

COP 4610L: Applications in the Enterprise Spring 2005

Introduction to MySQL– Part 1 –

Instructor : Mark Llewellyn
markl@cs.ucf.edu
CSB 242, 823-2790
<http://www.cs.ucf.edu/courses/cop4610L/spr2005>

School of Electrical Engineering and Computer Science
University of Central Florida



Database Access via JDBC

- The **Java Database Connectivity (JDBC)** interface enables any Java program to send SQL queries to any database, and receive back result tables with the desired data.
- Similar to the basic idea of Java in writing a program that will run on any hardware platform, JDBC enables the development of programs which function with nearly all commercially available DBMSs. Apart from the general popularity of Java, this is the fundamental reason for the widespread acceptance of JDBC.
- In order to guarantee the general database access, JDBC defines a certain core functionality supported by all DBMSs, This common denominator can be implemented by JDBC.
 - This implies that different product characteristics and manufacturer-specific optimizations are ignored by the JDBC standard.



Database Access via JDBC (cont.)

- One prerequisite for the use of JDBC is the availability of a **JDBC driver** for the database being utilized.
- The JDBC driver translates the JDBC queries of the Java database client into the respective supplier-specific calls.
- The simplest version on the Windows platform is the **Open Database Connectivity (ODBC)** interface. ODBC also enables different databases to function via a uniform interface.
- JDBC and ODBC are both based on the same idea. Using the JDBC-ODBC bridge, it is possible to access an ODBC data source via JDBC.

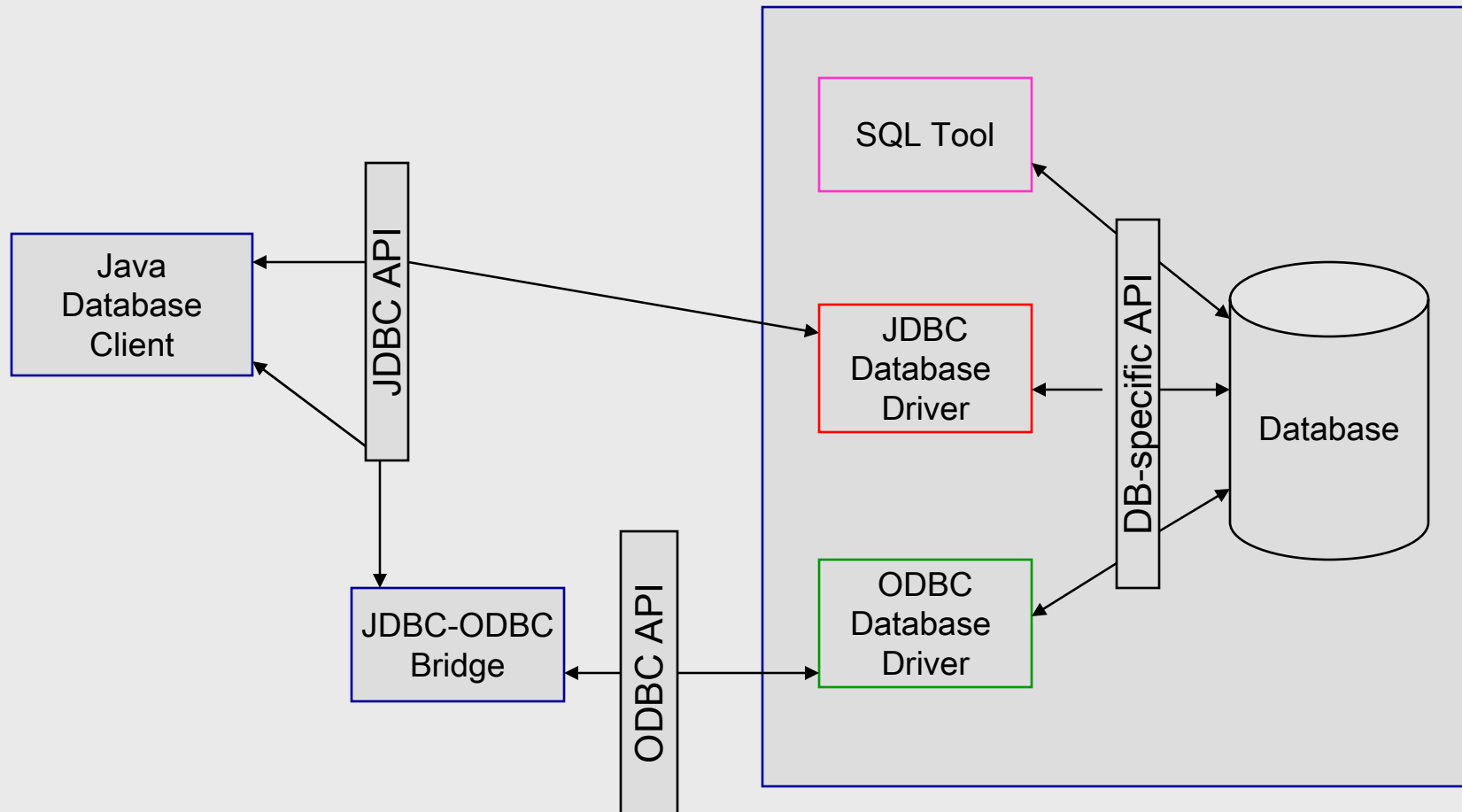


Different Methods for Database Access

- The figure on the next slide summarizes the various options available for accessing a database.
- The client software typically communicates with the server via a proprietary interface. The drivers translate JDBC or ODBC commands into the respective database specific calls.
- The user can also access the database using an SQL specific tool.
- One disadvantage of the ODBC solution is that every computer on which a Java database application is to run, the ODBC connection must be configured. This contradicts the Java principle “write once, run anywhere”. The way around this is through the use of servlets in which the only computer on which the ODBC must be configured is the one on which the servlet engine will run. If a database application is installed on several computers or distributed as an applet, the JDBC-ODBC bridge is not an option as the ODBC connection would have to be configured on every computer.



Summary of Database Access Methods



JDBC Database Access

- For the time being we will focus on the JDBC API for database access and not concern ourselves with ODBC (we'll look in more detail at ODBC later).
- JDBC is almost always used with a RDBMS. However, it can be used with any table-based data source. This means that it also works with applications like Excel.
- The separation of the JDBC API from the particular database drivers enables the application developer to change the underlying database without modifying the Java code that accesses the database.
- Most commercially available RDBMSs provide JDBC drivers and there are many third-party JDBC drivers available.
- We will focus on the JDBC and use it to manipulate a MySQL database. We'll discuss JDBC in more detail later.



MySQL RDBMS

- MySQL is a **database server** (although it does come with a set of simple client programs). The current version is 4.1.9 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 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.



Starting MySQL Server

- On Windows, MySQL runs as a server program, which means that it is a background process that sits patiently waiting for client connections.
- To start the MySQL server, open a command window, switch to the bin directory of your MySQL directory and enter `mysqld --console`. (Omitting `--console` sends error messages to `.err` file in data directory.)

```
C:\> Command Prompt (2) - mysqld --console
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

Z:\>c:

C:\>cd Program Files

C:\Program Files>cd MySQL 4.1.9

C:\Program Files\MySQL 4.1.9>cd MySQL Server 4.1

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1>cd bin

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqld --console
050209 12:48:37 InnoDB: Started; log sequence number 0 44044
mysqld: ready for connections.
Version: '4.1.9' socket: '' port: 3306 Source distribution
```

InnoDB is MySQL's ACID compliant storage engine. Server starts.

Specifying `--console` prevents error message from going to `.err` file.

Server finishes its startup sequence.



Starting MySQL Server (cont.)

- Once the MySQL server is started, open another command window in which to run the client process. The command window in which the MySQL server is executing will continue to display error messages, if any, but otherwise will not return a user prompt until the server exits.
- You can stop the MySQL server by executing the following command from the client window:

```
mysqladmin -u root shutdown
```

- The next page illustrates the execution of this command from the client window and its effect on the MySQL server.



Stopping MySQL Server

Client Window

```
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqladmin -u root shutdown -p
Enter password: ****
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
```

A user must have the privilege of stopping the server. The root user has this permission by default.

Server Window

```
C:\Program Files>cd MySQL 4.1.9
C:\Program Files\MySQL 4.1.9>cd MySQL Server 4.1
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1>cd bin
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqld --console
050209 12:48:37 InnoDB: Started; log sequence number 0 44044
mysqld: ready for connections.
Version: '4.1.9'  socket: ''  port: 3306  Source distribution
050209 13:24:22 [Note] mysqld: Normal shutdown
050209 13:24:22 InnoDB: Starting shutdown...
050209 13:24:25 InnoDB: Shutdown completed; log sequence number 0 4405
050209 13:24:25 [Note] mysqld: Shutdown complete
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
```

MySQL server responds with normal shutdown sequence and responses.

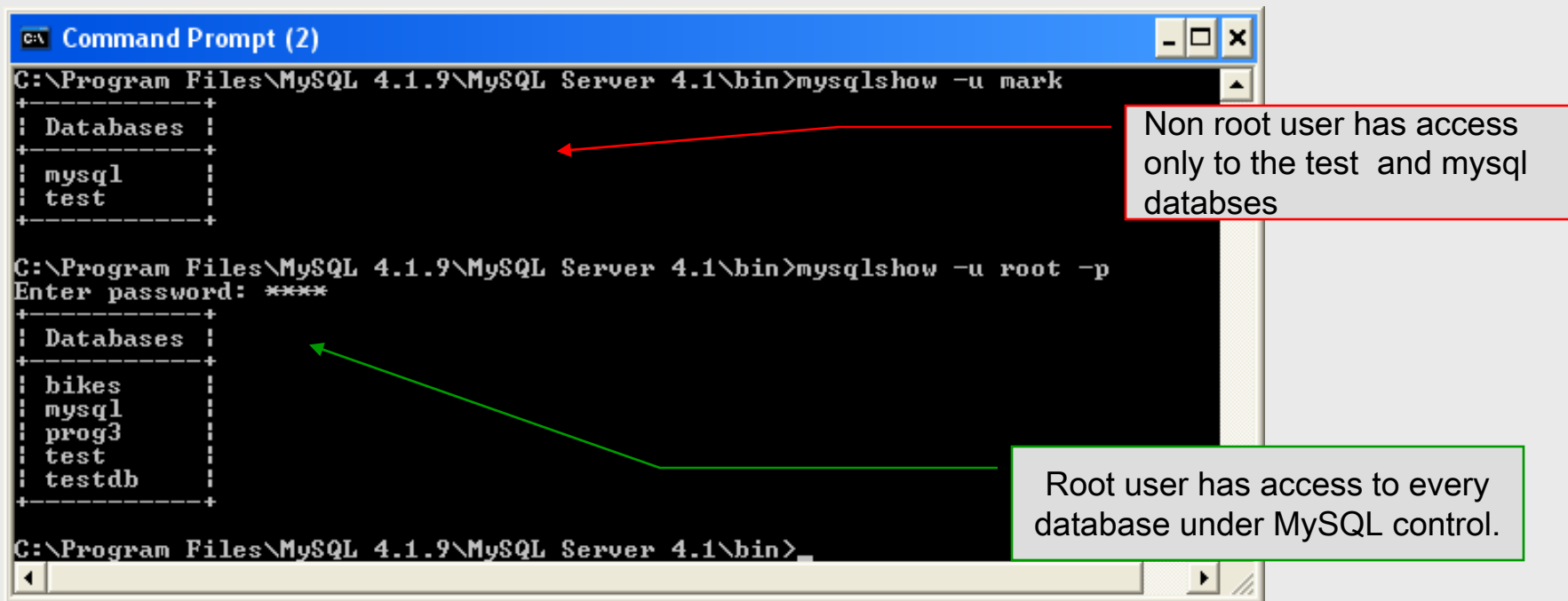


Running MySQL Client Programs Under Windows

- You can test whether the MySQL server is working by executing any of the following commands:
 - `C:\mysql\bin\mysqlshow`
 - `C:\mysql\bin\mysqlshow -u root`
 - `C:\mysql\bin\mysqlshow -u root mysql`
 - `C:\mysql\bin\mysqladmin version status proc`
 - `C:\mysql\bin\mysql test`
- An example of the first two formats is shown on the next slide.



Running MySQL Client Programs Under Windows (cont.)



```
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqlshow -u mark
+-----+
| Databases |
+-----+
| mysql    |
| test     |
+-----+

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqlshow -u root -p
Enter password: ****
+-----+
| Databases |
+-----+
| bikes    |
| mysql    |
| prog3    |
| test     |
| testdb   |
+-----+

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
```

Non root user has access only to the test and mysql databases

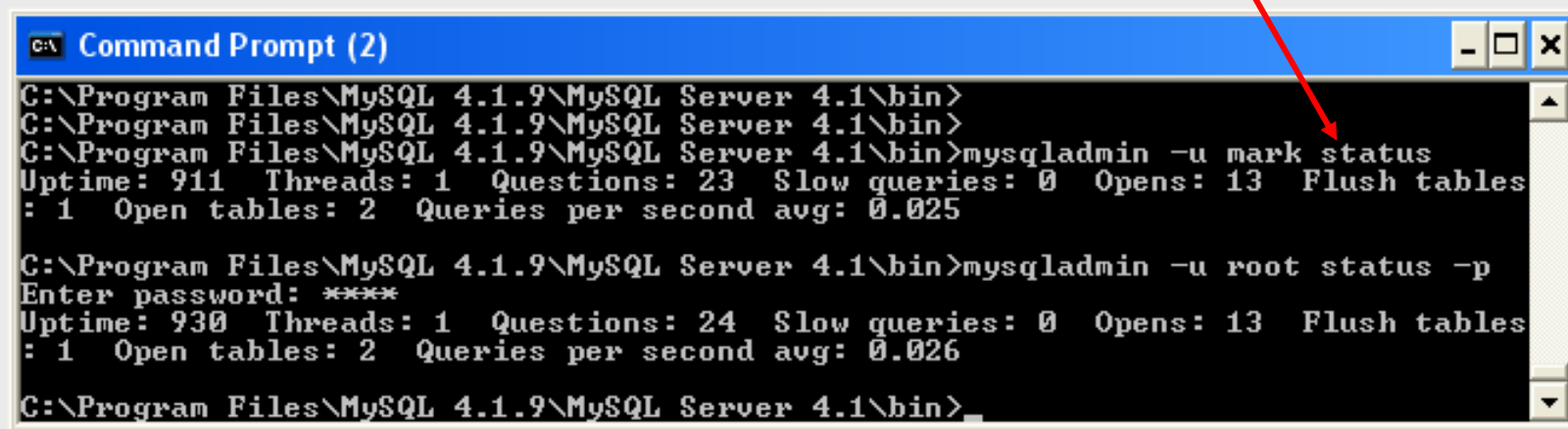
Root user has access to every database under MySQL control.

Note: The MySQL root user is not necessarily the same as the root user in Linux/Unix systems (it might be, but it doesn't have to be). All MySQL requires is that you have the necessary permission to execute the command entered. For right now we'll assume that the only user is the root user. We'll set permissions later.



Running MySQL Client Programs Under Windows (cont.)

This command allows you to see the current status of the MySQL server, in terms of the work being done.



```
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqladmin -u mark status
Uptime: 911  Threads: 1  Questions: 23  Slow queries: 0  Opens: 13  Flush tables
: 1  Open tables: 2  Queries per second avg: 0.025

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqladmin -u root status -p
Enter password: ****
Uptime: 930  Threads: 1  Questions: 24  Slow queries: 0  Opens: 13  Flush tables
: 1  Open tables: 2  Queries per second avg: 0.026

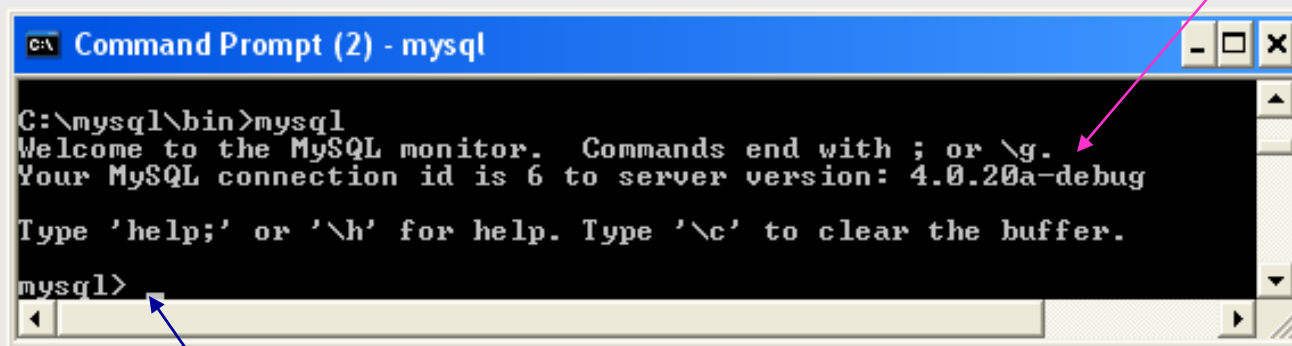
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
```



Starting A MySQL Client

- Once the MySQL server is started, open another command window in which to run the client process.
- You start a direct MySQL client application by executing the following command from the client window:

```
c:\mysql\bin> mysql
```



```
C:\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.20a-debug

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Normal start to MySQL

MySQL is awaiting commands



Starting A MySQL Client (cont.)

```
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysql -u mark
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21 to server version: 4.1.9

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

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

mysql> exit;
Bye

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22 to server version: 4.1.9

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+-----+
| Database |
+-----+
| bikes    |
| mysql    |
| prog3    |
| test     |
| testdb   |
+-----+
5 rows in set (0.00 sec)

mysql> exit;
Bye

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>
```

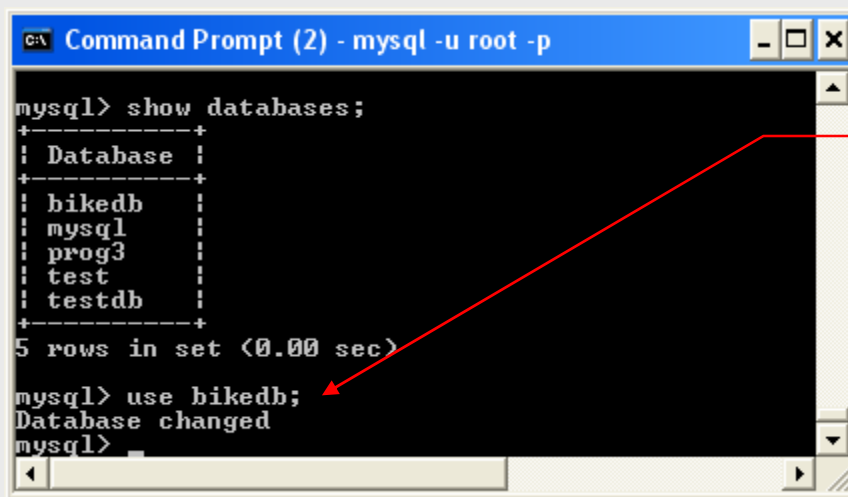
Start sequence for a non-root user

Start sequence for root user



Specifying A Database Within MySQL

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



```
C:\ Command Prompt (2) - mysql -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| test     |
| testdb   |
+-----+
5 rows in set (0.00 sec)

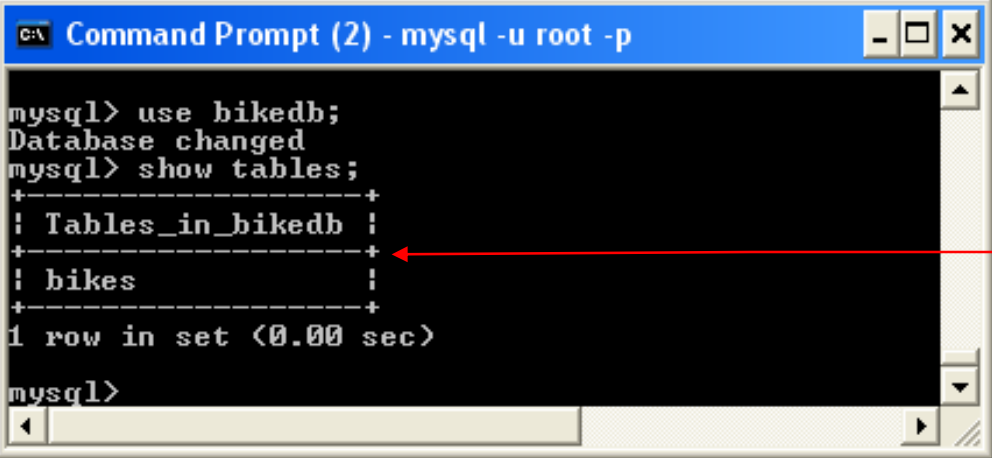
mysql> use bikedb;
Database changed
mysql>
```

SQL acknowledges selection of bikedb database.



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.



```
C:\ Command Prompt (2) - mysql -u root -p
mysql> use bikedb;
Database changed
mysql> show tables;
+-----+
| Tables_in_bikedb |
+-----+
| bikes              |
+-----+
1 row in set (0.00 sec)

mysql>
```

Show tables command lists all the relations within a database visible to the user. There is only a single table in this database.



Viewing the Schema of a Relation

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

```
C:\> Command Prompt (2) - mysql -u root -p
mysql> create table bikes (
-> bikename varchar(30) NOT NULL,
-> size int(2),
-> color varchar(15),
-> cost int(5),
-> purchased date,
-> mileage int(6),
-> primary key (bikename)
-> );
Query OK, 0 rows affected (0.05 sec)

mysql> describe bikes;
```

Field	Type	Null	Key	Default	Extra
bikename	varchar(30)		PRI		
size	int(2)	YES		NULL	
color	varchar(15)	YES		NULL	
cost	int(5)	YES		NULL	
purchased	date	YES		NULL	
mileage	int(6)	YES		NULL	

```
6 rows in set (0.00 sec)

mysql>
```

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



Running a Simple Select Query in MySQL

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

```
C:\ Command Prompt (2) - mysql -u root -p

mysql> describe bikes;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| bikename   | varchar(30)   |      | PRI |          |       |
| size       | int(2)        | YES  |     | NULL    |       |
| color      | varchar(15)   | YES  |     | NULL    |       |
| cost       | int(5)        | YES  |     | NULL    |       |
| purchased  | date          | YES  |     | NULL    |       |
| mileage    | int(6)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from bikes;
+-----+-----+-----+-----+-----+-----+
| bikename      | size | color      | cost | purchased   | mileage |
+-----+-----+-----+-----+-----+-----+
| Colnago Dream Rabobank | 60   | blue/orange | 5500 | 2002-06-27 | 4300    |
| Bianchi Evolution 3   | 58   | celeste    | 4800 | 2003-11-16 | 2000    |
| Eddy Merckx Molteni   | 58   | orange     | 5100 | 2004-08-12 | 0        |
| Eddy Merckx Domo      | 58   | blue/black | 5300 | 2004-02-02 | 0        |
| Battaglin Carrera    | 60   | red/white  | 4000 | 2001-03-14 | 11200   |
| Gianni Motta Personal | 59   | red/green  | 4400 | 2000-05-01 | 8700    |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

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



Creating a Database in MySQL

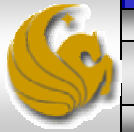
- There are two ways to create a MySQL database.
 1. From a client command window execute the `mysqladmin` script `create` and specify the name of the database.

```
C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqladmin -u root -p create sample
Enter password: ****

C:\Program Files\MySQL 4.1.9\MySQL Server 4.1\bin>mysqlshow -u root -p
Enter password: ****
+-----+
| Databases |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| sample   |
| test     |
| testdb   |
+-----+
```

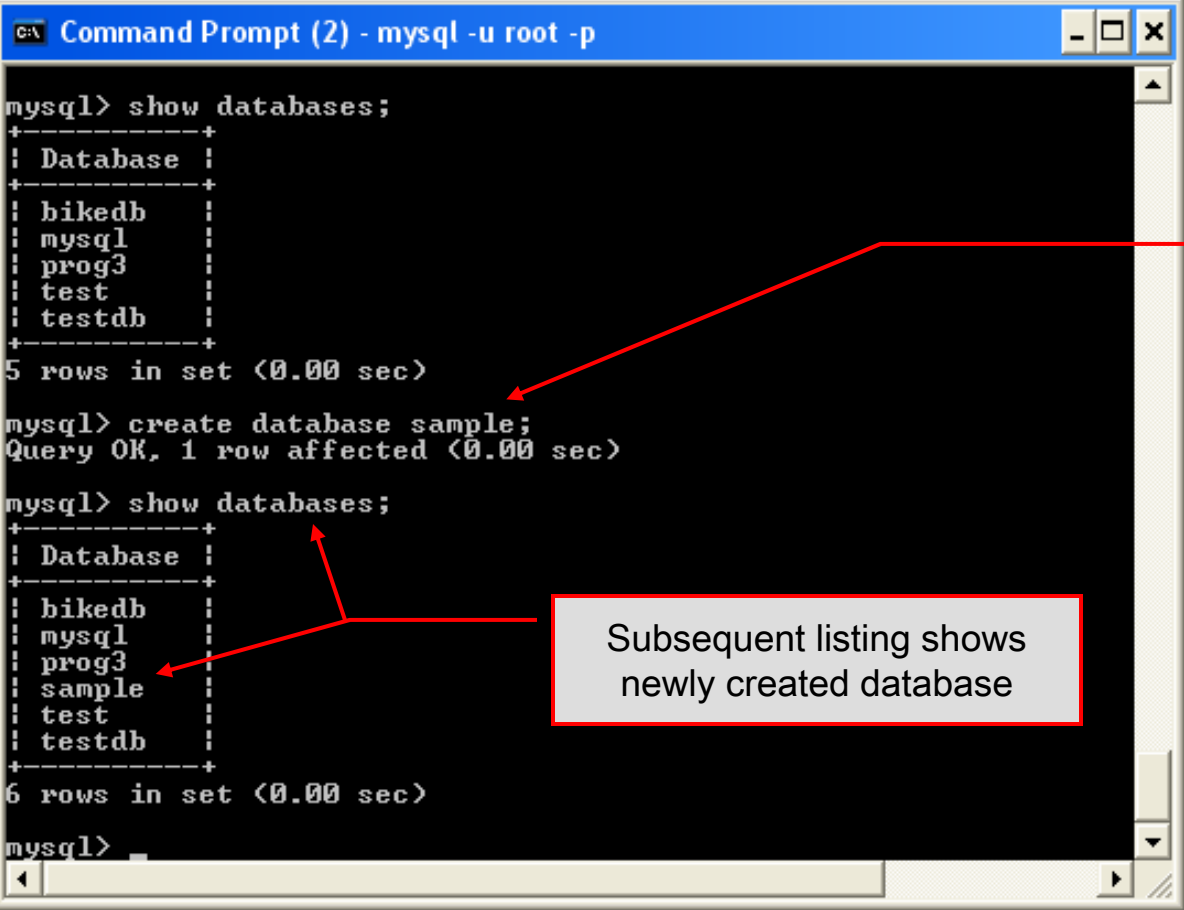
Specify creation of new database named sample

Subsequent listing of databases shows newly created sample database.



Creating a Database in MySQL (cont.)

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



```
C:\> Command Prompt (2) - mysql -u root -p

mysql> show databases;
+-----+
| Database |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| test     |
| testdb   |
+-----+
5 rows in set (0.00 sec)

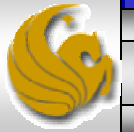
mysql> create database sample;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| sample   |
| test     |
| testdb   |
+-----+
6 rows in set (0.00 sec)

mysql>
```

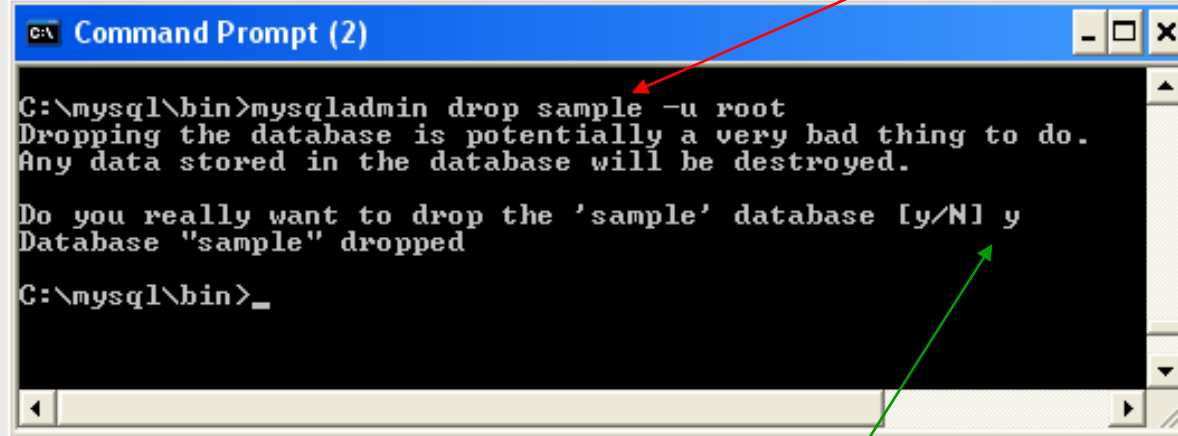
Create new database from within MySQL monitor.

Subsequent listing shows newly created database



Dropping a Database in MySQL

- There are two ways to destroy a MySQL database.
 1. From a client command window execute the `mysqladmin` script `drop` and specify the name of the database.



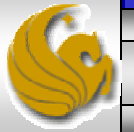
```
C:\mysql\bin>mysqladmin drop sample -u root
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'sample' database [y/N] y
Database "sample" dropped

C:\mysql\bin>_
```

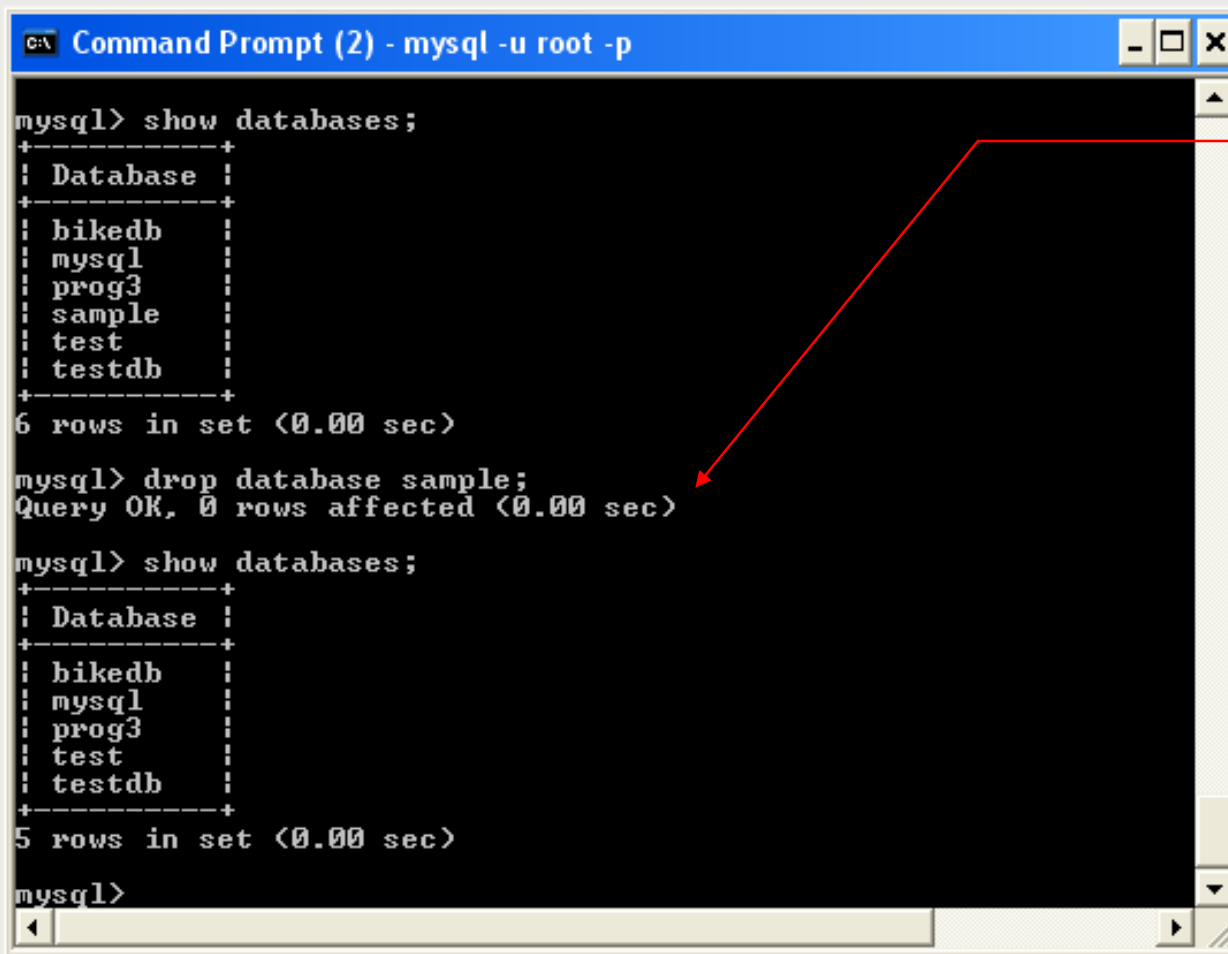
Specify dropping the database named sample.

MySQL gives you a warning and a bailout before the database is destroyed.



Dropping a Database in MySQL

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



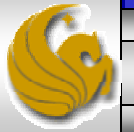
```
mysql> show databases;
+-----+
| Database |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| sample   |
| test     |
| testdb   |
+-----+
6 rows in set (0.00 sec)

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

mysql> show databases;
+-----+
| Database |
+-----+
| bikedb   |
| mysql    |
| prog3    |
| test     |
| testdb   |
+-----+
5 rows in set (0.00 sec)

mysql>
```

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



Manipulating Tables in MySQL

- The creation of a database does not place into the database any relations. Relations must be separately created.
- To create a table within a database, first select the database (or create one if you haven't already done so), then execute the create table command.

```
C:\ Command Prompt (2) - mysql -u root
mysql> create table articles <
-> article_id int(9) not null auto_increment,
-> headline text not null,
-> date_post datetime not null default '0000-00-00 00:00:00',
-> text_body text,
-> text_summary text,
-> who_created int(9) default null,
-> email_sent int(1) not null default '0',
-> date_email datetime default null,
-> who_approved int(9) default null,
-> pic varchar(255) default null,
-> primary key (article_id)
-> >;
Query OK, 0 rows affected (0.11 sec)
```



Manipulating Tables in MySQL (cont.)

Screen shot showing newly created table.

```
C:\> Command Prompt (2) - mysql -u root
mysql> describe articles;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          | Extra          |
+-----+-----+-----+-----+-----+-----+
| article_id     | int(9)        |      | PRI | NULL             | auto_incremen |
| headline       | text          |      |      |                  |                |
| date_post      | datetime      |      |      | 0000-00-00 00:00:00 |                |
| text_body      | text          | YES  |      | NULL             |                |
| text_summary   | text          | YES  |      | NULL             |                |
| who_created    | int(9)        | YES  |      | NULL             |                |
| email_sent     | int(1)        |      |      | 0                 |                |
| date_email     | datetime      | YES  |      | NULL             |                |
| who_approved   | int(9)        | YES  |      | NULL             |                |
| pic            | varchar(255)  | YES  |      | NULL             |                |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```



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

```
Command Prompt (2) - mysql -u root -p
mysql> select * from bikes;
+-----+-----+-----+-----+-----+-----+
| bikename      | size | color      | cost  | purchased | mileage |
+-----+-----+-----+-----+-----+-----+
| Colnago Dream Rabobank | 60   | blue/orange | 5500  | 2002-06-27 | 4300   |
| Bianchi Evolution 3   | 58   | celeste    | 4800  | 2003-11-16 | 2000   |
| Eddy Merckx Molteni   | 58   | orange     | 5100  | 2004-08-12 | 0      |
| Eddy Merckx Domo      | 58   | blue/black  | 5300  | 2004-02-02 | 0      |
| Battaglin Carrera     | 60   | red/white  | 4000  | 2001-03-14 | 11200  |
| Gianni Motta Personal | 59   | red/green  | 4400  | 2000-05-01 | 8700   |
| Gios Torino Super     | 60   | blue       | 3800  | 1998-11-08 | 9000   |
| Schwinn Paramount P14 | 60   | blue       | 1800  | 1992-03-01 | 200    |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> create temporary table bluebikes
-> select *
-> from bikes
-> where color = 'blue';
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> show tables;
+-----+
| Tables_in_bikedb |
+-----+
| bikes              |
+-----+
1 row in set (0.00 sec)

mysql> select * from bluebikes;
+-----+-----+-----+-----+-----+-----+
| bikename      | size | color | cost  | purchased | mileage |
+-----+-----+-----+-----+-----+-----+
| Gios Torino Super | 60   | blue  | 3800  | 1998-11-08 | 9000   |
| Schwinn Paramount P14 | 60   | blue  | 1800  | 1992-03-01 | 200    |
+-----+-----+-----+-----+-----+-----+
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.

