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

- **SQL Tool**
- **JDBC**
- **ODBC**
- **Database Driver**
- **Java Database Client**
- **JDBC API**
- **JDBC-ODBC Bridge**
- **ODBC API**
- **DB-specific API**
- **Database**
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 stable version is 5.0.18 and can be downloaded from www.mysql.com. (Any of the versions of MySQL 5.0 will be fine for our purposes.)

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

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

MySQL 5.0
- Generally Available (GA) release for production use

Upcoming Releases:
- MySQL 5.0 -- Alpha release, Test new features early
- Source code snapshots of the development branch

Older Releases:
- MySQL 4.1 -- Previous GA release
- Archives of Older Releases

MySQL Cluster

MySQL Cluster is included in version 5.0 of the MySQL database server, as part of the MySQL Max packages. Binaries and source are available from the MySQL 5.0 download page.

MySQL Tools

MySQL also develop Graphical User Interface applications for administering MySQL Server and working with data:
- MySQL Workbench -- Use this graphical client to work with your MySQL databases and run queries.
- MySQL Administrator -- Manage MySQL Server
- MySQL Query Browser -- Use this graphical client to work with your MySQL databases and run queries.

Drivers and Connectors

While many programming languages have included support for connecting to MySQL server, additional drivers are available:

- MySQL Connector/Java -- Connect to MySQL database applications.
Select version 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.
Go back to the main download page and also download MySQL Administrator and MySQL Query Browser.

Also download the MySQL Connector/J 3.1.
Installing MySQL 5.0

• Once you’ve got MySQL downloaded, go through the installation process. It may vary somewhat depending on platform.

• I’ve illustrated the basic install on Windows XP over the next few pages, just to give you an idea of what you should be seeing.
Installing MySQL 5.0

• Once you’ve got MySQL downloaded, go through the installation process. It may vary somewhat depending on platform.

• I’ve illustrated the basic install on Windows XP over the next few pages, just to give you an idea of what you should be seeing.

• Once the Window installer is running you should see the following window appear:
Installing MySQL 5.0 (cont.)

Your choice here. For this course, a typical set-up will be fine.
Installing MySQL 5.0 (cont.)

Select the destination folder for the install.

Current Settings:
Setup Type: Typical
Destination Folder: E:\Program Files\MySQL\MySQL Server 5.0\

< Back Install Cancel
Again, your choice here. If you want to skip the sign-up that's fine.
If everything has good well up to this point, you should see a window similar to this one. Click the Finish button, cross your fingers, and hang-on while the installer configures your system and gets MySQL up and running as a service.
Installing MySQL 5.0 (cont.)

The Configuration Wizard will allow you to configure the MySQL Server 5.0 server instance. To Continue, click Next.
Installing MySQL 5.0 (cont.)

Your choice here. If you are not sure if there is already a MySQL server on your machine, choose the detailed configuration setting.
Installing MySQL 5.0 (cont.)

Choose the developer machine option

Choose the developer machine option

My SQL Server Instance Configuration Wizard

MySQL Server Instance Configuration
Configure the MySQL Server 5.0 server instance.

Please select a server type. This will influence memory, disk and CPU usage.

- **Developer Machine**
  - This is a development machine, and many other applications will be run on it. MySQL Server should only use a minimal amount of memory.

- **Server Machine**
  - Several server applications will be running on this machine. Choose this option for web/application servers. MySQL will have medium memory usage.

- **Dedicated MySQL Server Machine**
  - This machine is dedicated to run the MySQL Database Server. No other servers, such as a web or mail server, will be run. MySQL will utilize up to all available memory.

< Back   Next >   Cancel
Installing MySQL 5.0 (cont.)

Choose the multifunctional database option

MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration

Configure the MySQL Server 5.0 server instance.

Please select the database usage.

- **Multifunctional Database**
  - General purpose databases. This will optimize the server for the use of the fast transactional InnoDB storage engine and the high speed MyISAM storage engine.

- **Transactional Database Only**
  - Optimized for application servers and transactional web applications. This will make InnoDB the main storage engine. Note that the MyISAM engine can still be used.

- **Non-Transactional Database Only**
  - Suited for simple web applications, monitoring or logging applications as well as analysis programs. Only the non-transactional MyISAM storage engine will be activated.
Installing MySQL 5.0 (cont.)

Choose the installation path to keep InnoDB tables in same area as other MySQL files.
Select manual setting for this option. The default is 15, I set mine to 10, but you can use any number you would like, but pick something greater than 2 or 3.
Installing MySQL 5.0 (cont.)

Accept all defaults in this window.

MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration
Configure the MySQL Server 5.0 server instance.

Please set the networking options.

**Enable TCP/IP Networking**
Enable this to allow TCP/IP connections. When disabled, only local connections through named pipes are allowed.

Port Number: 3306

Please set the server SQL mode.

**Enable Strict Mode**
This option forces the server to behave more like a traditional database server. It is recommended to enable this option.

< Back  Next >  Cancel
Installing MySQL 5.0 (cont.)

Your choice again
Installing MySQL 5.0 (cont.)

Accept default options

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

- **Install As Windows Service**: This is the recommended way to run the MySQL server on Windows.
  - **Service Name**: MySQL
  - Launch the MySQL Server automatically

- **Include Bin Directory in Windows PATH**: Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.
Installing MySQL 5.0 (cont.)

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

Do not enable this option
Installing MySQL 5.0 (cont.)

Configuration is about to begin. Now cross your fingers, toes, and anything else you have, take a deep breath, click the Execute button and close your eyes for a few seconds.
Installing MySQL 5.0 (cont.)

You've successfully installed MySQL!!

MySQL Server Instance Configuration Wizard

MySQL Server Instance Configuration
Configure the MySQL Server 5.0 server instance.

Processing configuration ...

- Prepare configuration
- Write configuration file (E:\Program Files\MySQL\MySQL Server 5.0\my.ini)
- Start service
- Apply security settings

Configuration file created.
Windows service MySQL installed.
Service started successfully.
Security settings applied.

Press [Finish] to close the Wizard.
Running MySQL 5.0

• If you’ve successfully installed MySQL, it should now be running as a service on your machine. It will start automatically when your machine boots.

• Go into your listing of programs (from the start menu at the bottom: All Programs) and you should see MySQL appear. Since you will be running MySQL clients a lot, it will be easier if you pin the MySQL Client to the start menu.

• To verify that MySQL is running properly as a service you can either check the process window or run a MySQL client.
Enter the password you provided during the MySQL installation procedure as the root user.

Hopefully, you see this output from MySQL. The MySQL server is now awaiting a command from this client.
List all databases managed by this MySQL server which are accessible to this client.

Terminate client connection.
Specifying A Database Within MySQL

• Unless, it is specifically stated, in the following slides we’ll assume that the user has root-level privileges.

• To select a database for use in MySQL the `use` command must be issued. In the example below, we’ll select the `bikedb` database.
Viewing the Schema of a Relation

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

```sql
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 in 0.05 sec
mysql> describe bikes;

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>bikename</td>
<td>varchar(30)</td>
<td>YES</td>
<td>PRI</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>int(2)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>color</td>
<td>varchar(15)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>cost</td>
<td>int(5)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>purchased</td>
<td>date</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
<tr>
<td>mileage</td>
<td>int(6)</td>
<td>YES</td>
<td>NULL</td>
<td>NULL</td>
<td></td>
</tr>
</tbody>
</table>
6 rows in set in 0.00 sec
mysql>
```

Specify which table’s schema to describe. All information regarding the schema visible to the user is displayed.
Viewing the Relations of a Database

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

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

Show tables command lists all the relations within a database visible to the user. There are two tables in this database.
Running a Simple Select Query in MySQL

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

```
mysql> select * from bikes;
+-----------+--------+--------+-------+----------+----------+
<table>
<thead>
<tr>
<th>bikename</th>
<th>size</th>
<th>color</th>
<th>cost</th>
<th>purchased</th>
<th>mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battaglin Carrera</td>
<td>60</td>
<td>red/white</td>
<td>4000</td>
<td>2001-03-14</td>
<td>11200</td>
</tr>
<tr>
<td>Bianchi Corse Evo 4</td>
<td>58</td>
<td>celeste</td>
<td>5700</td>
<td>2004-12-22</td>
<td>300</td>
</tr>
<tr>
<td>Bianchi Evolution 3</td>
<td>58</td>
<td>celeste</td>
<td>4800</td>
<td>2003-11-16</td>
<td>2000</td>
</tr>
<tr>
<td>Bianchi/Liquigas FG</td>
<td>58</td>
<td>celeste/blue</td>
<td>5600</td>
<td>2005-12-02</td>
<td>0</td>
</tr>
<tr>
<td>Colnago Dream Rabobank</td>
<td>60</td>
<td>blue/orange</td>
<td>5500</td>
<td>2002-07-27</td>
<td>4300</td>
</tr>
<tr>
<td>Colnago Superissimo</td>
<td>59</td>
<td>red</td>
<td>3800</td>
<td>1996-03-01</td>
<td>13000</td>
</tr>
<tr>
<td>Eddy Merckx Domo</td>
<td>58</td>
<td>blue/black</td>
<td>5300</td>
<td>2005-02-02</td>
<td>0</td>
</tr>
<tr>
<td>Eddy Merckx Molteni</td>
<td>58</td>
<td>orange</td>
<td>5100</td>
<td>2004-08-12</td>
<td>0</td>
</tr>
<tr>
<td>Gianni Motta Personal</td>
<td>59</td>
<td>red/green</td>
<td>4400</td>
<td>2000-05-01</td>
<td>8700</td>
</tr>
<tr>
<td>Gios Torino Super</td>
<td>60</td>
<td>blue</td>
<td>2000</td>
<td>1998-11-08</td>
<td>9000</td>
</tr>
<tr>
<td>Schwinn Paramount P14</td>
<td>60</td>
<td>blue</td>
<td>1800</td>
<td>1992-03-01</td>
<td>200</td>
</tr>
</tbody>
</table>
+-----------+--------+--------+-------+----------+----------+
11 rows in set (0.00 sec)
```
Creating a Database in MySQL

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

![MySQL Command Line Client]

Create new database from within MySQL monitor.

Subsequent listing shows newly created database.
Dropping a Database in MySQL

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

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

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

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

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

mysql> use sample;
Database changed

mysql> create table articles (  
    -> article_id int(9) not null auto_increment,  
    -> headline text not null,  
    -> data_post datetime not null default '0000-00-00 00:00:00',  
    -> text_body text,  
    -> who_created int(9) default null,  
    -> email_sent int(1) not null default '0',  
    -> date_email datetime default null,  
    -> who_approved int(9) default null,  
    -> pic varchar(255) default null,  
    -> primary key (article_id)
    -> );
Query OK, 0 rows affected (0.08 sec)

mysql> 
```
Screen shot showing newly created table.
Manipulating Tables in MySQL (cont.)

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

  ```sql
  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

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.

```
mysql> select * from bikes;
+-------------+--------+-------+--------+------------------+--------+
<table>
<thead>
<tr>
<th>bikename</th>
<th>size</th>
<th>color</th>
<th>cost</th>
<th>purchased</th>
<th>mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battaglin Carrera</td>
<td>60</td>
<td>red/white</td>
<td>4000</td>
<td>2001-03-14</td>
<td>11200</td>
</tr>
<tr>
<td>Bianchi Corse Evo 4</td>
<td>58</td>
<td>celeste</td>
<td>5700</td>
<td>2004-12-22</td>
<td>300</td>
</tr>
<tr>
<td>Bianchi Evolution 3</td>
<td>58</td>
<td>celeste</td>
<td>4800</td>
<td>2003-11-16</td>
<td>2000</td>
</tr>
<tr>
<td>Bianchi/Aliqua FG</td>
<td>58</td>
<td>celeste/blue</td>
<td>5600</td>
<td>2005-12-02</td>
<td>0</td>
</tr>
<tr>
<td>Colnago Dream Rabobank</td>
<td>60</td>
<td>blue/orange</td>
<td>5500</td>
<td>2002-07-27</td>
<td>300</td>
</tr>
<tr>
<td>Colnago Superissimo</td>
<td>59</td>
<td>red</td>
<td>3800</td>
<td>1996-03-01</td>
<td>13000</td>
</tr>
<tr>
<td>Eddy Merckx Domo</td>
<td>58</td>
<td>blue/black</td>
<td>5300</td>
<td>2005-02-02</td>
<td>0</td>
</tr>
<tr>
<td>Eddy Merckx Molteni</td>
<td>58</td>
<td>orange</td>
<td>5100</td>
<td>2004-06-12</td>
<td>0</td>
</tr>
<tr>
<td>Giacinto Rotta Personal</td>
<td>59</td>
<td>red/green</td>
<td>4400</td>
<td>2000-05-01</td>
<td>8700</td>
</tr>
<tr>
<td>Gios Torino Super</td>
<td>60</td>
<td>blue</td>
<td>2000</td>
<td>1998-11-00</td>
<td>9000</td>
</tr>
<tr>
<td>Schwinn Paramount P14</td>
<td>60</td>
<td>blue</td>
<td>1800</td>
<td>1992-03-01</td>
<td>200</td>
</tr>
</tbody>
</table>
+-------------+--------+-------+--------+------------------+--------+
11 rows in set (0.00 sec)
```

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

```
mysql> select * from celestebikes;
+-------------+--------+-------+--------+------------------+--------+
<table>
<thead>
<tr>
<th>bikename</th>
<th>size</th>
<th>color</th>
<th>cost</th>
<th>purchased</th>
<th>mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bianchi Corse Evo 4</td>
<td>58</td>
<td>celeste</td>
<td>5700</td>
<td>2004-12-22</td>
<td>300</td>
</tr>
<tr>
<td>Bianchi Evolution 3</td>
<td>58</td>
<td>celeste</td>
<td>4800</td>
<td>2003-11-16</td>
<td>2000</td>
</tr>
</tbody>
</table>
+-------------+--------+-------+--------+------------------+--------+
2 rows in set (0.00 sec)
```

Notice that temporary tables do not appear in a table listing.
A First Look At The MySQL Query Browser

- Query input window
- Database selection window
- Result set window