CNT 4714: Enterprise Computing Spring 2013

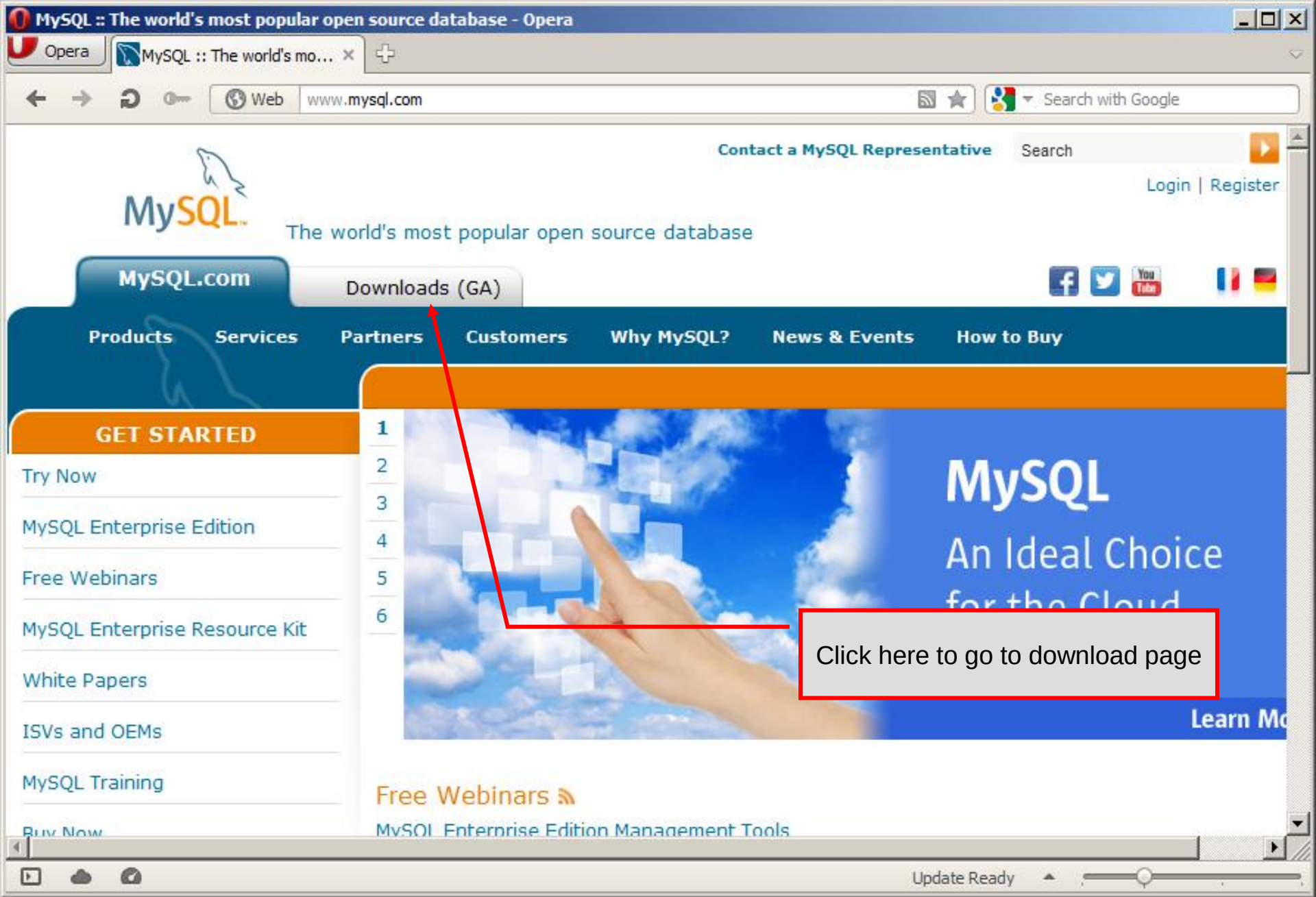
Introduction To MySQL
Installation Of MySQL 5.5.29

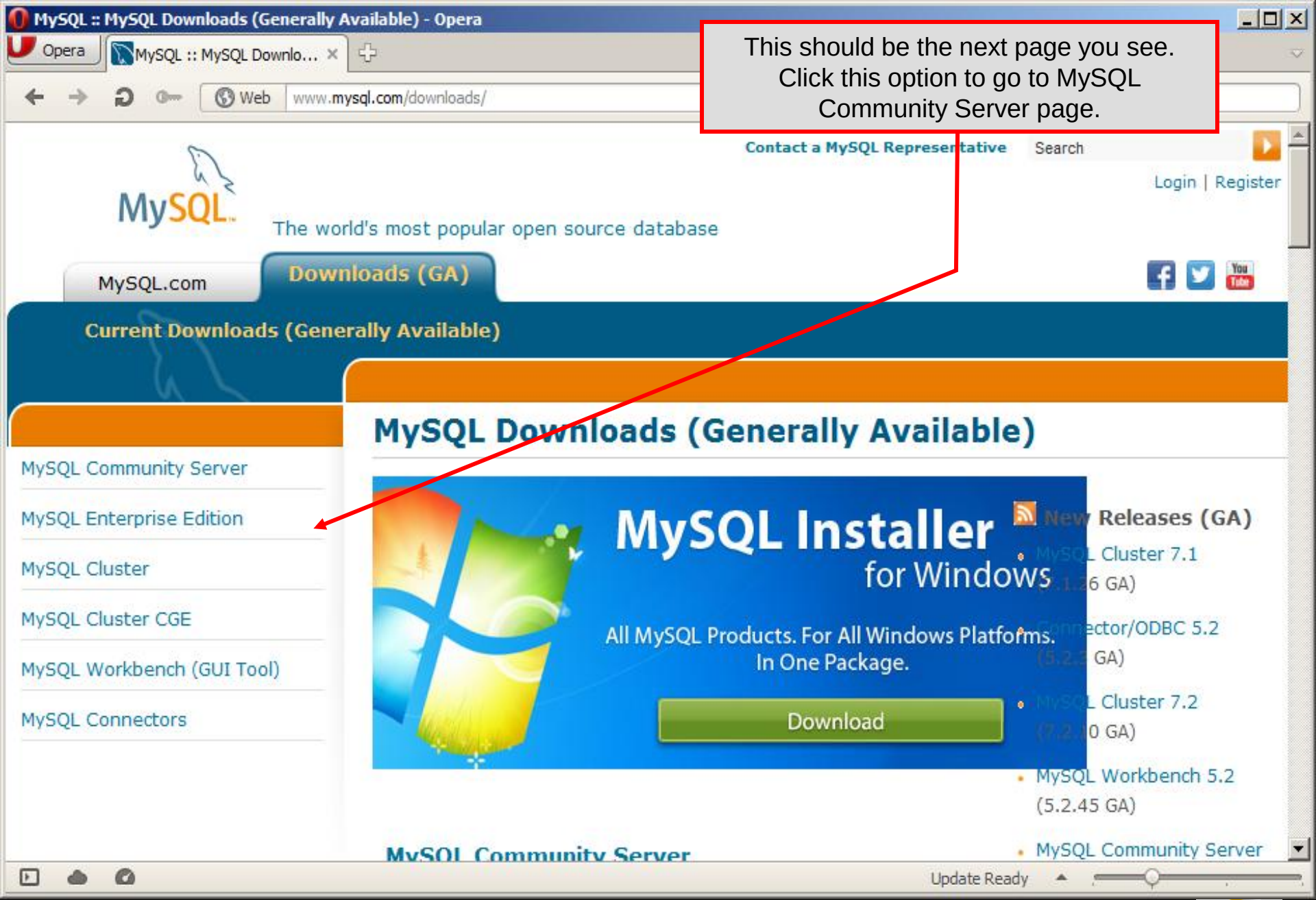
Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cnt4714/spr2013

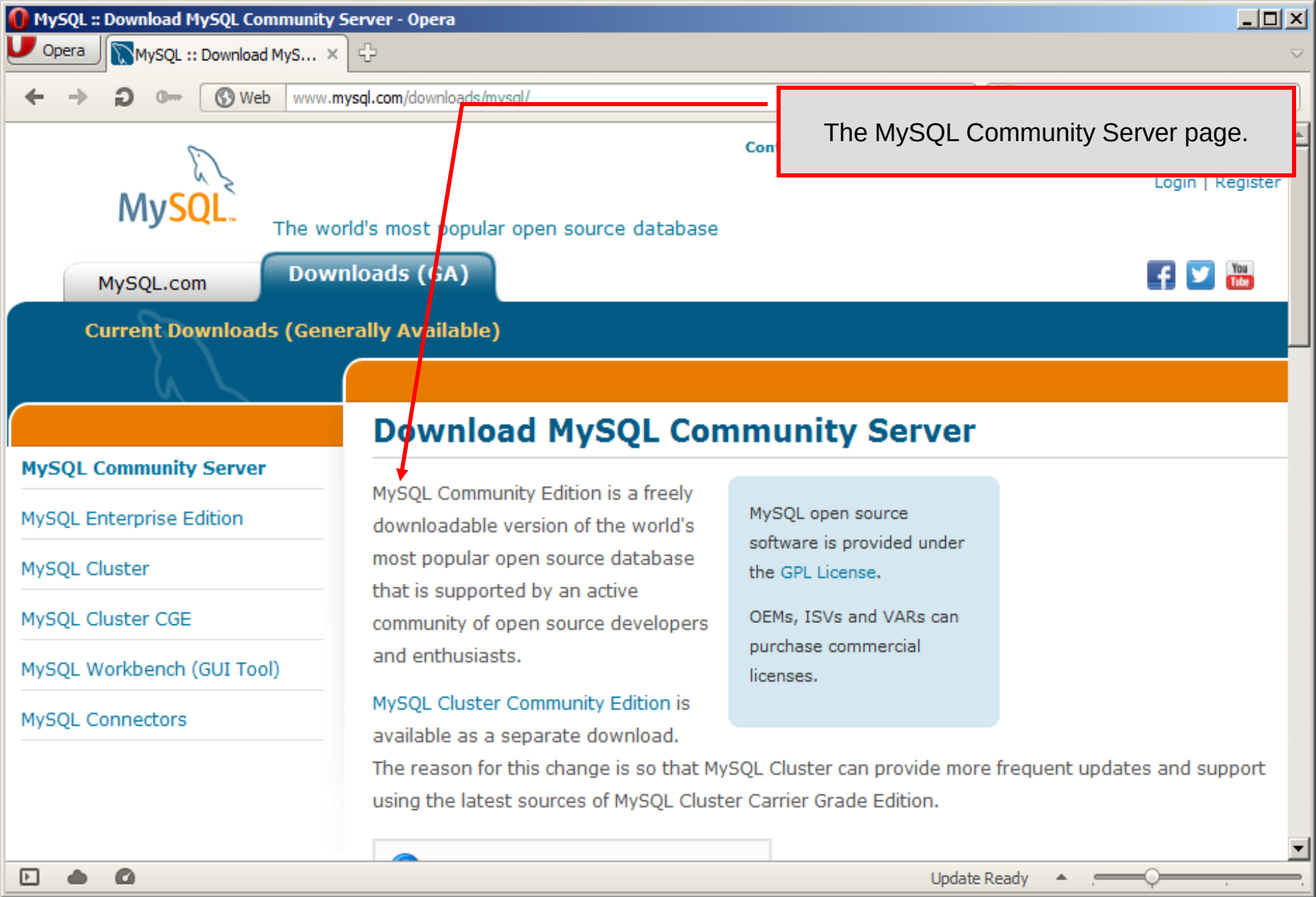
Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida

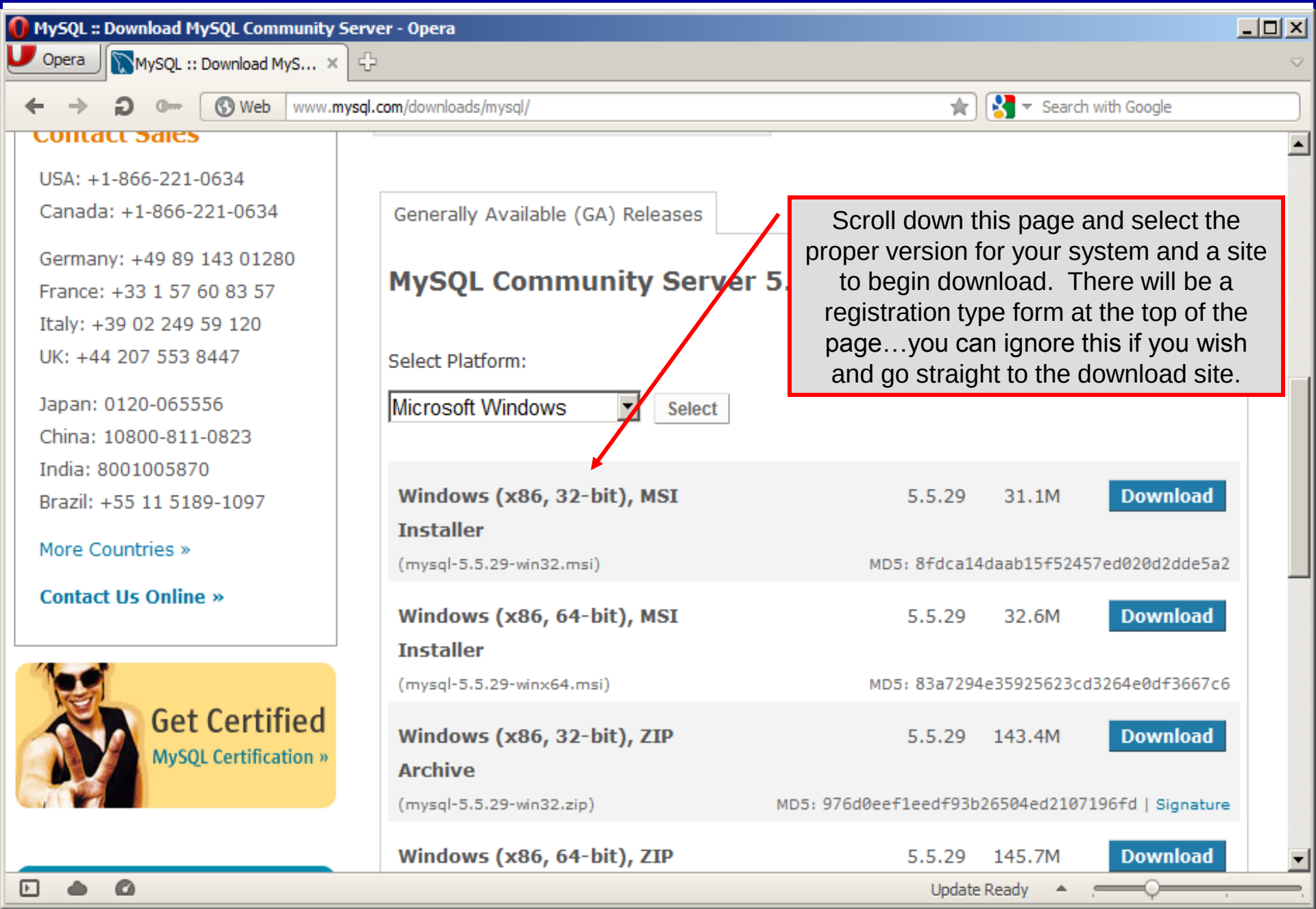
MySQL RDBMS

- MySQL is a database server (although it does come with a set of simple client programs). The current stable version is 5.5.29 and can be downloaded from www.mysql.com.
- 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.

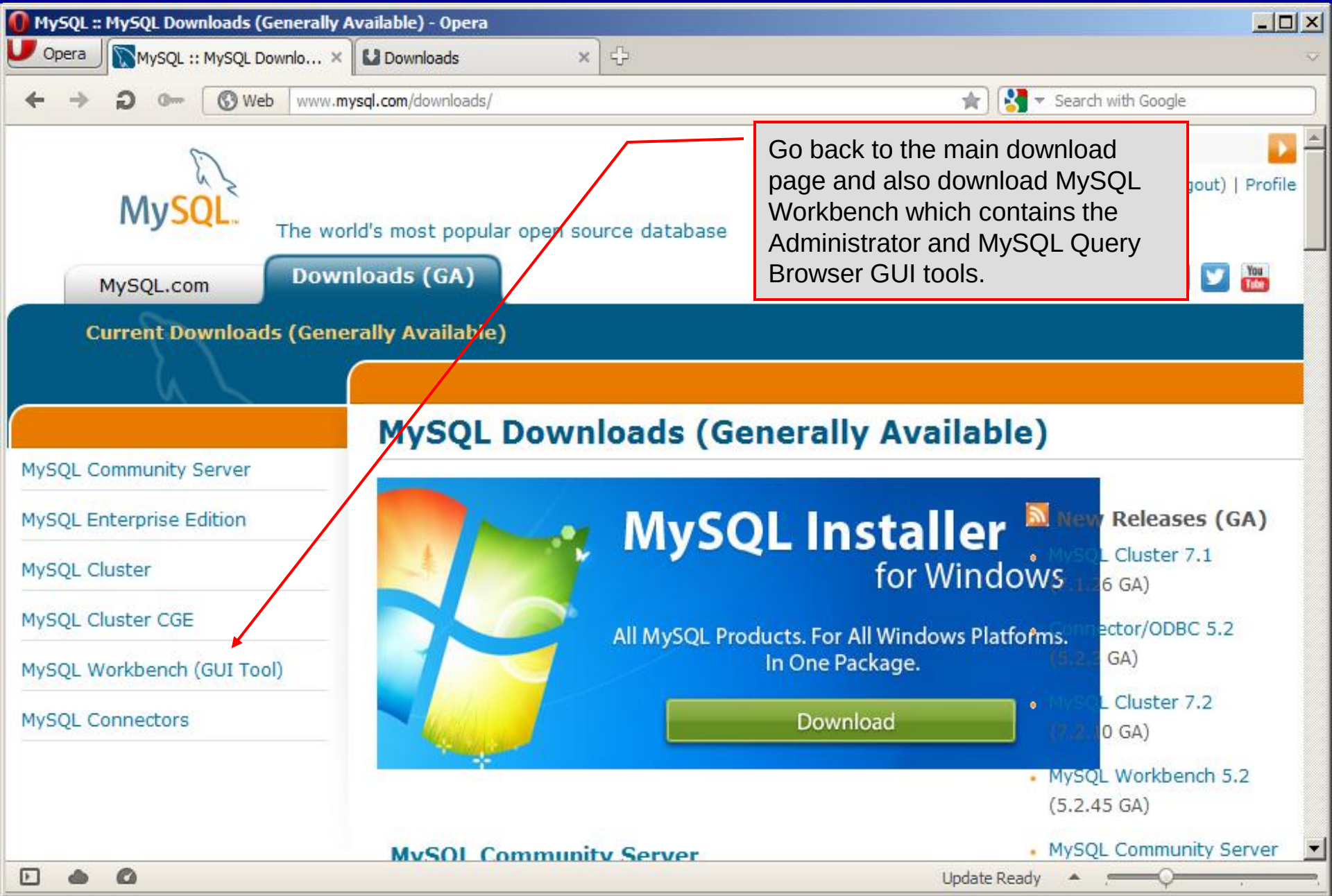

MySQL :: The world's most popular open source database - Opera
Opera
MySQL :: The world's mo...
Web
www.mysql.com
Search with Google
Contact a MySQL Representative
Search
Login | Register
MySQL
The world's most popular open source database
MySQL.com
Downloads (GA)
Products
Services
Partners
Customers
Why MySQL?
News & Events
How to Buy
GET STARTED
Try Now
MySQL Enterprise Edition
Free Webinars
MySQL Enterprise Resource Kit
White Papers
ISVs and OEMs
MySQL Training
Buy Now
1
2
3
4
5
6
MySQL
An Ideal Choice
Learn Mo
Click here to go to download page
Free Webinars
MySQL Enterprise Edition Management Tools
Update Ready

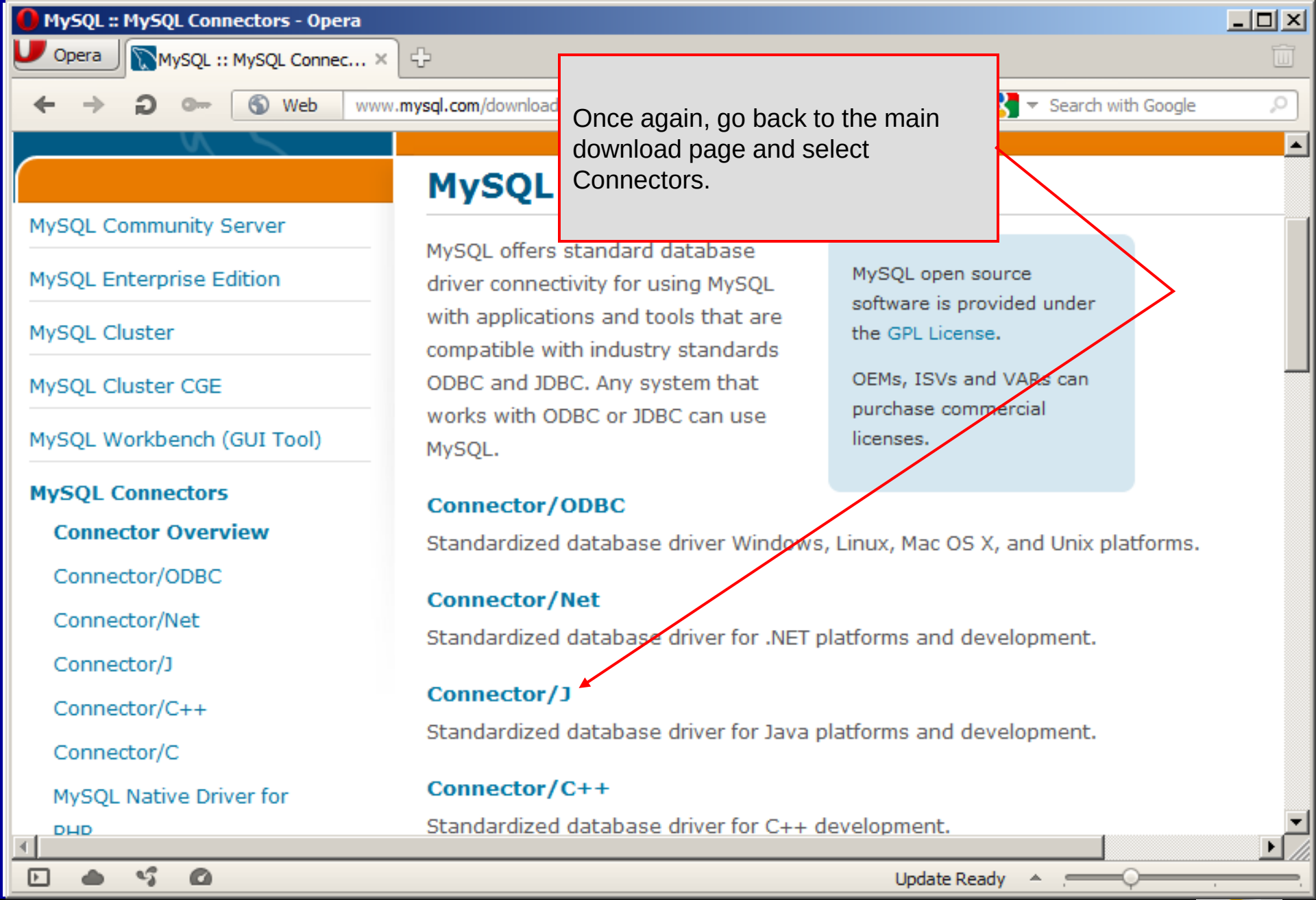

This should be the next page you see. Click this option to go to MySQL Community Server page.
MySQL :: MySQL Downloads (Generally Available) - Opera
Opera
MySQL :: MySQL Downlo...
Web
www.mysql.com/downloads/
Contact a MySQL Representative
Search
Login | Register
MySQL
The world's most popular open source database
MySQL.com
Downloads (GA)
Current Downloads (Generally Available)
MySQL Downloads (Generally Available)
MySQL Community Server
MySQL Enterprise Edition
MySQL Cluster
MySQL Cluster CGE
MySQL Workbench (GUI Tool)
MySQL Connectors
MySQL Installer
for Windows
All MySQL Products. For All Windows Platforms.
In One Package.
Download
New Releases (GA)
MySQL Cluster 7.1
MySQL Workbench 5.2
(5.2.45 GA)
MySQL Community Server
Update Ready

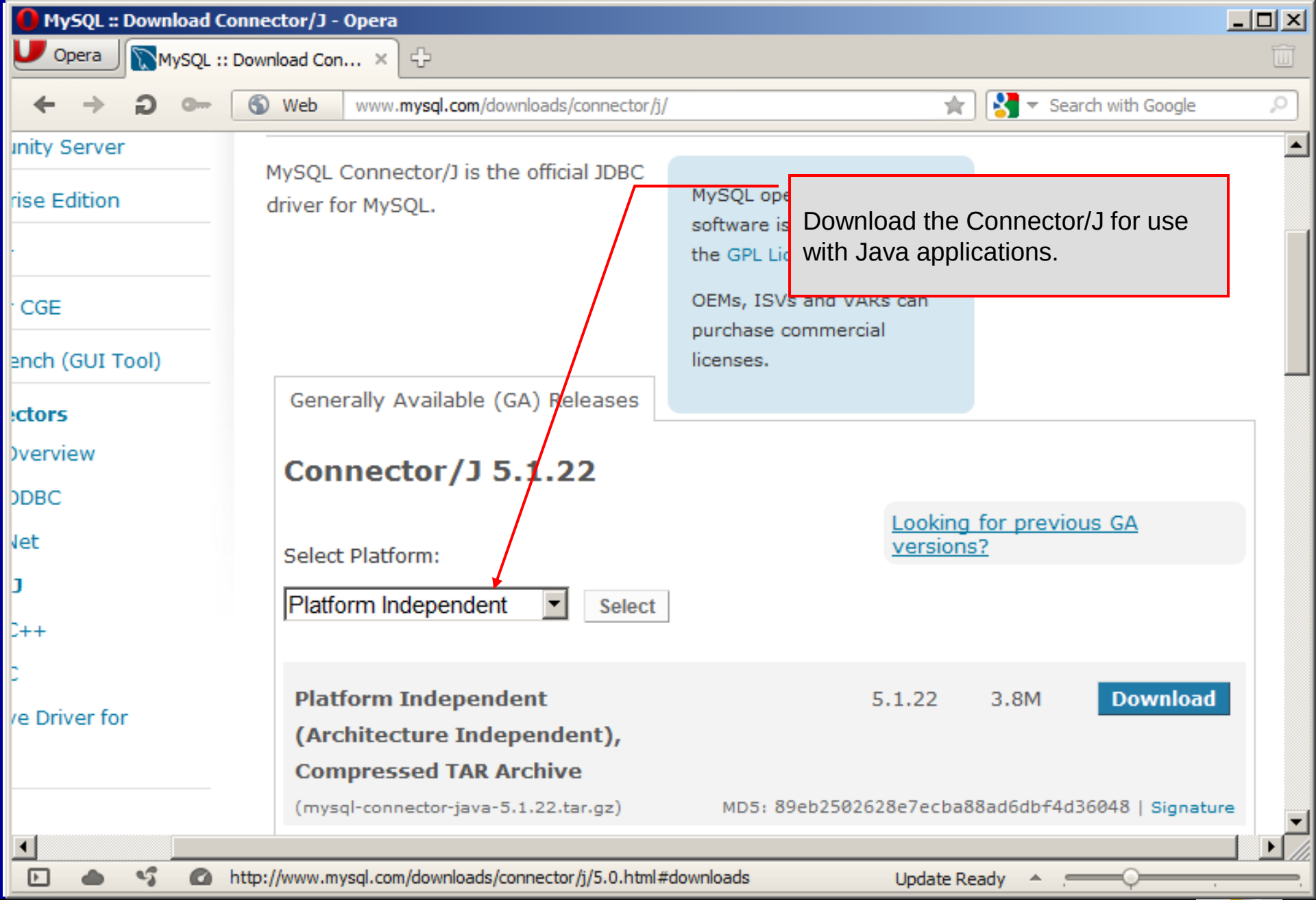

MySQL :: Download MySQL Community Server - Opera
Opera
MySQL :: Download MyS...
Web
www.mysql.com/downloads/mysql/
The MySQL Community Server page.
Login | Register
MySQL
The world's most popular open source database
MySQL.com
Downloads (GA)
Current Downloads (Generally Available)
Download MySQL Community Server
MySQL Community Server
MySQL Enterprise Edition
MySQL Cluster
MySQL Cluster CGE
MySQL Workbench (GUI Tool)
MySQL Connectors
MySQL Community Edition is a freely downloadable version of the world's most popular open source database that is supported by an active community of open source developers and enthusiasts.
MySQL open source software is provided under the GPL License.
OEMs, ISVs and VARs can purchase commercial licenses.
MySQL Cluster Community Edition is available as a separate download.
The reason for this change is so that MySQL Cluster can provide more frequent updates and support using the latest sources of MySQL Cluster Carrier Grade Edition.
Update Ready



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.


MySQL :: MySQL Downloads (Generally Available) - Opera
Opera
MySQL :: MySQL Downlo...
Downloads
Web
www.mysql.com/downloads/
Search with Google
MySQL
The world's most popular open source database
MySQL.com
Downloads (GA)
Current Downloads (Generally Available)
MySQL Downloads (Generally Available)
MySQL Community Server
MySQL Enterprise Edition
MySQL Cluster
MySQL Cluster CGE
MySQL Workbench (GUI Tool)
MySQL Connectors
MySQL Installer
for Windows
All MySQL Products. For All Windows Platforms.
In One Package.
Download
New Releases (GA)
MySQL Cluster 7.1
Connector/ODBC 5.2
MySQL Cluster 7.2
MySQL Workbench 5.2
(5.2.45 GA)
MySQL Community Server
MySQL Community Server
Update Ready
Go back to the main download page and also download MySQL Workbench which contains the Administrator and MySQL Query Browser GUI tools.


MySQL :: MySQL Connectors - Opera
Opera
MySQL :: MySQL Connec...
Web
www.mysql.com/download
Search with Google
Once again, go back to the main download page and select Connectors.
MySQL Community Server
MySQL Enterprise Edition
MySQL Cluster
MySQL Cluster CGE
MySQL Workbench (GUI Tool)
MySQL Connectors
Connector Overview
Connector/ODBC
Connector/Net
Connector/J
Connector/C++
Connector/C
MySQL Native Driver for PHP
MySQL
MySQL offers standard database driver connectivity for using MySQL with applications and tools that are compatible with industry standards ODBC and JDBC. Any system that works with ODBC or JDBC can use MySQL.
MySQL open source software is provided under the GPL License.
OEMs, ISVs and VARs can purchase commercial licenses.
Connector/ODBC
Standardized database driver Windows, Linux, Mac OS X, and Unix platforms.
Connector/Net
Standardized database driver for .NET platforms and development.
Connector/J
Standardized database driver for Java platforms and development.
Connector/C++
Standardized database driver for C++ development.
Update Ready


MySQL :: Download Connector/J - Opera
Opera
MySQL :: Download Con...
Web
www.mysql.com/downloads/connector/j/
Search with Google
unity Server
rise Edition
CGE
ench (GUI Tool)
ectors
Overview
DBC
Net
J
C++
C
ve Driver for
MySQL Connector/J is the official JDBC driver for MySQL.
MySQL ope
software is
the GPL Lic
OEMs, ISVs and VARs can purchase commercial licenses.
Download the Connector/J for use with Java applications.
Generally Available (GA) Releases
Connector/J 5.1.22
Looking for previous GA versions?
Select Platform:
Platform Independent
Select
Platform Independent (Architecture Independent), Compressed TAR Archive
5.1.22
3.8M
Download
(mysql-connector-java-5.1.22.tar.gz)
MD5: 89eb2502628e7ecba88ad6dbf4d36048 | Signature
http://www.mysql.com/downloads/connector/j/5.0.html#downloads
Update Ready

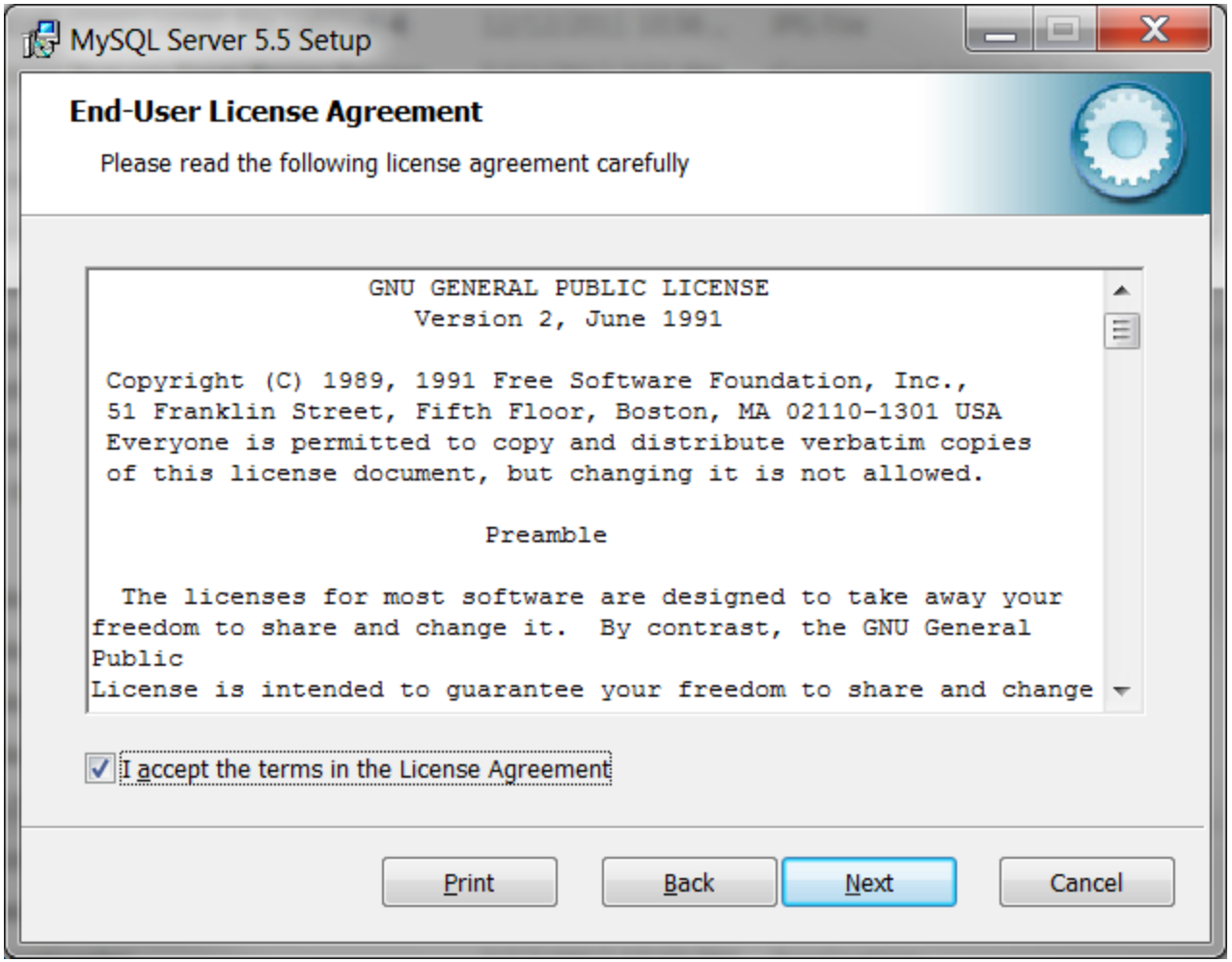
Installing MySQL 5.5.29

- 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 7 over the next few pages, just to give you an idea of what you should be seeing.

Installing MySQL 5.5.29

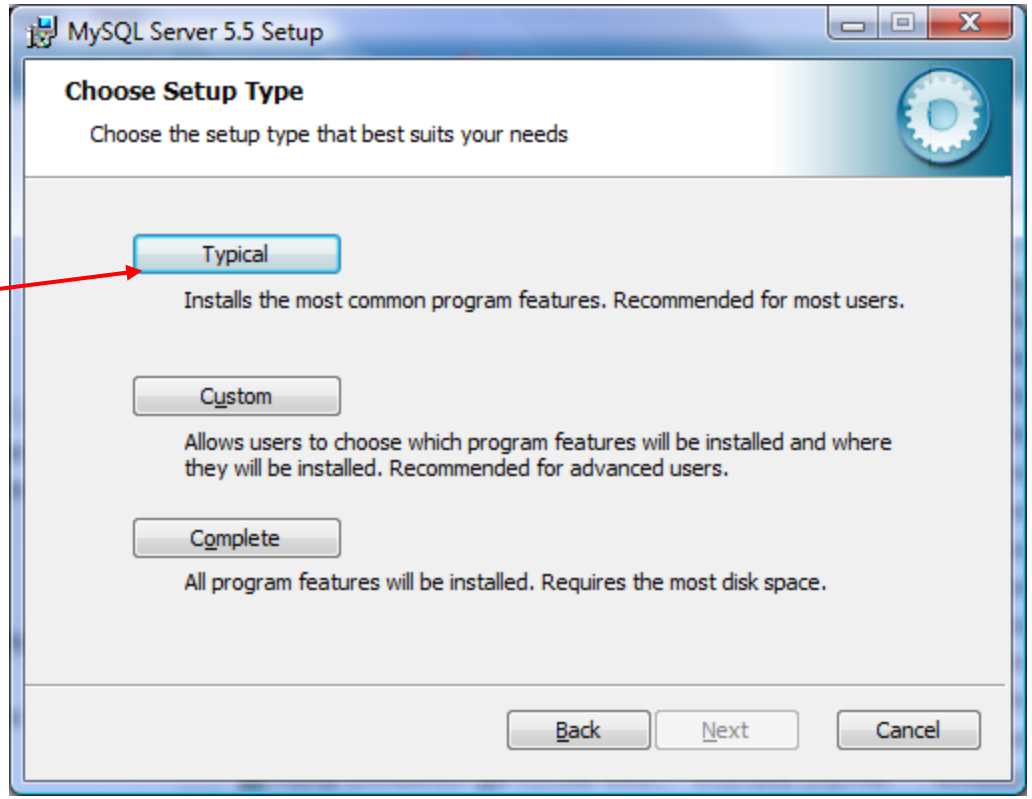
- Once the Window installer is running you should see the following window appear:
- Click Next and accept the terms on the next window.

Installing MySQL 5.5.29 (cont.)



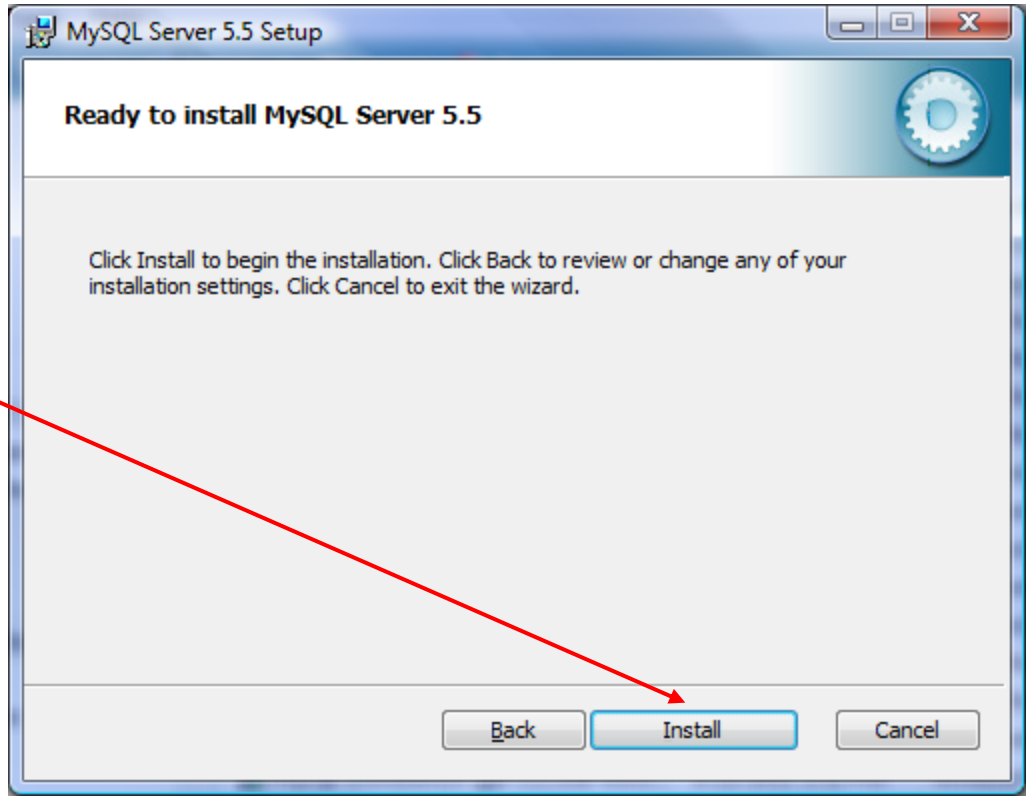
Installing MySQL 5.5.29 (cont.)

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

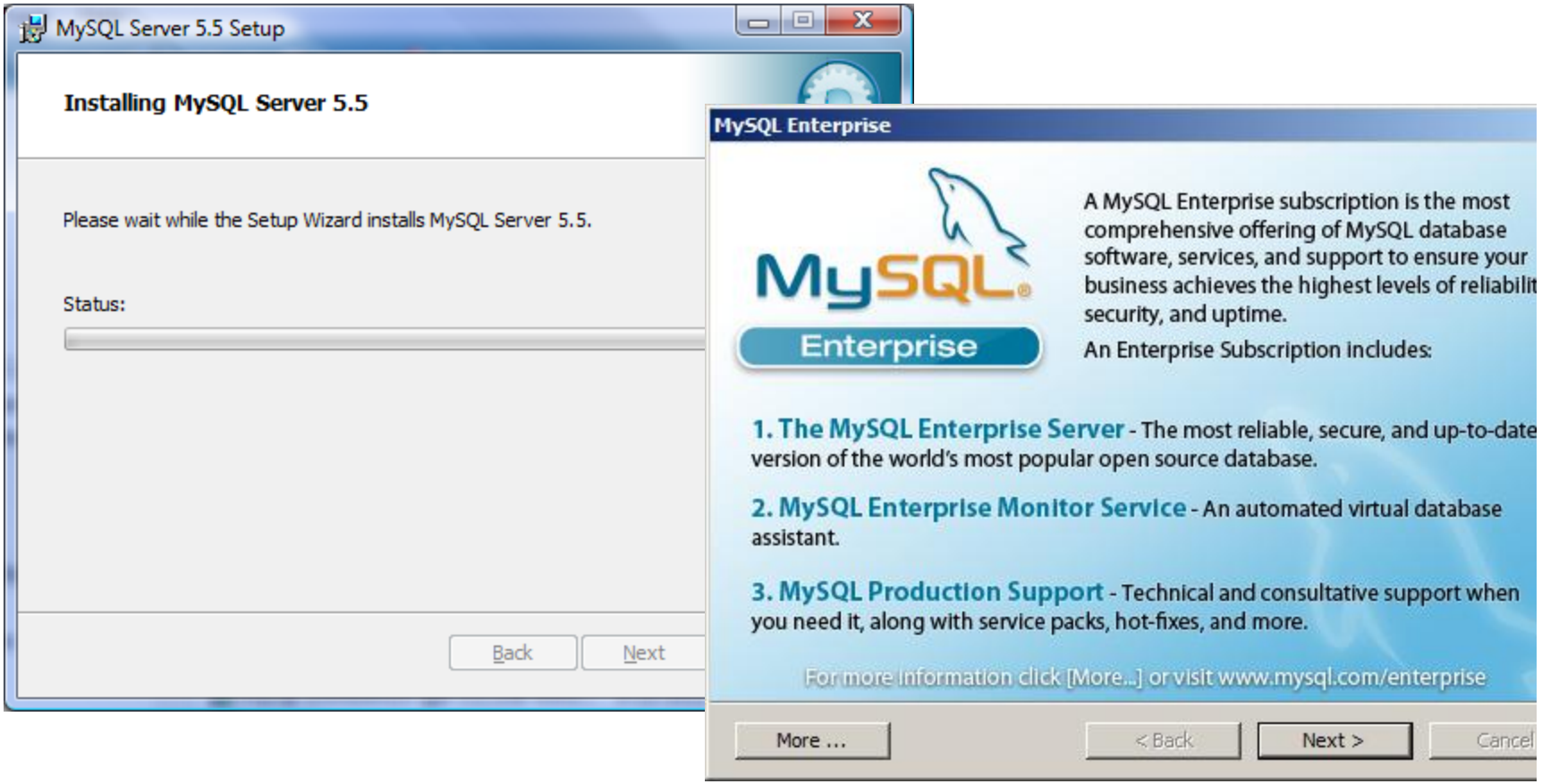


Installing MySQL 5.5.29 (cont.)

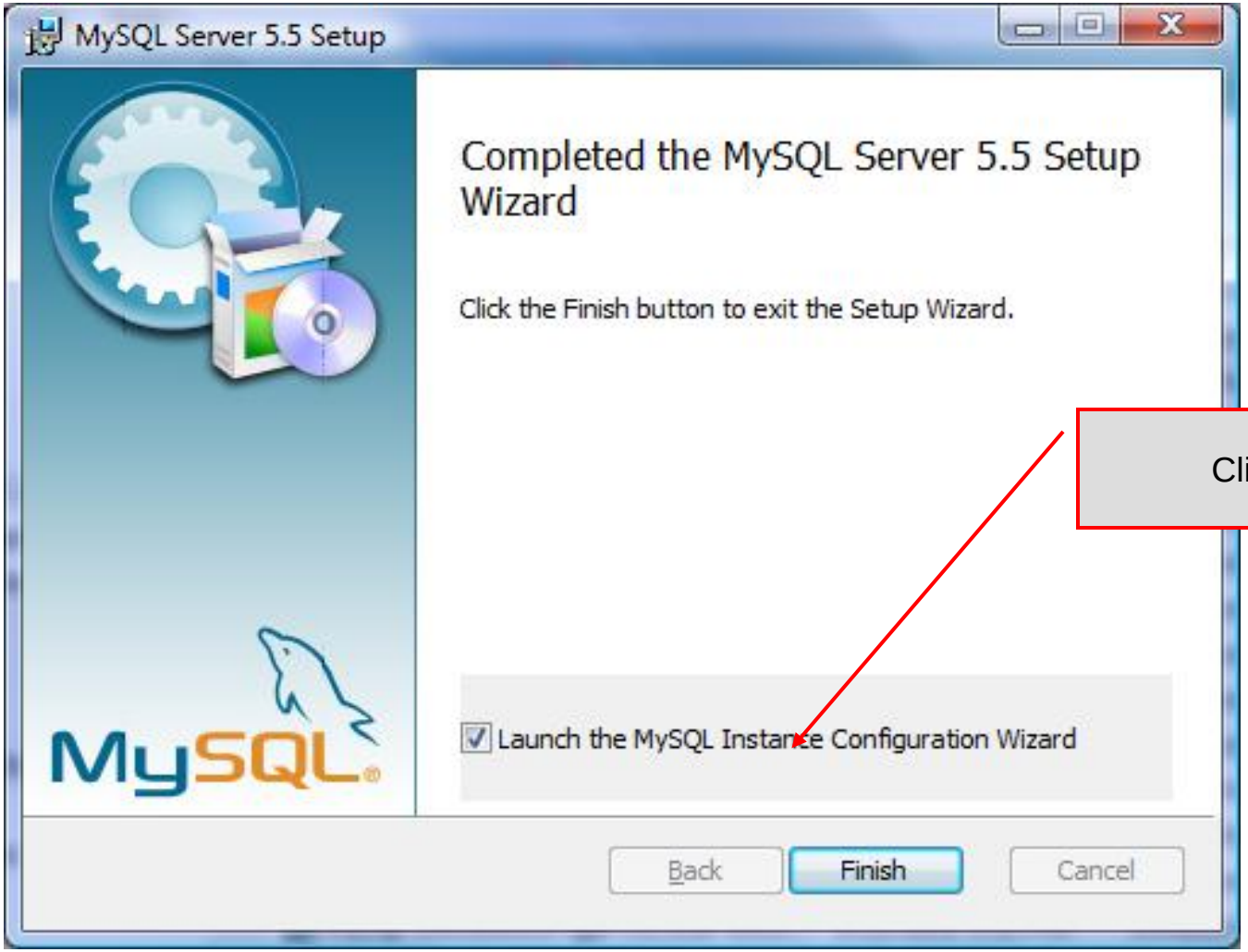
Click Install



Installing MySQL 5.5.29 (cont.)



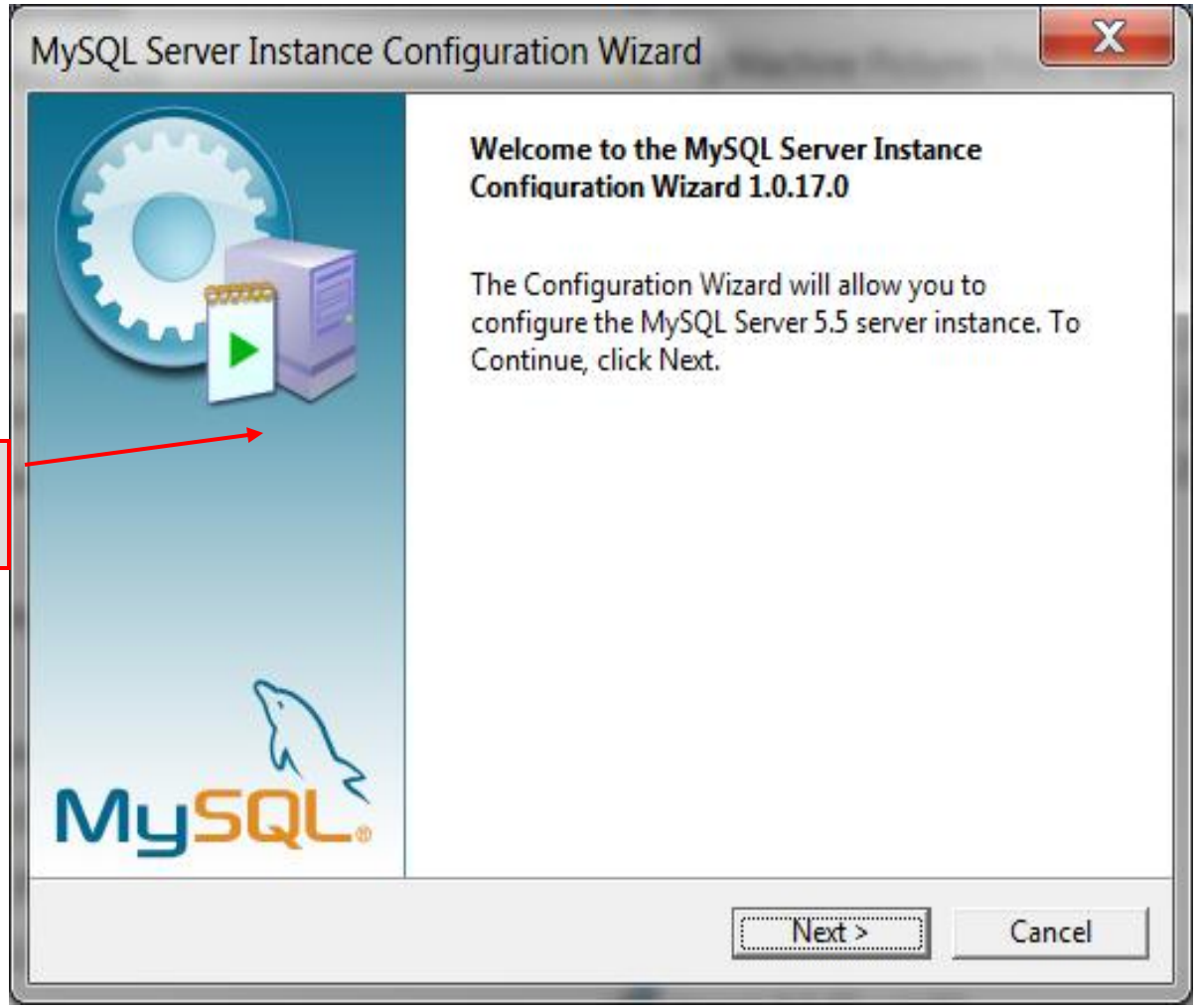
Installing MySQL 5.5.29 (cont.)



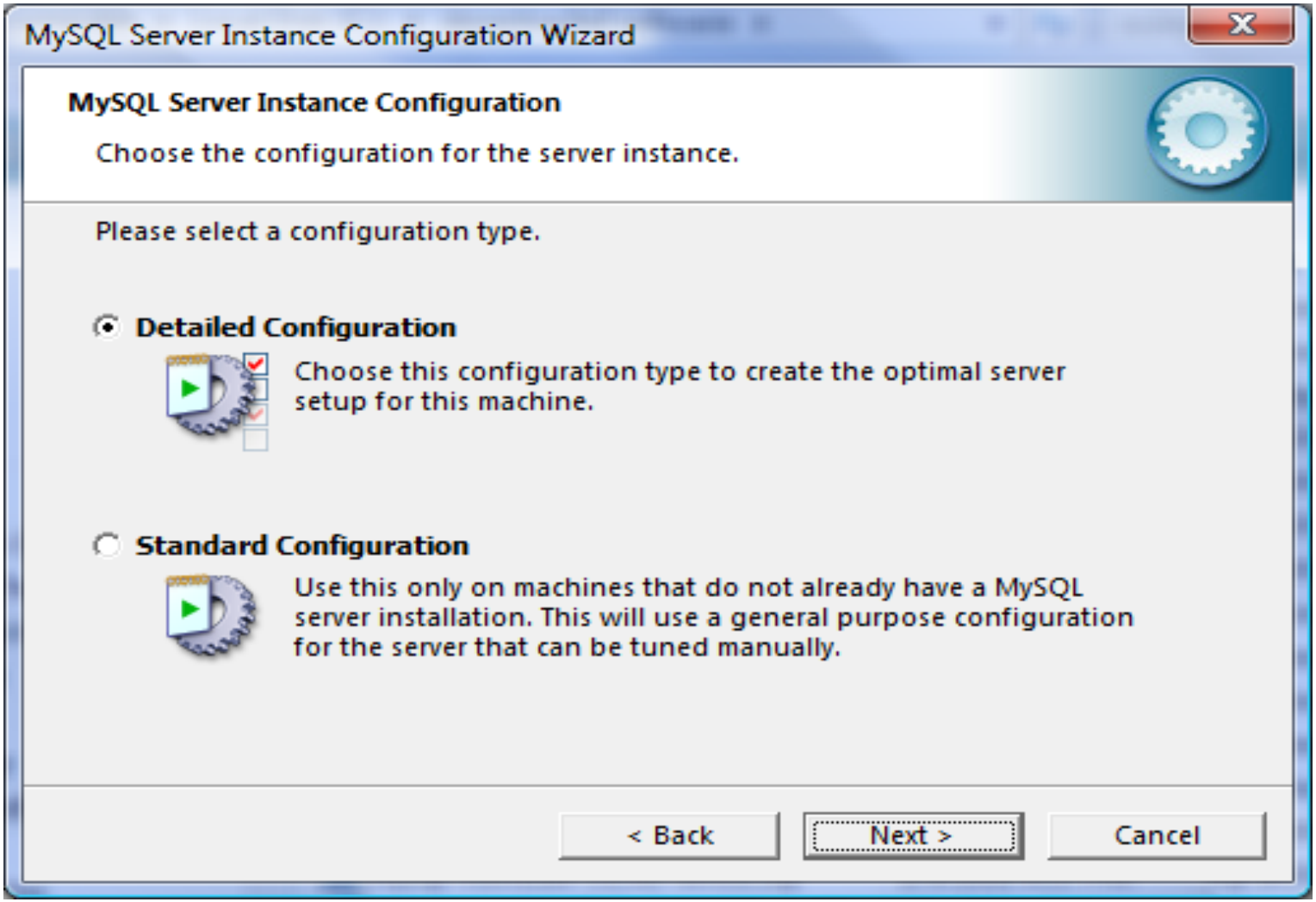
Click Finish

Installing MySQL 5.5.29 (cont.)

Initial server configuration window



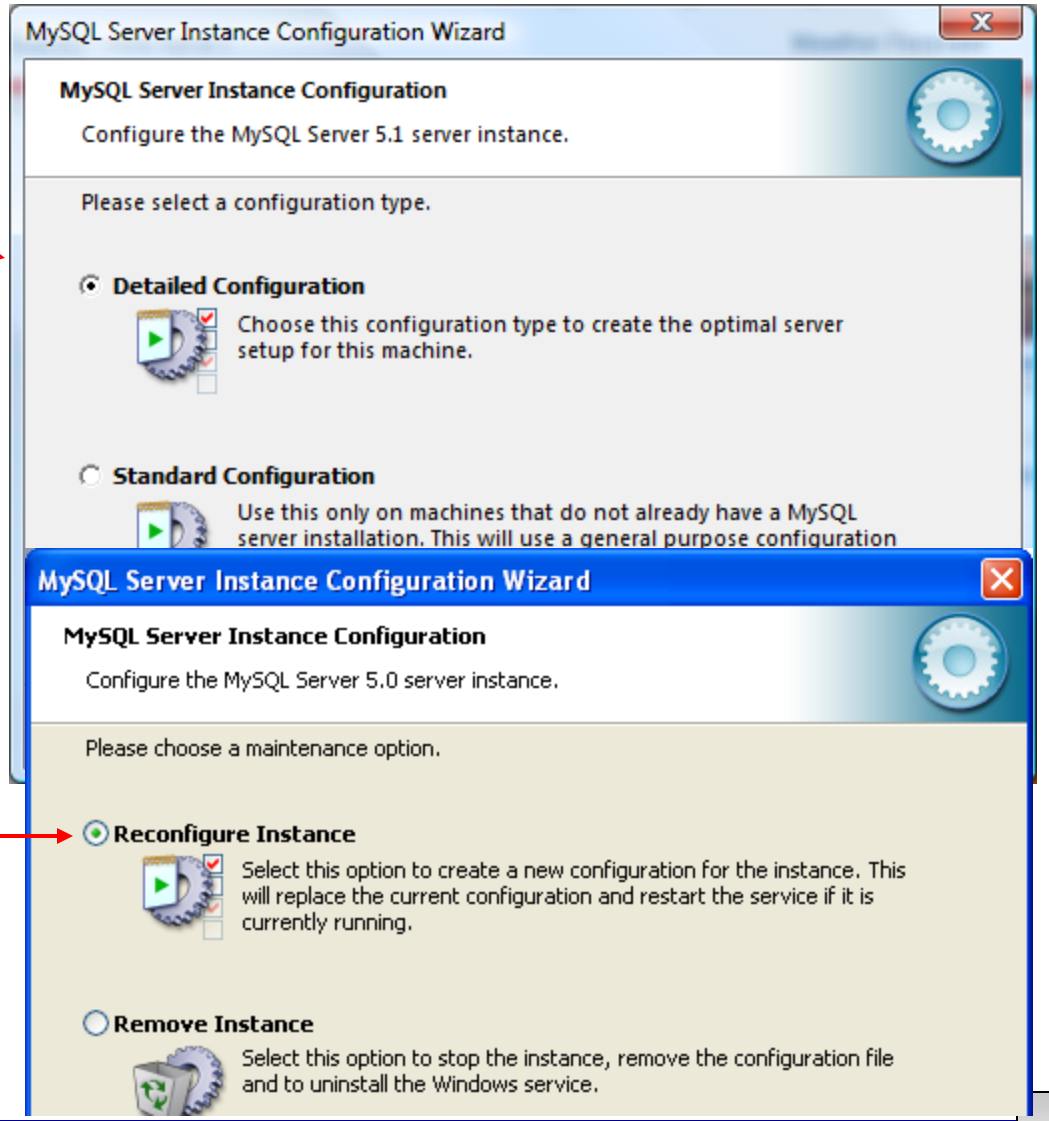
Installing MySQL 5.5.29 (cont.)



Installing MySQL 5.5.29 (cont.)

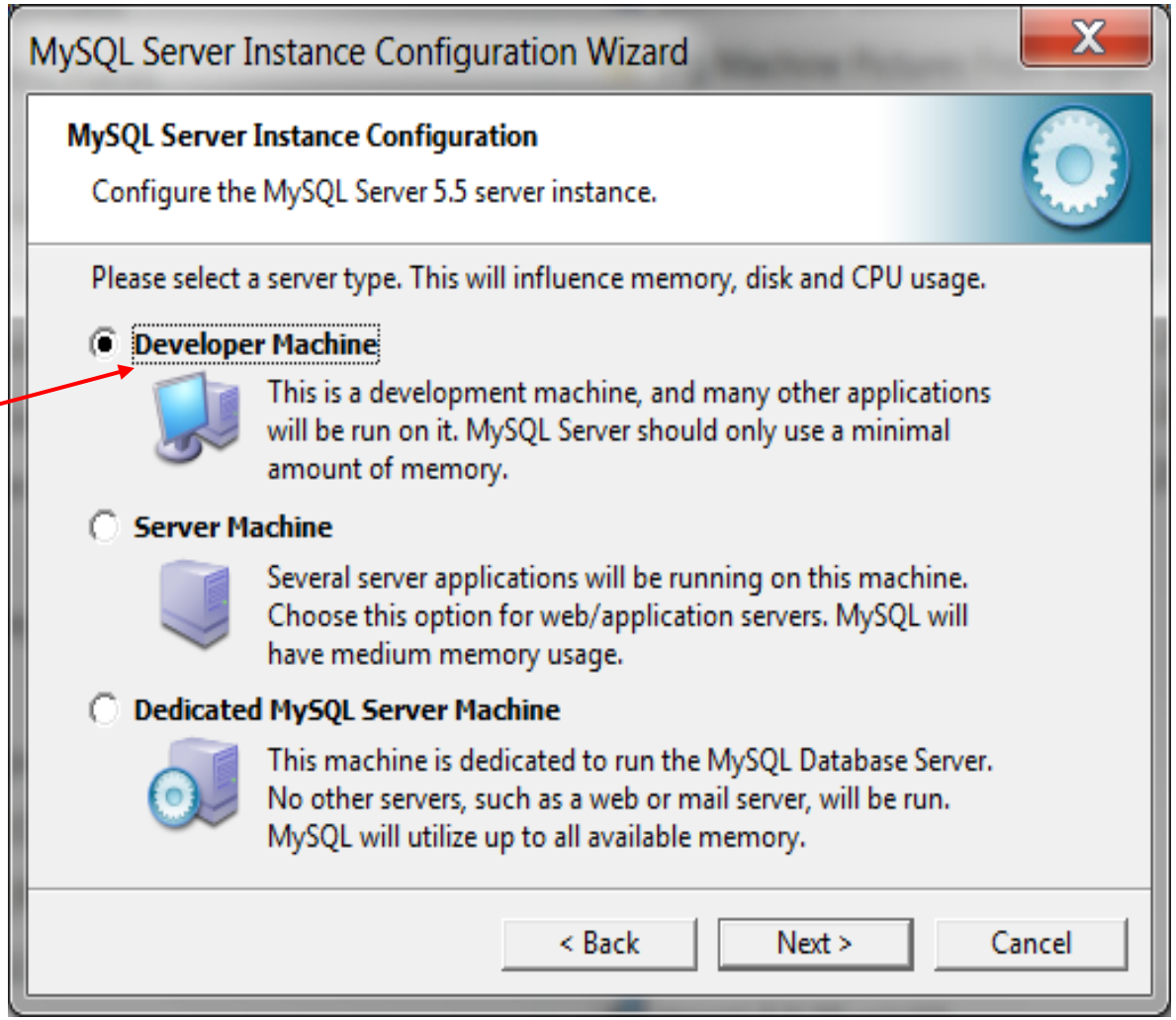
Your choice here. If you are not sure if there is already a MySQL server on your machine, choose the detailed configuration setting.

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.



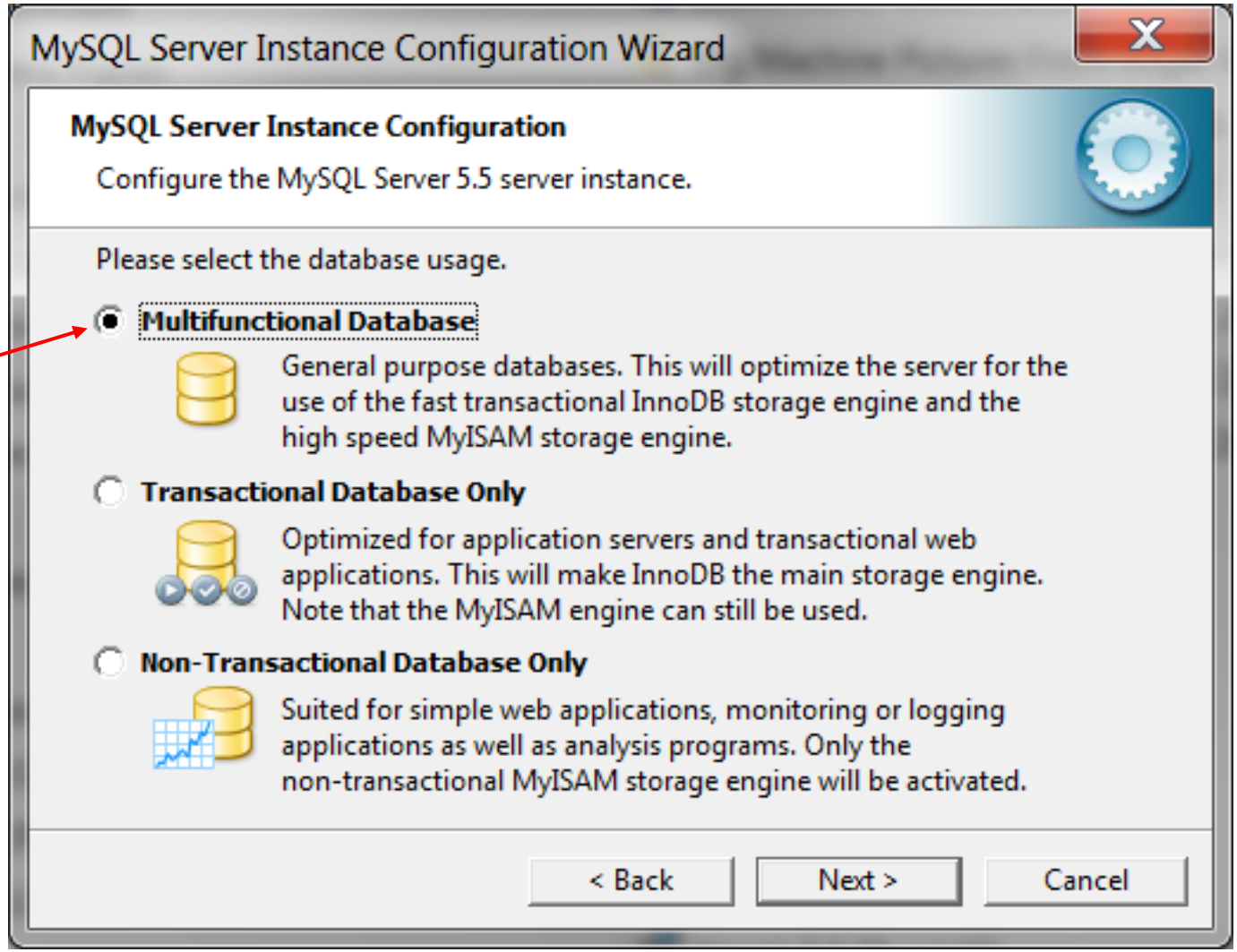
Installing MySQL 5.5.29 (cont.)

Choose the developer machine option



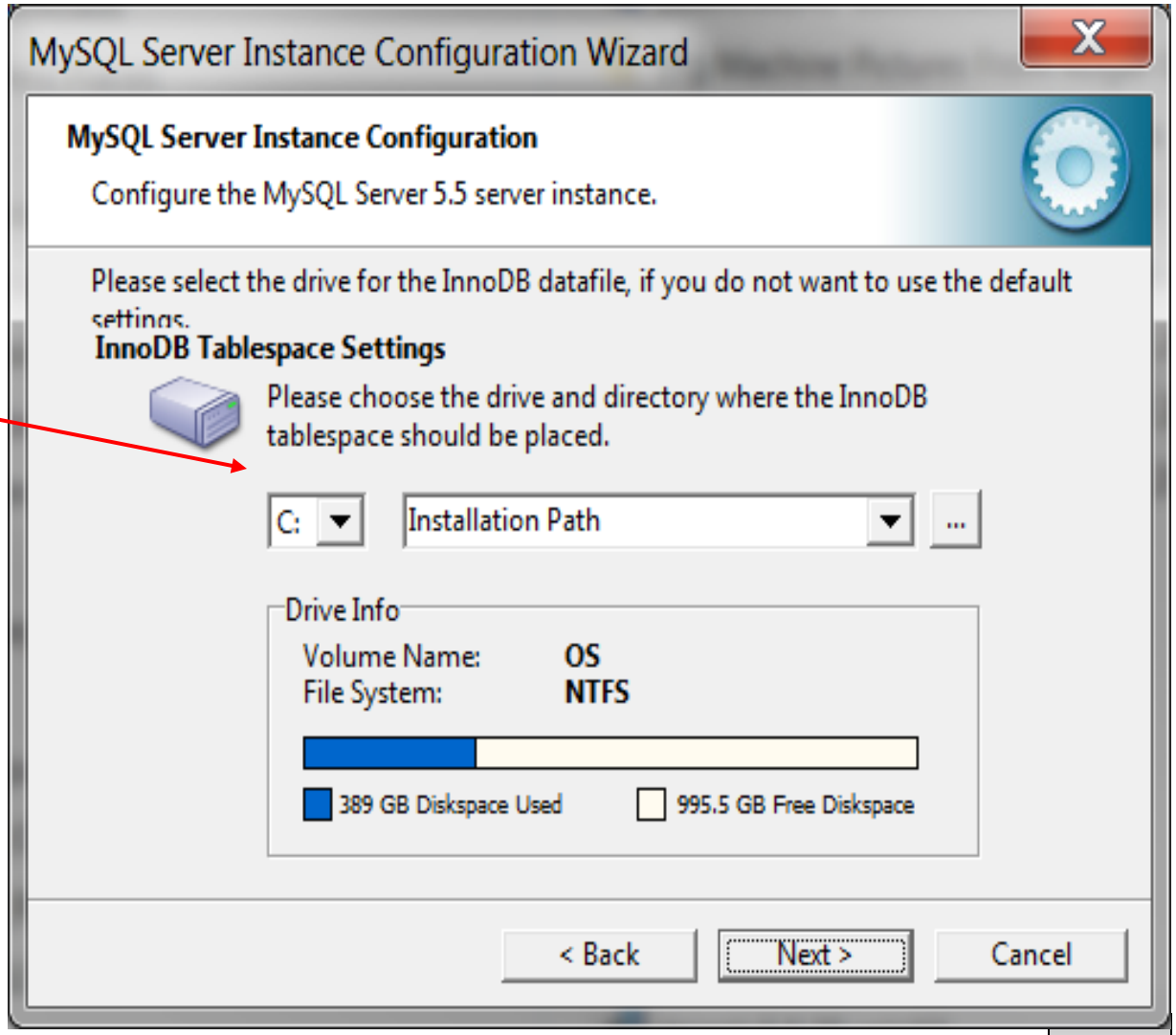
Installing MySQL 5.5.29 (cont.)

Choose the multifunctional database option



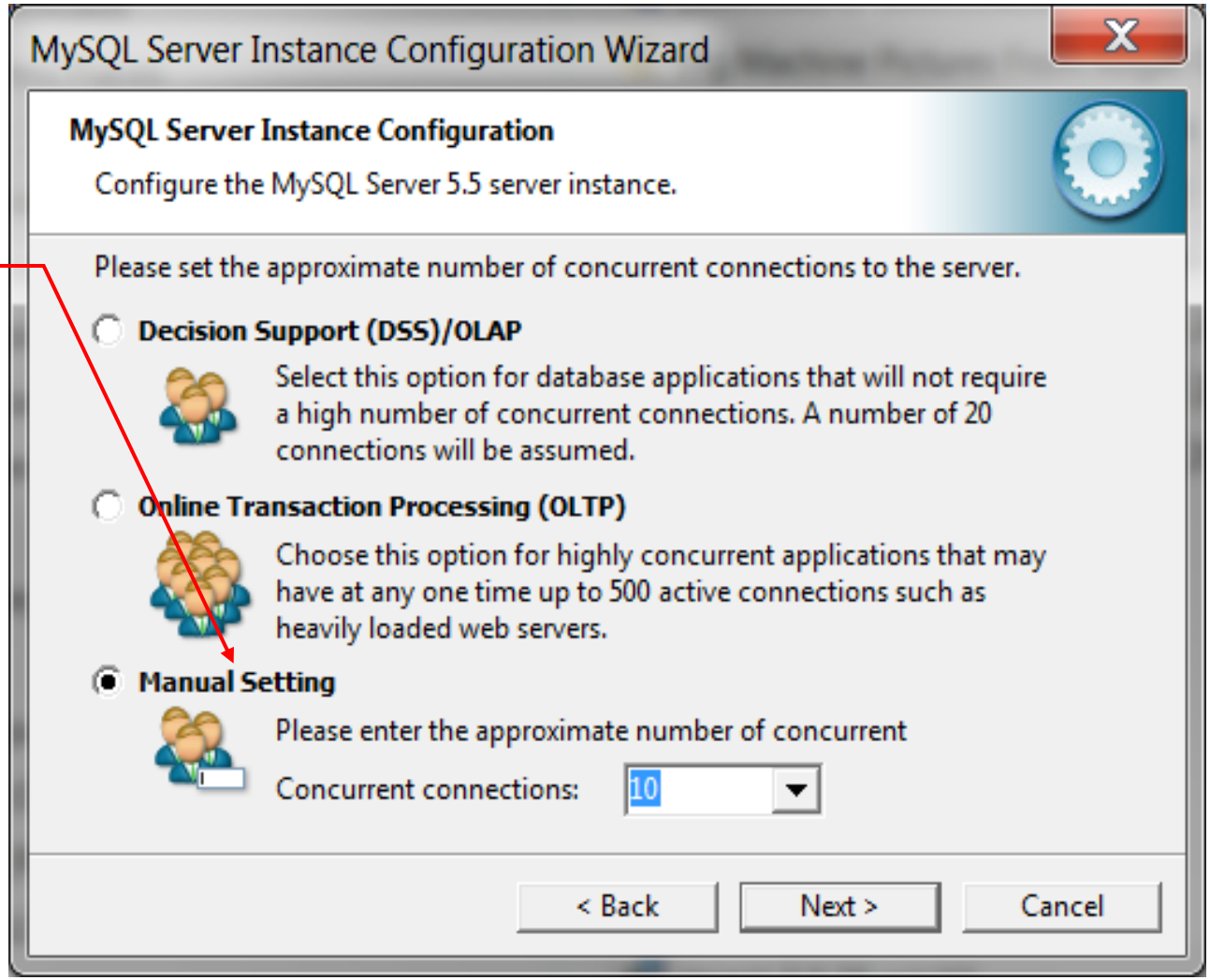
Installing MySQL 5.5.29 (cont.)

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



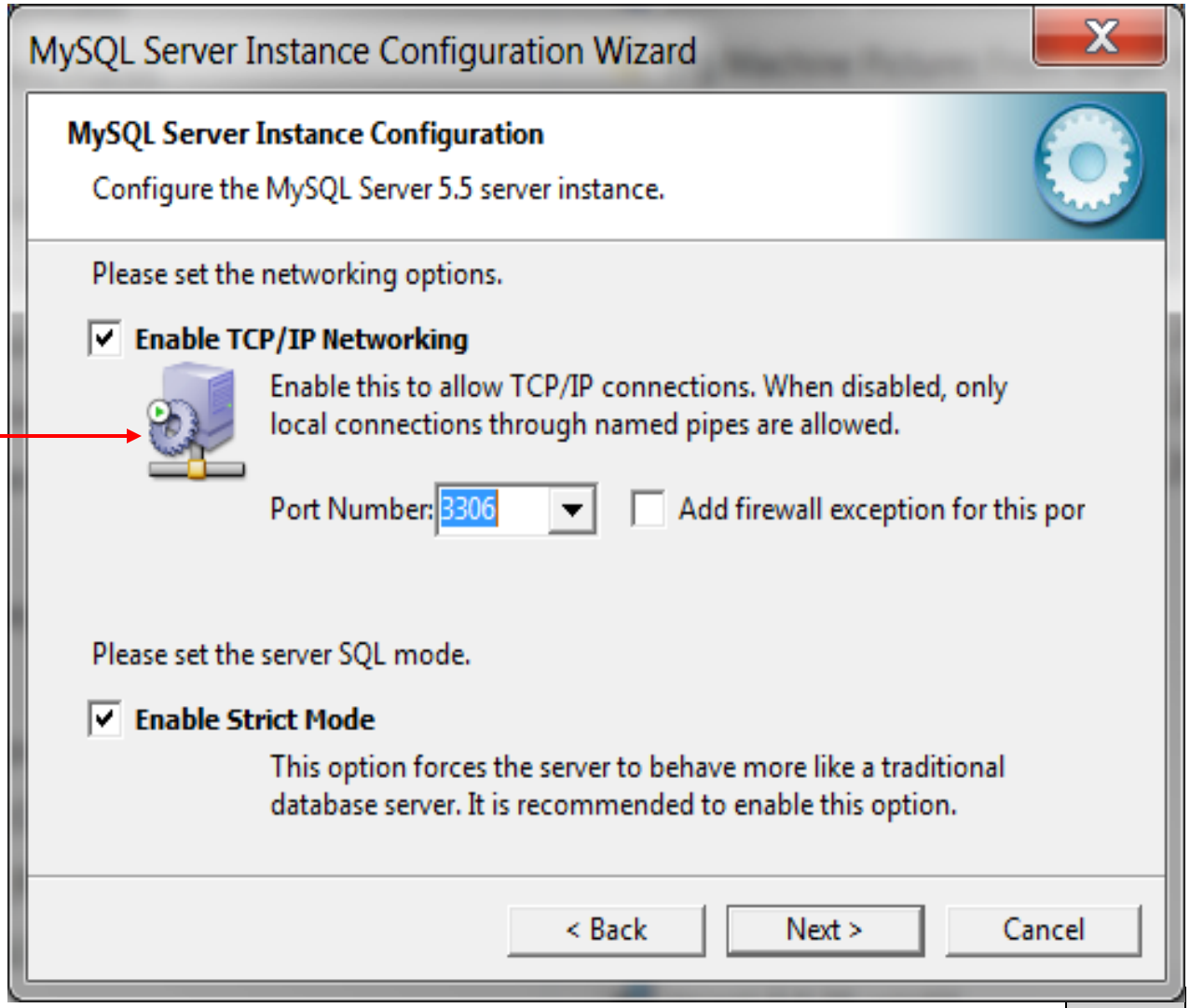
Installing MySQL 5.5.29 (cont.)

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



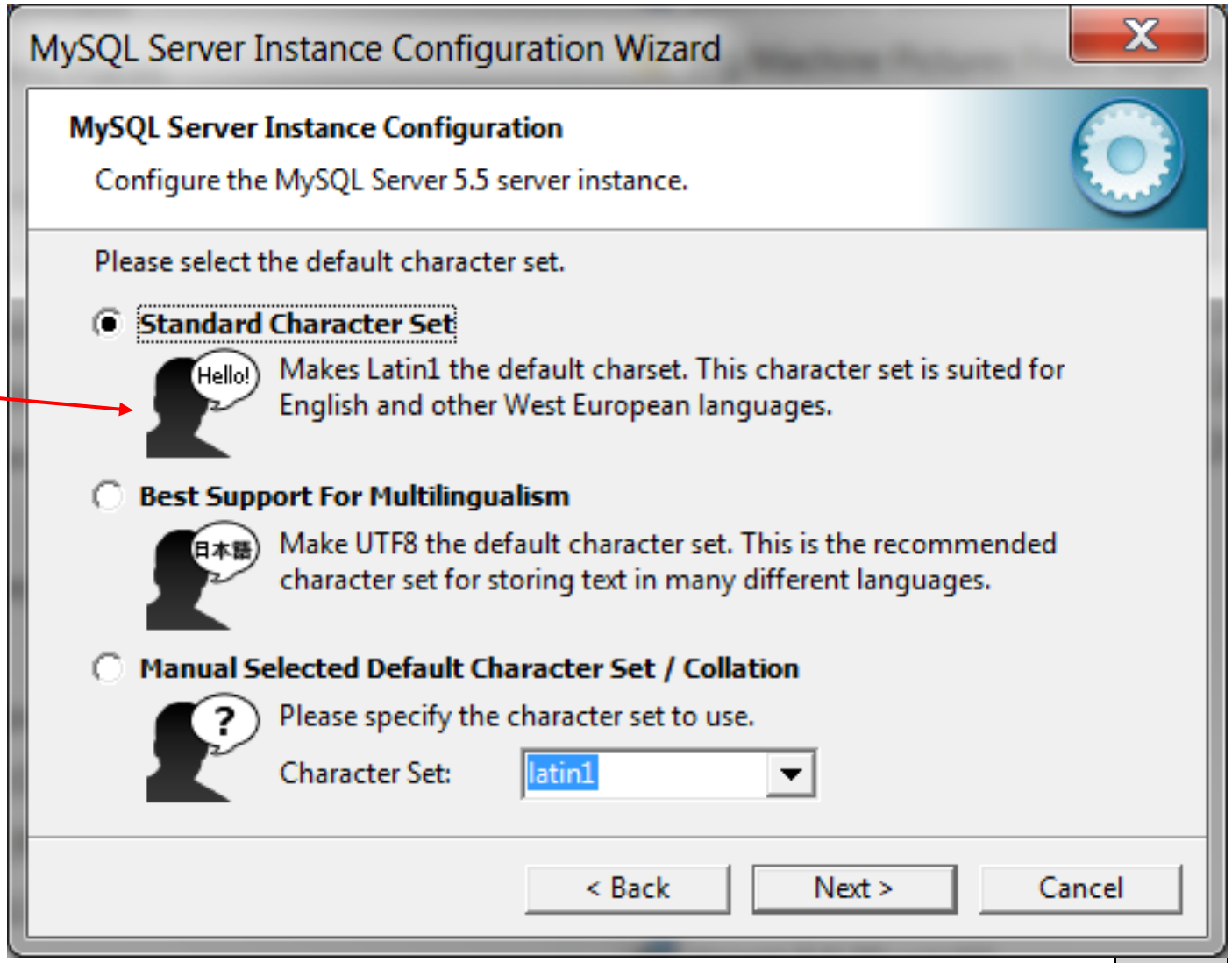
Installing MySQL 5.5.29 (cont.)

Accept all defaults in this window



Installing MySQL 5.5.29 (cont.)

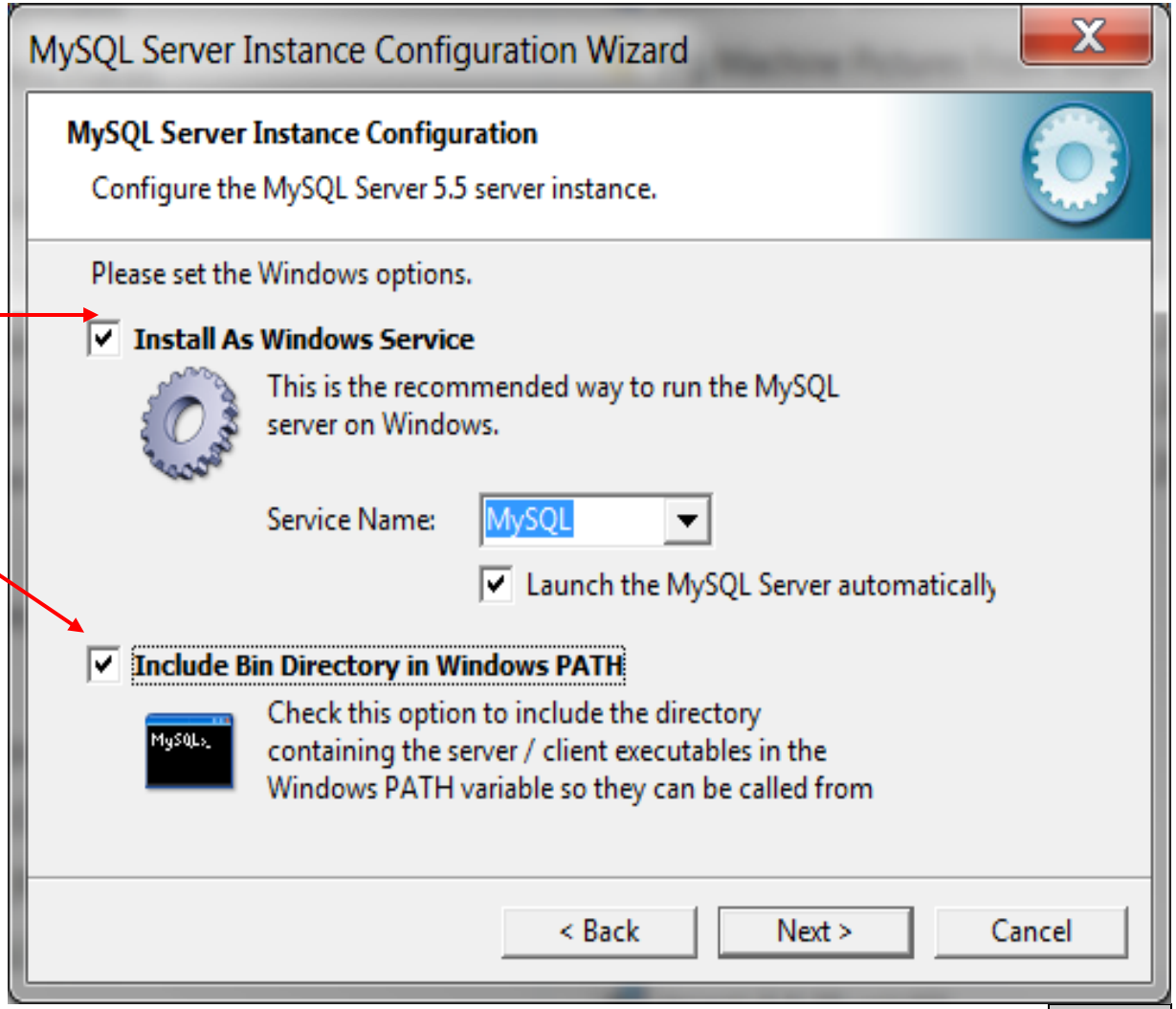
Your choice again



Installing MySQL 5.5.29 (cont.)

Accept default options

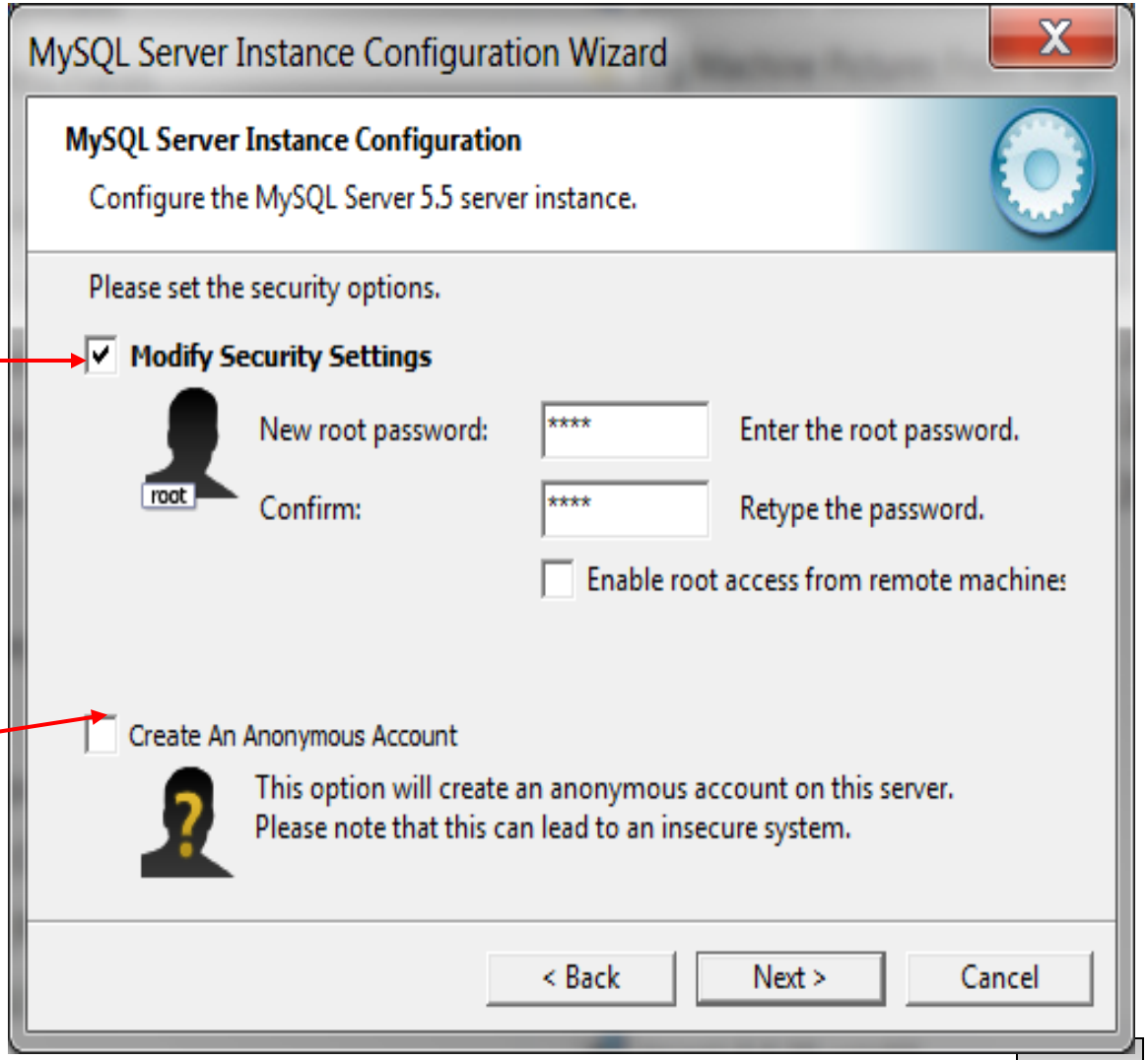
This option is not marked by default, but you can mark and accept it if you want to include MySQL file locations in your PATH statement.



Installing MySQL 5.5.29 (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.

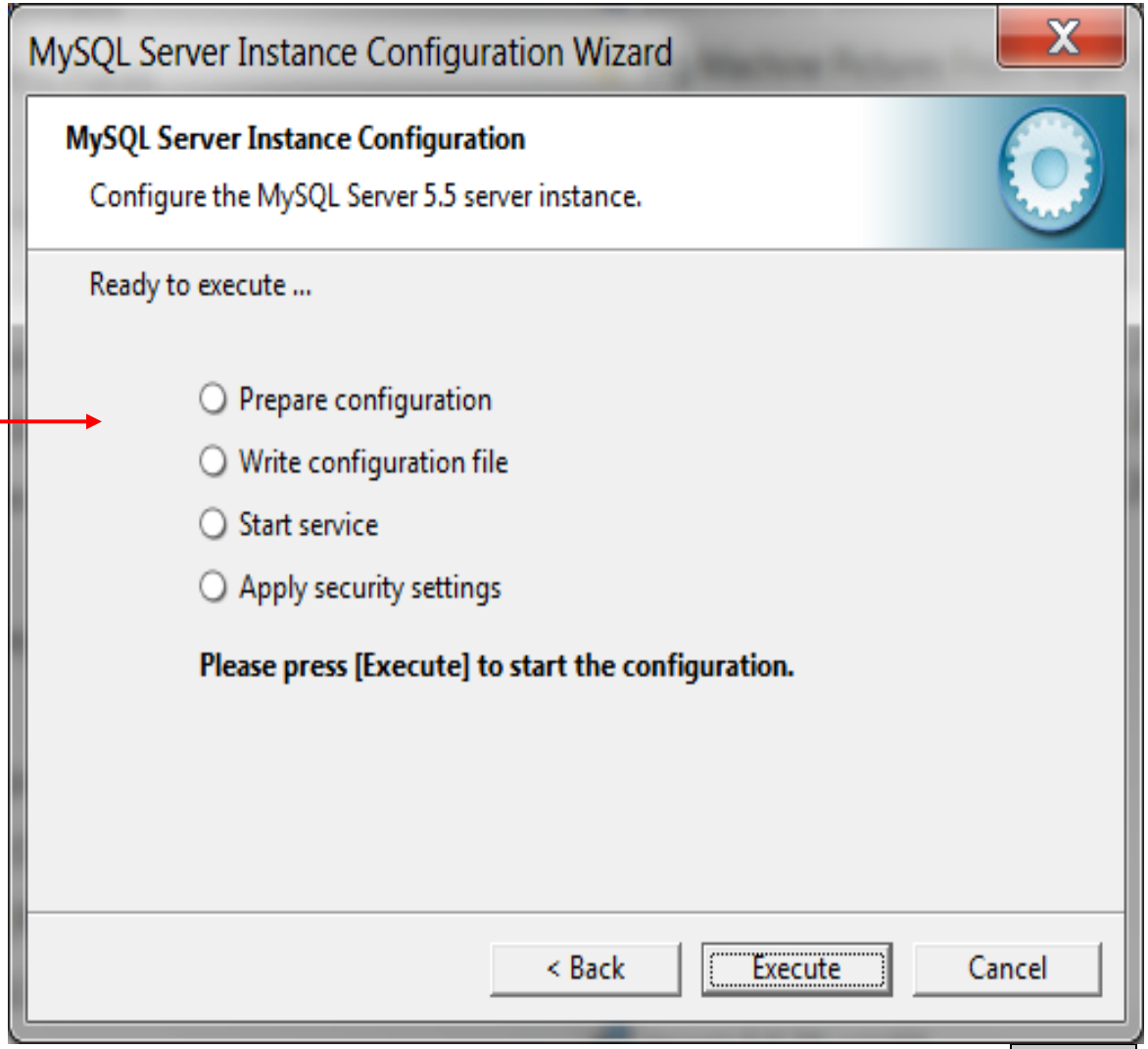
Do not enable this option



Installing MySQL 5.5.29 (cont.)

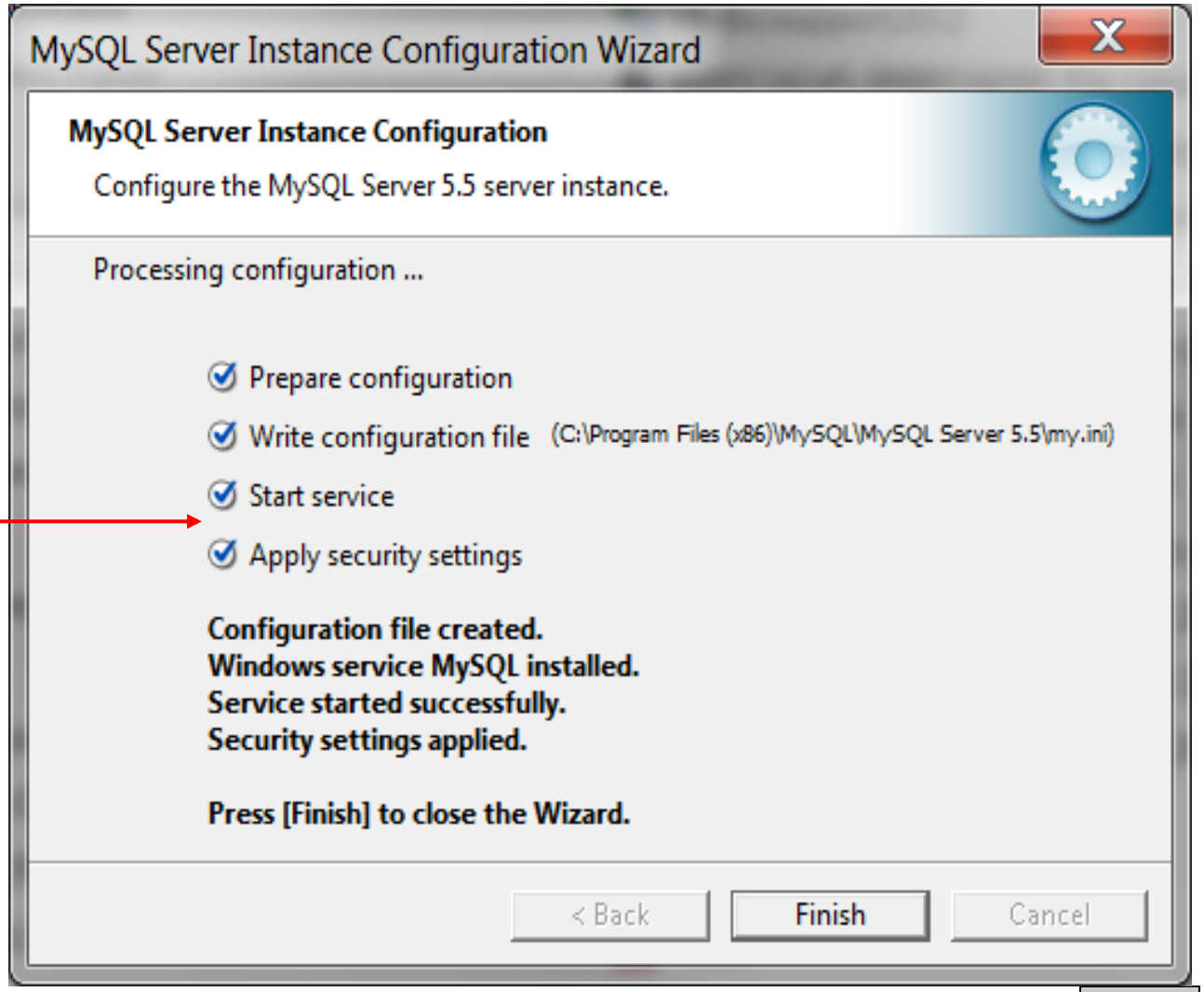
Configuration is about to begin. Now cross your fingers, toes, and anything else you have, take a deep breath, click the Execute button and close your eyes for a few seconds.

When they all have green check marks in them – you're good to go!



Installing MySQL 5.5.29 (cont.)

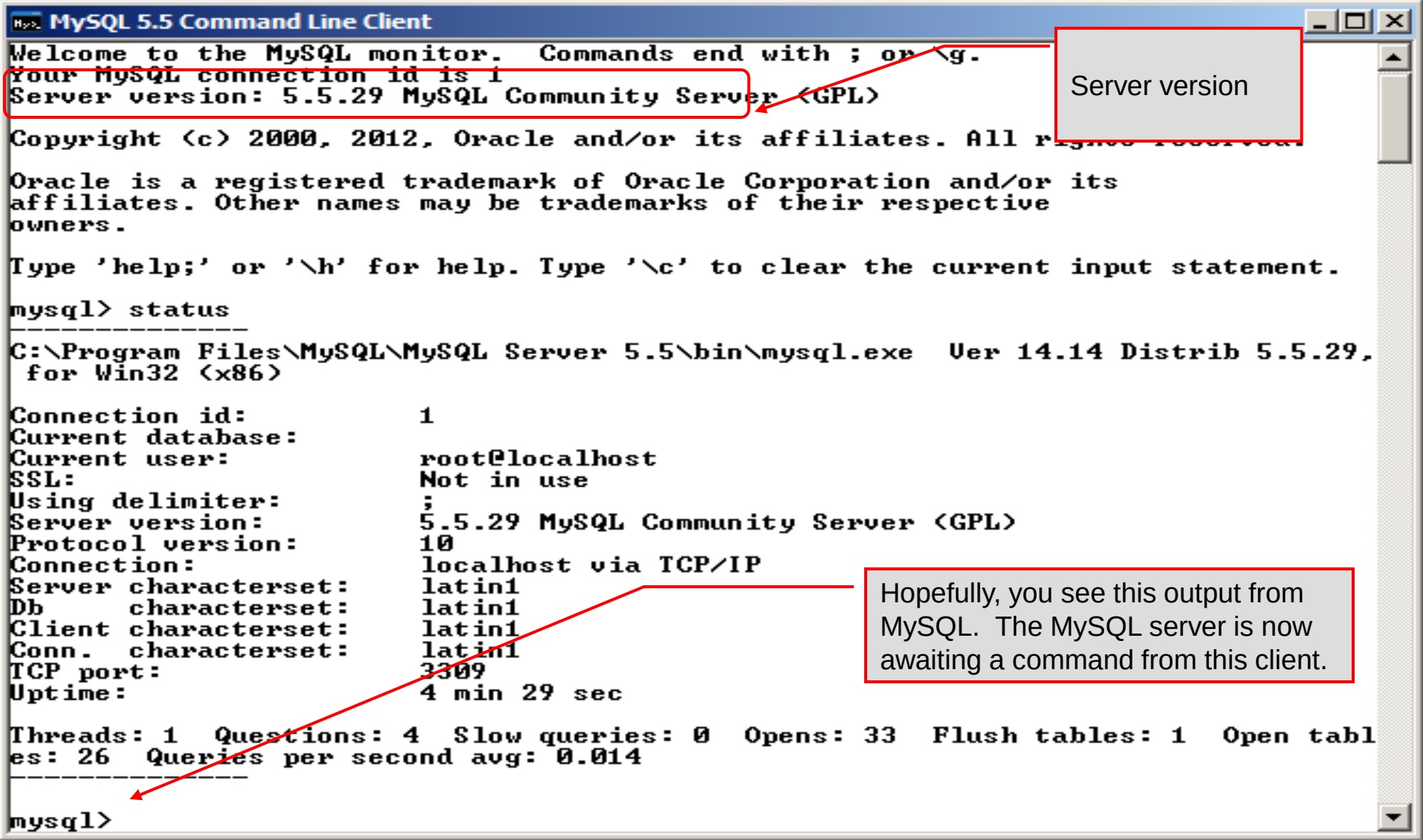
You've successfully installed MySQL!!



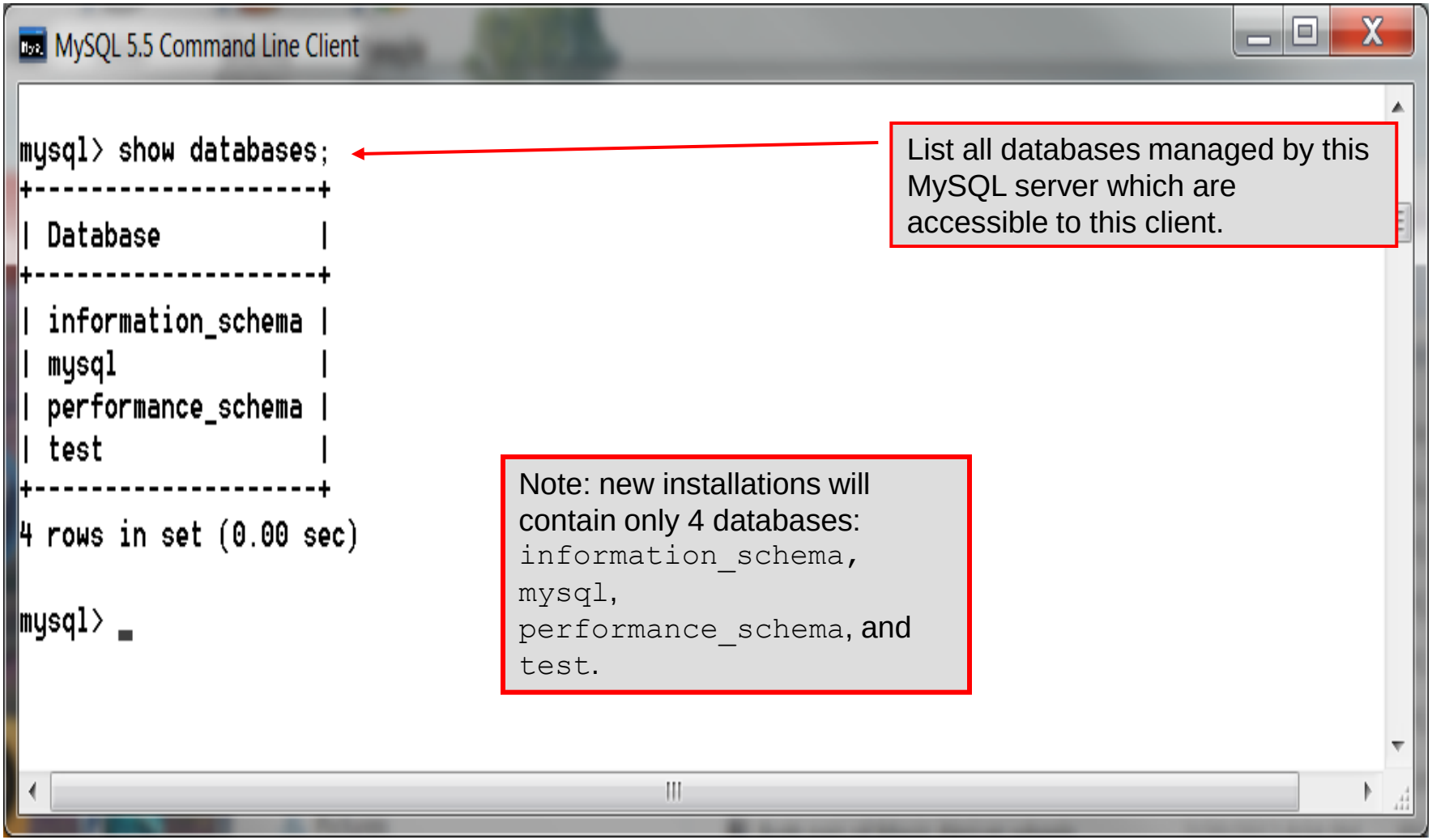
Running MySQL 5.5.29

- 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 5.5 Command Line 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.

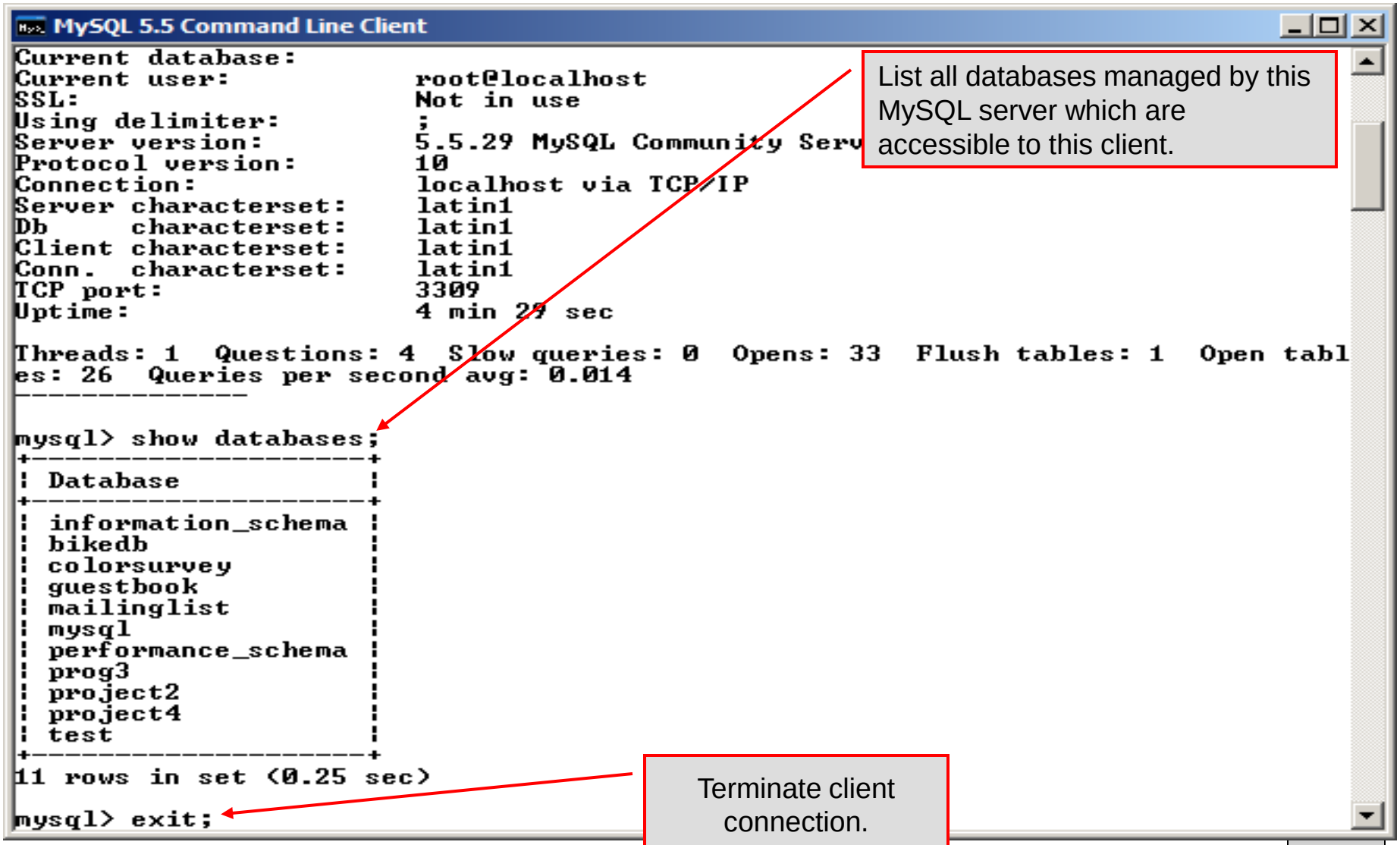
Running MySQL 5.5.29 (cont.)



Running MySQL 5.5.29 (cont.)

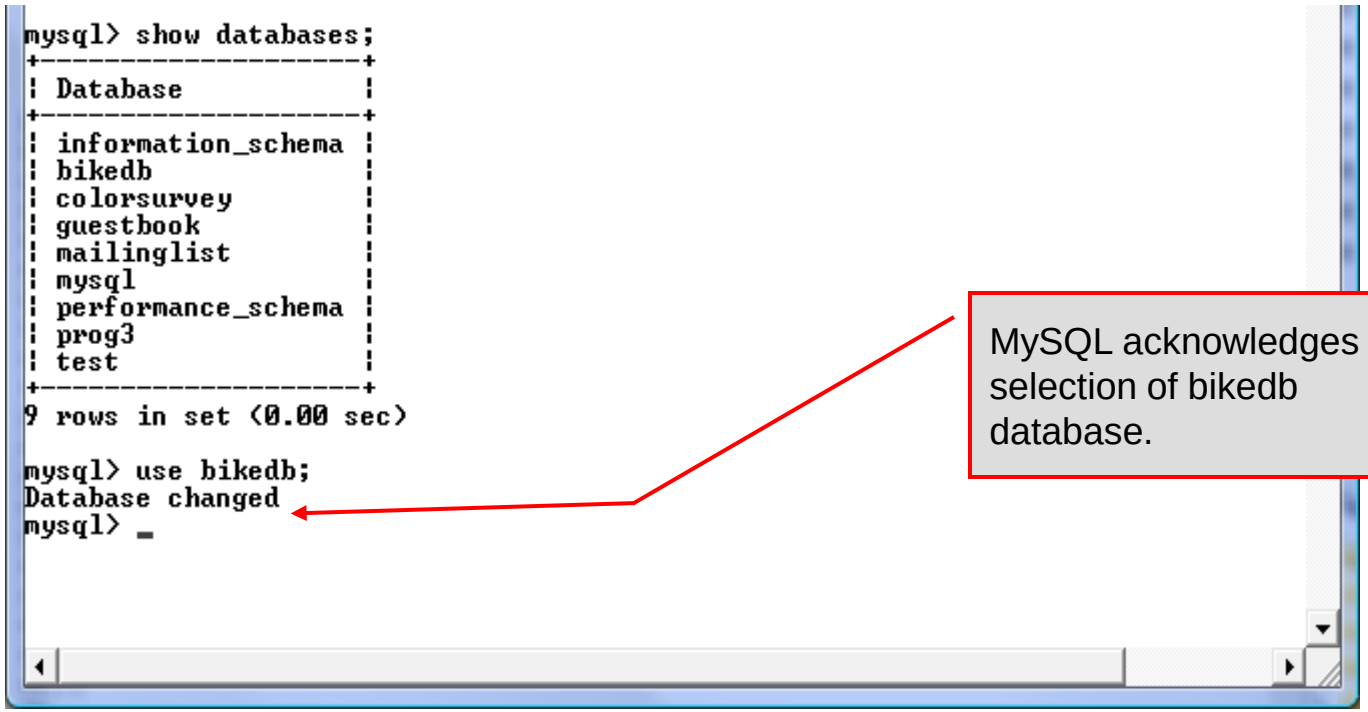


Running MySQL 5.5.29 (cont.)



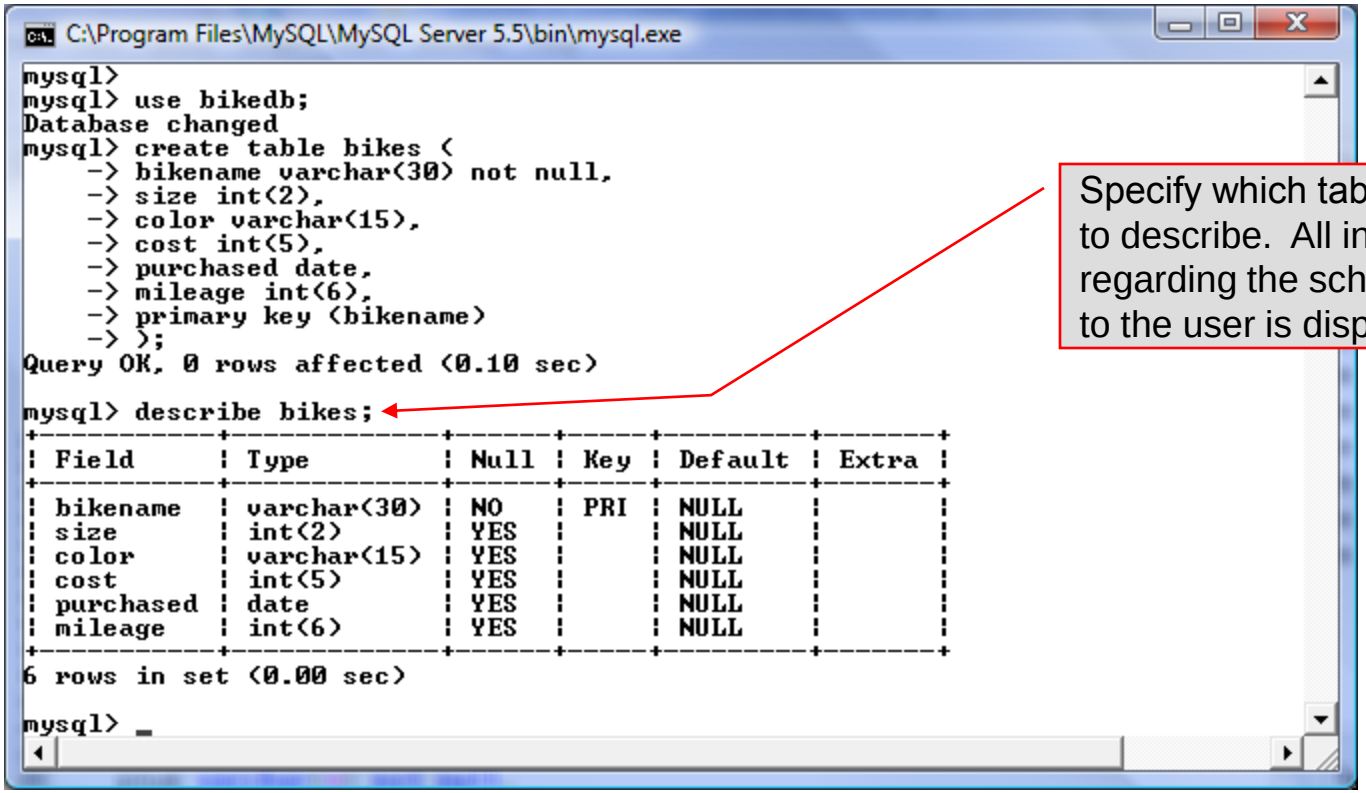
Specifying A Database Within MySQL

- 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.



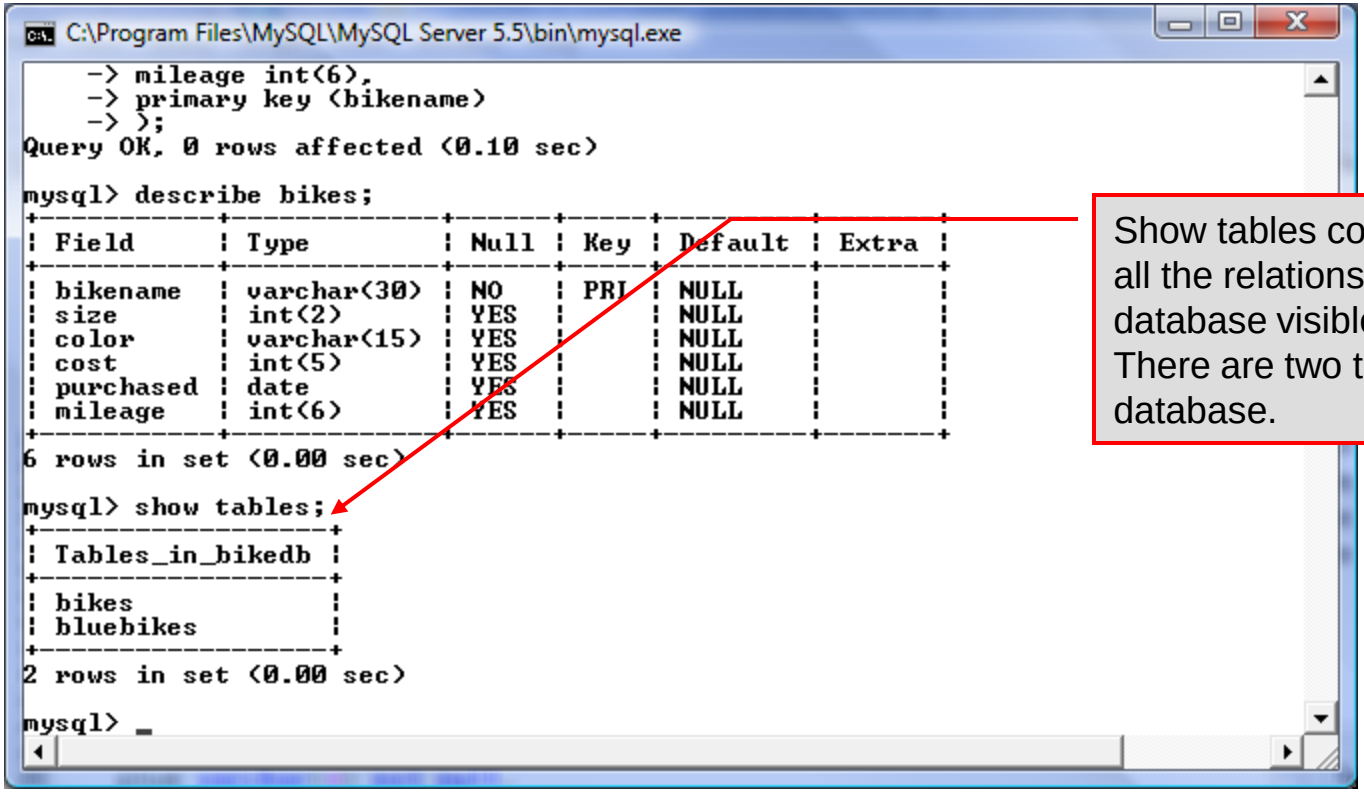
Viewing the Schema of a Relation

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



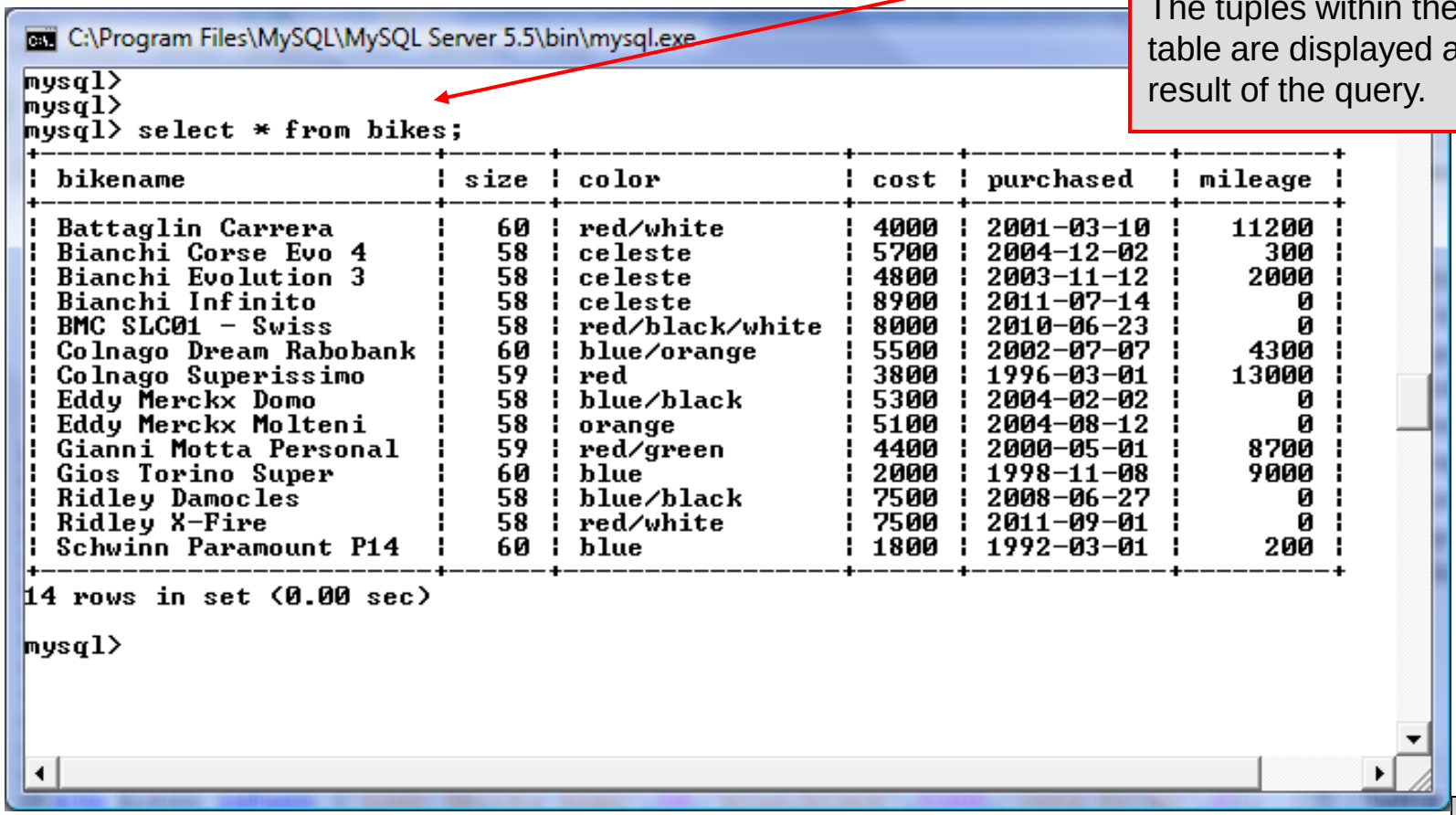
Viewing the Relations of a Database

- 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

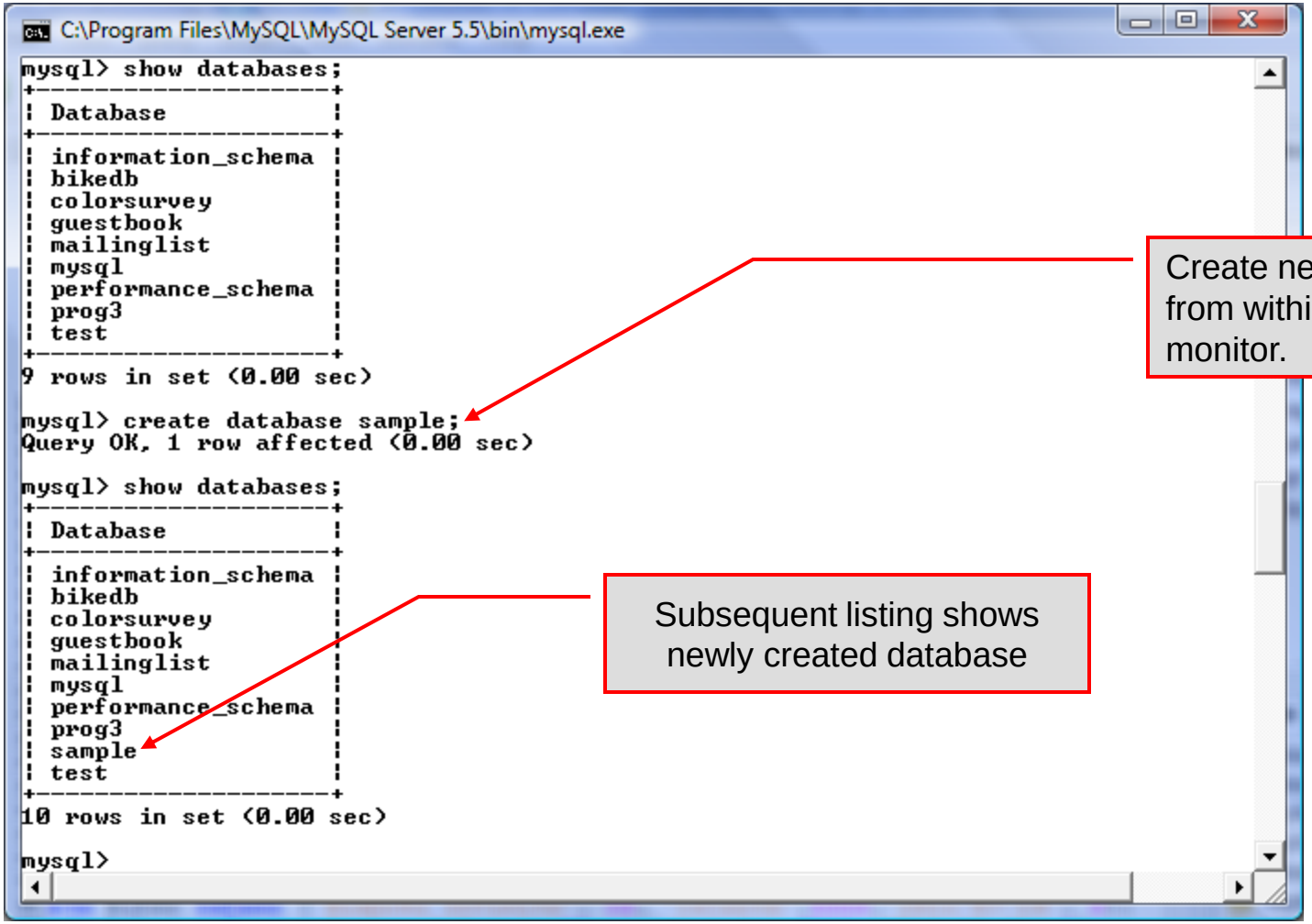
- Within the MySQL monitor, running an SQL query is straight forward. The example below illustrates a simple selection query on the `bikes` table of the `bikedb` database.



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

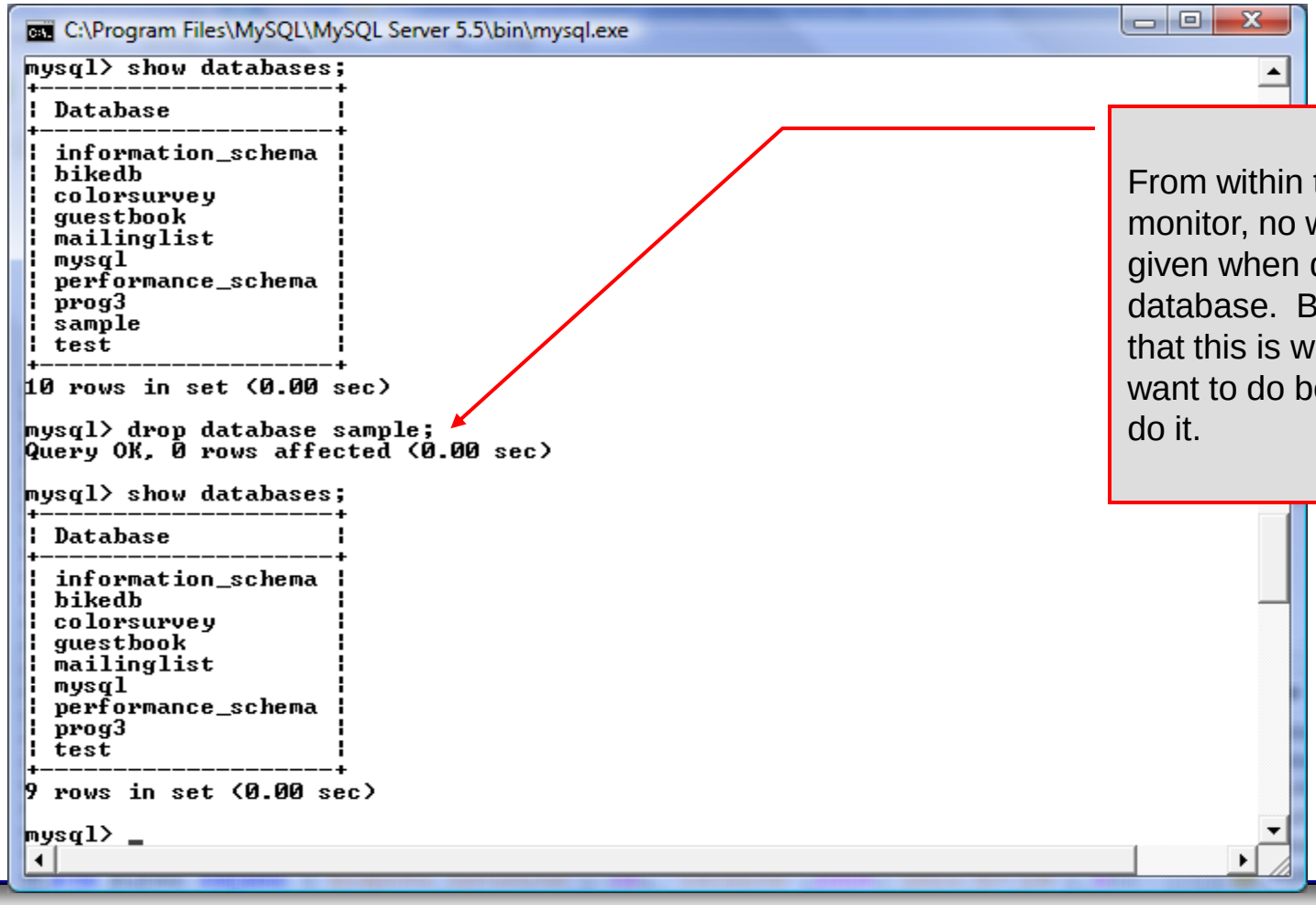
Creating a Database in MySQL

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



Dropping a Database in MySQL

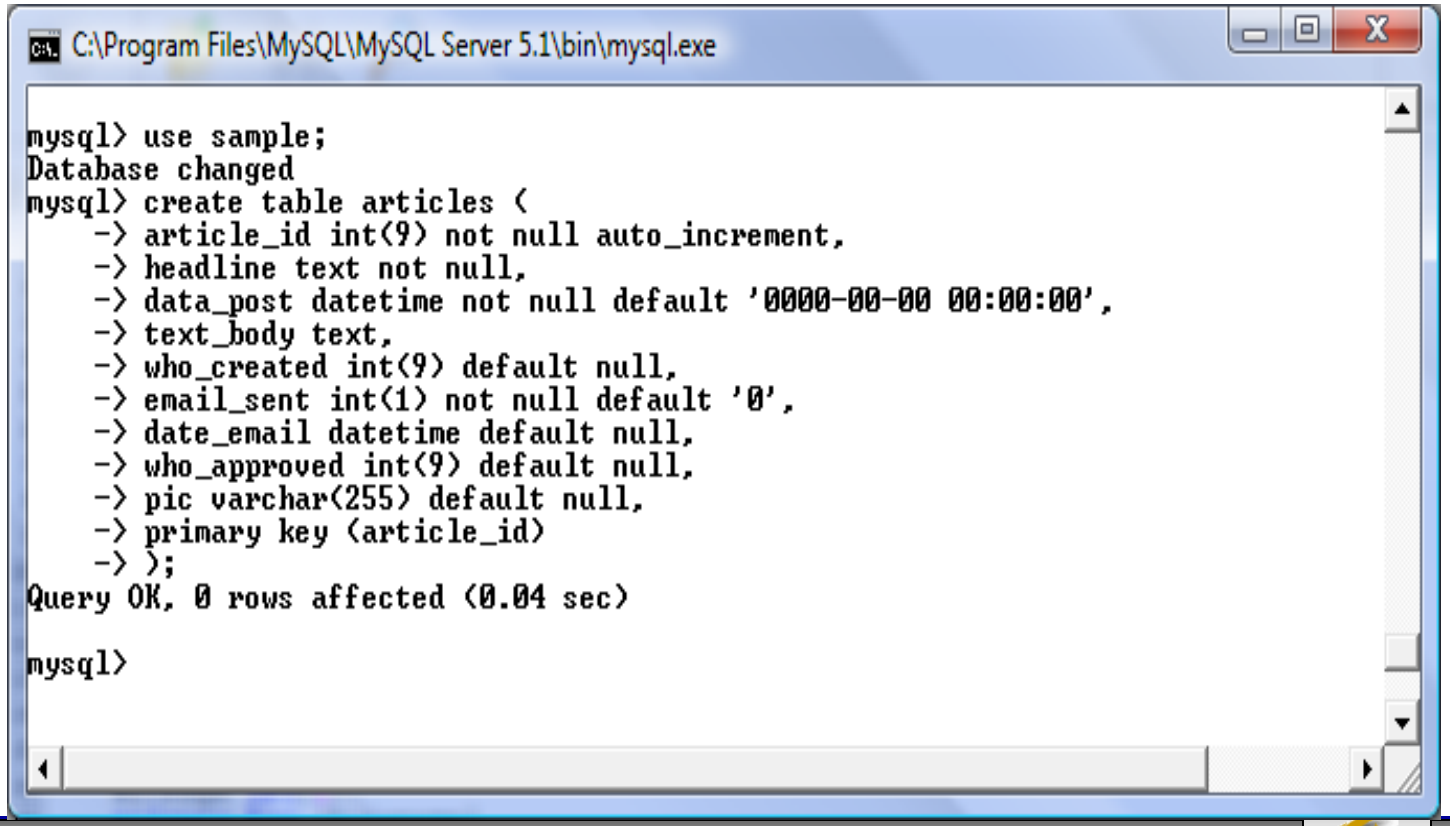
- From the MySQL monitor execute the `drop database <db name>` command.



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.

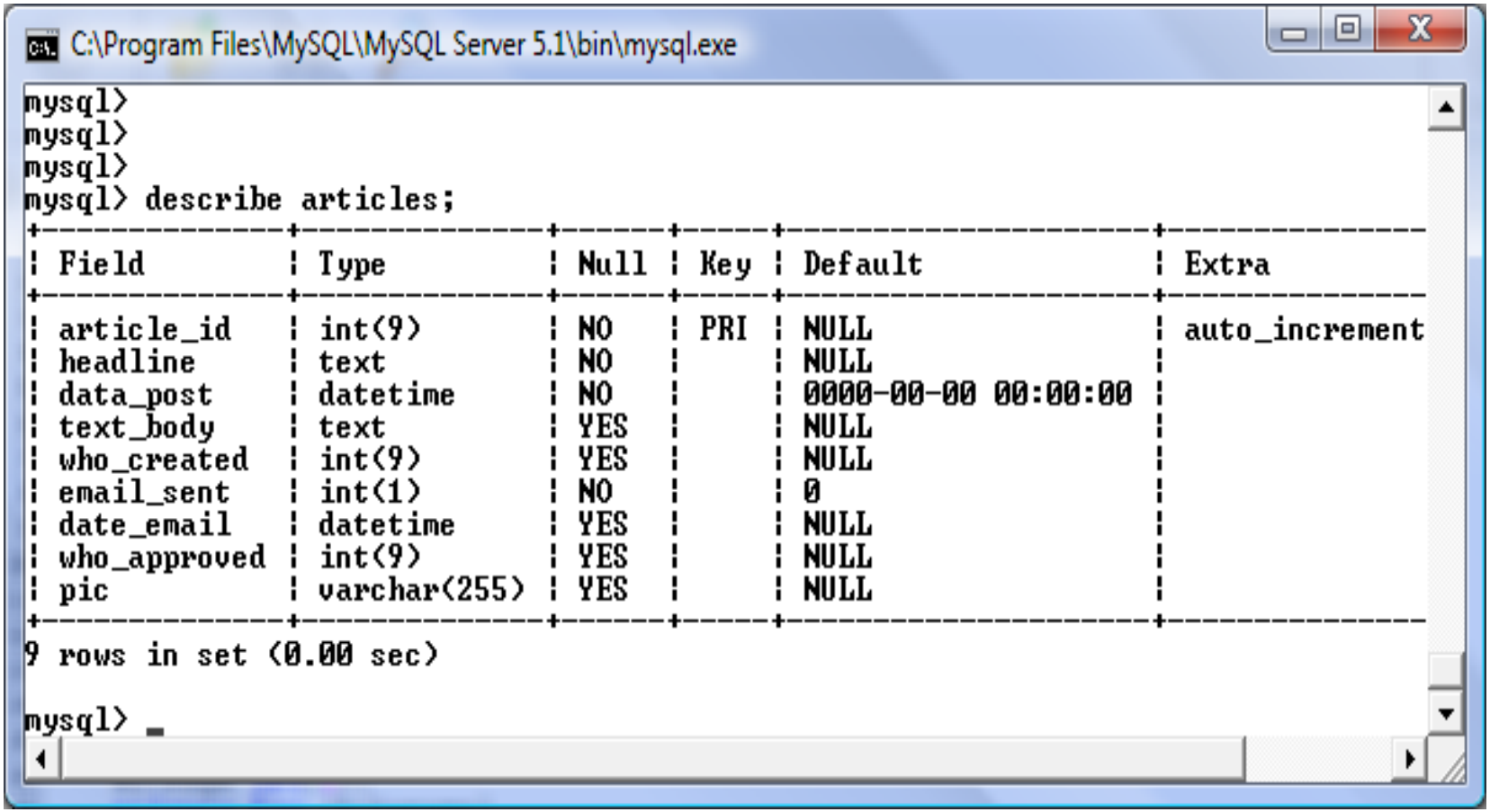
Manipulating Tables in MySQL

- 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 that describes the 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

C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql.exe

```
+-----------------------+------+-----------------+------+------------+---------+
| Battaglin Carrera     |   60 | red/white       | 4000 | 2001-03-10 |    1120 |
| Bianchi Corse Evo 4   |   58 | celeste         | 5700 | 2004-12-02 |     300 |
| Bianchi Evolution 3   |   58 | celeste         | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito      |   58 | celeste         | 8900 | 2011-07-14 |       0 |
| BMC SLC01 - Swiss     |   58 | red/black/white | 8000 | 2010-06-23 |       0 |
| Colnago Dream Rabobank |   60 | blue/orange     | 5500 | 2002-07-07 |     430 |
| Colnago Superissimo   |   59 | red             | 3800 | 1996-03-01 |    1300 |
| Eddy Merckx Domo      |   58 | blue/black      | 5300 | 2004-02-02 |       0 |
| Eddy Merckx Molteni   |   58 | orange          | 5100 | 2004-08-12 |       0 |
| Gianni Motta Personal |   59 | red/green       | 4400 | 2000-05-01 |     870 |
| Gios Torino Super     |   60 | blue            | 2000 | 1998-11-08 |     900 |
| Ridley Damocles       |   58 | blue/black      | 7500 | 2008-06-27 |       0 |
| Ridley X-Fire         |   58 | red/white       | 7500 | 2011-09-01 |       0 |
| Schwinn Paramount P14 |   60 | blue            | 1800 | 1992-03-01 |      20 |
+-----------------------+------+-----------------+------+------------+---------+
14 rows in set (0.00 sec)

mysql> create temporary table celestebikes
    -> select *
    -> from bikes
    -> where color = "celeste";
Query OK, 3 rows affected (0.11 sec)
Records: 3  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 | 2004-12-02 |     300 |
| Bianchi Evolution 3 |   58 | celeste | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito    |   58 | celeste | 8900 | 2011-07-14 |       0 |
+-----------------+------+---------+------+------------+---------+
3 rows in set (0.00 sec)
```

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 5.5.20.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.)

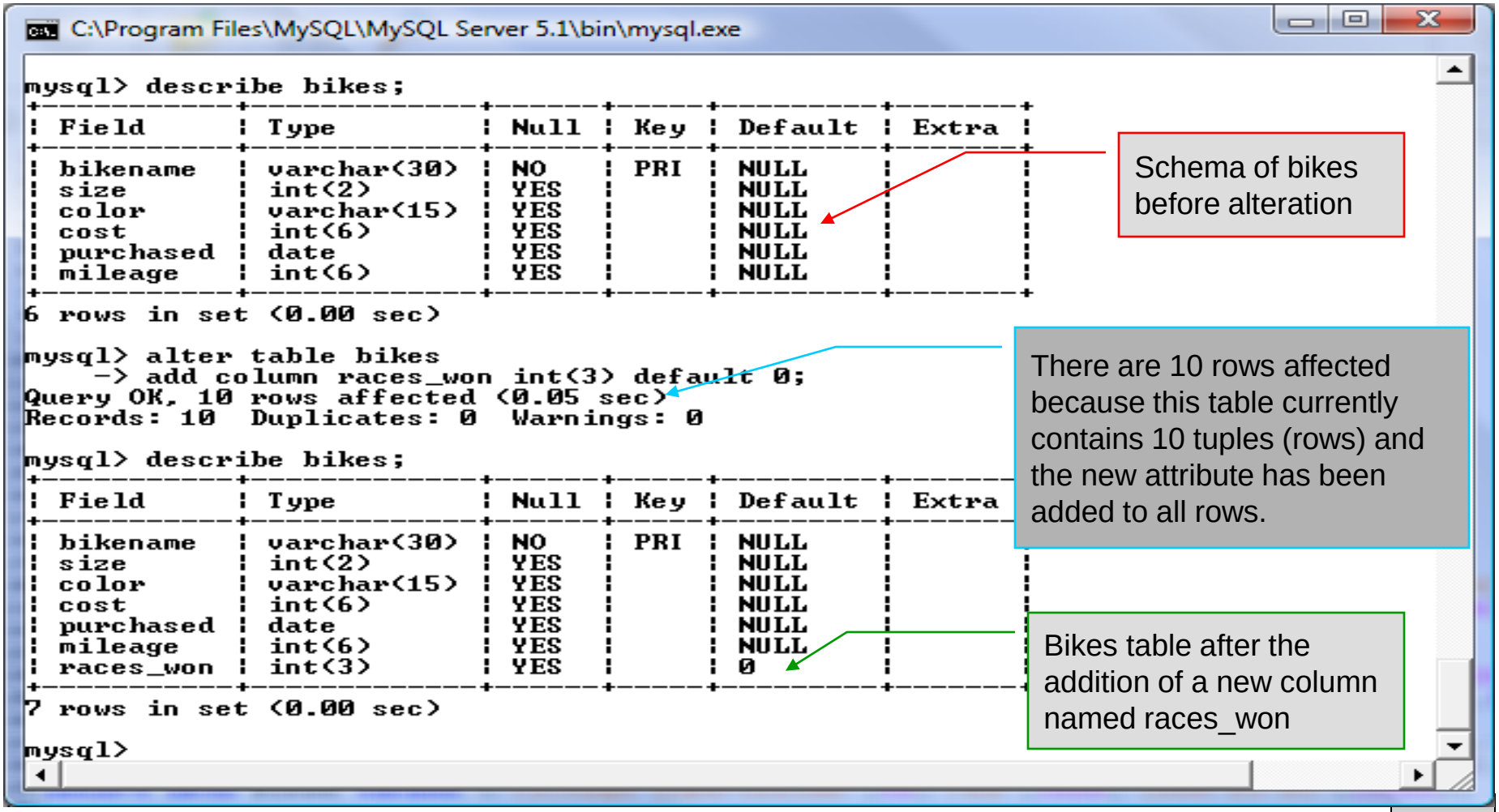
Action Syntax	Action Performed
`add [column] column_declaration [first | after column_name]`	Add a column to the table
`alter [column] column_name {set default literal | drop default}`	Specify new default value for a column or remove old default
`change [column] column_name column_declaration`	Modify 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.

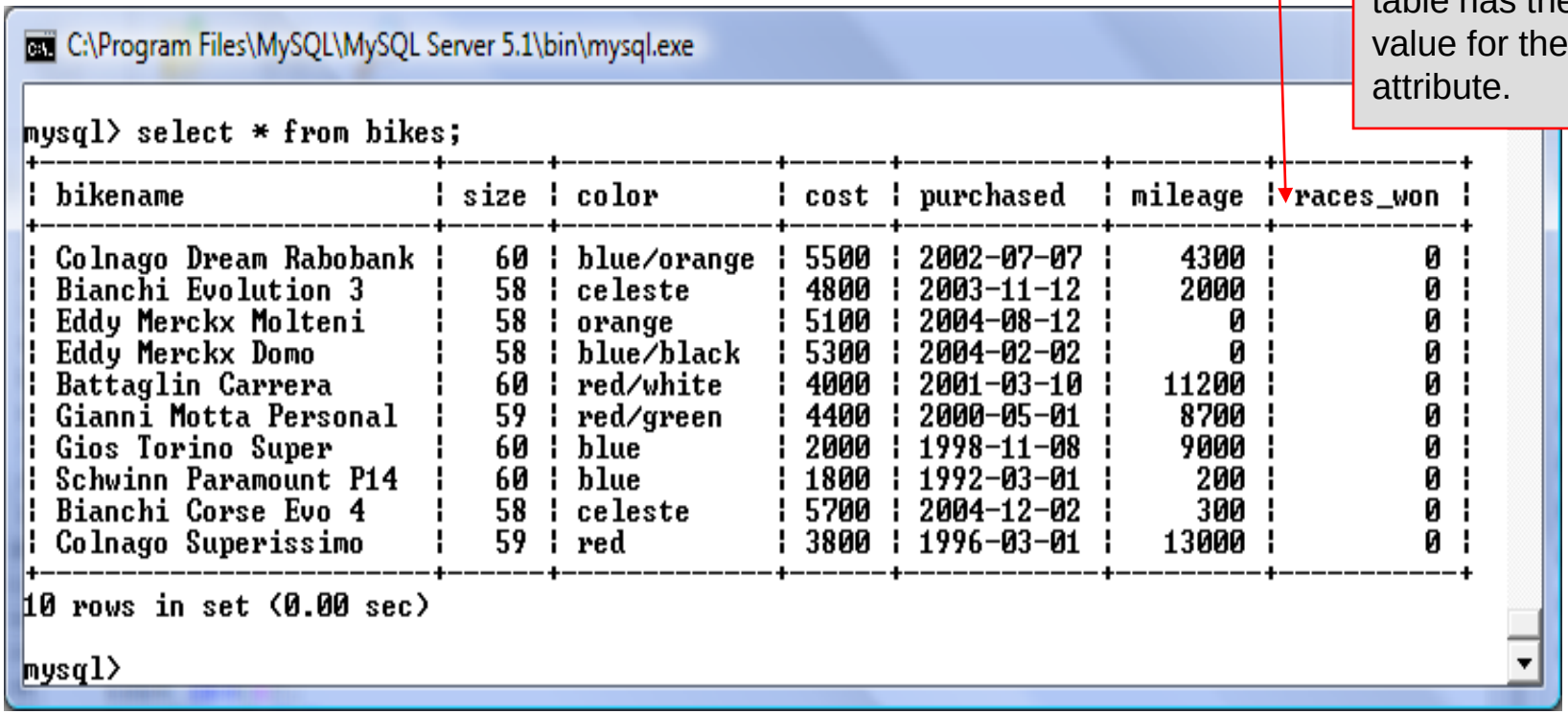
Altering A Table (cont.)

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



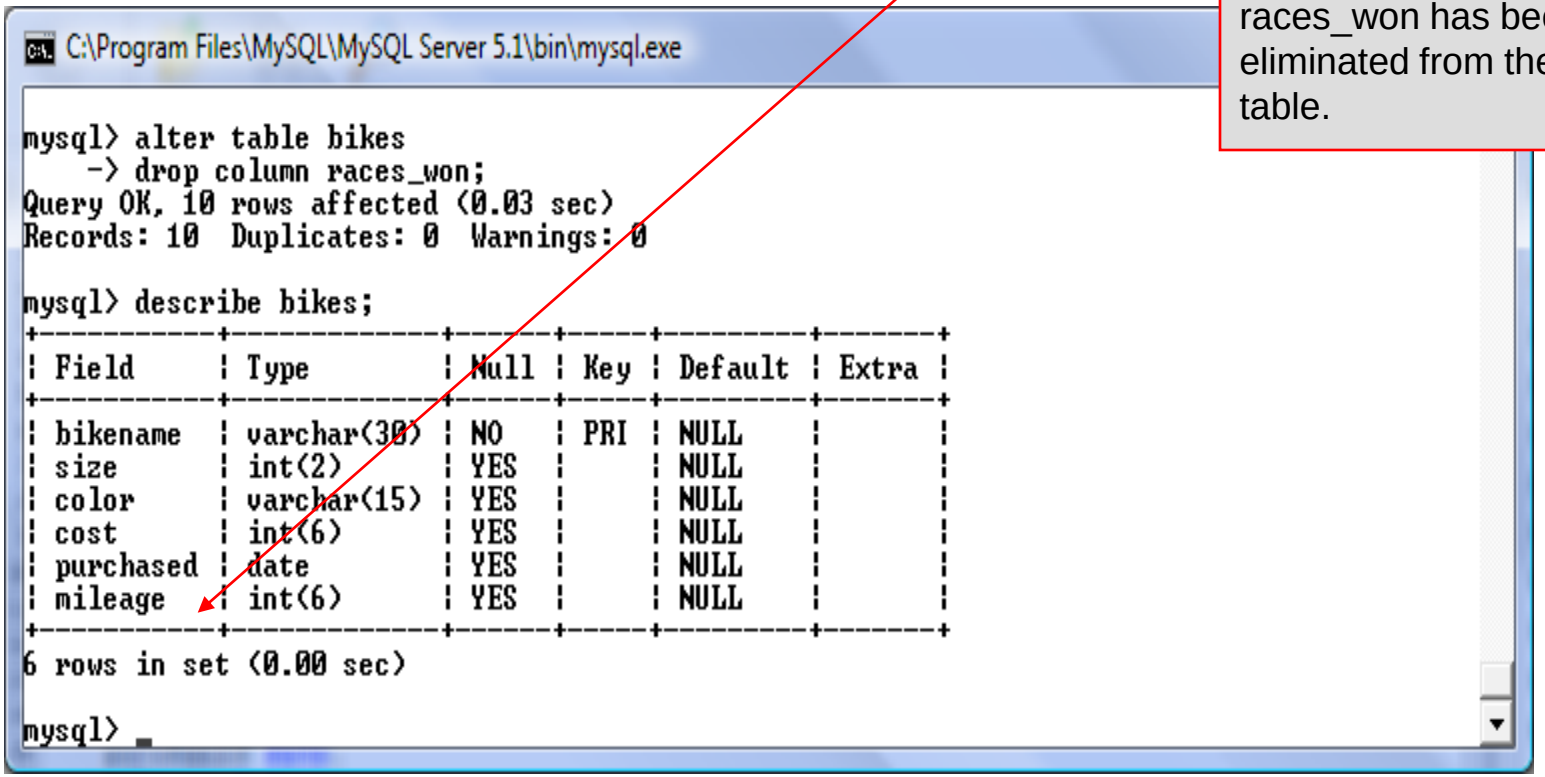
Altering A Table (cont.)

- 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.



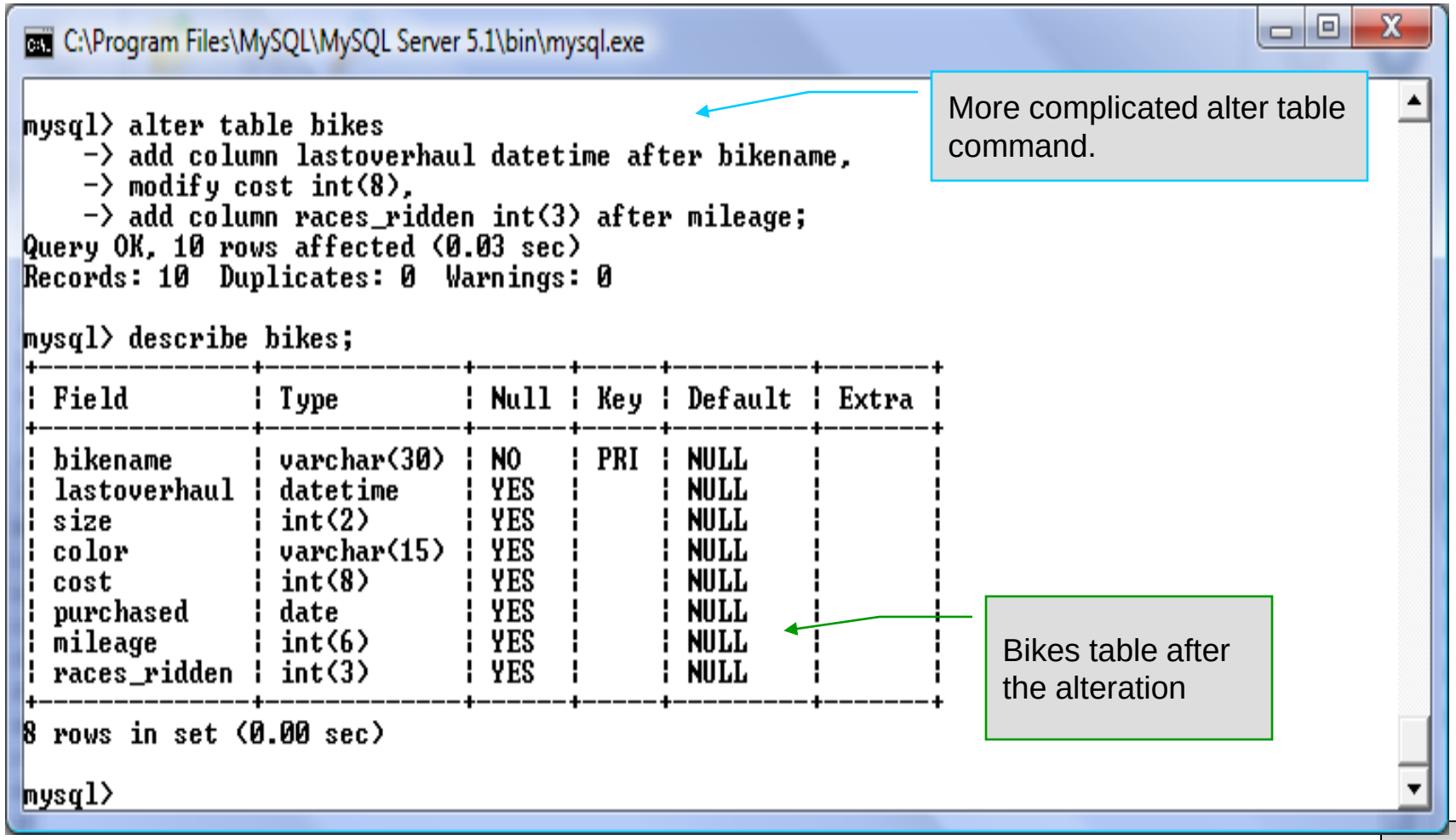
Altering A Table (cont.)

- 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.



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

C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql.exe

```
+-----------------------+------+-----------------+------+------------+---------+
| Battaglin Carrera     |   60 | red/white       | 4000 | 2001-03-10 |   11200 |
| Bianchi Corse Evo 4   |   58 | celeste         | 5700 | 2004-12-02 |     300 |
| Bianchi Evolution 3   |   58 | celeste         | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito      |   58 | celeste         | 8900 | 2011-07-14 |       0 |
| BMC SLC01 - Swiss     |   58 | red/black/white | 8000 | 2010-06-23 |       0 |
| Colnago Dream Rabobank |   60 | blue/orange     | 5500 | 2002-07-07 |    4300 |
| Colnago Superissimo   |   59 | red             | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo      |   58 | blue/black      | 5300 | 2004-02-02 |       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 |
| Ridley Damocles       |   58 | blue/black      | 7500 | 2008-06-27 |       0 |
| Ridley X-Fire         |   58 | red/white       | 7500 | 2011-09-01 |       0 |
| Schwinn Paramount P14 |   60 | blue            | 1800 | 1992-03-01 |     200 |
+-----------------------+------+-----------------+------+------------+---------+
14 rows in set (0.00 sec)

mysql> insert into bikes
    -> set bikename = "Eddy Merckx EM7",
    -> cost=9500,
    -> mileage=100,
    -> purchased="2011-01-01",
    -> color="red/white/blue",
    -> 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 | 2001-03-10 |   11200 |
| Bianchi Corse Evo 4   |   58 | celeste         | 5700 | 2004-12-02 |     300 |
| Bianchi Evolution 3   |   58 | celeste         | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito      |   58 | celeste         | 8900 | 2011-07-14 |       0 |
| BMC SLC01 - Swiss     |   58 | red/black/white | 8000 | 2010-06-23 |       0 |
| Colnago Dream Rabobank |   60 | blue/orange     | 5500 | 2002-07-07 |    4300 |
| Colnago Superissimo   |   59 | red             | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo      |   58 | blue/black      | 5300 | 2004-02-02 |       0 |
| Eddy Merckx EM7       |   58 | red/white/blue  | 9500 | 2011-01-01 |     100 |
| 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 |
| Ridley Damocles       |   58 | blue/black      | 7500 | 2008-06-27 |       0 |
| Ridley X-Fire         |   58 | red/white       | 7500 | 2011-09-01 |       0 |
| Schwinn Paramount P14 |   60 | blue            | 1800 | 1992-03-01 |     200 |
+-----------------------+------+-----------------+------+------------+---------+
15 rows in set (0.00 sec)

mysql> _
```

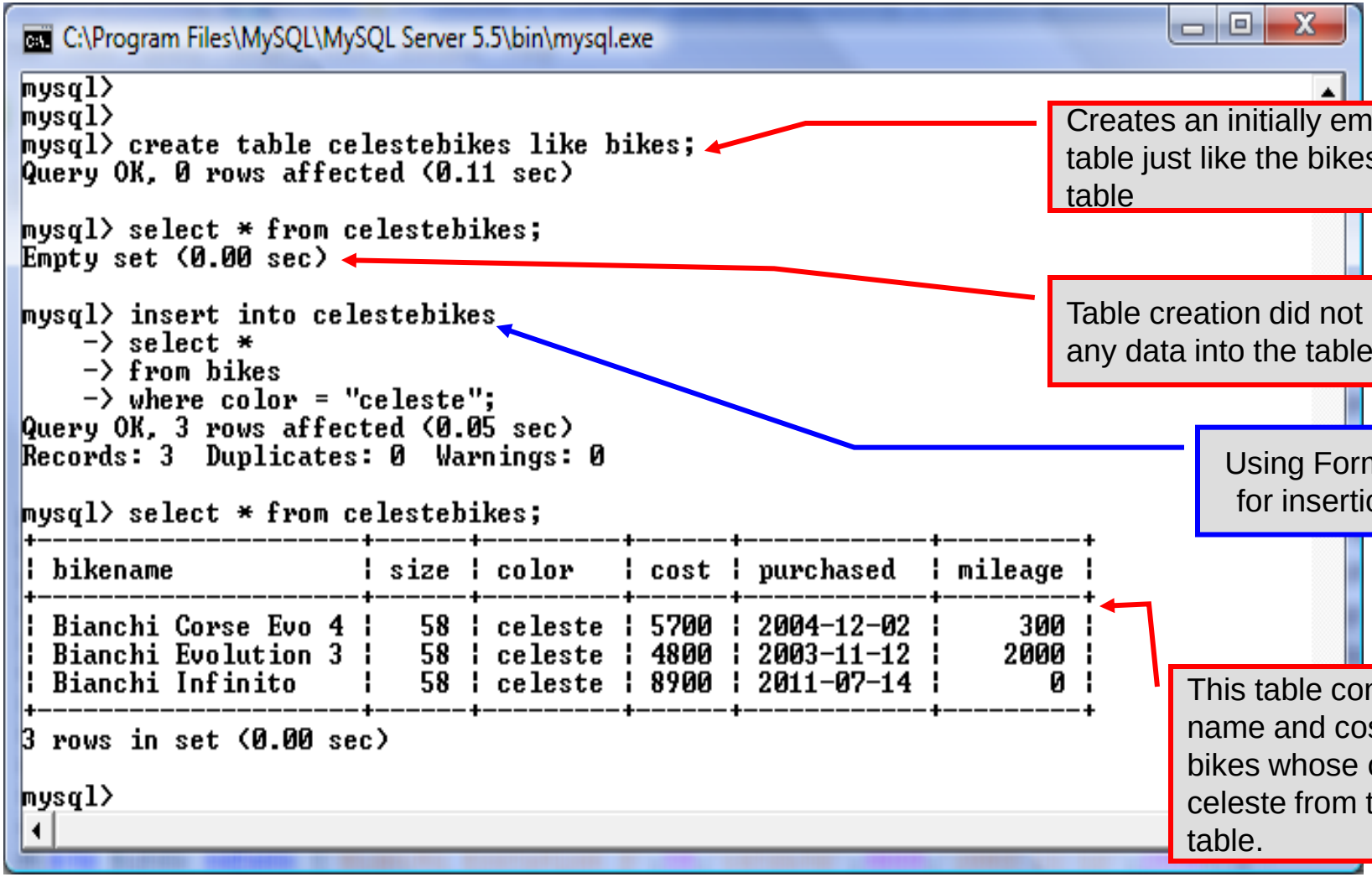
Using Form 1 for insertion – attribute order is not important.

C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql.exe

```
mysql> select * from bikes;
+-------------------------+------+-----------------+------+------------+---------+
| bikename                | size | color           | cost | purchased  | mileage |
+-------------------------+------+-----------------+------+------------+---------+
| Battaglin Carrera       |   60 | red/white       | 4000 | 2001-03-10 |   11200 |
| Bianchi Corse Evo 4     |   58 | celeste         | 5700 | 2004-12-02 |     300 |
| Bianchi Evolution 3     |   58 | celeste         | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito        |   58 | celeste         | 8900 | 2011-07-14 |       0 |
| BMC SLC01 - Swiss       |   58 | red/black/white | 8000 | 2010-06-23 |       0 |
| Colnago Dream Rabobank  |   60 | blue/orange     | 5500 | 2002-07-07 |    4300 |
| Colnago Superissimo     |   59 | red             | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo        |   58 | blue/black      | 5300 | 2004-02-02 |       0 |
| Eddy Merckx EM7         |   58 | red/white/blue  | 9500 | 2011-01-01 |     100 |
| 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 |
| Ridley Damocles         |   58 | blue/black      | 7500 | 2008-06-27 |       0 |
| Ridley X-Fire           |   58 | red/white       | 7500 | 2011-09-01 |       0 |
| Schwinn Paramount P14   |   60 | blue            | 1800 | 1992-03-01 |     200 |
+-------------------------+------+-----------------+------+------------+---------+
15 rows in set (0.00 sec)

mysql> insert into bikes
    -> values("Ridley Crosswind",58,"black",6500,"2010-04-05",2000);
Query OK, 1 row affected (0.05 sec)

mysql> select * from bikes;
+-------------------------+------+-----------------+------+------------+---------+
| bikename                | size | color           | cost | purchased  | mileage |
+-------------------------+------+-----------------+------+------------+---------+
| Battaglin Carrera       |   60 | red/white       | 4000 | 2001-03-10 |   11200 |
| Bianchi Corse Evo 4     |   58 | celeste         | 5700 | 2004-12-02 |     300 |
| Bianchi Evolution 3     |   58 | celeste         | 4800 | 2003-11-12 |    2000 |
| Bianchi Infinito        |   58 | celeste         | 8900 | 2011-07-14 |       0 |
| BMC SLC01 - Swiss       |   58 | red/black/white | 8000 | 2010-06-23 |       0 |
| Colnago Dream Rabobank  |   60 | blue/orange     | 5500 | 2002-07-07 |    4300 |
| Colnago Superissimo     |   59 | red             | 3800 | 1996-03-01 |   13000 |
| Eddy Merckx Domo        |   58 | blue/black      | 5300 | 2004-02-02 |       0 |
| Eddy Merckx EM7         |   58 | red/white/blue  | 9500 | 2011-01-01 |     100 |
| 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 |
| Ridley Crosswind        |   58 | black           | 6500 | 2010-04-05 |    2000 |
| Ridley Damocles         |   58 | blue/black      | 7500 | 2008-06-27 |       0 |
| Ridley X-Fire           |   58 | red/white       | 7500 | 2011-09-01 |       0 |
| Schwinn Paramount P14   |   60 | blue            | 1800 | 1992-03-01 |     200 |
+-------------------------+------+-----------------+------+------------+---------+
16 rows in set (0.00 sec)

mysql> _
```

Using Form 2 for insertion – attribute order is important.

Examples: Inserting Data Into A Table



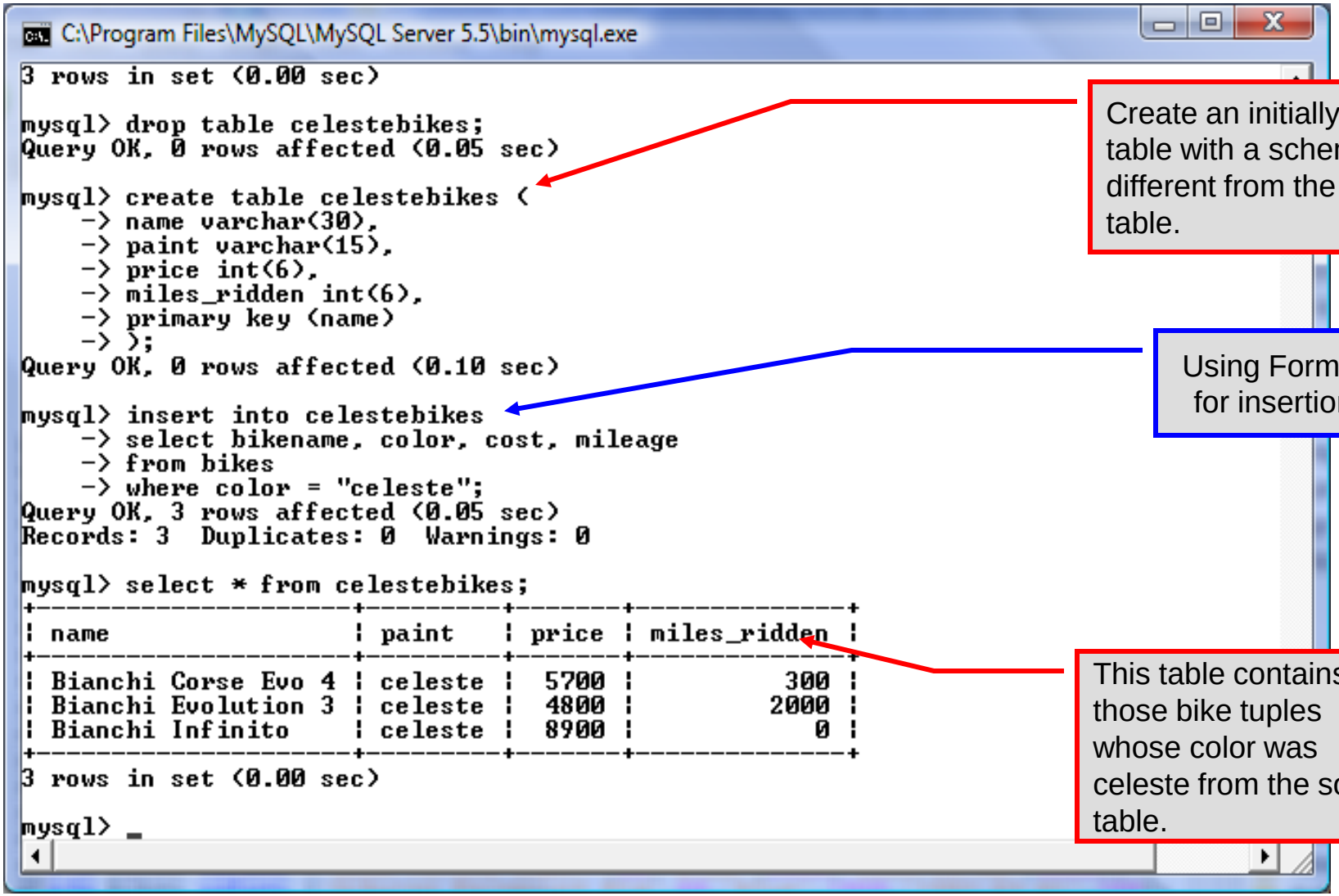
Creates an initially empty table just like the bikes table

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.

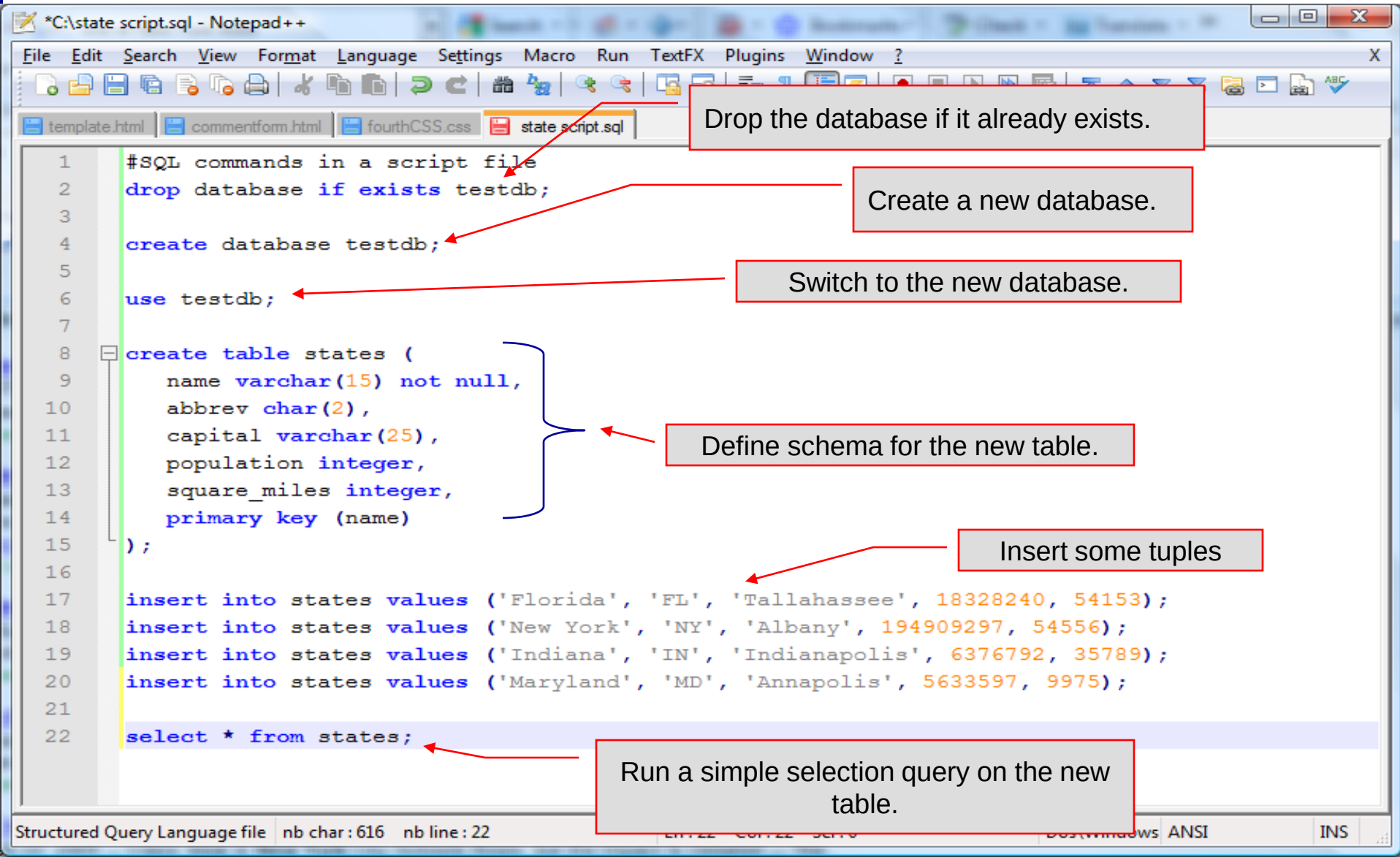
Examples: Inserting Data Into A Table



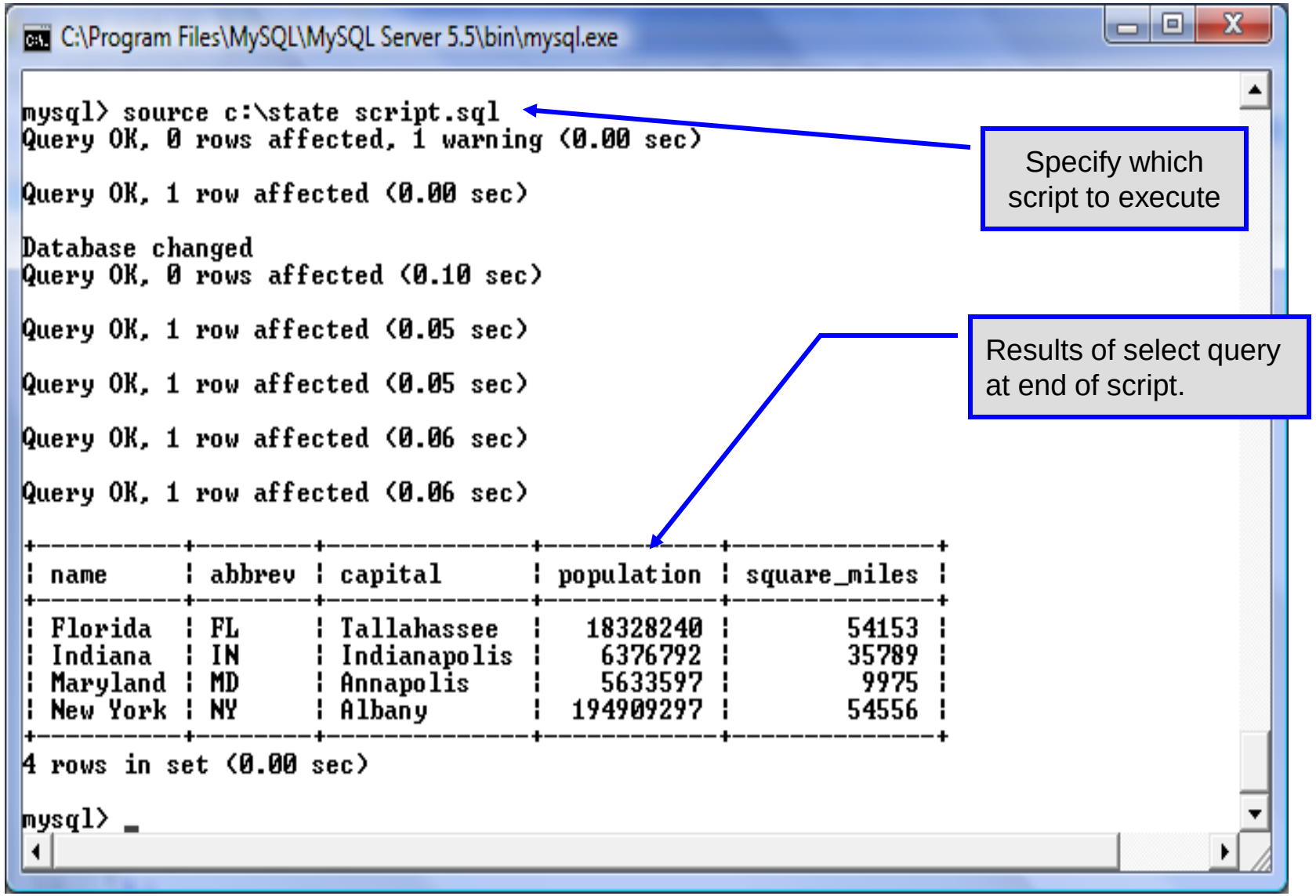
Using Scripts with MySQL

- 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. The third and more preferable option is to use the MySQL Workbench tool (see page 98 and on.)
- 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.)



Specify which script to execute

Results of select query at end of script.

Importing Data Using the `mysqlimport` Utility

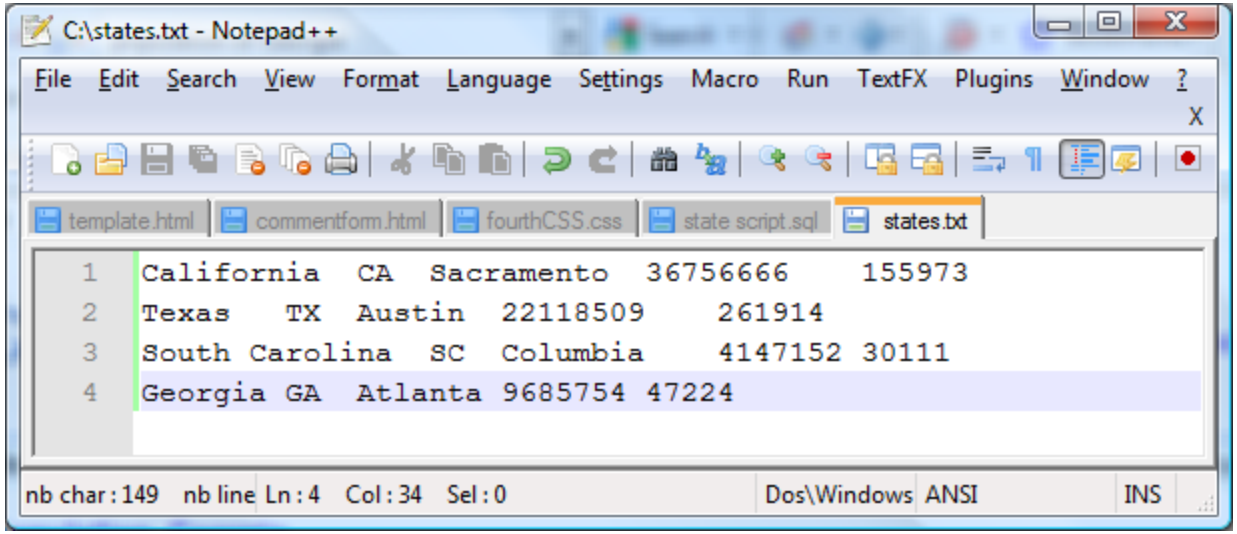
- As with many things in MySQL there are several ways to accomplish a specific task. For getting data into tables, the `mysqlimport` utility is also useful.
- The `mysqlimport` utility reads a range of data formats, including comma- and tab- delimited, and inserts the data into a specified database table.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

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

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

Importing Data Using the `mysqlimportUtility` (cont.)

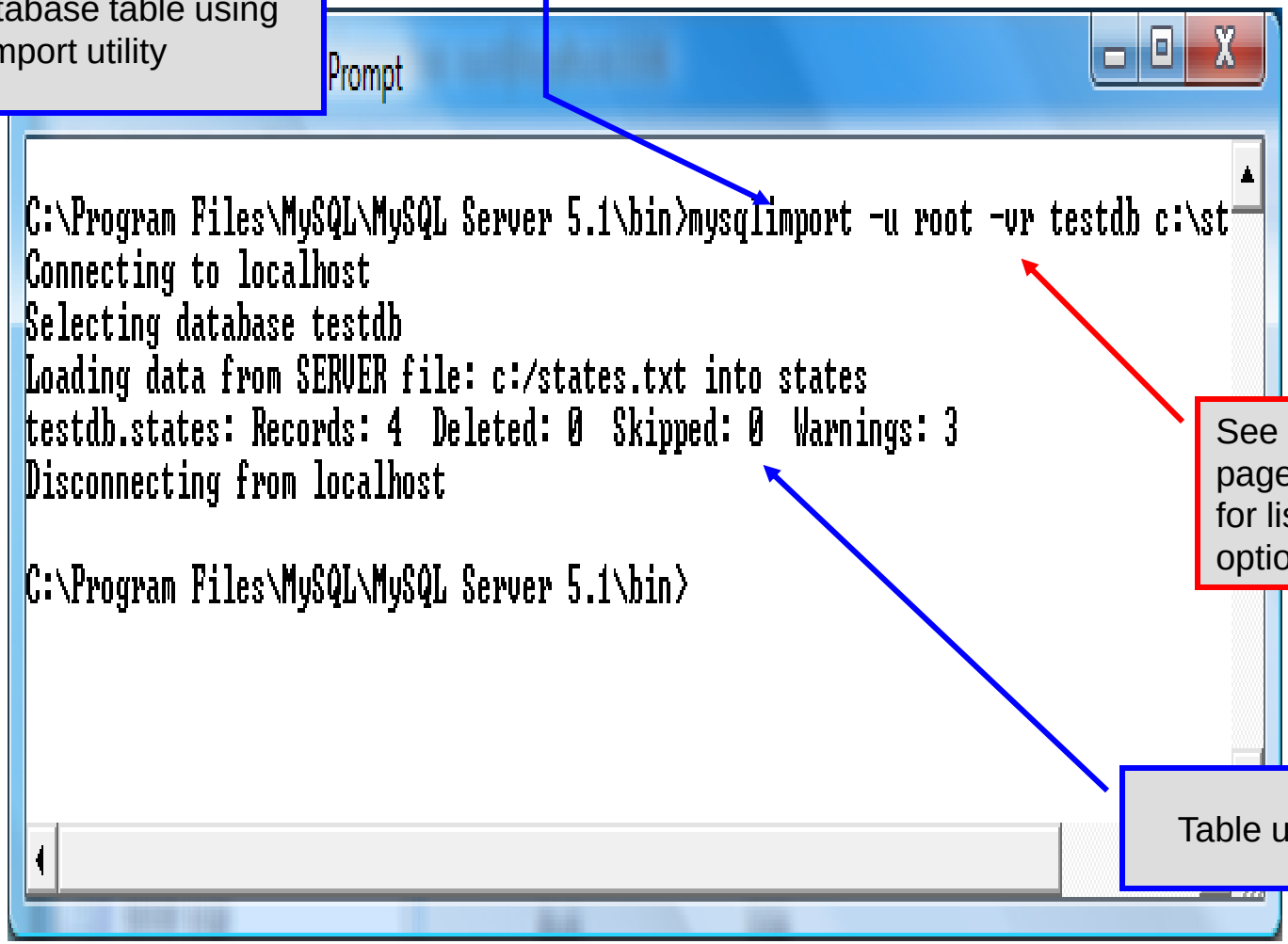
- The file shown below was created to import additional data into the states table within the testdb database used in the previous example.



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

Importing Data Using the `mysqlimportUtility`

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



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

Table updated

Importing Data Using the mysqlimportUtility

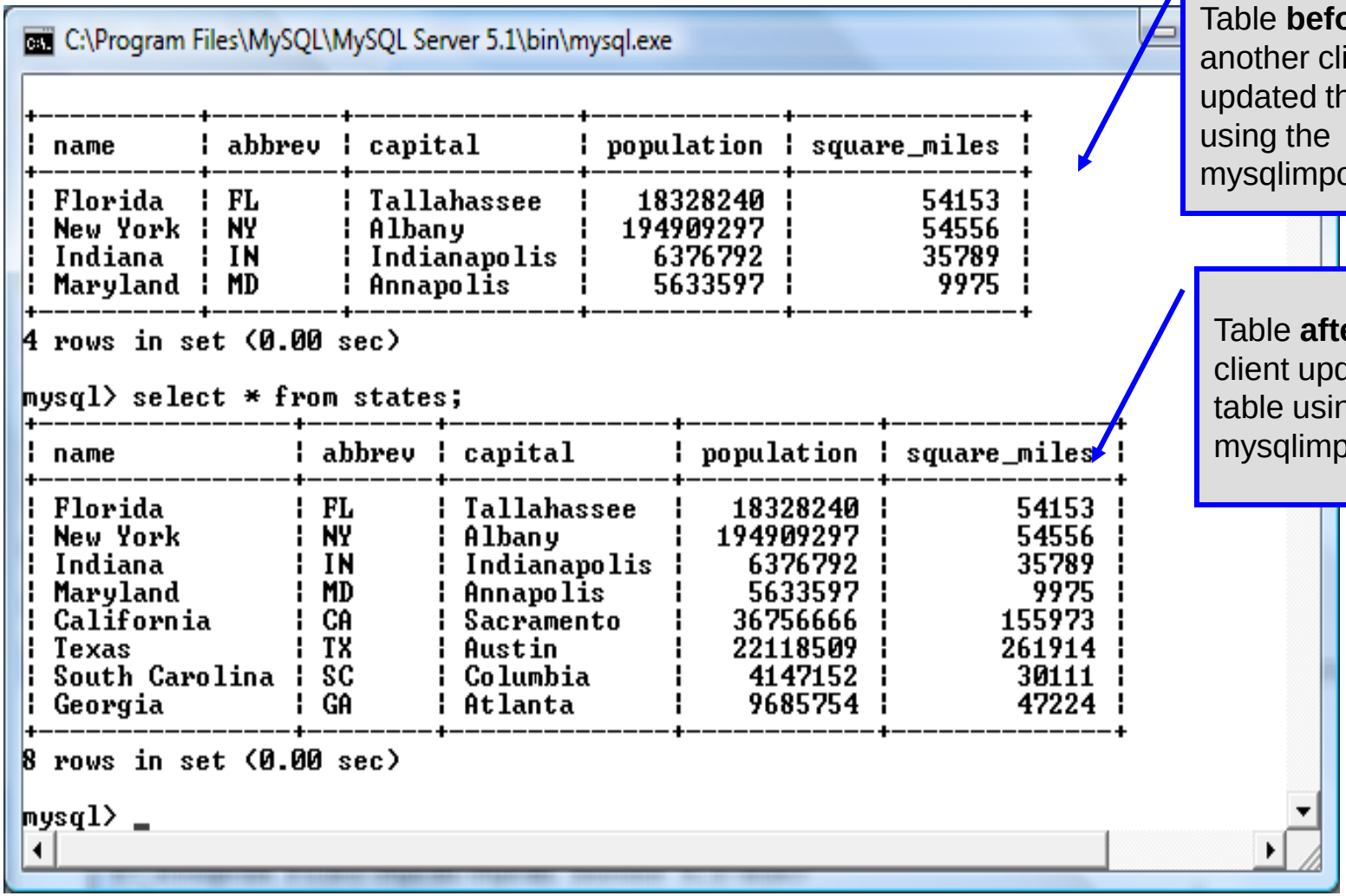


Table **before** another client updated the table using the mysqlimport utility.

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.

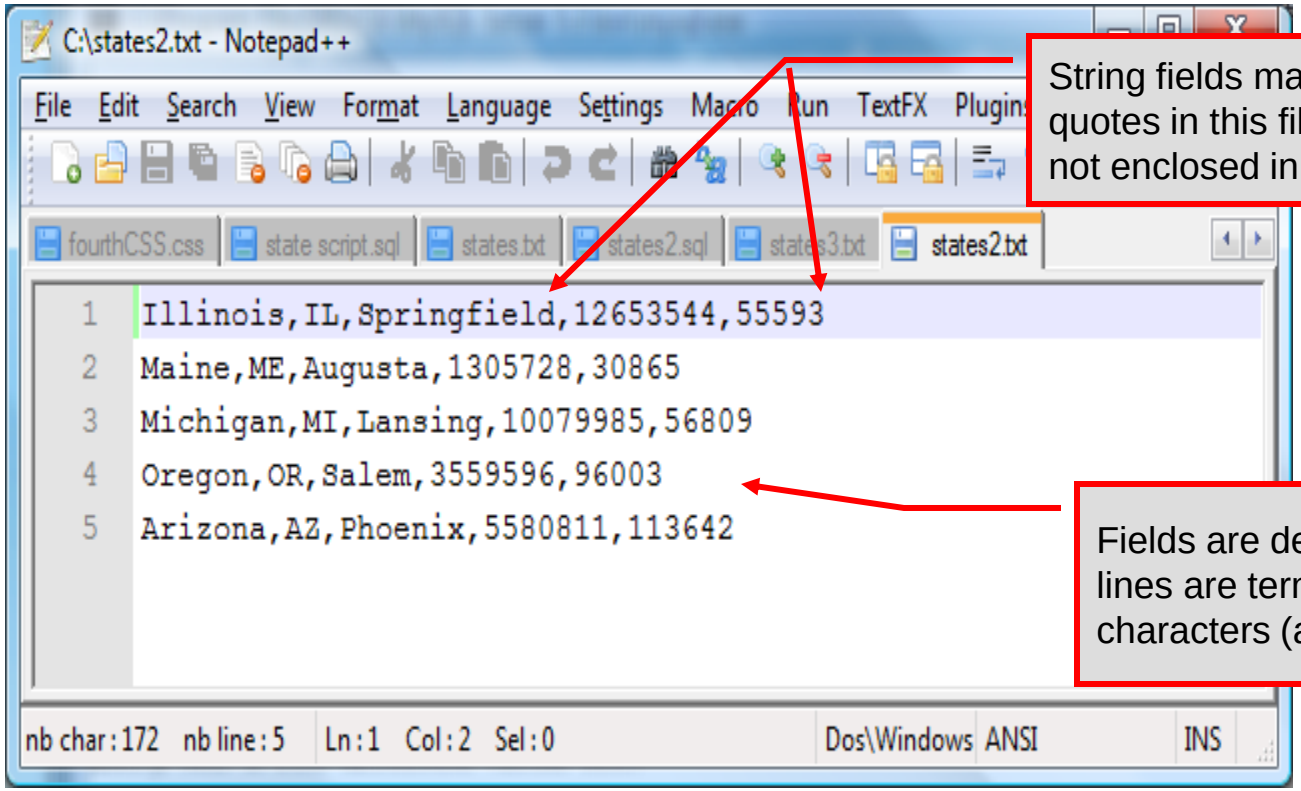
Same as –r and –i options in mysqlimport utility – either replace or ignore rows with duplicate keys.

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)

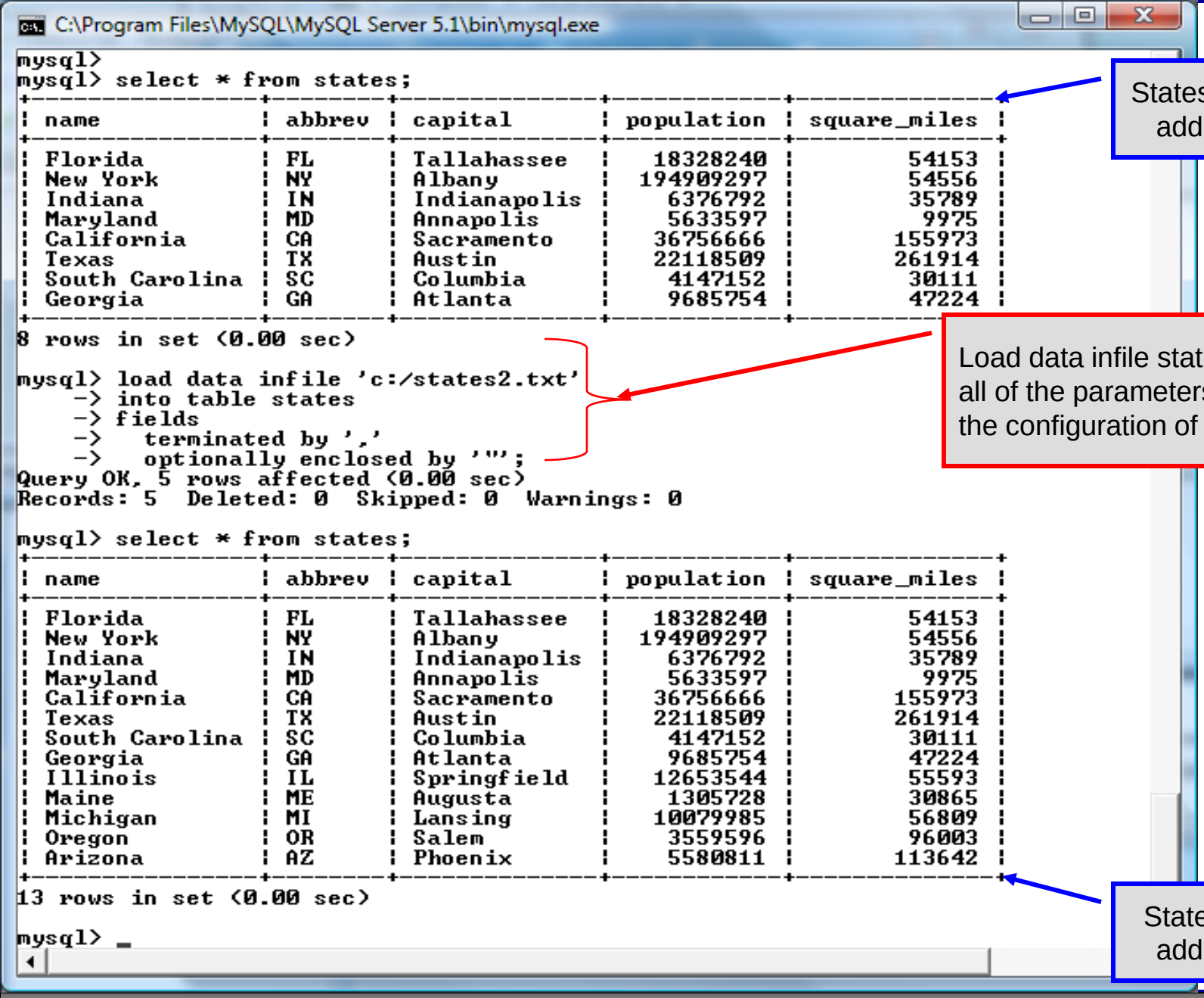
Load Data Infile Example



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

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

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

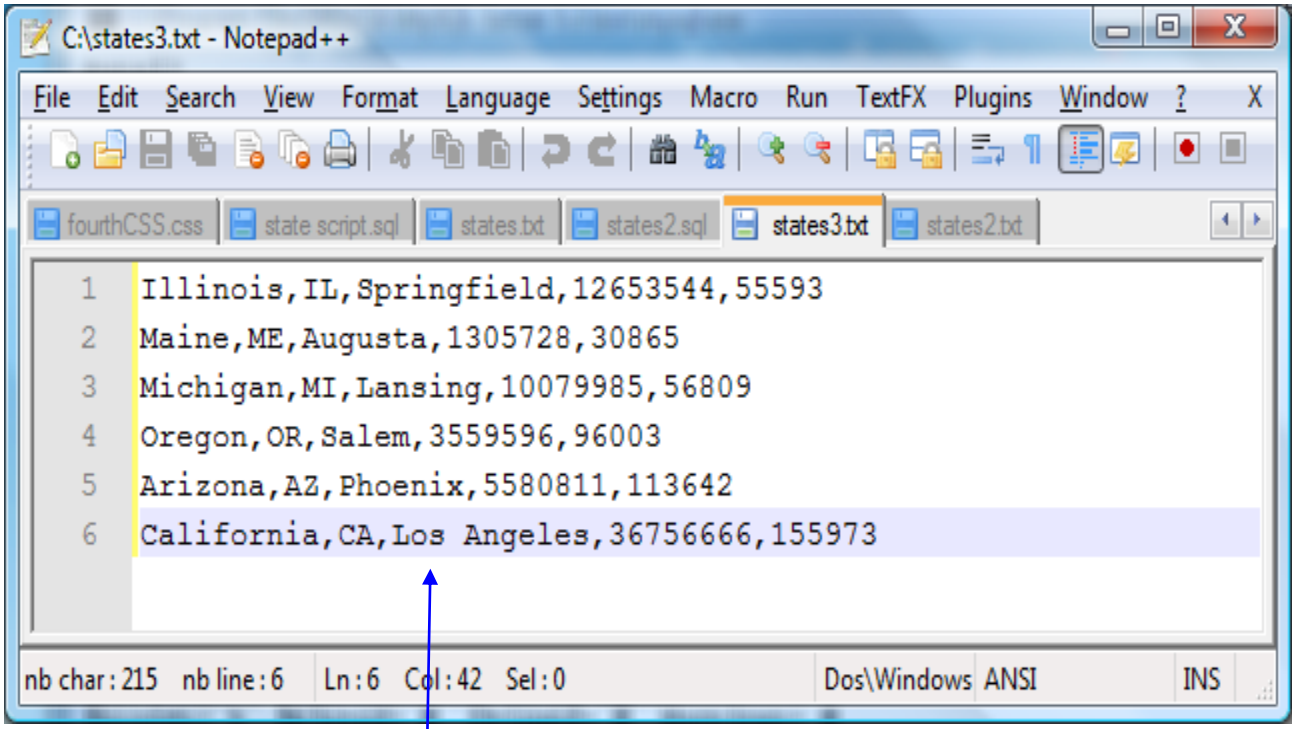


States table **before** addition of data

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



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

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

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

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

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

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

mysql> _
```

States table **before** addition of data

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

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

The `Ignore` Clause of the Insert Command

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

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

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

Low Priority and Delayed Inserts

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

Inserting/Replacing Data Using `Replace`

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

Form 1

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

Form 2

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

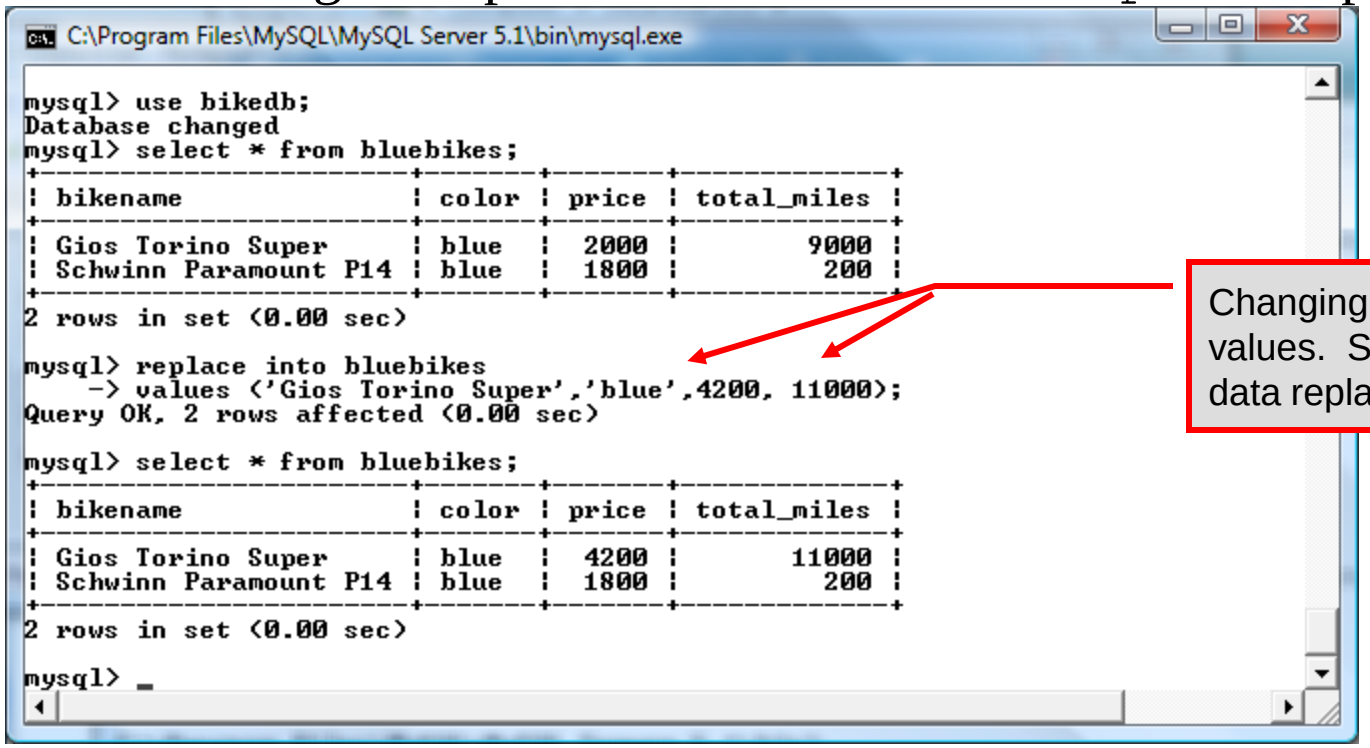
Form 3

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

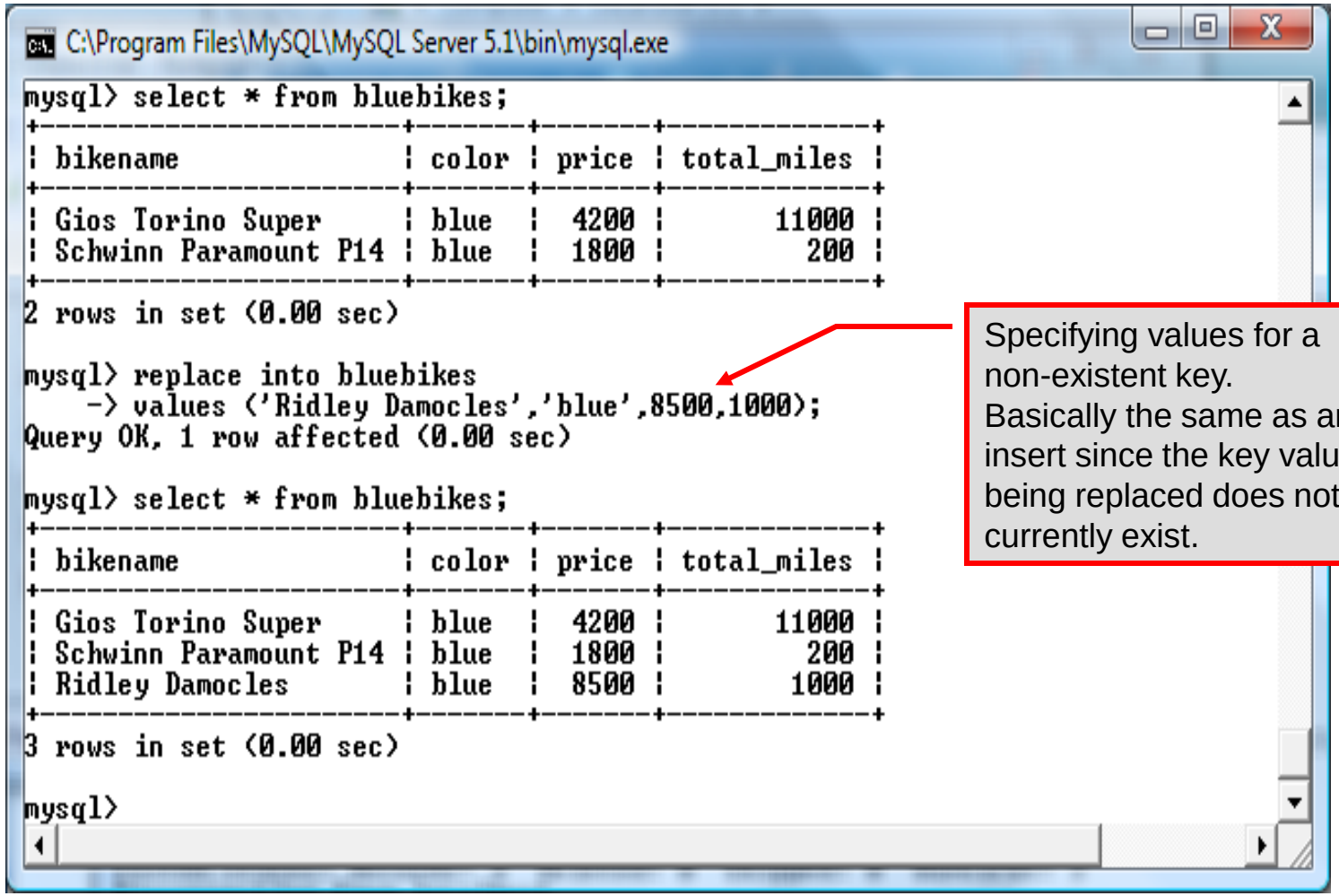

Using `replace`

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

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



Using `Replace` (cont.)



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

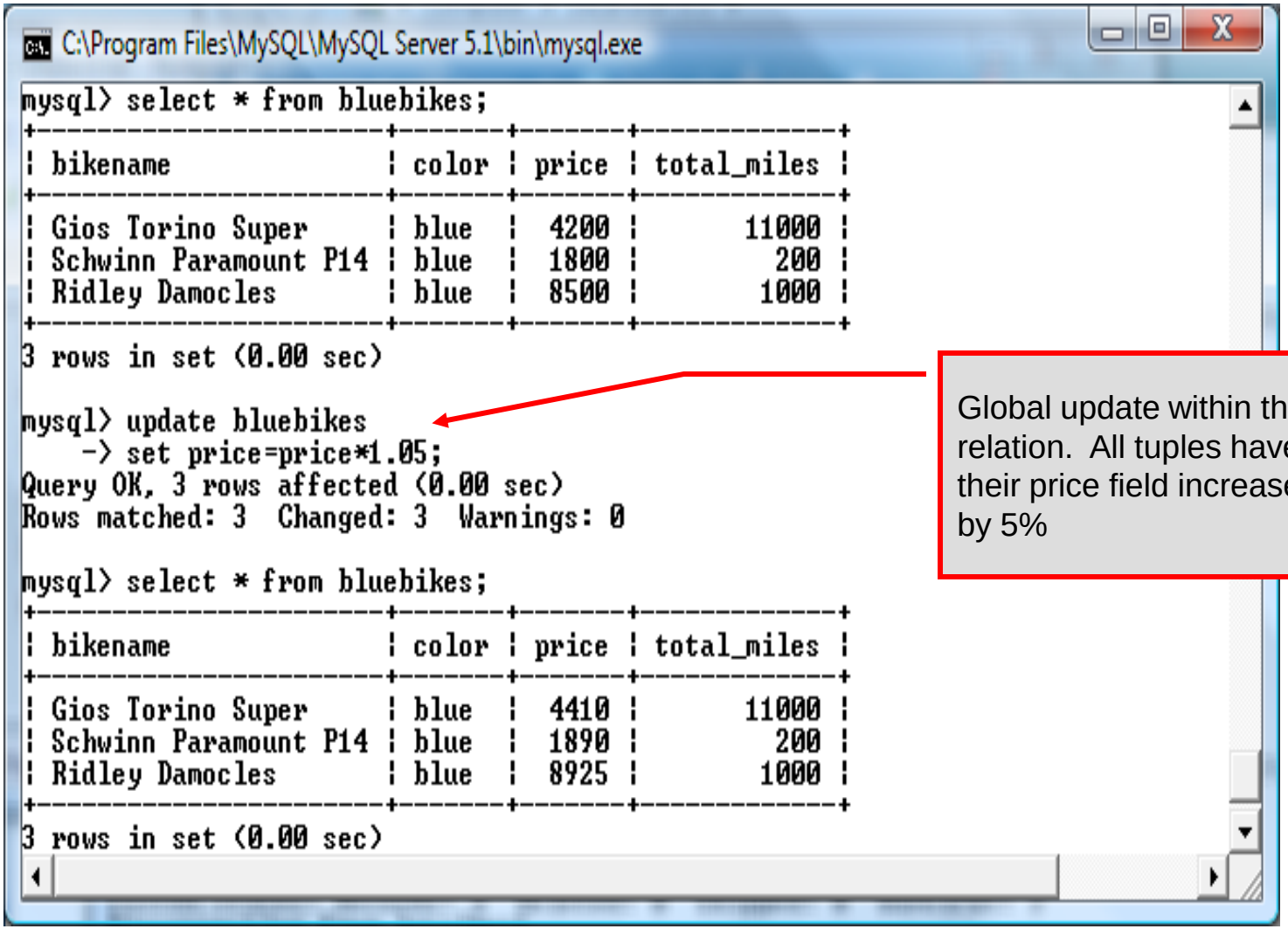
Performing Updates on Tables

- 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];
```

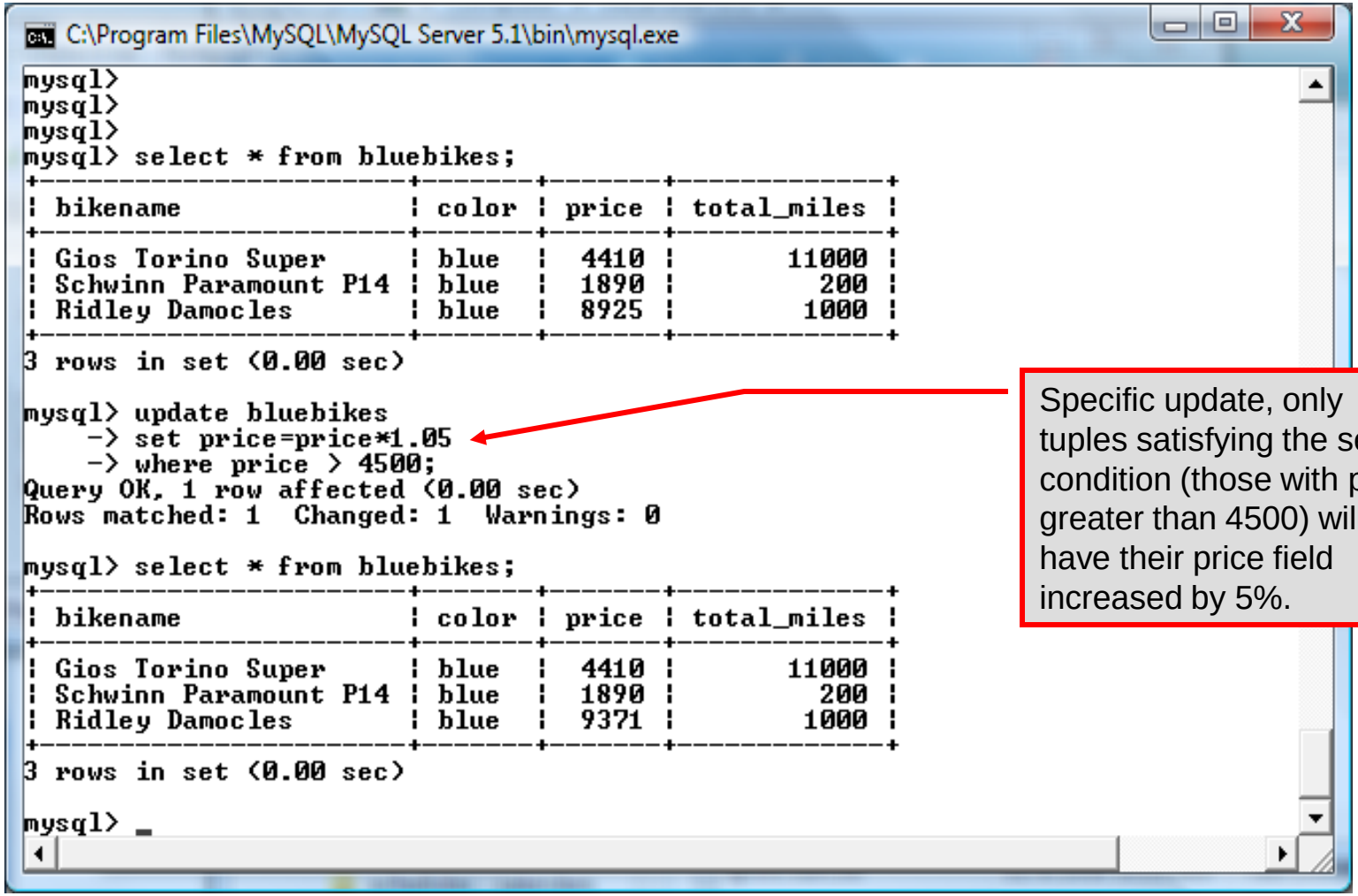
- There are basically two parts to the statement: the `set` portion to declare which column to set to what value; and the `where` portion, which defines which rows are to be affected.
- `Limit` restricts the number of rows affected to `num`.

Using `update` (cont.)



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

Using `update` (cont.)



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

Select Queries in MySQL

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

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


MySQL RDBMS (cont.)

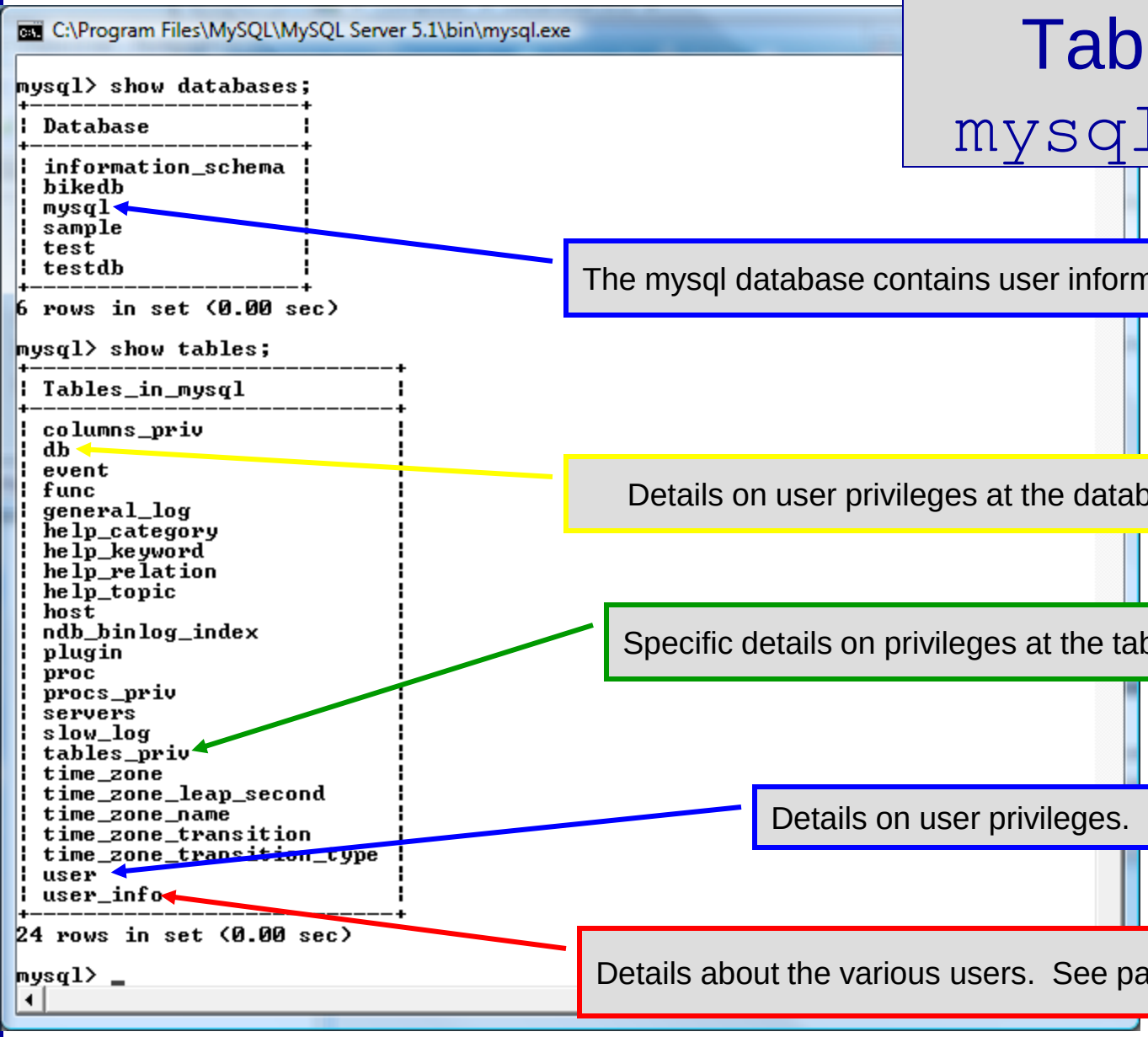
- MySQL features a user permissions system, which allows control over user's access to the databases under MySQL control.
- There are very few competitors of MySQL (Oracle, Sybase, DB2, and SQL Server) that can match the level of sophistication provided by MySQL's permissions system in terms of granularity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

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

Authorization in MySQL

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

Tables in the mysql Database



The mysql database contains user information

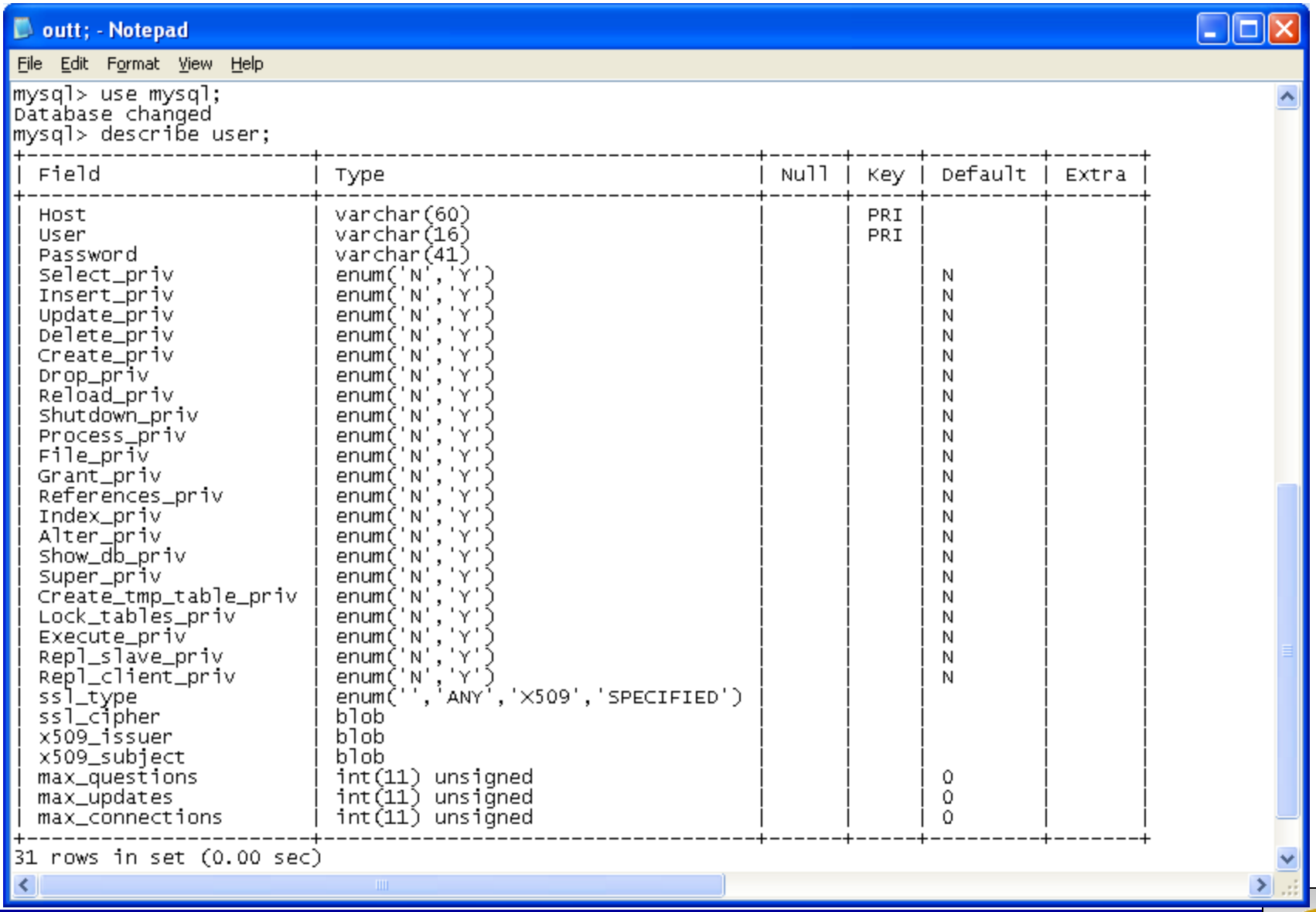
Details on user privileges at the database level. See page 94.

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

Details on user privileges. See page 91.

Details about the various users. See page 92.

Contents of the `user` Table

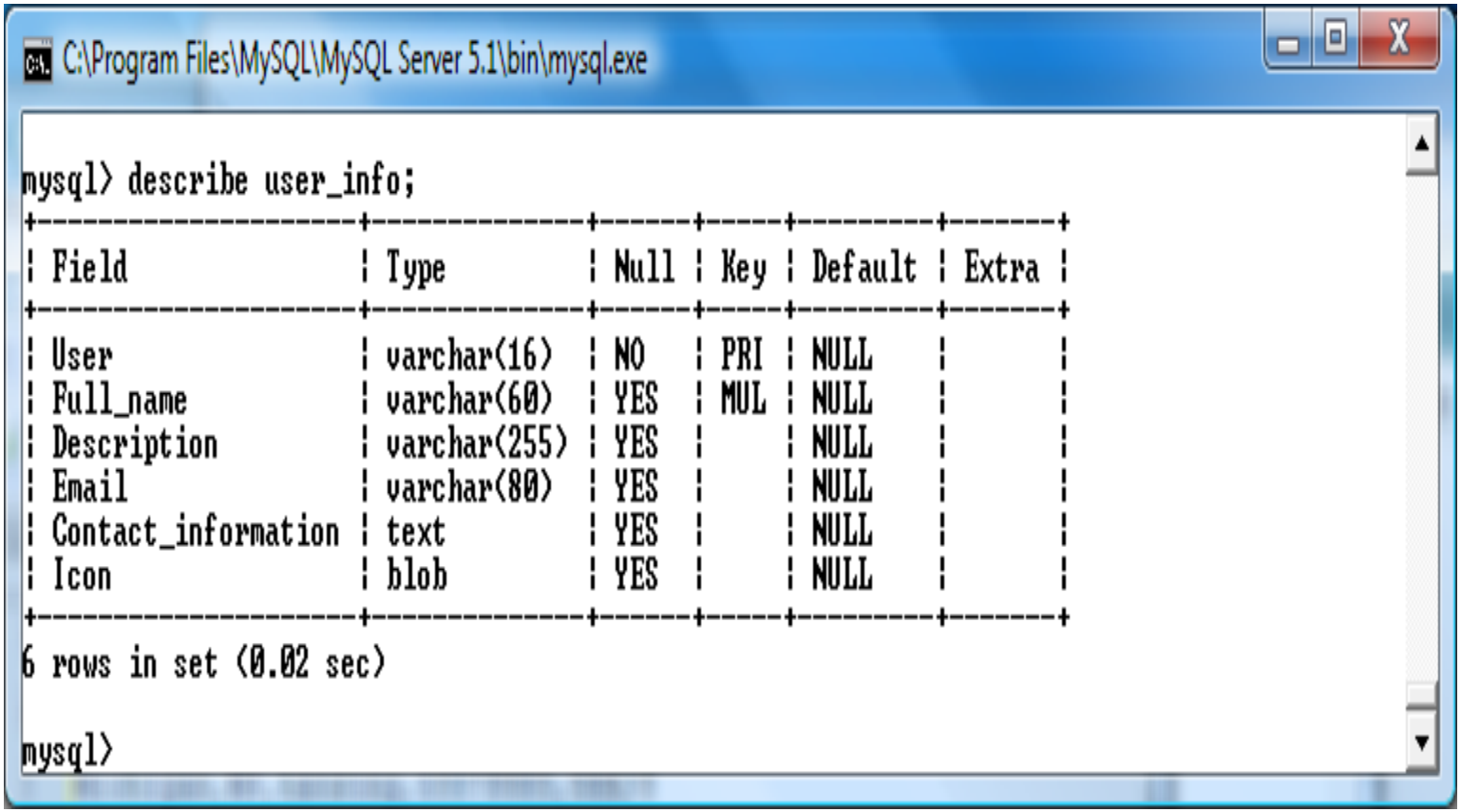


outt; - Notepad

File Edit Format View Help

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


Contents of the `user_info` Table



Contents of the `tables_priv` Table

outt; - Notepad

File Edit Format View Help

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

mysql
```

outt; - Notepad

File Edit Format View Help

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


Contents of the db Table

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

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

mysql>
```


How The Grant Tables Work

- The various grant tables work together to define access capabilities for the various users of the databases in MySQL. The tables represent a hierarchy which begins at the database level and moves downward to finer and finer granularity in access capabilities.
- To understand how the grant tables work, it is necessary to understand the process that MySQL goes through when considering a request from a client.

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

How The Grant Tables Work (cont.)

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

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

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

How The Grant Tables Work (cont.)

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

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

Managing User Privileges with GRANT and REVOKE

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

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


Some of the Privileges Assigned with GRANT

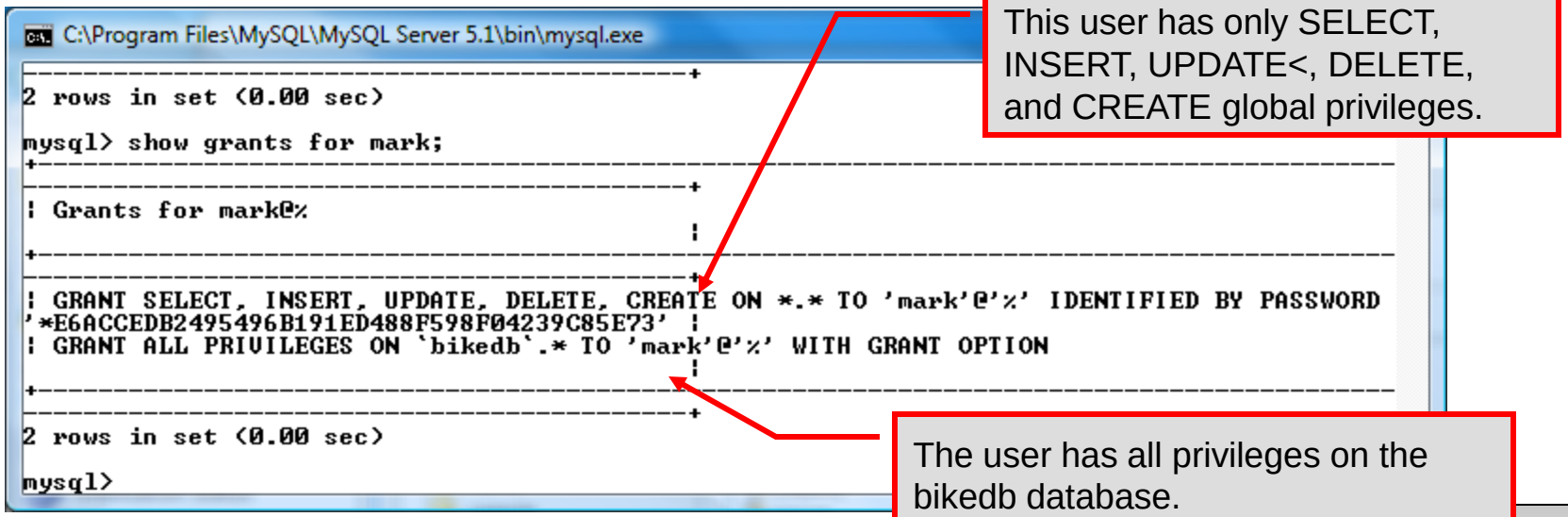
Privilege	Operations Permitted
ALL or ALL PRIVILEGES	All privileges except for GRANT
ALTER	Change a table definition using ALTER TABLE excluding the creation and dropping of indices.
CREATE	Create database or tables within a database.
CREATE TEMPORARY TABLES	Create temporary tables.
DELETE	Ability to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

Displaying Privileges with SHOW

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

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

- An example is shown below:



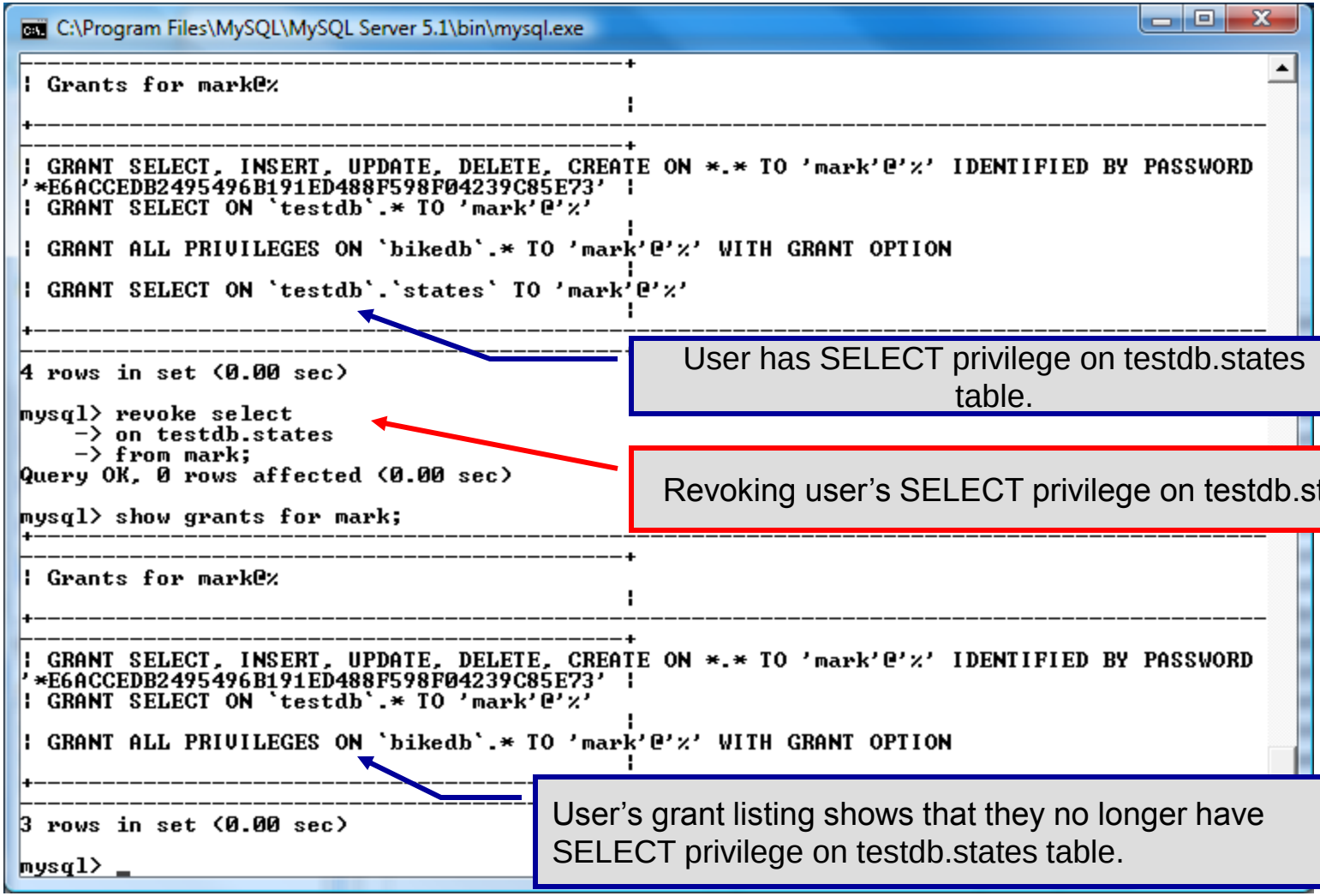
Revoking User Privileges with `REVOKE`

- 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.

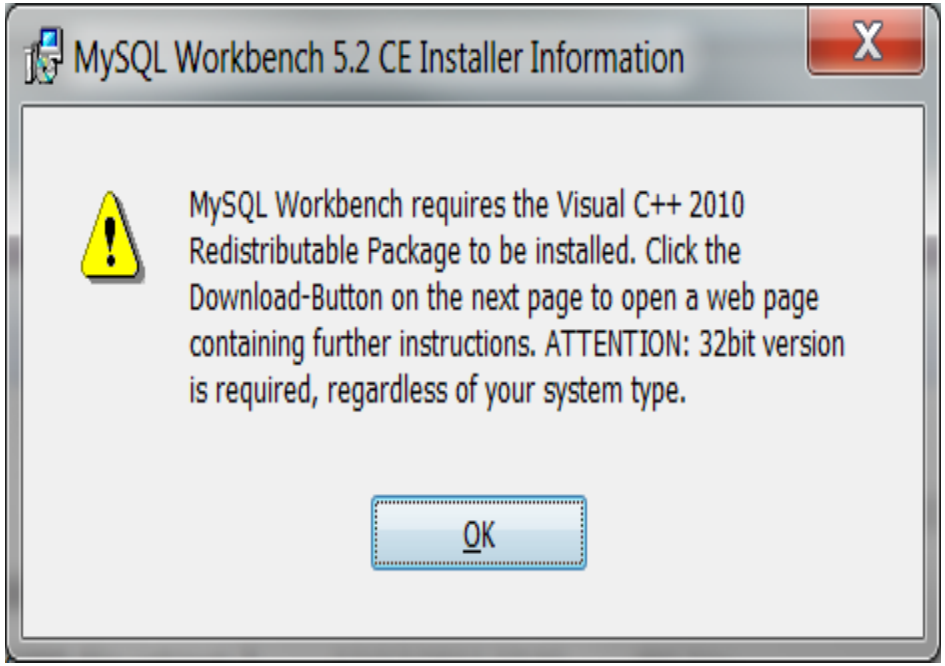
Example - Revoking User Privileges with REVOKE



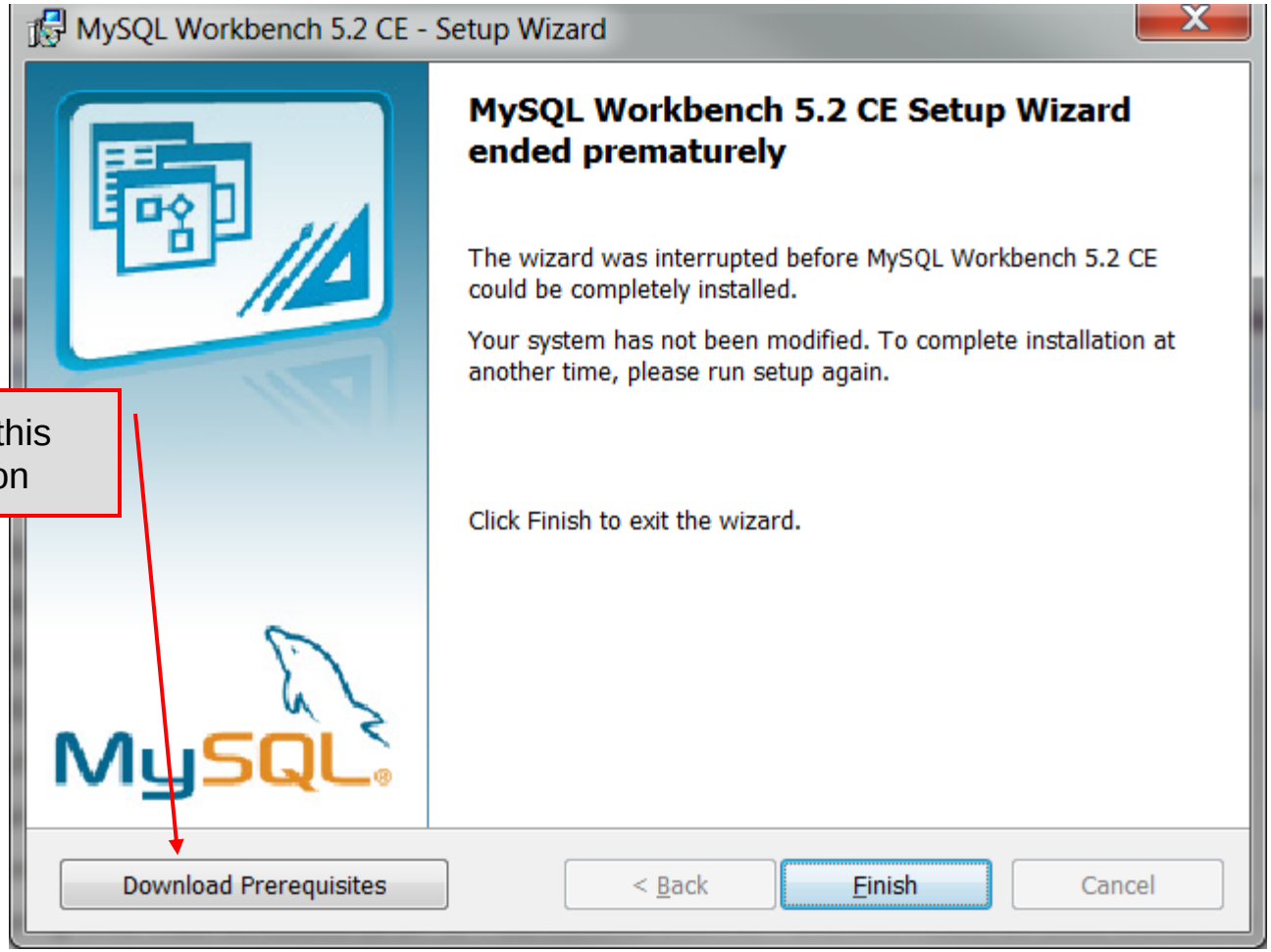
The MySQL Workbench

- From MySQL you can download a GUI-based tool that allows you to create, manipulate, and administer MySQL databases.
- The current version of this tool (5.2.45) does not implement full functionality of the GRANT command down to the attribute level.
- 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/ (see page 7).
- The install/set-up for this tool as well as a few screen shots of this tool and its capabilities are shown in the next few slides.

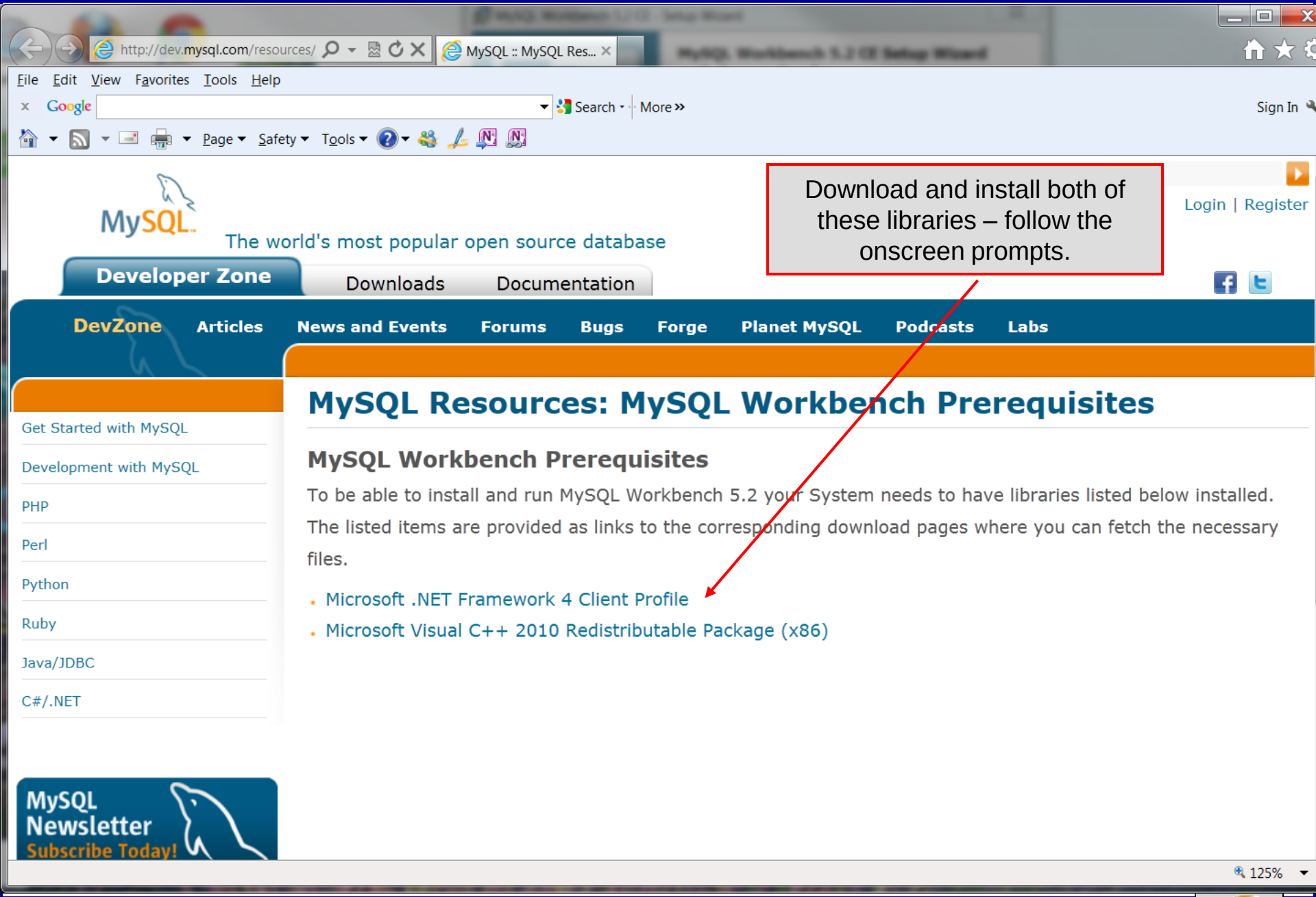
The MySQL Workbench



The MySQL Workbench

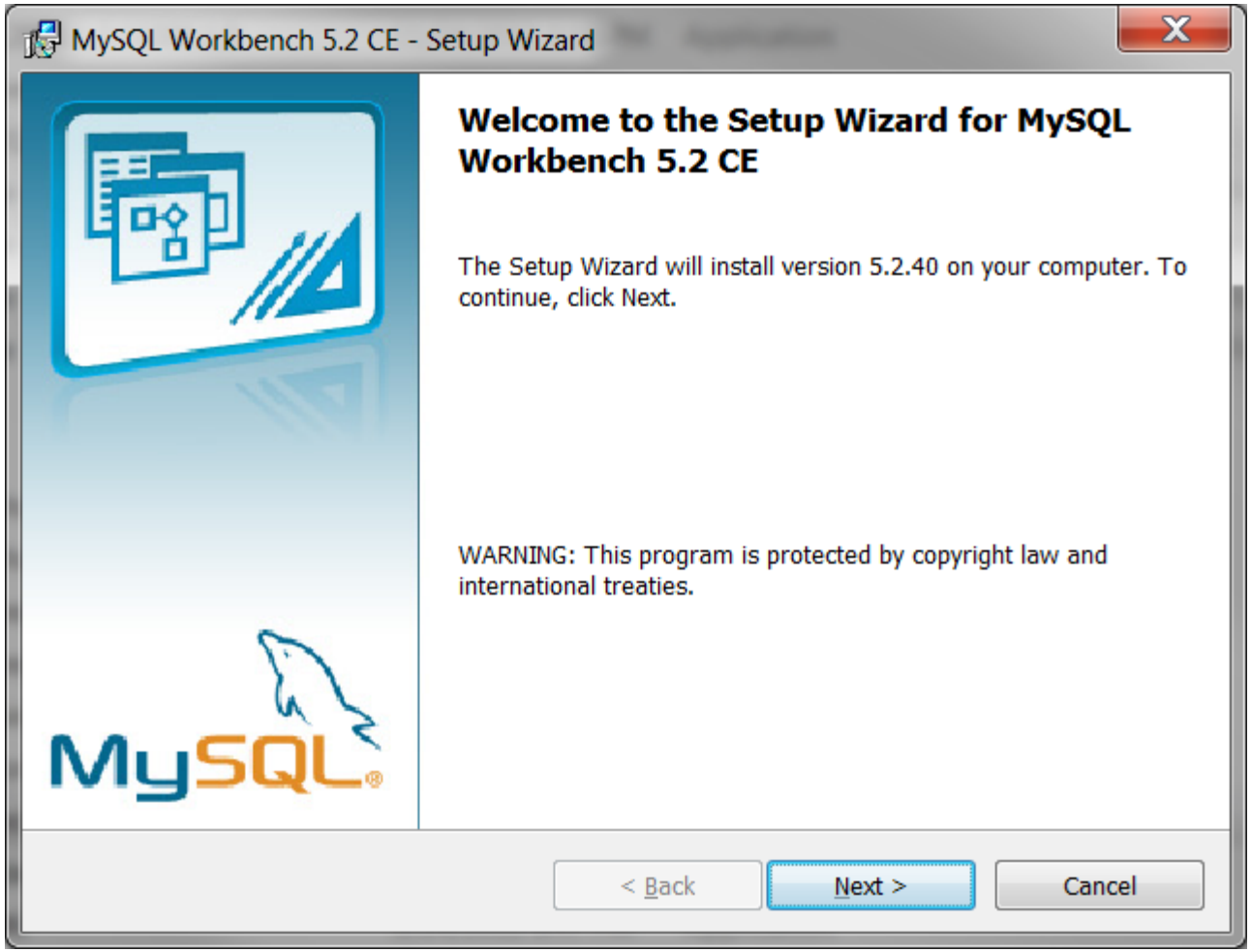


Click this button

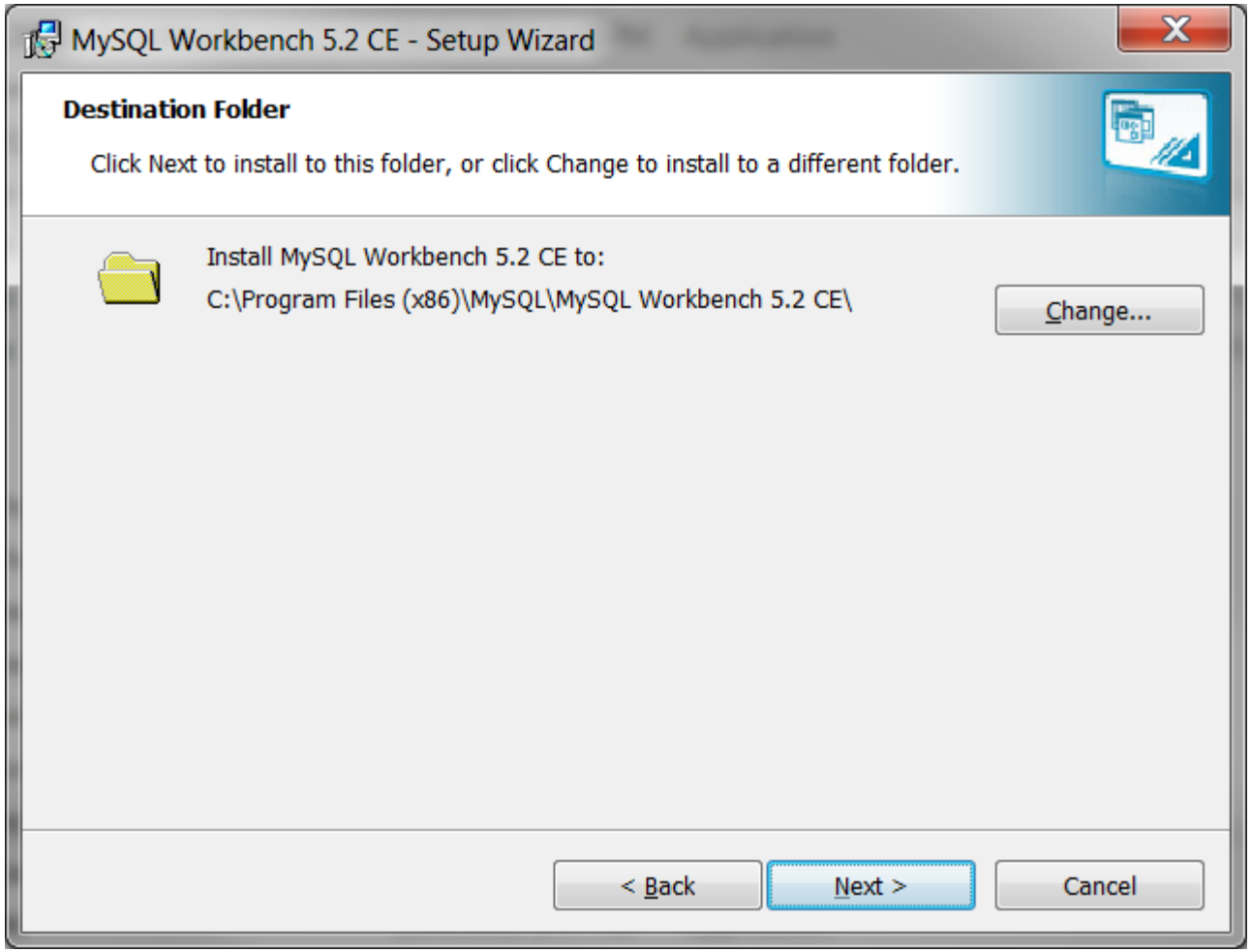


Download and install both of these libraries – follow the onscreen prompts.

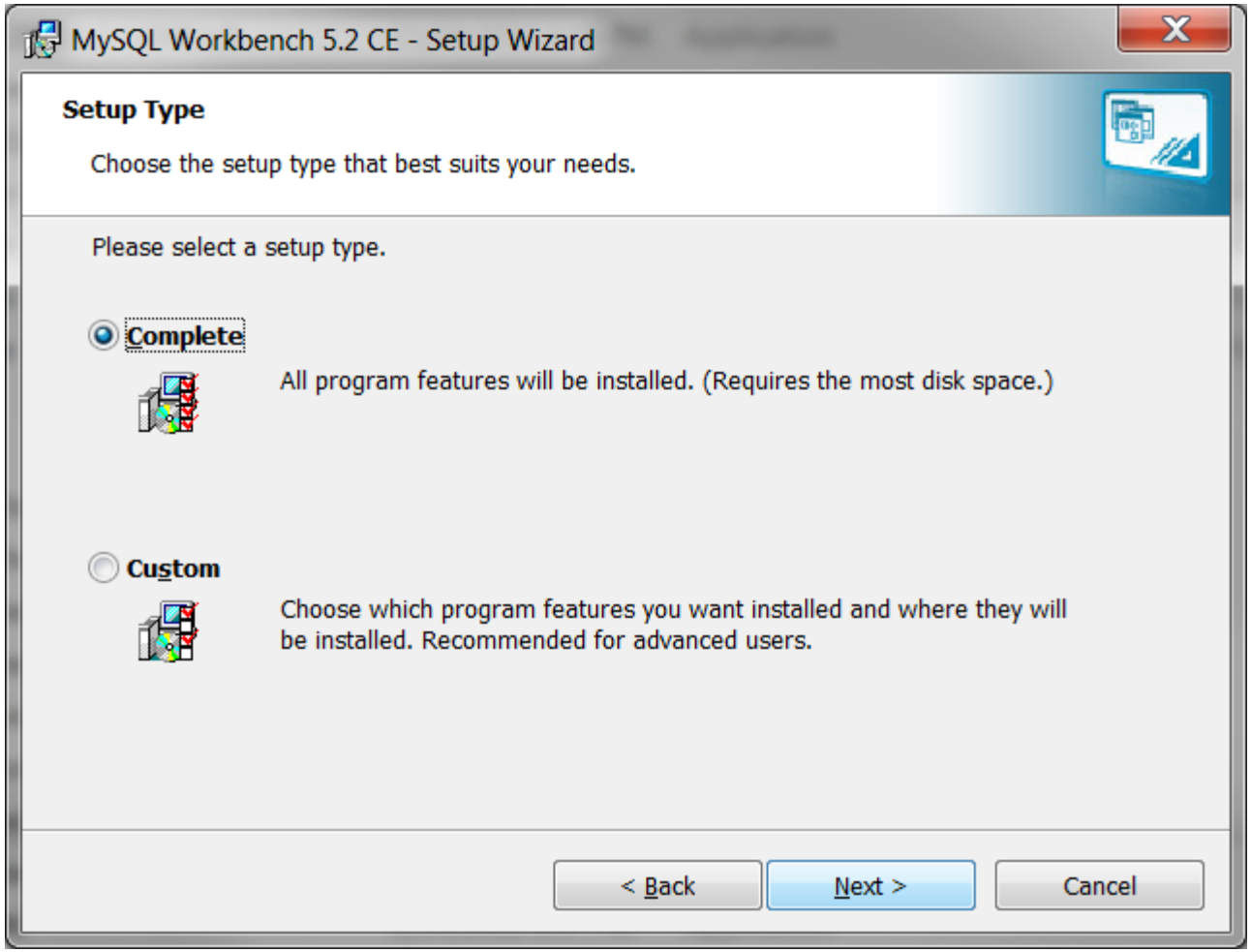
The MySQL Workbench



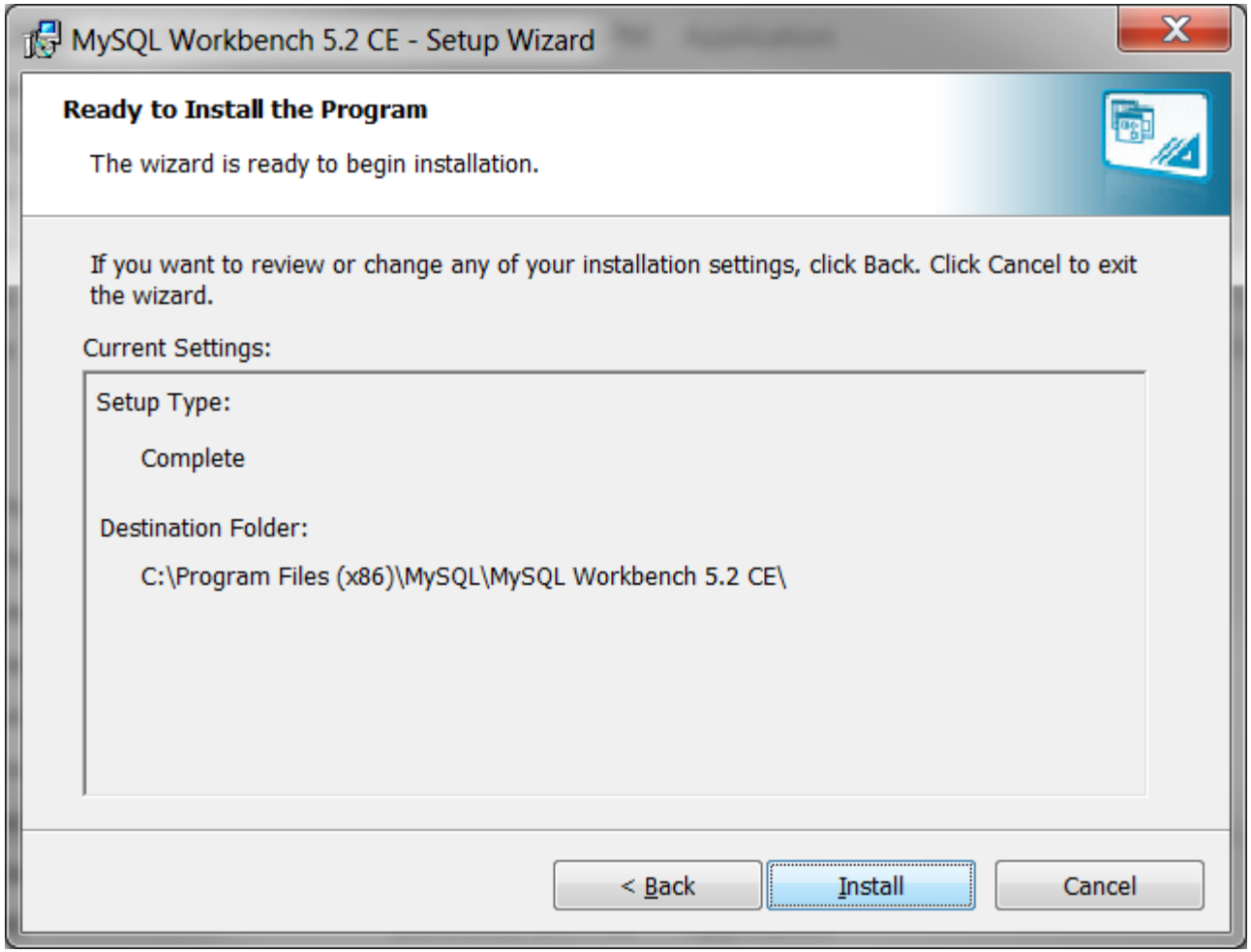
The MySQL Workbench



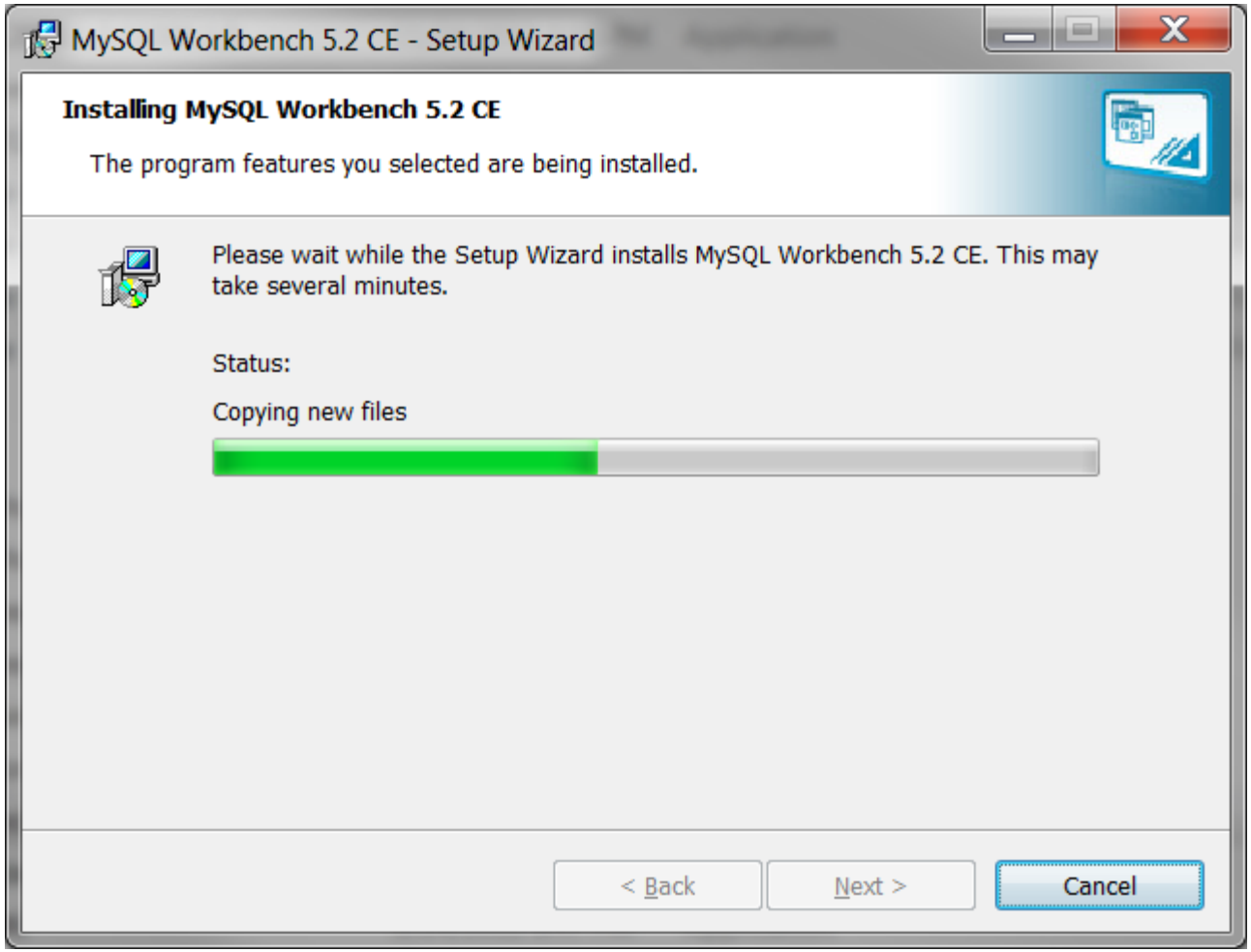
The MySQL Workbench



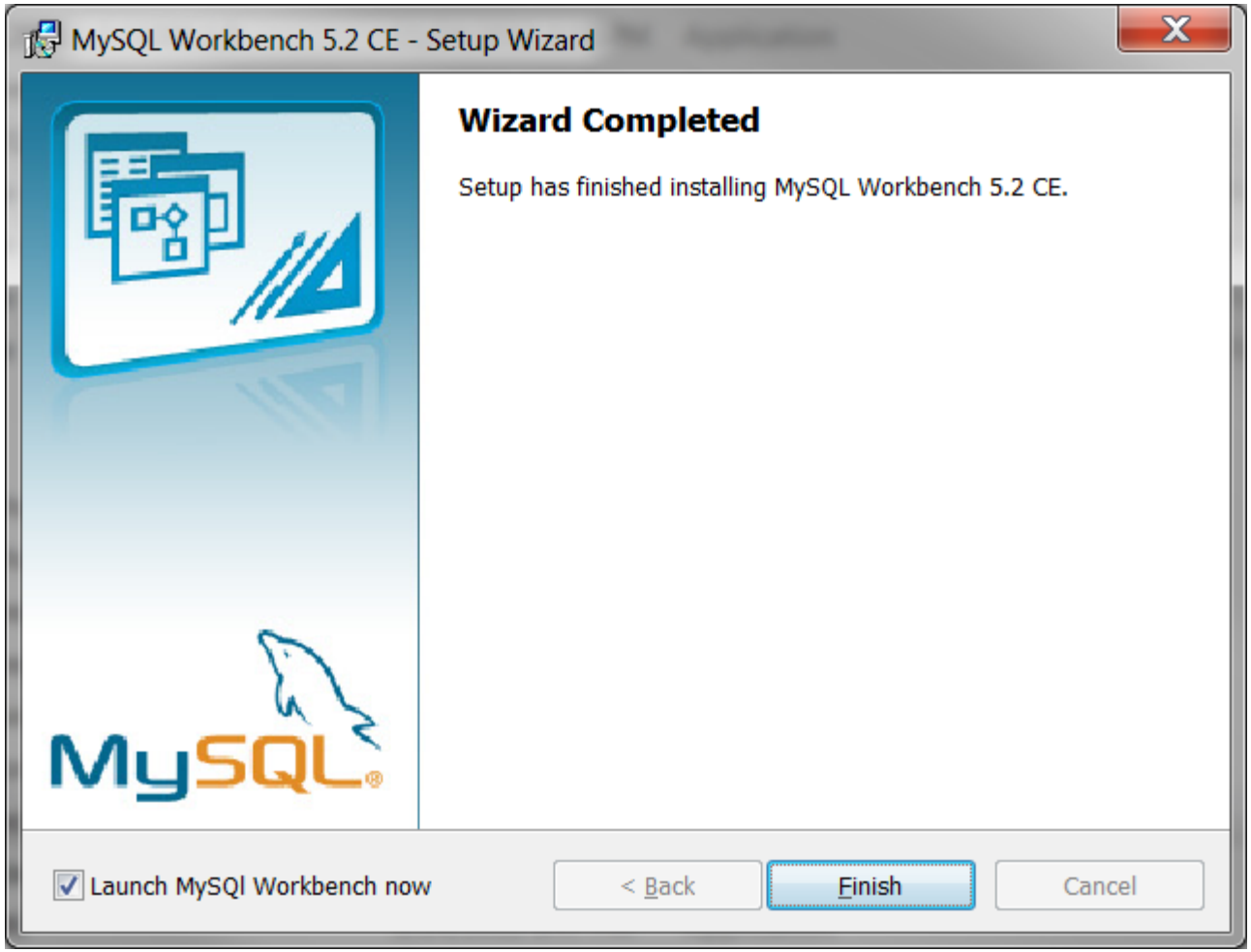
The MySQL Workbench

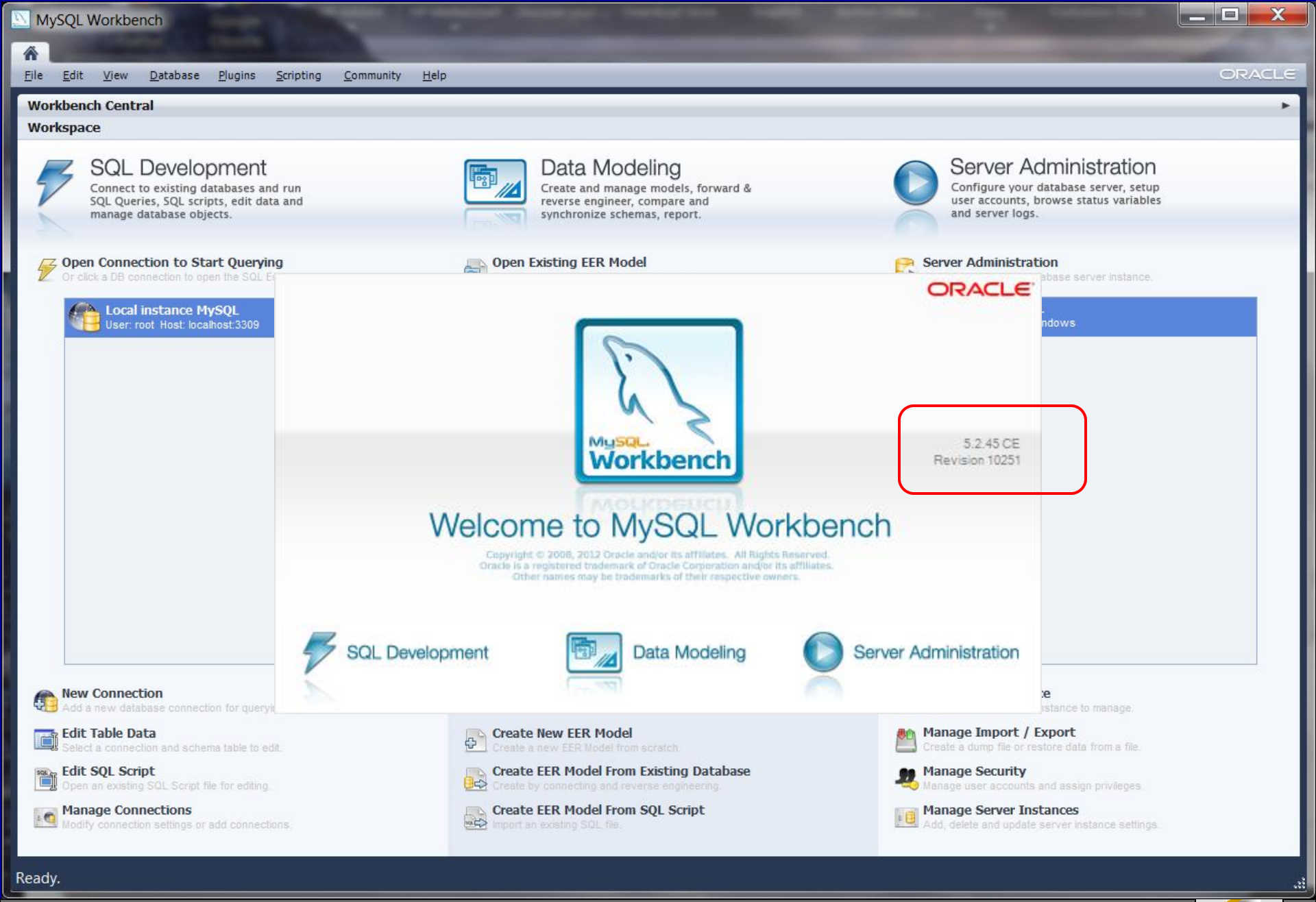


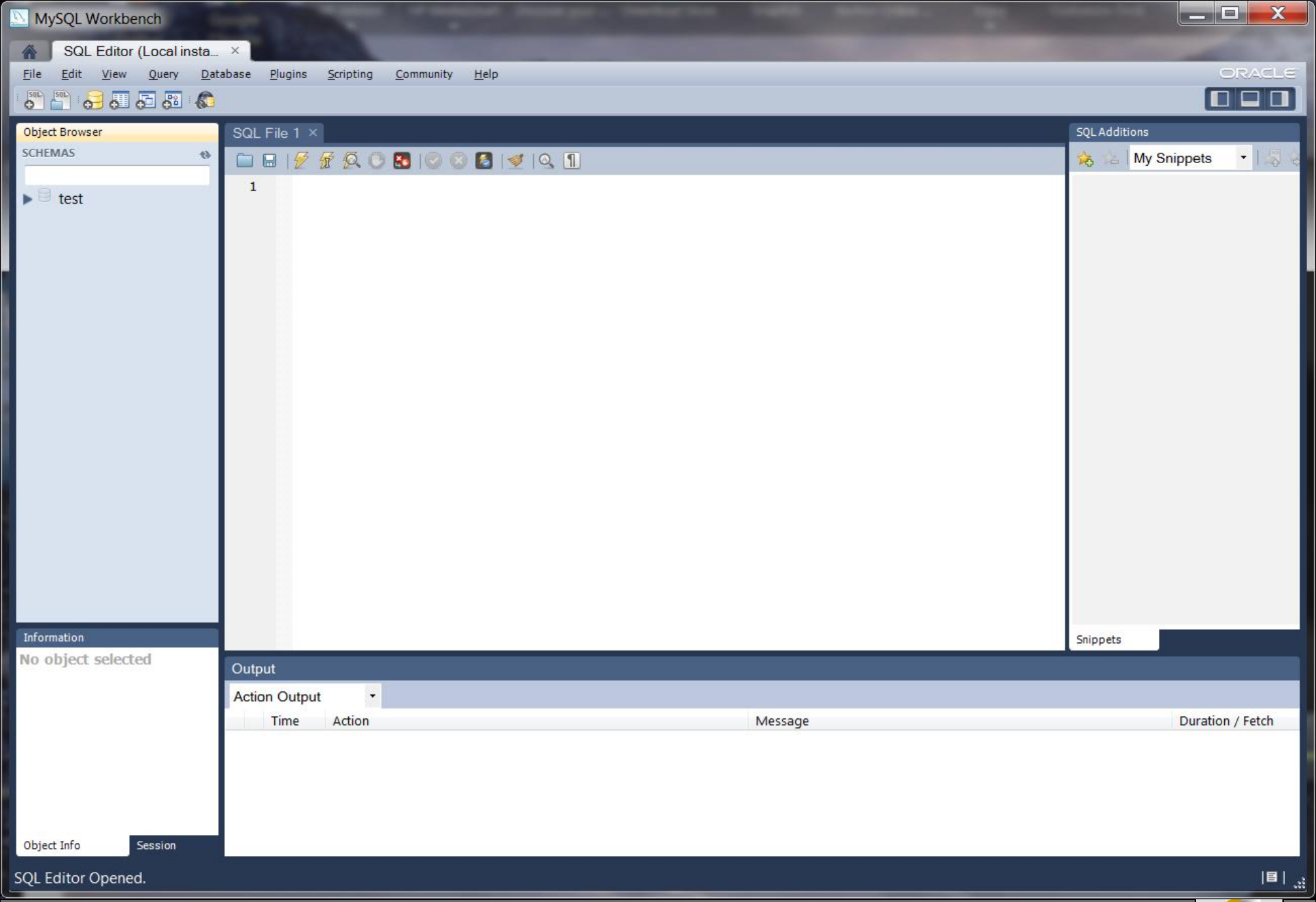
The MySQL Workbench

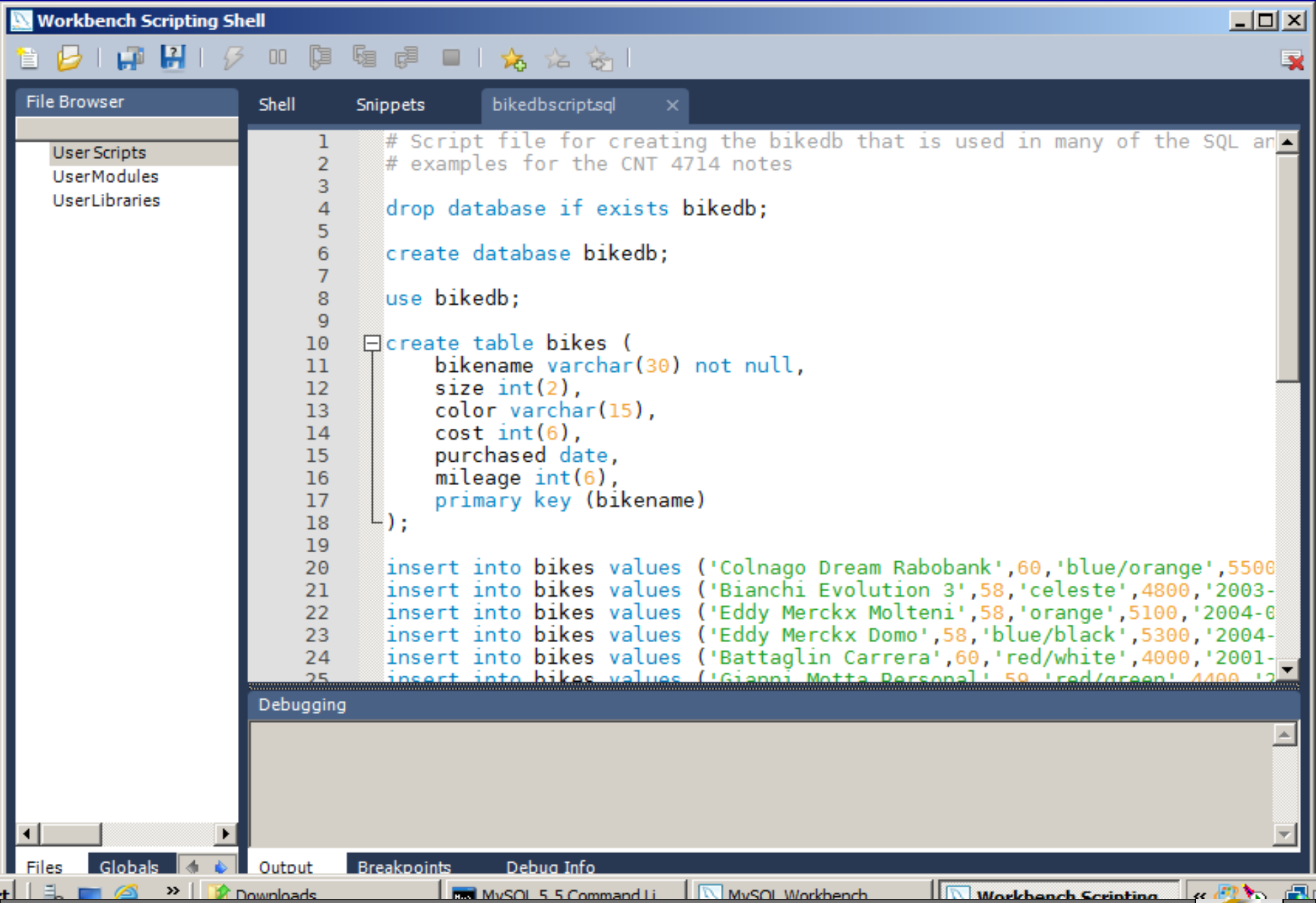


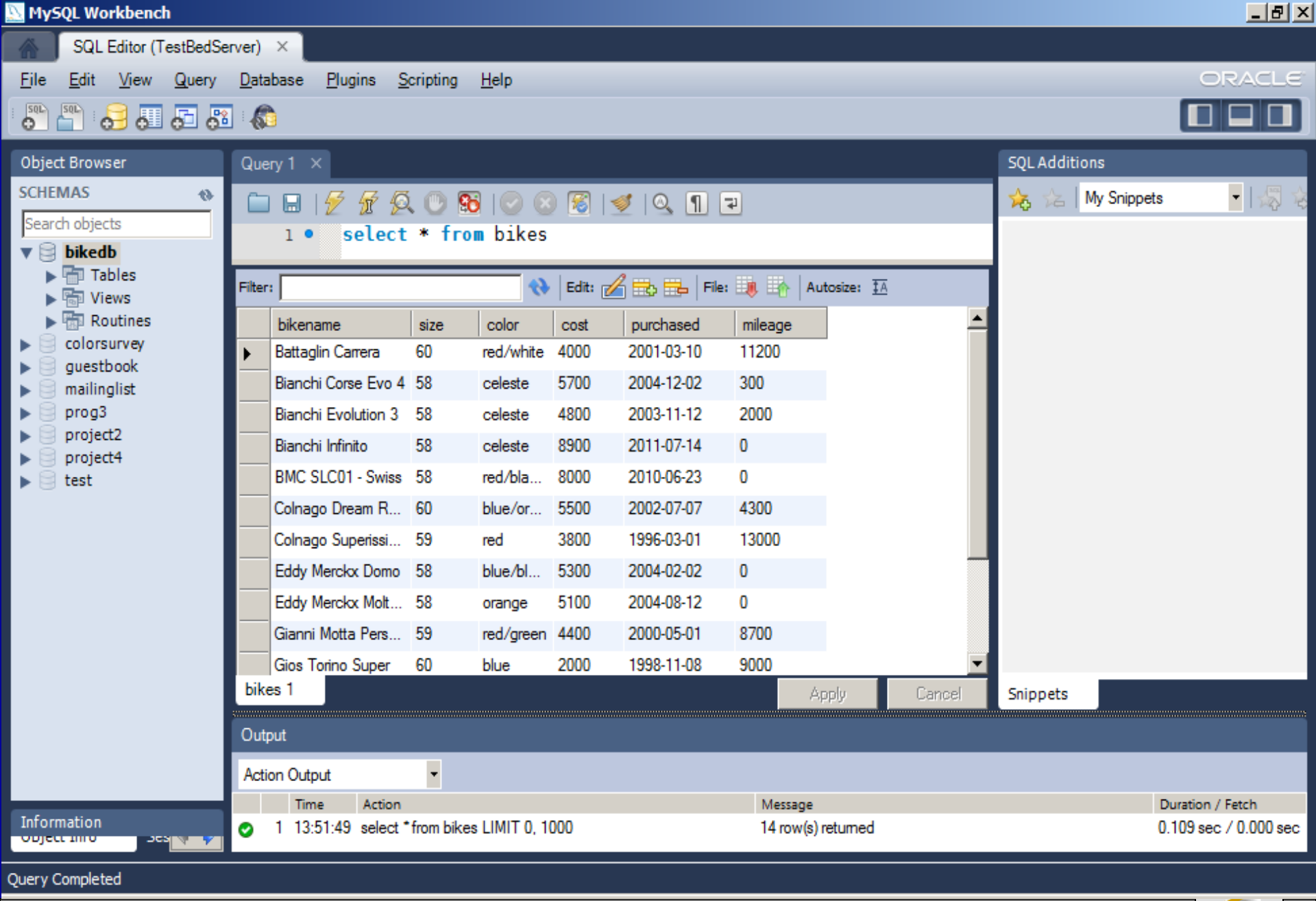
The MySQL Workbench

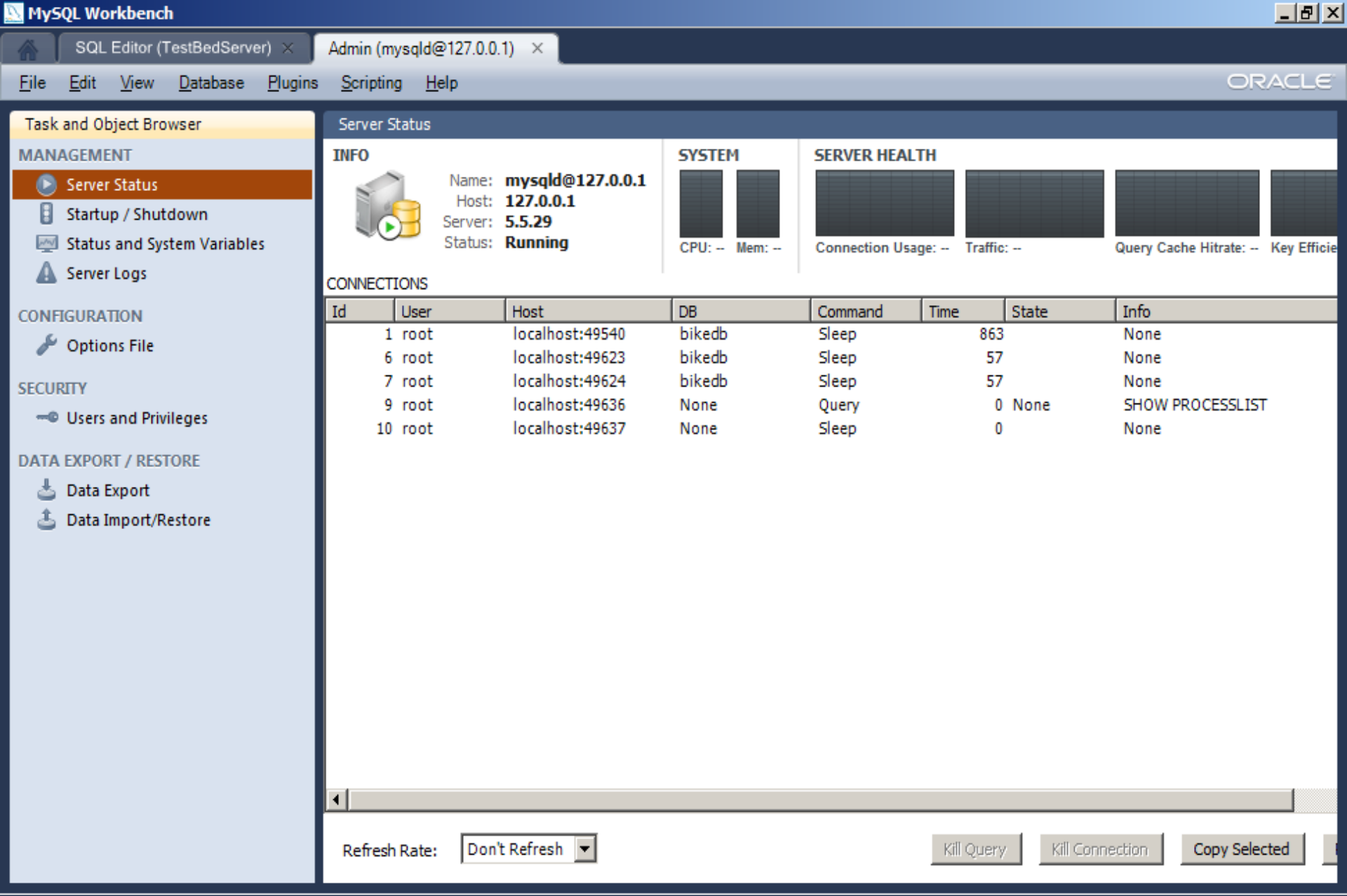


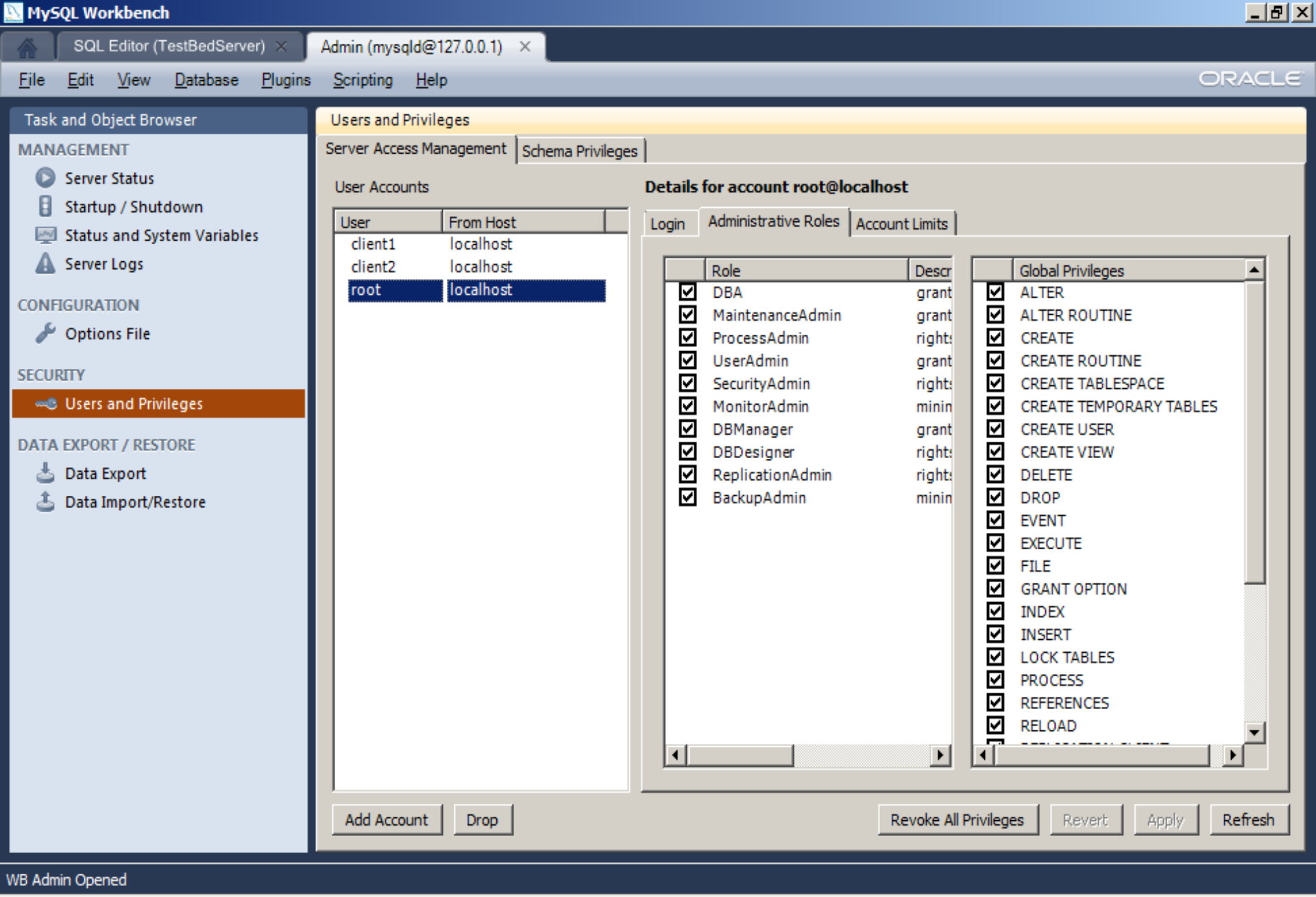

MySQL Workbench
File Edit View Database Plugins Scripting Community Help
ORACLE
Workbench Central
Workspace
SQL Development
Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.
Data Modeling
Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.
Server Administration
Configure your database server, setup user accounts, browse status variables and server logs.
Open Connection to Start Querying
Open Existing EER Model
Server Administration
Local instance MySQL
User: root Host: localhost:3309
ORACLE
MySQL Workbench
5.2.45 CE
Revision 10251
Welcome to MySQL Workbench
Copyright © 2008, 2012 Oracle and/or its affiliates. All Rights Reserved.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.
SQL Development
Data Modeling
Server Administration
New Connection
Edit Table Data
Edit SQL Script
Manage Connections
Create New EER Model
Create EER Model From Existing Database
Create EER Model From SQL Script
Manage Import / Export
Manage Security
Manage Server Instances
Ready.


MySQL Workbench
SQL Editor (Local insta...
File Edit View Query Database Plugins Scripting Community Help
ORACLE
Object Browser
SCHEMAS
test
SQL File 1
1
SQL Additions
My Snippets
Snippets
Information
No object selected
Object Info
Session
Output
Action Output
Time
Action
Message
Duration / Fetch
SQL Editor Opened.


Workbench Scripting Shell
File Browser
User Scripts
UserModules
UserLibraries
Shell
Snippets
bikedbscriptsql
1 # Script file for creating the bikedb that is used in many of the SQL an
2 # examples for the CNT 4714 notes
3
4 drop database if exists bikedb;
5
6 create database bikedb;
7
8 use bikedb;
9
10 create table bikes (
11 bikename varchar(30) not null,
12 size int(2),
13 color varchar(15),
14 cost int(6),
15 purchased date,
16 mileage int(6),
17 primary key (bikename)
18);
19
20 insert into bikes values ('Colnago Dream Rabobank',60,'blue/orange',5500
21 insert into bikes values ('Bianchi Evolution 3',58,'celeste',4800,'2003-
22 insert into bikes values ('Eddy Merckx Molteni',58,'orange',5100,'2004-0
23 insert into bikes values ('Eddy Merckx Domo',58,'blue/black',5300,'2004-
24 insert into bikes values ('Battaglin Carrera',60,'red/white',4000,'2001-
Debugging
Files
Globals
Output
Breakpoints
Debug Info
Downloads
MySQL 5.5 Command Li
MySQL Workbench
Workbench Scripting


MySQL Workbench
SQL Editor (TestBedServer)
File Edit View Query Database Plugins Scripting Help
ORACLE
Object Browser
SCHEMAS
Search objects
bikedb
Tables
Views
Routines
colorsurvey
guestbook
mailinglist
prog3
project2
project4
test
Query 1
1 • select * from bikes
Filter:
Edit:
File:
Autosize:
bikename size color cost purchased mileage
Battaglin Carrera 60 red/white 4000 2001-03-10 11200
Bianchi Corse Evo 4 58 celeste 5700 2004-12-02 300
Bianchi Evolution 3 58 celeste 4800 2003-11-12 2000
Bianchi Infinito 58 celeste 8900 2011-07-14 0
BMC SLC01 - Swiss 58 red/bla... 8000 2010-06-23 0
Colnago Dream R... 60 blue/or... 5500 2002-07-07 4300
Colnago Superissi... 59 red 3800 1996-03-01 13000
Eddy Merckx Domo 58 blue/bl... 5300 2004-02-02 0
Eddy Merckx Molt... 58 orange 5100 2004-08-12 0
Gianni Motta Pers... 59 red/green 4400 2000-05-01 8700
Gios Torino Super 60 blue 2000 1998-11-08 9000
bikes 1
Apply
Cancel
SQL Additions
My Snippets
Snippets
Output
Action Output
Time Action Message Duration / Fetch
1 13:51:49 select * from bikes LIMIT 0, 1000 14 row(s) returned 0.109 sec / 0.000 sec
Information
Query Completed


MySQL Workbench
SQL Editor (TestBedServer)
Admin (mysqld@127.0.0.1)
File Edit View Database Plugins Scripting Help
ORACLE
Task and Object Browser
MANAGEMENT
Server Status
Startup / Shutdown
Status and System Variables
Server Logs
CONFIGURATION
Options File
SECURITY
Users and Privileges
DATA EXPORT / RESTORE
Data Export
Data Import/Restore
Server Status
INFO
Name: mysqld@127.0.0.1
Host: 127.0.0.1
Server: 5.5.29
Status: Running
SYSTEM
CPU: -- Mem: --
SERVER HEALTH
Connection Usage: -- Traffic: -- Query Cache Hitrate: -- Key Efficie
CONNECTIONS
Id User Host DB Command Time State Info
1 root localhost:49540 bikedb Sleep 863 None
6 root localhost:49623 bikedb Sleep 57 None
7 root localhost:49624 bikedb Sleep 57 None
9 root localhost:49636 None Query 0 None SHOW PROCESSLIST
10 root localhost:49637 None Sleep 0 None
Refresh Rate: Don't Refresh
Kill Query
Kill Connection
Copy Selected
WB Admin Opened


MySQL Workbench
SQL Editor (TestBedServer)
Admin (mysqld@127.0.0.1)
File Edit View Database Plugins Scripting Help
ORACLE
Task and Object Browser
MANAGEMENT
Server Status
Startup / Shutdown
Status and System Variables
Server Logs
CONFIGURATION
Options File
SECURITY
Users and Privileges
DATA EXPORT / RESTORE
Data Export
Data Import/Restore
Users and Privileges
Server Access Management
Schema Privileges
User Accounts
User
From Host
client1 localhost
client2 localhost
root localhost
Details for account root@localhost
Login
Administrative Roles
Account Limits
Role
Descr
DBA grant
MaintenanceAdmin grant
ProcessAdmin right
UserAdmin grant
SecurityAdmin right
MonitorAdmin minin
DBManager grant
DBDesigner right
ReplicationAdmin right
BackupAdmin minin
Global Privileges
ALTER
ALTER ROUTINE
CREATE
CREATE ROUTINE
CREATE TABLESPACE
CREATE TEMPORARY TABLES
CREATE USER
CREATE VIEW
DELETE
DROP
EVENT
EXECUTE
FILE
GRANT OPTION
INDEX
INSERT
LOCK TABLES
PROCESS
REFERENCES
RELOAD
Add Account
Drop
Revoke All Privileges
Revert
Apply
Refresh
WB Admin Opened