

# COP 4710: Database Systems Fall 2009

## Introduction To MySQL

Instructor : Dr. Mark Llewellyn  
markl@cs.ucf.edu  
HEC 236, 407-823-2790  
<http://www.cs.ucf.edu/courses/cop4710/fall2009>

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 5.1.38 and can be downloaded from [www.mysql.com](http://www.mysql.com). (Any of the versions of MySQL 5.1.38 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 Bookmarks Sign In

MySQL :: The world's ... [Valid] Markup Validation ...

MySQL.com Developer Zone Partners & Solutions Customer Login

Products Downloads Documentation Services Customers Why MySQL? News & Events How to Buy

GET STARTED

- MySQL Enterprise
- Free Web Seminars
- Test Drive MySQL Query Analyzer
- MySQL Enterprise 30-day Trial
- MySQL Training
- Buy Now
- Contact Us

Sage Group narrows costs and opens new market with MySQL as cross-product line standard

Click here to go to download page

PRODUCTS MYSQL IN ACTION RESOURCES WHAT'S NEW

Internet | Protected Mode: Off 100%



MySQL :: MySQL Downloads - Windows Internet Explorer

http://dev.mysql.com/downloads/

File Edit View Favorites Tools Help

Google Search

MySQL :: MySQL Down... [Valid] Markup Validation ...

MySQL Community Server

MySQL Proxy

MySQL Cluster

MySQL Workbench

GUI Tools

Connectors

Snapshots

Archives

Mirrors

Recommended Servers for MySQL


The world's most popular open source database

MySQL.com Developer Zone Partners & Solutions Customer Login

DevZone • Downloads • Documentation • Librarian • Articles • Forums • Bugs • Forge • Planet MySQL • Labs

Search

## MySQL Downloads



### MySQL Downloads

- ▶ MySQL software is provided under the
- ▶ OEMs, ISVs and VARs can purchase Cor
- ▶ Learn about [MySQL Products](#) and [MySQL](#)

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 5.1 GA - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search

MySQL :: MySQL 5.1 GA

Recommended Servers for MySQL

The world's most popular open source database

Login | Register

MySQL.com Developer Zone Partners & Solutions Customer Login

DevZone • Downloads • Documentation • Librarian • Articles • Forums • Bugs • Forge • Planet MySQL • Labs

Search

MySQL Community Server

5.4

5.1

5.0

4.1

MySQL Proxy

MySQL Cluster

MySQL Workbench

MySQL 5.1 GA

MySQL Downloads

- MySQL software is provided under the
- OEMs, ISVs and VARs can purchase Cor
- Learn about [MySQL Products](#) and [MySQL](#)

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

Internet | Protected Mode: Off

100%

The MySQL Community Server page.





MySQL :: MySQL 5.1 GA - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search

MySQL :: MySQL 5.1 GA

MySQL Workbench  
GUI Tools  
Connectors  
Snapshots  
Archives  
Mirrors

If You Like Free Software,  
You'll Love Our 60-Day  
Free Trial Offer on Systems

Learn More »

Download software  
from premier

	MySQL Community Server	MySQL Enterprise Subscription
	<a href="#">Download</a>	<a href="#">Learn More</a> <a href="#">30-day Trial</a>
List Price (US)	No cost, freely available under the GPL License	Starting at \$599/server/year
Special Offers	Not Applicable	<ul style="list-style-type: none"><li><a href="#">Unlimited Servers for Less</a></li></ul>
Audience	For users or organizations looking to maintain their own solutions.	For <b>businesses, public sector institutions</b> and users looking for the highest reliability.
System Updates	I have my own method of keeping my systems up to date and am comfortable configuring	I want <b>automated notifications</b> and predictable releases of well-tested updates and upgrades

Internet | Protected Mode: Off

100%

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



MySQL :: MySQL 5.1 GA - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search

MySQL :: MySQL 5.1 GA

View the MySQL 5.1 List of Changes

! We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

- [Windows](#)
- [Windows x64](#)
- [Linux \(non RPM packages\)](#)
- [Linux \(non RPM, Intel C/C++ compiled, glibc-2.3\)](#)
- [Red Hat Enterprise Linux 3 RPM \(x86\)](#)
- [Red Hat Enterprise Linux 3 RPM \(AMD64 / Intel EM64T\)](#)
- [Red Hat Enterprise Linux 3 RPM \(Intel IA64\)](#)
- [Red Hat Enterprise Linux 4 RPM \(x86\)](#)
- [Red Hat Enterprise Linux 4 RPM \(AMD64 / Intel EM64T\)](#)
- [Red Hat Enterprise Linux 4 RPM \(Intel IA64\)](#)
- [Red Hat Enterprise Linux 5 RPM \(x86\)](#)
- [Red Hat Enterprise Linux 5 RPM \(AMD64 / Intel EM64T\)](#)
- [Red Hat Enterprise Linux 5 RPM \(Intel IA64\)](#)

Internet | Protected Mode: Off 100%

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



MySQL :: MySQL 5.1 GA - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search

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	<a href="#">Pick a mirror</a>
MD5: 5a077abefee447cbb271e2aa7f6d5a47   <a href="#">Signature</a>			
Windows MSI Installer (x86)	5.1.38	103.5M	<a href="#">Pick a mirror</a>
MD5: 4011f11770fd4cbe2f96ed7b61b5d3ba   <a href="#">Signature</a>			
Without installer (unzip in C:\)	5.1.38	126.9M	<a href="#">Pick a mirror</a>
MD5: 78328ae74e464b0f333e6b0825c6fd6b   <a href="#">Signature</a>			

### Windows x64 downloads (platform notes)

Windows Essentials (AMD64 / Intel EM64T)	5.1.38	29.4M	<a href="#">Pick a mirror</a>
MD5: 40b3eb7055f296c88f66262602d1645f   <a href="#">Signature</a>			
Windows MSI Installer (AMD64 / Intel EM64T)	5.1.38	93.5M	<a href="#">Pick a mirror</a>
MD5: 27ba306cde7df2711450cea983123084   <a href="#">Signature</a>			

Internet | Protected Mode: Off 100%





MySQL :: MySQL GUI Tools - Windows Internet Explorer

http://dev.mysql.com/downloads/gui-tools/5.0.html

File Edit View Favorites Tools Help

Google Search Bookmarks Sign In

MySQL.com Developer Zone Partners

DevZone Downloads Documentation Librarian Articles For

MySQL GUI Tools

MySQL Downloads

- MySQL software is provided under the G
- OEMs, ISVs and VARs can purchase Compr
- Learn about MySQL Products and MySQL

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

MySQL Community Server

MySQL Proxy

MySQL Cluster

MySQL Workbench

GUI Tools

5.0

Connectors

Snapshots

Archives

Internet | Protected Mode: Off

100%



MySQL :: MySQL GUI Tools - Windows Internet Explorer


http://dev.mysql.com/downloads/gui-tools/5.0.html

File Edit View Favorites Tools Help

Google Search Bookmarks Sign In

MySQL Cluster  
MySQL Workbench  
**GUI Tools**  
5.0  
Connectors  
Snapshots  
Archives  
Mirrors

Get Certified  
MySQL certification



The MySQL GUI Tools Bundle for 5.0 includes the following products which are also supported for MySQL 5.1.

- MySQL Administrator 1.2
- MySQL Query Browser 1.2
- MySQL Migration Toolkit 1.1

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

Internet | Protected Mode: Off 100%



MySQL :: MySQL Connectors - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search Bookmarks Sign In

MySQL Proxy  
MySQL Cluster  
MySQL Workbench  
GUI Tools  
**Connectors**  
Connector/ODBC  
Connector/J  
Connector/Net  
Connector/MXJ  
Connector/C++  
Connector/C  
MySQL native driver for PHP  
MySQL Connector for OpenOffice.org

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



- ▶ MySQL software is provided under the G
- ▶ OEMs, ISVs and VARs can purchase Compr
- ▶ Learn about [MySQL Products](#) and [MySQL](#)

MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. This enables MySQL to work easily with standard development tools on Windows, Linux, Macintosh and Unix platforms. Any system that works with ODBC or JDBC can use MySQL.

- [Connector/ODBC](#): Standardized database driver Windows, Linux, Mac OS X, and Unix platforms.

**NEW! MySQL Connector/C++**

- ▶ Free On-Demand Web Seminar
- "The New MySQL Connector, by

Internet | Protected Mode: Off 100%



MySQL :: MySQL Connectors - Windows Internet Explorer

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

File Edit View Favorites Tools Help

Google Search Bookmarks Sign In

MySQL native driver for PHP  
MySQL Connector for OpenOffice.org

Snapshots  
Archives  
Mirrors

ODBC, JDBC, & XQuery Access to MySQL  
DataDirect Technologies Supports Access to MySQL  
→ Evaluate It Now

Related pages:

- Newsletter
- Training

Windows, Linux, Macintosh and Unix that works with ODBC or JDBC can use

- Connector/ODBC: Standardized database driver for 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.
- Connector/C++: Standardized database driver for C++ development.
- Connector/C (libmysql): A client library for C development.
- MySQL native driver for PHP - mysqlnd: The MySQL native driver for PHP is an additional, alternative way to connect from PHP 5.3 or newer to the MySQL Server 4.1 or newer.
- MySQL Connector for OpenOffice.org: Easy to use MySQL driver for OpenOffice.org 3.1 or newer.

Download the Connector/J for use with Java applications. You can go ahead now and also download connector for PHP for use later in the semester.

Internet | Protected Mode: Off 100%



# Installing MySQL 5.1.38

- 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.





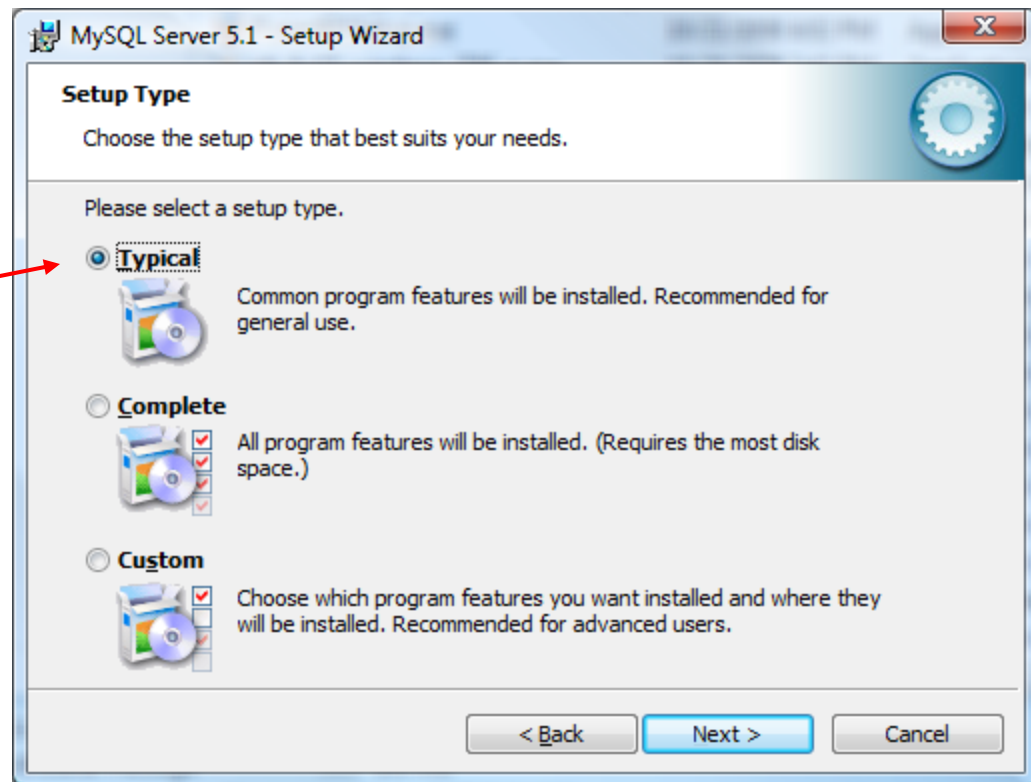
# Installing MySQL 5.1.38

- 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 Windows installer is running you should see the following window appear:

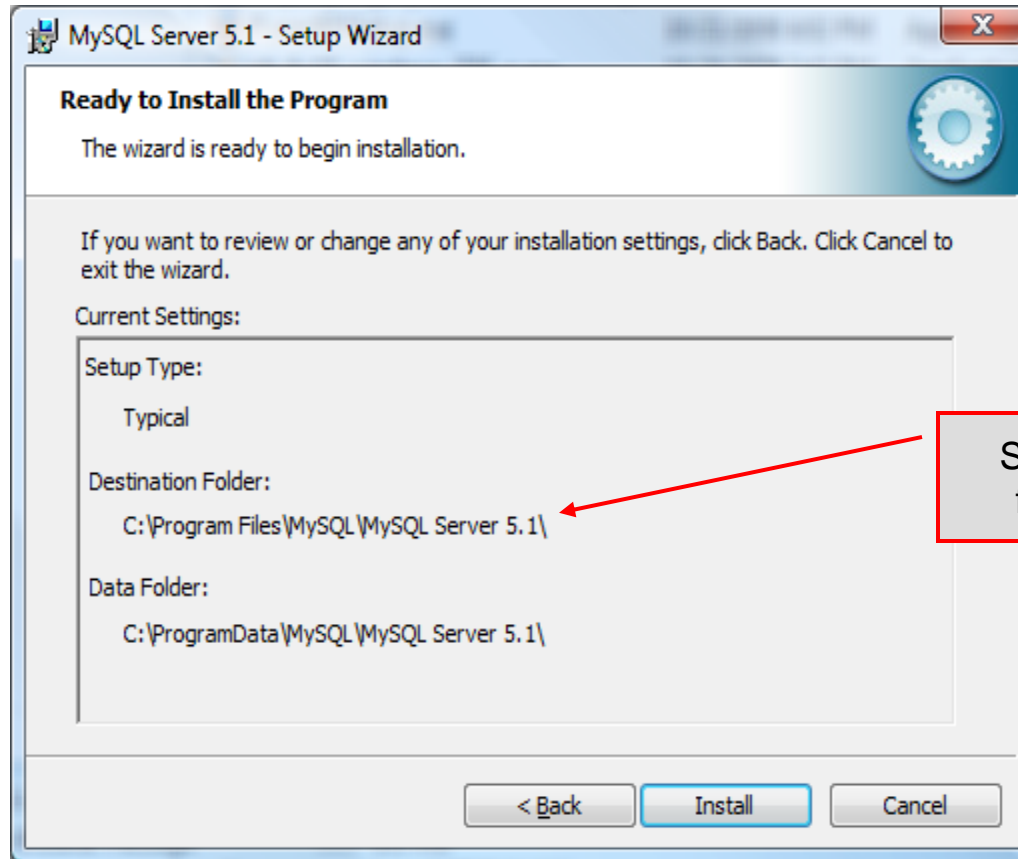


# Installing MySQL 5.1.38 (cont.)

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



# Installing MySQL 5.1.38 (cont.)



Select the destination folder for the install.



# Installing MySQL 5.1.38 (cont.)

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



The image shows a Windows-style dialog box titled "MySQL.com Sign Up - Setup Wizard". The main heading is "MySQL.com Sign-Up" with a subtext "Login or create a new MySQL.com account." Below this, it says "Please log in or select the option to create a new account." There are three radio button options: "Create a new free MySQL.com account" (which is selected), "Login to MySQL.com", and "Skip Sign-Up". The "Login to MySQL.com" option has a subtext "Select this option if you already have a MySQL.com account. Please specify your login information below." followed by input fields for "Email address:" and "Password:". At the bottom right, there are "Next >" and "Cancel" buttons. A red arrow points from the text box on the left to the "Skip Sign-Up" radio button.

**MySQL.com Sign Up - Setup Wizard**

**MySQL.com Sign-Up**  
Login or create a new MySQL.com account.

Please log in or select the option to create a new account.

☒ **Create a new free MySQL.com account**  
If you do not yet have a MySQL.com account, select this option and complete the following three steps.

☐ **Login to MySQL.com**  
Select this option if you already have a MySQL.com account. Please specify your login information below.

Email address:

Password:

☐ **Skip Sign-Up**

Next > Cancel



# Installing MySQL 5.1.38 (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 5.1.38 (cont.)

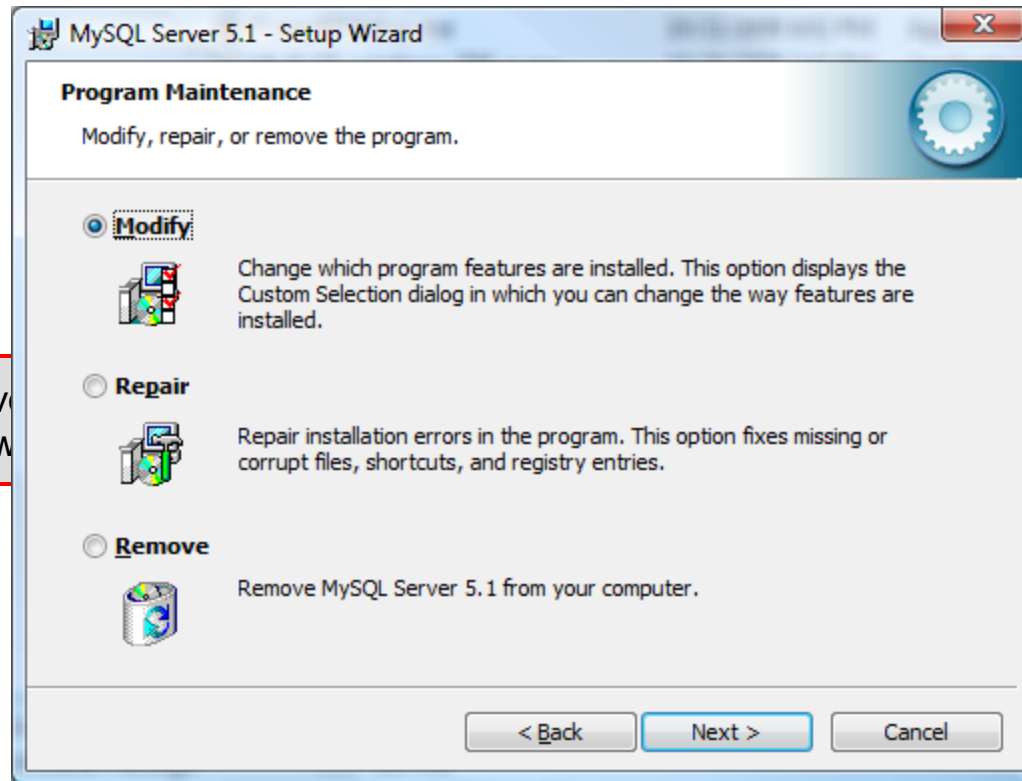


Initial server  
configuration w

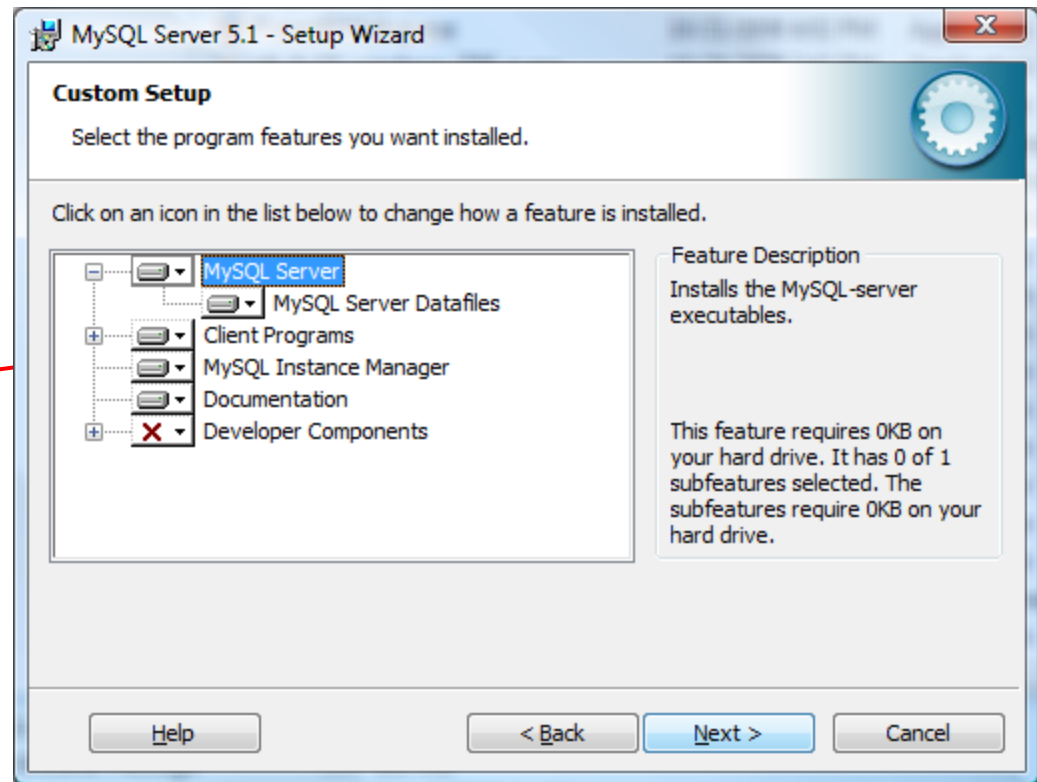


# Installing MySQL 5.1.38 (cont.)

Initial server  
configuration w



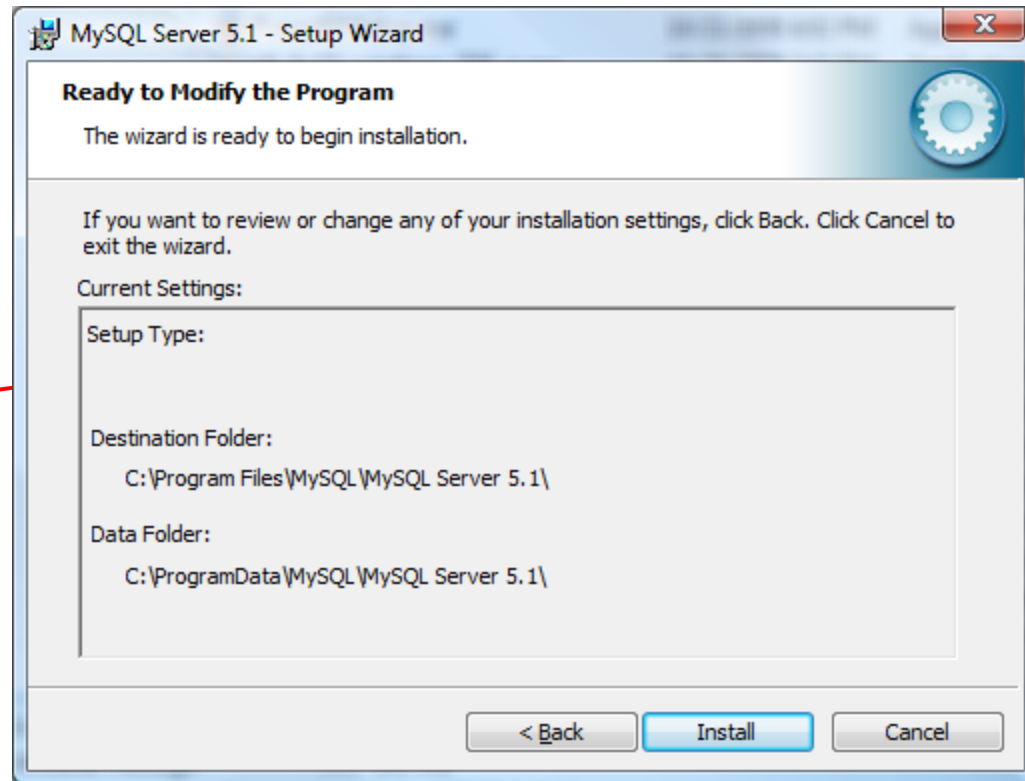
# Installing MySQL 5.1.38 (cont.)



Initial server  
configuration window



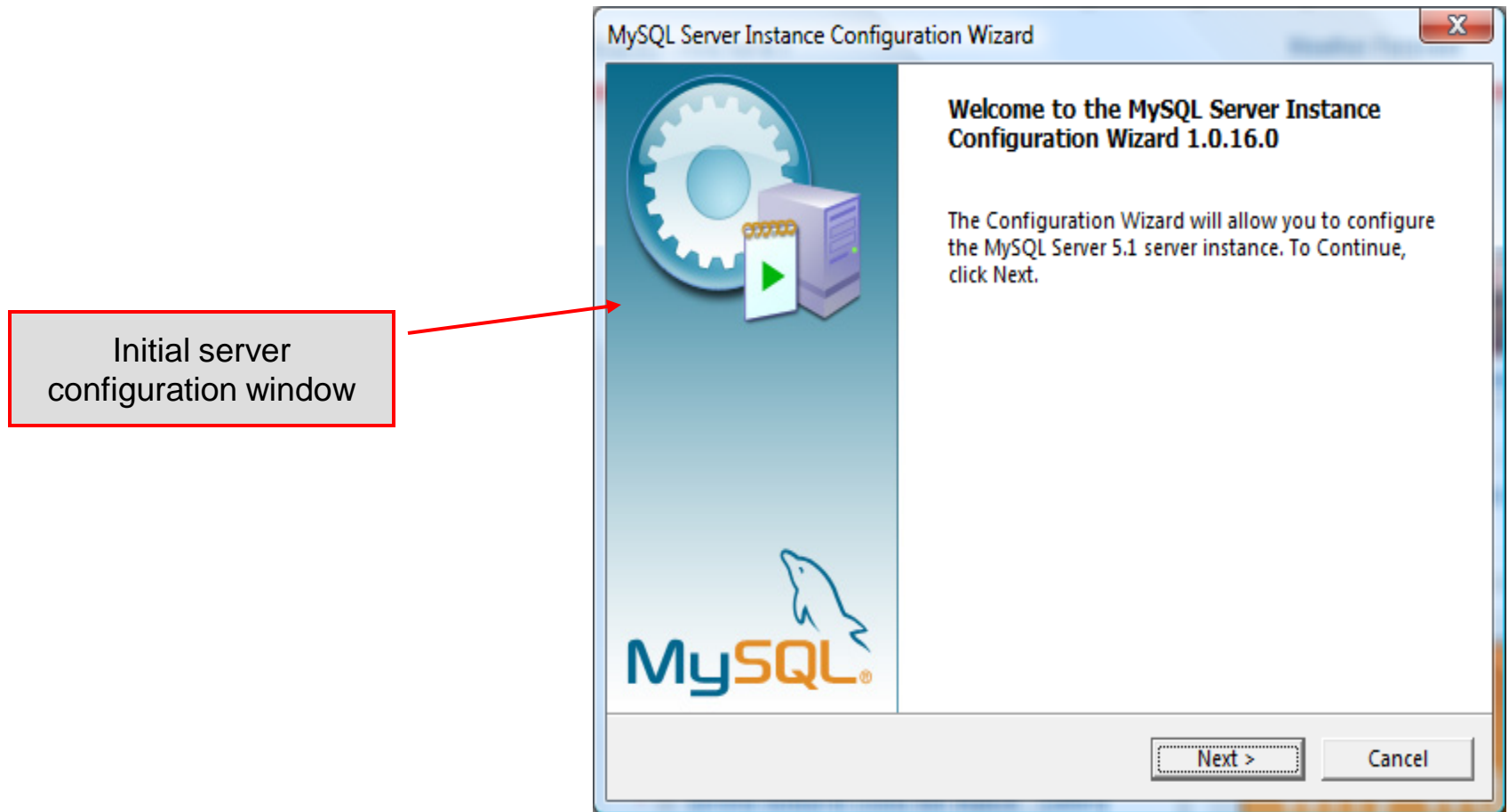
# Installing MySQL 5.1.38 (cont.)



Initial server  
configuration window



# Installing MySQL 5.1.38 (cont.)

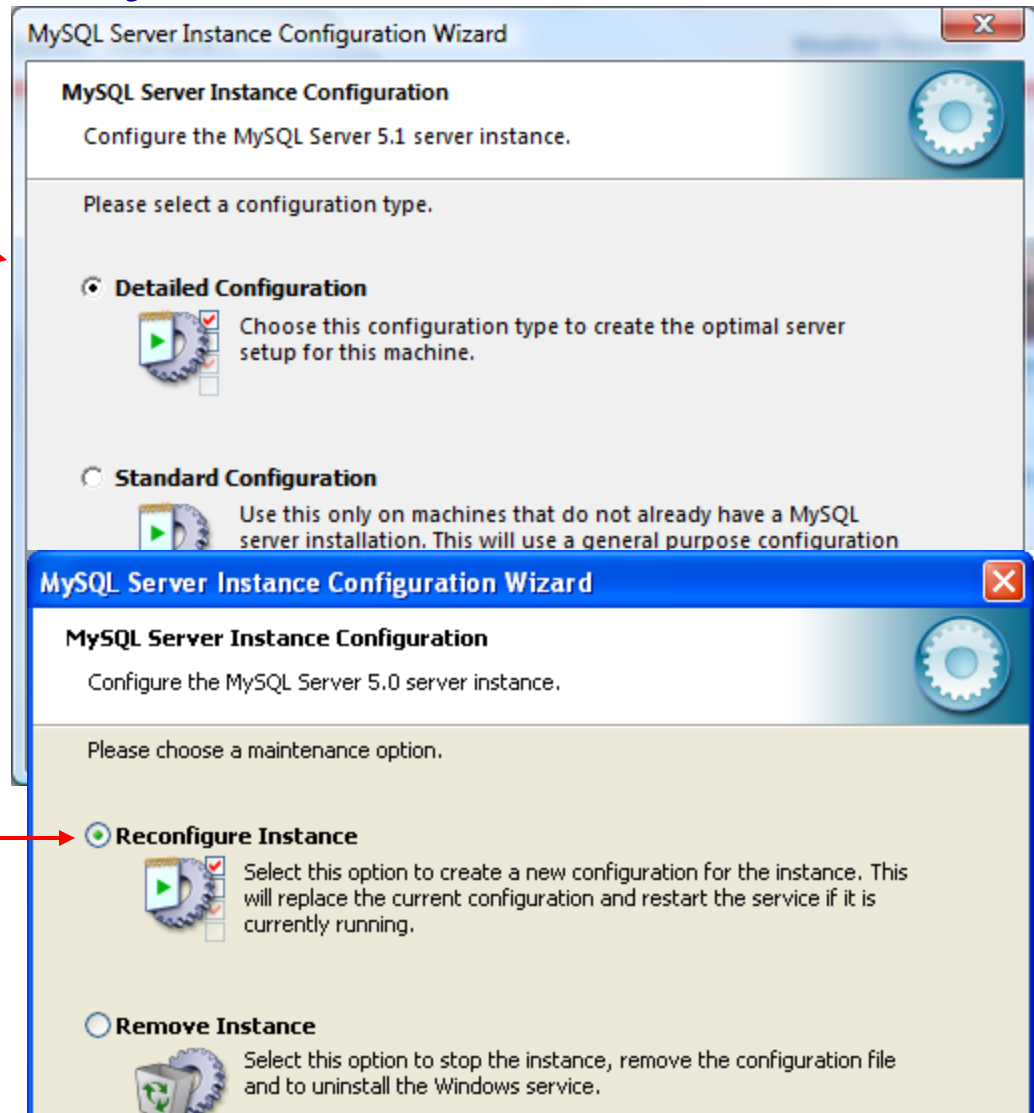




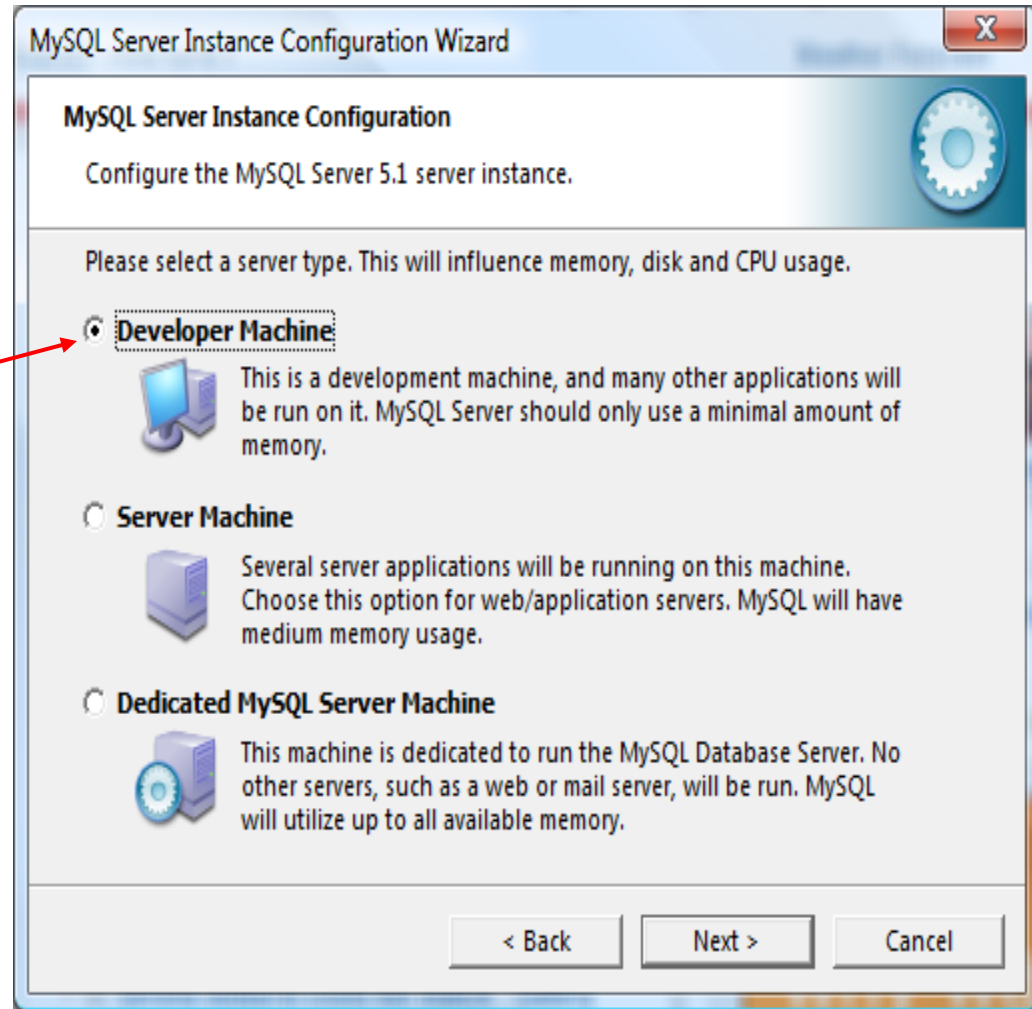
# Installing MySQL 5.1.38 (cont.)

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 5.1.38 (cont.)

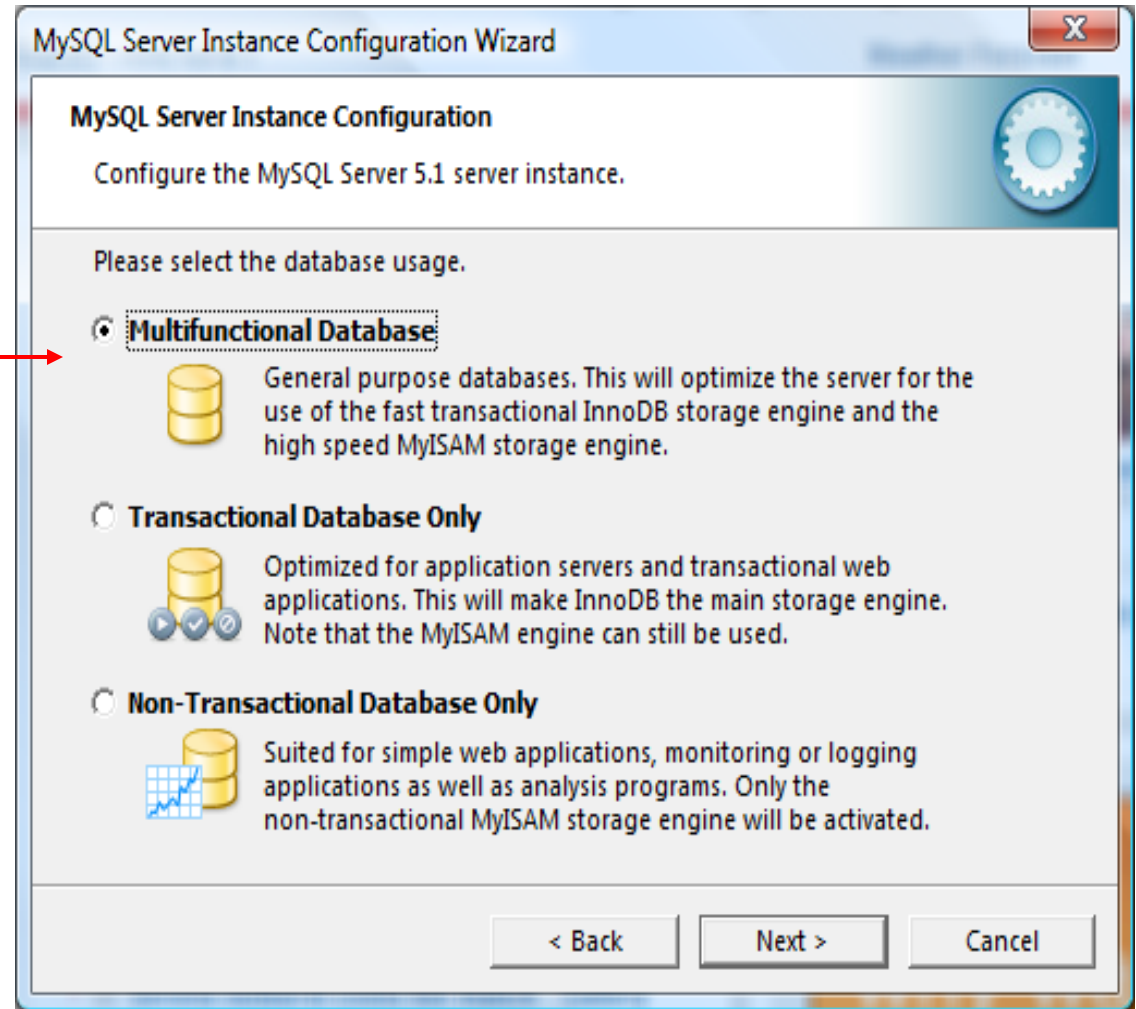


Choose the  
developer machine  
option



# Installing MySQL 5.1.38 (cont.)

Choose the  
multifunctional  
database option



# Installing MySQL 5.1.38 (cont.)


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.1 server instance.

Please select the drive for the InnoDB datafile, if you do not want to use the default settings.


**InnoDB Tablespace Settings**

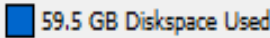
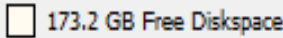
 Please choose the drive and directory where the InnoDB tablespace should be placed.

C: Installation Path ...

**Drive Info**

Volume Name:  
File System: NTFS



 59.5 GB Diskspace Used  173.2 GB Free Diskspace

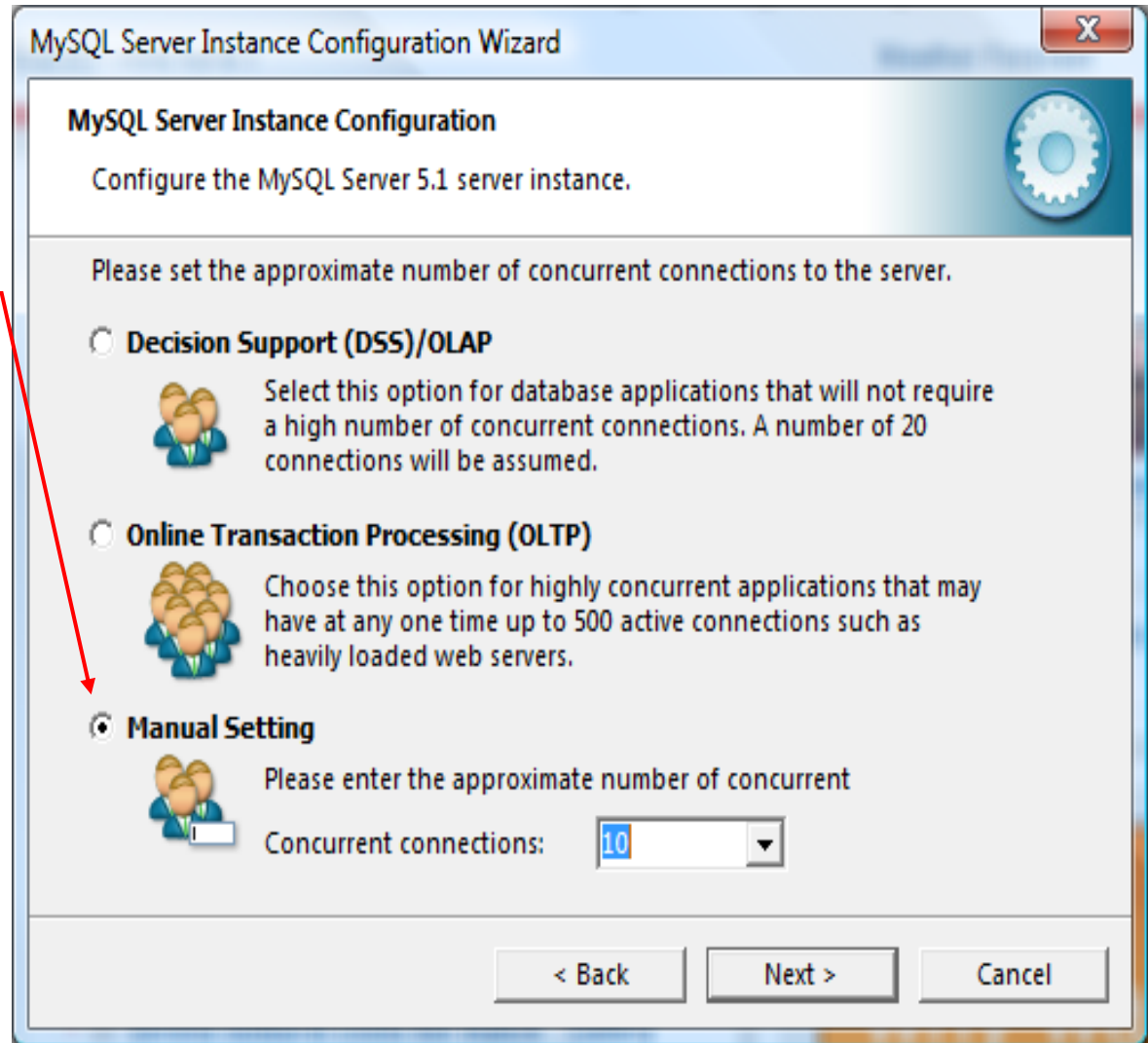
< Back Next > Cancel

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



# Installing MySQL 5.1.38 (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 2 or 3.



The image shows the 'MySQL Server Instance Configuration Wizard' window. The title bar reads 'MySQL Server Instance Configuration Wizard'. The main heading is 'MySQL Server Instance Configuration' with a subtext 'Configure the MySQL Server 5.1 server instance.' and a gear icon. The instruction says 'Please set the approximate number of concurrent connections to the server.' There are three radio button options: 'Decision Support (DSS)/OLAP', 'Online Transaction Processing (OLTP)', and 'Manual Setting'. The 'Manual Setting' option is selected. Below it, there is a text input field for 'Concurrent connections' with the value '10' entered. At the bottom are buttons for '< Back', 'Next >', and 'Cancel'. A red arrow points from the text box on the left to the 'Manual Setting' radio button.

MySQL Server Instance Configuration Wizard

**MySQL Server Instance Configuration**

Configure the MySQL Server 5.1 server instance.

Please set the approximate number of concurrent connections to the server.

☐ **Decision Support (DSS)/OLAP**

Select this option for database applications that will not require a high number of concurrent connections. A number of 20 connections will be assumed.

☐ **Online Transaction Processing (OLTP)**

Choose this option for highly concurrent applications that may have at any one time up to 500 active connections such as heavily loaded web servers.

☒ **Manual Setting**

Please enter the approximate number of concurrent

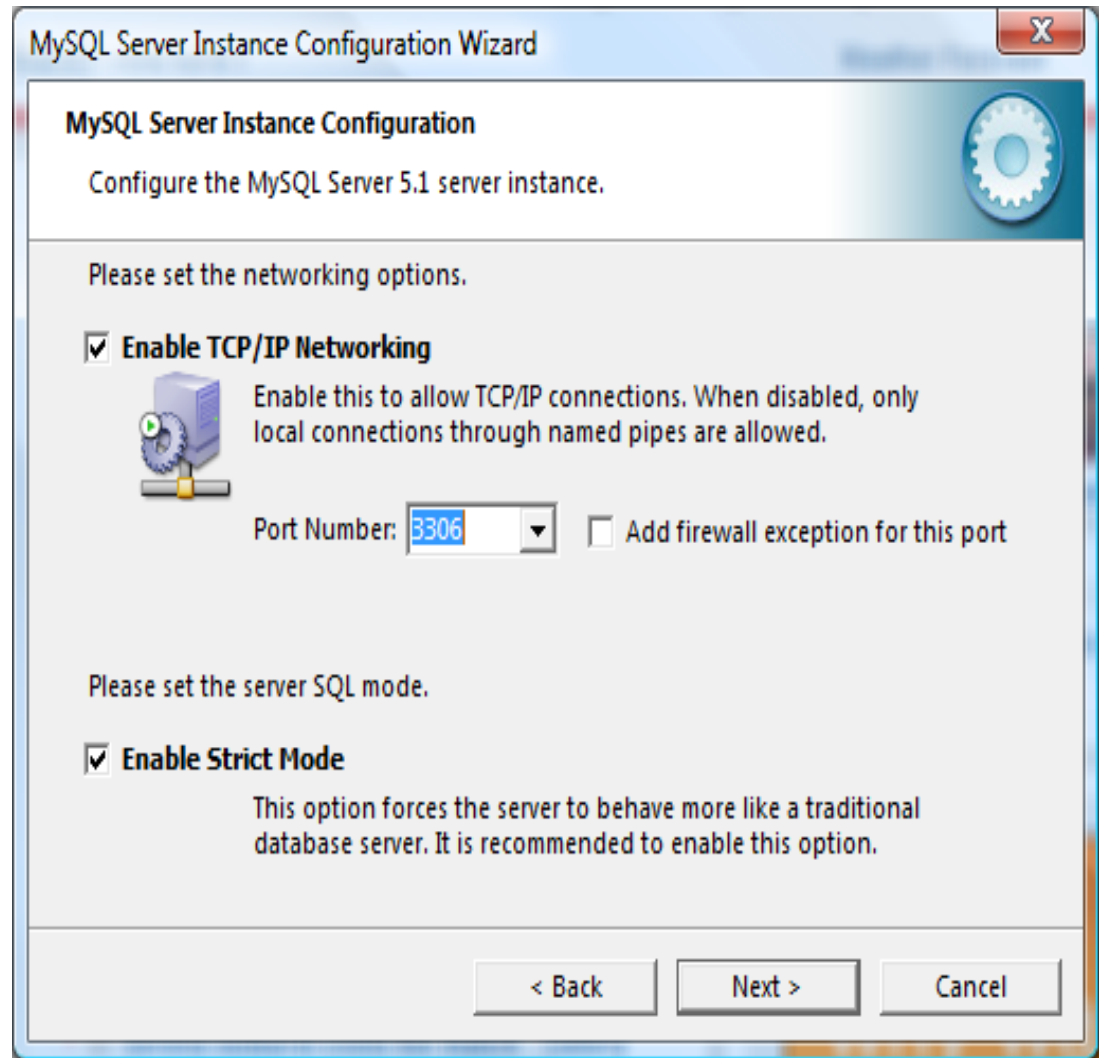
Concurrent connections:

< Back    Next >    Cancel





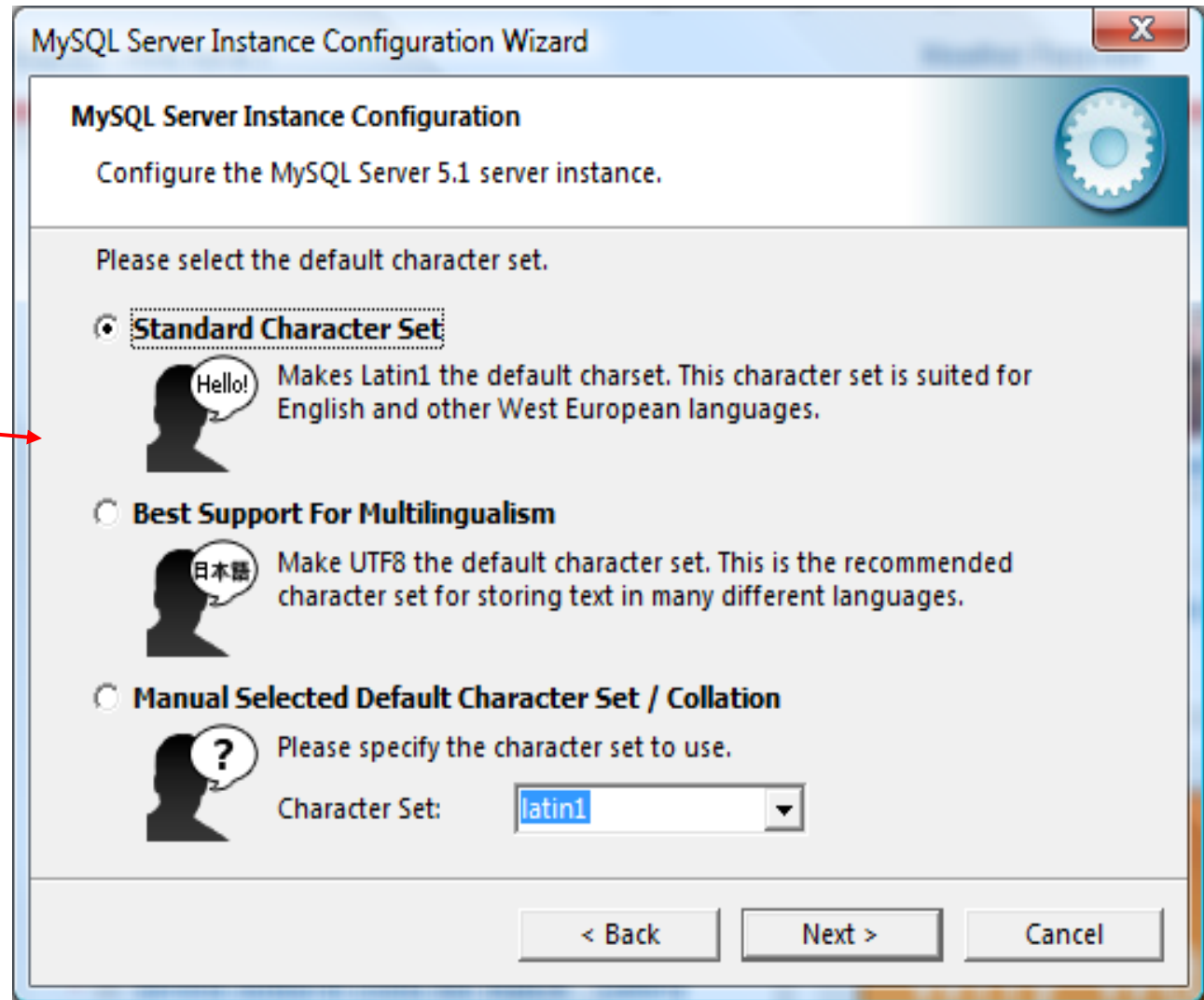
# Installing MySQL 5.1.38 (cont.)



Accept all  
defaults in this  
window



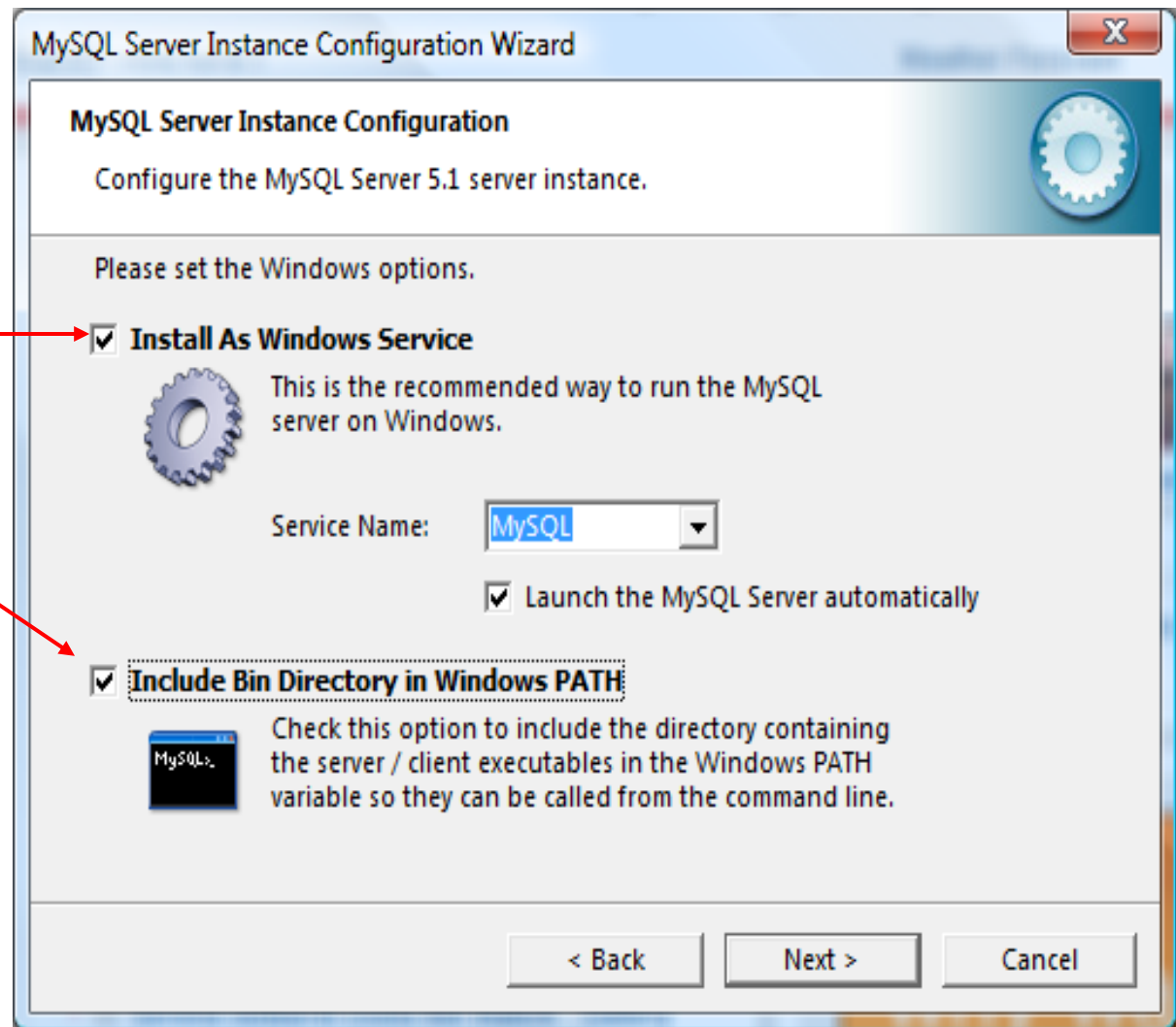
# Installing MySQL 5.1.38 (cont.)



Your choice  
again



# Installing MySQL 5.1.38 (cont.)



Accept default options

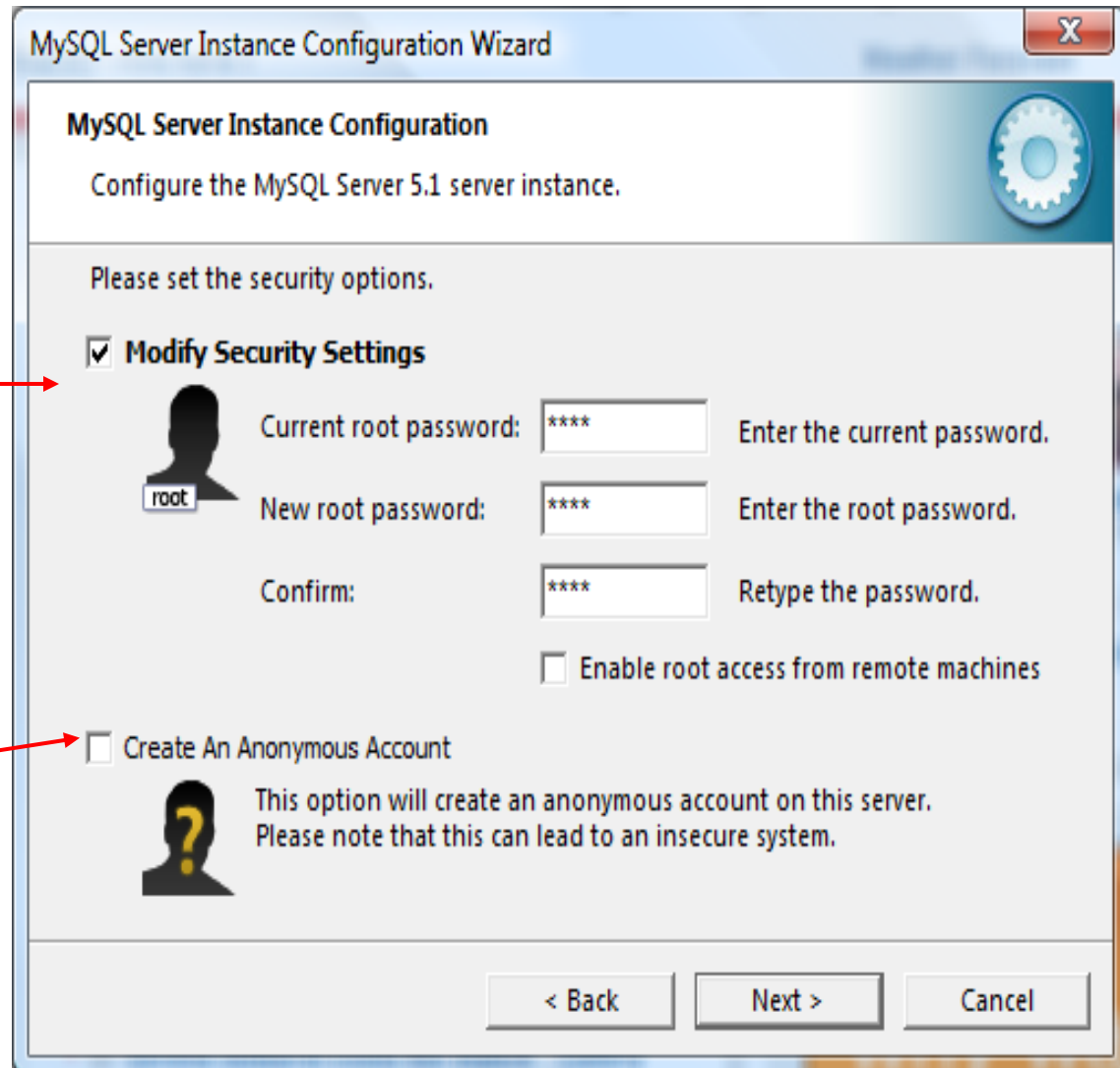
This option is not marked by default, but you can mark and accept it if you want to include MySQL file locations in your PATH statement.



# Installing MySQL 5.1.38 (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



The image shows the 'MySQL Server Instance Configuration Wizard' window. The title bar says 'MySQL Server Instance Configuration Wizard'. The main title is 'MySQL Server Instance Configuration' with a subtitle 'Configure the MySQL Server 5.1 server instance.' Below this, it says 'Please set the security options.' There are two main sections. The first section is 'Modify Security Settings', which is checked. It contains three password fields: 'Current root password:' (with '\*\*\*\*' in it), 'New root password:' (with '\*\*\*\*' in it), and 'Confirm:' (with '\*\*\*\*' in it). To the right of these fields are labels: 'Enter the current password.', 'Enter the root password.', and 'Retype the password.' Below these fields is an unchecked checkbox labeled 'Enable root access from remote machines'. The second section is 'Create An Anonymous Account', which is also unchecked. It features a question mark icon and the text: 'This option will create an anonymous account on this server. Please note that this can lead to an insecure system.' At the bottom of the window are three buttons: '< Back', 'Next >', and 'Cancel'.


MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.1 server instance.

Please set the security options.

☒ **Modify Security Settings**


 Current root password:  Enter the current password.

New root password:  Enter the root password.

Confirm:  Retype the password.

☐ Enable root access from remote machines

☐ **Create An Anonymous Account**

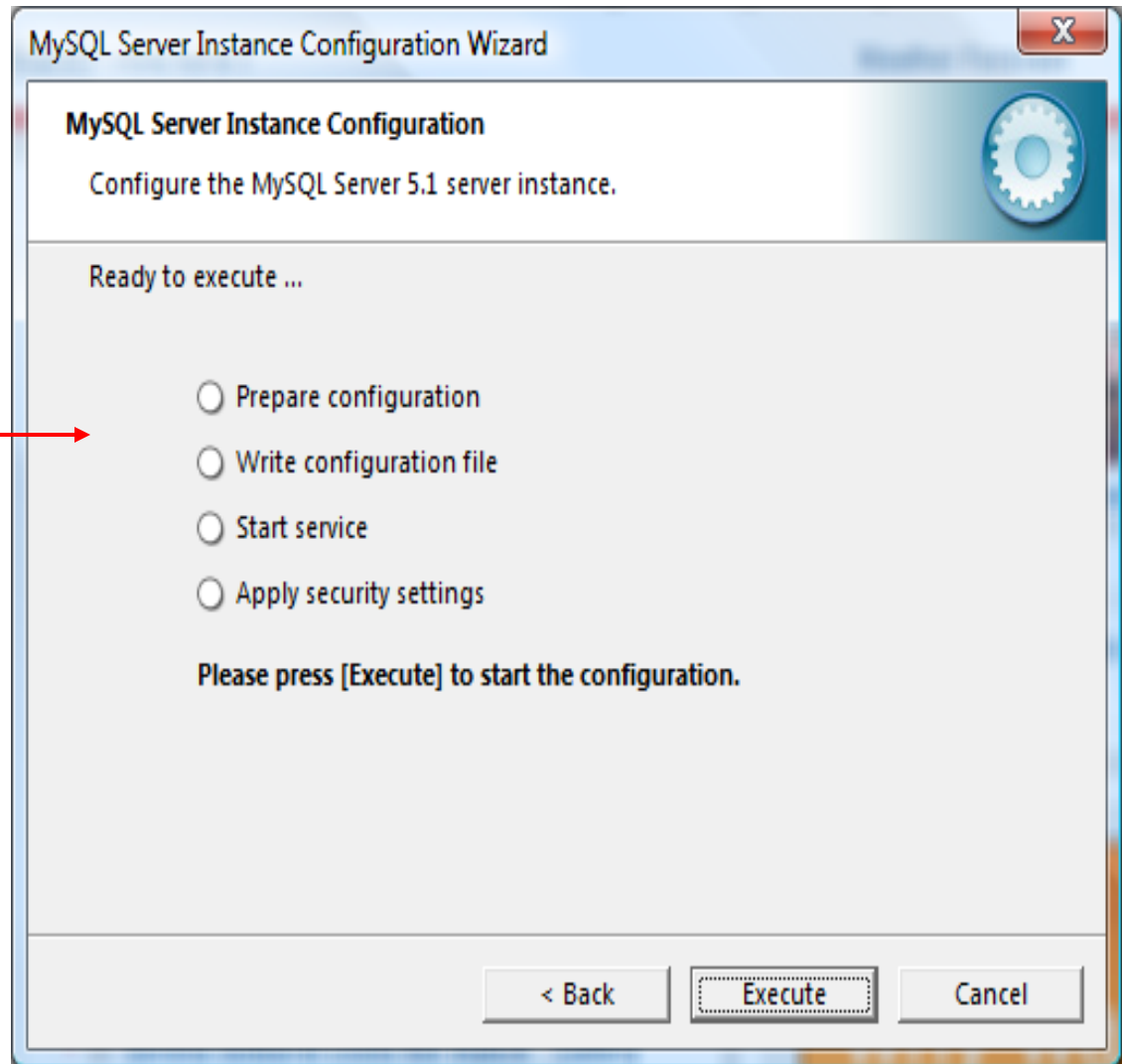
 This option will create an anonymous account on this server. Please note that this can lead to an insecure system.

< Back    Next >    Cancel

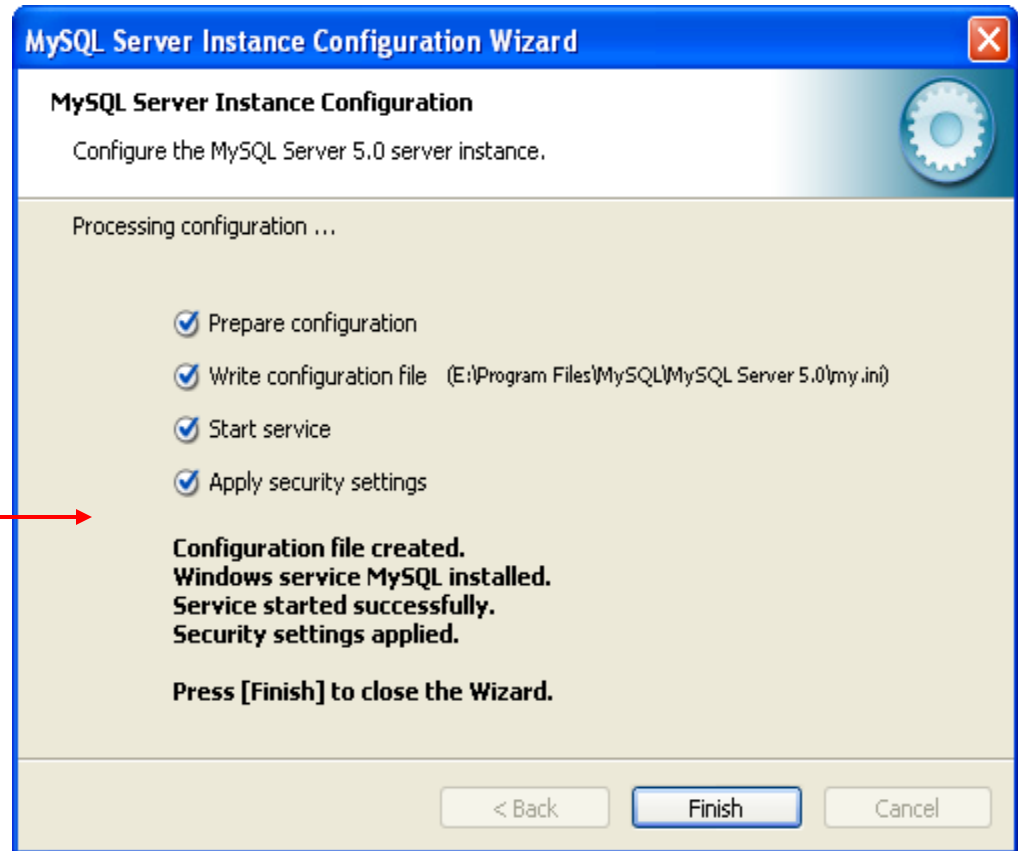


# Installing MySQL 5.1.38 (cont.)

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 5.1.38 (cont.)



You've successfully  
installed MySQL!!



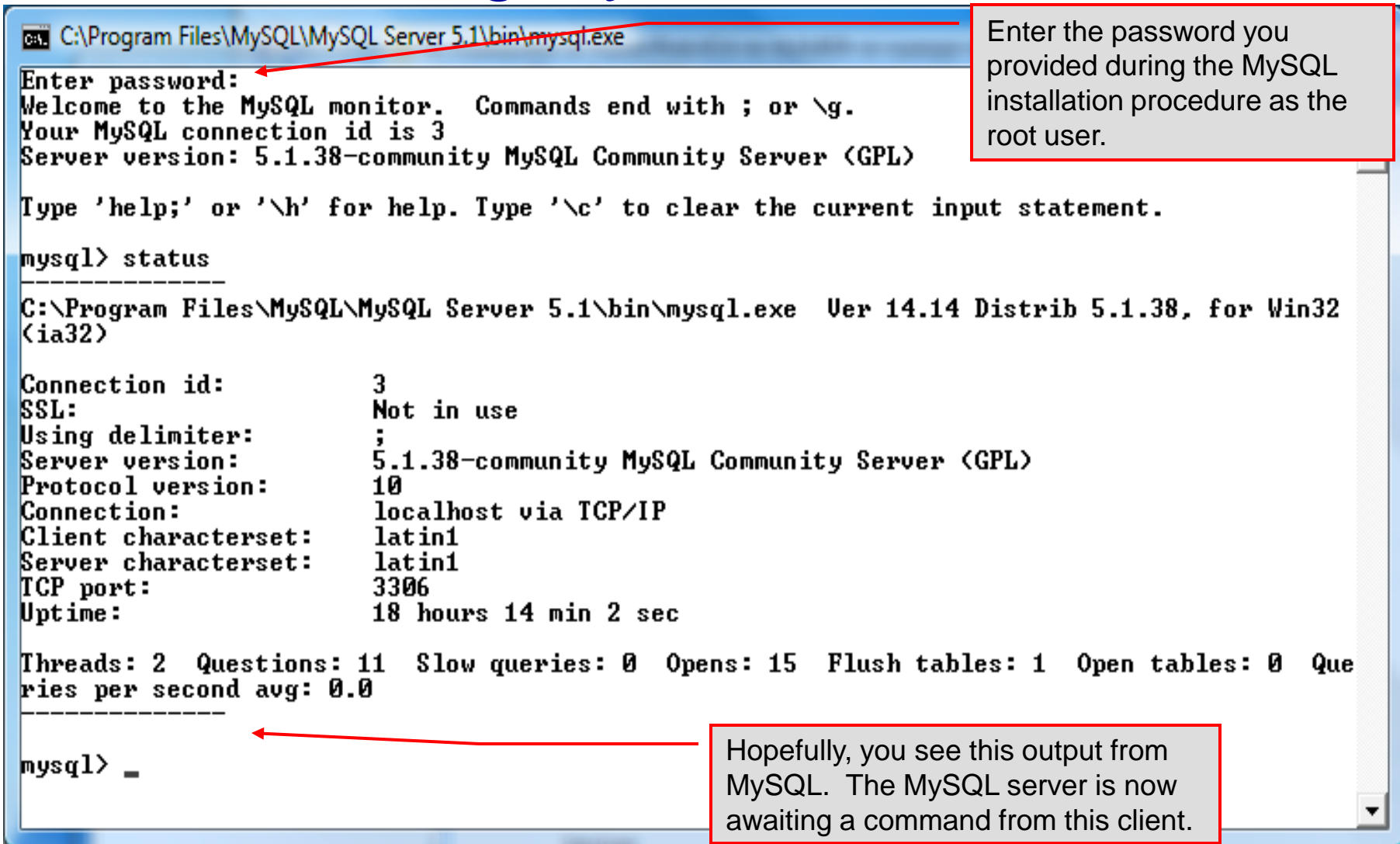
# Running MySQL 5.1.38

- 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 5.1.38 (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 3
Server version: 5.1.38-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 5.1.38, for Win32
(ia32)

Connection id:          3
SSL:                   Not in use
Using delimiter:        ;
Server version:         5.1.38-community MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Client characterset:    latin1
Server characterset:    latin1
TCP port:               3306
Uptime:                 18 hours 14 min 2 sec

Threads: 2  Questions: 11  Slow queries: 0  Opens: 15  Flush tables: 1  Open tables: 0  Que
ries per second avg: 0.0

mysql> _
```

Enter the password you provided during the MySQL installation procedure as the root user.

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



# Running MySQL 5.1.38 (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 3
Server version: 5.1.38-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 5.1.38, for Win32
(ia32)

Connection id:          3
SSL:                    Not in use
Using delimiter:        ;
Server version:         5.1.38-community MySQL Community Server (GPL)
Protocol version:       10
Connection:             localhost via TCP/IP
Client characterset:     latin1
Server characterset:     latin1
TCP port:               3306
Uptime:                 18 hours 14 min 2 sec

Threads: 2  Questions: 11  Slow queries: 0  Opens: 15  Flush tables: 1  Open tables: 0  Que
ries per second avg: 0.0
-----

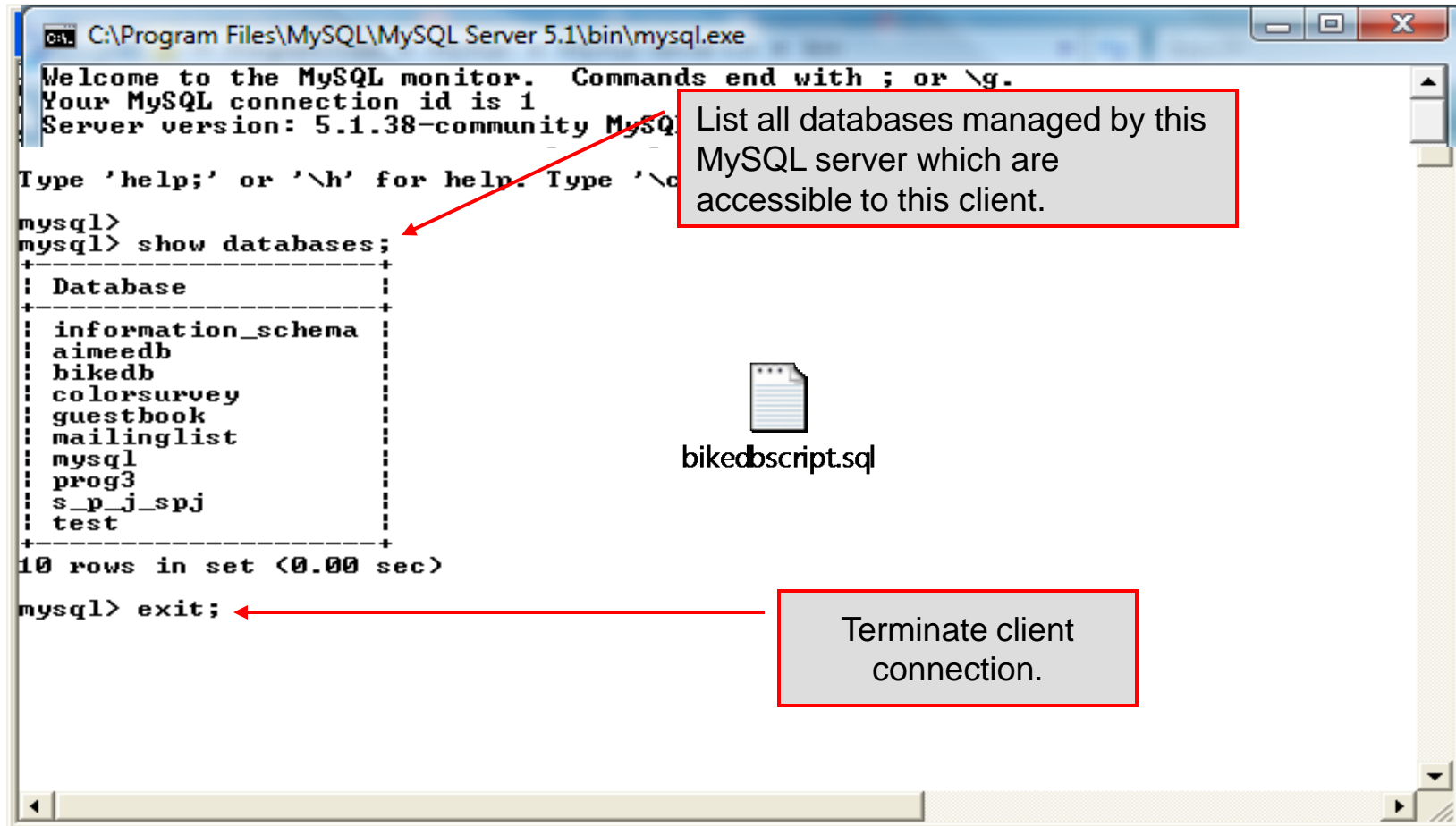
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.03 sec)
```

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

New Installations Only!



# Running MySQL 5.1.38 (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: 5.1.38-community MySQL

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

mysql>
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| aineedb    |
| bikedb     |
| colorsurvey |
| guestbook  |
| mailinglist |
| mysql      |
| prog3      |
| s_p_j_spj  |
| test       |
+-----+
10 rows in set (0.00 sec)

mysql> exit;
```

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

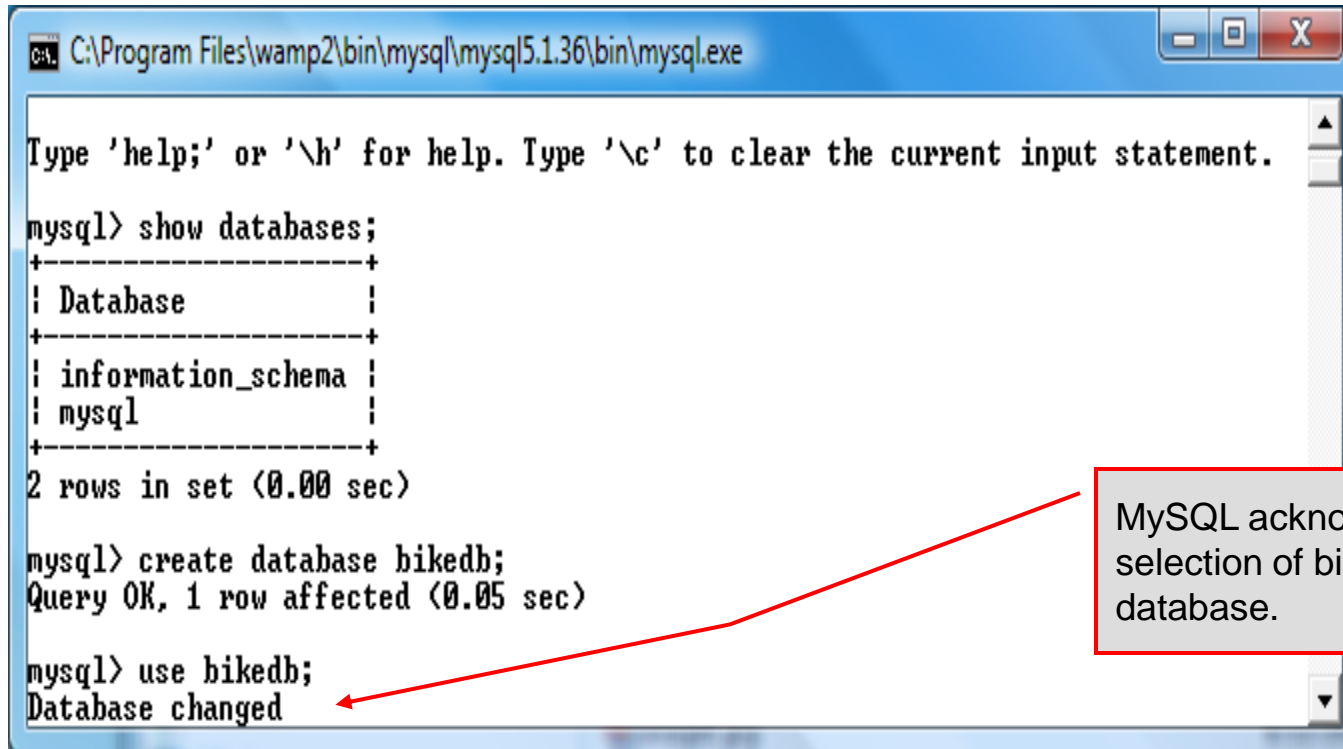
bikedbscript.sql

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.



A screenshot of a MySQL command-line window. The title bar shows the path `C:\Program Files\wamp2\bin\mysql\mysql5.1.36\bin\mysql.exe`. The window contains the following text:

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

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
+-----+
2 rows in set (0.00 sec)

mysql> create database bikedb;
Query OK, 1 row affected (0.05 sec)

mysql> use bikedb;
Database changed
```

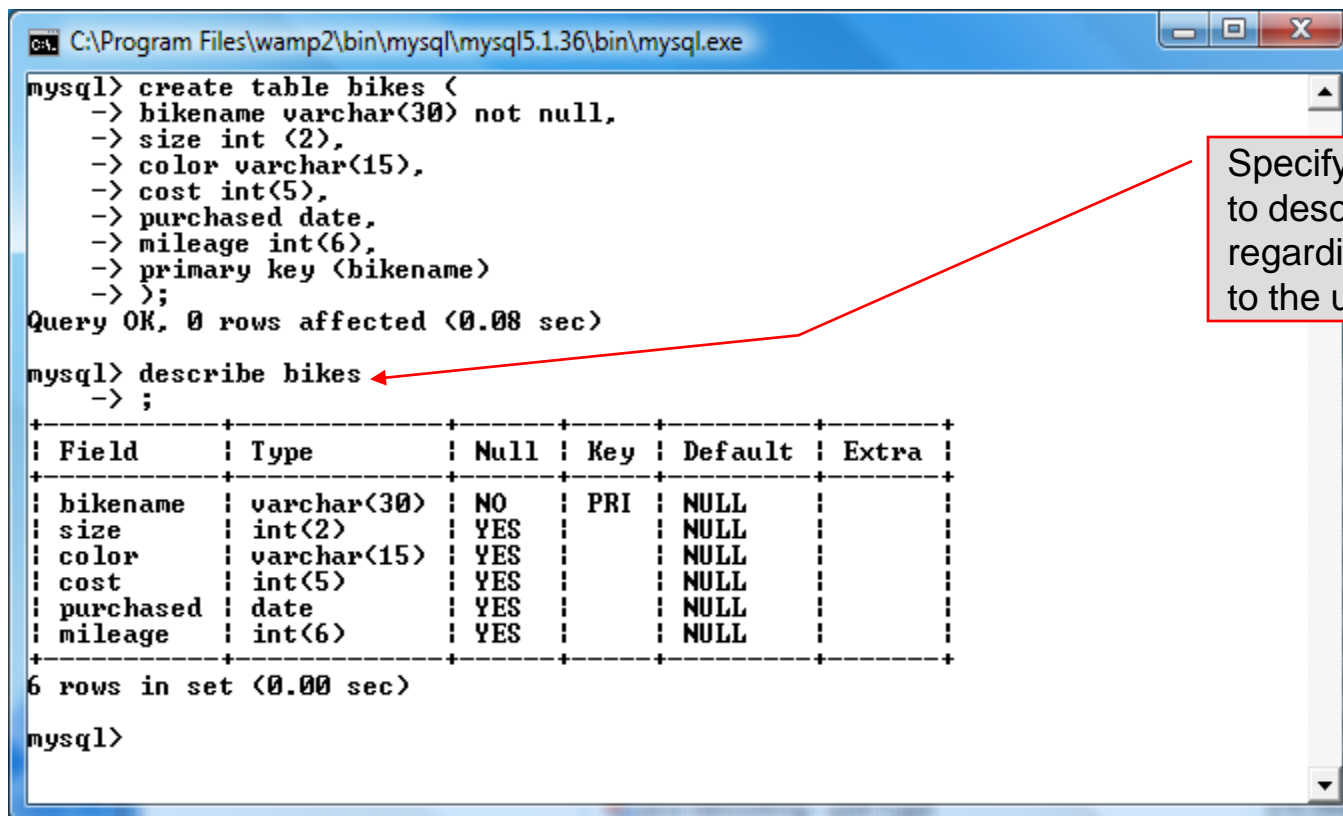
A red arrow points from a text box on the right to the `use bikedb;` command and its output.

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\wamp2\bin\mysql\mysql5.1.36\bin\mysql.exe
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 (0.00 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)

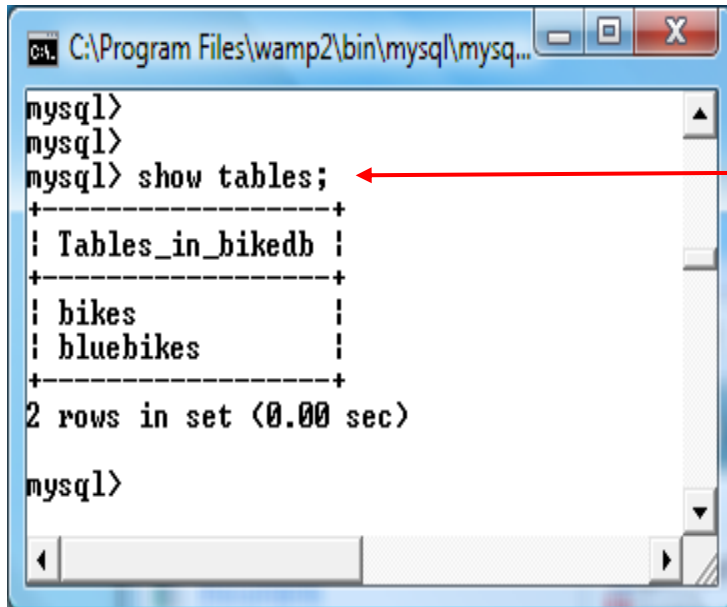
mysql>
```

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.



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

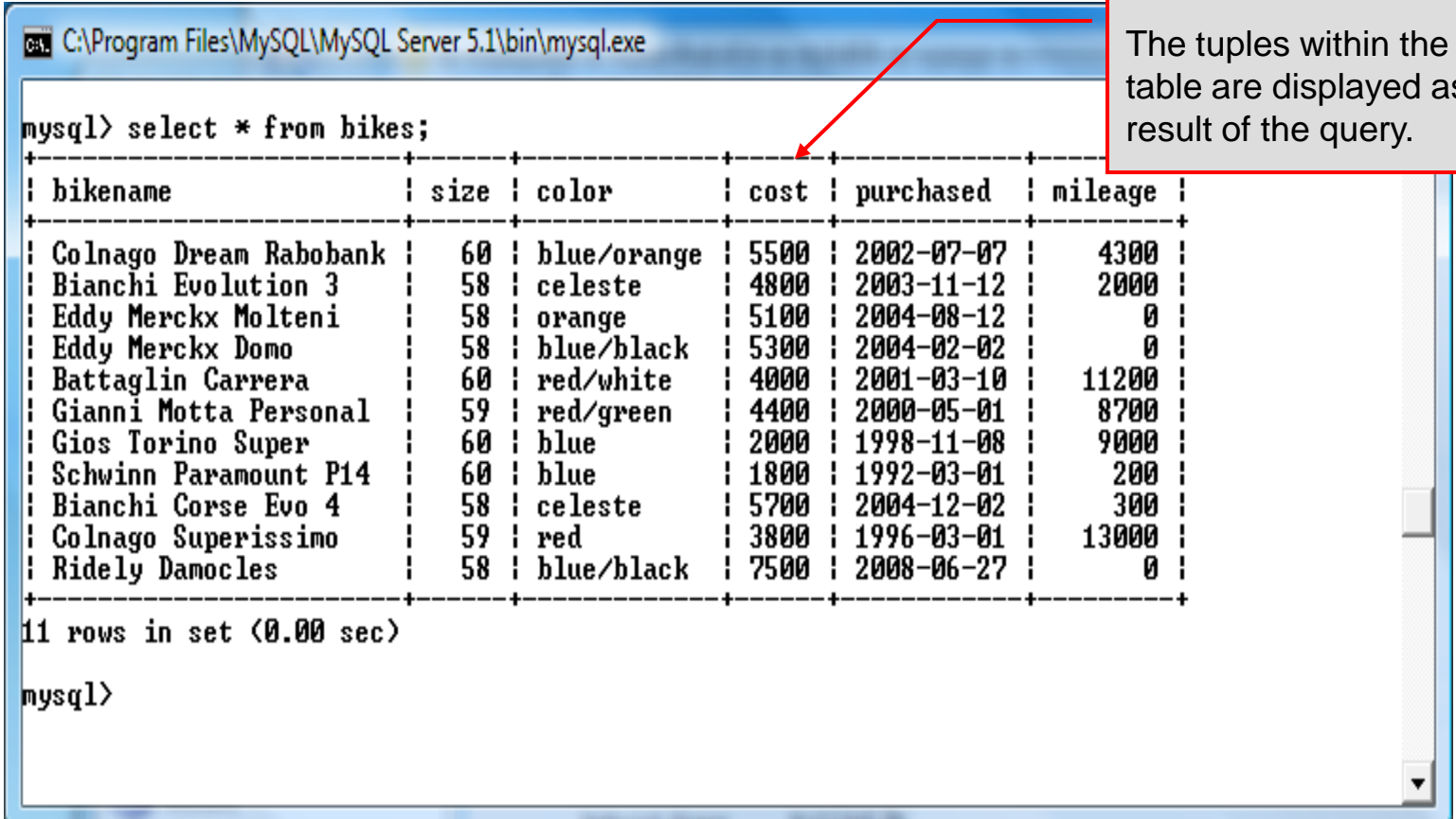
mysql>
```

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.



The screenshot shows a Windows command prompt window titled "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe". The prompt is `mysql>`. The user has entered the query `select * from bikes;`. The result is displayed as a table with 6 columns: `bikename`, `size`, `color`, `cost`, `purchased`, and `mileage`. There are 11 rows of data. Below the table, it says "11 rows in set (0.00 sec)". The prompt is now `mysql>`. A red arrow points from a text box to the `cost` column header.

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

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
Ridely Damocles	58	blue/black	7500	2008-06-27	0

11 rows in set (0.00 sec)

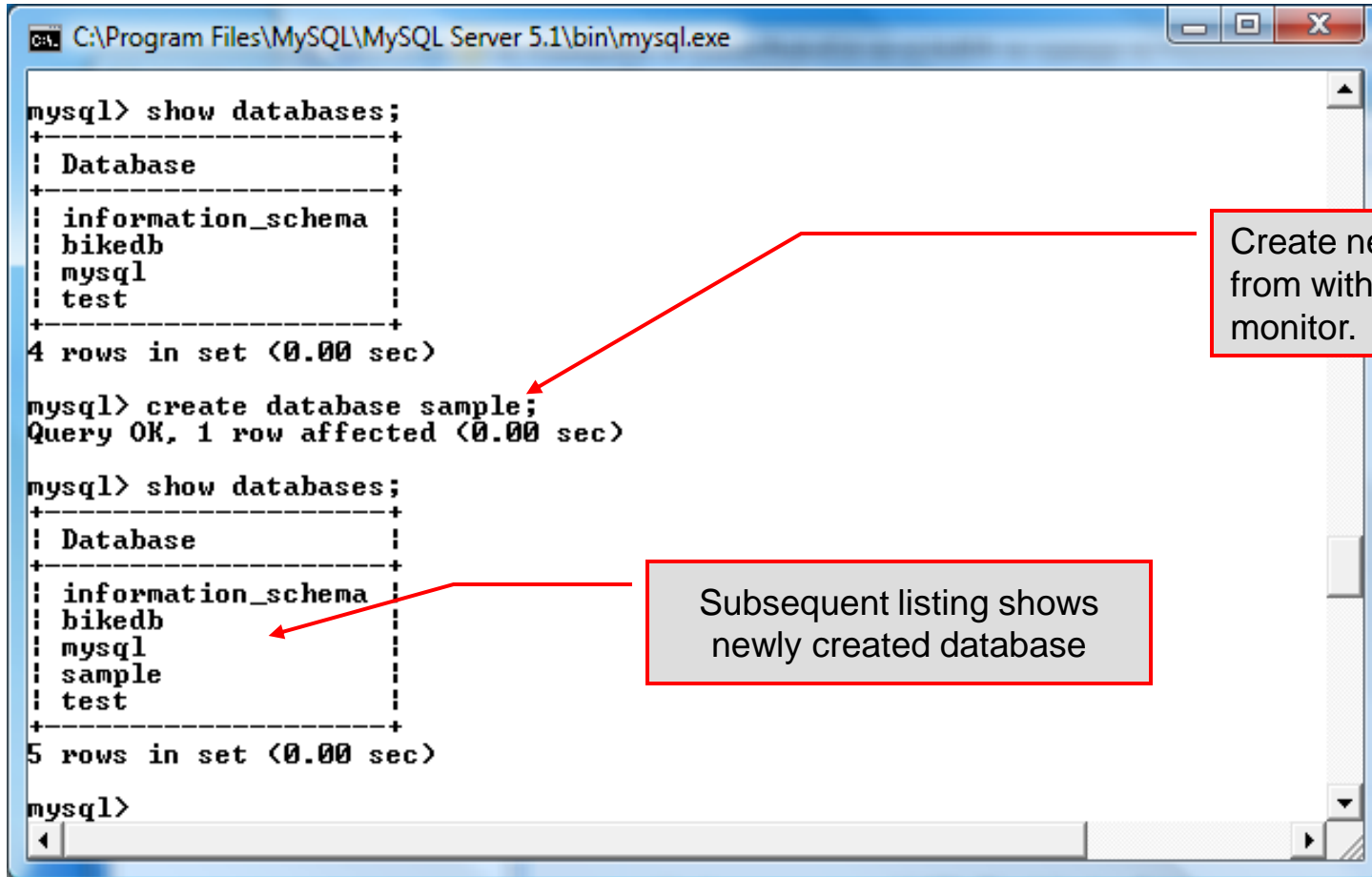
mysql>





# Creating a Database in MySQL

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



The screenshot shows a Windows command prompt window titled "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe". The prompt is "mysql>". The user enters the command "show databases;", which returns a table listing four databases: information\_schema, bikedb, mysql, and test. The user then enters "create database sample;", which returns "Query OK, 1 row affected (0.00 sec)". The user enters "show databases;" again, which returns a table listing five databases: information\_schema, bikedb, mysql, sample, and test. A red arrow points from the "sample" database in the second table to a text box on the right. Another red arrow points from the "sample" database in the second table to a text box below it.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| bikedb         |
| mysql          |
| test           |
+-----+
4 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         |
| mysql          |
| sample         |
| test           |
+-----+
5 rows in set (0.00 sec)

mysql>
```

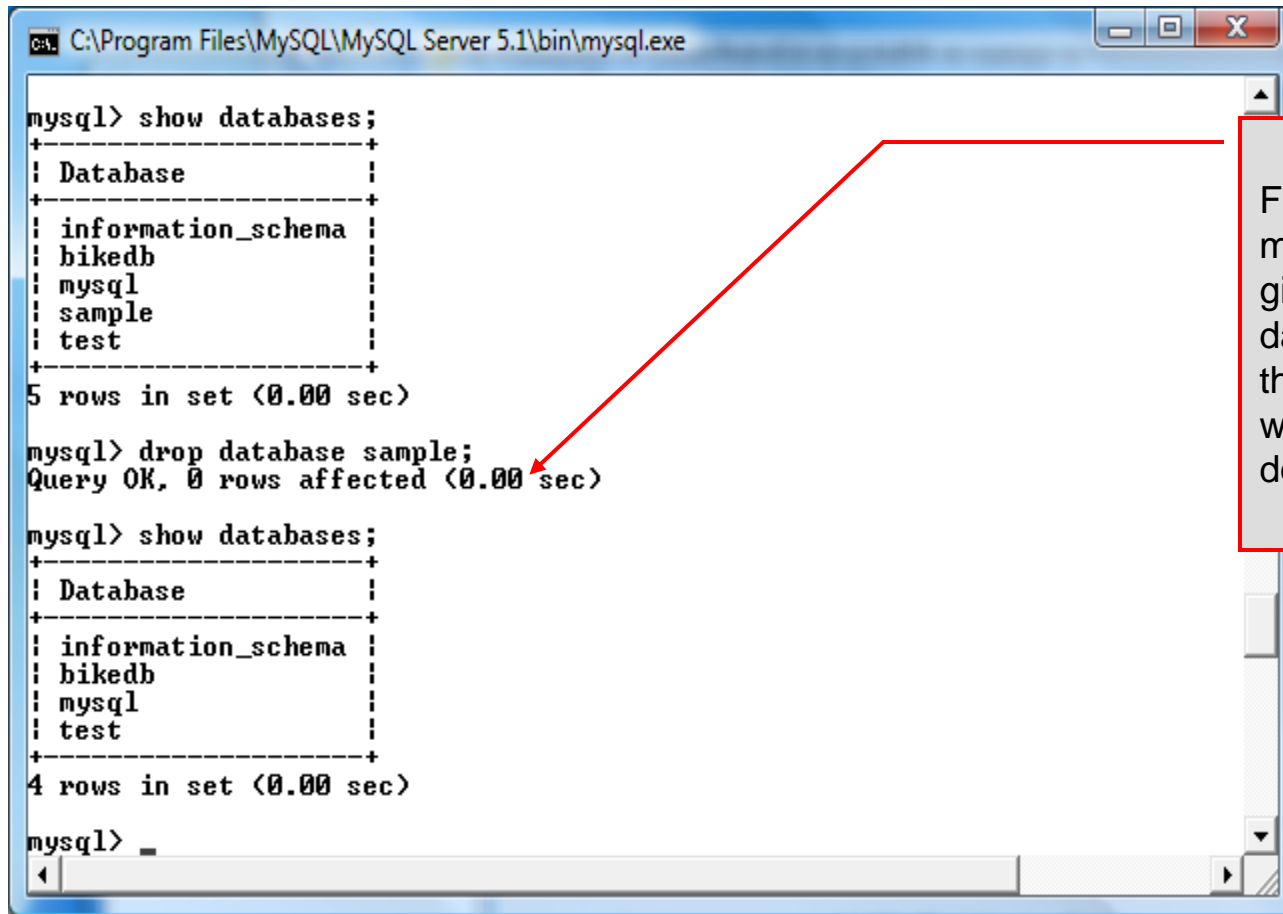
Create new database from within MySQL monitor.

Subsequent listing shows newly created database



# Dropping a Database in MySQL

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



The screenshot shows a Windows command prompt window titled "C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe". The prompt is "mysql>". The user enters "show databases;" and the output is a table with one column "Database" and five rows: "information\_schema", "bikedb", "mysql", "sample", and "test". Below the table, it says "5 rows in set (0.00 sec)". The user then enters "drop database sample;" and the output is "Query OK, 0 rows affected (0.00 sec)". The user enters "show databases;" again, and the output is a table with one column "Database" and four rows: "information\_schema", "bikedb", "mysql", and "test". Below the table, it says "4 rows in set (0.00 sec)". The prompt is now "mysql> \_". A red arrow points from the text box on the right to the "drop database sample;" command.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| bikedb       |
| mysql        |
| sample       |
| test         |
+-----+
5 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       |
| mysql        |
| test         |
+-----+
4 rows in set (0.00 sec)

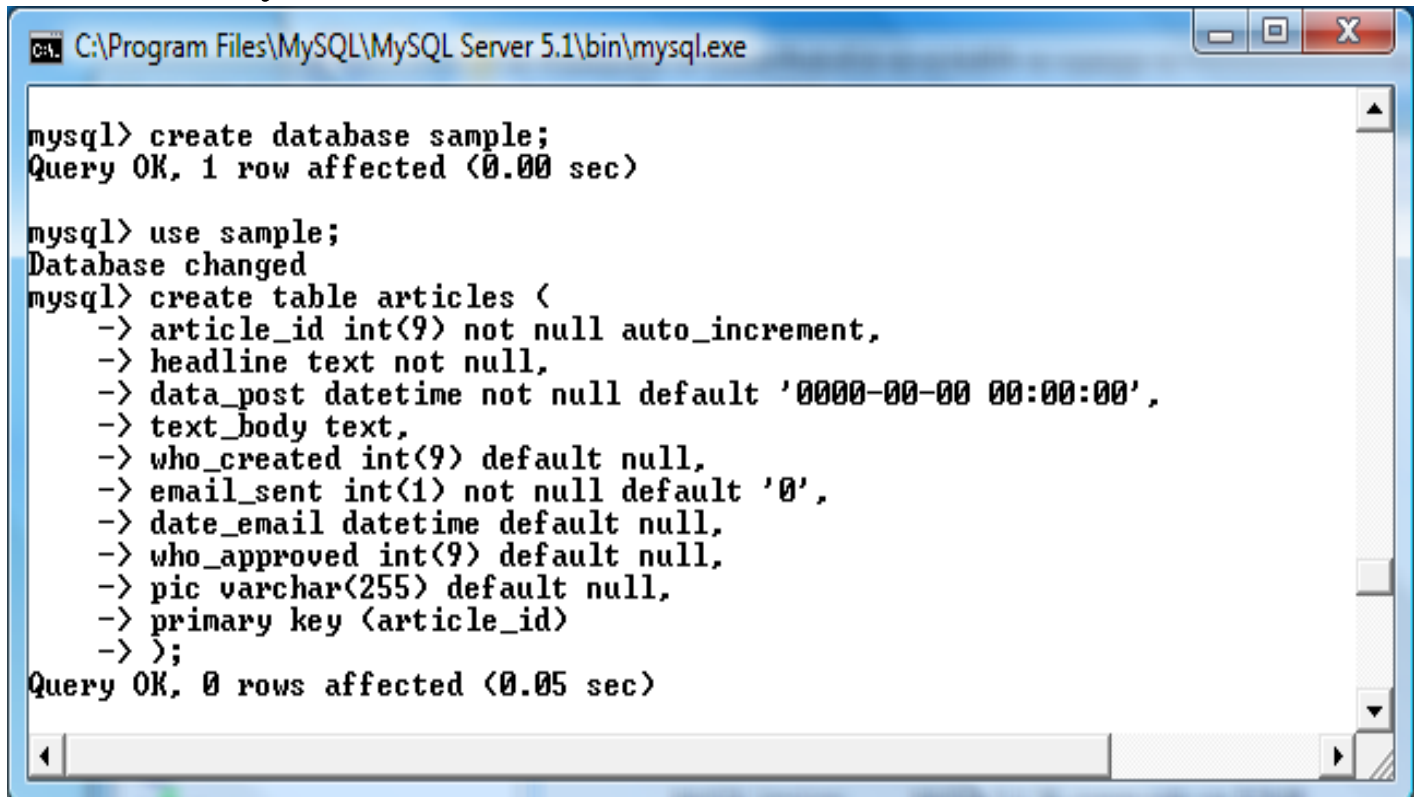
mysql> _
```

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

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 (0.05 sec)
```



# Manipulating Tables in MySQL (cont.)

Screen shot that describes the newly created table.

```
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 (0.02 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

Database changed

```
mysql> select * from bikes;
```

bikename	size	color	cost	purchased	mile
Colnago Dream Rabobank	60	blue/orange	5500	2002-07-07	4
Bianchi Evolution 3	58	celeste	4800	2003-11-12	2
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	11
Gianni Motta Personal	59	red/green	4400	2000-05-01	8
Gios Torino Super	60	blue	2000	1998-11-08	9
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	13
Ridely Damocles	58	blue/black	7500	2008-06-27	

11 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	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> _
```

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 5.1.38.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
<code>add [column] column_declaration [first   after column_name]</code>	Add a column to the table
<code>alter [column] column_name {set default literal   drop default}</code>	Specify new default value for a column or remove old default
<code>change [column] column_name column_declaration</code>	Modify column declaration with renaming of column
<code>modify [column] column_declaration</code>	Modify column declaration without renaming column
<code>drop [column] column_name</code>	Drop a column and all data contained within it.
<code>rename [as] new_table_name</code>	Rename a table
<code>table_options</code>	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.

The screenshot shows a MySQL command prompt window with the following content:

```
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, 11 rows affected (0.05 sec)  
Records: 11 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> _
```

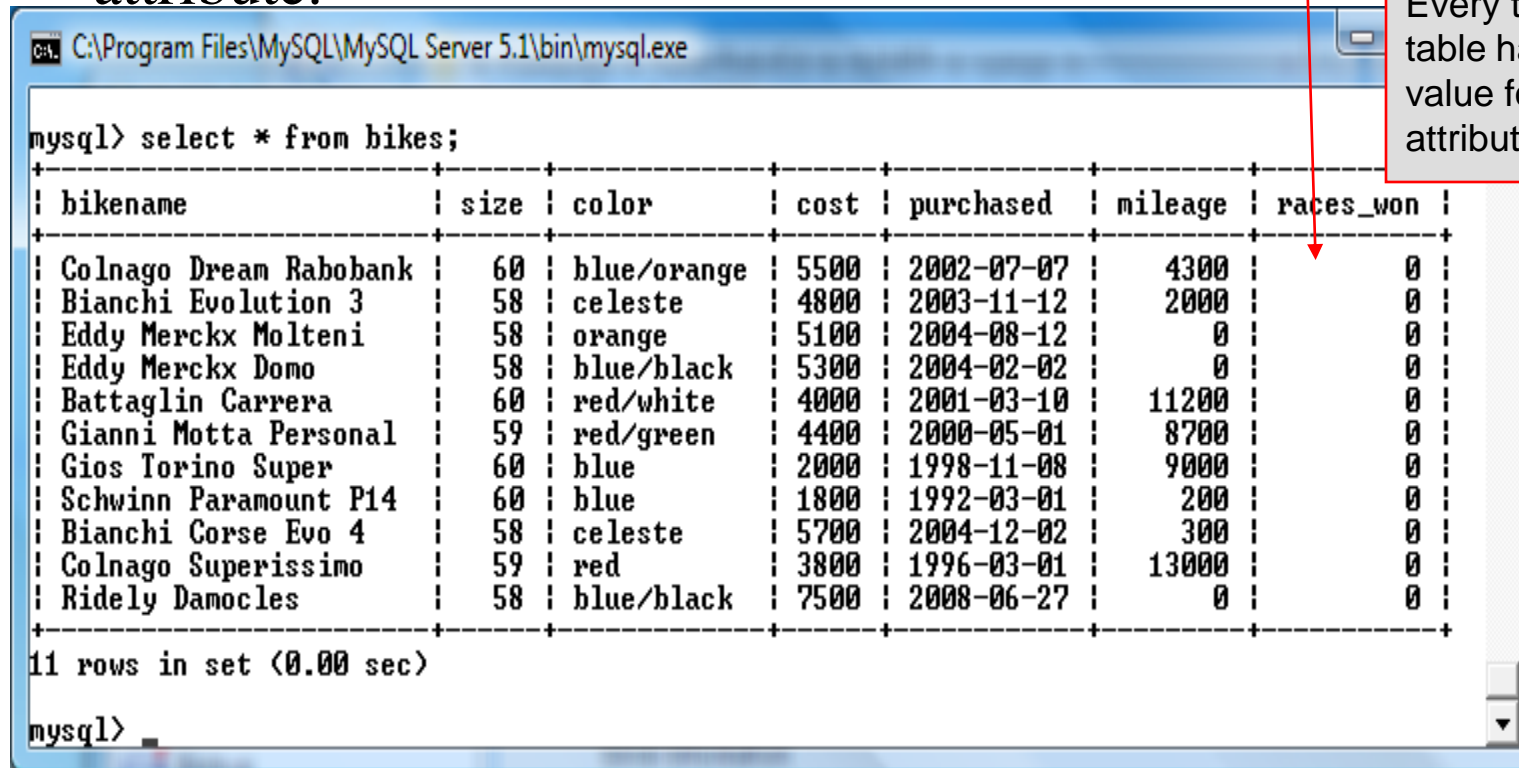
Annotations:

- A red box labeled "Schema of bikes before alteration" points to the first table.
- A blue box labeled "There are 11 rows affected because this table currently contains 11 tuples (rows) and the new attribute has been added to all rows." points to the command output.
- A green box labeled "Bikes table after the addition of a new column named races\_won" points to the second table.



## 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.



```
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
Ridely Damocles	58	blue/black	7500	2008-06-27	0	0

11 rows in set (0.00 sec)

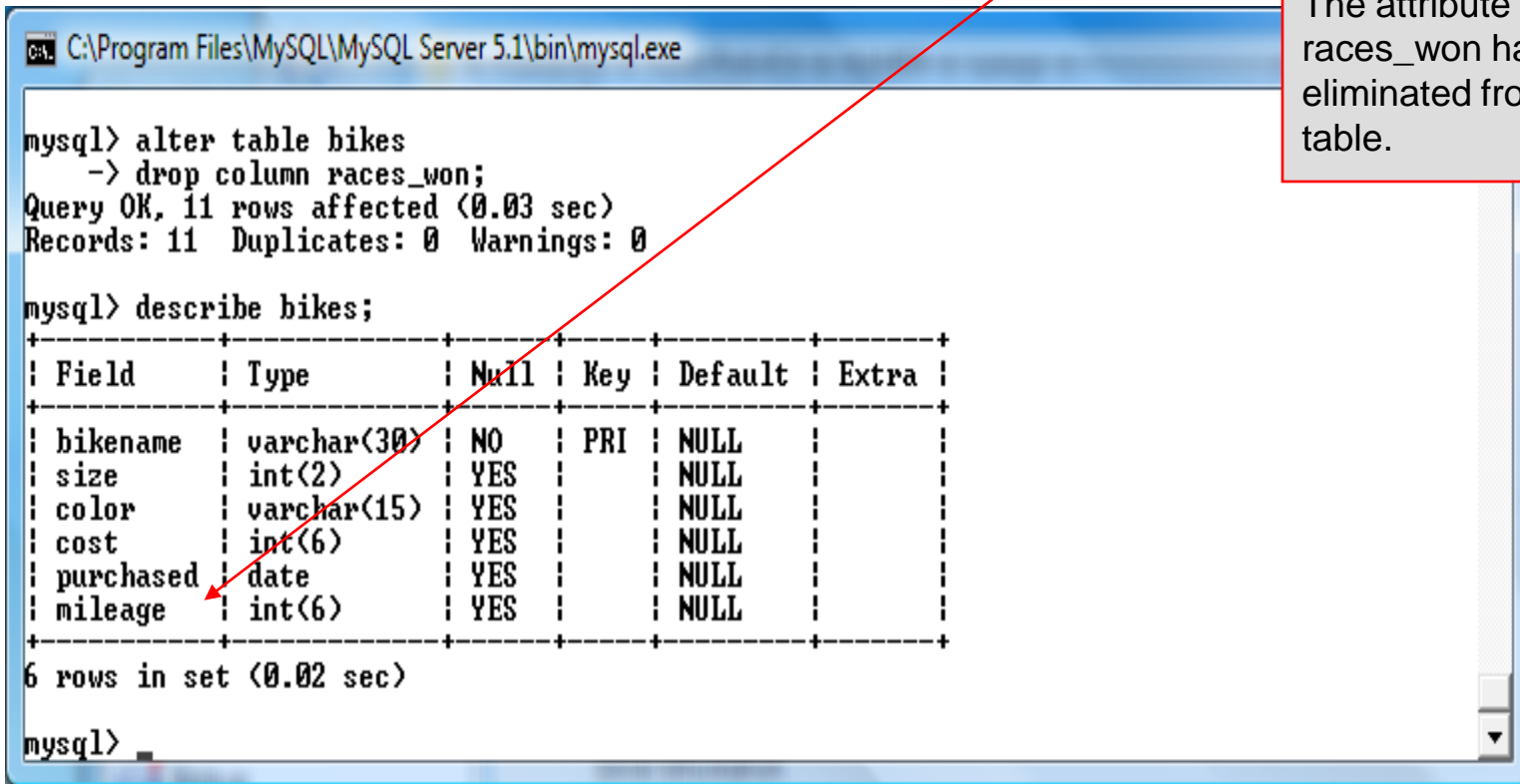
```
mysql>
```

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



# 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.



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

mysql> alter table bikes
  -> drop column races_won;
Query OK, 11 rows affected (0.03 sec)
Records: 11  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 (0.02 sec)

mysql>

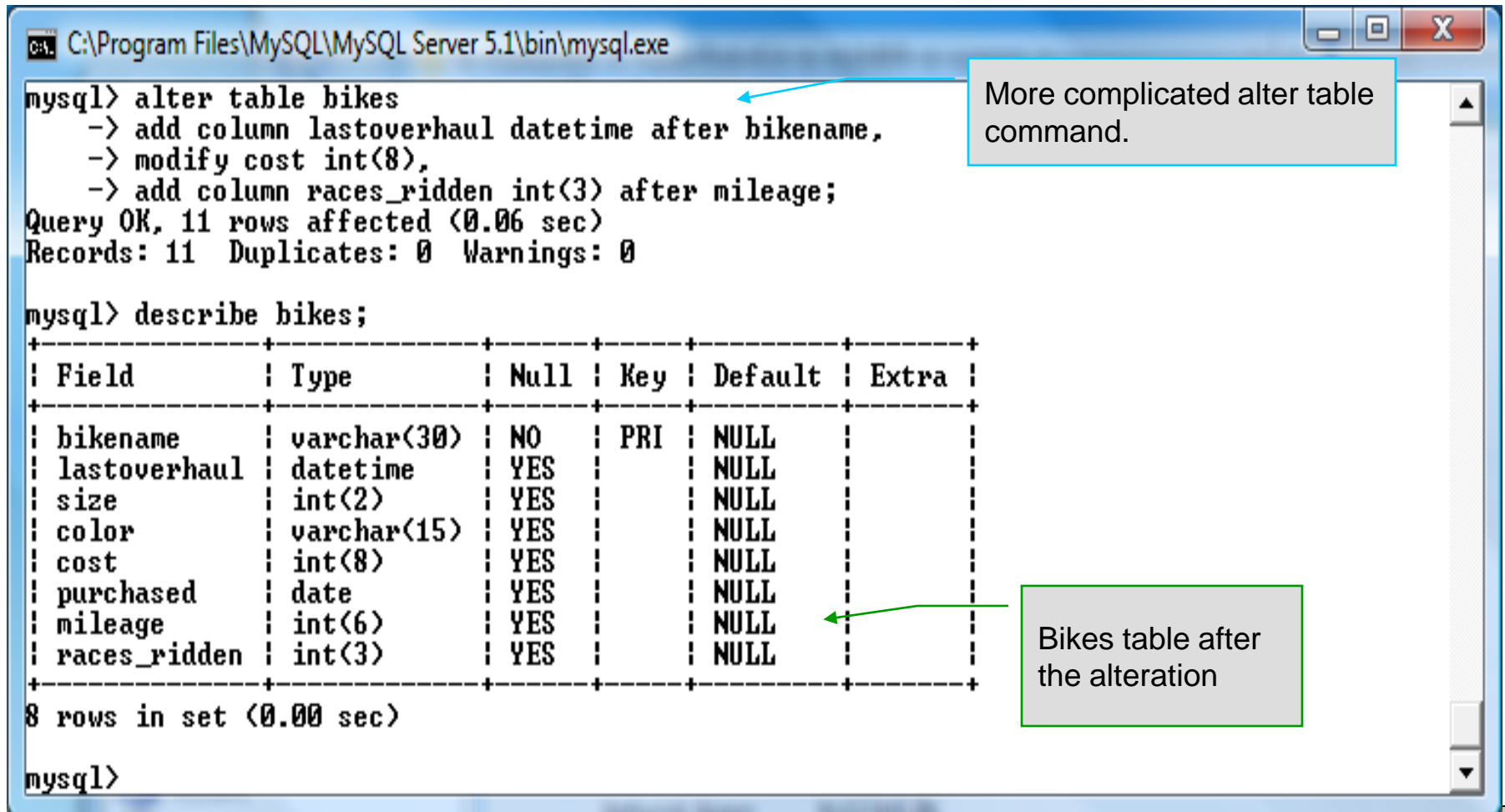
The attribute races\_won has been eliminated from the table.





# Altering A Table (cont.)

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



The screenshot shows a MySQL command window with the following text:

```
mysql> alter table bikes
  -> add column lastoverhaul datetime after bikename,
  -> modify cost int(8),
  -> add column races_ridden int(3) after mileage;
Query OK, 11 rows affected (0.06 sec)
Records: 11  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>

Annotations:

- A blue arrow points from the text "More complicated alter table command." to the `alter table bikes` command.
- A green arrow points from the text "Bikes table after the alteration" to the `describe bikes;` output table.



# 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.



```
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
-> set bikename="Eddy Merckx MXM",
-> 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	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 in set (0.00 sec)

mysql>

# Examples: Inserting Data Into A Table

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 MXM", 58, "black/red",8000,"2008-02-02",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 MXM       | 58   | black/red  | 8000  | 2008-02-02 | 150    |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.01 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> create table celestebikes like bikes;
Query OK, 0 rows affected (0.00 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

```
1 #SQL commands in a script file
2 drop database if exists testdb;
3
4 create database testdb;
5
6 use testdb;
7
8 create table states (
9     name varchar(15) not null,
10    abbrev char(2),
11    capital varchar(25),
12    population integer,
13    square_miles integer,
14    primary key (name)
15 );
16
17 insert into states values ('Florida', 'FL', 'Tallahassee', 18328240, 54153);
18 insert into states values ('New York', 'NY', 'Albany', 194909297, 54556);
19 insert into states values ('Indiana', 'IN', 'Indianapolis', 6376792, 35789);
20 insert into states values ('Maryland', 'MD', 'Annapolis', 5633597, 9975);
21
22 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.

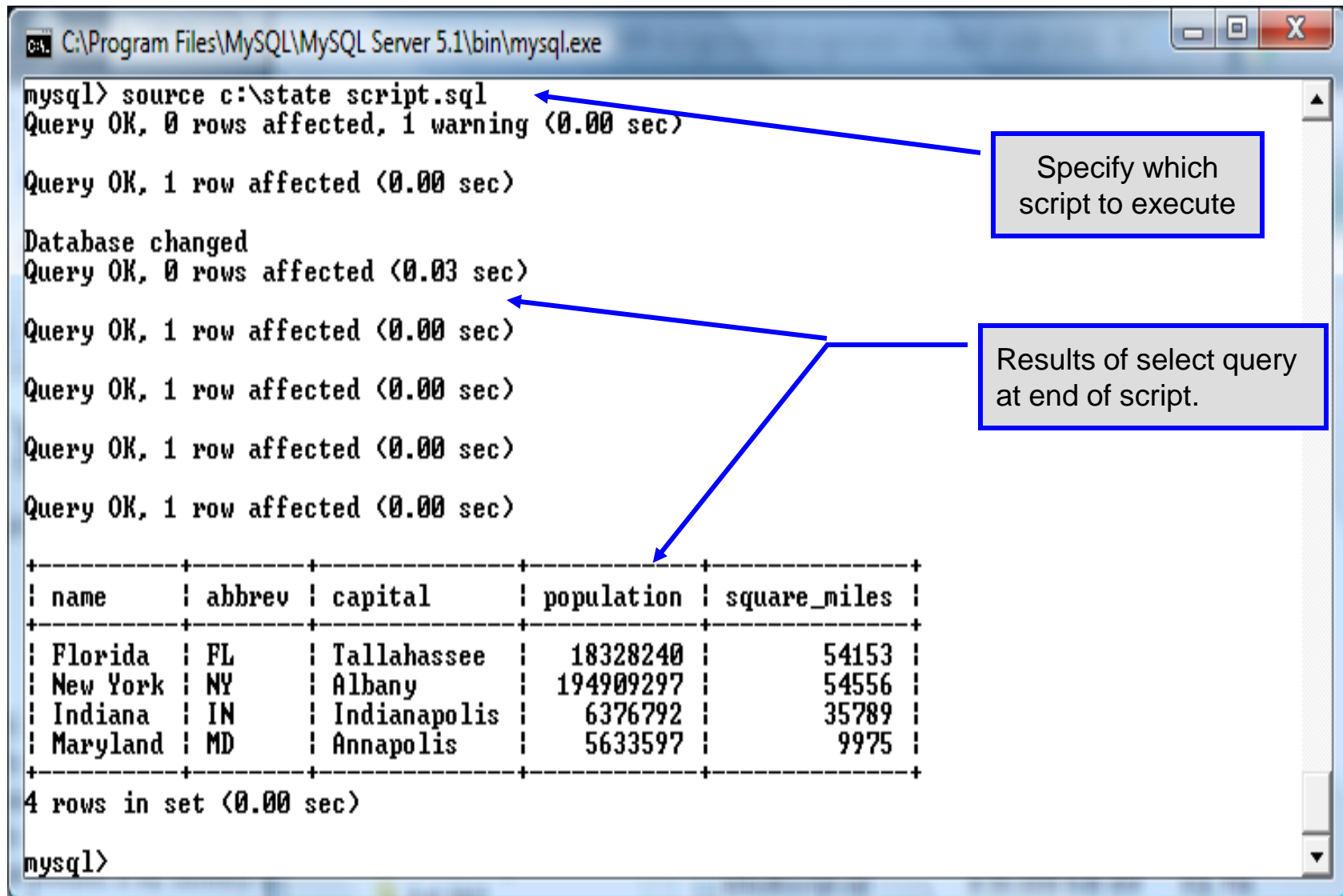
Insert some tuples

Run a simple selection query on the new table.

Structured Query Language file nb char : 616 nb line : 22



# 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:

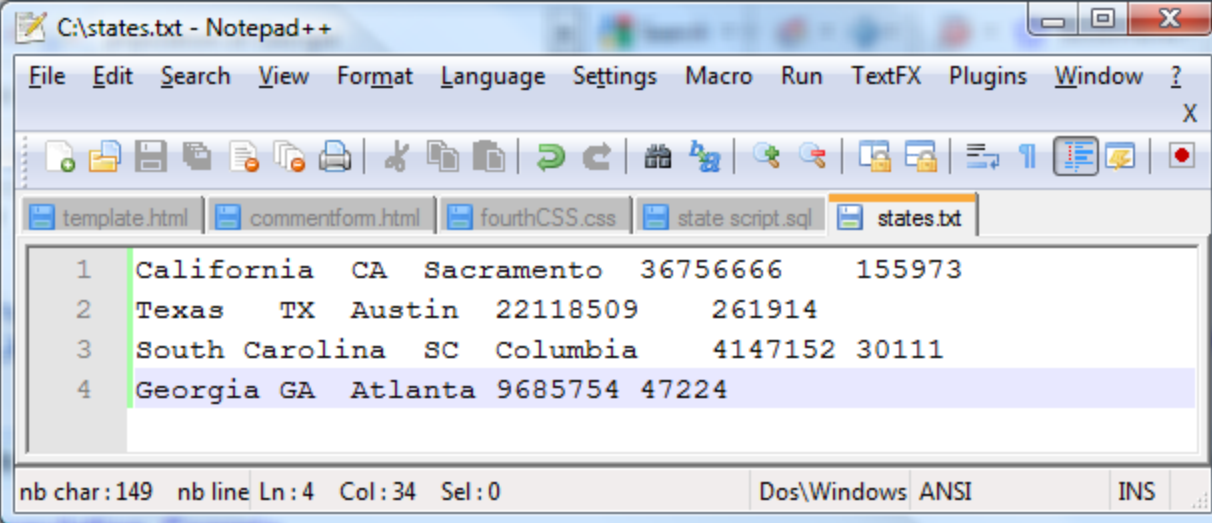
```
mysqlimport [options] database_name file1 file2 ...
```

- 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 `mysqlimport` Utility (cont.)

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



```
1 California CA Sacramento 36756666 155973
2 Texas TX Austin 22118509 261914
3 South Carolina SC Columbia 4147152 30111
4 Georgia GA Atlanta 9685754 47224
```

- 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 `mysqlimport` Utility

Importing a “data file” into a MySQL database table using the `mysqlimport` utility

```
Prompt

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.

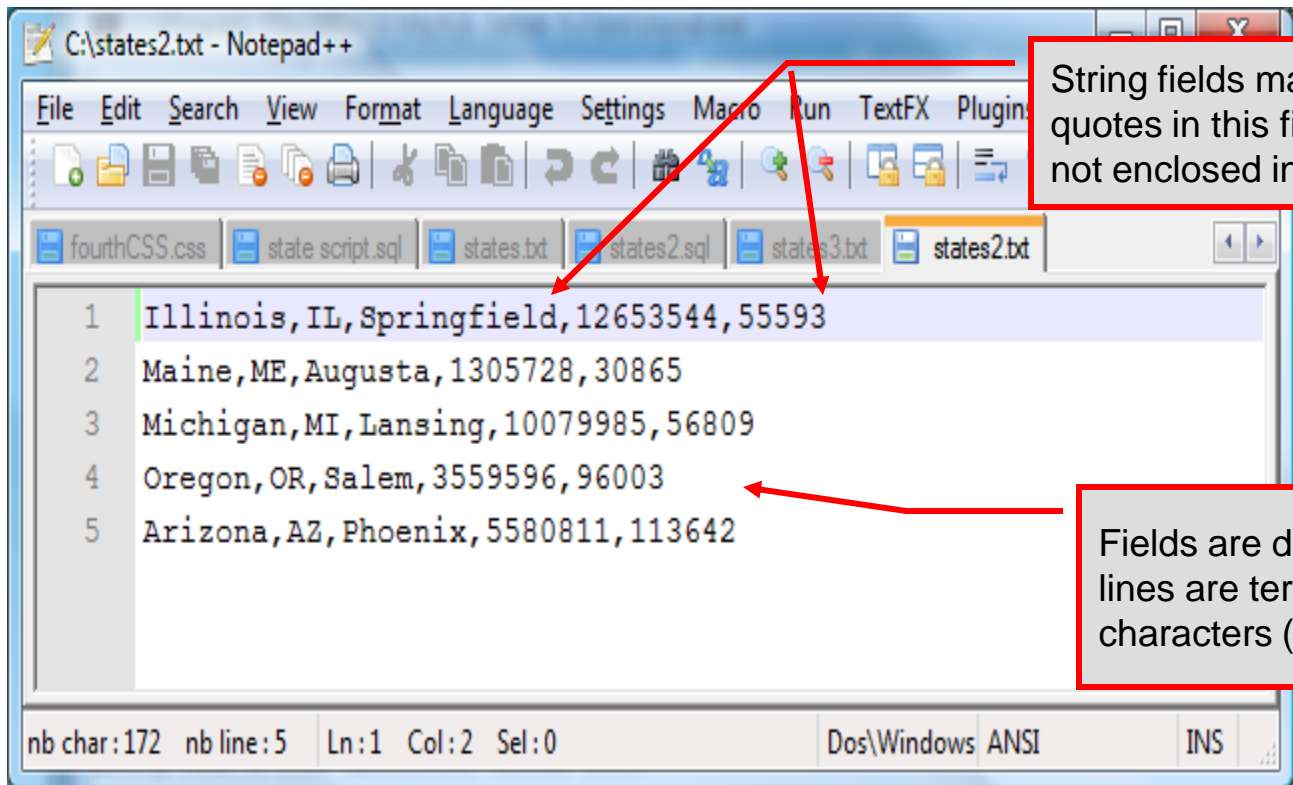
Sets the characters that delimit and enclose the fields and lines in the data file. Similar to `mysqlimport` syntax.

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

Used to load only certain columns (not entire rows)



# Load Data Infile Example



The screenshot shows a Notepad++ window titled 'C:\states2.txt - Notepad++'. The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, TextFX, and Plugin. The toolbar contains various icons for file operations and editing. The tab bar shows several files, with 'states2.txt' selected. The text area contains the following data:

```
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
```

Two callout boxes provide additional information:

- A box at the top right states: "String fields may be enclosed by double quotes in this file. Numeric values are not enclosed in quotes." Red arrows point from this box to the state names and the numeric values in the first line of the data.
- A box at the bottom right states: "Fields are delimited by commas and lines are terminated by newline characters (an invisible \n)". A red arrow points from this box to the comma between 'Salem' and '3559596' in the fourth line.

The status bar at the bottom indicates 'nb char:172 nb line:5 Ln:1 Col:2 Sel:0' and 'Dos\Windows ANSI' encoding.

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



```
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> _
```

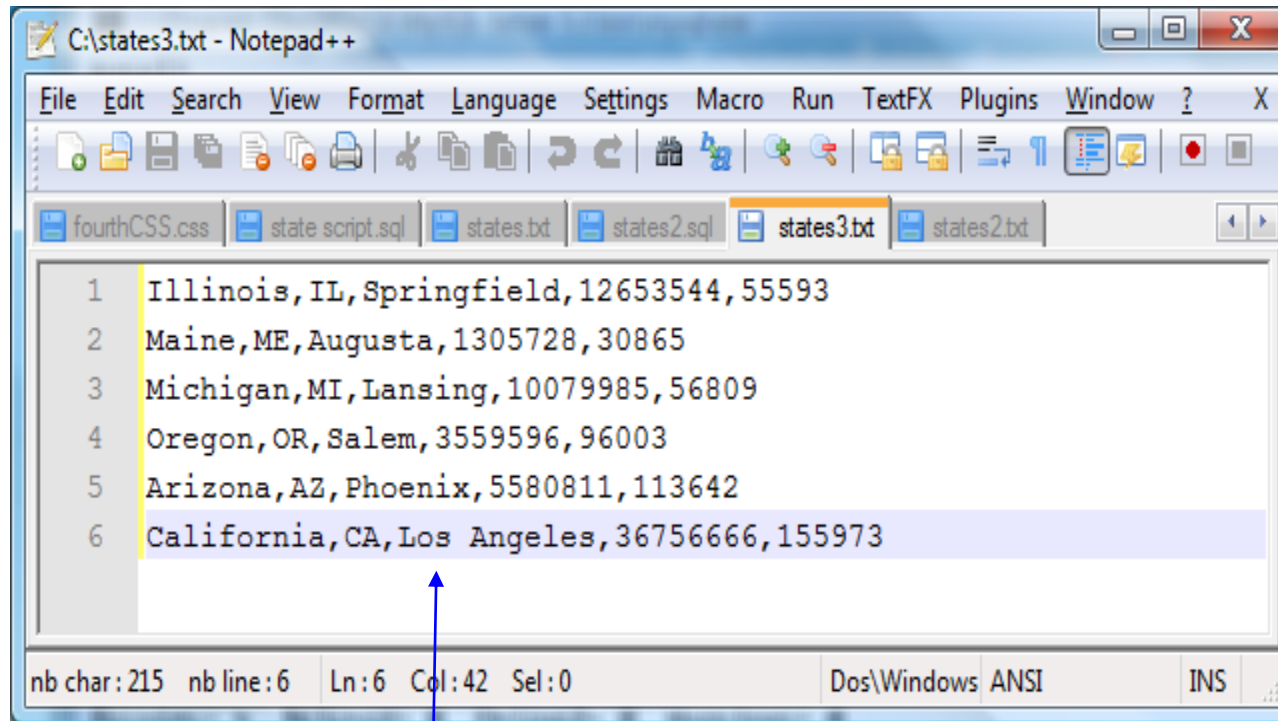
States table **before** addition of data

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



The screenshot shows a Notepad++ window titled 'C:\states3.txt - Notepad++'. The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, TextFX, Plugins, Window, and Help. The toolbar contains various icons for file operations and editing. The tab bar shows several files: fourthCSS.css, state script.sql, states.txt, states2.sql, states3.txt (active), and states2.txt. The text area contains six lines of data, with the last line highlighted in blue. A blue arrow points from the text below to the highlighted line.

```
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
```

nb char: 215 nb line: 6 Ln: 6 Col: 42 Sel: 0 Dos\Windows ANSI INS

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.



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 an 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,...), (...)`

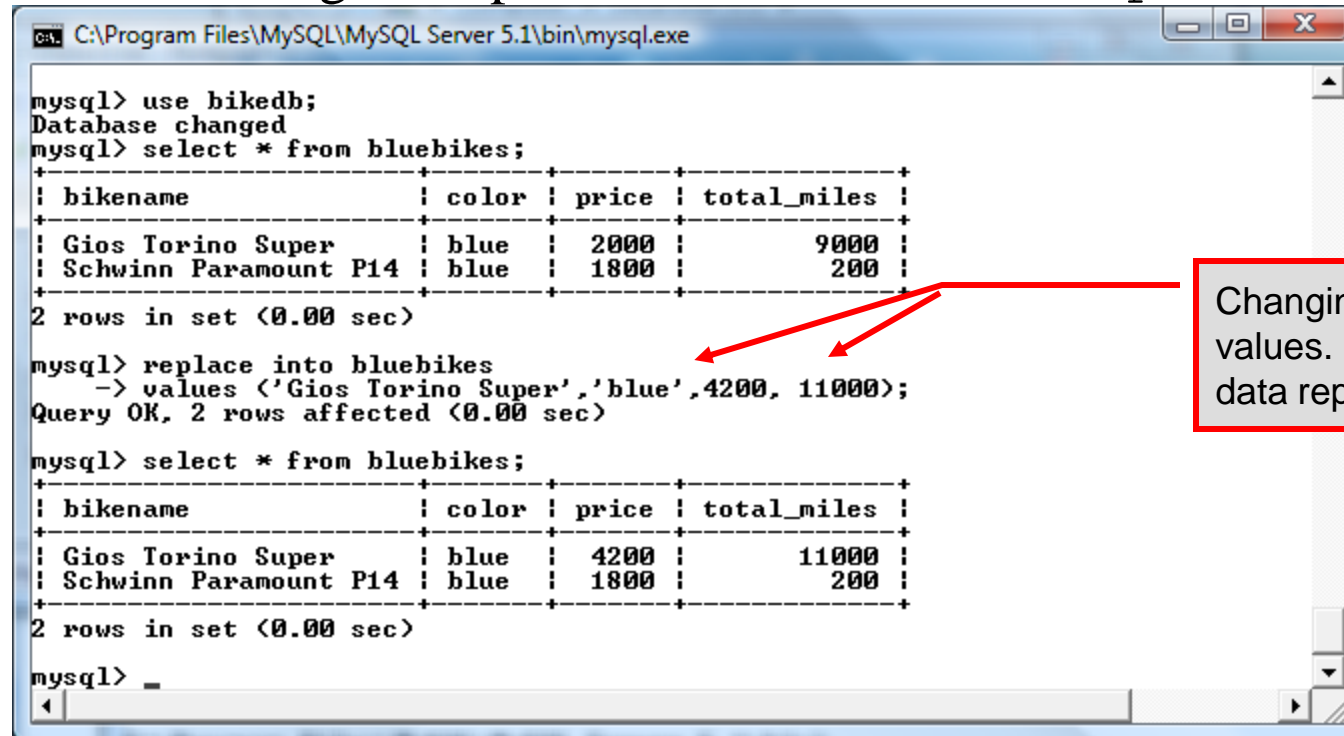
---

**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] [SQL_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

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.

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> _
```



# 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

```
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','Alter') |
| Column_priv | set('Select','Insert','Update','References') |
+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql>
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| tables_priv     |
+-----+

mysql> describe tables_priv;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Host  | char(60) | YES | PRI |          |       |
| Db    | char(64) | YES | PRI |          |       |
| User  | char(16) | YES | PRI |          |       |
| Table_name | char(64) | YES | PRI |          |       |
| Grantor | char(77) | YES | MUL |          |       |
| Timestamp | timestamp | YES |     | CURRENT_TIMESTAMP |       |
| Table_priv | set('Select','Insert','Update','Delete','Create','Drop','Grant','References','Index','Alter') | YES |     |          |       |
| Column_priv | set('Select','Insert','Update','References') | YES |     |          |       |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```



# 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`.`states` TO `mark`@'%' |
+-----+
| GRANT ALL PRIVILEGES ON `hikedb`.`*` 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`.`states` TO `mark`@'%' |
+-----+
| GRANT ALL PRIVILEGES ON `hikedb`.`*` 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.

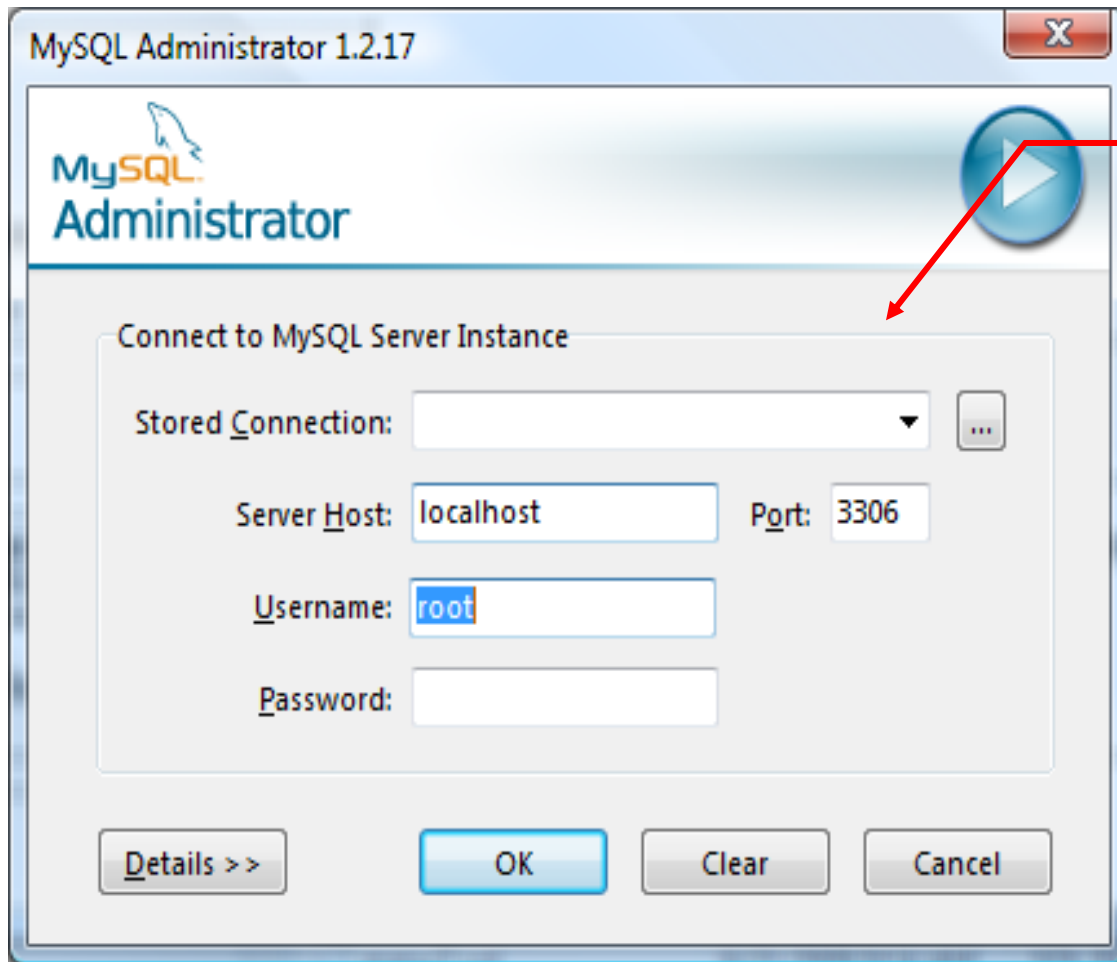


# The MySQL Administrator Tool

- 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.**

MySQL

Connected to MySQL Server Instance

Username:	root
Hostname:	localhost
Port:	3306

Server Information

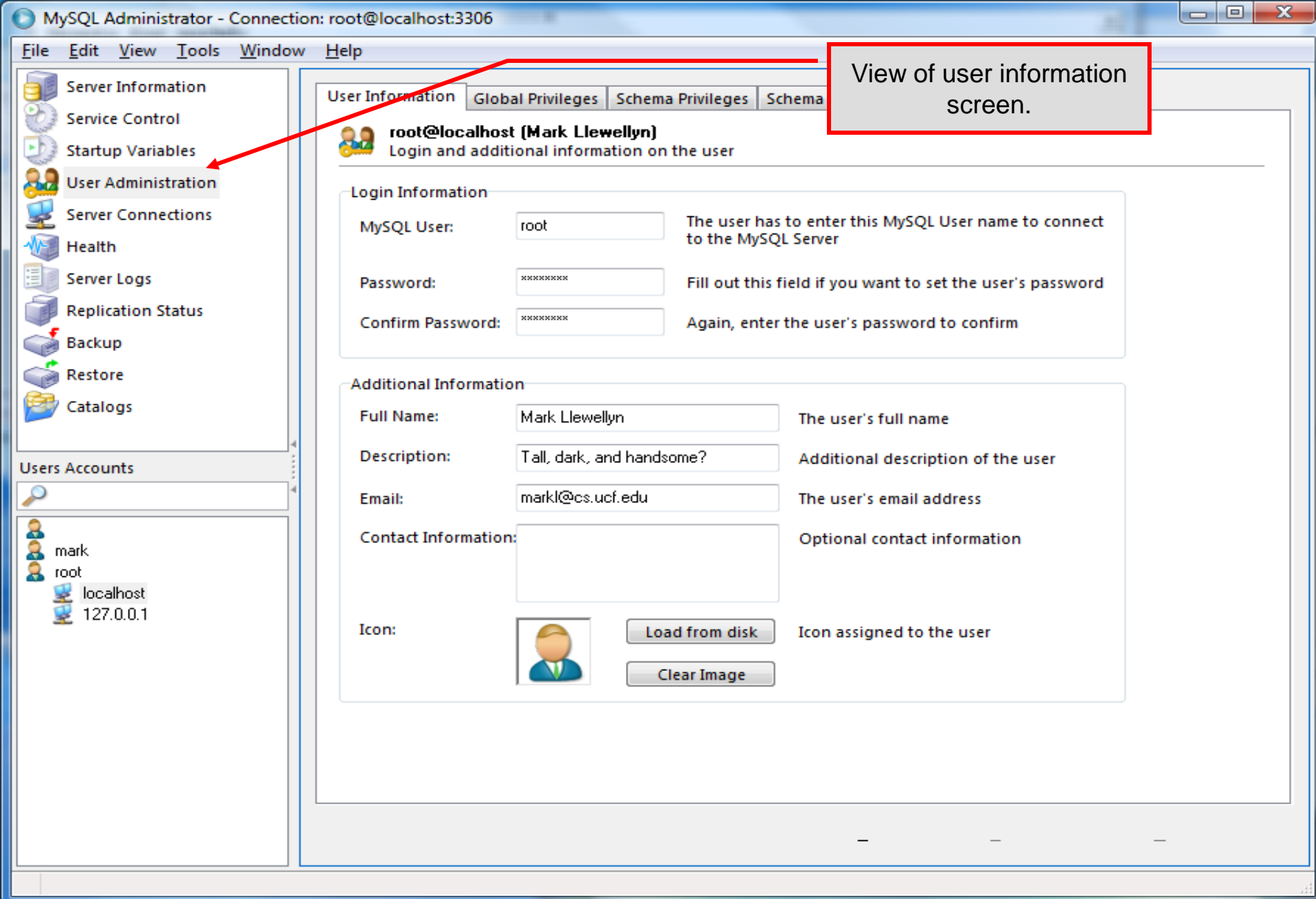
MySQL Version:	MySQL 5.1.38-community via TCP/IP
Network Name:	9VQ2JH1-PC
IP:	127.0.0.1

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

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

View of user privileges for root user on bikedb

Schemata

- bikedb
- information\\_schema
- mysql
- sample
- test
- testdb

Assigned Privileges

Available Privileges

- SELECT Grants the SELECT privilege to t...
- INSERT Grants the INSERT privilege to t...
- UPDATE Grants the UPDATE privilege to ...
- DELETE Grants the DELETE privilege to t...
- 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...
- EVENT Grants the EVENT privilege to th...
- TRIGGER Grants the TRIGGER privilege to...

Add new user 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

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

Schemata

- bikedb
- information\\_schema
- mysql
- 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...
- EVENT Grants the EVENT privilege to th...

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

Add new user 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  
information\_schema  
mysql  
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	2009-09-17 18:46:09
bluebikes	MyISAM	3	112 B	2 kB	2009-09-21 09:38:29
celestebikes	MyISAM	2	88 B	2 kB	2009-09-17 18:52:12

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

Num. of Tables: 3 Rows: 17 Data Len: 768 B Index Len: 6 kB

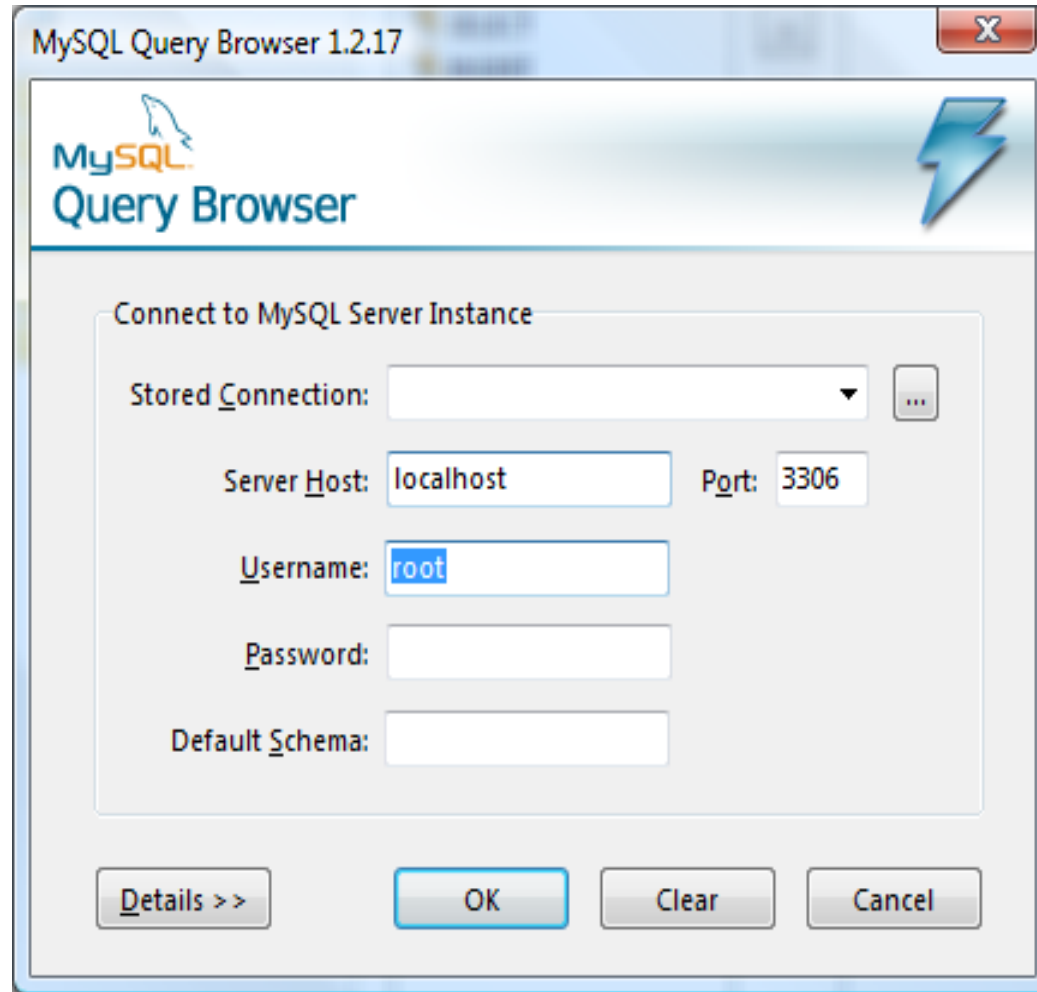


# The MySQL Query Browser Tool

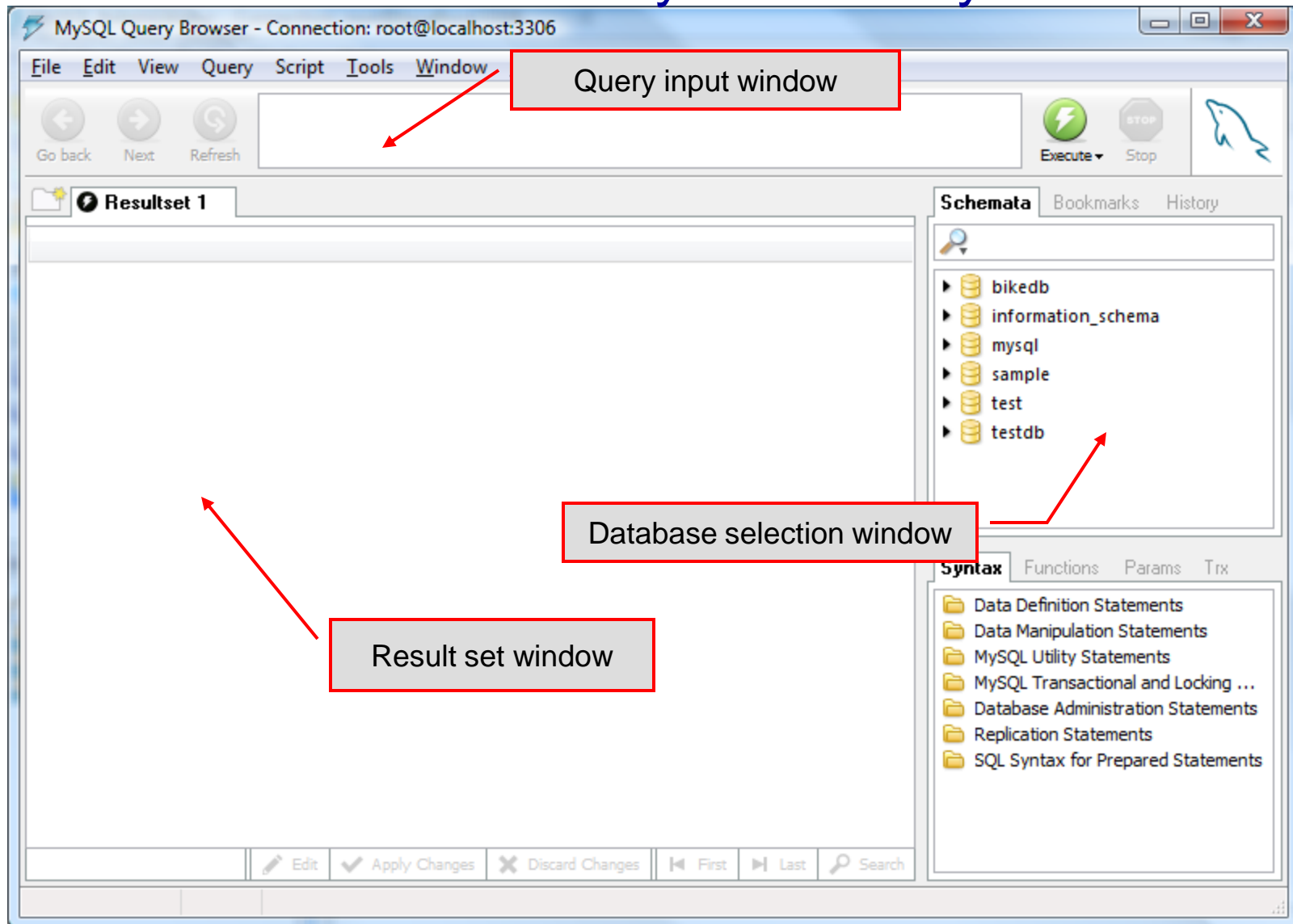
- From MySQL you can also download a GUI-based query browser tool.
- This tool implements all of the basic DML side of SQL with some limitation. For example, editing result sets is possible only if the result set was generated from a single table. Join-based result sets are not editable. This tool also 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





# A First Look At The MySQL Query Browser

The screenshot shows the MySQL Query Browser window. The title bar indicates the connection is 'root@localhost:3306 / bikedb'. The menu bar includes File, Edit, View, Query, Script, Tools, Window, and Help. Below the menu is a toolbar with 'Go back', 'Next', and 'Refresh' buttons. The 'Query input window' contains the SQL query: `select * from bikes`. To the right of the query input are 'Execute' and 'Stop' buttons. Below the query input is the 'Resultset 1' tab, which displays a table of bike data. The table has columns: bikename, size, color, cost, purchased, and mileage. The 'Result set window' is highlighted with a red box. To the right of the result set is the 'Schemata' panel, showing a tree view of databases: bikedb, information\_schema, mysql, sample, test, and testdb. Below the schemata panel is the 'Syntax' panel, showing 'Data Definition Statements' and 'Data Manipulation Statements'. At the bottom of the window is a status bar showing '12 rows fetched in 0.0038s (0.0)' and buttons for 'Edit', 'Apply Changes', 'Discard Changes', 'First', 'Last', and 'Search'.

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

File Edit View Query Script Tools Window Help

Go back Next Refresh

Query input window

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

Schemata Bookmarks History

- bikedb
- information\_schema
- mysql
- sample
- test
- testdb

Syntax Functions Params Trx

- Data Definition Statements
- Data Manipulation Statements

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.

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

1: 1



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 x

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

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.

