

# JAVA BOOTCAMP

# MySQL

# INTRO TO MYSQL

# TOPICS

- Introduction to MySQL
- Workbench
- Creating a database
- Data manipulation

# **INTRO TO MYSQL**

## **RELATIONAL DATABASES**

- A relational database consists of one or more tables that consist of rows (records) and columns (fields).
- These table are related by keys.
- The primary key in a table is the one that uniquely identifies each of the rows in the table.
- A foreign key is used to relate the rows in one table to the rows in another table.

# WORKBENCH

# **MYSQL**

## **MYSQL IS:**

- Inexpensive
- Fast
- Easy to use
- Portable

# **MYSQL**

## **MYSQL PROVIDES:**

- Support for SQL
- Support for multiple clients
- Connectivity
- Security
- Referential integrity
- Transaction processing

# MYSQL TOOLS

- MySQL - The MySQL database server, which manages databases and tables, controls user access, and processes SQL queries.
- MySQL-client - MySQL client programs, which make it possible to connect to and interact with the server.
- MySQL-devel - Libraries and header files that come in handy when compiling other programs that use MySQL.
- MySQL-shared - Shared libraries for the MySQL client.
- MySQL-bench - Benchmark and performance

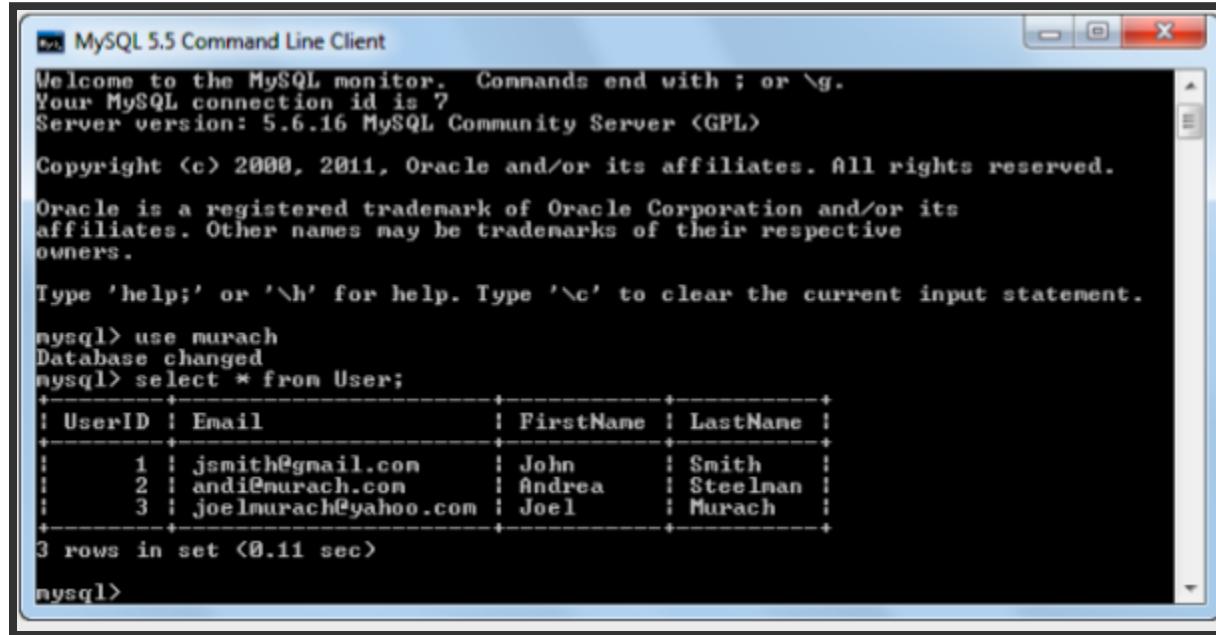
testing tools for the MySQL database server.

## INSTALLING MYSQL

- Go to <http://www.mysql.com/downloads/>
- Step by step installation

# CREATING A DATABASE

# USING MYSQL



The screenshot shows a Windows command-line window titled "MySQL 5.5 Command Line Client". The window displays the MySQL monitor welcome message, copyright information, and a help message. It then shows the user switching to the "murach" database and executing a SELECT query to retrieve data from the "User" table. The output shows three rows of data with columns: UserID, Email, FirstName, and LastName.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.6.16 MySQL Community Server <GPL>

Copyright (c) 2000, 2011, 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.

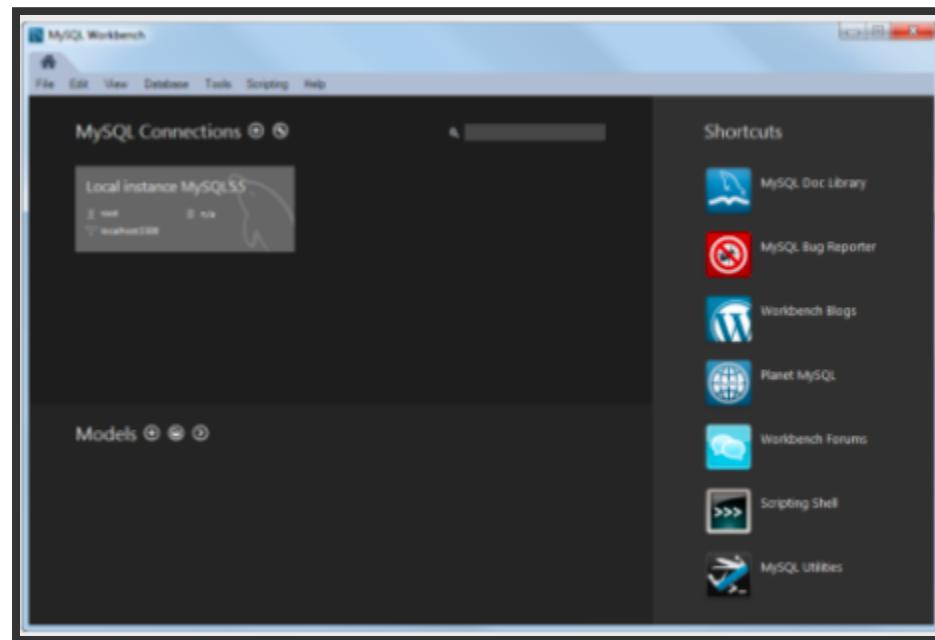
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use murach
Database changed
mysql> select * from User;
+-----+-----+-----+-----+
| UserID | Email           | FirstName | LastName |
+-----+-----+-----+-----+
|      1 | jsmith@gmail.com | John       | Smith     |
|      2 | andi@murach.com | Andrea    | Steelman  |
|      3 | joelmurach@yahoo.com | Joel      | Murach   |
+-----+-----+-----+-----+
3 rows in set (0.11 sec)

mysql>
```

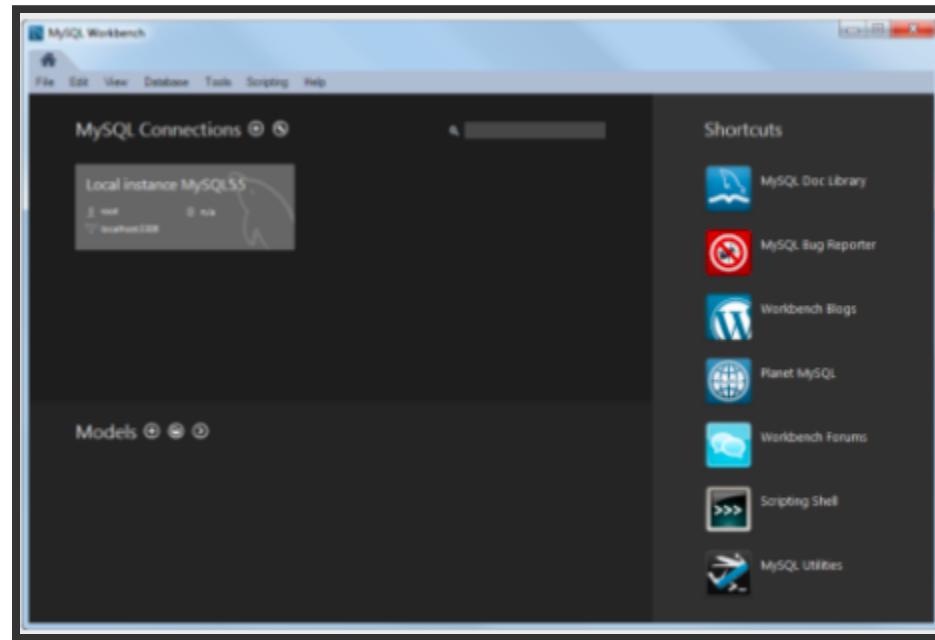
A commandline-tool

# USING MYSQL



The home tab

# USING MYSQL



Dialog box for opening database connections.

# CREATING TABLES

# CREATING TABLES

- CREATE TABLE table\_name (column\_name column\_type);
- Example:

```
CREATE TABLE tutorials_tbl (
    tutorial_id      INT
NOT NULL AUTO_INCREMENT,
    tutorial_title   VARCHAR(100) NOT NULL,
    tutorial_author  VARCHAR(40)  NOT NULL,
    submission_date  DATE,
    PRIMARY KEY ( tutorial_id )
);
```

# CREATING TABLES

- PRIMARY KEY
- NOT NULL
- AUTO\_INCREMENT
- Check constraints

# DATA TYPES

# MYSQL DATATYPES

Selecting a datatype for your fields

- Properly defining the fields in a table is important to the overall optimization of your database.
- You should use only the type and size of field you really need to use; don't define a field as 10 characters wide if you know you're only going to use 2 characters. These types of fields (or columns) are also referred to as data types, after the type of data you will be storing in those fields.

# MYSQL DATATYPES

MySQL uses many different data types broken into three categories:

- Numeric.
- Date and time.
- String types.

# NUMERIC DATATYPES

- MySQL uses all the standard ANSI SQL numeric data types
  - *INT (4 byte)*: A normal-sized integer that can be signed or unsigned. If signed, the allowable range is from -2147483648 to 2147483647. If unsigned, the allowable range is from 0 to 4294967295.
  - *TINYINT (1 byte)*: A very small integer that can be signed or unsigned. If signed, the allowable range is from -128 to 127. If unsigned, the allowable range is from 0 to 255.

# NUMERIC DATATYPES

- *SMALLINT (2 byte)* - A small integer that can be signed or unsigned. If signed, the allowable range is from -32768 to 32767. If unsigned, the allowable range is from 0 to 65535.
- *MEDIUMINT* - A medium-sized integer that can be signed or unsigned. If signed, the allowable range is from -8388608 to 8388607. If unsigned, the allowable range is from 0 to 16777215.

# NUMERIC DATATYPES

- *BIGINT (8 bytes)*- A large integer that can be signed or unsigned. If signed, the allowable range is from -9223372036854775808 to 9223372036854775807. If unsigned, the allowable range is from 0 to 18446744073709551615.

# NUMERIC DATATYPES

- *FLOAT(M,D)* - A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.
- *DOUBLE(M,D)* - A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.

# DATE AND TIME TYPES

- *DATE* - A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.
- *DATETIME* - A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.

# DATE AND TIME TYPES

- ***TIMESTAMP*** - A timestamp between midnight, January 1, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 ( YYYYMMDDHHMMSS ).
- ***TIME*** - Stores the time in HH:MM:SS format.
- ***YEAR(M)*** - Stores a year in 2-digit or 4-digit format. If the length is specified as 2 (for example YEAR(2)), YEAR can be 1970 to 2069 (70 to 69). If the length is specified as 4, YEAR can be 1901 to 2155. The default length is 4.

1901 to 2155. The default length is 4.

## STRING TYPES

- *CHAR(M)* - A fixed-length string between 1 and 255 characters in length (for example CHAR(5)), right-padded with spaces to the specified length when stored. Defining a length is not required, but the default is 1.
- *VARCHAR(M)* - A variable-length string between 1 and 255 characters in length; for example VARCHAR(25). You must define a length when creating a VARCHAR field.

# STRING TYPES

- *BLOB* or *TEXT*- A field with a maximum length of 65535 characters. BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data; the difference between the two is that sorts and comparisons on stored data are case sensitive on BLOBs and are not case sensitive in TEXT fields. You do not specify a length with BLOB or TEXT.

# CREATING DATABASES

# **CREATING DBS**

## **CREATING DATABASES**

Use MySQL to create databases

# TABLES

## CREATE AND DROP A TABLE

- We use the *CREATE TABLE* statement to create a table and the *DROP TABLE* statement to delete a table.
- We can use the *DROP TABLE IF EXISTS* statement to guard against an error resulting from attempting to delete a table that does not exist.

# DATA QUERY

## SELECTING DATA FROM A SINGLE TABLE

- A *SELECT* statement is a DML statement that returns a result set that consists of the specified rows and columns.
- We specify columns with the *SELECT* clause and rows with the *WHERE* clause.

# DATA QUERY

## SELECTING DATA FROM A SINGLE TABLE

- We specify the table the data should come from using the *FROM* clause.
- The *ORDER BY* clause specifies the way data should be sorted.

# DATA QUERY

## SELECTING DATA FROM MULTIPLE TABLES

- To return a result set that contains data from two tables, we need to join them using a *JOIN clause*. This is usually going to be an *INNER JOIN* (the default) so that rows are only included when the key of a row in the first table matches the key of a row in the second table.
- In a *left outer join*, the data for all the rows in the first table (the one on the left) are included, but only data for matching rows in the second table. For a *right outer join*, the opposite is true.

# DATA MANIPULATION

# **SQL STATEMENTS**

## **STATEMENT TYPES**

- Data Definition Language: DDL (Creating tables and databases)
- Data Manipulation Language: DML (Adding, modifying, and deleting data)
- Data Control language: DCL (Control access to data)

# DML

## *INSERT STATEMENT*

```
INSERT INTO table-name [ (column-list) ]  
VALUES (value-list);
```

# DML

## *INSERT STATEMENT*

```
INSERT INTO Movie (name, category)
VALUES ('Airplane', 'Comedy');
```

# DML

## *UPDATE STATEMENT*

```
UPDATE table-name  
SET expression-1 [, expression-2] ...  
WHERE selection-criteria;
```

# DML

## *UPDATE STATEMENT*

```
UPDATE Movie  
SET category = 'scifi'  
WHERE name = 'Alien';
```

# DML

## ***DELETE STATEMENT***

```
DELETE FROM table-name  
WHERE selection-criteria;
```

# DML

## ***DELETE STATEMENT***

```
DELETE FROM Movie  
WHERE name = 'Star Wars: Episode I';
```

# **BACKUP AND RESTORE**

## **BACKUP AND RESTORE USING MYSQL**

Use MySQL to backup and restore data.

# RECAP

# RECAP

## WHAT YOU SHOULD KNOW AT THIS POINT

- What are relational databases
- Know different relational DB products
- How MySQL differs from other DB products
- What are the different MySQL tools
- How to install and configure MySQL

# RECAP

## WHAT YOU SHOULD KNOW AT THIS POINT

- Use MySQL command prompt and workbench
- How to create tables
- How to create DBs (schema)
- Know MySQL datatypes
- How to use MySQL to query and modify data
- How to backup and restore data on MySQL