warning is given when dropping a database. Be very sure that this is what you want to do before you do it.

Manipulating Tables in MySQL

- The creation of a database does not place any relations into the database. Relations must be separately created.

- To create a table within a database, first select the database (or create one if you haven't already done so), then execute the `create table` command.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> use sample;
Database changed
mysql> create table articles (
    -> article_id int(9) not null auto_increment,
    -> headline text not null,
    -> data_post datetime not null default '0000-00-00 00:00:00',
    -> text_body text,
    -> who_created int(9) default null,
    -> email_sent int(1) not null default '0',
    -> date_email datetime default null,
    -> who_approved int(9) default null,
    -> pic varchar(255) default null,
    -> primary key (article_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
```


Manipulating Tables in MySQL (cont.)

Screen shot that describes the newly created table.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql>
mysql>
mysql>
mysql> describe articles;
+--------------+--------------+------+-----+---------------------+----------------
| Field        | Type         | Null | Key | Default             | Extra
+--------------+--------------+------+-----+---------------------+----------------
| article_id   | int(9)       | NO   | PRI | NULL                | auto_increment
| headline     | text         | NO   |     | NULL                |
| data_post    | datetime     | NO   |     | 0000-00-00 00:00:00 |
| text_body    | text         | YES  |     | NULL                |
| who_created  | int(9)       | YES  |     | NULL                |
| email_sent   | int(1)       | NO   |     | 0                   |
| date_email   | datetime     | YES  |     | NULL                |
| who_approved | int(9)       | YES  |     | NULL                |
| pic          | varchar(255) | YES  |     | NULL                |
+--------------+--------------+------+-----+---------------------+----------------
9 rows in set (0.00 sec)

mysql> _
```


Manipulating Tables in MySQL (cont.)

- The `create table` command has the following general format:

```
create [temporary] table
[if not exists] tablename
[(create_definition, ...)]
[table_options] [select_statement];
```

- If the [if not exists] clause is present, MySQL will produce an error message if a table with the specified name already exists in the database, otherwise the table is created.

Manipulating Tables in MySQL (cont.)

- A temporary table exists only for the life of the current database connection. It is automatically destroyed when the connection is closed or dies.
- Two different connections can use the same name for a temporary table without conflicting with one another.
- Temporary tables are most useful when queries get complex and intermediate results become useful. Also, versions of MySQL earlier than version 4.1 do not have subselect capability and temporary tables are a convenient way to simulate subselect query results.

Note: Non-root users require special permission to be able to create temporary tables. These users must have the Create_tmp_tables privilege set in the user grant table. We'll see more on this later.

Creating A Temporary Table From A Select Query

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> use bikedb;
Database changed
mysql> select * from bikes;
+-------------------------+------+--------------+------+----------
| bikename                | size | color        | cost | purchased
+-------------------------+------+--------------+------+----------
| Colnago Dream Rabobank  |   60 | blue/orange  | 5500 | 2002-07-0
| Bianchi Evolution 3     |   58 | celeste      | 4800 | 2003-11-1
| Eddy Merckx Molteni     |   58 | orange       | 5100 | 2004-08-1
| Eddy Merckx Domo        |   58 | blue/black   | 5300 | 2004-02-0
| Battaglin Carrera       |   60 | red/white    | 4000 | 2001-03-1
| Gianni Motta Personal   |   59 | red/green    | 4400 | 2000-05-0
| Gios Torino Super       |   60 | blue         | 2000 | 1998-11-0
| Schwinn Paramount P14   |   60 | blue         | 1800 | 1992-03-0
| Bianchi Corse Evo 4     |   58 | celeste      | 5700 | 2004-12-0
| Colnago Superissimo     |   59 | red          | 3800 | 1996-03-0
+-------------------------+------+--------------+------+----------
10 rows in set (0.00 sec)

mysql> create temporary table celestebikes
    -> select *
    -> from bikes
    -> where color = "celeste";
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> show tables;
+-------------------+
| Tables_in_bikedb  |
+-------------------+
| bikes             |
| bluebikes         |
+-------------------+
2 rows in set (0.00 sec)

mysql> select * from celestebikes;
+-----------------------+------+---------+------+-------------+---
| bikename              | size | color   | cost | purchased   | mi
+-----------------------+------+---------+------+-------------+---
| Bianchi Evolution 3   |   58 | celeste | 4800 | 2003-11-12  |
| Bianchi Corse Evo 4   |   58 | celeste | 5700 | 2004-12-02  |
+-----------------------+------+---------+------+-------------+---
2 rows in set (0.00 sec)

mysql>
```

A SELECT query produces a result set which has been extracted from one or more tables. A table can be created with the results of this data using the create table command.

Notice that temporary tables do not appear in a table listing.

Manipulating Tables in MySQL (cont.)

- Recall that the `create table` command has the following general format:

```
create [temporary] table
[if not exists] tablename
[(create_definition, ...)]
[table_options]
[select_statement];
```

- The table options allow you to specify the MySQL table type. The table type can be anyone of the six types listed in the table on the next slide.

Manipulating Tables in MySQL (cont.)

Table Type	Description
ISAM	MySQL's original table handler
HEAP	The data for this table is only stored in memory
MyISAM	A binary portable table handler that has replaced ISAM
MERGE	A collection of MyISAM tables used as one table
BDB	Transaction-safe tables with page locking
InnoDB	Transaction-safe tables with row locking

MySQL Table Types

ISAM, HEAP, and MyISAM are available for MySQL versions 3.23.6 or later.

MERGE, BDB, and InnoDB are available for MySQL versions 4.0 and later.

Default table type is InnoDB for MySQL versions 5.1.43.x.

Altering A Table

- After a table has been created, it is possible to change the specifications of its schema. This is done through the `alter table` command:

  `alter table` *`table_name action_list`*

  – Note: Changing the schema of a table in a database is not something that is done very often once the database has been created. The time for altering the schema is during the design phase. Altering the schema of an operational database is a very dangerous thing.

- Multiple changes to the table can be made at the same time by separating actions with commas in the `action_list`.

- The possible attribute (column) actions that can be used are shown in the table on the following slide.

Altering A Table (cont.)

Action Syntax	Action Performed
`add [column] column_declaration [first | after column_name]`	Add a column to the table
`alter [column] column_name {set default literal | drop default}`	Specify new default value for a column or remove old default
`change [column] column_name column_declaration`	Modify column declaration with renaming of column
`modify [column] column_declaration`	Modify column declaration without renaming column
`drop [column] column_name`	Drop a column and all data contained within it.
`rename [as] new_table_name`	Rename a table
table_options	Change the table options

Actions performed by `alter table` (column related) command

column_name represents the current name of the column, *column_declaration* represents the new declaration, in the same format as if it were in a `create` command.

Altering A Table (cont.)

- The screen shot below shows an example of altering a table.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> describe bikes;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| bikename  | varchar(30)  | NO   | PRI | NULL    |       |
| size      | int(2)       | YES  |     | NULL    |       |
| color     | varchar(15)  | YES  |     | NULL    |       |
| cost      | int(6)       | YES  |     | NULL    |       |
| purchased | date         | YES  |     | NULL    |       |
| mileage   | int(6)       | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> alter table bikes
    -> add column races_won int(3) default 0;
Query OK, 10 rows affected (0.05 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> describe bikes;
+-----------+--------------+------+-----+---------+-------
| Field     | Type         | Null | Key | Default | Extra
+-----------+--------------+------+-----+---------+-------
| bikename  | varchar(30)  | NO   | PRI | NULL    |
| size      | int(2)       | YES  |     | NULL    |
| color     | varchar(15)  | YES  |     | NULL    |
| cost      | int(6)       | YES  |     | NULL    |
| purchased | date         | YES  |     | NULL    |
| mileage   | int(6)       | YES  |     | NULL    |
| races_won | int(3)       | YES  |     | 0       |
+-----------+--------------+------+-----+---------+-------
7 rows in set (0.00 sec)

mysql>
```

Schema of bikes before alteration

There are 10 rows affected because this table currently contains 10 tuples (rows) and the new attribute has been added to all rows.

Bikes table after the addition of a new column named races_won

Altering A Table (cont.)

- The screen shot below shows the tuples currently in the bikes table after the addition of the new attribute illustrating that all of the tuples have assumed the default value on the new attribute.

Every tuple in the table has the default value for the new attribute.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from bikes;
+-----------------------+------+-------------+------+------------+---------+-----------+
| bikename              | size | color       | cost | purchased  | mileage | races_won |
+-----------------------+------+-------------+------+------------+---------+-----------+
| Colnago Dream Rabobank |   60 | blue/orange | 5500 | 2002-07-07 |    4300 |         0 |
| Bianchi Evolution 3   |   58 | celeste     | 4800 | 2003-11-12 |    2000 |         0 |
| Eddy Merckx Molteni   |   58 | orange      | 5100 | 2004-08-12 |       0 |         0 |
| Eddy Merckx Domo      |   58 | blue/black  | 5300 | 2004-02-02 |       0 |         0 |
| Battaglin Carrera     |   60 | red/white   | 4000 | 2001-03-10 |   11200 |         0 |
| Gianni Motta Personal |   59 | red/green   | 4400 | 2000-05-01 |    8700 |         0 |
| Gios Torino Super     |   60 | blue        | 2000 | 1998-11-08 |    9000 |         0 |
| Schwinn Paramount P14 |   60 | blue        | 1800 | 1992-03-01 |     200 |         0 |
| Bianchi Corse Evo 4   |   58 | celeste     | 5700 | 2004-12-02 |     300 |         0 |
| Colnago Superissimo   |   59 | red         | 3800 | 1996-03-01 |   13000 |         0 |
+-----------------------+------+-------------+------+------------+---------+-----------+
10 rows in set (0.00 sec)

mysql>
```


Altering A Table (cont.)

- The screen shot below illustrates dropping a column from a table.
- Note that in general, this type of operation may not always be allowed due to constraint violations.

The attribute races_won has been eliminated from the table.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> alter table bikes
    -> drop column races_won;
Query OK, 10 rows affected (0.03 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> describe bikes;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| bikename  | varchar(30) | NO   | PRI | NULL    |       |
| size      | int(2)      | YES  |     | NULL    |       |
| color     | varchar(15) | YES  |     | NULL    |       |
| cost      | int(6)      | YES  |     | NULL    |       |
| purchased | date        | YES  |     | NULL    |       |
| mileage   | int(6)      | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```


Altering A Table (cont.)

- The screen shot below shows a more complicated example of altering a table.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> alter table bikes
    -> add column lastoverhaul datetime after bikename,
    -> modify cost int(8),
    -> add column races_ridden int(3) after mileage;
Query OK, 10 rows affected (0.03 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> describe bikes;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| bikename     | varchar(30) | NO   | PRI | NULL    |       |
| lastoverhaul | datetime    | YES  |     | NULL    |       |
| size         | int(2)      | YES  |     | NULL    |       |
| color        | varchar(15) | YES  |     | NULL    |       |
| cost         | int(8)      | YES  |     | NULL    |       |
| purchased    | date        | YES  |     | NULL    |       |
| mileage      | int(6)      | YES  |     | NULL    |       |
| races_ridden | int(3)      | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql>
```

More complicated alter table command.

Bikes table after the alteration

Inserting Data Into A Table

- Data can be entered into a MySQL table using either the `insert` or `replace` commands.
- The `insert` statement is the primary way of getting data into the database and has the following form:

Form 1

```
insert [low priority | delayed] [ignore] [into]table_name
        [set] column_name1 = expression1,
                column_name2 = expression2, …
```

Form 2

```
insert [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)]values (expression,…), (…)…
```

Form 3

```
insert [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)] select…
```


Inserting Data Into A Table (cont.)

- Form 1 of the insert statement is the most verbose, but also the most common. The `set` clause explicitly names each column and states what value (evaluated from each `expression`) should be put into the table.

- Form 2 (insert values) requires just a comma separated list of the data. For each row inserted, each data value must correspond with a column. In other words, the number of values listed must match the number of columns and the order of the value list must be the same as the columns. (In form 1, the order is not critical since each column is named.)

- Form 3 is used to insert data into a table which is the result set of a `select` statement. This is similar to the temporary table example seen earlier in the notes.

- The following couple of pages give some examples of the different forms of the `insert` command.

Examples: Inserting Data Into A Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql>
mysql> select * from bikes;
+-------------------------+------+--------------+------+-------------+---
| bikename                | size | color        | cost | purchased   | mi
+-------------------------+------+--------------+------+-------------+---
| Colnago Dream Rabobank  |   60 | blue/orange  | 5500 | 2002-07-07  |
| Bianchi Evolution 3     |   58 | celeste      | 4800 | 2003-11-12  |
| Eddy Merckx Molteni     |   58 | orange       | 5100 | 2004-08-12  |
| Eddy Merckx Domo        |   58 | blue/black   | 5300 | 2004-02-02  |
| Battaglin Carrera       |   60 | red/white    | 4000 | 2001-03-10  |
| Gianni Motta Personal   |   59 | red/green    | 4400 | 2000-05-01  |
| Gios Torino Super       |   60 | blue         | 2000 | 1998-11-08  |
| Schwinn Paramount P14   |   60 | blue         | 1800 | 1992-03-01  |
| Bianchi Corse Evo 4     |   58 | celeste      | 5700 | 2004-12-02  |
| Colnago Superissimo     |   59 | red          | 3800 | 1996-03-01  |   13000 |
| Ridley Damocles         |   58 | blue/black   | 7500 | 2008-06-27  |       0 |
+-------------------------+------+--------------+------+-------------+---------+
11 rows in set (0.00 sec)

mysql> insert into bikes
    -> set bikename="Eddy Merckx AX5",
    -> cost=8000,
    -> mileage=150,
    -> purchased="2008-02-02",
    -> color="black/red",
    -> size=58;
Query OK, 1 row affected (0.00 sec)

mysql> select * from bikes;
+-------------------------+------+--------------+------+-------------+---
| bikename                | size | color        | cost | purchased   | mile
+-------------------------+------+--------------+------+-------------+---
| Colnago Dream Rabobank  |   60 | blue/orange  | 5500 | 2002-07-07  |    4300 |
| Bianchi Evolution 3     |   58 | celeste      | 4800 | 2003-11-12  |    2000 |
| Eddy Merckx Molteni     |   58 | orange       | 5100 | 2004-08-12  |       0 |
| Eddy Merckx Domo        |   58 | blue/black   | 5300 | 2004-02-02  |       0 |
| Battaglin Carrera       |   60 | red/white    | 4000 | 2001-03-10  |   11200 |
| Gianni Motta Personal   |   59 | red/green    | 4400 | 2000-05-01  |    8700 |
| Gios Torino Super       |   60 | blue         | 2000 | 1998-11-08  |    9000 |
| Schwinn Paramount P14   |   60 | blue         | 1800 | 1992-03-01  |     200 |
| Bianchi Corse Evo 4     |   58 | celeste      | 5700 | 2004-12-02  |     300 |
| Colnago Superissimo     |   59 | red          | 3800 | 1996-03-01  |   13000 |
| Ridley Damocles         |   58 | blue/black   | 7500 | 2008-06-27  |       0 |
| Eddy Merckx AX5         |   58 | black/red    | 8000 | 2008-02-02  |     150 |
+-------------------------+------+--------------+------+-------------+---------+
12 rows in set (0.00 sec)

mysql> _
```

Using Form 1 for insertion – attribute order is not important.

Examples: Inserting Data Into A Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from bikes;
+-------------------------+------+--------------+------+------------+---------+
| bikename                | size | color        | cost | purchased  | mileage |
+-------------------------+------+--------------+------+------------+---------+
| Colnago Dream Rabobank  |   60 | blue/orange  | 5500 | 2002-07-07 |    4300 |
| Bianchi Evolution 3     |   58 | celeste      | 4800 | 2003-11-12 |    2000 |
| Eddy Merckx Molteni     |   58 | orange       | 5100 | 2004-08-12 |       0 |
| Eddy Merckx Domo        |   58 | blue/black   | 5300 | 2004-02-02 |       0 |
| Battaglin Carrera       |   60 | red/white    | 4000 | 2001-03-10 |   11200 |
| Gianni Motta Personal   |   59 | red/green    | 4400 | 2000-05-01 |    8700 |
| Gios Torino Super       |   60 | blue         | 2000 | 1998-11-08 |    9000 |
| Schwinn Paramount P14   |   60 | blue         | 1800 | 1992-03-01 |     200 |
| Bianchi Corse Evo 4     |   58 | celeste      | 5700 | 2004-12-02 |     300 |
| Colnago Superissimo     |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Ridley Damocles         |   58 | blue/black   | 7500 | 2008-06-27 |       0 |
+-------------------------+------+--------------+------+------------+---------+
11 rows in set (0.00 sec)

mysql> insert into bikes
    -> values ("Eddy Merckx AX5", 58, "black/red", 8000, "2010-02-02", 150);
Query OK, 1 row affected (0.00 sec)

mysql> select * from bikes;
+-------------------------+------+--------------+------+------------+---------+
| bikename                | size | color        | cost | purchased  | mileage |
+-------------------------+------+--------------+------+------------+---------+
| Colnago Dream Rabobank  |   60 | blue/orange  | 5500 | 2002-07-07 |    4300 |
| Bianchi Evolution 3     |   58 | celeste      | 4800 | 2003-11-12 |    2000 |
| Eddy Merckx Molteni     |   58 | orange       | 5100 | 2004-08-12 |       0 |
| Eddy Merckx Domo        |   58 | blue/black   | 5300 | 2004-02-02 |       0 |
| Battaglin Carrera       |   60 | red/white    | 4000 | 2001-03-10 |   11200 |
| Gianni Motta Personal   |   59 | red/green    | 4400 | 2000-05-01 |    8700 |
| Gios Torino Super       |   60 | blue         | 2000 | 1998-11-08 |    9000 |
| Schwinn Paramount P14   |   60 | blue         | 1800 | 1992-03-01 |     200 |
| Bianchi Corse Evo 4     |   58 | celeste      | 5700 | 2004-12-02 |     300 |
| Colnago Superissimo     |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Ridley Damocles         |   58 | blue/black   | 7500 | 2008-06-27 |       0 |
| Eddy Merckx AX5         |   58 | black/red    | 8000 | 2010-02-02 |     150 |
+-------------------------+------+--------------+------+------------+---------+
12 rows in set (0.00 sec)

mysql> _
```

Using Form 2 for insertion – attribute order is important.

Examples: Inserting Data Into A Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> show tables;
+------------------+
| Tables_in_bikedb |
+------------------+
| bikes            |
| bluebikes        |
+------------------+
2 rows in set (0.00 sec)

mysql> create table celestebikes like bikes;
Query OK, 0 rows affected (0.04 sec)

mysql> select * from celestebikes;
Empty set (0.00 sec)

mysql> insert into celestebikes
    -> select *
    -> from bikes
    -> where color = "celeste";
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from celestebikes;
+---------------------+------+---------+------+------------+---------+
| bikename            | size | color   | cost | purchased  | mileage |
+---------------------+------+---------+------+------------+---------+
| Bianchi Evolution 3 |   58 | celeste | 4800 | 2003-11-12 |    2000 |
| Bianchi Corse Evo 4 |   58 | celeste | 5700 | 2004-12-02 |     300 |
+---------------------+------+---------+------+------------+---------+
2 rows in set (0.00 sec)

mysql> _
```

Creates an initially empty table just like the bikes table

Table creation did not place any data into the table

Using Form 3 for insertion

This table contains the name and cost of those bikes whose color was celeste from the source table.

Examples: Inserting Data Into A Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> create table celestebikes (
    -> name varchar(30),
    -> paint varchar(15),
    -> price int(6),
    -> miles_ridden int(6),
    -> primary key (name)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> insert into celestebikes
    -> select bikename, color, cost, mileage
    -> from bikes
    -> where color="celeste";
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from celestebikes;
+---------------------+---------+-------+--------------+
| name                | paint   | price | miles_ridden |
+---------------------+---------+-------+--------------+
| Bianchi Evolution 3 | celeste |  4800 |         2000 |
| Bianchi Corse Evo 4 | celeste |  5700 |          300 |
+---------------------+---------+-------+--------------+
2 rows in set (0.00 sec)

mysql> _
```

Create an initially empty table with a schema different from the base table.

Using Form 3 for insertion

This table contains the those bike tuples whose color was celeste from the source table.

Using Scripts with MySQL

- Entering data to create sample databases using conventional SQL commands is tedious and prone to errors. A much simpler technique is to use scripts. The following illustrates two techniques for invoking scripts in MySQL.
- Create your script file using the text editor of your choice.
- Comments in the SQL script files begin with a # symbol.
- In the script file example shown on the next slide, I drop the database in the first SQL command. Without the if exists clause, this will generate an error if the database does not exist. The first time the script executes (or subsequent executions if the database is dropped independently) the error will be generated…simply ignore the error.

Using Scripts with MySQL (cont.)

*C:\state script.sql - Notepad++

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?

template.html | commentform.html | fourthCSS.css | state script.sql

```sql
#SQL commands in a script file
drop database if exists testdb;

create database testdb;

use testdb;

create table states (
   name varchar(15) not null,
   abbrev char(2),
   capital varchar(25),
   population integer,
   square_miles integer,
   primary key (name)
);

insert into states values ('Florida', 'FL', 'Tallahassee', 18328240, 54153);
insert into states values ('New York', 'NY', 'Albany', 194909297, 54556);
insert into states values ('Indiana', 'IN', 'Indianapolis', 6376792, 35789);
insert into states values ('Maryland', 'MD', 'Annapolis', 5633597, 9975);

select * from states;
```

Drop the database if it already exists.

Create a new database.

Switch to the new database.

Define schema for the new table.

Insert some tuples

Run a simple selection query on the new table.

Structured Query Language file | nb char : 616 | nb line : 22 | ANSI | INS

Using Scripts with MySQL (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> source c:\state script.sql
Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.03 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

+----------+--------+--------------+------------+--------------+
| name     | abbrev | capital      | population | square_miles |
+----------+--------+--------------+------------+--------------+
| Florida  | FL     | Tallahassee  |   18328240 |        54153 |
| New York | NY     | Albany       |  194909297 |        54556 |
| Indiana  | IN     | Indianapolis |    6376792 |        35789 |
| Maryland | MD     | Annapolis    |    5633597 |         9975 |
+----------+--------+--------------+------------+--------------+
4 rows in set (0.00 sec)

mysql>
```

Specify which script to execute

Results of select query at end of script.

Importing Data Using the `mysqlimport` Utility

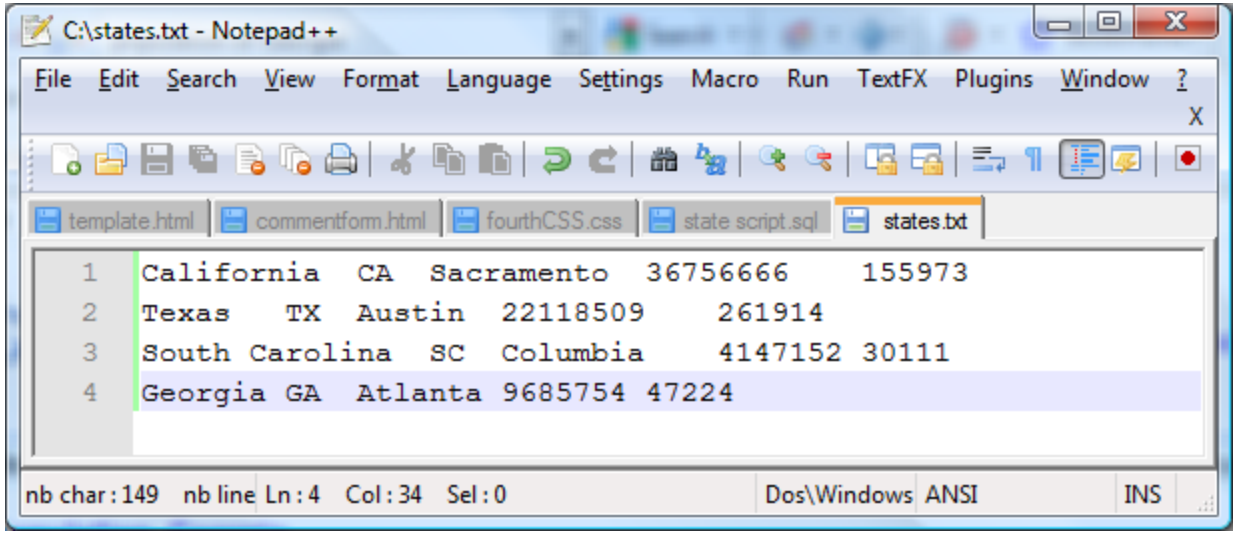
- As with many things in MySQL there are several ways to accomplish a specific task. For getting data into tables, the `mysqlimport` utility is also useful.
- The `mysqlimport` utility reads a range of data formats, including comma- and tab- delimited, and inserts the data into a specified database table. The syntax for `mysqlimport` is:

```
mysqlimport [options] database_name file1 file2 …
```

- This utility is designed to be invoked from the command line.
- The name of the file (excluding the extension) must match the name of the database table into which the data import will occur. Failure to match names will result in an error.

Importing Data Using the `mysqlimportUtility` (cont.)

- The file shown below was created to import additional data into the states table within the testdb database used in the previous example.



- In this case, the default field delimiter (tab), default field enclosure (nothing), and the default line delimiter (\n) were used. Many options are available and are illustrated in the table on pages 65-66.

Importing Data Using the `mysqlimportUtility`

Importing a "data file" into a MySQL database table using the mysqlimport utility

```
C:\Program Files\MySQL\MySQL Server 5.1\bin>mysqlimport -u root -vr testdb c:\st
Connecting to localhost
Selecting database testdb
Loading data from SERVER file: c:/states.txt into states
testdb.states: Records: 4  Deleted: 0  Skipped: 0  Warnings: 3
Disconnecting from localhost

C:\Program Files\MySQL\MySQL Server 5.1\bin>
```

See tables on pages 23-24 for listing of options.

Table updated

Importing Data Using the `mysqlimportUtility`

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
+----------+--------+--------------+------------+--------------+
| name     | abbrev | capital      | population | square_miles |
+----------+--------+--------------+------------+--------------+
| Florida  | FL     | Tallahassee  |   18328240 |        54153 |
| New York | NY     | Albany       |  194909297 |        54556 |
| Indiana  | IN     | Indianapolis |    6376792 |        35789 |
| Maryland | MD     | Annapolis    |    5633597 |         9975 |
+----------+--------+--------------+------------+--------------+
4 rows in set (0.00 sec)

mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name           | abbrev | capital      | population | square_miles |
+----------------+--------+--------------+------------+--------------+
| Florida        | FL     | Tallahassee  |   18328240 |        54153 |
| New York       | NY     | Albany       |  194909297 |        54556 |
| Indiana        | IN     | Indianapolis |    6376792 |        35789 |
| Maryland       | MD     | Annapolis    |    5633597 |         9975 |
| California     | CA     | Sacramento   |   36756666 |       155973 |
| Texas          | TX     | Austin       |   22118509 |       261914 |
| South Carolina | SC     | Columbia     |    4147152 |        30111 |
| Georgia        | GA     | Atlanta      |    9685754 |        47224 |
+----------------+--------+--------------+------------+--------------+
8 rows in set (0.00 sec)

mysql> _
```

Table **before** another client updated the table using the mysqlimport utility.

Table **after** another client updated the table using the mysqlimport utility.

`mysqlimportUtility` Options

Option	Action
-r or –replace	Causes imported rows to overwrite existing rows if they have the same unique key value.
-i or –ignore	Ignores rows that have the same unique key value as existing rows.
-f or –force	Forces mysqlimport to continue inserting data even if errors are encountered.
-l or –lock	Lock each table before importing (a good idea in general and especially on a busy server).
-d or –delete	Empty the table before inserting data.
--fields-terminated-by=‘char’	Specify the separator used between values of the same row, default \t (tab).
--fields-enclosed-by=‘char’	Specify the delimiter that encloses each field, default is none.

mysqlimport Utility Options (cont.)

Option	Action
--fields-optionally-enclosed-by='char'	Same as –fields-enclosed-by, but delimiter is used only to enclosed string-type columns, default is none.
--fields-escaped-by='char'	Specify the escape character placed before special characters; default is \.
--lines-terminated-by='char'	Specify the separator used to terminate each row of data, default is \n (newline).
-u or –user	Specify your username
-p or –password	Specify your password
-h or –host	Import into MySQL on the named host; default is localhost.
-s or –silent	Silent mode, output appears only when errors occur.
-v or –verbose	Verbose mode, print more commentary on action.
-? or –help	Print help message and exit

Importing Data From A File With SQL Statement `Load Data Infile`

- Using the utility `mysqlimport` to load data into a table from an external file works well if the user has access to a command window or command line.
- If you have access via a connection to only the MySQL database, or you are importing data from within an executing application, you will need to use the SQL statement `Load Data Infile`.
- The Load Data Infile statement also provides a bit more flexibility since the file name does not need to match the table name. Other than that the options are basically the same and the same results are accomplished.
- The example on page 70 illustrates this SQL command which is available in MySQL.

Importing Data From A File With SQL Statement `Load Data Infile` (cont.)

- The basic form of the Load Data Infile statement is:

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'filename'
[REPLACE | IGNORE]
INTO TABLE tablename
[FIELDS
     [TERMINATED BY 'char']
     [ [OPTIONALLY] ENCLOSED BY 'char']
     [ESCAPED BY '\char']  ]
[LINES
     [STARTING BY 'char']
     [TERMINATED BY 'char']  ]
[IGNORE number LINES]
[(column_name, … )]
```

Either allow concurrent update or block until no other clients are reading from the specified table. See page 75.

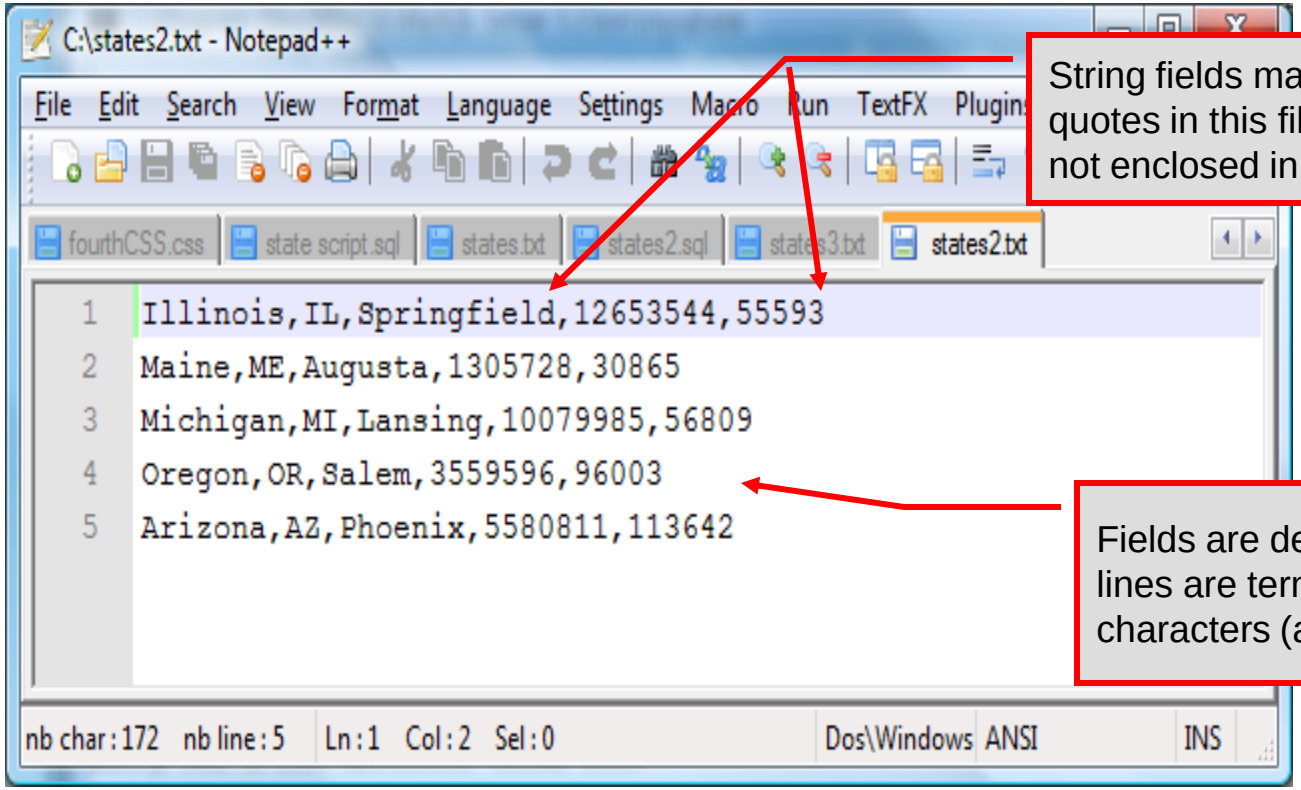
Same as –r and –i options in mysqlimport utility – either replace or ignore rows with duplicate keys.

Sets the characters that delimit and enclose the fields and lines in the data file. Similar to mysqlimport syntax.

Ignores lines at the start of the file (miss header info)

Used to load only certain columns (not entire rows)

Load Data Infile Example



String fields may be enclosed by double quotes in this file. Numeric values are not enclosed in quotes.

Fields are delimited by commas and lines are terminated by newline characters (an invisible \n)

Text file containing the data to be loaded into the database table.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql>
mysql> select * from states;
+----------------+--------+---------------+------------+--------------+
| name           | abbrev | capital       | population | square_miles |
+----------------+--------+---------------+------------+--------------+
| Florida        | FL     | Tallahassee   |   18328240 |        54153 |
| New York       | NY     | Albany        |  194909297 |        54556 |
| Indiana        | IN     | Indianapolis  |    6376792 |        35789 |
| Maryland       | MD     | Annapolis     |    5633597 |         9975 |
| California     | CA     | Sacramento    |   36756666 |       155973 |
| Texas          | TX     | Austin        |   22118509 |       261914 |
| South Carolina | SC     | Columbia      |    4147152 |        30111 |
| Georgia        | GA     | Atlanta       |    9685754 |        47224 |
+----------------+--------+---------------+------------+--------------+
8 rows in set (0.00 sec)

mysql> load data infile 'c:/states2.txt'
    -> into table states
    -> fields
    ->   terminated by ','
    ->   optionally enclosed by '"';
Query OK, 5 rows affected (0.00 sec)
Records: 5  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from states;
+----------------+--------+---------------+------------+--------------+
| name           | abbrev | capital       | population | square_miles |
+----------------+--------+---------------+------------+--------------+
| Florida        | FL     | Tallahassee   |   18328240 |        54153 |
| New York       | NY     | Albany        |  194909297 |        54556 |
| Indiana        | IN     | Indianapolis  |    6376792 |        35789 |
| Maryland       | MD     | Annapolis     |    5633597 |         9975 |
| California     | CA     | Sacramento    |   36756666 |       155973 |
| Texas          | TX     | Austin        |   22118509 |       261914 |
| South Carolina | SC     | Columbia      |    4147152 |        30111 |
| Georgia        | GA     | Atlanta       |    9685754 |        47224 |
| Illinois       | IL     | Springfield   |   12653544 |        55593 |
| Maine          | ME     | Augusta       |    1305728 |        30865 |
| Michigan       | MI     | Lansing       |   10079985 |        56809 |
| Oregon         | OR     | Salem         |    3559596 |        96003 |
| Arizona        | AZ     | Phoenix       |    5580811 |       113642 |
+----------------+--------+---------------+------------+--------------+
13 rows in set (0.00 sec)

mysql> _
```

States table **before** addition of data

Load data infile statement indicating all of the parameters which describe the configuration of the input file.

States table **after** addition of data

Load Data Infile Example 2

C:\states3.txt - Notepad++

```
1 Illinois,IL,Springfield,12653544,55593
2 Maine,ME,Augusta,1305728,30865
3 Michigan,MI,Lansing,10079985,56809
4 Oregon,OR,Salem,3559596,96003
5 Arizona,AZ,Phoenix,5580811,113642
6 California,CA,Los Angeles,36756666,155973
```

Text file containing the data to be loaded into the database table.

California already exists in the states table – this one will replace the value of the capital with a different value.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from states;
+----------------+--------+---------------+------------+--------------+
| name           | abbrev | capital       | population | square_miles |
+----------------+--------+---------------+------------+--------------+
| Florida        | FL     | Tallahassee   |   18328240 |        54153 |
| New York       | NY     | Albany        |  194909297 |        54556 |
| Indiana        | IN     | Indianapolis  |    6376792 |        35789 |
| Maryland       | MD     | Annapolis     |    5633597 |         9975 |
| California     | CA     | Sacramento    |   36756666 |       155973 |
| Texas          | TX     | Austin        |   22118509 |       261914 |
| South Carolina | SC     | Columbia      |    4147152 |        30111 |
| Georgia        | GA     | Atlanta       |    9685754 |        47224 |
| Illinois       | IL     | Springfield   |   12653544 |        55593 |
| Maine          | ME     | Augusta       |    1305728 |        30865 |
| Michigan       | MI     | Lansing       |   10079985 |        56809 |
| Oregon         | OR     | Salem         |    3559596 |        96003 |
| Arizona        | AZ     | Phoenix       |    5580811 |       113642 |
+----------------+--------+---------------+------------+--------------+
13 rows in set (0.00 sec)

mysql> load data infile 'c:/states3.txt'
    -> replace into table states
    -> fields
    ->   terminated by ','
    ->   optionally enclosed by '"';
Query OK, 12 rows affected (0.00 sec)
Records: 6  Deleted: 6  Skipped: 0  Warnings: 0

mysql> select * from states;
+----------------+--------+---------------+------------+--------------+
| name           | abbrev | capital       | population | square_miles |
+----------------+--------+---------------+------------+--------------+
| Florida        | FL     | Tallahassee   |   18328240 |        54153 |
| New York       | NY     | Albany        |  194909297 |        54556 |
| Indiana        | IN     | Indianapolis  |    6376792 |        35789 |
| Maryland       | MD     | Annapolis     |    5633597 |         9975 |
| California     | CA     | Los Angeles   |   36756666 |       155973 |
| Texas          | TX     | Austin        |   22118509 |       261914 |
| South Carolina | SC     | Columbia      |    4147152 |        30111 |
| Georgia        | GA     | Atlanta       |    9685754 |        47224 |
| Illinois       | IL     | Springfield   |   12653544 |        55593 |
| Maine          | ME     | Augusta       |    1305728 |        30865 |
| Michigan       | MI     | Lansing       |   10079985 |        56809 |
| Oregon         | OR     | Salem         |    3559596 |        96003 |
| Arizona        | AZ     | Phoenix       |    5580811 |       113642 |
+----------------+--------+---------------+------------+--------------+
13 rows in set (0.00 sec)

mysql> _
```

States table **before** addition of data

Same basic configuration as in previous example except that we have instructed MySQL to replace duplicate key value rows with new values (in this case replacing California's capital).

States table **after** addition of data. Note that California's capital has been changed!

The `Ignore` Clause of the Insert Command

- While the normal issues of data type compatibility are always of concern, there are other issues to deal with when inserting data into tables.
- There is the possibility that a duplicate of a key may be entered. If so, you will see an error like this:

```
ERROR 1062: Duplicate entry '2' for key 1
```

- It is possible to subdue errors by using the keyword `ignore` in the `insert` statement. By using `ignore` any duplicate rows will simply be ignored. They won't be imported, and the data at the related row of the target table will be left untouched.
  - In your application, you would be wise to check how many rows were affected (imported) whenever using `ignore` because ignoring a record may constitute a failure condition in your application that needs to be handled.

Low Priority and Delayed Inserts

- If you specify `insert low-priority`, the insert waits until all other clients have finished reading from the table before the insert is executed.
- If you specify `insert delayed`, the client performing the action gets and instant acknowledgement that the insert has been performed, although in fact the data will only be inserted when the table is not in use by another thread.
  - This may be useful if you have an application that needs to complete its process in minimum time, or simply where there is no need for it to wait for the effect of an insert to take place. For example, when you're adding data to a log or audit trail.
  - This feature applies only to ISAM or MyISAM type files.

Inserting/Replacing Data Using `Replace`

- Data can also be entered into a MySQL table using the `replace` command.
- The `replace` statement has forms similar to the `insert` statement:

Form 1

```
replace [low priority | delayed] [ignore] [into]table_name
        [set] column_name1 = expression1,
              column_name2 = expression2, …
```

Form 2

```
replace [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)]values (expression,…), (…)…
```

Form 3

```
replace [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)] select…
```


Using `replace`

- The `replace` statement works similar to `insert`. It always tries to insert the new data, but when it tries to insert a new row with the same primary or unique key as an existing row, it deletes the old row and replaces it with the new values.

- The following examples will illustrate how `replace` operates.

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> use bikedb;
Database changed
mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  2000 |        9000 |
| Schwinn Paramount P14 | blue  |  1800 |         200 |
+-----------------------+-------+-------+-------------+
2 rows in set (0.00 sec)

mysql> replace into bluebikes
    -> values ('Gios Torino Super','blue',4200, 11000);
Query OK, 2 rows affected (0.00 sec)

mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4200 |       11000 |
| Schwinn Paramount P14 | blue  |  1800 |         200 |
+-----------------------+-------+-------+-------------+
2 rows in set (0.00 sec)

mysql> _
```

Changing non-key values. Simplest form of data replacement.

Using `Replace` (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4200 |       11000 |
| Schwinn Paramount P14 | blue  |  1800 |         200 |
+-----------------------+-------+-------+-------------+
2 rows in set (0.00 sec)

mysql> replace into bluebikes
    -> values ('Ridley Damocles','blue',8500,1000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4200 |       11000 |
| Schwinn Paramount P14 | blue  |  1800 |         200 |
| Ridley Damocles       | blue  |  8500 |        1000 |
+-----------------------+-------+-------+-------------+
3 rows in set (0.00 sec)

mysql>
```

Specifying values for a non-existent key. Basically the same as an insert since the key value being replaced does not currently exist.

Performing Updates on Tables

- The `update` command allows you to modify the values of the existing data in a table. The basic format of the statement is:

```
update [low priority] [ignore] table_name
    set column_name1 = expression1,
        column_name2 = expression2, …
    [where where_definition]
    [limit num];
```

- There are basically two parts to the statement: the `set` portion to declare which column to set to what value; and the `where` portion, which defines which rows are to be affected.
- `Limit` restricts the number of rows affected to `num`.

Using `update` (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4200 |       11000 |
| Schwinn Paramount P14 | blue  |  1800 |         200 |
| Ridley Damocles       | blue  |  8500 |        1000 |
+-----------------------+-------+-------+-------------+
3 rows in set (0.00 sec)

mysql> update bluebikes
    -> set price=price*1.05;
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4410 |       11000 |
| Schwinn Paramount P14 | blue  |  1890 |         200 |
| Ridley Damocles       | blue  |  8925 |        1000 |
+-----------------------+-------+-------+-------------+
3 rows in set (0.00 sec)
```

Global update within the relation. All tuples have their price field increased by 5%

Using `update` (cont.)

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql>
mysql>
mysql>
mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4410 |       11000 |
| Schwinn Paramount P14 | blue  |  1890 |         200 |
| Ridley Damocles       | blue  |  8925 |        1000 |
+-----------------------+-------+-------+-------------+
3 rows in set (0.00 sec)

mysql> update bluebikes
    -> set price=price*1.05
    -> where price > 4500;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from bluebikes;
+-----------------------+-------+-------+-------------+
| bikename              | color | price | total_miles |
+-----------------------+-------+-------+-------------+
| Gios Torino Super     | blue  |  4410 |       11000 |
| Schwinn Paramount P14 | blue  |  1890 |         200 |
| Ridley Damocles       | blue  |  9371 |        1000 |
+-----------------------+-------+-------+-------------+
3 rows in set (0.00 sec)

mysql> _
```

Specific update, only tuples satisfying the select condition (those with price greater than 4500) will have their price field increased by 5%.

Select Queries in MySQL

- The `select` command in MySQL is basically the same as in the standard SQL, however, it does have some additional features. The basic format of the statement is (not all options are shown – for complete details see the SQL Manual):

```
SELECT [ALL | DISTINCT | DISTINCTROW][HIGH_PRIORITY]
       [STRAIGHT JOIN] [SQL_SMALL_RESULT][SQL_BIG_RESULT]
       [SQL_BUFFER_RESULT][SQ_CACHE | SQL_NO_CACHE]
       select_expression, …
   [INTO {OUTFILE | DUMPFILE} 'path/to/filename' export_options]
   [FROM table_references
        WHERE where_definition]
        [GROUP BY {col_name | col_alias | col_pos | formula}
                [asc |desc], …]
        [HAVING where_definition]
        [ORDER BY {col_name | col_alias | col_pos | formula}
                [asc | desc], …]
        [LIMIT [offset, ] num_rows]
        [PROCEDURE procedure_name];
```


MySQL RDBMS (cont.)

- MySQL features a user permissions system, which allows control over user's access to the databases under MySQL control.

- There are very few competitors of MySQL (Oracle, Sybase, DB2, and SQL Server) that can match the level of sophistication provided by MySQL's permissions system in terms of granularity and level of security provided.

Note that I did not include Microsoft Access in the list above. There are a couple of reasons for this; Access concentrates on the client front-end, although available in shareable versions, it lacks the management system that is a key part of any RDBMS. Access provides virtually no user authentication capabilities nor does it have multithreading processing capabilities, in its normal form.

Authorization in MySQL

- `mysql` and the various utility programs such as `mysqladmin`, `mysqlshow`, and `mysqlimport` can only be invoked by a valid MySQL user.
- Permissions for various users are recorded in grant tables maintained by MySQL.
- As the root user, you have access to all the databases and tables maintained by the MySQL Server.
- One of these databases is named `mysql`.and contains the various information on the users who have access to this installation of MySQL. Some of the tables which comprise this database are shown on the next few pages.

Tables in the `mysql` Database

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bikedb             |
| mysql              |
| sample             |
| test               |
| testdb             |
+--------------------+
6 rows in set (0.00 sec)

mysql> show tables;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| event                     |
| func                      |
| general_log               |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| host                      |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| servers                   |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
| user_info                 |
+---------------------------+
24 rows in set (0.00 sec)

mysql> _
```

The mysql database contains user information

Details on user privileges at the database level. See page 94.

Specific details on privileges at the table level. See page 93

Details on user privileges. See page 91.

Details about the various users. See page 92.

Contents of the `user` Table

outt; - Notepad

File Edit Format View Help

```
mysql> use mysql;
Database changed
mysql> describe user;
+-----------------------+-----------------------------------+------+-----+---------+-------+
| Field                 | Type                              | Null | Key | Default | Extra |
+-----------------------+-----------------------------------+------+-----+---------+-------+
| Host                  | varchar(60)                       |      | PRI |         |       |
| User                  | varchar(16)                       |      | PRI |         |       |
| Password              | varchar(41)                       |      |     |         |       |
| Select_priv           | enum('N','Y')                     |      |     | N       |       |
| Insert_priv           | enum('N','Y')                     |      |     | N       |       |
| Update_priv           | enum('N','Y')                     |      |     | N       |       |
| Delete_priv           | enum('N','Y')                     |      |     | N       |       |
| Create_priv           | enum('N','Y')                     |      |     | N       |       |
| Drop_priv             | enum('N','Y')                     |      |     | N       |       |
| Reload_priv           | enum('N','Y')                     |      |     | N       |       |
| Shutdown_priv         | enum('N','Y')                     |      |     | N       |       |
| Process_priv          | enum('N','Y')                     |      |     | N       |       |
| File_priv             | enum('N','Y')                     |      |     | N       |       |
| Grant_priv            | enum('N','Y')                     |      |     | N       |       |
| References_priv       | enum('N','Y')                     |      |     | N       |       |
| Index_priv            | enum('N','Y')                     |      |     | N       |       |
| Alter_priv            | enum('N','Y')                     |      |     | N       |       |
| Show_db_priv          | enum('N','Y')                     |      |     | N       |       |
| Super_priv            | enum('N','Y')                     |      |     | N       |       |
| Create_tmp_table_priv | enum('N','Y')                     |      |     | N       |       |
| Lock_tables_priv      | enum('N','Y')                     |      |     | N       |       |
| Execute_priv          | enum('N','Y')                     |      |     | N       |       |
| Repl_slave_priv       | enum('N','Y')                     |      |     | N       |       |
| Repl_client_priv      | enum('N','Y')                     |      |     | N       |       |
| ssl_type              | enum('','ANY','X509','SPECIFIED') |      |     |         |       |
| ssl_cipher            | blob                              |      |     |         |       |
| x509_issuer           | blob                              |      |     |         |       |
| x509_subject          | blob                              |      |     |         |       |
| max_questions         | int(11) unsigned                  |      |     | 0       |       |
| max_updates           | int(11) unsigned                  |      |     | 0       |       |
| max_connections       | int(11) unsigned                  |      |     | 0       |       |
+-----------------------+-----------------------------------+------+-----+---------+-------+
31 rows in set (0.00 sec)
```

Contents of the `user_info` Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> describe user_info;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| User               | varchar(16)  | NO   | PRI | NULL    |       |
| Full_name          | varchar(60)  | YES  | MUL | NULL    |       |
| Description        | varchar(255) | YES  |     | NULL    |       |
| Email              | varchar(80)  | YES  |     | NULL    |       |
| Contact_information | text        | YES  |     | NULL    |       |
| Icon               | blob         | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
6 rows in set (0.02 sec)

mysql>
```


Contents of the `tables_priv` Table

outt; - Notepad

File Edit Format View Help

```
mysql> \t;
mysql> describe tables_priv;
+-------------+-----------------------------------------------------------------------------------------
| Field       | Type
+-------------+-----------------------------------------------------------------------------------------
| Host        | char(60)
| Db          | char(64)
| User        | char(16)
| Table_name  | char(64)
| Grantor     | char(77)
| Timestamp   | timestamp
| Table_priv  | set('Select','Insert','Update','Delete','Create','Drop','Grant','References','Index','
| Column_priv | set('Select','Insert','Update','References')
+-------------+-----------------------------------------------------------------------------------------
8 rows in set (0.00 sec)

mysql
```

outt; - Notepad

File Edit Format View Help

```
--------------------------------+------+-----+-------------------+-------+
                                | Null | Key | Default           | Extra |
--------------------------------+------+-----+-------------------+-------+
                                |      | PRI |                   |       |
                                |      | PRI |                   |       |
                                |      | PRI |                   |       |
                                |      | PRI |                   |       |
                                |      | MUL |                   |       |
                                | YES  |     | CURRENT_TIMESTAMP |       |
:','References','Index','Alter') |      |     |                   |       |
--------------------------------+------+-----+-------------------+-------+
```


Contents of the db Table

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe

```
mysql> describe db;
+-----------------------+---------------+------+-----+---------+-------+
| Field                 | Type          | Null | Key | Default | Extra |
+-----------------------+---------------+------+-----+---------+-------+
| Host                  | char(60)      | NO   | PRI |         |       |
| Db                    | char(64)      | NO   | PRI |         |       |
| User                  | char(16)      | NO   | PRI |         |       |
| Select_priv           | enum('N','Y') | NO   |     | N       |       |
| Insert_priv           | enum('N','Y') | NO   |     | N       |       |
| Update_priv           | enum('N','Y') | NO   |     | N       |       |
| Delete_priv           | enum('N','Y') | NO   |     | N       |       |
| Create_priv           | enum('N','Y') | NO   |     | N       |       |
| Drop_priv             | enum('N','Y') | NO   |     | N       |       |
| Grant_priv            | enum('N','Y') | NO   |     | N       |       |
| References_priv       | enum('N','Y') | NO   |     | N       |       |
| Index_priv            | enum('N','Y') | NO   |     | N       |       |
| Alter_priv            | enum('N','Y') | NO   |     | N       |       |
| Create_tmp_table_priv | enum('N','Y') | NO   |     | N       |       |
| Lock_tables_priv      | enum('N','Y') | NO   |     | N       |       |
| Create_view_priv      | enum('N','Y') | NO   |     | N       |       |
| Show_view_priv        | enum('N','Y') | NO   |     | N       |       |
| Create_routine_priv   | enum('N','Y') | NO   |     | N       |       |
| Alter_routine_priv    | enum('N','Y') | NO   |     | N       |       |
| Execute_priv          | enum('N','Y') | NO   |     | N       |       |
| Event_priv            | enum('N','Y') | NO   |     | N       |       |
| Trigger_priv          | enum('N','Y') | NO   |     | N       |       |
+-----------------------+---------------+------+-----+---------+-------+
22 rows in set (0.00 sec)

mysql>
```


How The Grant Tables Work

- The various grant tables work together to define access capabilities for the various users of the databases in MySQL. The tables represent a hierarchy which begins at the database level and moves downward to finer and finer granularity in access capabilities.

- To understand how the grant tables work, it is necessary to understand the process that MySQL goes through when considering a request from a client.

Step 1: A user attempts to connect to the MySQL server. The `user` table is consulted, and on the basis of the username, password, and host from which the connection is occurring, the connection is either refused or accepted. (MySQL actually sorts the user table and looks for the first match.)

How The Grant Tables Work (cont.)

Step 2: If the connection is accepted, any privilege fields in the `user` table that are set to 'Y' will allow the user to perform that action on any database under the server's control. For administrative actions such as shutdown and reload, the entry in the `user` table is deemed absolute, and no further grant tables are consulted.

Step 3: Where the user makes a database-related request and the `user` table does not allow the user to perform that operations (the privilege is set to 'N'), MySQL consults the `db` table (see page 84).

Step 4: The `db` table is consulted to see if there is an entry for the user, database, and host. If there is a match, the `db` privilege fields determine whether the user can perform the request.

How The Grant Tables Work (cont.)

Step 5: If there is a match on the `db` table's `Db` and `User` files but `Host` is blank, the `host` table is consulted to see whether there is a match on all three fields. If there is, the privilege fields in the `host` table will determine whether the use can perform the requested operation. Corresponding entries in the `db` and `host` tables must both be 'Y' for the request to be granted. Thus, an 'N' in either table will block the request.

Step 6: If the user's request is not granted, MySQL checks the `tables_priv` (see page 83) and `columns_priv` tables. It looks for a match on the user, host, database, and table to which the request is made (and the column, if there is an entry in the `columns_priv` table). It adds any privileges it finds in these tables to the privileges already granted. The sum of these privileges determines if the request can be granted.

Managing User Privileges with GRANT and REVOKE

- The basic granting and revocation of privileges in MySQL are accomplished through the `grant` and `revoke` commands.
- The format of the `grant` command is:

```
GRANT privileges [(column_list)]
 ON  database_name.table_name
 TO username@hostname [IDENTIFIED BY 'password']
 [REQUIRE [SSL | X509]
    [CIPHER  cipher [AND] ]
    [ISSUER  issuer [AND] ]
    [SUBJECT subject ] ]
 [WITH GRANT OPTION |
     MAX_QUERIES_PER_HOUR  num  |
     MAX_UPDATES_PER_HOUR  num  |
     MAX_CONNECTIONS_PER_HOUR  num ]
```


Some of the Privileges Assigned with GRANT

Privilege	Operations Permitted
ALL or ALL PRIVILEGES	All privileges except for GRANT
ALTER	Change a table definition using ALTER TABLE excluding the creation and dropping of indices.
CREATE	Create database or tables within a database.
CREATE TEMPORARY TABLES	Create temporary tables.
DELETE	Ability to perform deletions from tables. (Delete DML statements).
DROP	Ability to drop databases or tables.
INSERT	Ability to insert data into tables.
SHUTDOWN	Ability to shutdown the MySQL server.

Displaying Privileges with SHOW

- The SQL command SHOW is used to display the grant privileges for a given user.
- The syntax for the SHOW command is:

```
SHOW GRANTS FOR username@hostname
```

- An example is shown below:

```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
+-----------------------------------------------+
2 rows in set (0.00 sec)

mysql> show grants for mark;
+-----------------------------------------------+
| Grants for mark@%                             |
+-----------------------------------------------+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON *.* TO 'mark'@'%' IDENTIFIED BY PASSWORD
'*E6ACCEDB2495496B191ED488F598F04239C85E73' |
| GRANT ALL PRIVILEGES ON `bikedb`.* TO 'mark'@'%' WITH GRANT OPTION |
+-----------------------------------------------+
2 rows in set (0.00 sec)

mysql>
```

This user has only SELECT, INSERT, UPDATE<, DELETE, and CREATE global privileges.

The user has all privileges on the bikedb database.

Revoking User Privileges with `REVOKE`

- Revocation of privileges in MySQL is accomplished with the `revoke` command.
- The format of the `revoke` command is:

```
REVOKE privileges [(column_list)]
 ON  database_name.table_name
 FROM username@hostname
```

- An example is shown on the next page.

Example - Revoking User Privileges with REVOKE

```
C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
+-----------------------------------------------------------------------------------------------+
| Grants for mark@%                                                                             |
+-----------------------------------------------------------------------------------------------+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON *.* TO 'mark'@'%' IDENTIFIED BY PASSWORD
'*E6ACCEDB2495496B191ED488F598F04239C85E73' |
| GRANT SELECT ON `testdb`.* TO 'mark'@'%'                                                      |
| GRANT ALL PRIVILEGES ON `bikedb`.* TO 'mark'@'%' WITH GRANT OPTION                            |
| GRANT SELECT ON `testdb`.`states` TO 'mark'@'%'                                               |
+-----------------------------------------------------------------------------------------------+
4 rows in set (0.00 sec)

mysql> revoke select
    -> on testdb.states
    -> from mark;
Query OK, 0 rows affected (0.00 sec)

mysql> show grants for mark;
+-----------------------------------------------------------------------------------------------+
| Grants for mark@%                                                                             |
+-----------------------------------------------------------------------------------------------+
| GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON *.* TO 'mark'@'%' IDENTIFIED BY PASSWORD
'*E6ACCEDB2495496B191ED488F598F04239C85E73' |
| GRANT SELECT ON `testdb`.* TO 'mark'@'%'                                                      |
| GRANT ALL PRIVILEGES ON `bikedb`.* TO 'mark'@'%' WITH GRANT OPTION                            |
+-----------------------------------------------------------------------------------------------+
3 rows in set (0.00 sec)

mysql> _
```

User has SELECT privilege on testdb.states table.

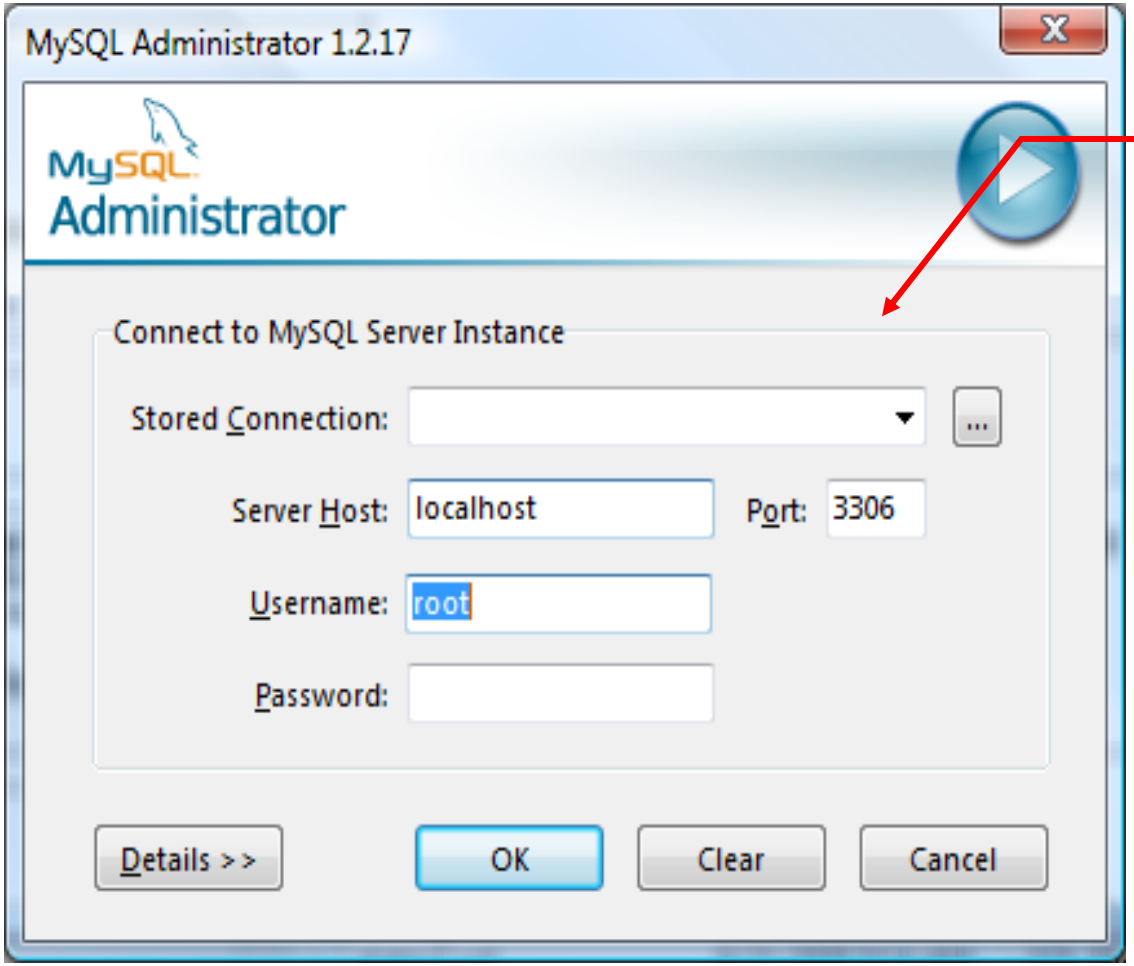
Revoking user's SELECT privilege on testdb.states.

User's grant listing shows that they no longer have SELECT privilege on testdb.states table.

The MySQL Administrator Tool

- From MySQL you can download a GUI-based administrator tool to help you administer your MySQL databases.
- This tool implements all of the GRANT, REVOKE, and SHOW functionality available in SQL.
- This tool also contains some system administrator functionality for monitoring system resources and utilization.
- You can download this tool at: http://www.mysql.com/products/.
- A few screen shots of this tool and its capabilities are shown in the next few slides.

The MySQL Administrator Tool – Screen Shots



Initial login screen

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

- Server Information
- Service Control
- Startup Variables
- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore
- Catalogs

Server status:
MySQL Server is running.

Connected to MySQL Server Instance

Username:	root
Hostname:	localhost
Port:	3306

Server Information

MySQL Version:	MySQL 5.1.43-community via TCP/IP
Network Name:	9VQ2JH1-PC
IP:	127.0.0.1

Client Information

Version:	MySQL Client Version 5.1.11
Network Name:	9VQ2JH1-PC
IP:	10.173.203.112
Operating System:	unknown
Hardware:	4x Intel(R) Core(TM)2 Quad CPU Q6700 @ 2.66GHz, 3.2 GB RAM

Initial screen after successful login.

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

View of user information screen.

- Server Information
- Service Control
- Startup Variables
- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore
- Catalogs

Users Accounts

- mark
- root

User Information | Global Privileges | Schema Privileges | Schema Obj

mark@any host (Mark Llewellyn)
Login and additional information on the user

Login Information

MySQL User:	mark	The user has to enter this MySQL User name to connect to the MySQL Server
Password:	********	Fill out this field if you want to set the user's password
Confirm Password:	********	Again, enter the user's password to confirm

Additional Information

Full Name:	Mark Llewellyn	The user's full name
Description:	Tall, dark and handsome	Additional description of the user
Email:	markl@cs.ucf.edu	The user's email address
Contact Information:		Optional contact information
Icon:	Load from disk / Clear Image	Icon assigned to the user

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

- Server Information
- Service Control
- Startup Variables
- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore
- Catalogs

Users Accounts

- mark
- root

User Information | Global Privileges | Schema Privileges | Schema Object Pr

View of user privileges for mark user on bikedb

mark@any host (Mark Llewellyn)
Schema Privileges assigned to the User

Schemata	Assigned Privileges	Available Privileges	
bikedb		SELECT	Grants the SELECT privilege to t...
colorsurvey		INSERT	Grants the INSERT privilege to t...
emps		UPDATE	Grants the UPDATE privilege to ...
information_schema		DELETE	Grants the DELETE privilege to t...
mailinglist		CREATE	Grants the CREATE privilege to t...
mysql		DROP	Grants the DROP privilege to th...
prog2		GRANT	Grants the GRANT privilege to t...
sample		REFERENCES	Grants the REFERENCES privileg...
test		INDEX	Grants the INDEX privilege to th...
testdb		ALTER	Grants the ALTER privilege to th...
		CREATE_TMP_TA...	Grants the CREATE_TMP_TABLE ...
		LOCK_TABLES	Grants the LOCK_TABLES privile...
		CREATE_VIEW	Grants the CREATE_VIEW privile...
		SHOW_VIEW	Grants the SHOW_VIEW privileg...
		CREATE_ROUTINE	Grants the CREATE_ROUTINE pri...
		ALTER_ROUTINE	Grants the ALTER_ROUTINE privi...
		EXECUTE	Grants the EXECUTE privilege to...

Apply changes | Discard changes

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

Server Information
Service Control
Startup Variables
User Administration
Server Connections
Health
Server Logs
Replication Status
Backup
Restore
Catalogs

Users Accounts

mark
root
localhost
127.0.0.1

User Information | Global Privileges | Schema Privileges | Schema Object Pr

root@localhost (Mark Llewellyn)
Schema Privileges assigned to the User

Schemata

bikedb
colorsurvey
emps
information_schema
mailinglist
mysql
prog2
sample
test
testdb

Assigned Privileges

SELECT
INSERT
UPDATE
DELETE
CREATE
DROP
GRANT
REFERENCES
INDEX
ALTER
CREATE_TMP...
LOCK_TABLES
CREATE_VIEW
SHOW_VIEW
CREATE_RO...
ALTER_ROUT...
EXECUTE

Availa

View of user privileges for root user on bikedb. Root user has all privileges by default.

Apply changes | Discard changes

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

- Server Information
- Service Control
- Startup Variables
- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore
- Catalogs

Users Accounts

- mark
- root
  - localhost
  - 127.0.0.1

User Information | Global Privileges | Schema Privileges | Schema Object Privileges | Resources

mark@any host (Mark Llewellyn)
Schema Privileges assigned to the User

Schemata

- bikedb
- colorsurvey
- emps
- information_schema
- mailinglist
- mysql
- prog2
- sample
- test
- testdb

Assigned Privileges

- SELECT
- INSERT
- UPDATE
- DELETE

< > << >>

Available Privileges

CREATE	Grants the CREATE privilege to t...
DROP	Grants the DROP privilege to th...
GRANT	Grants the GRANT privilege to t...
REFERENCES	Grants the REFERENCES privileg...
INDEX	Grants the INDEX privilege to th...
ALTER	Grants the ALTER privilege to th...
CREATE_TMP_TA...	Grants the CREATE_TMP_TABLE ...
LOCK_TABLES	Grants the LOCK_TABLES privile...
CREATE_VIEW	Grants the CREATE_VIEW privile...
SHOW_VIEW	Grants the SHOW_VIEW privileg...
CREATE_ROUTINE	Grants the CREATE_ROUTINE pri...
ALTER_ROUTINE	Grants the ALTER_ROUTINE privi...
EXECUTE	Grants the EXECUTE privilege to...

Select a user and a database to grant or revoke privileges.

Apply changes | Discard changes

MySQL Administrator - Connection: root@localhost:3306

File Edit View Tools Window Help

- Server Information
- Service Control
- Startup Variables
- User Administration
- Server Connections
- Health
- Server Logs
- Replication Status
- Backup
- Restore
- Catalogs

Schemata

- bikedb
- colorsurvey
- emps
- information_schema
- mailinglist
- mysql
- prog2
- sample
- test
- testdb

Schema Tables | Schema Indices | Views | Stored procedures

bikedb
All tables of the bikedb schema

Table Name	Engine	Rows	Data length	Index length	Update time
bikes	MyISAM	12	568 B	2 kB	2010-02-15 15:17:42
bluebikes	MyISAM	2	76 B	2 kB	2010-02-15 15:16:55
celestebikes	MyISAM	2	96 B	2 kB	2010-02-15 15:20:41

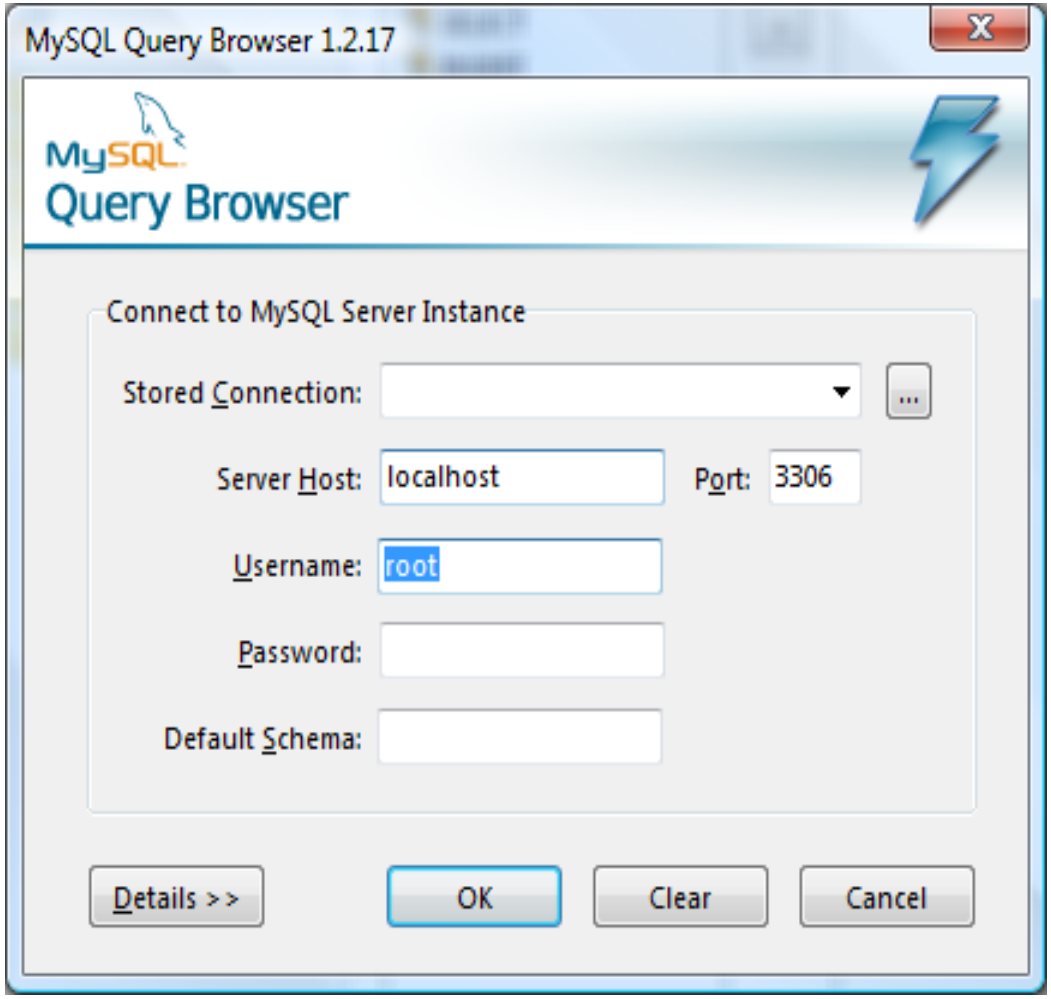
Num. of Tables: 3 | Rows: 16 | Data Len: 740 B | Index Len: 6 kB

View of system catalogs which describe the databases maintained by the server.

The MySQL Query Browser Tool

- From MySQL you can also download a GUI-based query browser tool.
- This tool implements all of the basic DML side of SQL with some limitation. For example, editing result sets is possible only if the result set was generated from a single table. Join-based result sets are not editable. This tool also implements many DDL commands.
- This tool is helpful for developing and testing queries.
- A few screen shots of this tool and its capabilities are shown in the next few slides.

The MySQL Query Browser Tool



A First Look At The MySQL Query Browser

MySQL Query Browser - Connection: root@localhost:3306

File Edit View Query Script Tools Window

Query input window

Go back Next Refresh Execute Stop

Resultset 1

Schemata Bookmarks History

bikedb
information_schema
mysql
sample
test
testdb

Database selection window

Syntax Functions Params Trx

Data Definition Statements
Data Manipulation Statements
MySQL Utility Statements
MySQL Transactional and Locking ...
Database Administration Statements
Replication Statements
SQL Syntax for Prepared Statements

Result set window

Edit Apply Changes Discard Changes First Last Search

A First Look At The MySQL Query Browser

MySQL Query Browser - Connection: root@localhost:3306 / bikedb

File Edit View Query Script Tools Window Help

Go back Next Refresh

```
select * from bikes
```

Execute Stop

Resultset 1

bikename	size	color	cost	purchased	mileage
Colnago Dream Rabobank	60	blue/orange	5500	2002-07-07	4300
Bianchi Evolution 3	58	celeste	4800	2003-11-12	2000
Eddy Merckx Molteni	58	orange	5100	2004-08-12	0
Eddy Merckx Domo	58	blue/black	5300	2004-02-02	0
Battaglin Carrera	60	red/white	4000	2001-03-10	11200
Gianni Motta Personal	59	red/green	4400	2000-05-01	8700
Gios Torino Super	60	blue	2000	1998-11-08	9000
Schwinn Paramount P14	60	blue	1800	1992-03-01	200
Bianchi Corse Evo 4	58	celeste	5700	2004-12-02	300
Colnago Superissimo	59	red	3800	1996-03-01	13000
Ridley Damocles	58	blue/black	7500	2008-06-27	0
Eddy Merckx MXM	58	black/red	8000	2008-02-02	150

12 rows fetched in 0.0038s (0.0 | Edit | Apply Changes | Discard Changes | First | Last | Search

1: 1

Schemata Bookmarks History

- bikedb
- information_schema
- mysql
- sample
- test
- testdb

Syntax Functions Params Trx

- Data Definition Statements
- Data Manipulation Statements

Query input window

Result set window

Result set shown for this query. Note that this query is based on a single table, so the result set is editable.

MySQL Query Browser - Connection: root@localhost:3306 / bikedb

File Edit View Query Script Tools Window Help

Go back Next Refresh

```
select * from bikes where cost >4500
```

Execute Stop

Resultset 1 Resultset 2 **Resultset 3**

bikename	size	color	cost	purchased	mileage
Colnago Dream Rabo...	60	blue/or...	5500	2002-07-07	4300
Bianchi Evolution 3	58	celeste	4800	2003-11-12	2000
Eddy Merckx Molteni	58	orange	5100	2004-08-12	0
Eddy Merckx Domo	58	blue/bla...	5300	2004-02-02	0
Bianchi Corse Evo 4	58	celeste	5700	2004-12-02	300
Ridley Damocles	58	blue/bla...	7500	2008-06-27	0
Eddy Merckx MXM	58	black/red	8000	2008-02-02	150

7 rows fetched in 0.0027s (0.00 | Edit | Apply Changes | Discard Changes | First | Last | Search

1: 1

Schemata Bookmarks History

- bikedb
- information_schema
- mysql
- sample
- test
- testdb

Syntax Functions Params Trx

- Data Definition Statements
- Data Manipulation Statements
- MySQL Utility Statements
- MySQL Transactional and Locking ...
- Database Administration Statements
- Replication Statements
- SQL Syntax for Prepared Statements

You can manage multiple result sets simultaneously. Statistics on query execution are always available.