COP 4710: Database Systems
Spring 2007

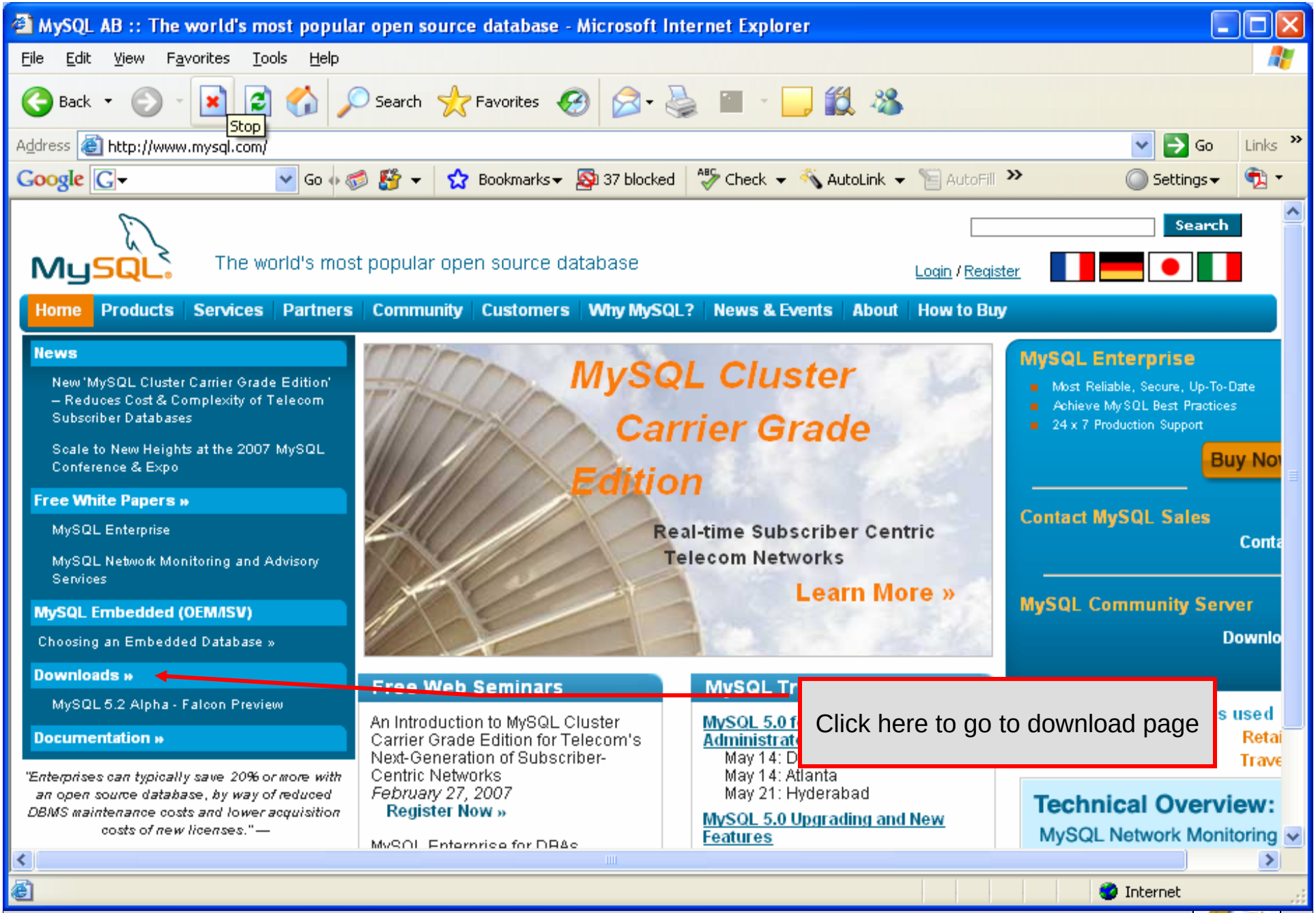
Introduction To MySQL

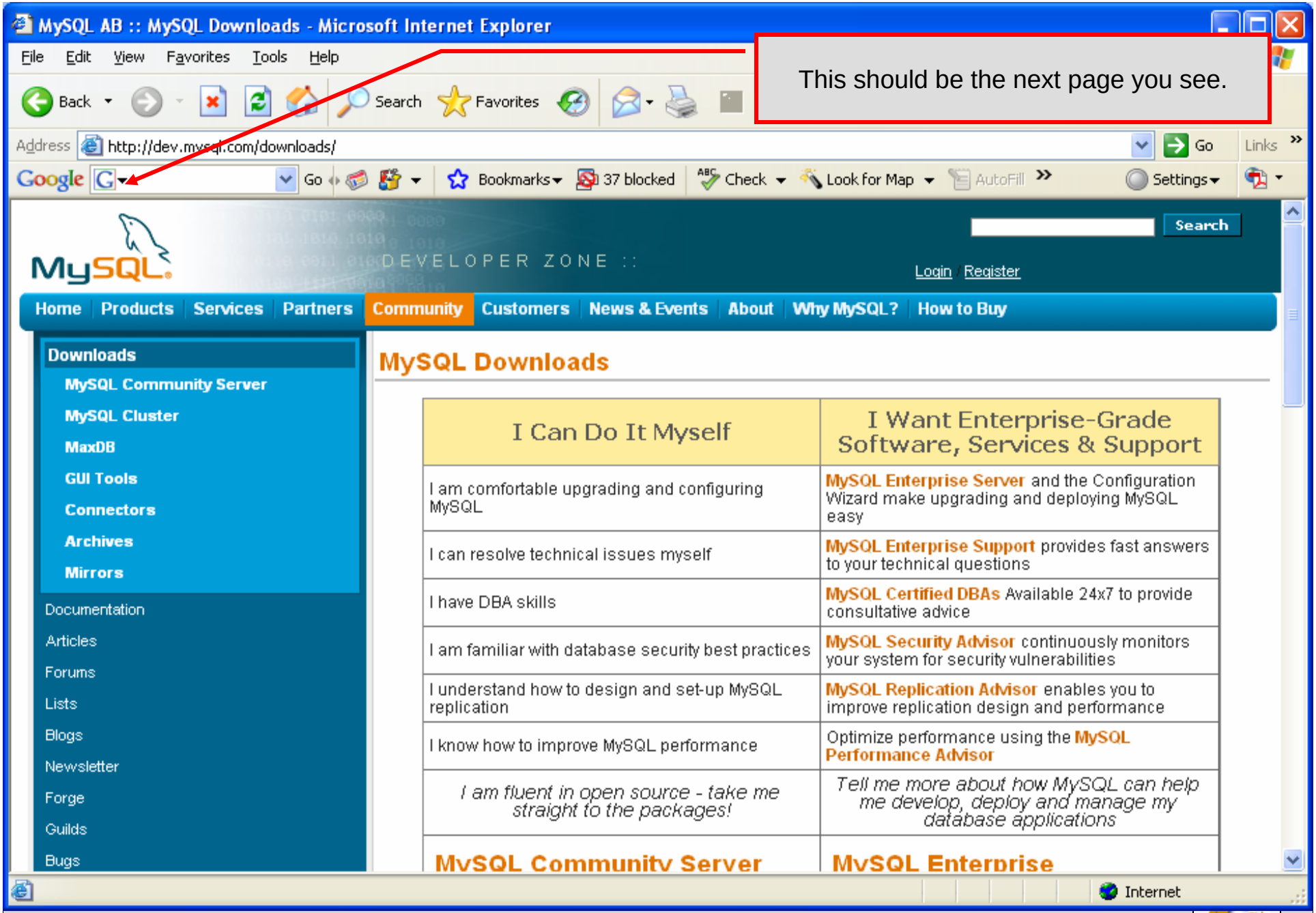
Instructor : Mark Llewellyn
markl@cs.ucf.edu
ENG3 236, 823-2790
http://www.cs.ucf.edu/courses/cop4710/spr2007

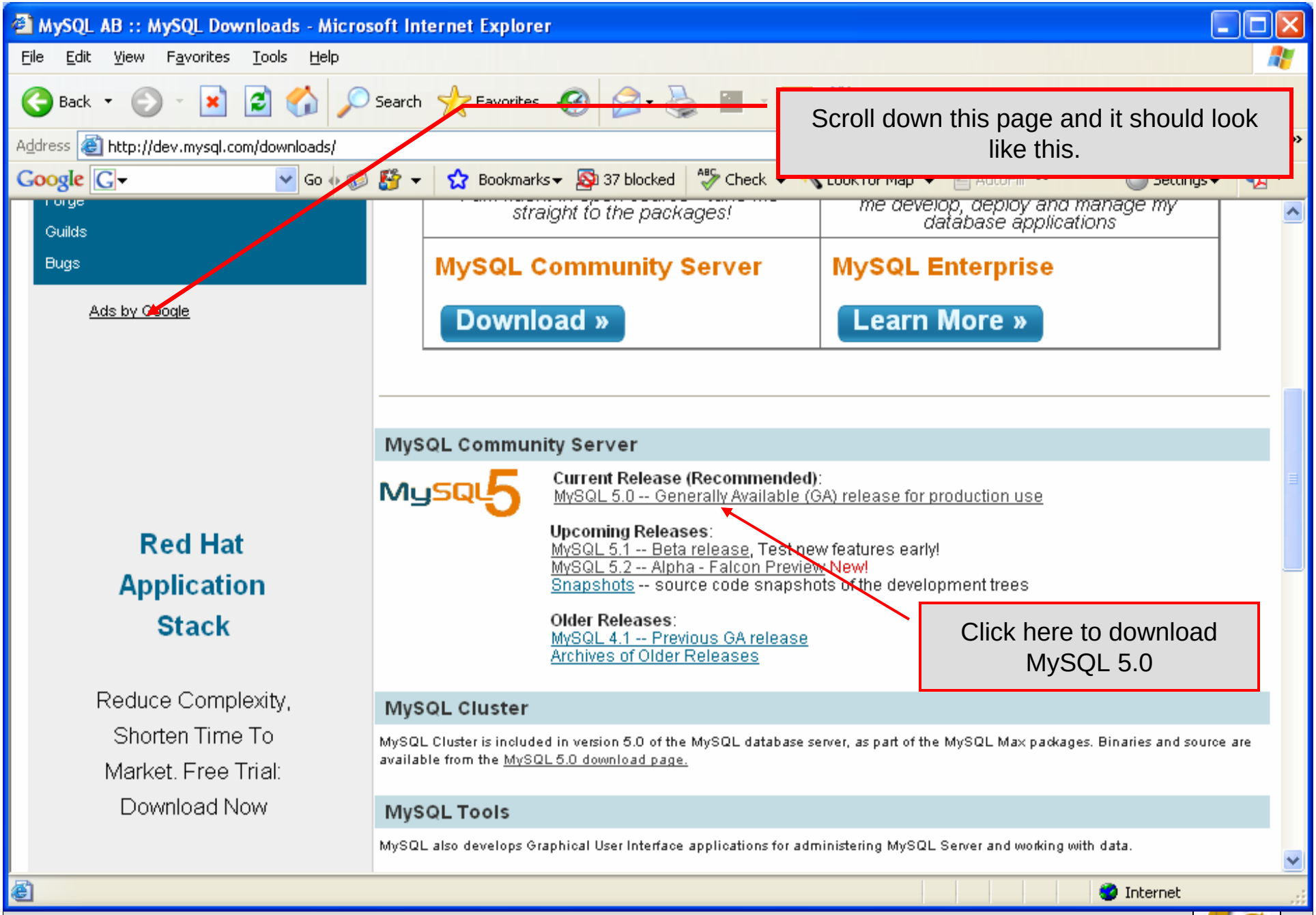
School of Electrical Engineering and Computer Science
University of Central Florida

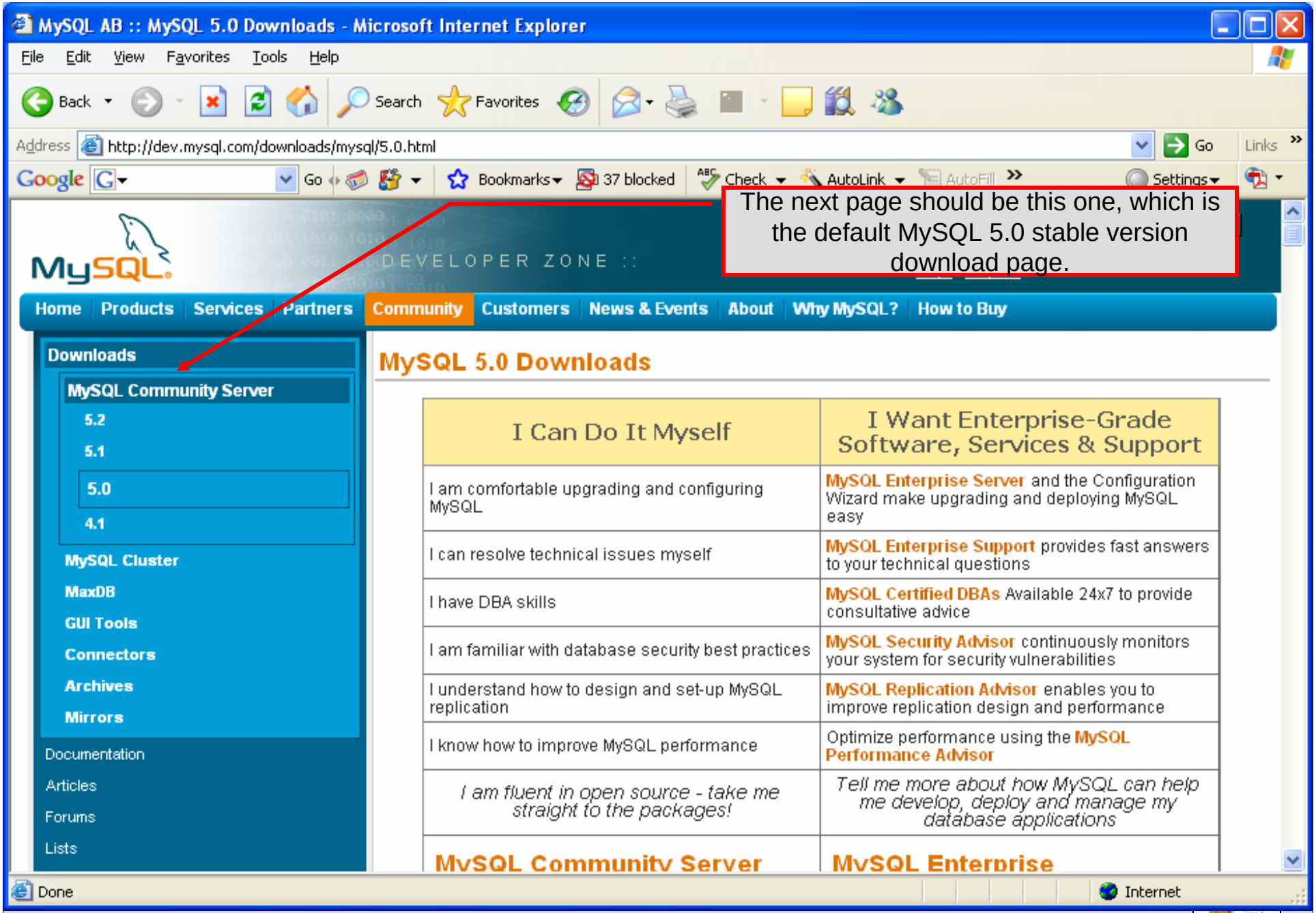
MySQL RDBMS

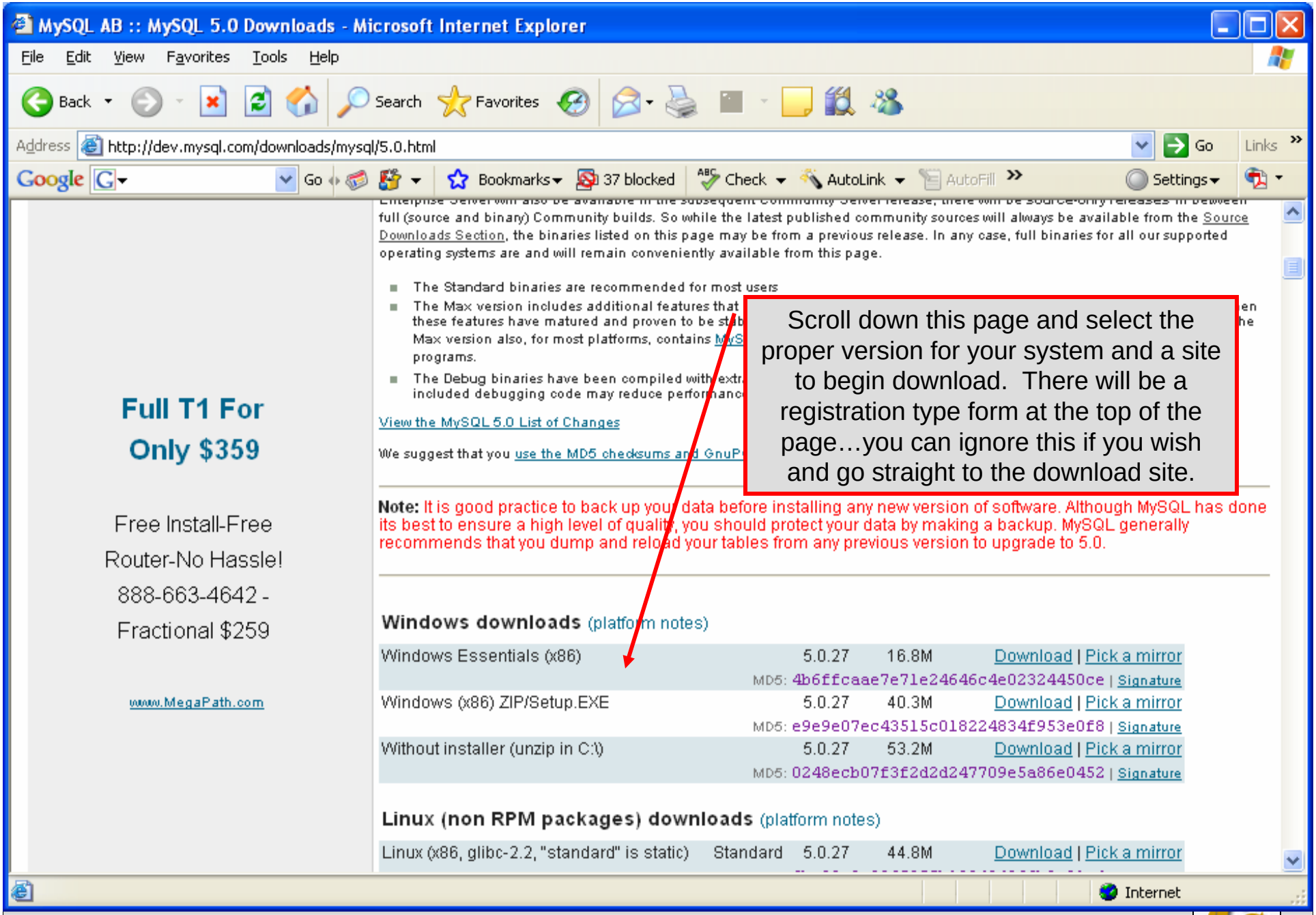
- MySQL is a database server (although it does come with a set of simple client programs). The current stable version is 5.0.27 and can be downloaded from www.mysql.com. (Any of the versions of MySQL 5.0.15 or greater will be fine for our purposes.)
- It is typically used in thin client environments. 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.

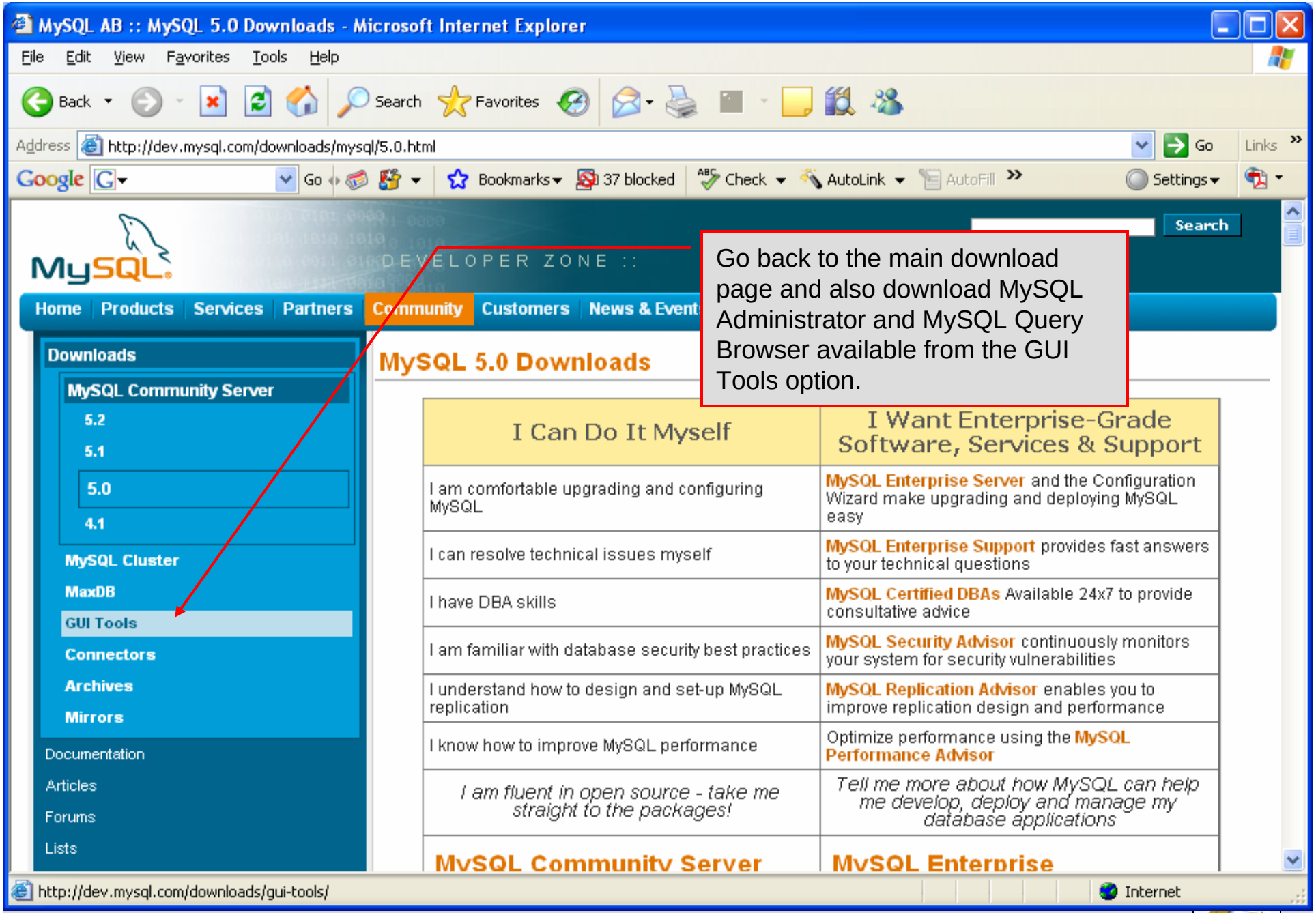

MySQL AB :: The world's most popular open source database - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Stop
Address http://www.mysql.com/
Go Links
Google Go Bookmarks 37 blocked Check AutoLink AutoFill Settings
MySQL
The world's most popular open source database
Search
Login / Register
Home Products Services Partners Community Customers Why MySQL? News & Events About How to Buy
News
New 'MySQL Cluster Carrier Grade Edition' – Reduces Cost & Complexity of Telecom Subscriber Databases
Scale to New Heights at the 2007 MySQL Conference & Expo
Free White Papers »
MySQL Enterprise
MySQL Network Monitoring and Advisory Services
MySQL Embedded (OEM/ISV)
Choosing an Embedded Database »
Downloads »
MySQL 5.2 Alpha - Falcon Preview
Documentation »
"Enterprises can typically save 20% or more with an open source database, by way of reduced DBMS maintenance costs and lower acquisition costs of new licenses."—
MySQL Cluster Carrier Grade Edition
Real-time Subscriber Centric Telecom Networks
Learn More »
Free Web Seminars
An Introduction to MySQL Cluster Carrier Grade Edition for Telecom's Next-Generation of Subscriber-Centric Networks
February 27, 2007
Register Now »
MySQL Tr
MySQL 5.0 f
Administrat
May 14: D
May 14: Atlanta
May 21: Hyderabad
MySQL 5.0 Upgrading and New Features
MySQL Enterprise
Most Reliable, Secure, Up-To-Date
Achieve MySQL Best Practices
24 x 7 Production Support
Buy Now
Contact MySQL Sales
Conta
MySQL Community Server
Downlo
s used
Retai
Trave
Technical Overview:
MySQL Network Monitoring
Click here to go to download page
Internet

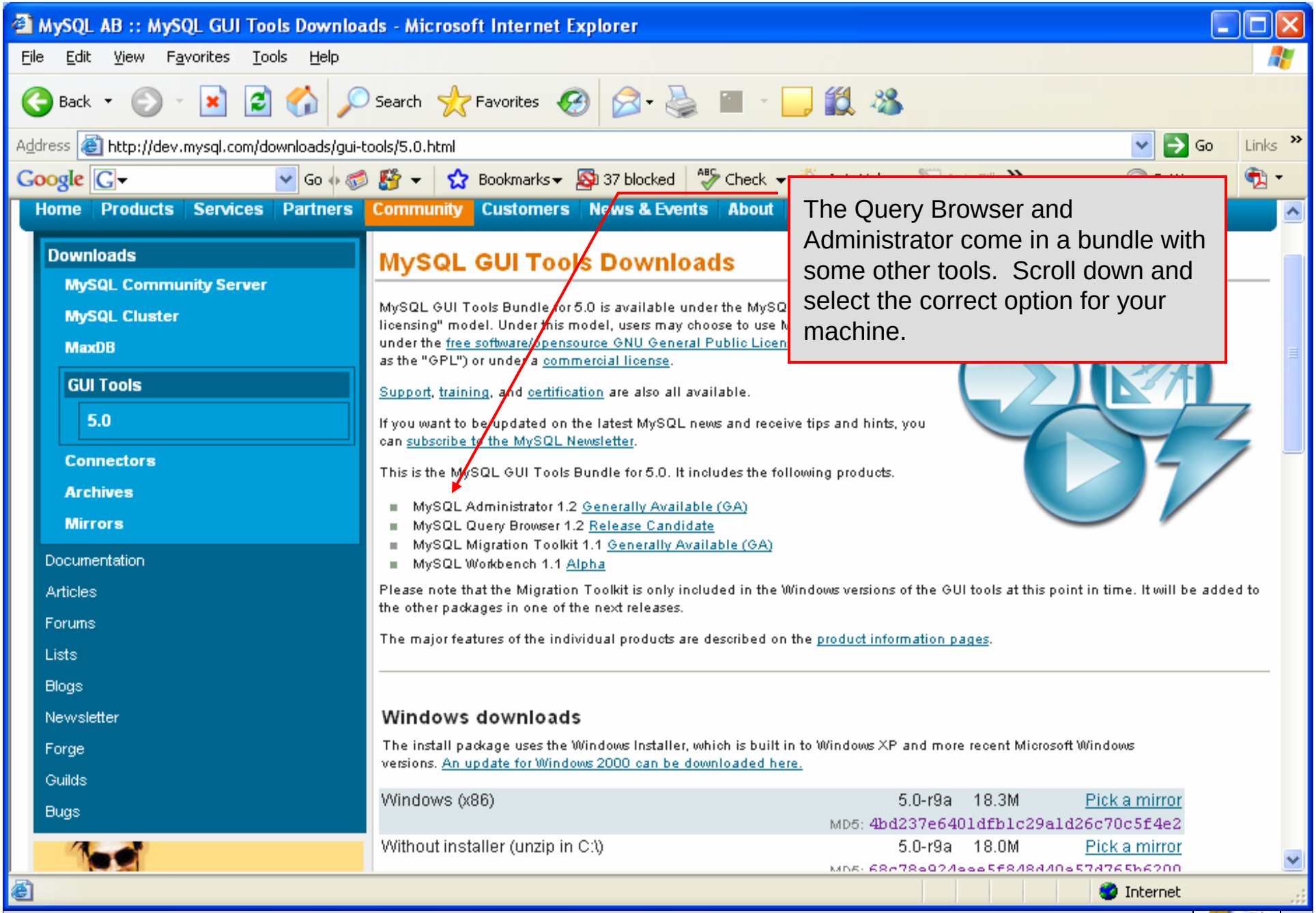

MySQL AB :: MySQL Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
This should be the next page you see.
Address http://dev.mysql.com/downloads/
Go Links
Google Go Bookmarks 37 blocked Check Look for Map AutoFill Settings
MySQL
DEVELOPER ZONE ::
Search
Login | Register
Home Products Services Partners Community Customers News & Events About Why MySQL? How to Buy
Downloads
MySQL Community Server
MySQL Cluster
MaxDB
GUI Tools
Connectors
Archives
Mirrors
Documentation
Articles
Forums
Lists
Blogs
Newsletter
Forge
Guilds
Bugs
MySQL Downloads
I Can Do It Myself
I Want Enterprise-Grade Software, Services & Support
I am comfortable upgrading and configuring MySQL
MySQL Enterprise Server and the Configuration Wizard make upgrading and deploying MySQL easy
I can resolve technical issues myself
MySQL Enterprise Support provides fast answers to your technical questions
I have DBA skills
MySQL Certified DBAs Available 24x7 to provide consultative advice
I am familiar with database security best practices
MySQL Security Advisor continuously monitors your system for security vulnerabilities
I understand how to design and set-up MySQL replication
MySQL Replication Advisor enables you to improve replication design and performance
I know how to improve MySQL performance
Optimize performance using the MySQL Performance Advisor
I am fluent in open source - take me straight to the packages!
Tell me more about how MySQL can help me develop, deploy and manage my database applications
MySQL Community Server
MySQL Enterprise
Internet


MySQL AB :: MySQL Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://dev.mysql.com/downloads/
Google Go Bookmarks 37 blocked Check
Scroll down this page and it should look like this.
Guilds
Bugs
Ads by Google
straight to the packages!
me develop, deploy and manage my database applications
MySQL Community Server
Download »
MySQL Enterprise
Learn More »
MySQL Community Server
Current Release (Recommended):
MySQL 5.0 -- Generally Available (GA) release for production use
Upcoming Releases:
MySQL 5.1 -- Beta release, Test new features early!
MySQL 5.2 -- Alpha - Falcon Preview New!
Snapshots -- source code snapshots of the development trees
Older Releases:
MySQL 4.1 -- Previous GA release
Archives of Older Releases
Click here to download MySQL 5.0
Red Hat Application Stack
Reduce Complexity, Shorten Time To Market. Free Trial: Download Now
MySQL Cluster
MySQL Cluster is included in version 5.0 of the MySQL database server, as part of the MySQL Max packages. Binaries and source are available from the MySQL 5.0 download page.
MySQL Tools
MySQL also develops Graphical User Interface applications for administering MySQL Server and working with data.
Internet

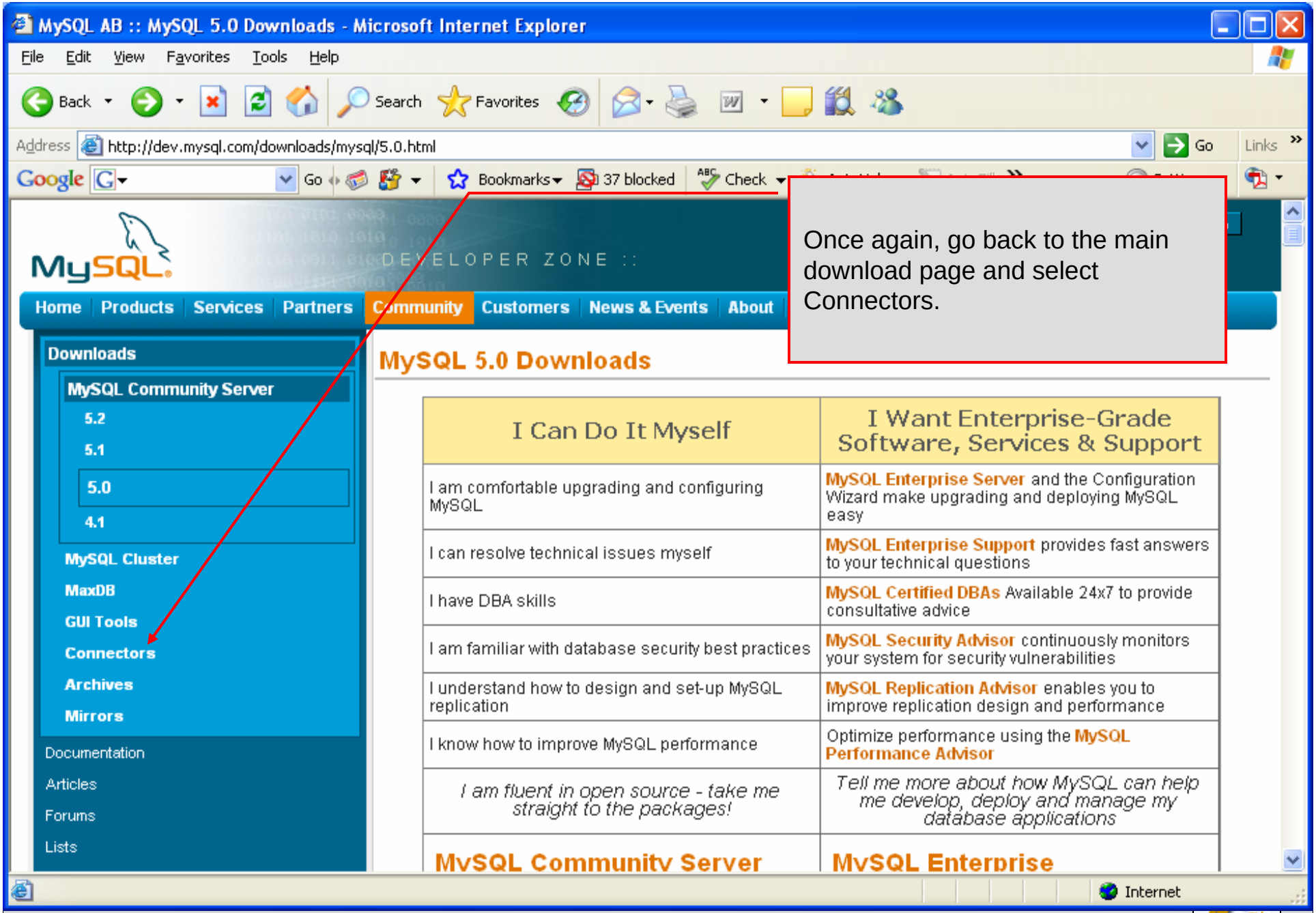

MySQL AB :: MySQL 5.0 Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://dev.mysql.com/downloads/mysql/5.0.html
Go Links
Google Go Bookmarks 37 blocked Check AutoLink AutoFill Settings
The next page should be this one, which is the default MySQL 5.0 stable version download page.
MySQL DEVELOPER ZONE ::
Home Products Services Partners Community Customers News & Events About Why MySQL? How to Buy
Downloads
MySQL Community Server
5.2
5.1
5.0
4.1
MySQL Cluster
MaxDB
GUI Tools
Connectors
Archives
Mirrors
Documentation
Articles
Forums
Lists
MySQL 5.0 Downloads
I Can Do It Myself
I Want Enterprise-Grade Software, Services & Support
I am comfortable upgrading and configuring MySQL
MySQL Enterprise Server and the Configuration Wizard make upgrading and deploying MySQL easy
I can resolve technical issues myself
MySQL Enterprise Support provides fast answers to your technical questions
I have DBA skills
MySQL Certified DBAs Available 24x7 to provide consultative advice
I am familiar with database security best practices
MySQL Security Advisor continuously monitors your system for security vulnerabilities
I understand how to design and set-up MySQL replication
MySQL Replication Advisor enables you to improve replication design and performance
I know how to improve MySQL performance
Optimize performance using the MySQL Performance Advisor
I am fluent in open source - take me straight to the packages!
Tell me more about how MySQL can help me develop, deploy and manage my database applications
MySQL Community Server
MySQL Enterprise
Done
Internet


MySQL AB :: MySQL 5.0 Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://dev.mysql.com/downloads/mysql/5.0.html
Go Links
Google Go Bookmarks 37 blocked Check AutoLink AutoFill Settings
full (source and binary) Community builds. So while the latest published community sources will always be available from the Source Downloads Section, the binaries listed on this page may be from a previous release. In any case, full binaries for all our supported operating systems are and will remain conveniently available from this page.
The Standard binaries are recommended for most users
The Max version includes additional features that
these features have matured and proven to be
Max version also, for most platforms, contains
programs.
The Debug binaries have been compiled with
included debugging code may reduce
View the MySQL 5.0 List of Changes
We suggest that you use the MD5 checksums and
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.
Note: It is good practice to back up your data before installing any new version of software. Although MySQL has done its best to ensure a high level of quality, you should protect your data by making a backup. MySQL generally recommends that you dump and reload your tables from any previous version to upgrade to 5.0.
Windows downloads (platform notes)
Windows Essentials (x86) 5.0.27 16.8M Download | Pick a mirror
MD5: 4b6ffcaae7e71e24646c4e02324450ce | Signature
Windows (x86) ZIP/Setup.EXE 5.0.27 40.3M Download | Pick a mirror
MD5: e9e9e07ec43515c018224834f953e0f8 | Signature
Without installer (unzip in C:\) 5.0.27 53.2M Download | Pick a mirror
MD5: 0248ecb07f3f2d2d247709e5a86e0452 | Signature
Linux (non RPM packages) downloads (platform notes)
Linux (x86, glibc-2.2, "standard" is static) Standard 5.0.27 44.8M Download | Pick a mirror
Full T1 For Only $359
Free Install-Free Router-No Hassle!
888-663-4642 - Fractional $259
www.MegaPath.com
Internet

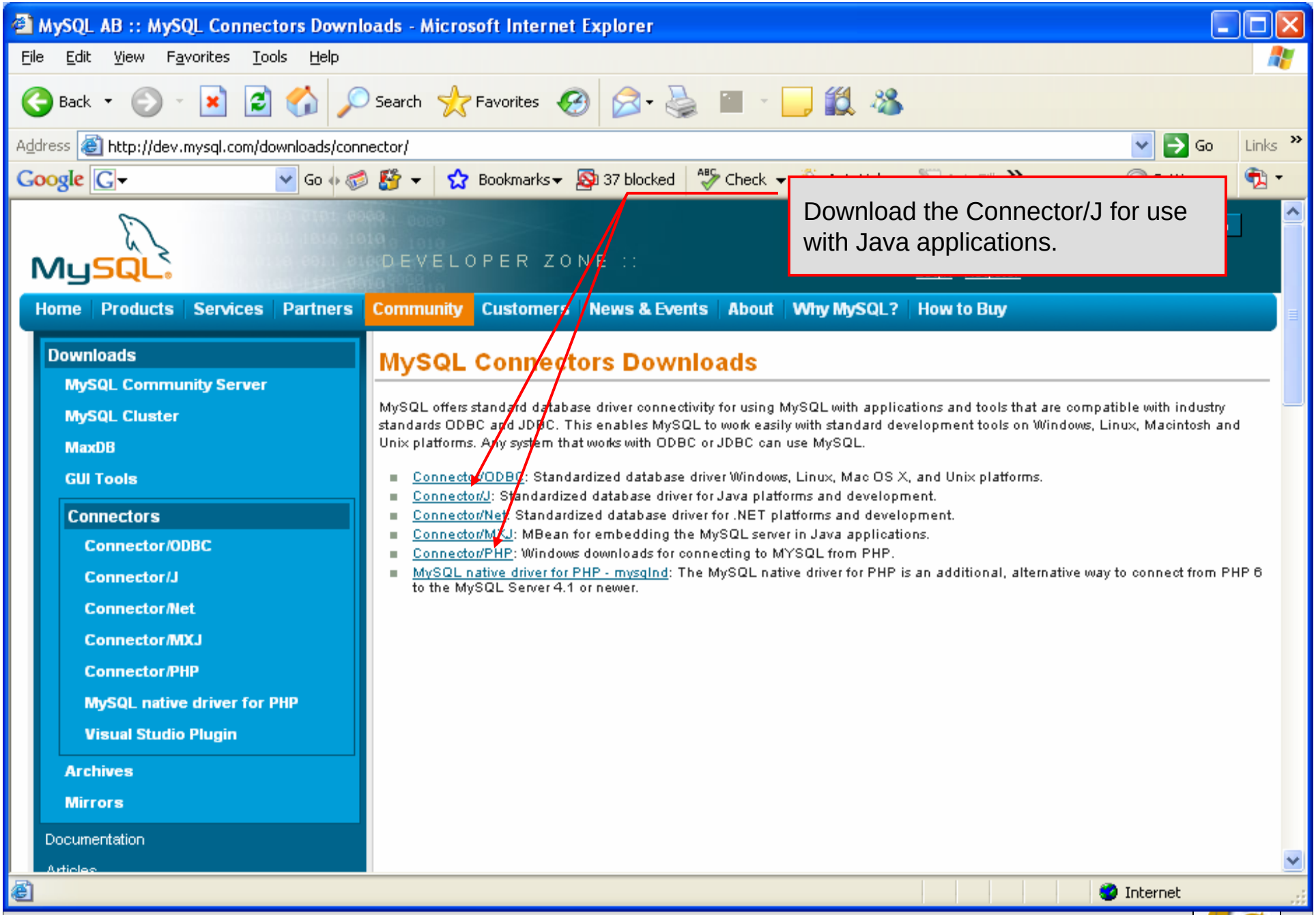

MySQL AB :: MySQL 5.0 Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://dev.mysql.com/downloads/mysql/5.0.html
Go Links
Google Go Bookmarks 37 blocked Check AutoLink AutoFill Settings
MySQL
DEVELOPER ZONE ::
Search
Home Products Services Partners Community Customers News & Event
Downloads
MySQL Community Server
5.2
5.1
5.0
4.1
MySQL Cluster
MaxDB
GUI Tools
Connectors
Archives
Mirrors
Documentation
Articles
Forums
Lists
MySQL 5.0 Downloads
Go back to the main download page and also download MySQL Administrator and MySQL Query Browser available from the GUI Tools option.
I Can Do It Myself
I Want Enterprise-Grade Software, Services & Support
I am comfortable upgrading and configuring MySQL
MySQL Enterprise Server and the Configuration Wizard make upgrading and deploying MySQL easy
I can resolve technical issues myself
MySQL Enterprise Support provides fast answers to your technical questions
I have DBA skills
MySQL Certified DBAs Available 24x7 to provide consultative advice
I am familiar with database security best practices
MySQL Security Advisor continuously monitors your system for security vulnerabilities
I understand how to design and set-up MySQL replication
MySQL Replication Advisor enables you to improve replication design and performance
I know how to improve MySQL performance
Optimize performance using the MySQL Performance Advisor
I am fluent in open source - take me straight to the packages!
Tell me more about how MySQL can help me develop, deploy and manage my database applications
MySQL Community Server
MySQL Enterprise
http://dev.mysql.com/downloads/gui-tools/
Internet



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



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

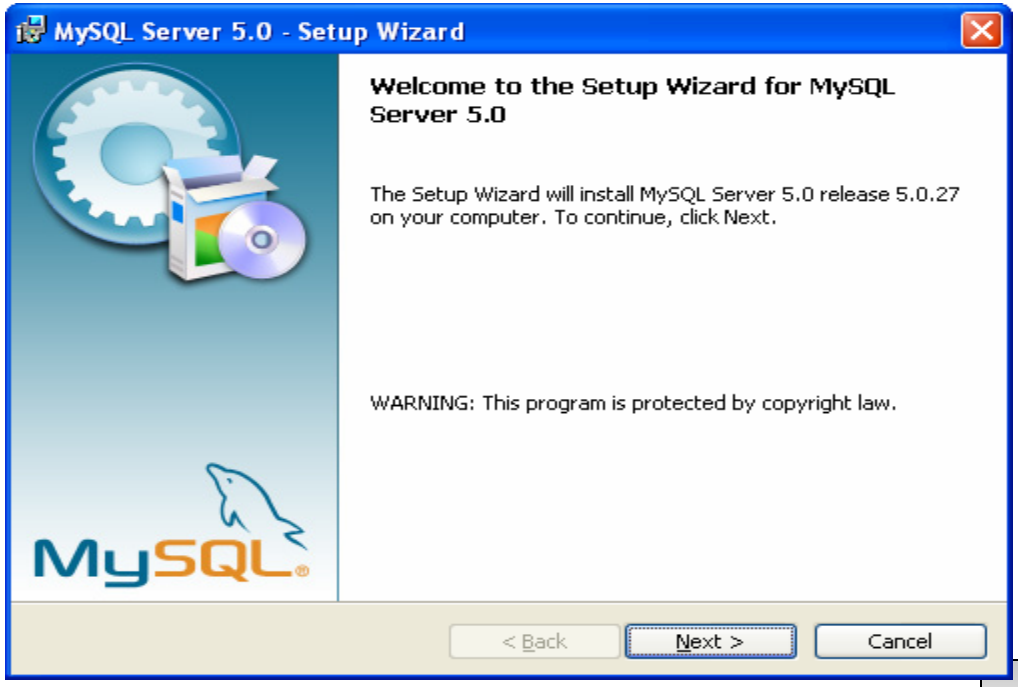

MySQL AB :: MySQL Connectors Downloads - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://dev.mysql.com/downloads/connector/
Go Links
Google Go Bookmarks 37 blocked Check
Download the Connector/J for use with Java applications.
MySQL
DEVELOPER ZONE ::
Home Products Services Partners Community Customers News & Events About Why MySQL? How to Buy
Downloads
MySQL Community Server
MySQL Cluster
MaxDB
GUI Tools
Connectors
Connector/ODBC
Connector/J
Connector/Net
Connector/MXJ
Connector/PHP
MySQL native driver for PHP
Visual Studio Plugin
Archives
Mirrors
Documentation
MySQL Connectors Downloads
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.
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/PHP: Windows downloads for connecting to MYSQL from PHP.
MySQL native driver for PHP - mysqlnd: The MySQL native driver for PHP is an additional, alternative way to connect from PHP 6 to the MySQL Server 4.1 or newer.
Internet

Installing MySQL 5.0.27

- 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.

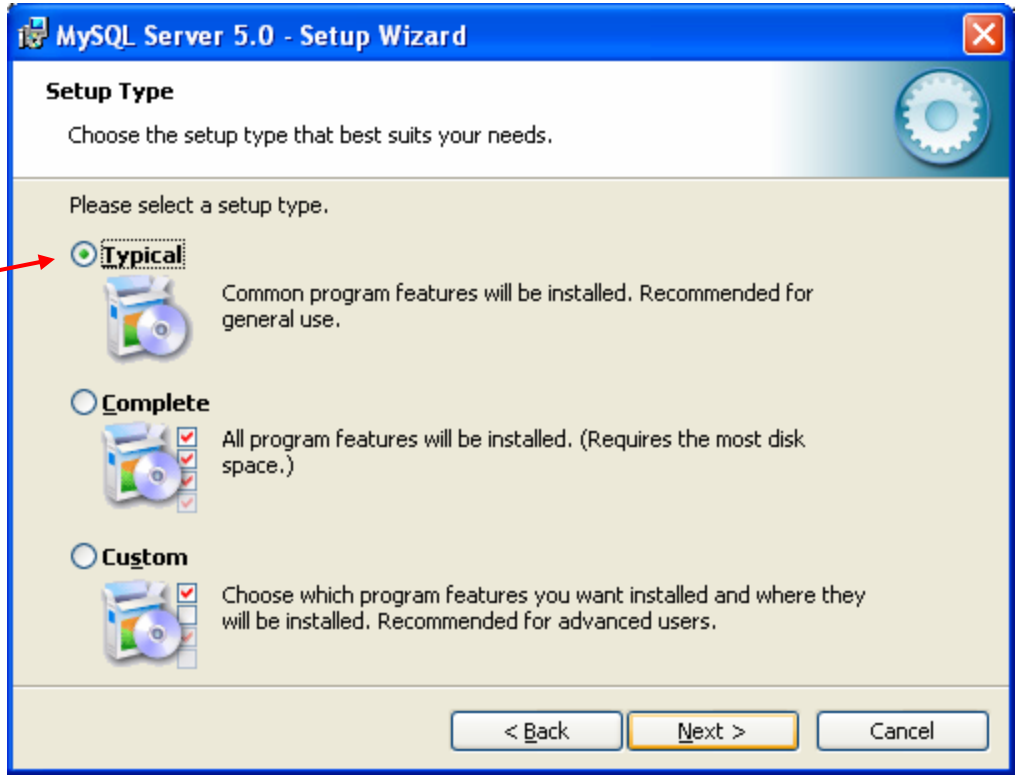
Installing MySQL 5.0.27

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

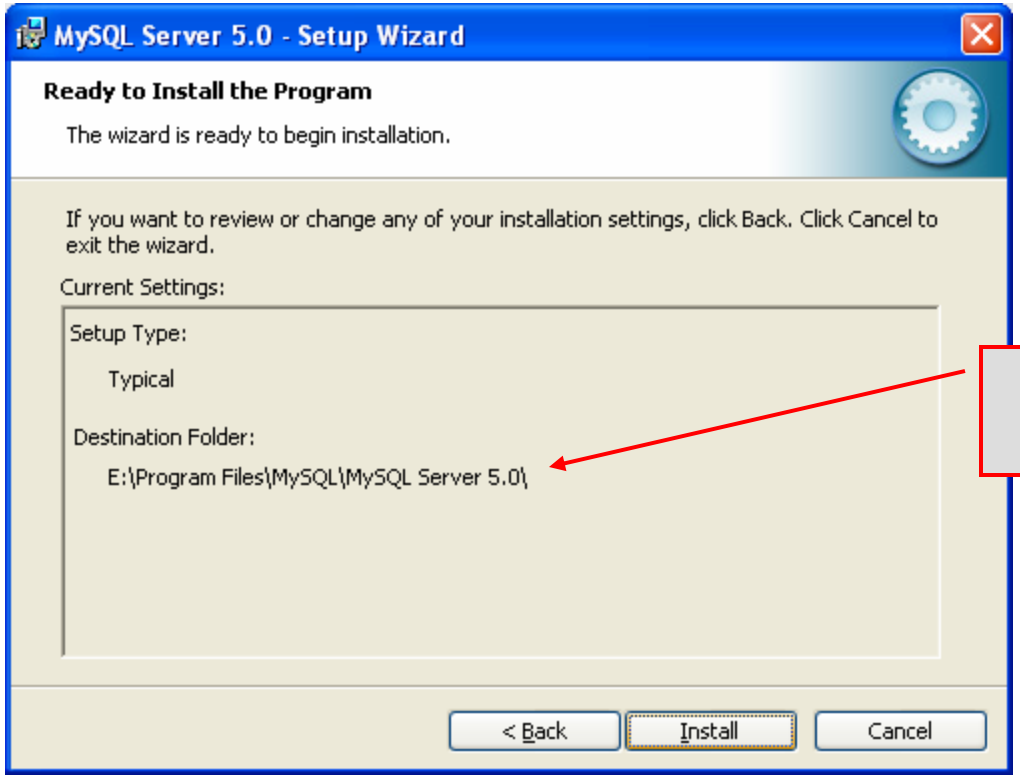


Installing MySQL 5.0 (cont.)

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



Installing MySQL 5.0 (cont.)



Select the destination folder for the install.

Installing MySQL 5.0 (cont.)

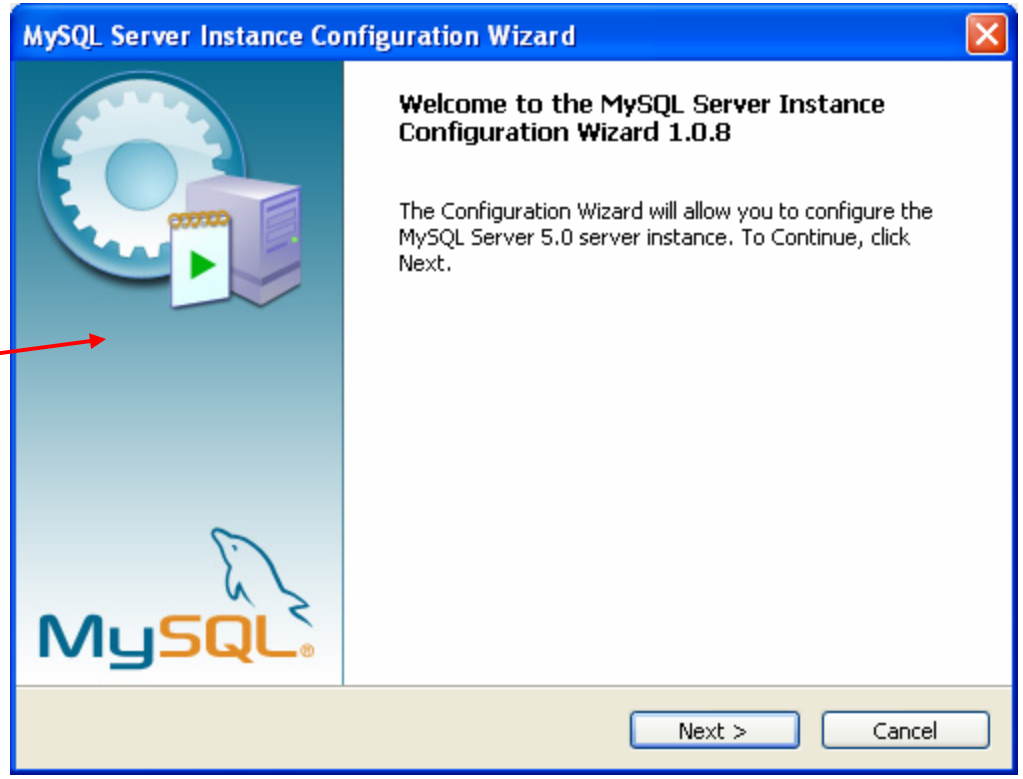
Again, your choice here. If you want to skip the sign-up that's fine.

Installing MySQL 5.0 (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 5.0 (cont.)

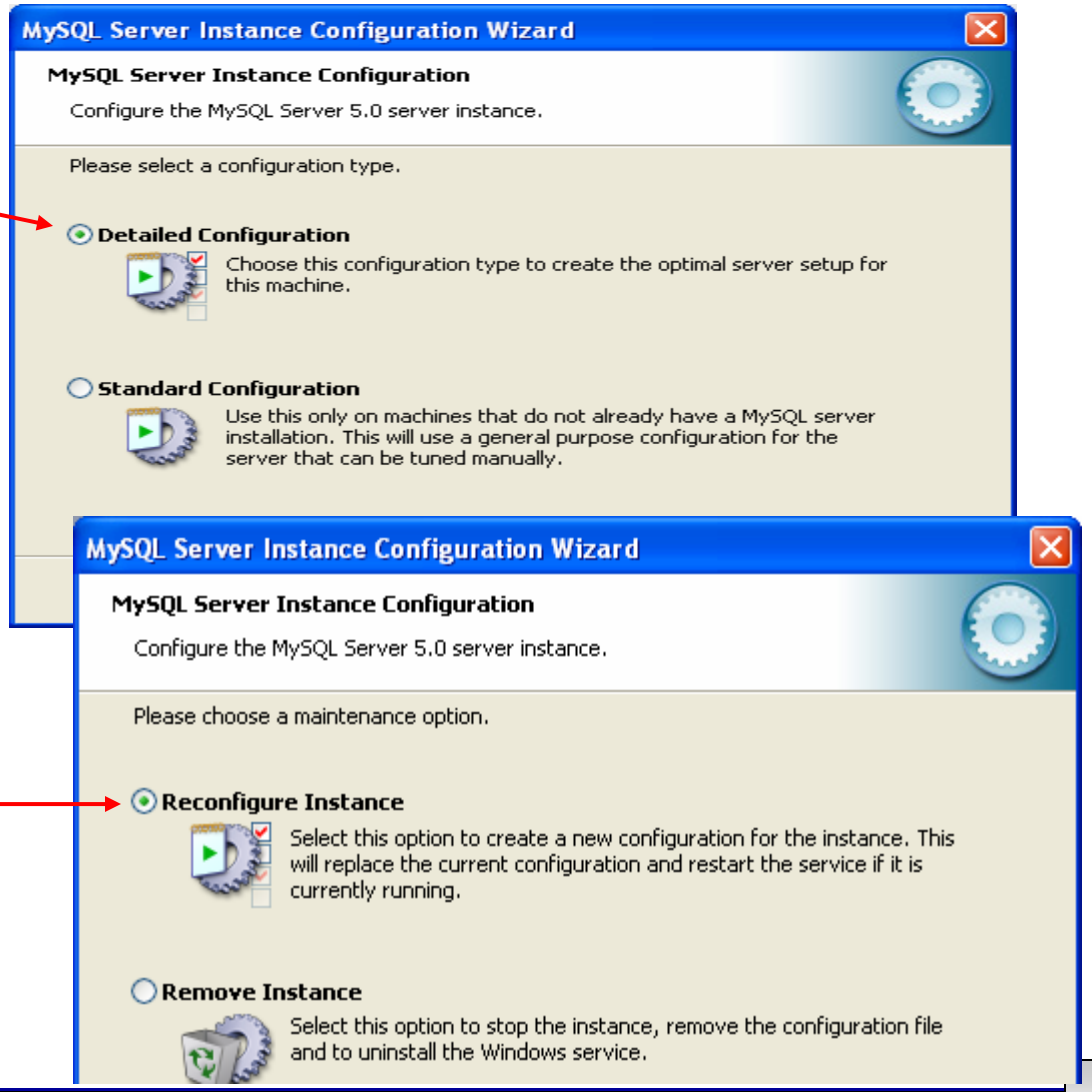
Initial server configuration window



Installing MySQL 5.0 (cont.)

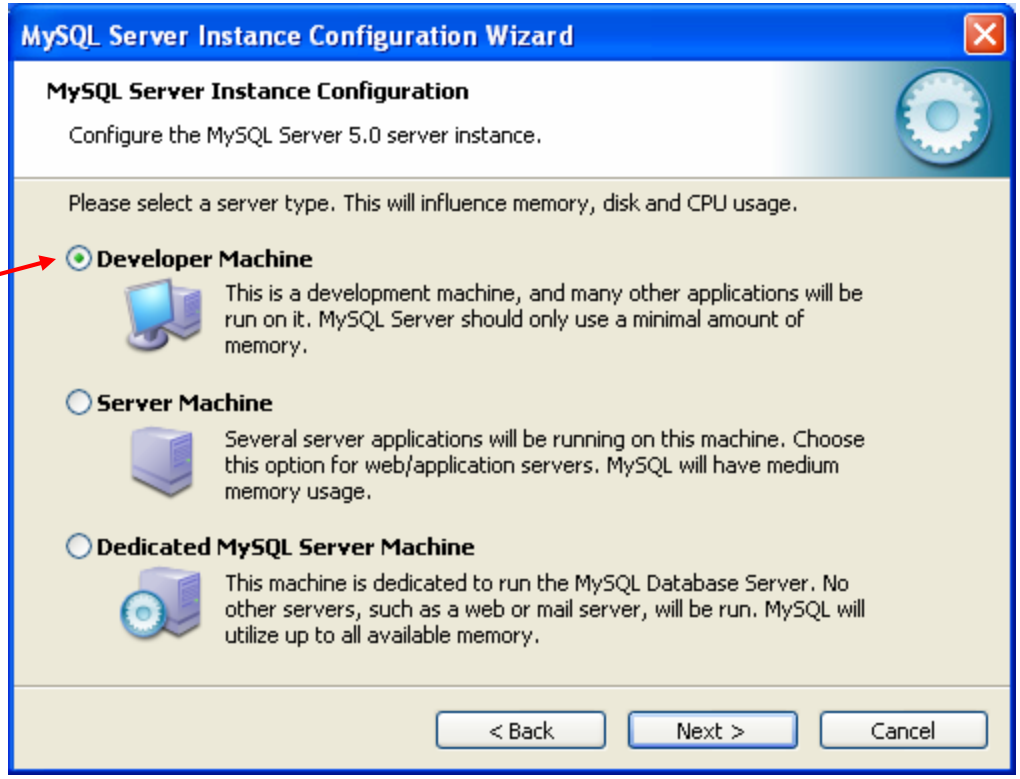
Your choice here. If you are not sure if there is already a MySQL server on your machine, choose the detailed configuration setting.

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.



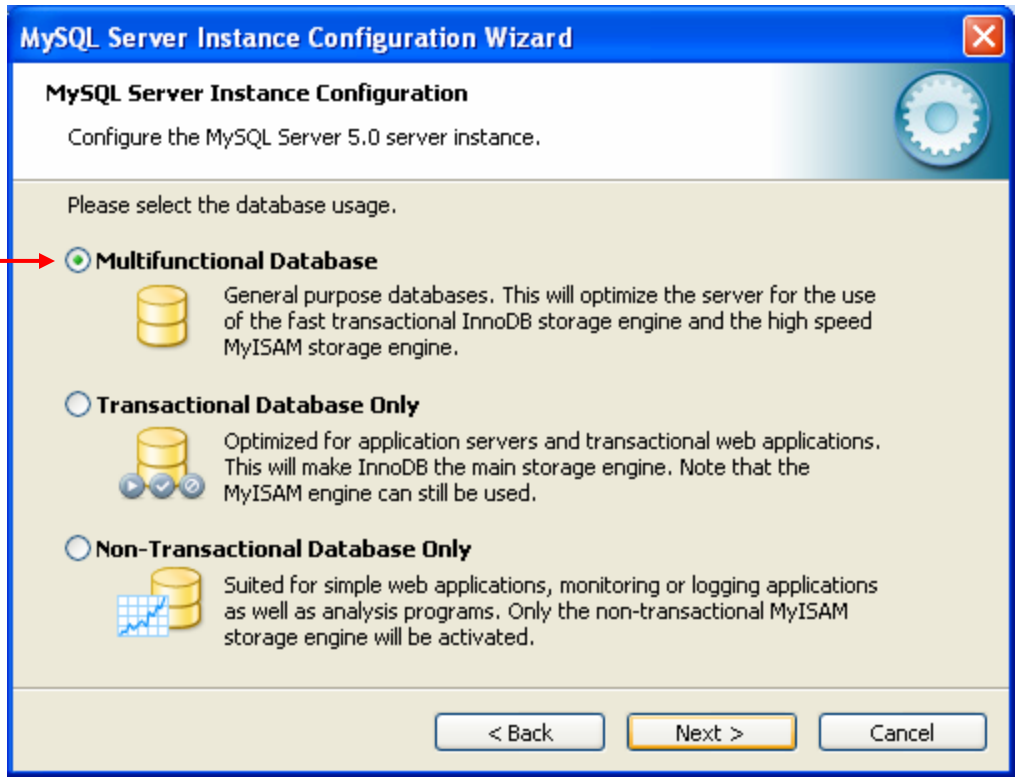
Installing MySQL 5.0 (cont.)

Choose the developer machine option



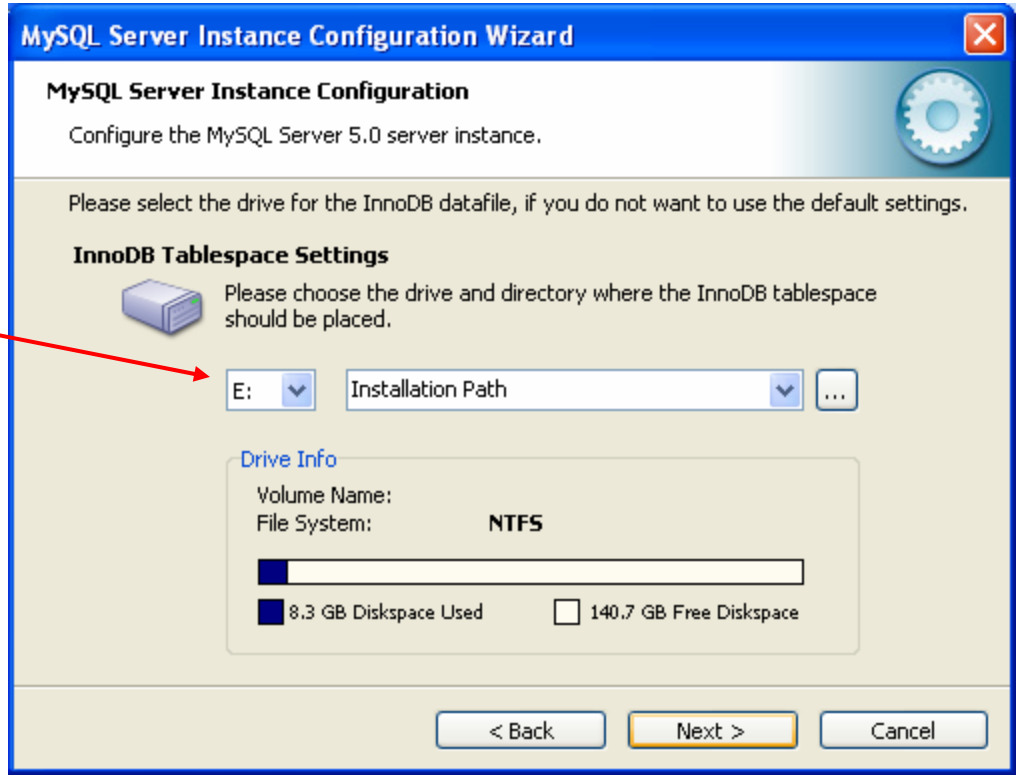
Installing MySQL 5.0 (cont.)

Choose the multifunctional database option



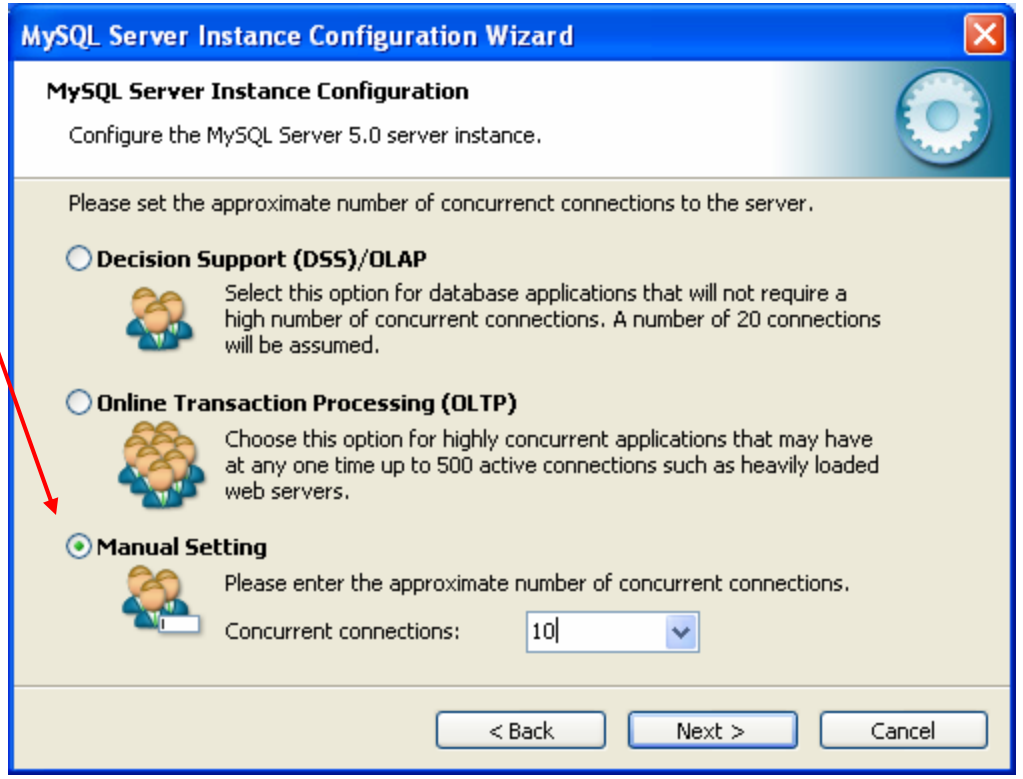
Installing MySQL 5.0 (cont.)

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



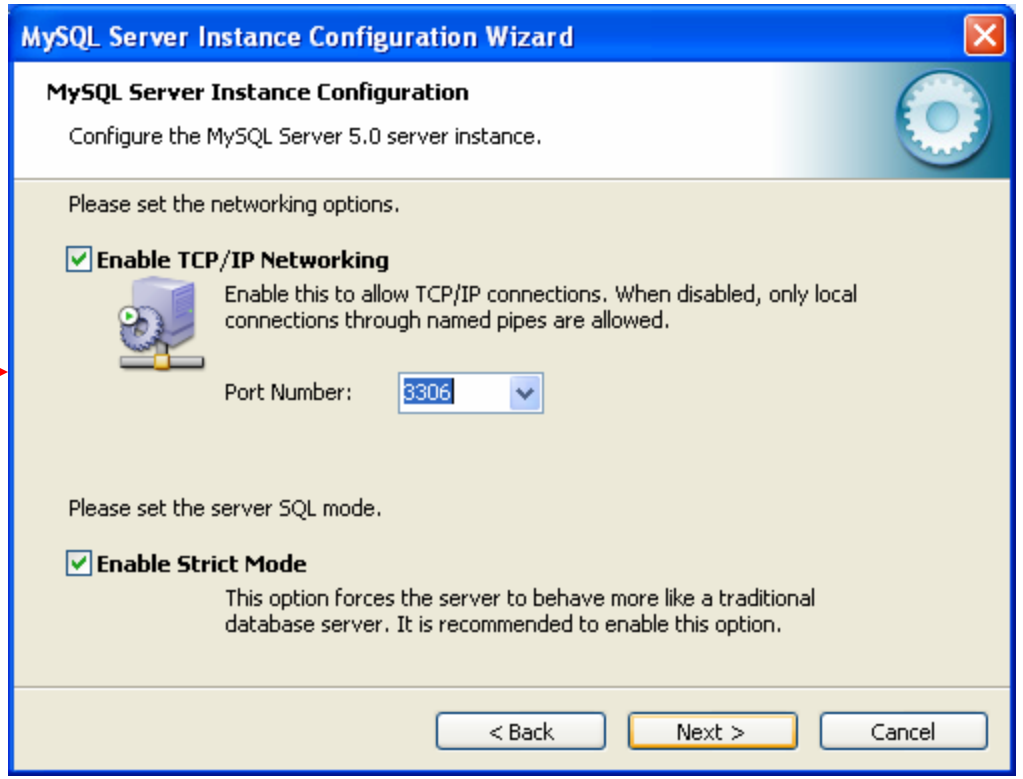
Installing MySQL 5.0 (cont.)

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.



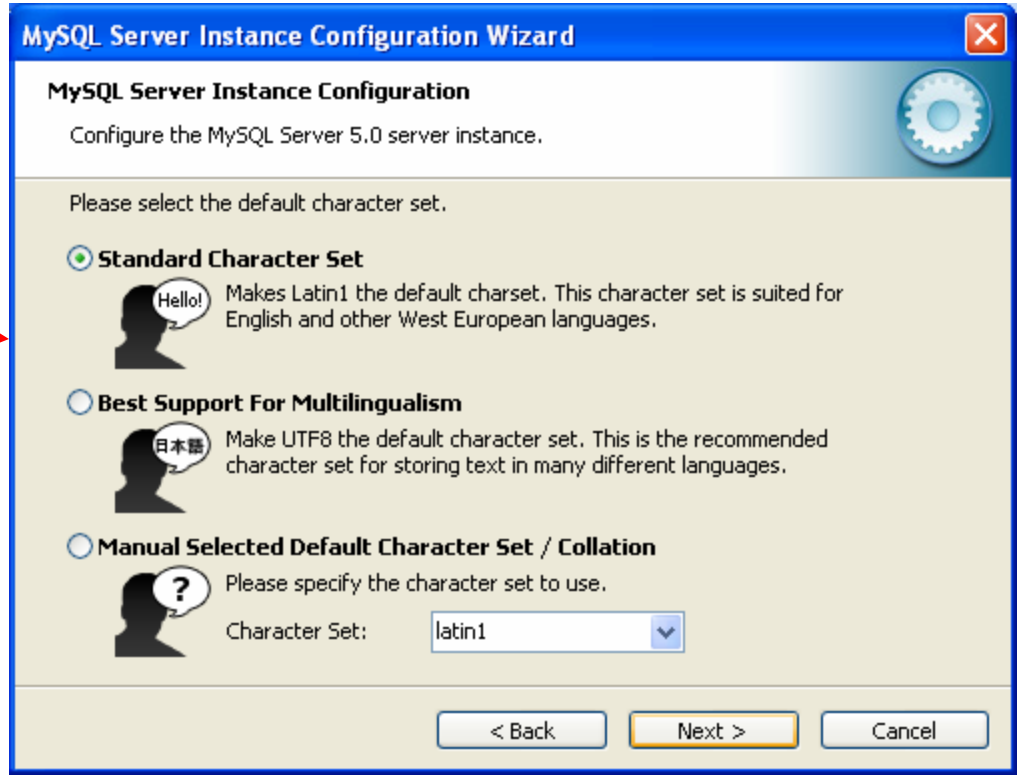
Installing MySQL 5.0 (cont.)

Accept all defaults in this window



Installing MySQL 5.0 (cont.)

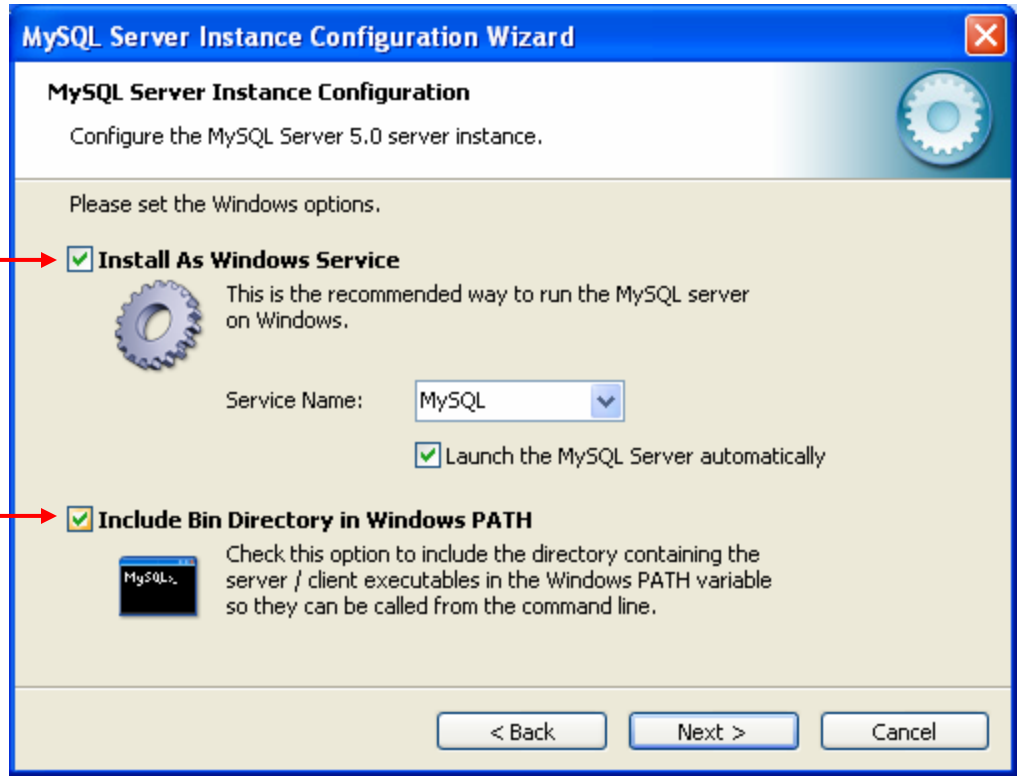
Your choice again



Installing MySQL 5.0 (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.



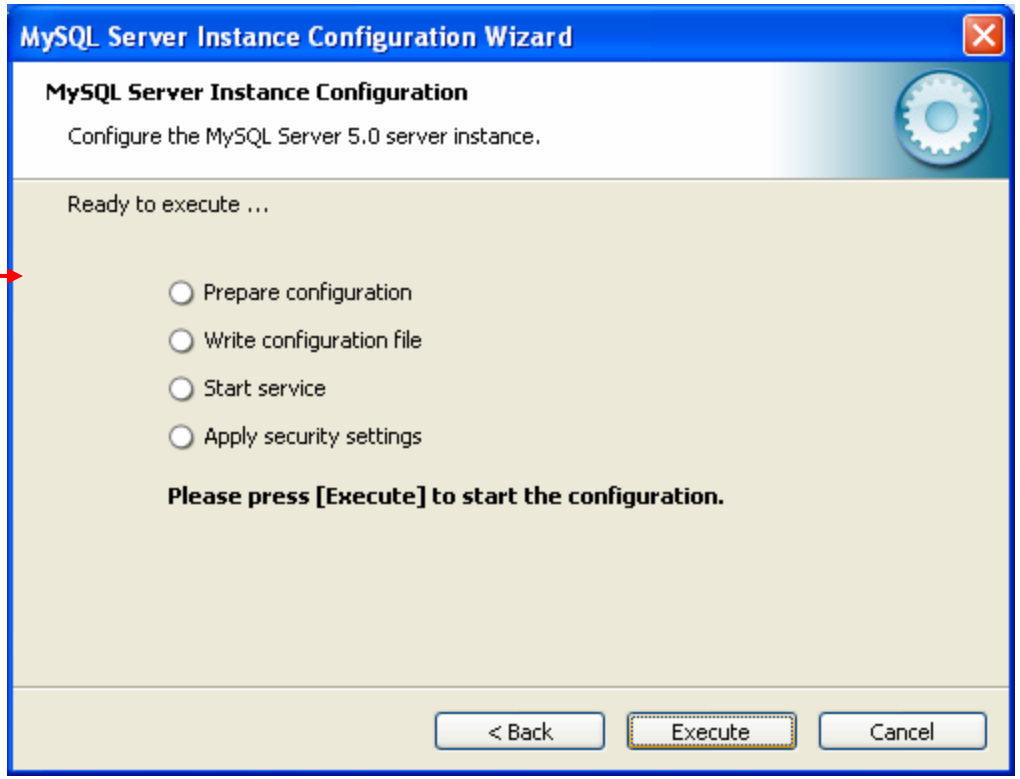
Installing MySQL 5.0 (cont.)

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

Do not enable this option

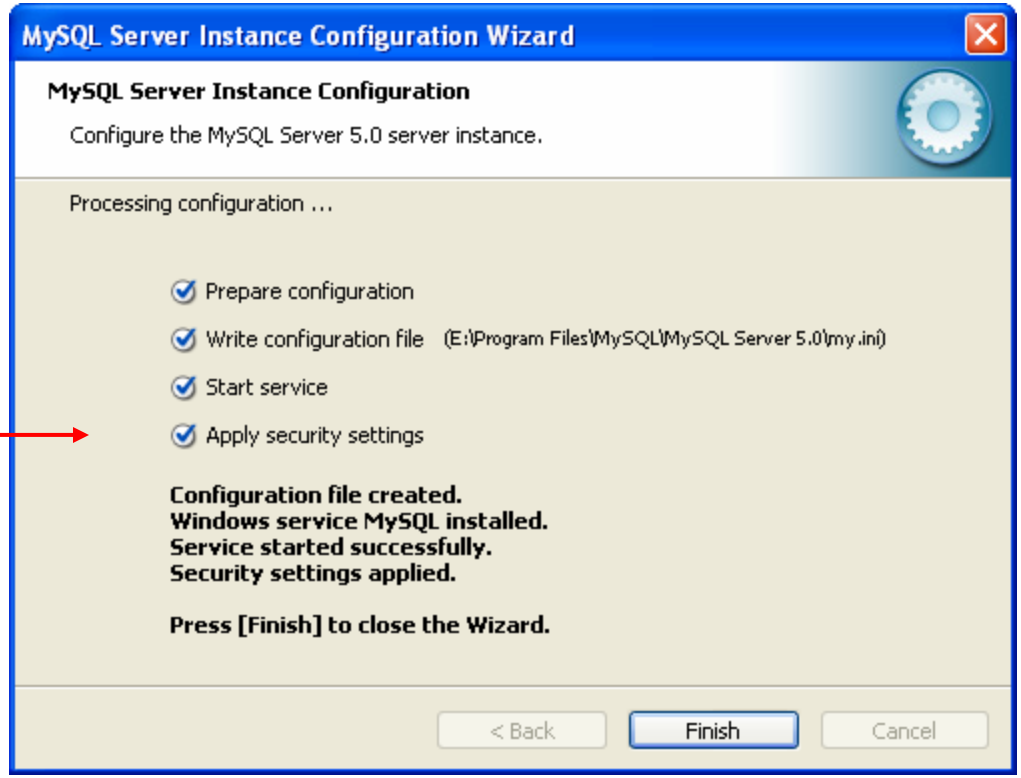
Installing MySQL 5.0 (cont.)

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 5.0 (cont.)

You've successfully installed MySQL!!

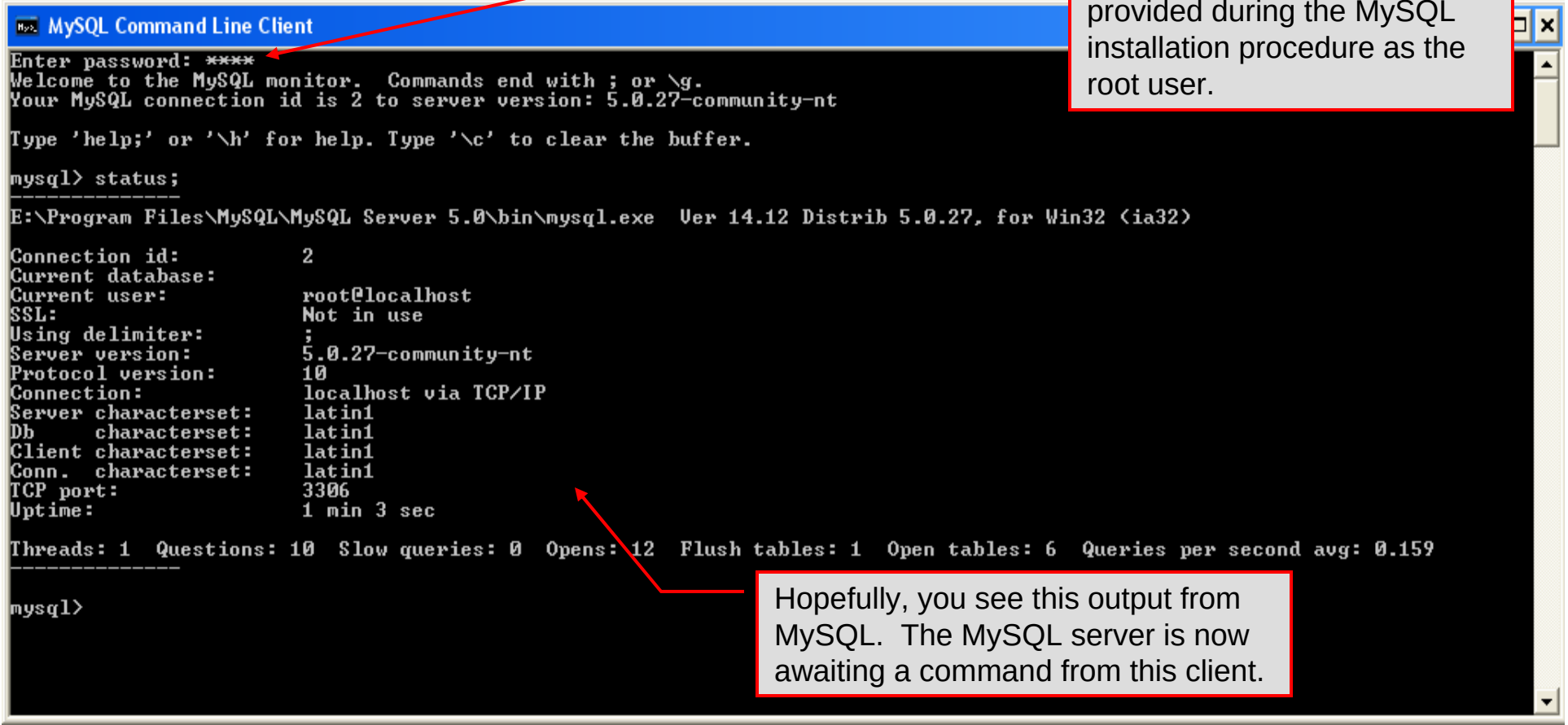


Running MySQL 5.0.27

- 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.

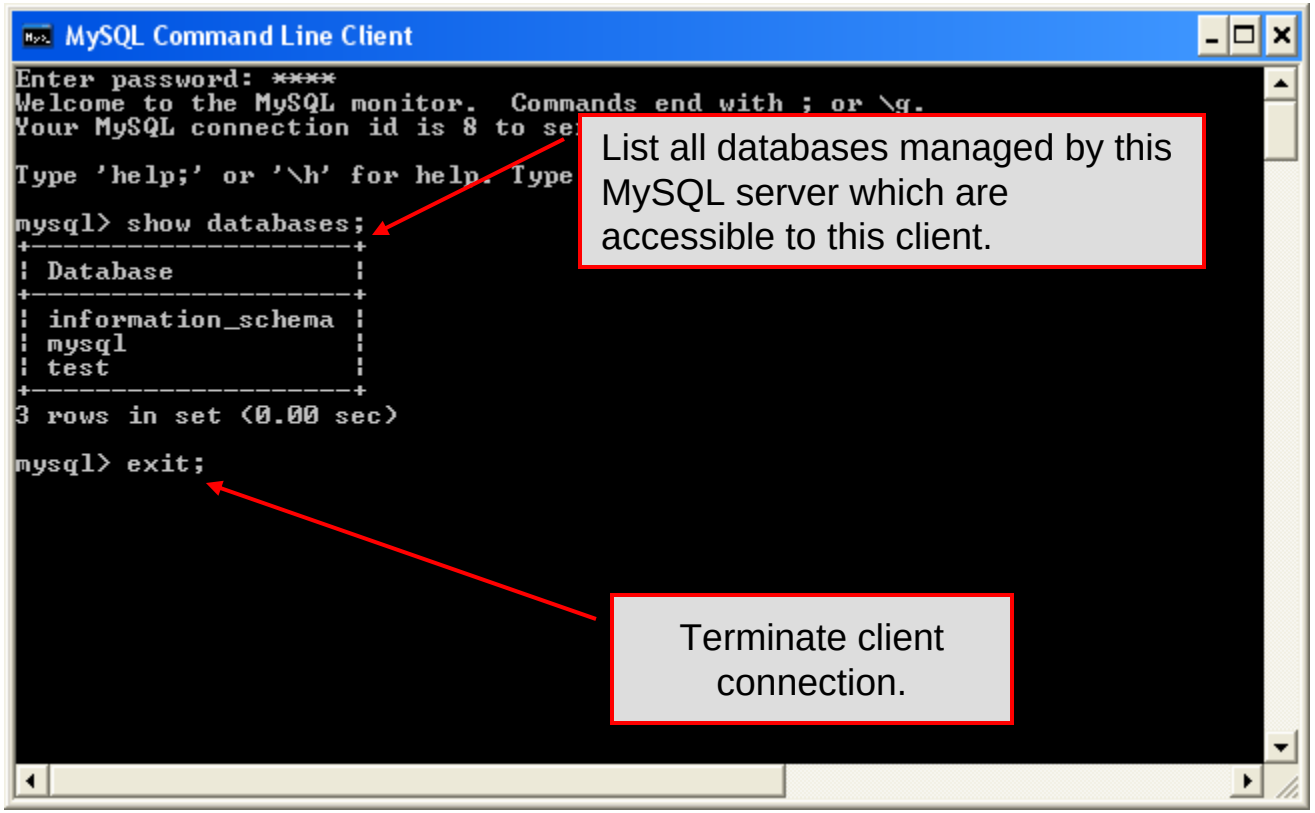
Running MySQL 5.0.27 (cont.)

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



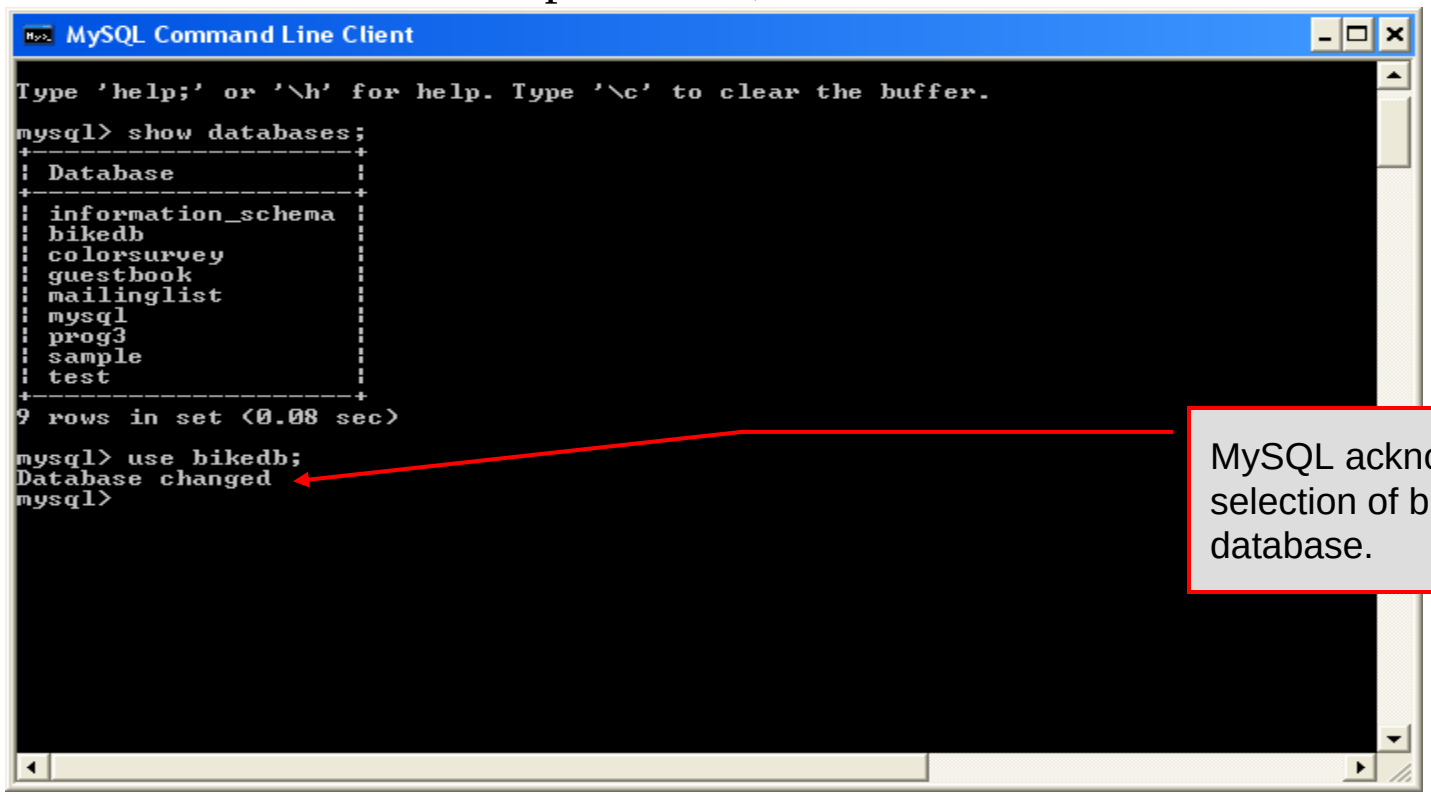
Hopefully, you see this output from MySQL. The MySQL server is now awaiting a command from this client.

Running MySQL 5.0.27 (cont.)



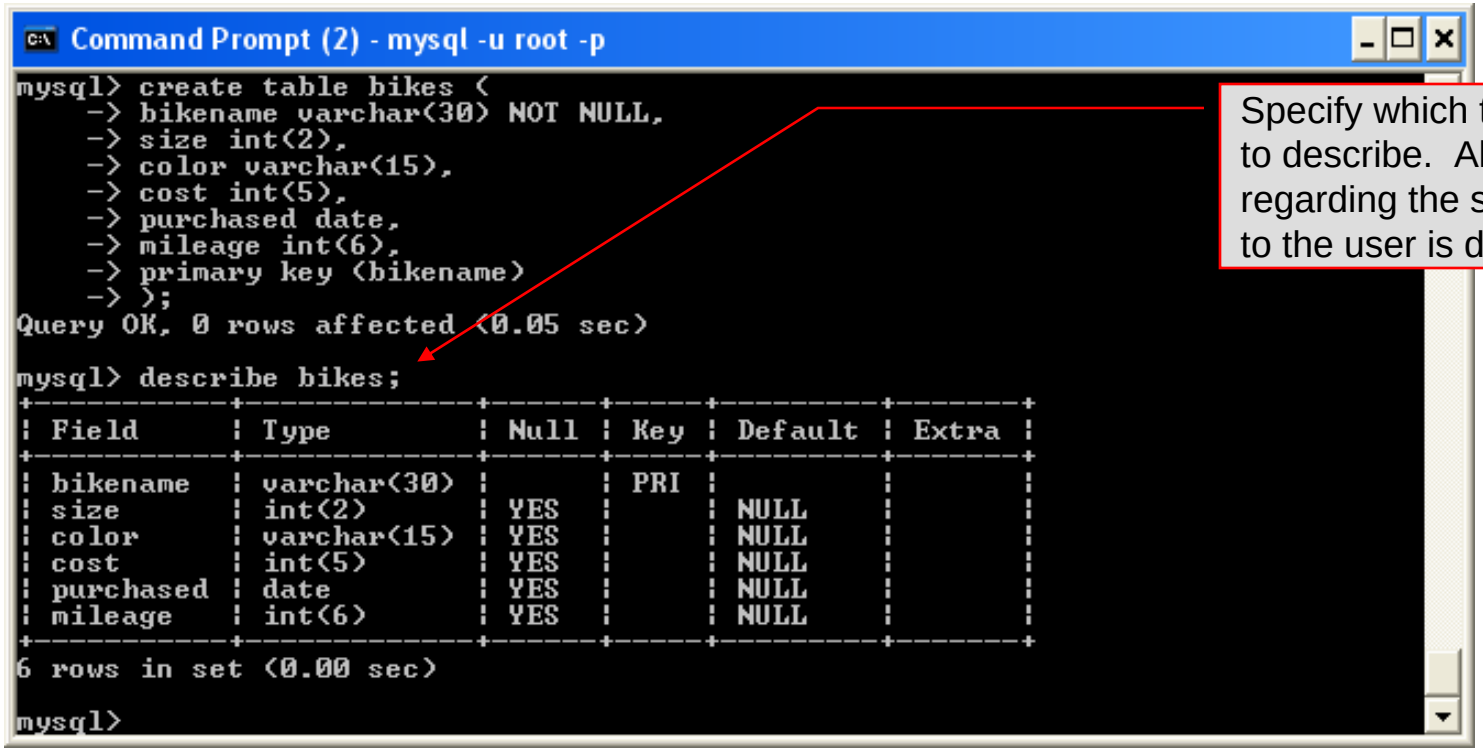
Specifying A Database Within MySQL

- 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.



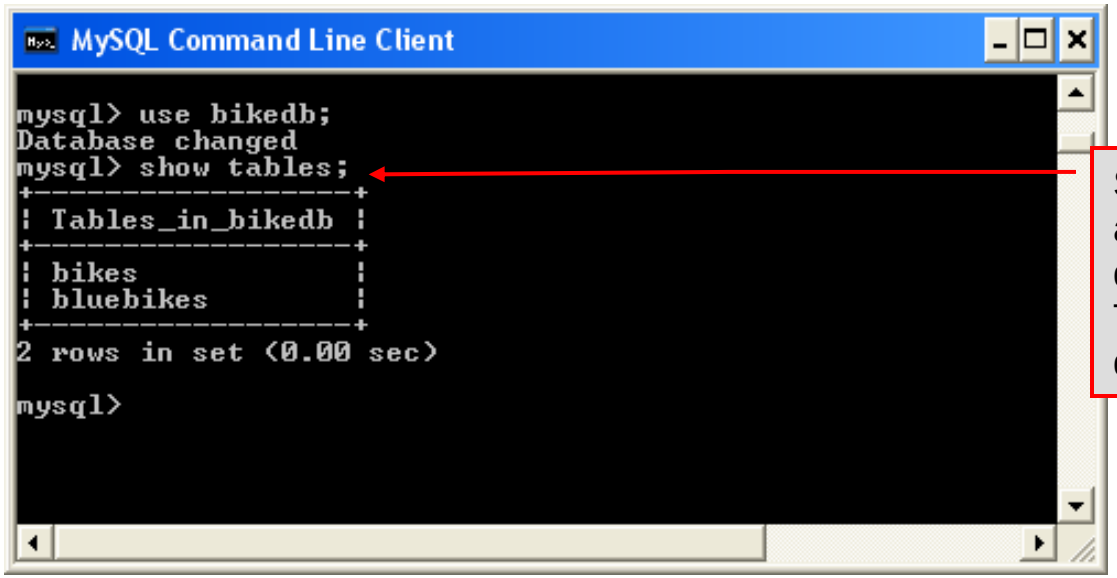
Viewing the Schema of a Relation

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



Viewing the Relations of a Database

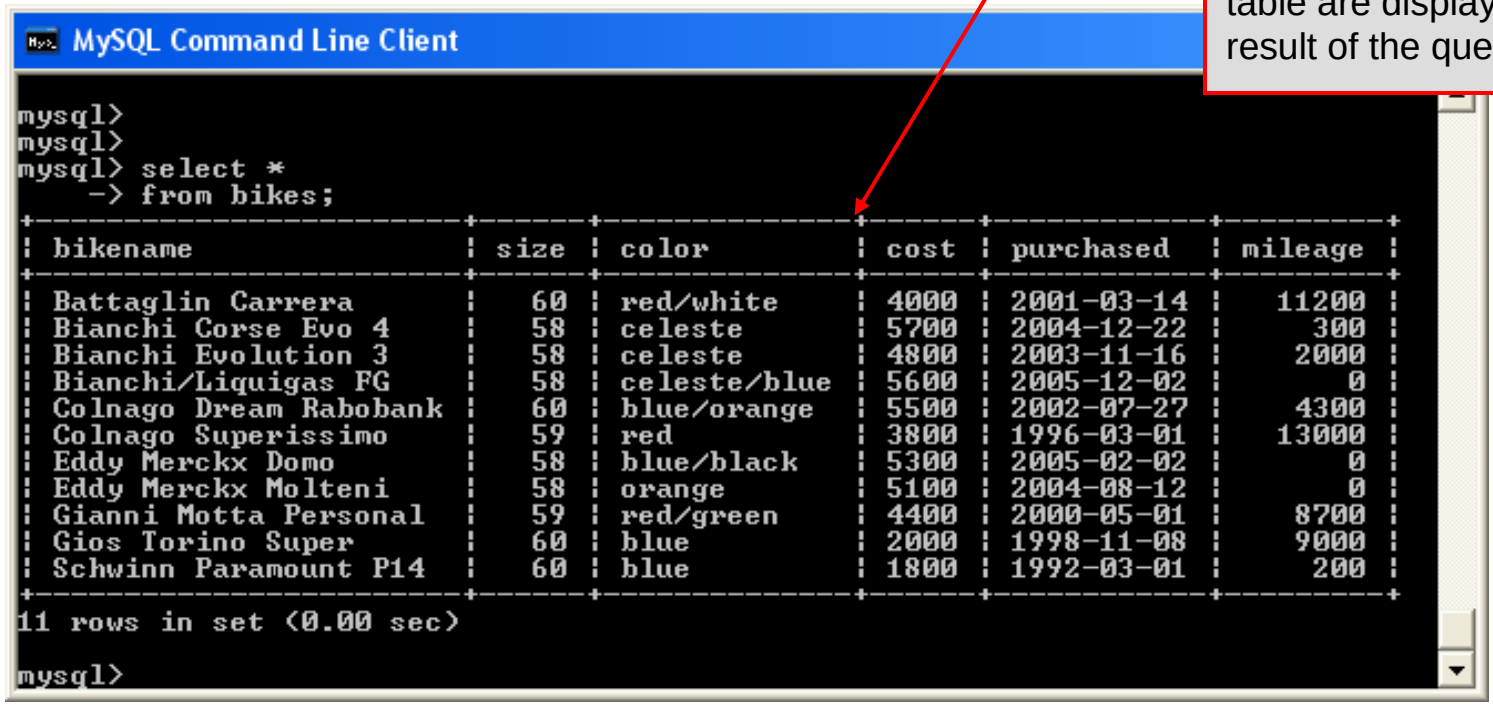
- Once a database has been selected you can see the relations (tables) within that database with the `show tables` command as illustrated below.



Running a Simple Select Query in MySQL

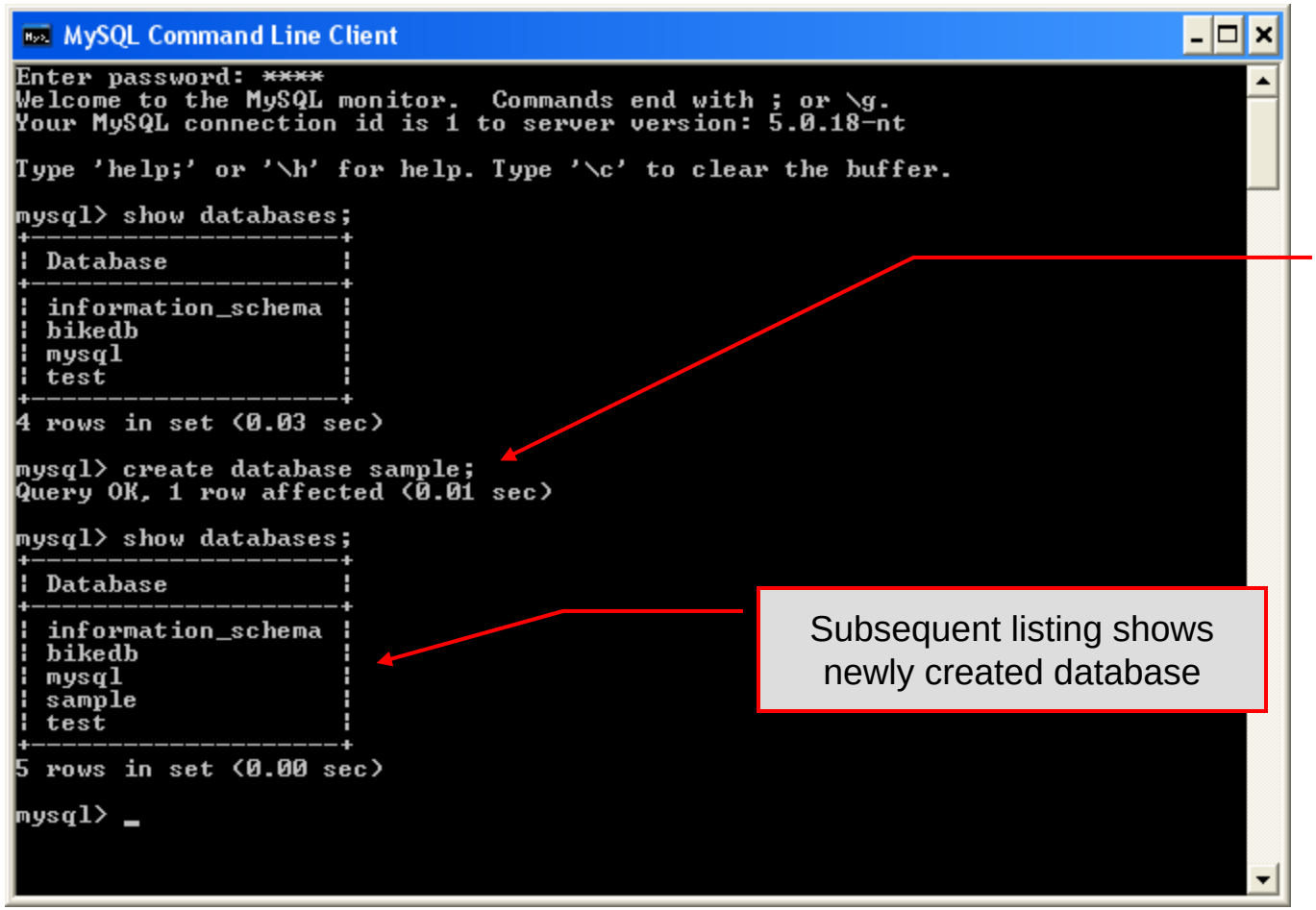
- Within the MySQL monitor, running an SQL query is straight forward. The example below illustrates a simple selection query on the `bikes` table of the `bikedb` database.

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



Creating a Database in MySQL

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

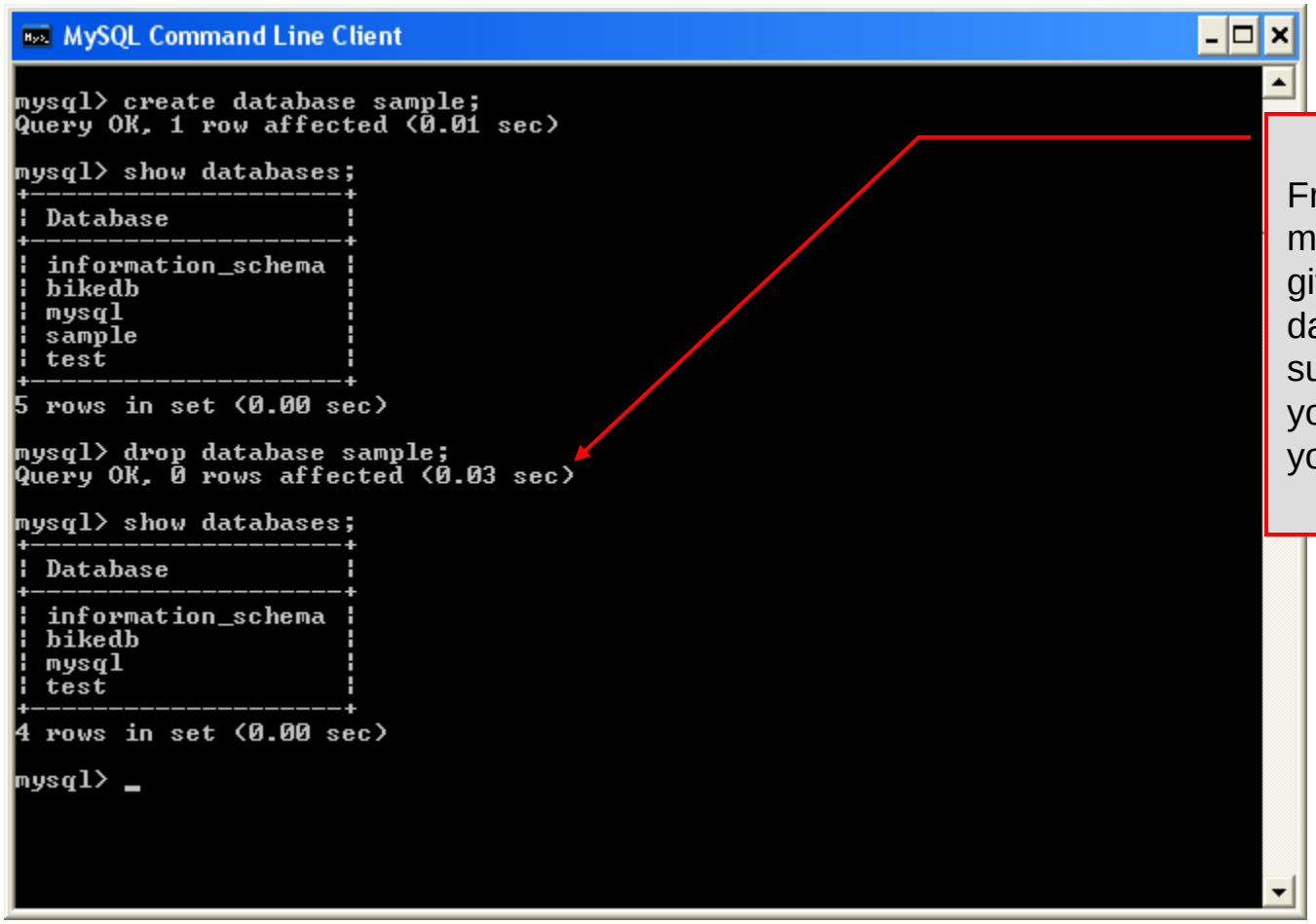


Create new database from within MySQL monitor.

Subsequent listing shows newly created database

Dropping a Database in MySQL

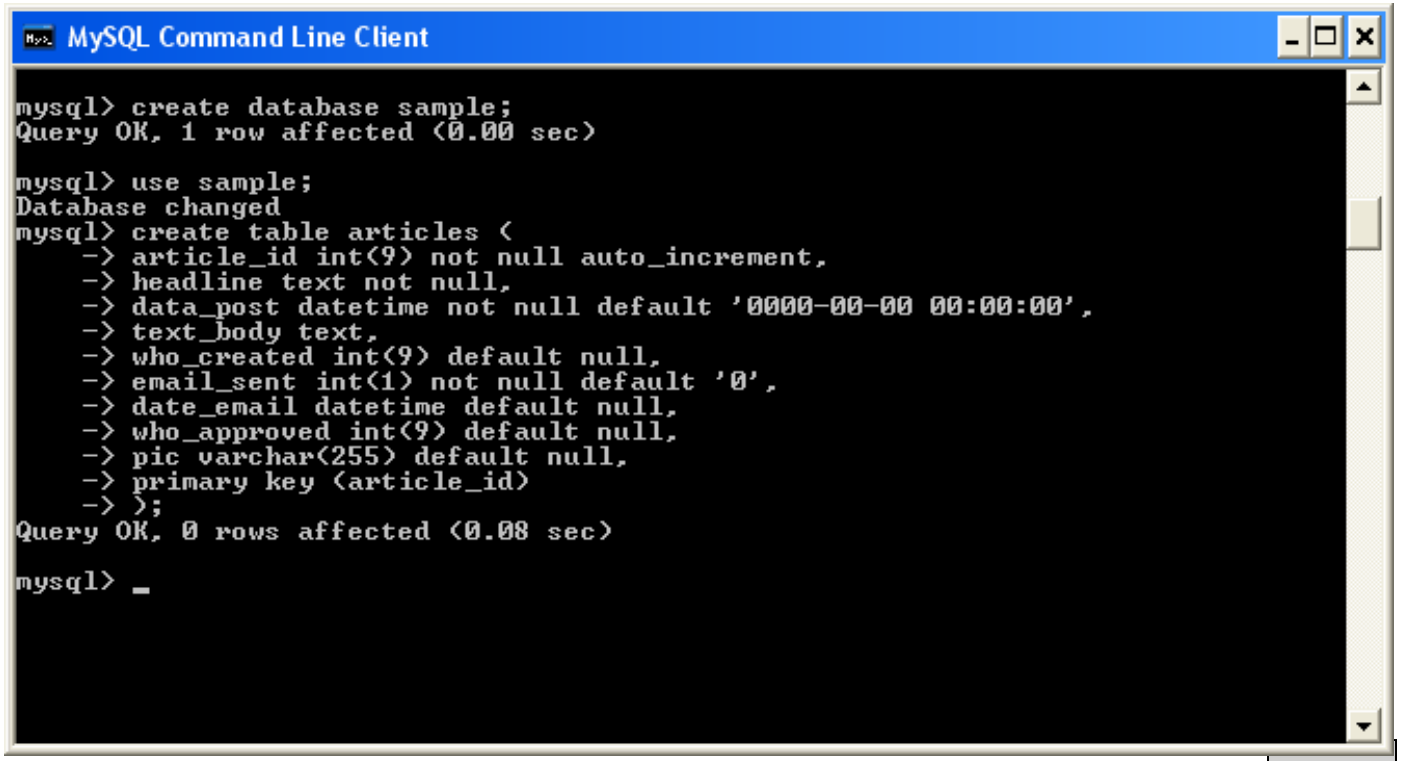
- From the MySQL monitor execute the `drop database <db name>` command.



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.

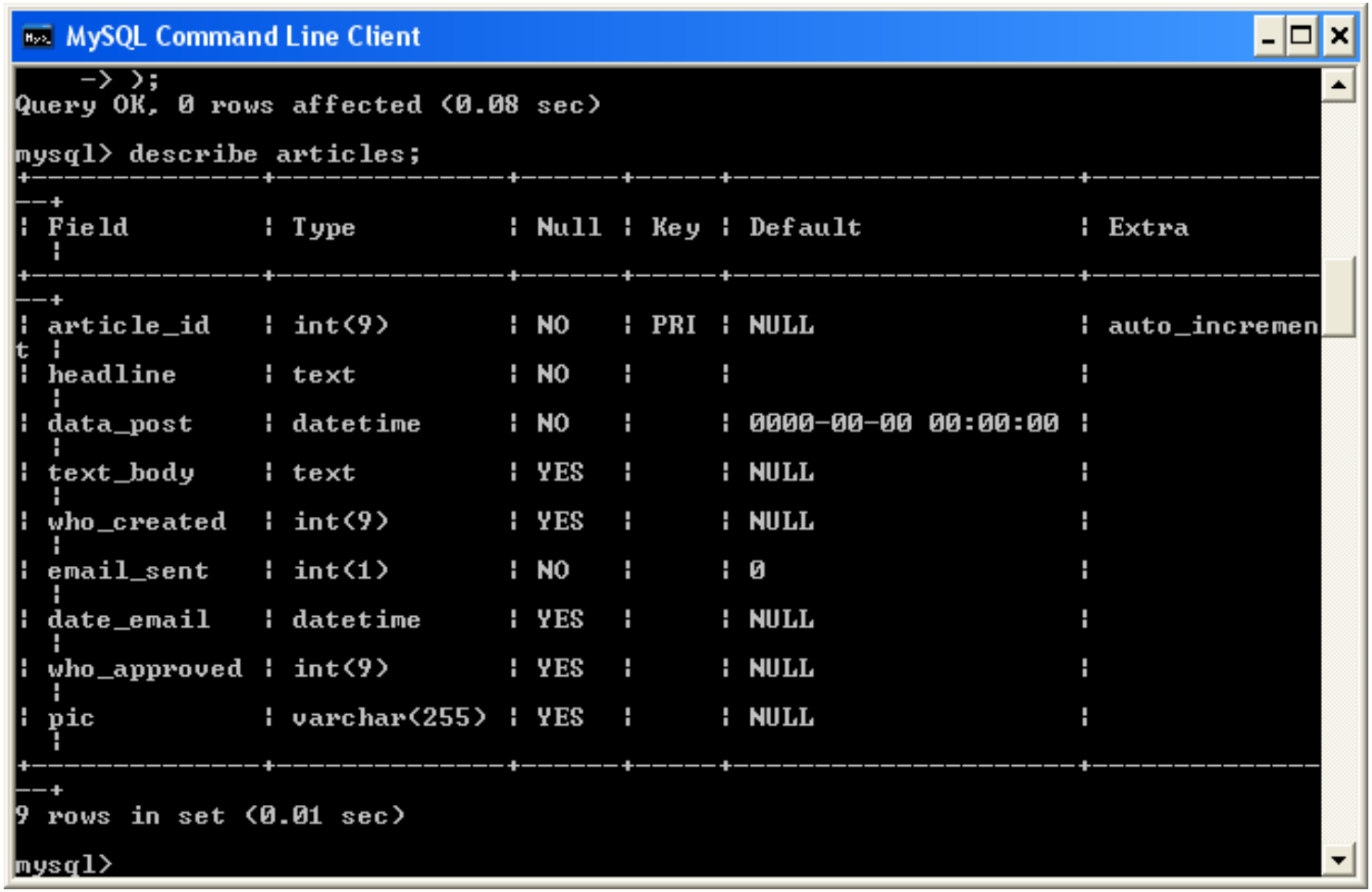
Manipulating Tables in MySQL

- 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.



Manipulating Tables in MySQL (cont.)

Screen shot showing newly created table.



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

MySQL Command Line Client

```
mysql> select * from bikes;
+-----------------------+------+--------------+------+------------+---------+
| bikename              | size | color        | cost | purchased  | mileage |
+-----------------------+------+--------------+------+------------+---------+
| Battaglin Carrera     |   60 | red/white    | 4000 | 2001-03-14 |   11200 |
| Bianchi Corse Evo 4   |   58 | celeste      | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3   |   58 | celeste      | 4800 | 2003-11-16 |    2000 |
| Bianchi/Liquigas FG   |   58 | celeste/blue | 5600 | 2005-12-02 |       0 |
| Colnago Dream Rabobank|   60 | blue/orange  | 5500 | 2002-07-27 |    4300 |
| Colnago Superissimo   |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo      |   58 | blue/black   | 5300 | 2005-02-02 |       0 |
| Eddy Merckx Molteni   |   58 | orange       | 5100 | 2004-08-12 |       0 |
| 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 |
+-----------------------+------+--------------+------+------------+---------+
11 rows in set (0.00 sec)

mysql> create temporary table celestebikes
    -> select *
    -> from bikes
    -> where color = 'celeste';
Query OK, 2 rows affected (0.09 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 Corse Evo 4 |   58 | celeste | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3 |   58 | celeste | 4800 | 2003-11-16 |    2000 |
+-------------------+------+---------+------+------------+---------+
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 5.0.x.

Altering A Table

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

  `alter table` *`table_name action_list`*

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

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

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

Altering A Table (cont.)

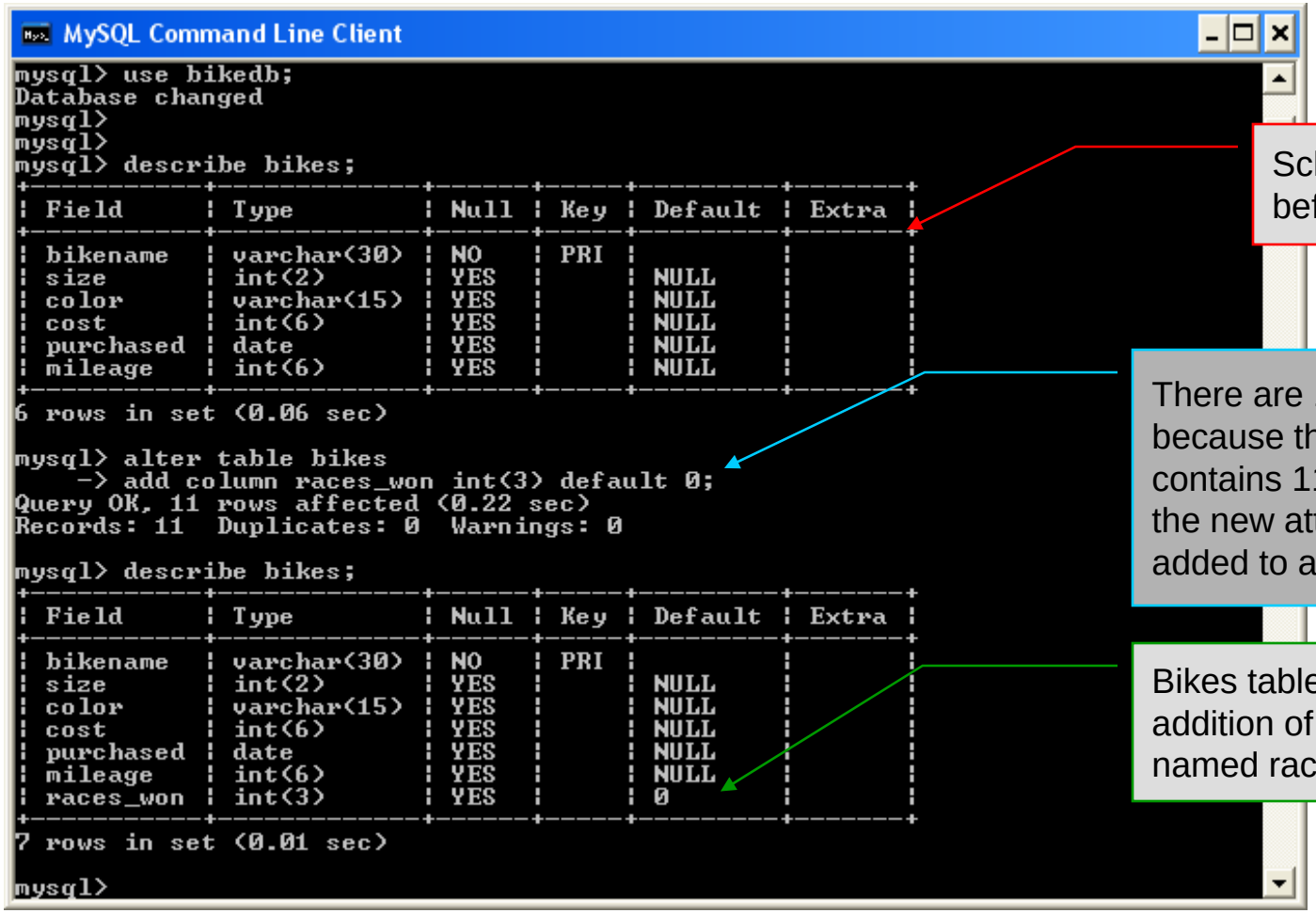
Action Syntax	Action Performed
`add [column] column_declaration [first | after column_name]`	Add a column to the table
`alter [column] column_name {set default literal | drop default}`	Specify new default value for a column or remove old default
`change [column] column_name column_declaration`	Modify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

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

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

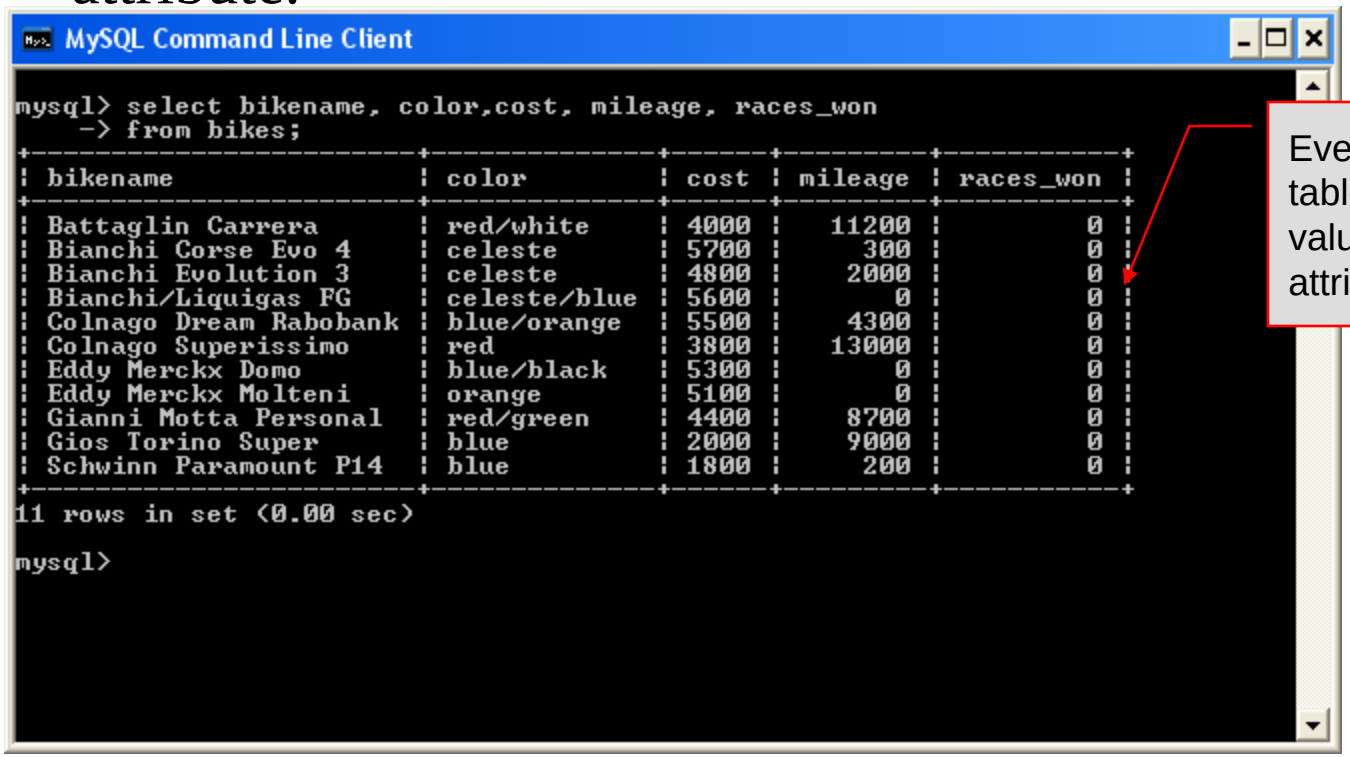
Altering A Table (cont.)

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



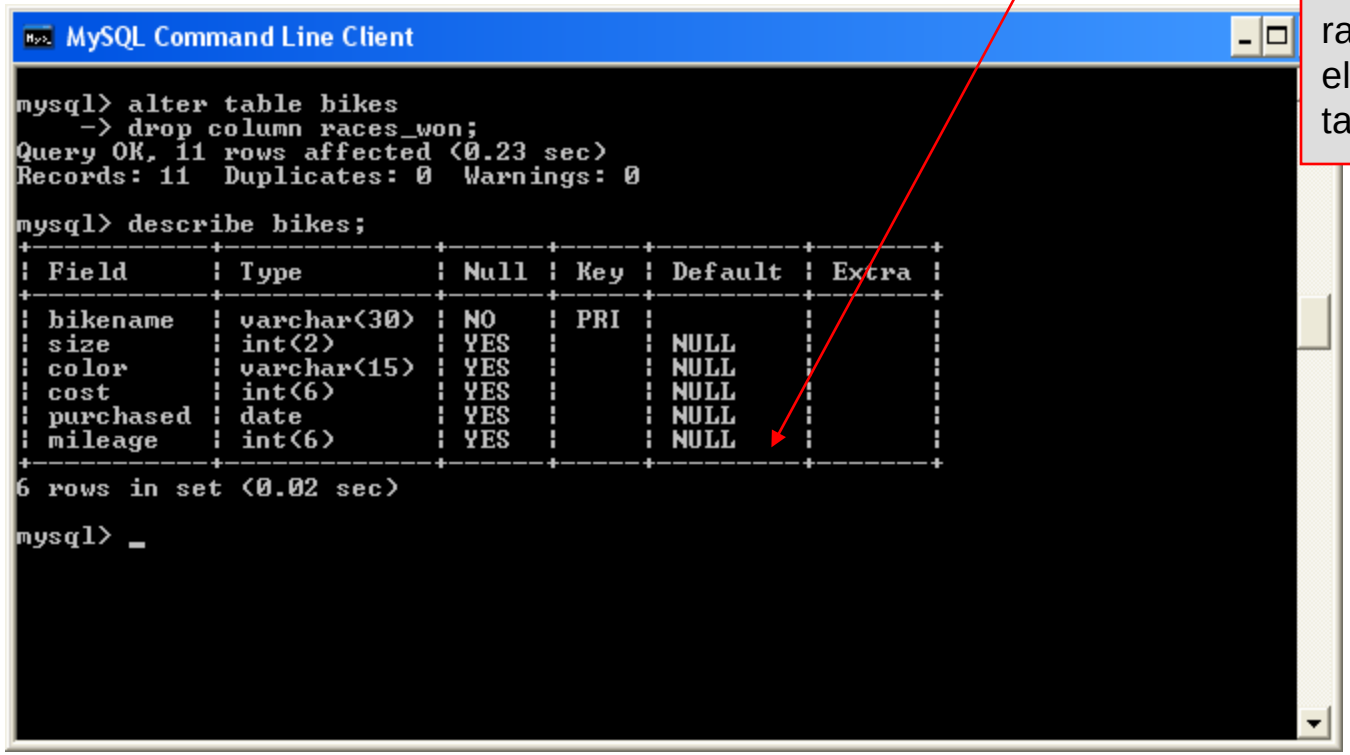
Altering A Table (cont.)

- 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.



Altering A Table (cont.)

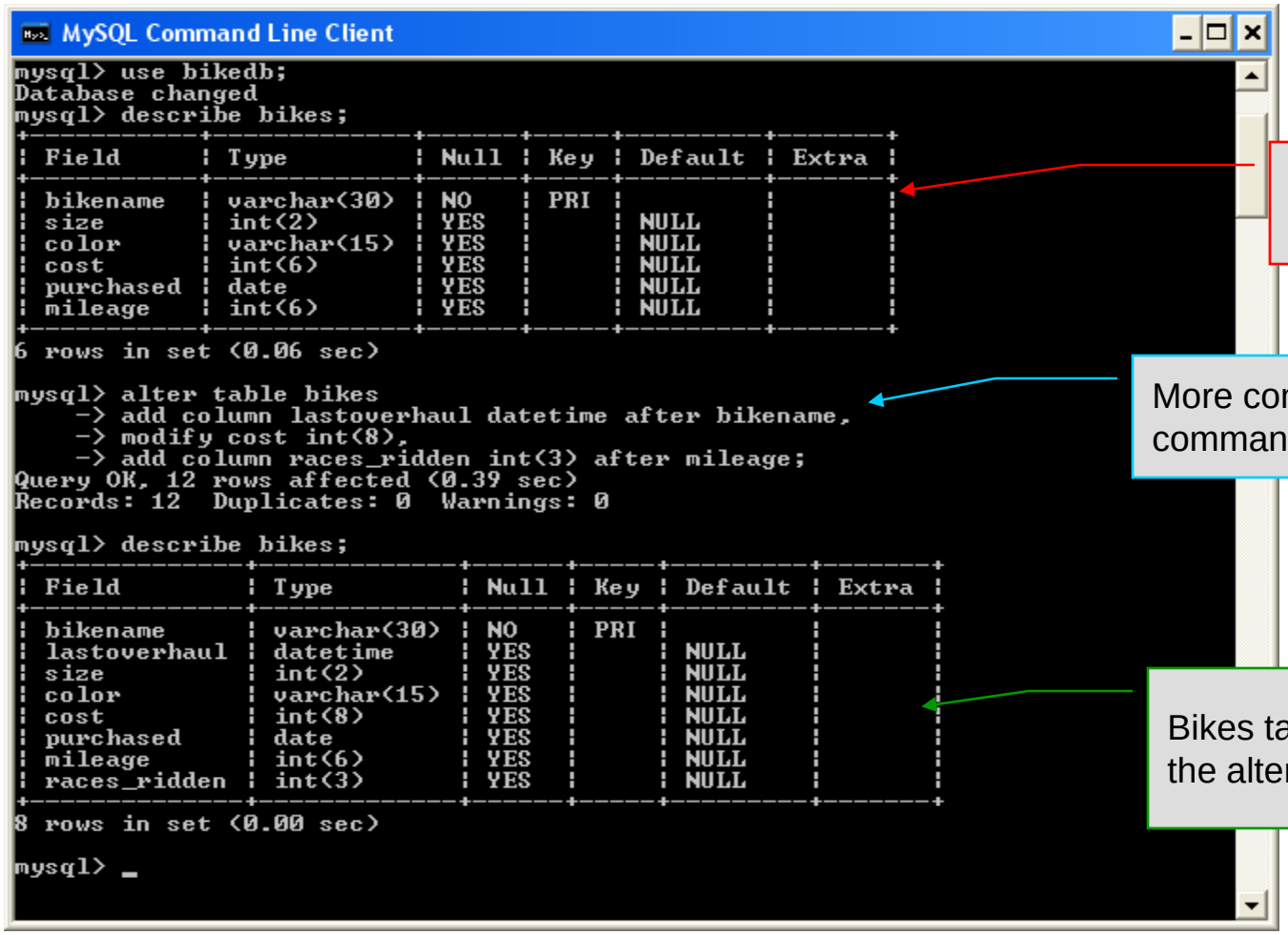
- The screen shot below illustrates dropping a column from a table.
- Note that in general, this type of operation may not always be allowed due to constraint violations.



The attribute races_won has been eliminated from the table.

Altering A Table (cont.)

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

Examples: Inserting Data Into A Table

MySQL Command Line Client

```
mysql> select * from bikes;
+-----------------------+------+--------------+------+------------+---------+
| bikename              | size | color        | cost | purchased  | mileage |
+-----------------------+------+--------------+------+------------+---------+
| Battaglin Carrera     |   60 | red/white    | 4000 | 2001-03-14 |   11200 |
| Bianchi Corse Evo 4   |   58 | celeste      | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3   |   58 | celeste      | 4800 | 2003-11-16 |    2000 |
| Bianchi/Liquigas FG   |   58 | celeste/blue | 5600 | 2005-12-02 |       0 |
| Colnago Dream Rabobank|   60 | blue/orange  | 5500 | 2002-07-27 |    4300 |
| Colnago Superissimo   |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo      |   58 | blue/black   | 5300 | 2005-02-02 |       0 |
| Eddy Merckx Molteni   |   58 | orange       | 5100 | 2004-08-12 |       0 |
| 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 |
+-----------------------+------+--------------+------+------------+---------+
11 rows in set (0.00 sec)

mysql> insert into bikes
    -> set bikename='Eddy Merckx MXM',
    -> cost=8200,
    -> mileage=150,
    -> purchased='2006-01-14',
    -> color='black/red',
    -> size=58;
Query OK, 1 row affected (0.03 sec)

mysql> select * from bikes;
+-----------------------+------+--------------+------+------------+---------+
| bikename              | size | color        | cost | purchased  | mileage |
+-----------------------+------+--------------+------+------------+---------+
| Battaglin Carrera     |   60 | red/white    | 4000 | 2001-03-14 |   11200 |
| Bianchi Corse Evo 4   |   58 | celeste      | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3   |   58 | celeste      | 4800 | 2003-11-16 |    2000 |
| Bianchi/Liquigas FG   |   58 | celeste/blue | 5600 | 2005-12-02 |       0 |
| Colnago Dream Rabobank|   60 | blue/orange  | 5500 | 2002-07-27 |    4300 |
| Colnago Superissimo   |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo      |   58 | blue/black   | 5300 | 2005-02-02 |       0 |
| Eddy Merckx Molteni   |   58 | orange       | 5100 | 2004-08-12 |       0 |
| Eddy Merckx MXM       |   58 | black/red    | 8200 | 2006-01-14 |     150 |
| 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 |
+-----------------------+------+--------------+------+------------+---------+
12 rows in set (0.00 sec)

mysql>
```

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

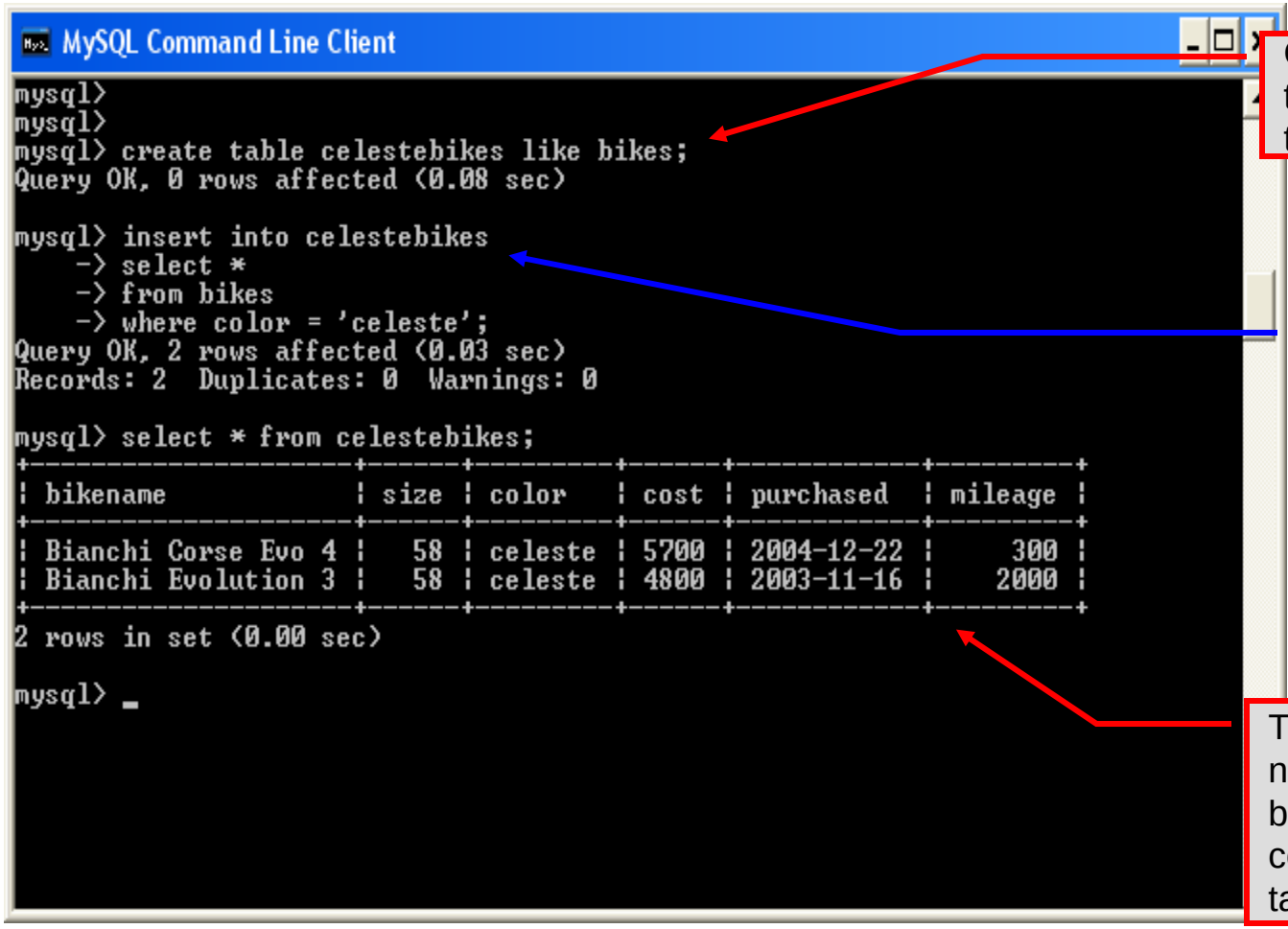
Examples: Inserting Data Into A Table

MySQL Command Line Client

```
mysql> select * from bikes;
+----------------------+------+--------------+------+------------+---------+
| bikename             | size | color        | cost | purchased  | mileage |
+----------------------+------+--------------+------+------------+---------+
| Battaglin Carrera    |   60 | red/white    | 4000 | 2001-03-14 |   11200 |
| Bianchi Corse Evo 4  |   58 | celeste      | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3  |   58 | celeste      | 4800 | 2003-11-16 |    2000 |
| Bianchi/Liquigas FG  |   58 | celeste/blue | 5600 | 2005-12-02 |       0 |
| Colnago Dream Rabobank |   60 | blue/orange | 5500 | 2002-07-27 |    4300 |
| Colnago Superissimo  |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo     |   58 | blue/black   | 5300 | 2005-02-02 |       0 |
| Eddy Merckx Molteni  |   58 | orange       | 5100 | 2004-08-12 |       0 |
| 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 |
+----------------------+------+--------------+------+------------+---------+
11 rows in set (0.00 sec)

mysql> insert into bikes
    -> values ('Eddy Merckx MXM',58,'black/red',8200,'2006-01-14',150);
Query OK, 1 row affected (0.01 sec)

mysql> select * from bikes;
+----------------------+------+--------------+------+------------+---------+
| bikename             | size | color        | cost | purchased  | mileage |
+----------------------+------+--------------+------+------------+---------+
| Battaglin Carrera    |   60 | red/white    | 4000 | 2001-03-14 |   11200 |
| Bianchi Corse Evo 4  |   58 | celeste      | 5700 | 2004-12-22 |     300 |
| Bianchi Evolution 3  |   58 | celeste      | 4800 | 2003-11-16 |    2000 |
| Bianchi/Liquigas FG  |   58 | celeste/blue | 5600 | 2005-12-02 |       0 |
| Colnago Dream Rabobank |   60 | blue/orange | 5500 | 2002-07-27 |    4300 |
| Colnago Superissimo  |   59 | red          | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo     |   58 | blue/black   | 5300 | 2005-02-02 |       0 |
| Eddy Merckx Molteni  |   58 | orange       | 5100 | 2004-08-12 |       0 |
| Eddy Merckx MXM      |   58 | black/red    | 8200 | 2006-01-14 |     150 |
| 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 |
+----------------------+------+--------------+------+------------+---------+
12 rows in set (0.00 sec)

mysql> _
```

Using Form 2 for insertion – attribute order is important.

Examples: Inserting Data Into A Table

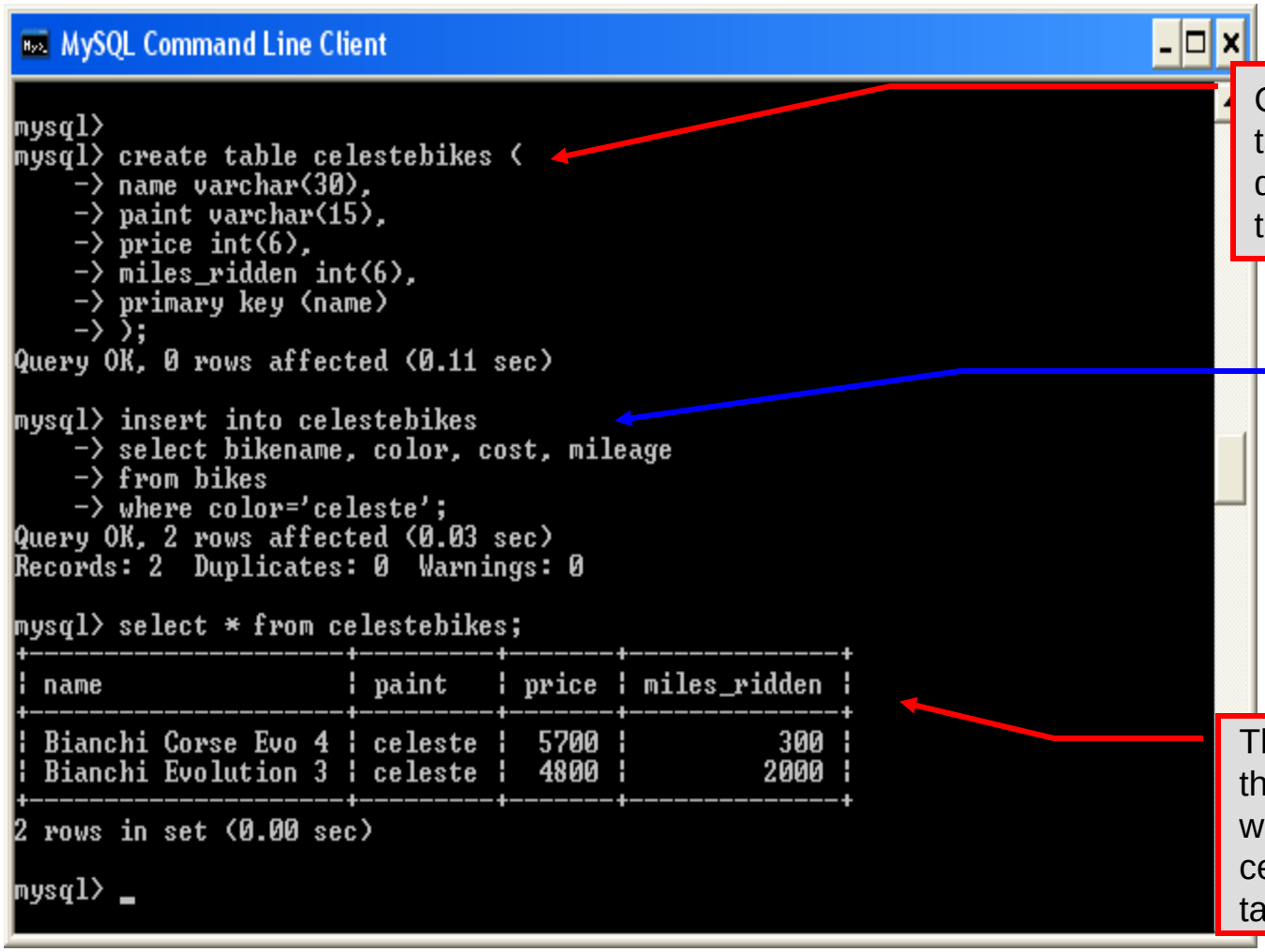


Creates an initially empty table just like the bikes table

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



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

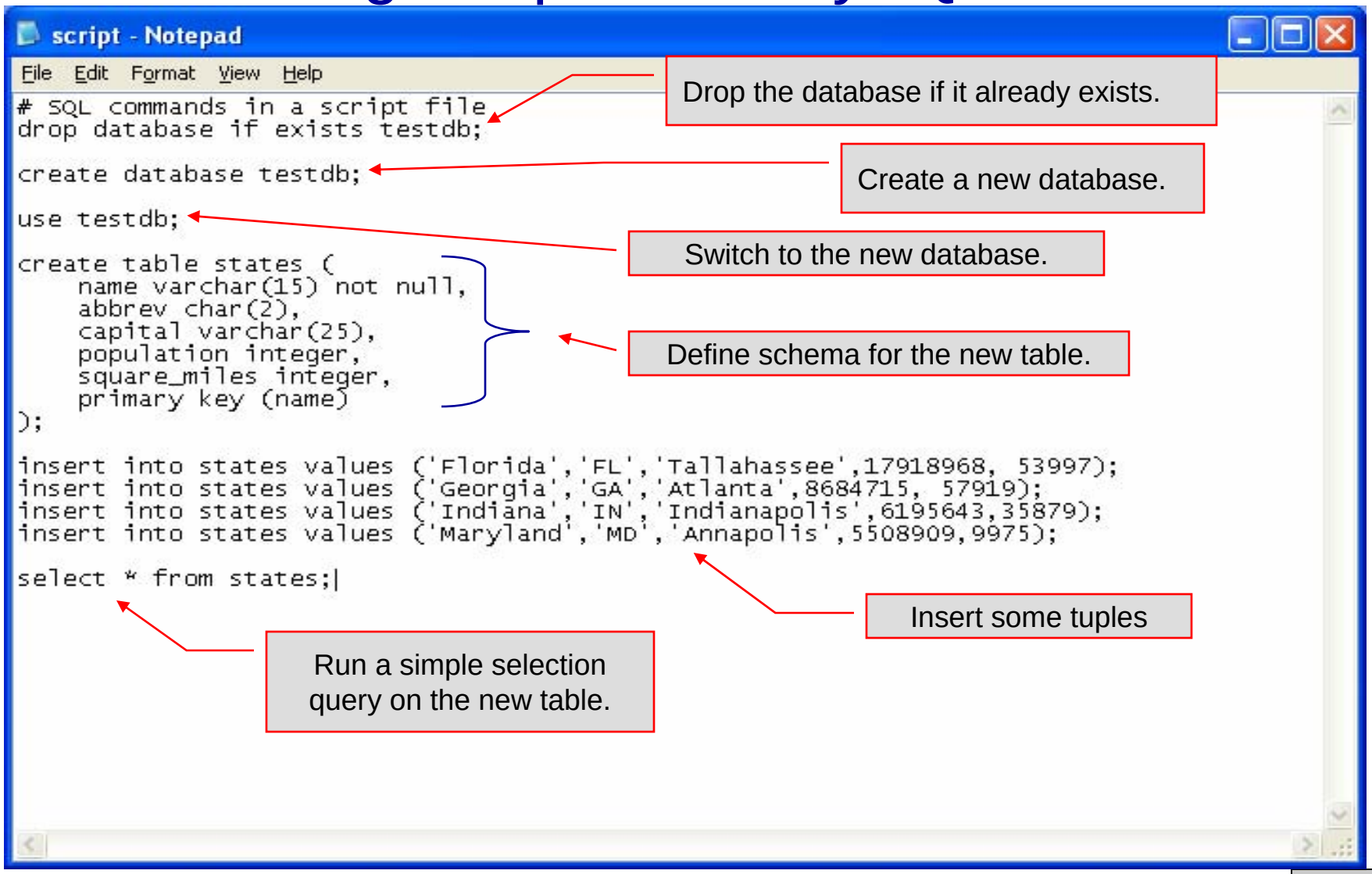
Using Form 3 for insertion

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

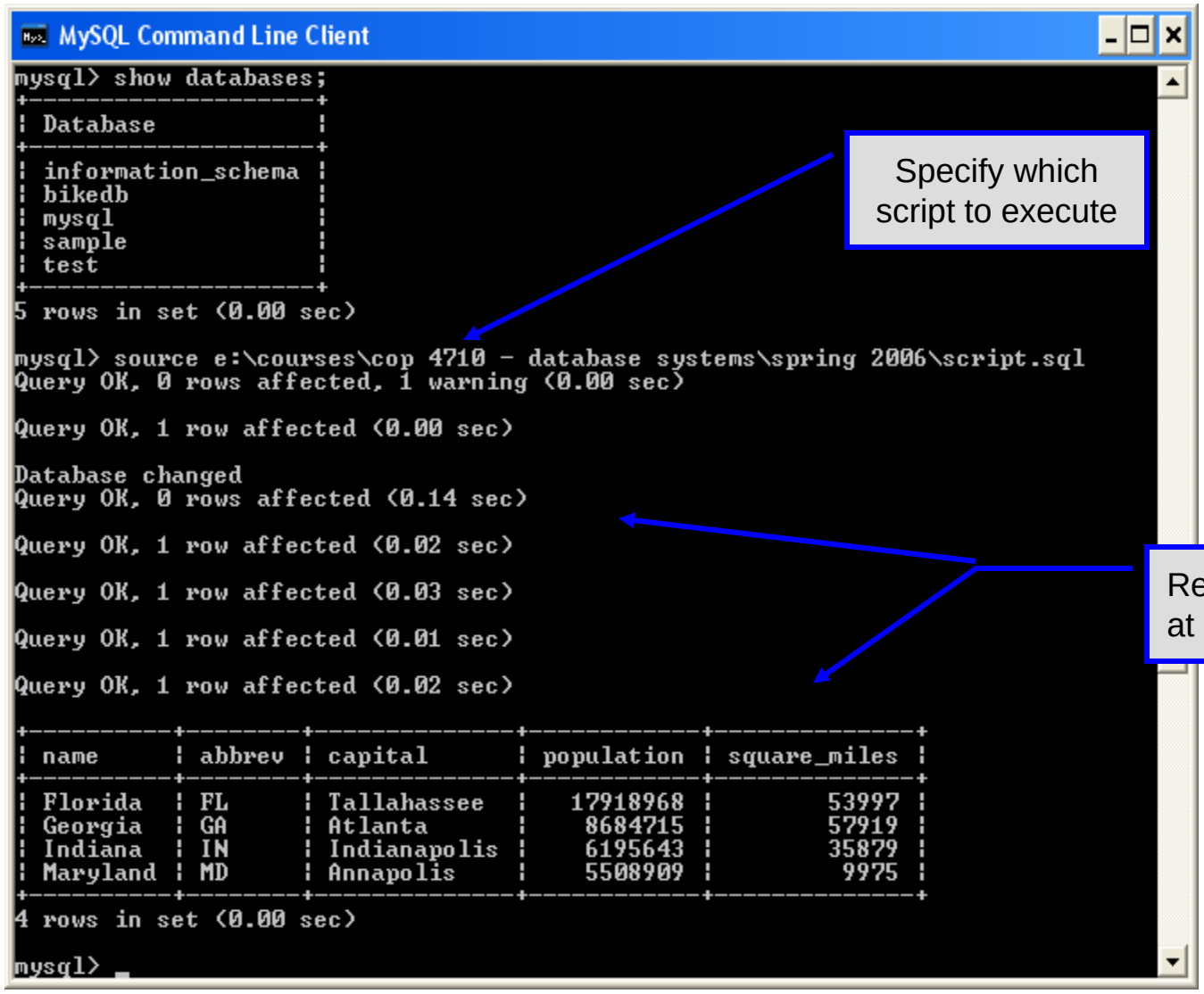
Using Scripts with MySQL

- 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.)



Using Scripts with MySQL (cont.)



Importing Data Using the `mysqlimport` Utility

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

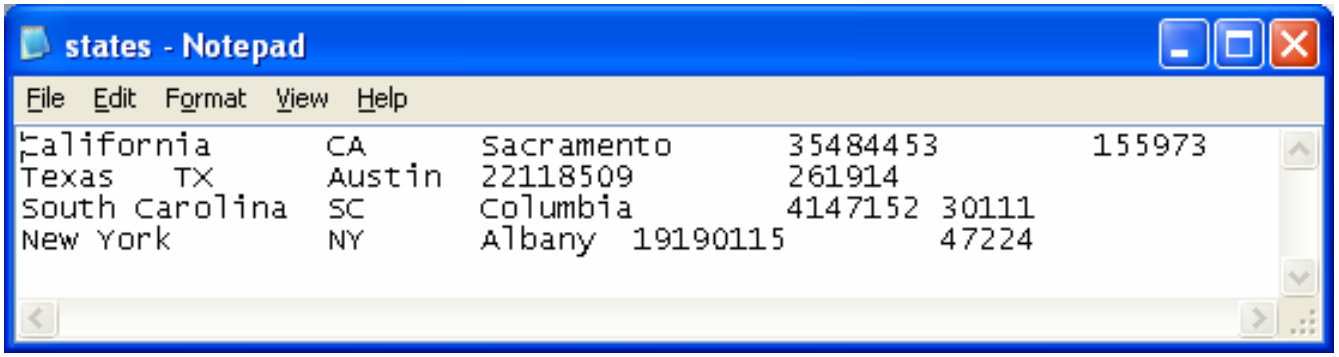
```
mysqlimport [options] database_name file1 file2 …
```

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

Importing Data Using the `mysqlimportUtility`

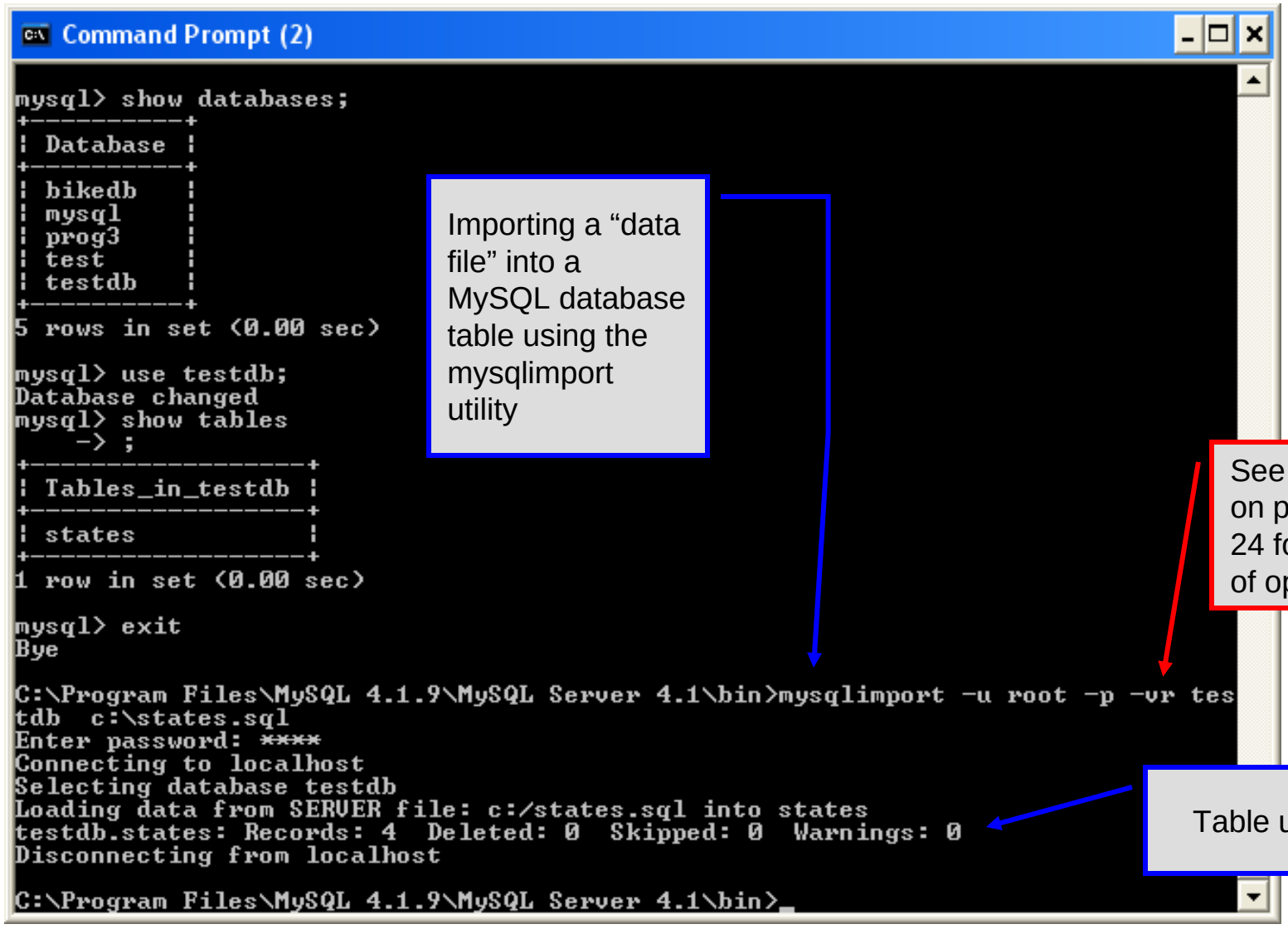
(cont.)

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



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

Importing Data Using the `mysqlimportUtility`



Importing Data Using the `mysqlimportUtility`

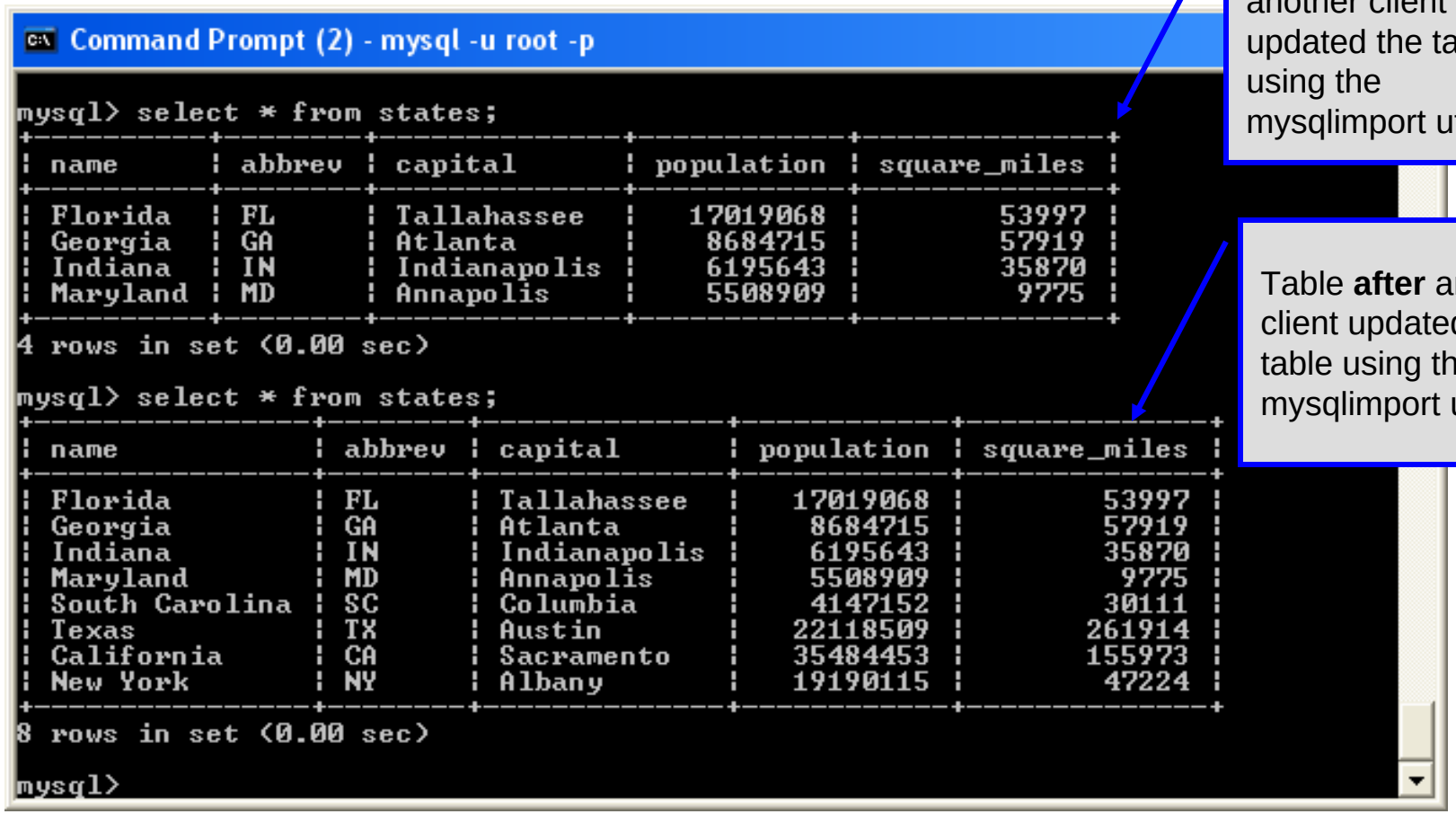


Table **before** another client updated the table using the mysqlimport utility.

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

mysqlimportUtility Options

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

mysqlimport Utility Options (cont.)

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

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

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

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

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

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

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

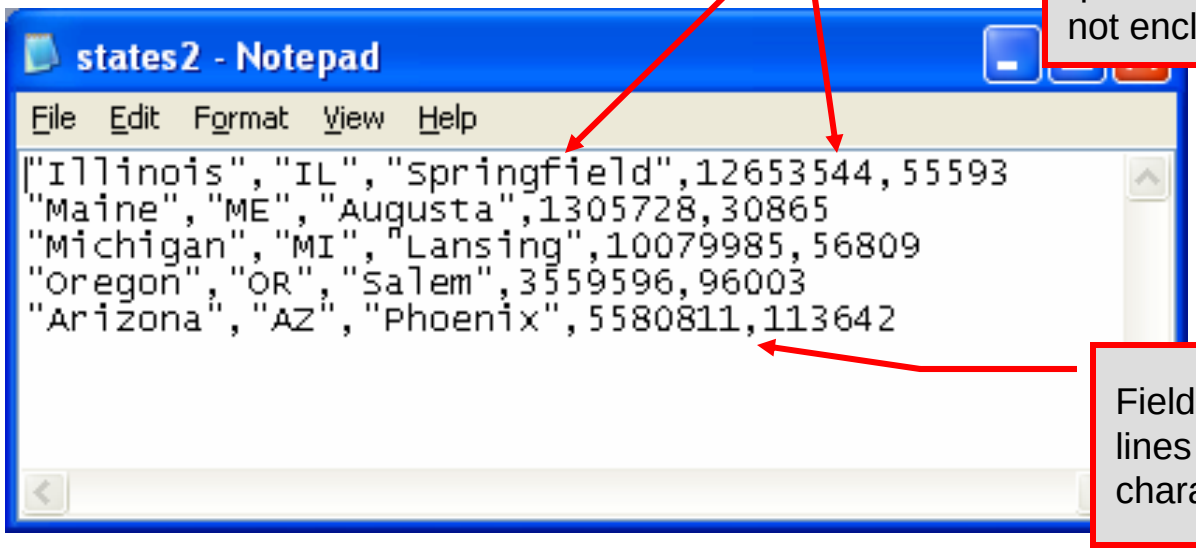
Same as –r and –i options in mysqlimport utility – either replace or ignore rows with duplicate keys.

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

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

Used to load only certain columns (not entire rows)

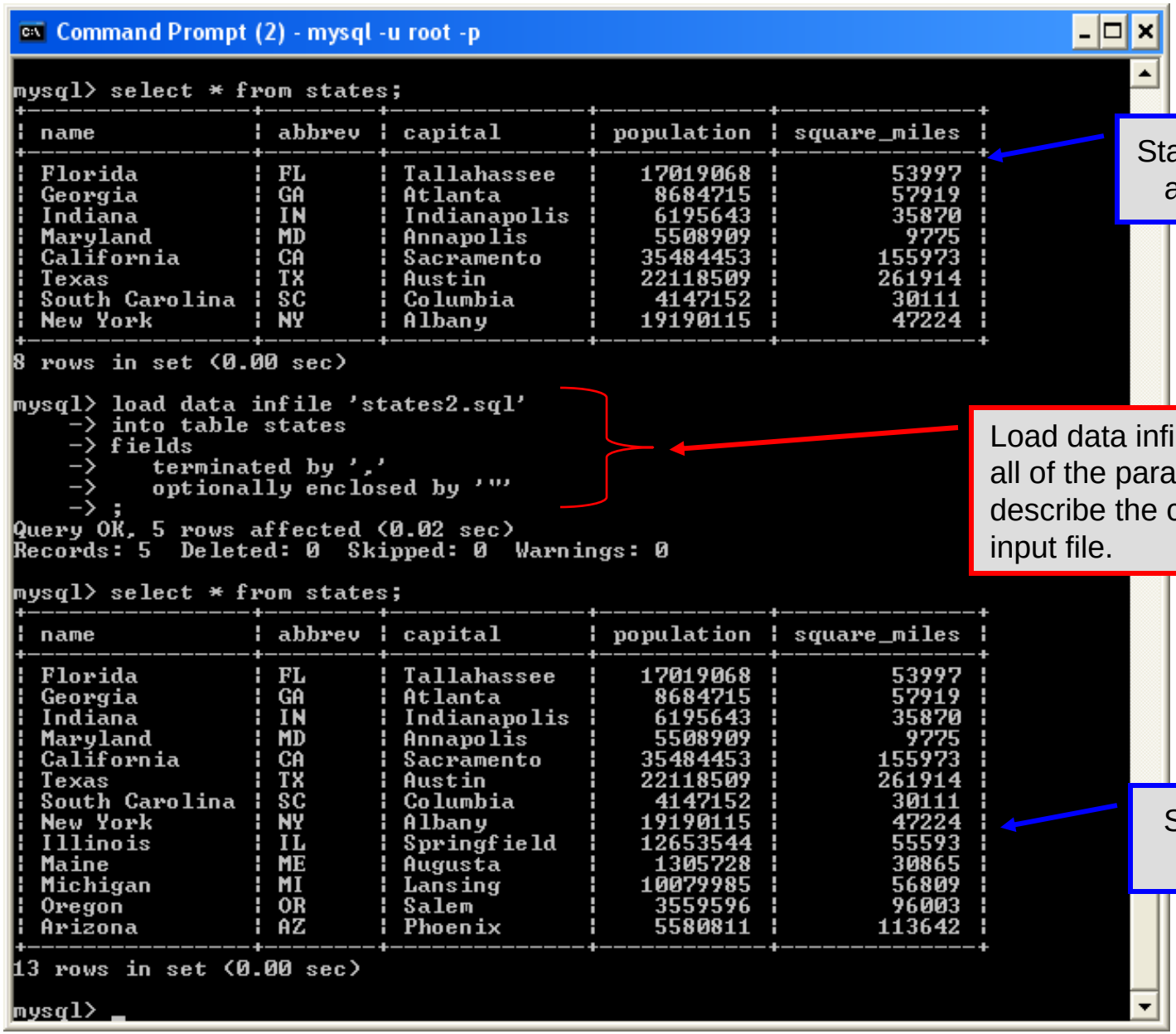
Load Data Infile Example



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

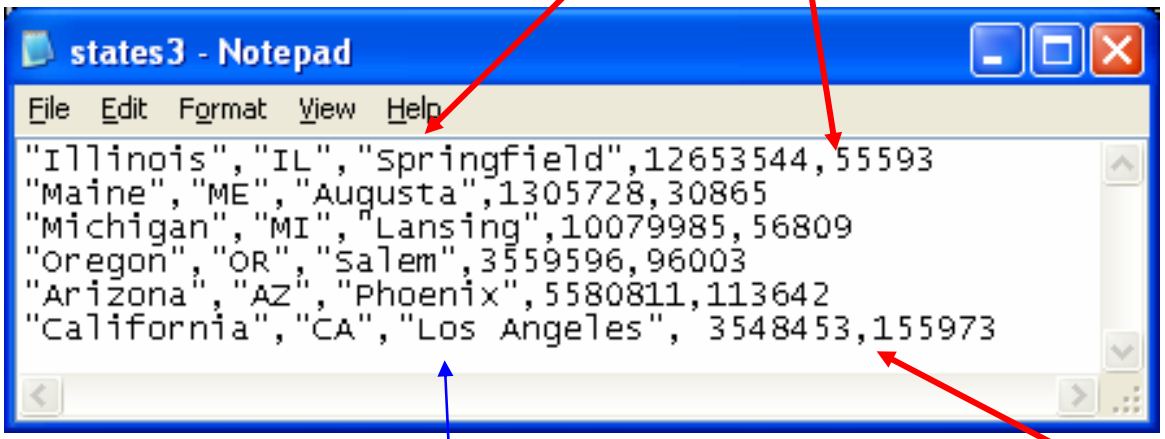
Fields are delimited by commas and lines are terminated by newline characters (an invisible \n)

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


Command Prompt (2) - mysql -u root -p
mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name | abbrev | capital | population | square_miles |
+----------------+--------+--------------+------------+--------------+
Florida	FL	Tallahassee	17019068	53997
Georgia	GA	Atlanta	8684715	57919
Indiana	IN	Indianapolis	6195643	35870
Maryland	MD	Annapolis	5508909	9775
California	CA	Sacramento	35484453	155973
Texas	TX	Austin	22118509	261914
South Carolina	SC	Columbia	4147152	30111
New York	NY	Albany	19190115	47224
+----------------+--------+--------------+------------+--------------+
8 rows in set (0.00 sec)
mysql> load data infile 'states2.sql'
-> into table states
-> fields
-> terminated by ','
-> optionally enclosed by '"'
-> ;
Query OK, 5 rows affected (0.02 sec)
Records: 5 Deleted: 0 Skipped: 0 Warnings: 0
mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name | abbrev | capital | population | square_miles |
+----------------+--------+--------------+------------+--------------+
Florida	FL	Tallahassee	17019068	53997
Georgia	GA	Atlanta	8684715	57919
Indiana	IN	Indianapolis	6195643	35870
Maryland	MD	Annapolis	5508909	9775
California	CA	Sacramento	35484453	155973
Texas	TX	Austin	22118509	261914
South Carolina	SC	Columbia	4147152	30111
New York	NY	Albany	19190115	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

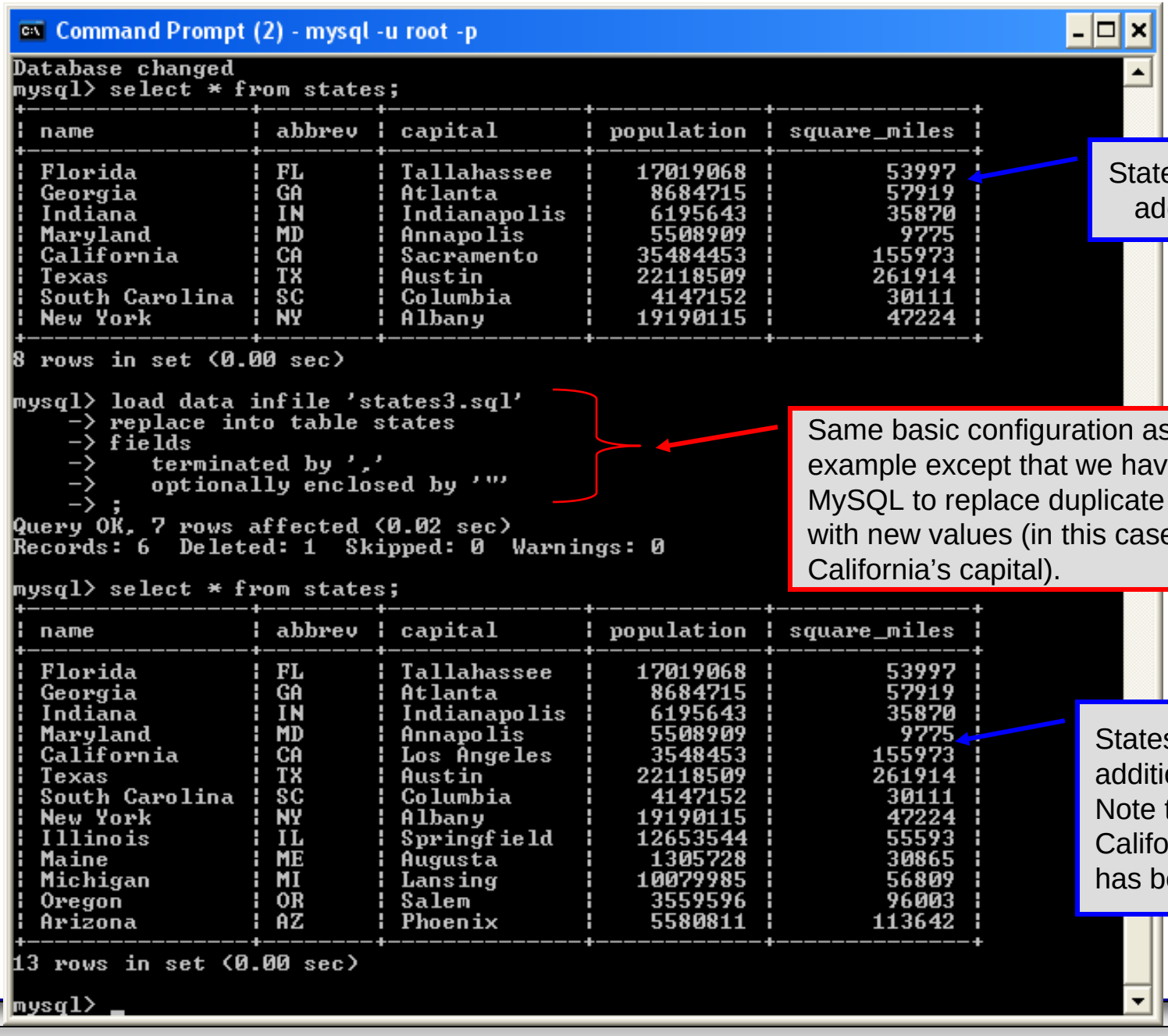
String fields are enclosed by double quotes in this file. Numeric values are not enclosed in quotes.

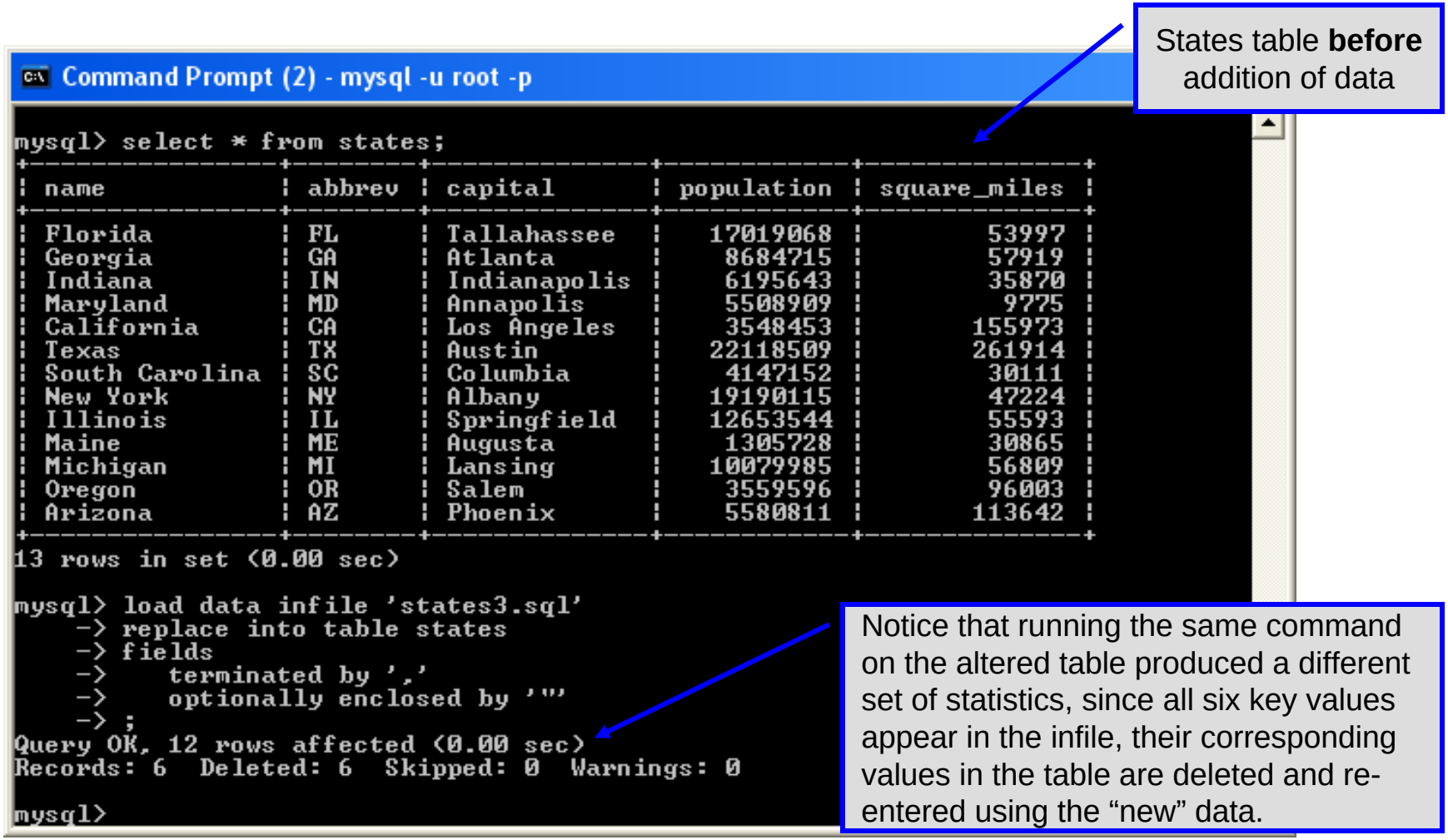


Text file containing the data to be loaded into the database table.

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

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


Command Prompt (2) - mysql -u root -p
Database changed
mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name | abbrev | capital | population | square_miles |
+----------------+--------+--------------+------------+--------------+
Florida	FL	Tallahassee	17019068	53997
Georgia	GA	Atlanta	8684715	57919
Indiana	IN	Indianapolis	6195643	35870
Maryland	MD	Annapolis	5508909	9775
California	CA	Sacramento	35484453	155973
Texas	TX	Austin	22118509	261914
South Carolina	SC	Columbia	4147152	30111
New York	NY	Albany	19190115	47224
+----------------+--------+--------------+------------+--------------+
8 rows in set (0.00 sec)
mysql> load data infile 'states3.sql'
-> replace into table states
-> fields
-> terminated by ','
-> optionally enclosed by '"'
-> ;
Query OK, 7 rows affected (0.02 sec)
Records: 6 Deleted: 1 Skipped: 0 Warnings: 0
mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name | abbrev | capital | population | square_miles |
+----------------+--------+--------------+------------+--------------+
Florida	FL	Tallahassee	17019068	53997
Georgia	GA	Atlanta	8684715	57919
Indiana	IN	Indianapolis	6195643	35870
Maryland	MD	Annapolis	5508909	9775
California	CA	Los Angeles	3548453	155973
Texas	TX	Austin	22118509	261914
South Carolina	SC	Columbia	4147152	30111
New York	NY	Albany	19190115	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!


Command Prompt (2) - mysql -u root -p
mysql> select * from states;
+----------------+--------+--------------+------------+--------------+
| name | abbrev | capital | population | square_miles |
+----------------+--------+--------------+------------+--------------+
Florida	FL	Tallahassee	17019068	53997
Georgia	GA	Atlanta	8684715	57919
Indiana	IN	Indianapolis	6195643	35870
Maryland	MD	Annapolis	5508909	9775
California	CA	Los Angeles	3548453	155973
Texas	TX	Austin	22118509	261914
South Carolina	SC	Columbia	4147152	30111
New York	NY	Albany	19190115	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 'states3.sql'
-> 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>
States table **before** addition of data
Notice that running the same command on the altered table produced a different set of statistics, since all six key values appear in the infile, their corresponding values in the table are deleted and re-entered using the "new" data.

The `Ignore` Clause of the Insert Command

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

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

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

Low Priority and Delayed Inserts

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

Inserting/Replacing Data Using Replace

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

Form 1

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

Form 2

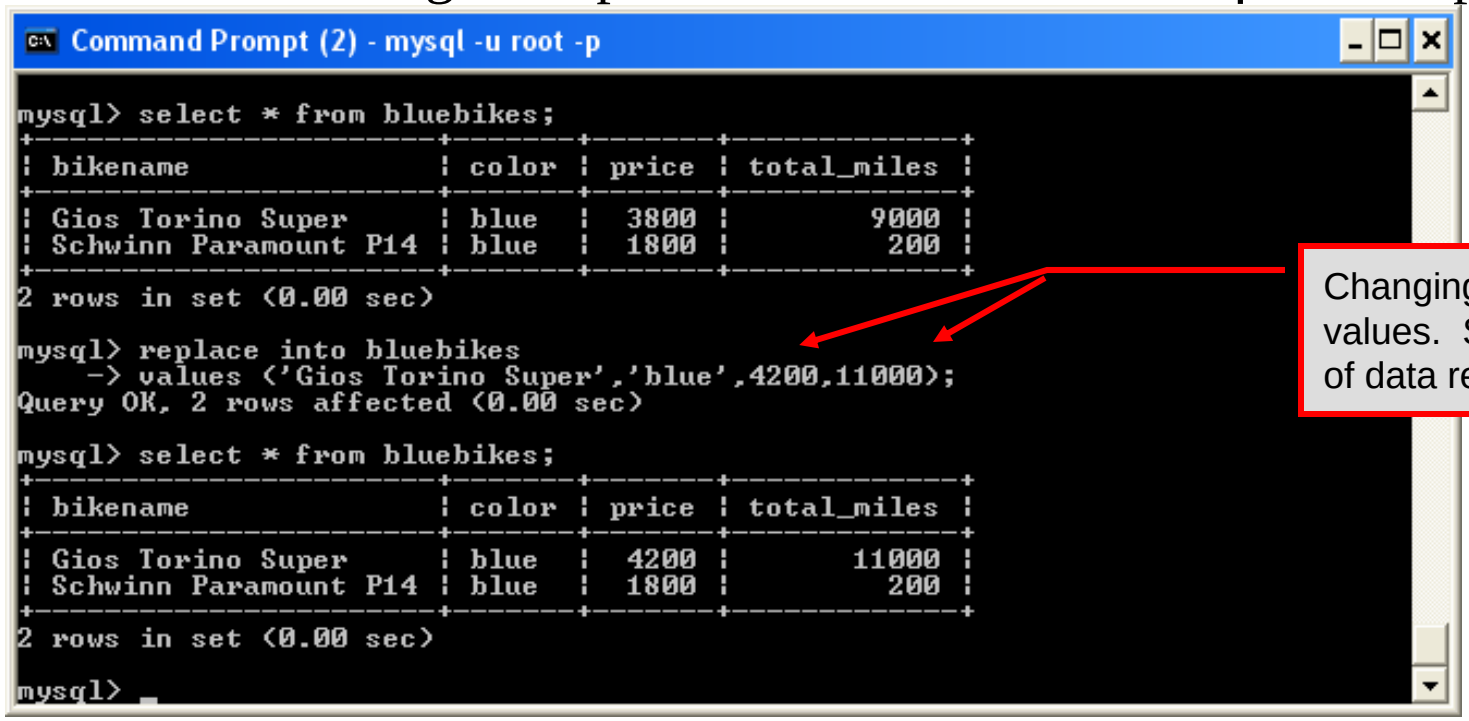
```
replace [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)]values (expression,…), (…)…
```

Form 3

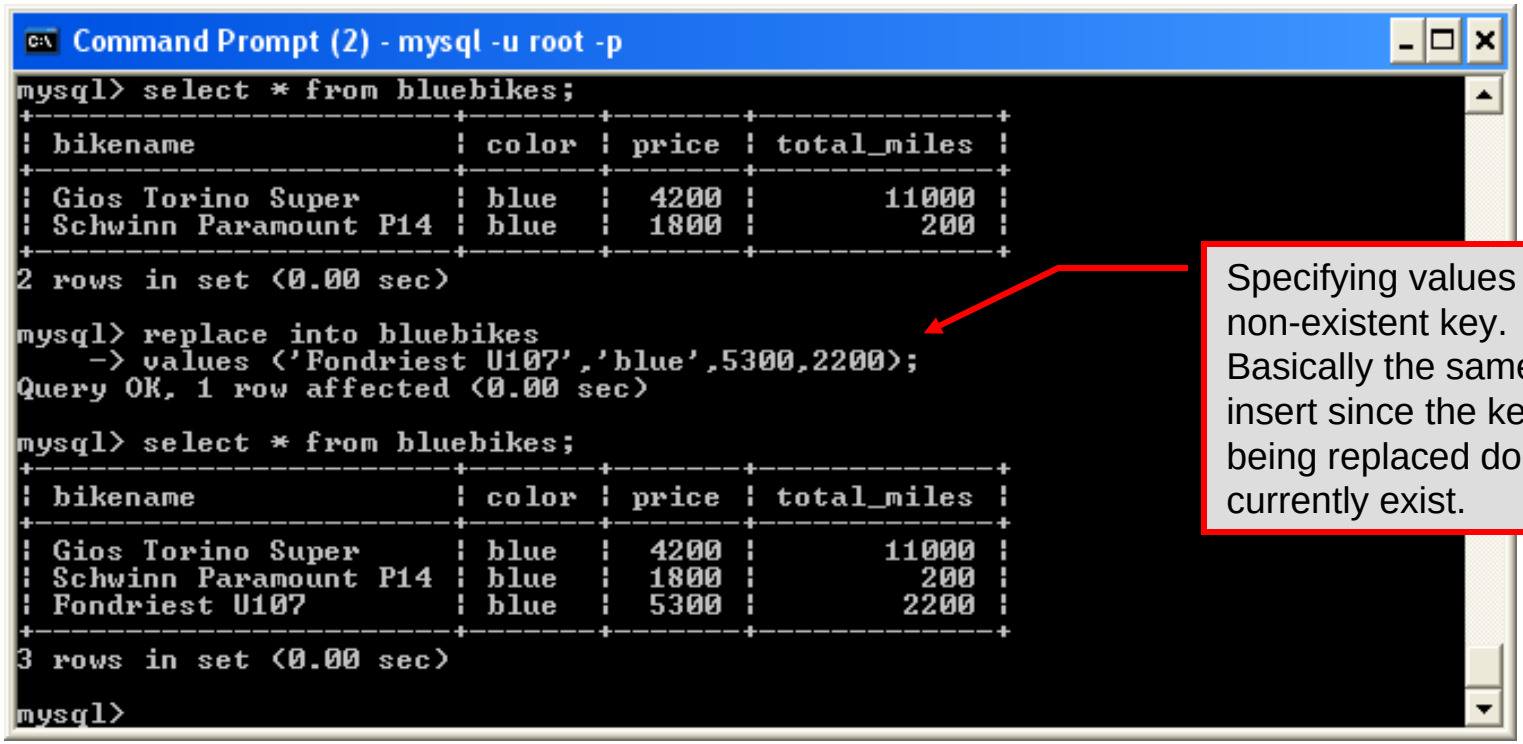
```
replace [low priority | delayed] [ignore] [into]table_name
        [(column_name,…)] select…
```

Using replace

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



Using Replace (cont.)



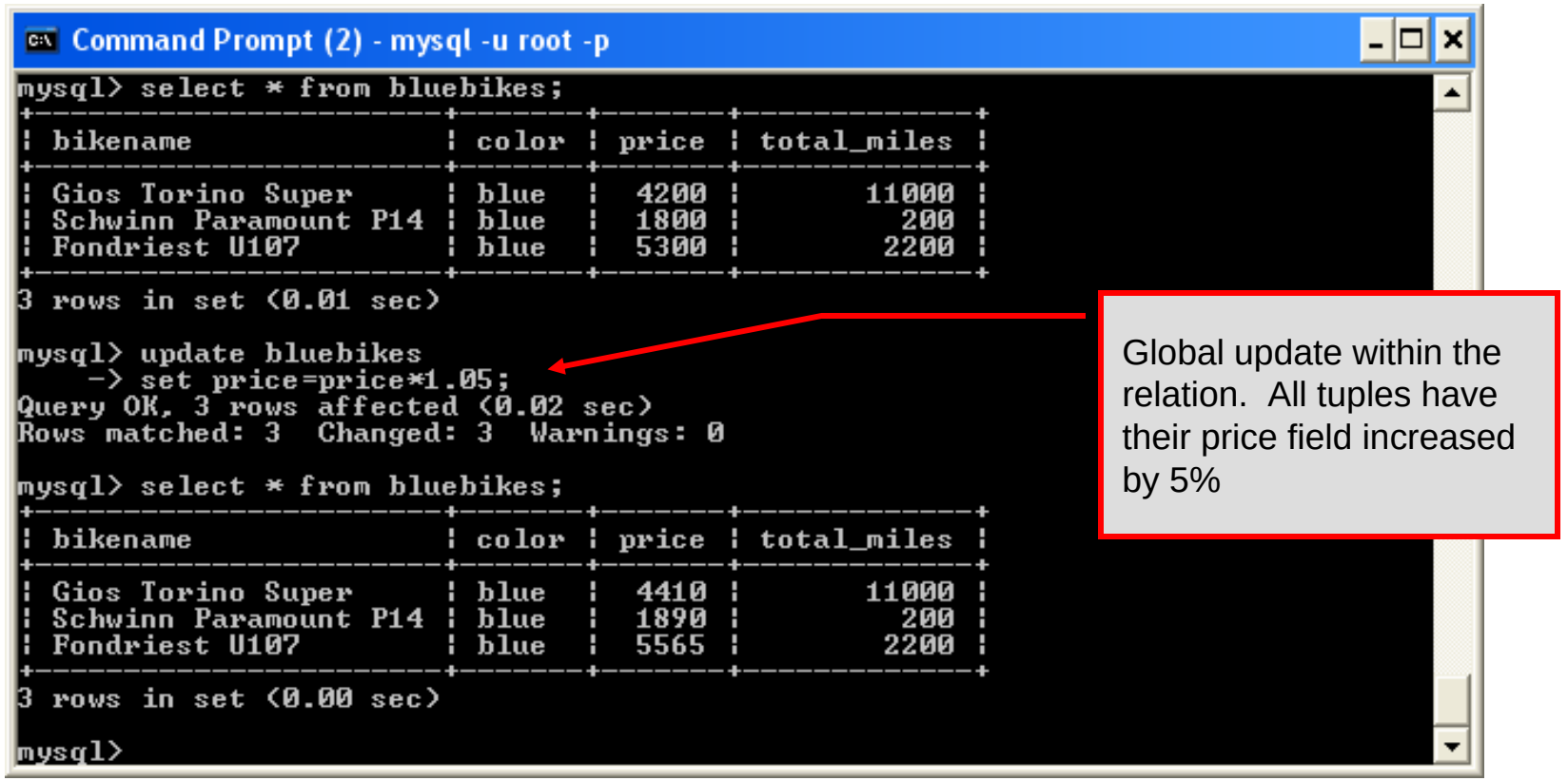
Performing Updates on Tables

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

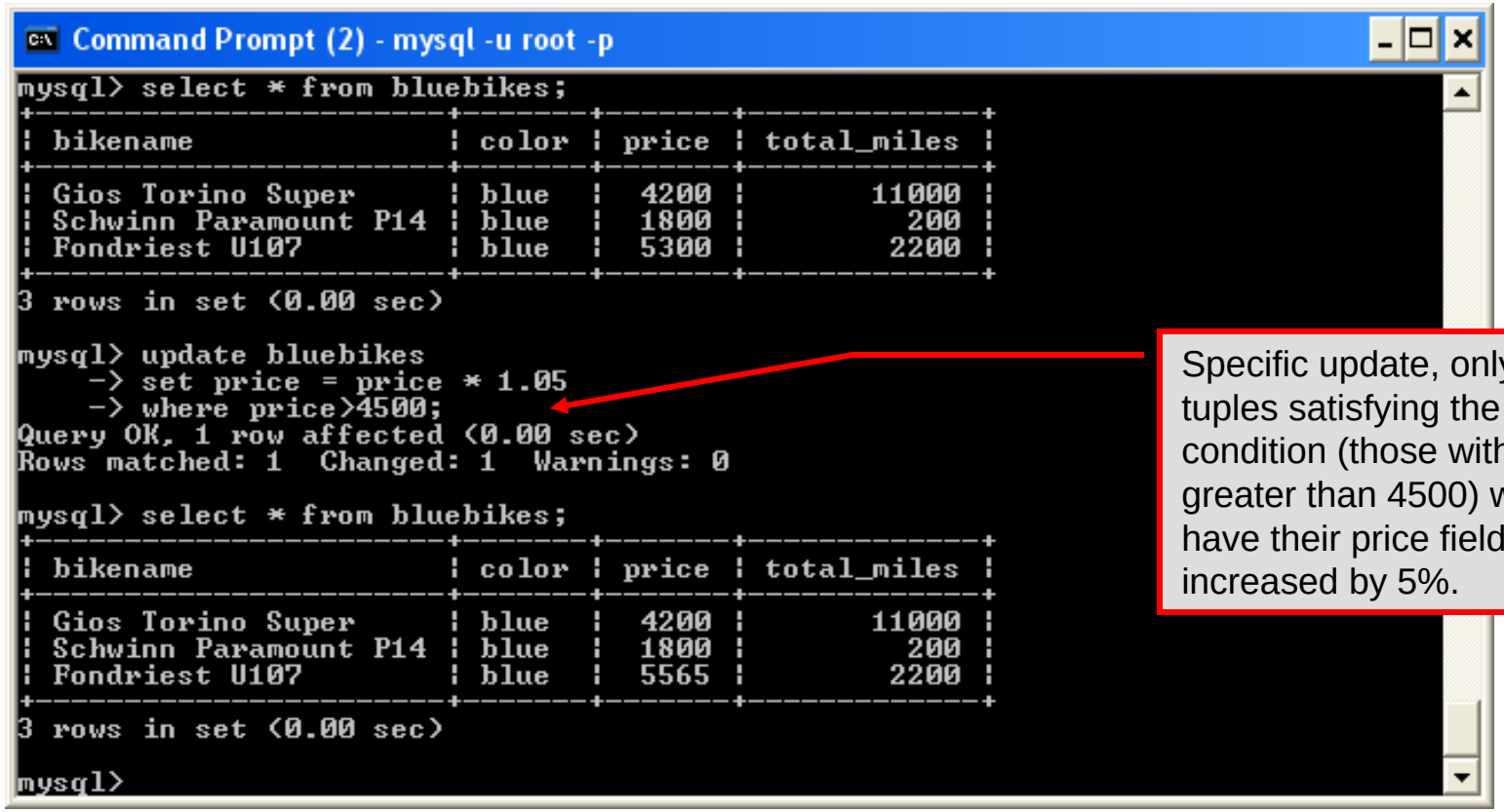
```
update [low priority] [ignore] table_name
    set column_name1 = expression1,
        column_name2 = expression2, …
    [where where_definition]
    [limit num];
```

- 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.)



Using update (cont.)



Select Queries in MySQL

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

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

MySQL RDBMS (cont.)

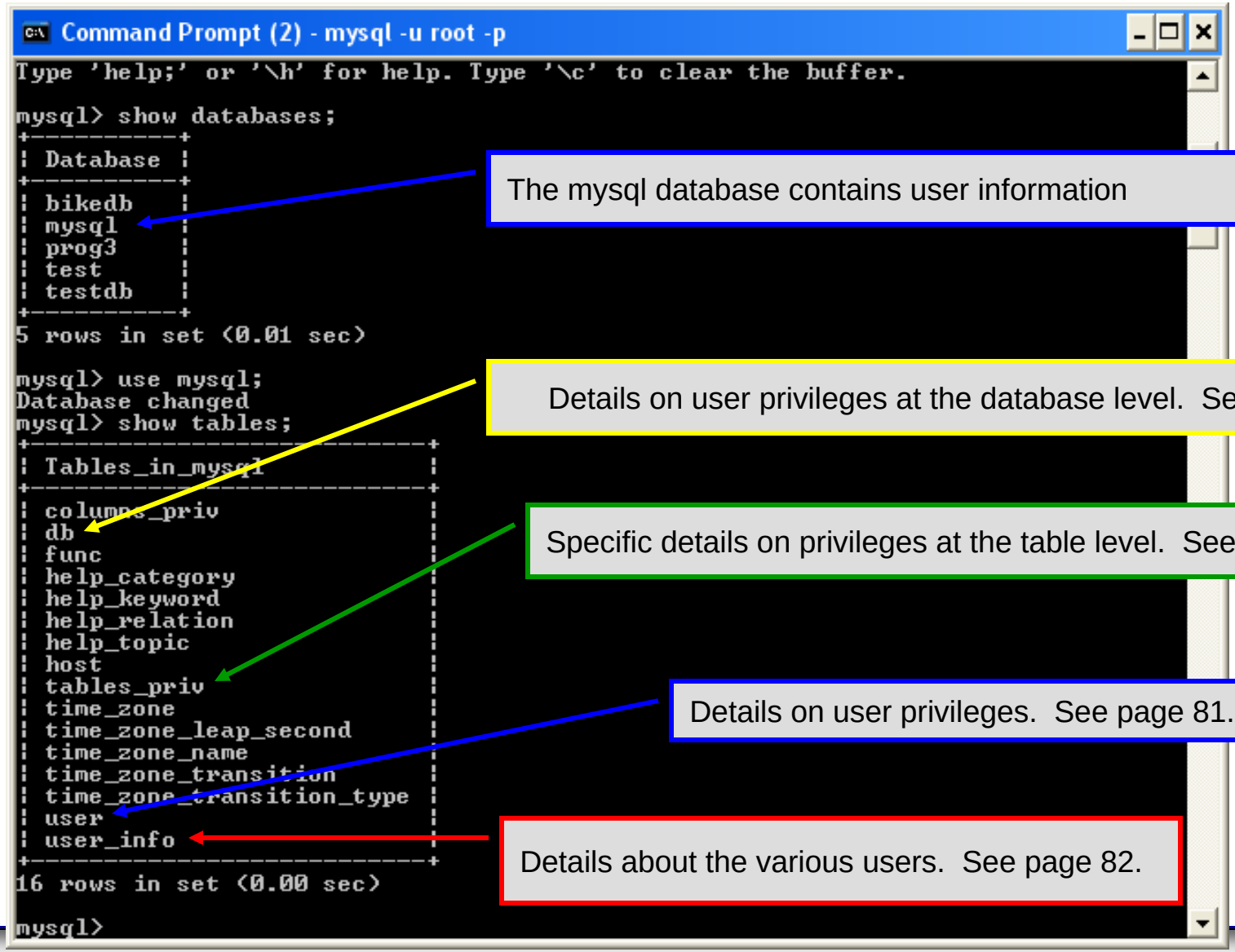
- MySQL features a user permissions system, which allows control over user's access to the databases under MySQL control.
- There are very few competitors of MySQL (Oracle, Sybase, DB2, and SQL Server) that can match the level of sophistication provided by MySQL's permissions system in terms of granularity and level of security provided.

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

Authorization in MySQL

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

Tables in the mysql Database



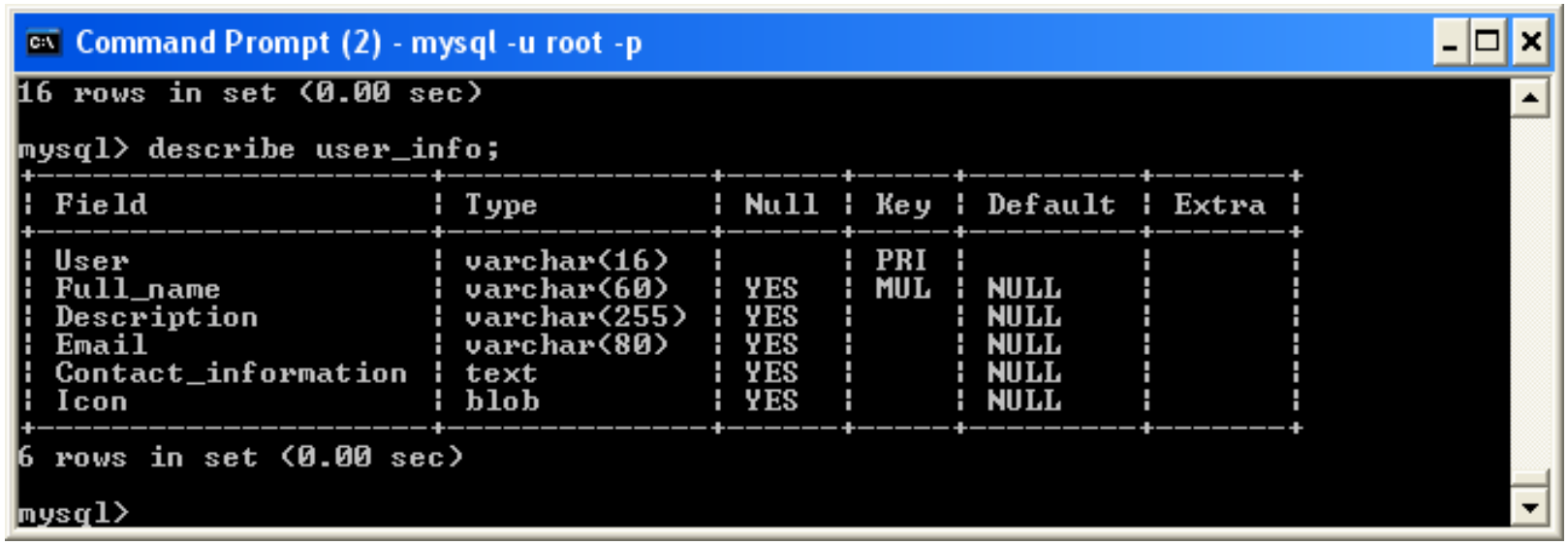
Contents of the user Table

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



Contents of the `tables_priv` Table

outt; - Notepad

File Edit Format View Help

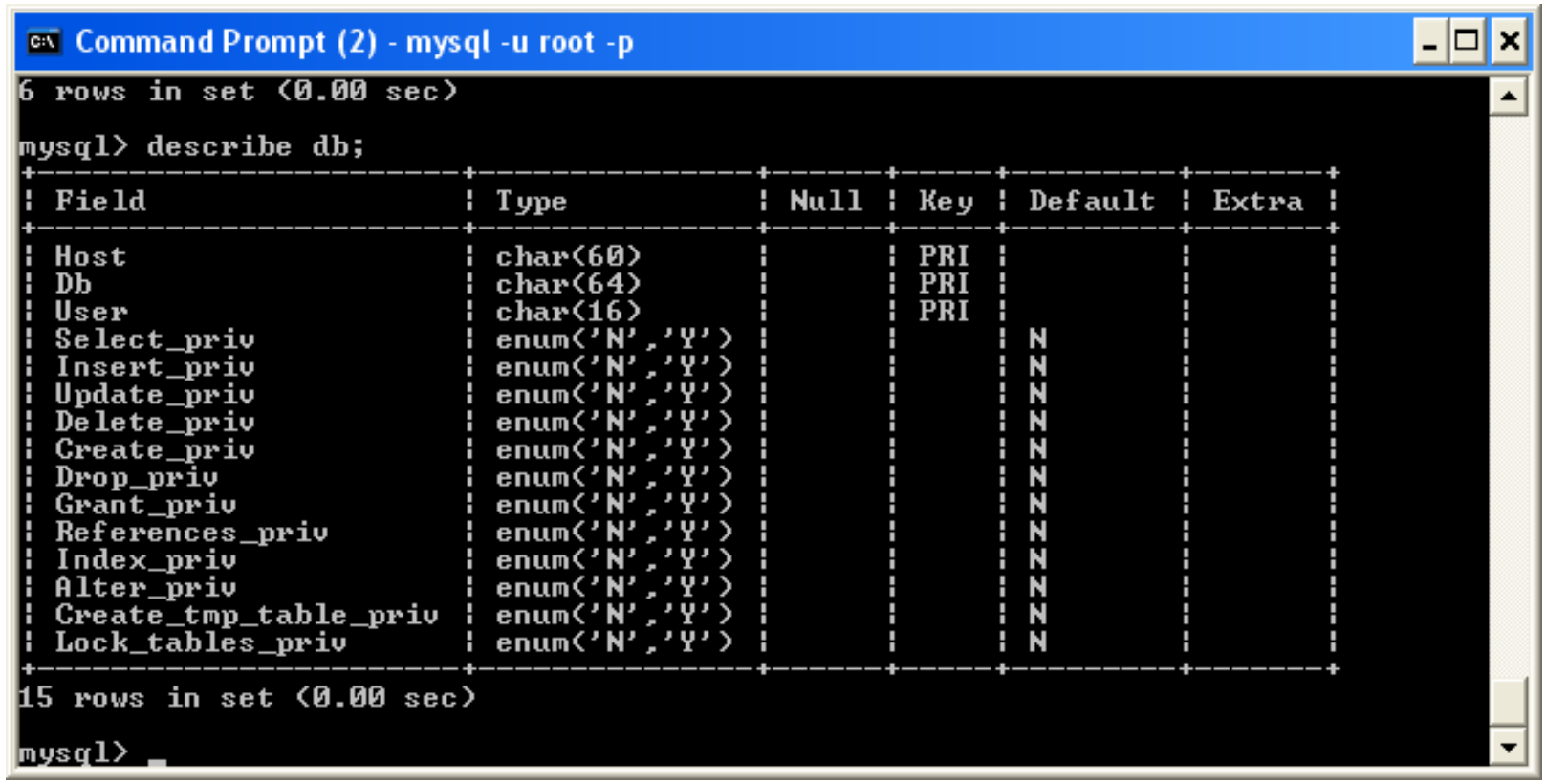
```
mysql> \t;
mysql> describe tables_priv;
+-------------+-----------------------------------------------------------------------------------------------+
| Field       | Type                                                                                          |
+-------------+---------------------------------------------------------------------------------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

mysql
```

outt; - Notepad

File Edit Format View Help

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

Contents of the db Table



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

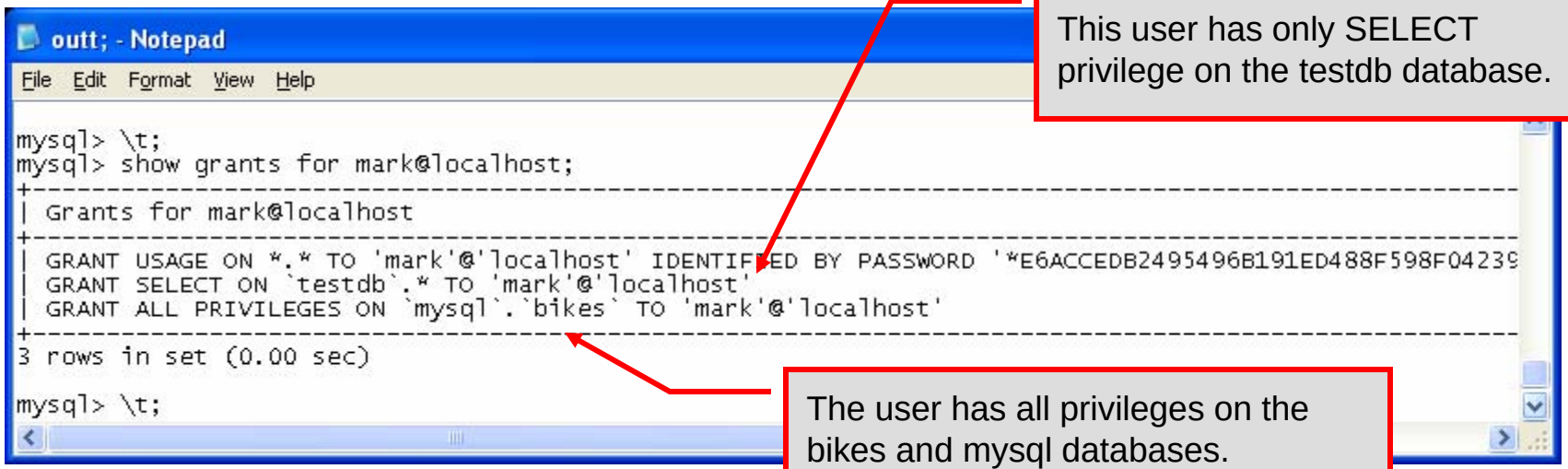
Privilege	Operations Permitted
ALL or ALL PRIVILEGES	All privileges except for GRANT
ALTER	Change a table definition using ALTER TABLE excluding the creation and dropping of indices.
CREATE	Create database or tables within a database.
CREATE TEMPORARY TABLES	Create temporary tables.
DELETE	Ability to 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:



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

outt; - Notepad

File Edit Format View Help

```
mysql> show grants for mark@localhost;
+-----------------------------------------------------------------------------------------------------------------------------------+
| Grants for mark@localhost
+-----------------------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'mark'@'localhost' IDENTIFIED BY PASSWORD '*E6ACCEDB2495496B191ED488F598F04239C85E73'
| GRANT ALL PRIVILEGES ON `mysql`.`bikes` TO 'mark'@'localhost'
| GRANT SELECT ON `testdb`.`states` TO 'mark'@'localhost'
+-----------------------------------------------------------------------------------------------------------------------------------+
3 rows in set (0.00 sec)

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

mysql> show grants for mark@localhost;
+-----------------------------------------------------------------------------------------------------------------------------------+
| Grants for mark@localhost
+-----------------------------------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'mark'@'localhost' IDENTIFIED BY PASSWORD '*E6ACCEDB2495496B191ED488F598F04239C85E73'
| GRANT ALL PRIVILEGES ON `mysql`.`bikes` TO 'mark'@'localhost'
+-----------------------------------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> \t;
```

User has SELECT privilege on testdb.states table.

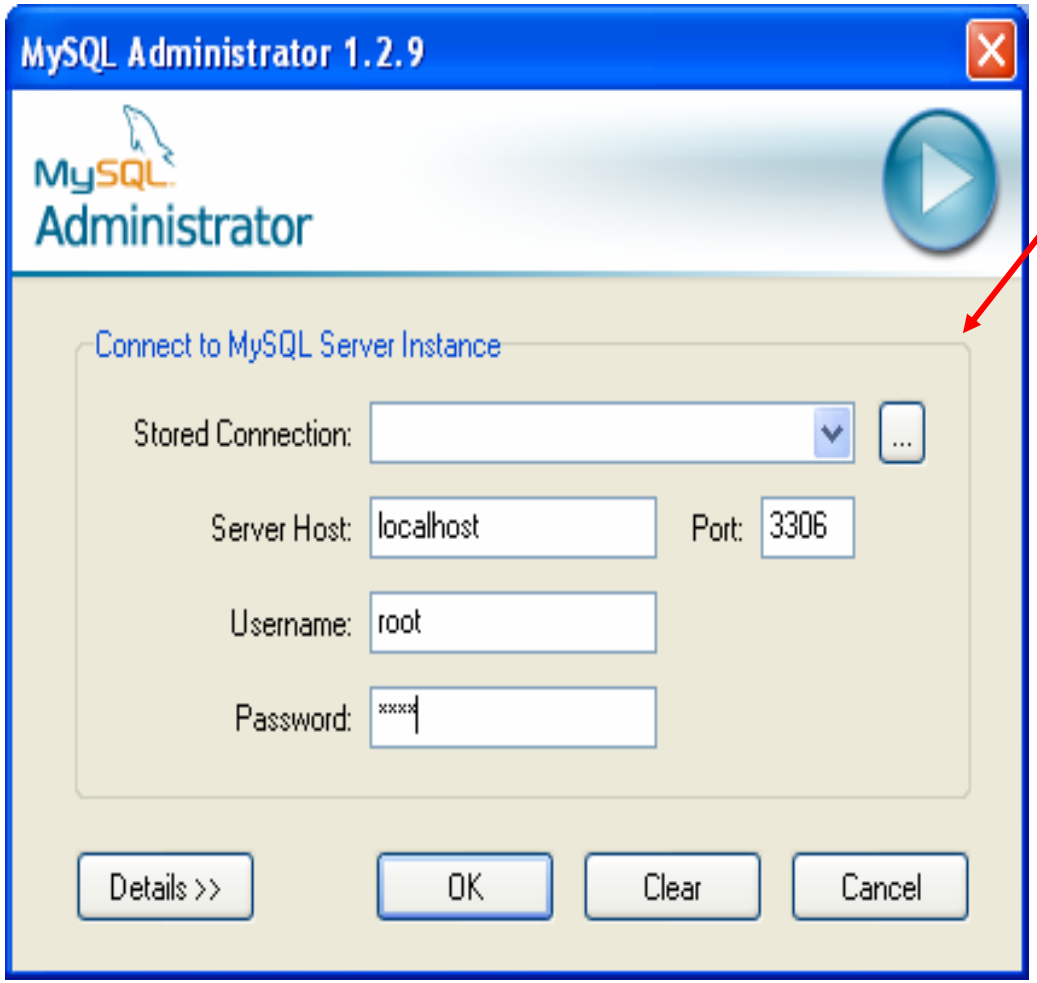
Revoking user's SELECT privilege on testdb.states.

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

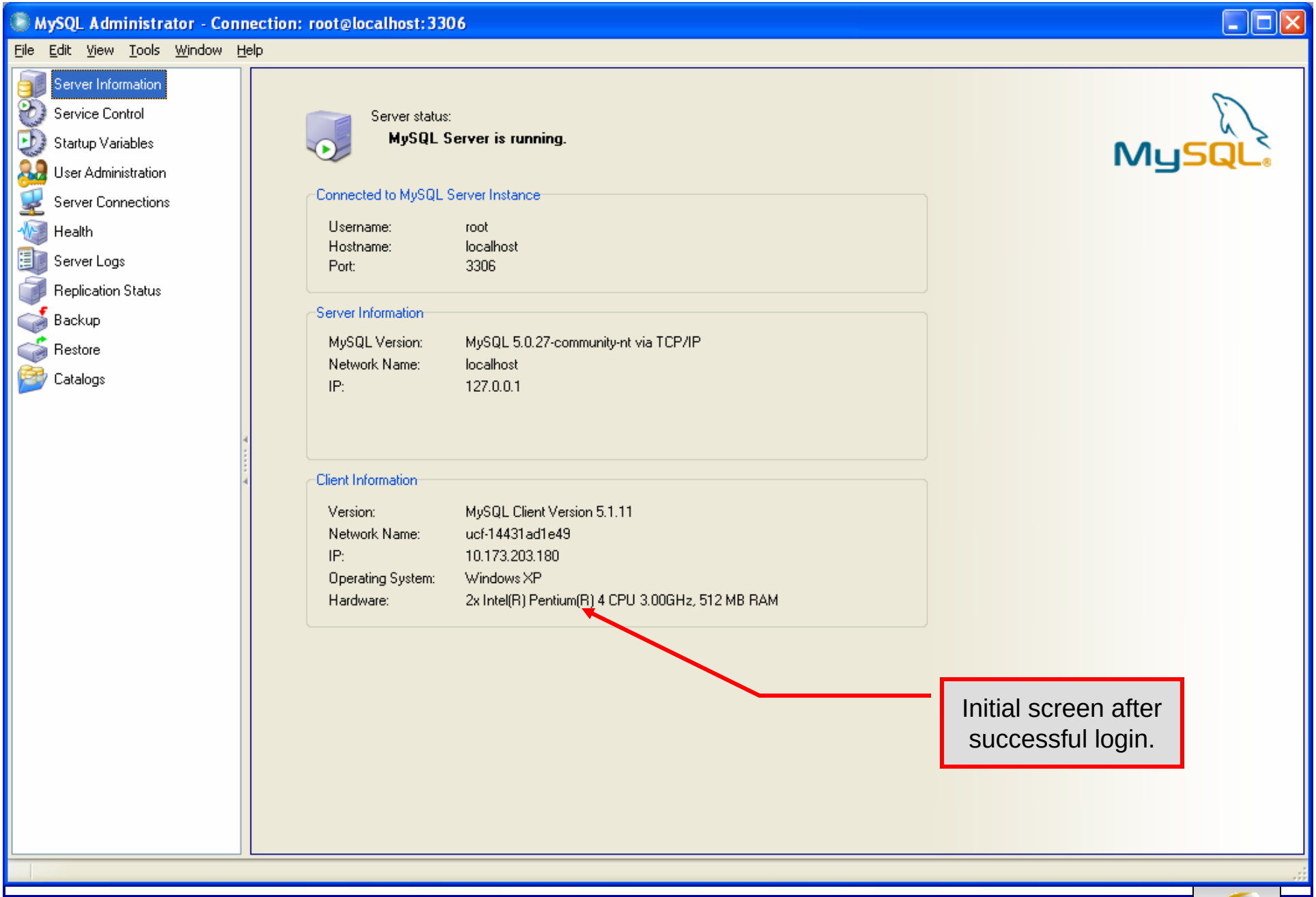
The MySQL Administrator Tool

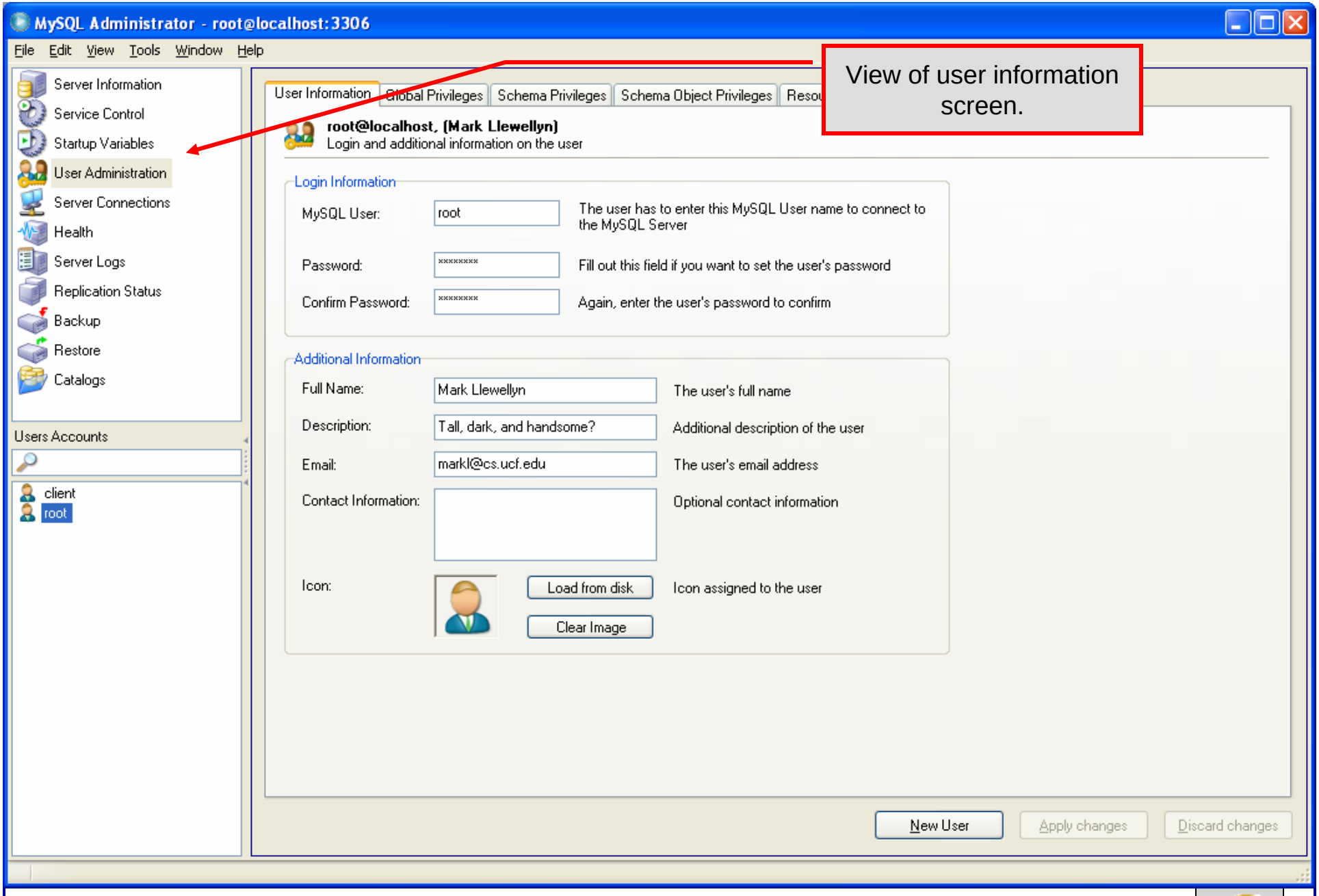
- From MySQL you can download a GUI-based administrator tool to help you administer your MySQL databases.
- This tool implements all of the GRANT, REVOKE, and SHOW functionality available in SQL.
- This tool also contains some system administrator functionality for monitoring system resources and utilization.
- 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.

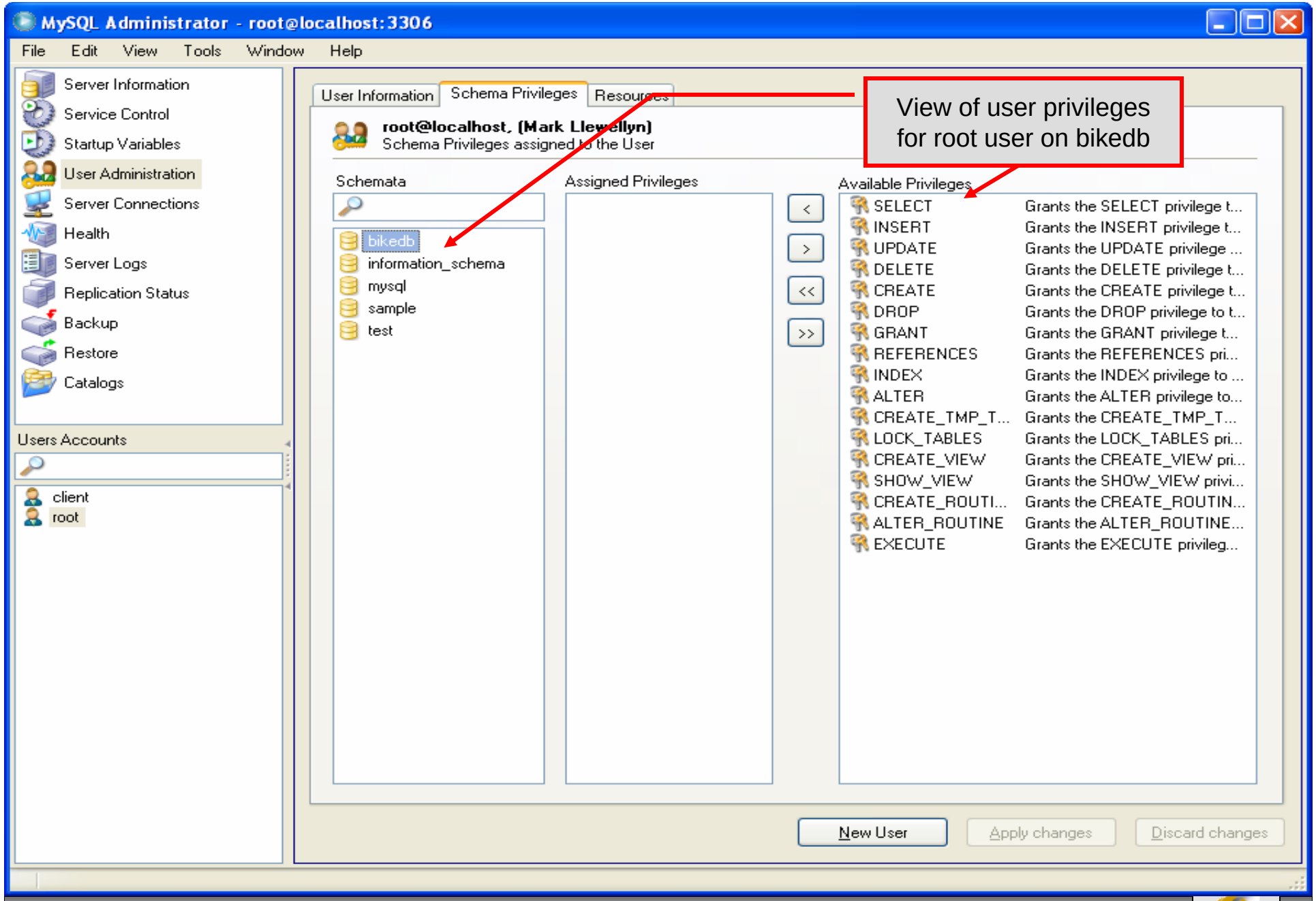
The MySQL Administrator Tool – Screen Shots

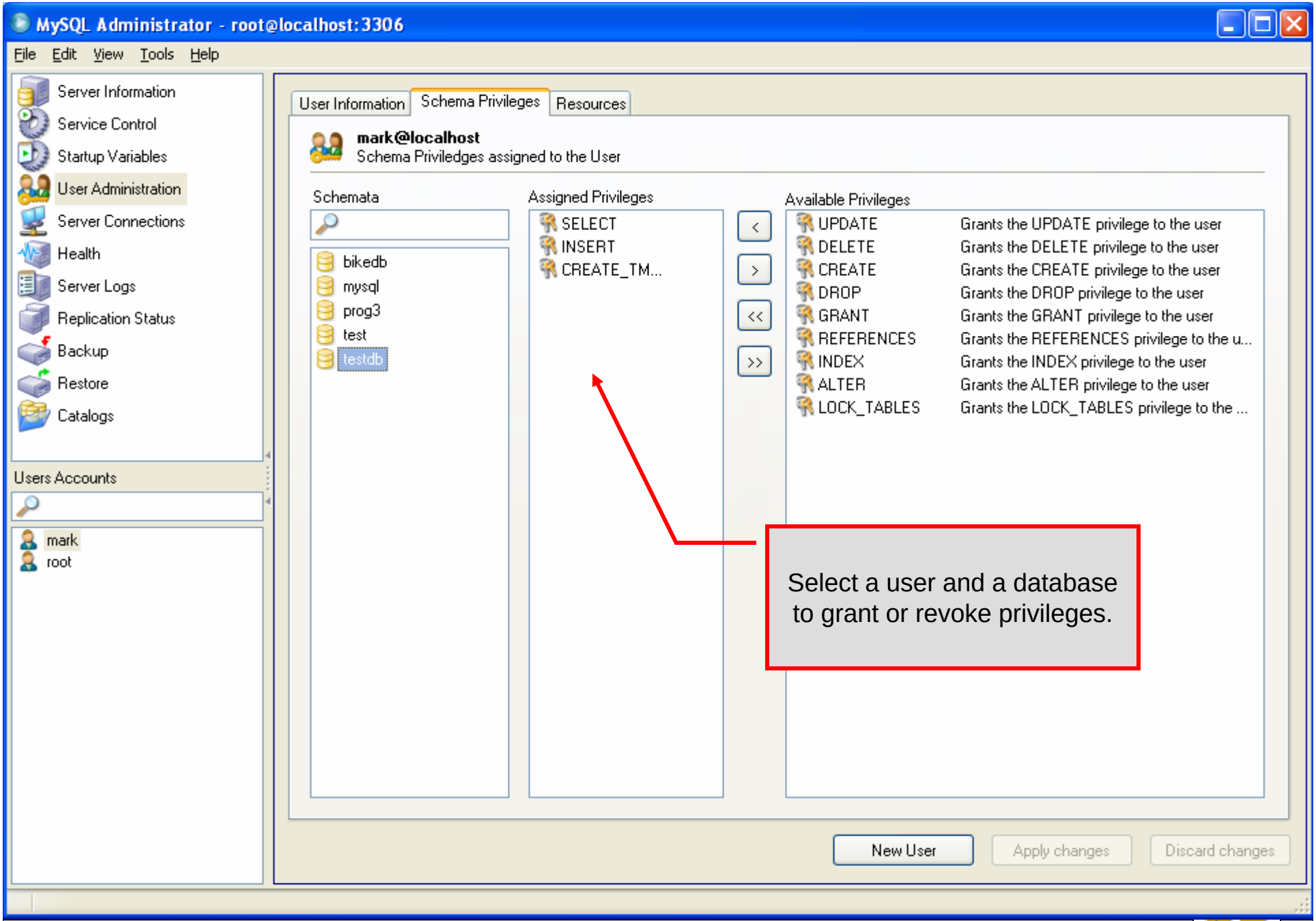


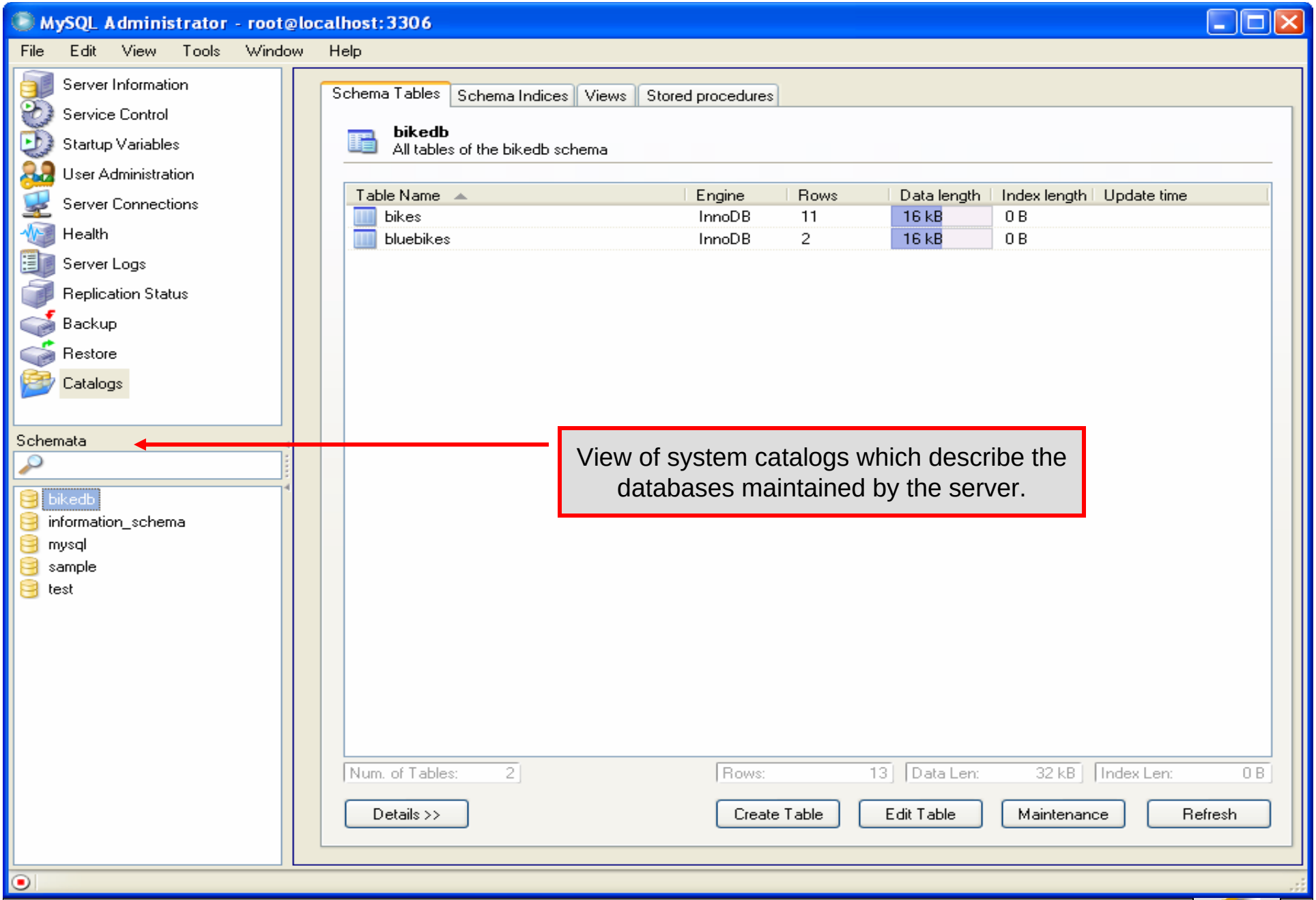
Initial login screen


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 5.0.27-community-nt via TCP/IP
Network Name: localhost
IP: 127.0.0.1
Client Information
Version: MySQL Client Version 5.1.11
Network Name: ucf-14431ad1e49
IP: 10.173.203.180
Operating System: Windows XP
Hardware: 2x Intel(R) Pentium(R) 4 CPU 3.00GHz, 512 MB RAM
Initial screen after successful login.


MySQL Administrator - 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
client
root
User Information
Global Privileges
Schema Privileges
Schema Object Privileges
root@localhost, (Mark Llewellyn)
Login and additional information on the user
Login Information
MySQL User:
root
The user has to enter this MySQL User name to connect to the MySQL Server
Password:
Fill out this field if you want to set the user's password
Confirm Password:
Again, enter the user's password to confirm
Additional Information
Full Name:
Mark Llewellyn
The user's full name
Description:
Tall, dark, and handsome?
Additional description of the user
Email:
markl@cs.ucf.edu
The user's email address
Contact Information:
Optional contact information
Icon:
Load from disk
Clear Image
Icon assigned to the user
New User
Apply changes
Discard changes
View of user information screen.


MySQL Administrator - 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
client
root
User Information
Schema Privileges
Resources
root@localhost, (Mark Llewellyn)
Schema Privileges assigned to the User
Schemata
bikedb
information_schema
mysql
sample
test
Assigned Privileges
Available Privileges
SELECT Grants the SELECT privilege t...
INSERT Grants the INSERT privilege t...
UPDATE Grants the UPDATE privilege ...
DELETE Grants the DELETE privilege t...
CREATE Grants the CREATE privilege t...
DROP Grants the DROP privilege to t...
GRANT Grants the GRANT privilege t...
REFERENCES Grants the REFERENCES pri...
INDEX Grants the INDEX privilege to ...
ALTER Grants the ALTER privilege to...
CREATE_TMP_T... Grants the CREATE_TMP_T...
LOCK_TABLES Grants the LOCK_TABLES pri...
CREATE_VIEW Grants the CREATE_VIEW pri...
SHOW_VIEW Grants the SHOW_VIEW privi...
CREATE_ROUTI... Grants the CREATE_ROUTIN...
ALTER_ROUTINE Grants the ALTER_ROUTINE...
EXECUTE Grants the EXECUTE privileg...
New User
Apply changes
Discard changes
View of user privileges for root user on bikedb

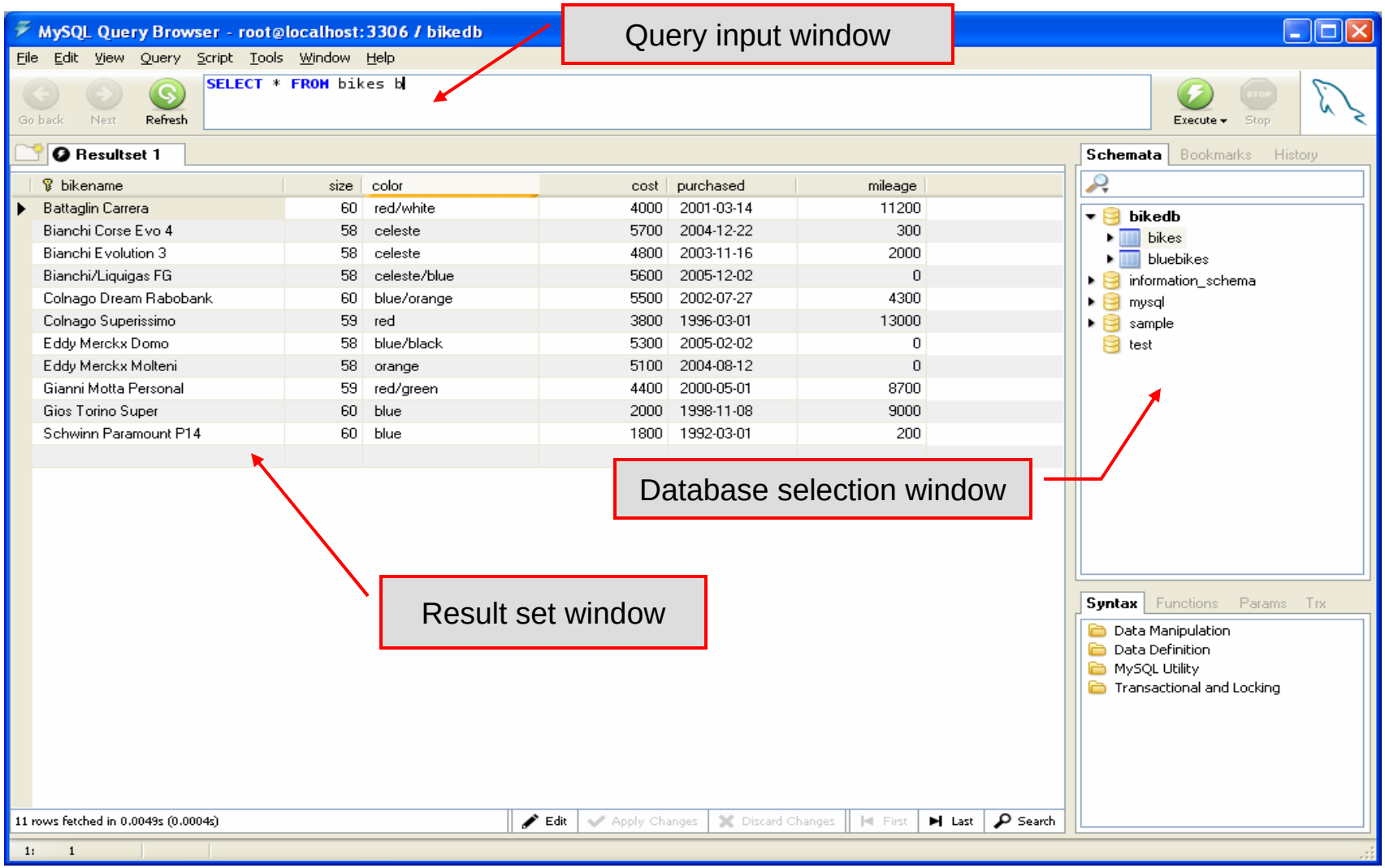

MySQL Administrator - root@localhost:3306
File Edit View Tools Help
Server Information
Service Control
Startup Variables
User Administration
Server Connections
Health
Server Logs
Replication Status
Backup
Restore
Catalogs
Users Accounts
mark
root
User Information
Schema Privileges
Resources
mark@localhost
Schema Priviledges assigned to the User
Schemata
bikedb
mysql
prog3
test
testdb
Assigned Privileges
SELECT
INSERT
CREATE_TM...
Available Privileges
UPDATE Grants the UPDATE privilege to the user
DELETE Grants the DELETE privilege to the user
CREATE Grants the CREATE privilege to the user
DROP Grants the DROP privilege to the user
GRANT Grants the GRANT privilege to the user
REFERENCES Grants the REFERENCES privilege to the u...
INDEX Grants the INDEX privilege to the user
ALTER Grants the ALTER privilege to the user
LOCK_TABLES Grants the LOCK_TABLES privilege to the ...
Select a user and a database to grant or revoke privileges.
New User
Apply changes
Discard changes

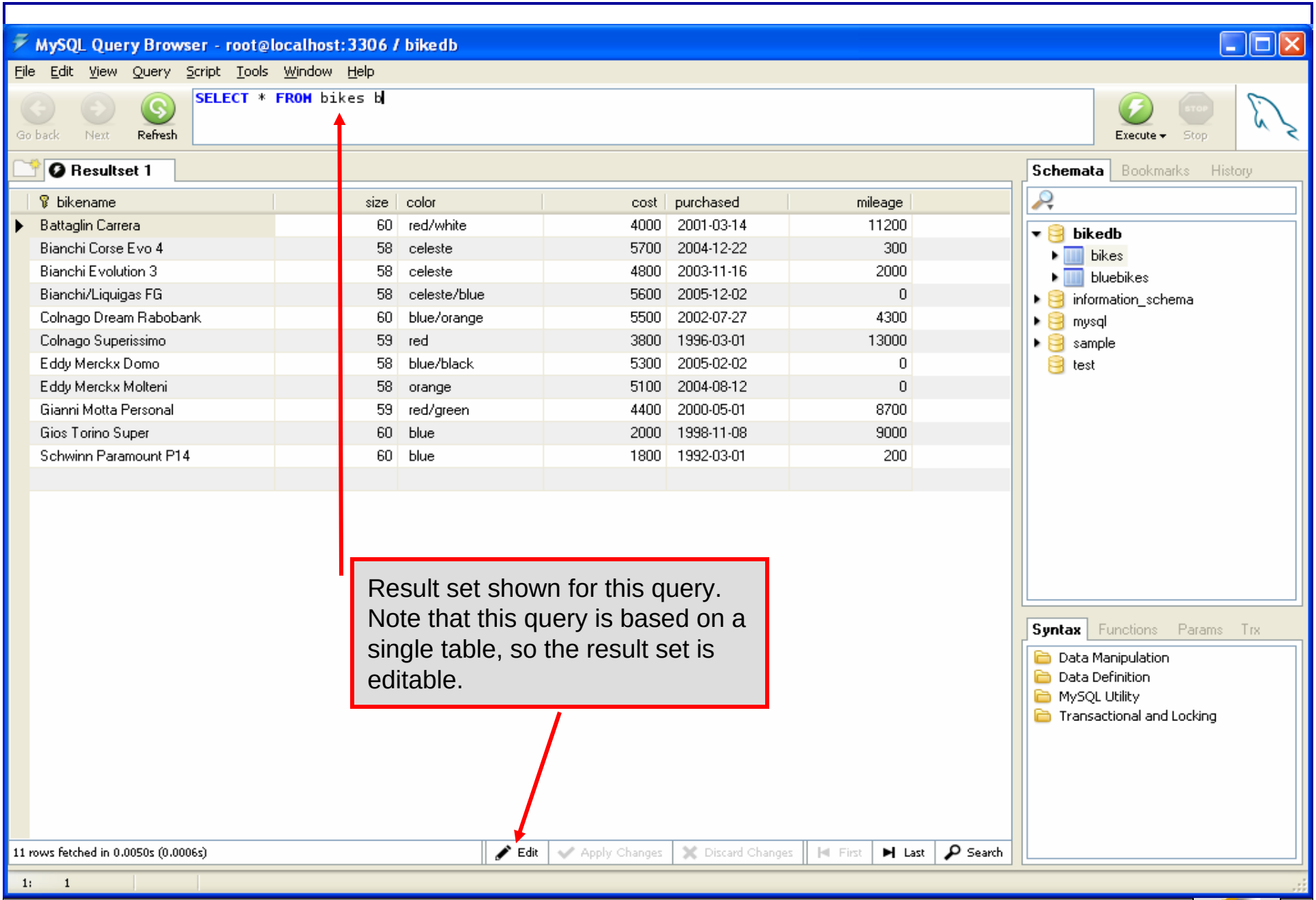

MySQL Administrator - root@localhost:3306
File Edit View Tools Window Help
Server Information
Service Control
Startup Variables
User Administration
Server Connections
Health
Server Logs
Replication Status
Backup
Restore
Catalogs
Schemata
bikedb
information_schema
mysql
sample
test
Schema Tables
Schema Indices
Views
Stored procedures
bikedb
All tables of the bikedb schema
Table Name
Engine
Rows
Data length
Index length
Update time
bikes
InnoDB
11
16 kB
0 B
bluebikes
InnoDB
2
16 kB
0 B
Num. of Tables: 2
Rows: 13
Data Len: 32 kB
Index Len: 0 B
Details >>
Create Table
Edit Table
Maintenance
Refresh
View of system catalogs which describe the databases maintained by the server.

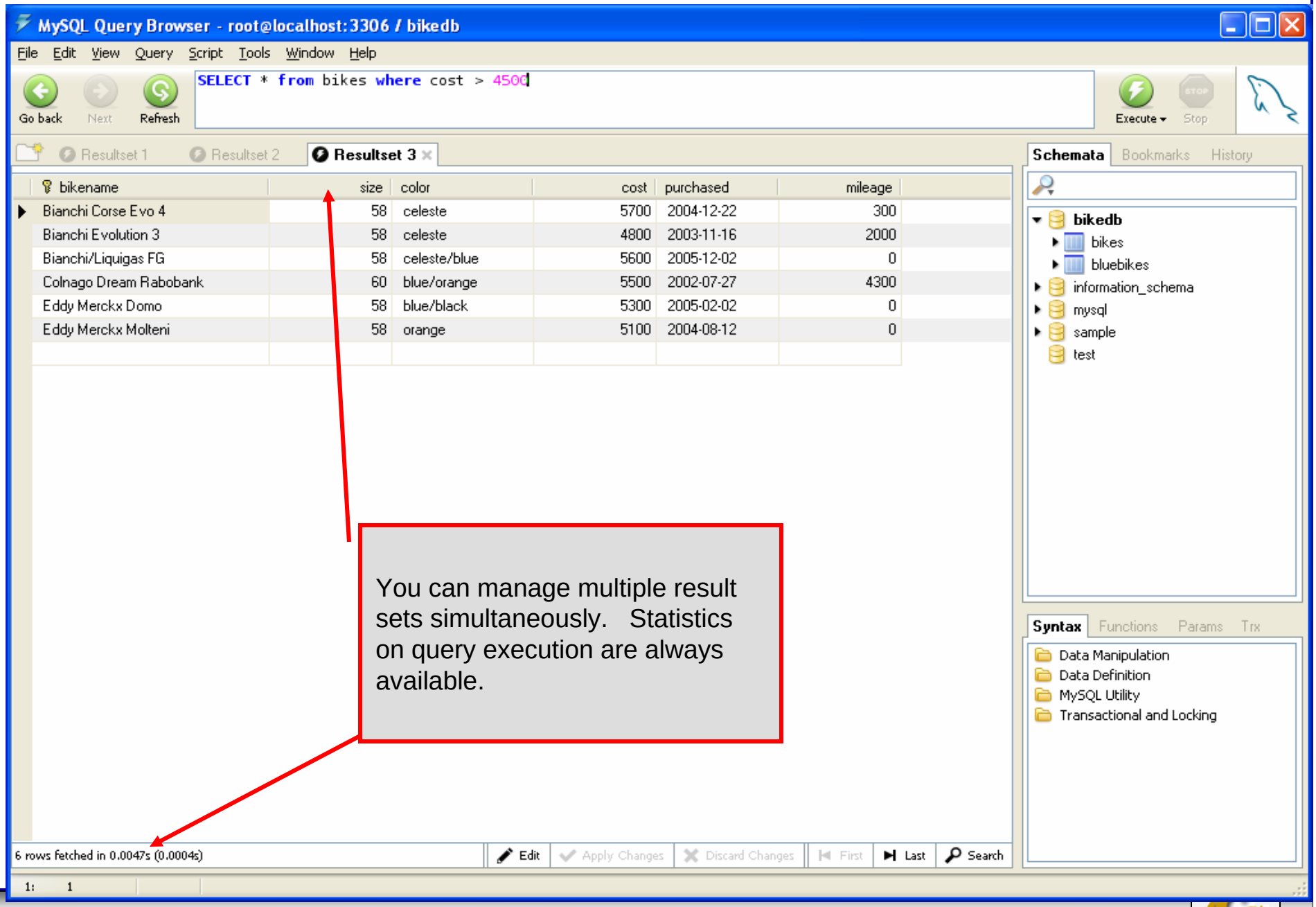
The MySQL Query Browser Tool

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

A First Look At The MySQL Query Browser




MySQL Query Browser - root@localhost:3306 / bikedb
File Edit View Query Script Tools Window Help
SELECT * FROM bikes b
Go back
Next
Refresh
Execute
Stop
Resultset 1
bikename size color cost purchased mileage
Battaglin Carrera 60 red/white 4000 2001-03-14 11200
Bianchi Corse Evo 4 58 celeste 5700 2004-12-22 300
Bianchi Evolution 3 58 celeste 4800 2003-11-16 2000
Bianchi/Liquigas FG 58 celeste/blue 5600 2005-12-02 0
Colnago Dream Rabobank 60 blue/orange 5500 2002-07-27 4300
Colnago Superissimo 59 red 3800 1996-03-01 13000
Eddy Merckx Domo 58 blue/black 5300 2005-02-02 0
Eddy Merckx Molteni 58 orange 5100 2004-08-12 0
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
Schemata Bookmarks History
bikedb
bikes
bluebikes
information_schema
mysql
sample
test
Syntax Functions Params Trx
Data Manipulation
Data Definition
MySQL Utility
Transactional and Locking
Result set shown for this query. Note that this query is based on a single table, so the result set is editable.
11 rows fetched in 0.0050s (0.0006s)
Edit
Apply Changes
Discard Changes
First
Last
Search
1: 1


MySQL Query Browser - root@localhost:3306 / bikedb
File Edit View Query Script Tools Window Help
SELECT * from bikes where cost > 4500
Go back
Next
Refresh
Execute
Stop
Resultset 1
Resultset 2
Resultset 3
bikename size color cost purchased mileage
Bianchi Corse Evo 4 58 celeste 5700 2004-12-22 300
Bianchi Evolution 3 58 celeste 4800 2003-11-16 2000
Bianchi/Liquigas FG 58 celeste/blue 5600 2005-12-02 0
Colnago Dream Rabobank 60 blue/orange 5500 2002-07-27 4300
Eddy Merckx Domo 58 blue/black 5300 2005-02-02 0
Eddy Merckx Molteni 58 orange 5100 2004-08-12 0
Schemata
Bookmarks
History
bikedb
bikes
bluebikes
information_schema
mysql
sample
test
Syntax
Functions
Params
Trx
Data Manipulation
Data Definition
MySQL Utility
Transactional and Locking
You can manage multiple result sets simultaneously. Statistics on query execution are always available.
6 rows fetched in 0.0047s (0.0004s)
Edit
Apply Changes
Discard Changes
First
Last
Search
1: 1