

CSCI-UA.60-1 Database Design and Web Implementation

MySQL - Part I

Prof Deena Engel

Department of Computer Science

deena@cs.nyu.edu

MySQL – Part 1

- ▶ We will start working in MySQL using one table at a time, to review our work in SQLite.
- ▶ Goals:
 - Learning to work in MySQL on i6
 - Learning to create, populate, update and manipulate tables in MySQL
 - Using MySQL for data analysis
 - Using LOAD DATA to import data from external sites for data analysis



MySQL Environment on i6

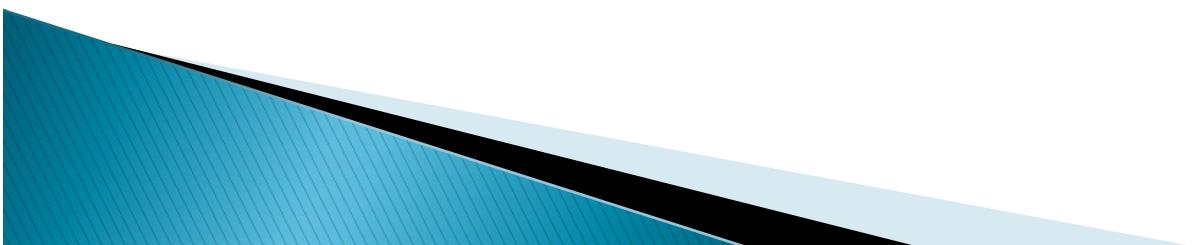
- ▶ For instructions for running MySQL on i6, see the instructions on the course site
- ▶ If you wish to view your data from time to time within a graphical environment, you can also use PHPMYADMIN which is available at <https://cims.nyu.edu/phpMyAdmin/>
 - When using PHPMYADMIN, be sure to login with your netid and your database password (**not** your i6 password)

MySQL Statements

- ▶ As in SQLite, commands in MySQL are called queries; queries can be used to create tables, cull data, update records, delete data, alter tables, etc.
- ▶ Full documentation on MySQL statements is found at
<http://dev.mysql.com/doc/refman/5.6/en/sql-syntax.html>

MySQL: Working with Tables

- ▶ SHOW TABLES – to see all of the tables in your database
 - This is similar to `.tables` in SQLite
- ▶ CREATE TABLE – to create a new table
 - Note that if a table already exists by that name, that you will not be able to create it again; you must first ...
- ▶ DROP TABLE – to remove a table from your database
- ▶ DESC TABLE – to describe a table listing all of the fields and attributes
 - This is similar to `.schema` in SQLite



Using INSERT to add records

- ▶ Use the INSERT query to add records to your table. If your data values are in the same order as the fields in the table, you do not need to specify the field names. For example:
- ▶ For example:

```
mysql> CREATE TABLE novels (
    -> title VARCHAR(30),
    -> author VARCHAR(25)
    -> );
mysql> INSERT INTO novels VALUES("Pride and Prejudice", "Austen,
Jane");
```

Using AUTO_INCREMENT

- ▶ The keyword AUTO_INCREMENT in a CREATE TABLE statement with an integer-typed field will cause MySQL to start at 1 and add 1 to each subsequent record in the table. This will give you a unique value in that field for identification purposes. (This is similar to your “N#” here at NYU.) This is a *surrogate key*.
- ▶ For example:

```
CREATE TABLE books2 ( ID SMALLINT UNSIGNED NOT NULL  
AUTO_INCREMENT ...
```

Use SELECT to cull data from the table

- ▶ There are three steps in culling data from one or more tables.
 - PROJECTION – which columns or fields do you want to see?
 - SELECTION – which rows or records do you want to see?
 - JOIN – working with two or more tables by building on the relationships among the tables.
- ▶ All of the work in the next few slides is analogous to our classwork in SQLite except
 - There is one exception: the concatenation operator is an operator in SQLite (||) and a function in MySQL (CONCAT(x, y, z ...))

PROJECTION

- ▶ Very confusing ... but we use the first line of the SELECT statement to PROJECT our results!
- ▶ For example

```
SELECT title,author,price  
      -> FROM books
```

- ▶ ... will display three columns, one each for title, author and price of the books table we created in class.



SELECT statements: keywords

- ▶ SELECT – projects the fields or columns
- ▶ DISTINCT – unique results
- ▶ AS – alias for a field (usually a calculated result)
- ▶ FROM – the table or tables you are using
- ▶ WHERE – data selection criteria
- ▶ GROUP BY – to aggregate data
- ▶ ORDER BY – the order of the records in the result set
- ▶ LIMIT n – how many records you wish to see
- ▶ *Note that SELECT and FROM are required; the rest is optional.*
- ▶ *Note that this is analogous to our work in SQLite.*

More on SELECT statements

- ▶ Use * and % as wildcards; for example,

```
SELECT *
FROM books;
```

- ▶ ... will yield all columns.

```
SELECT *
FROM books
WHERE author LIKE "T%";
```

- ▶ ... will yield all columns for books whose author's name begins with the letter "T".

SELECT statements, continued

- ▶ Use AS to alias (or “re-name”) projected columns that are a result of a calculation. For example:

```
SELECT CONCAT(title, ',author,') AS book ...
```

```
SELECT title,author,price*1.0875 AS priceWithTax...
```

SELECT statements, continued

- ▶ Use **and** and **or** to further clarify your WHERE clause in your record selection; for example,

```
SELECT title,author,year_written,price  
-> FROM books  
-> WHERE year_written >=1900  
-> AND year_written <2000  
-> AND price > 10  
-> ORDER BY price DESC;
```



Data Aggregation

- ▶ The GROUP BY clause will allow you to aggregate or sub-total data. For example, what is the average price of books for each publisher in our collection?

```
SELECT edition, AVG(price) AS avgPrice  
-> FROM books  
-> GROUP BY edition  
-> ORDER BY edition;
```



Using LOAD DATA

- ▶ MySQL supports data imports in a number of formats.
- ▶ LOAD DATA in MySQL is analogous to `.import` in SQLite.
- ▶ For example:

```
LOAD DATA LOCAL INFILE 'book_data.txt' INTO TABLE books;
```
- ▶ Details on the syntax are posted on the MySQL Readings pages of the course site.

Using the DELETE query

- ▶ DELETE can be used by itself to “empty” a table; however, in such a case the table still exists (whereas DROP TABLE will remove the table entirely).

```
DELETE FROM books;
```

- ▶ A condition will allow you to delete records selectively, as in this case, only “Penguin” editions will be deleted from our books collection:

```
DELETE  
-> FROM books  
-> WHERE edition="Penguin";
```



Using the UPDATE query

- ▶ An UPDATE query can be run against the entire table; in this case, we will raise all prices by 20%:

```
UPDATE books  
-> SET price = price*1.2;
```

- ▶ An UPDATE query can also be applied selectively; here we raised the prices of Penguin editions in our collection by an additional 30%:

```
UPDATE books  
-> SET price=price*1.3  
-> WHERE edition="Penguin";
```

Functions and Operators

- ▶ MySQL and all of the SQL languages offer a large library of built-in functions.
- ▶ A Summary of functions we will use in class is posted to the course site
- ▶ ... and additional documentation is posted to
<http://dev.mysql.com/doc/refman/5.6/en/functions.html>

