CSC 337

Relational Databases and SQL

Rick Mercer

Relational databases

- Relational database: A method of structuring data as tables associated to each other by shared attributes
- A table row corresponds to a unit of data called a record; a column corresponds to an attribute of that record
- Relational databases typically use Structured Query Language (SQL) to define, manage, and search data

One view of a Database

- Like a List of objects

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

Why use a database?

- Powerful: can search, filter data, and combine data from multiple sources
- Fast: can search/filter a database very quickly compared to a file
- Big: scale well up to very large data sizes
- Safe: built-in mechanisms for failure recovery
- Multi-user: concurrency features let many users view/edit data at same time
- Abstract: provides layer of abstraction between stored data and app(s)
  - Many database programs understand the same SQL commands

Why use a Relational database?

Much time, money, and care have gone to make efficient, fast, reliable, robust DBMSs

Database software

- There are many database management systems
  - Oracle
  - MySQL Oracle bought this with Java

  Microsoft SQL Server (powerful)
  - Microsoft Access (simple, part of MS Office)
  - PostgreSQL (powerful/complex free open-source database system)
  - SQLite (transportable, lightweight free open-source database system)
    - Comes with most Macs and linux distributes
    - This software is on your cell phone
      - Android and Iphone

Our system

- MariaDB (free open-source database system)
  - A fork from Oracle's MySQL
  - Many servers run "LAMP" (Linux, Apache, MySQL, and PHP)
  - Wikipedia is run on PHP and MySQL
  - We'll use MySQL in this course
  - You already have it
    - It comes with XAMMP

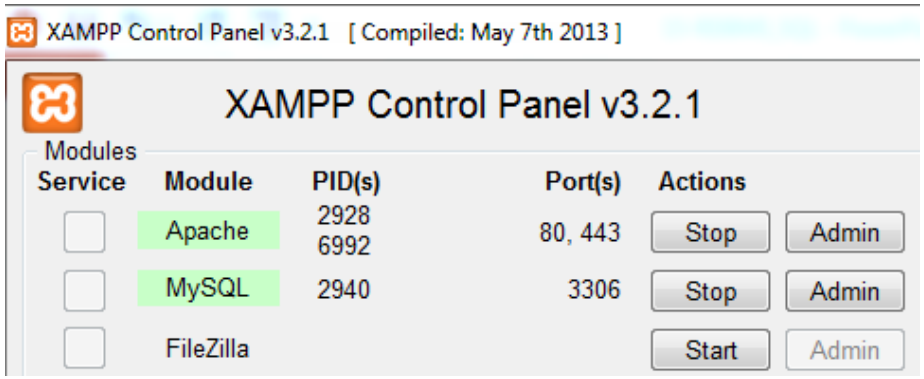
Northwind data base sample

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

- The first 4 records in a table named customers in a data base named Northwind
  - Let us build it together, at least two columns

Creating a Database 1

- Need a data base, we're using XAMPP
- Can start up MariaDB with mysql command
- The server in XAMPP must be running
  - Start XAMPP
  - Start both the Apache and MySQL servers



Creating a Database 2

- Get to the command line, change to the directory with your CSC 337 files, and start MariaDB with mysql

```
c:\xampp\mysql\bin\mysql -u root                    // Windows
```

or

```
/Applications/XAMPP/xamppfiles/bin/mysql -u root    // Mac
```

- **root** has a default password of an empty string, so if prompted for a password, simply press enter
  - Also, never put this on server with real data

Optional: Add to the PATH

- Or you can add mysql to the your path and start MariaDB with `mysql -u root`

Mac or Linux: Edit the file `.bash_profile` in your home folder

```
[Ricks-MBP:~ mercer$ pwd
 /Users/mercer
 Ricks-MBP:~ mercer$ pico .bash_profile
```

Creating a Database 3

c:\xampp\mysql\bin\mysql -u root // Windows
/Applications/XAMPP/xamppfiles/bin/mysql -u root // Mac
Note: Maria DB is case insensitive CREATE==Create CuStoMerS==custOmeRs

- Once in MariaDB (the mysql program), create a database `first` and one table `customer`, add 4 records

```
MariaDB> CREATE DATABASE first;
MariaDB> USE first;
MariaDB> CREATE TABLE customers(ID int, Name varchar(20));
```

- Add four records to table customers in database first

```
MariaDB> insert into customers values (1, 'Maria');
insert into customers values (2, 'Ana');
insert into customers values (3, 'Antonio');
insert into customers values (4, 'Thomas');
```

Reading a Data base 4

- SELECT returns certain rows (* means every row)

```
MariaDB [first]> SELECT * from customers;
+------+---------+
| ID   | Name    |
+------+---------+
|    1 | Maria   |
|    3 | Antonio |
|    2 | Ana     |
|    4 | Thomas  |
+------+---------+
```

Reading a Database 5

```
MariaDB [first]> SELECT ID FROM Customers;
+------+
| ID   |
+------+
|    1 |
|    3 |
|    2 |
|    4 |
+------+

MariaDB [first]> SELECT Name FROM Customers;
+---------+
| Name    |
+---------+
| Maria   |
| Antonio |
| Ana     |
| Thomas  |
+---------+
```

Reading a Database 6

```
MariaDB [first]> SELECT * from Customers where ID < 3;
+------+-------+
| ID   | Name  |
+------+-------+
|    1 | Maria |
|    2 | Ana   |
+------+-------+

MariaDB [first]> SELECT * from Customers where name > "Leo";
+------+--------+
| ID   | Name   |
+------+--------+
|    1 | Maria  |
|    4 | Thomas |
+------+--------+
```

Updating a data base 7

- Add a column to the table named **`City`**
  - Populate that column with any four cities you want
  - One answer, could copy and paste

```
ALTER TABLE customers ADD COLUMN City varchar(33) NOT NULL;
UPDATE Customers SET City = 'Tucson' where NAME = 'Maria';
UPDATE Customers SET City = 'Douglas' where NAME = 'Antonio';
UPDATE Customers SET City = 'Yuma' where NAME = 'Ana';
UPDATE Customers SET City = 'Phoenix' where NAME = 'Thomas';
```

- Confirm Changes

```
MariaDB [first]> SELECT * FROM Customers;
+------+---------+---------+
| ID   | Name    | City    |
+------+---------+---------+
|    1 | Maria   | Tucson  |
|    3 | Antonio | Douglas |
|    2 | Ana     | Yuma    |
|    4 | Thomas  | Phoenix |
+------+---------+---------+
```

Backing up a data base 8

- You need to do the following when turning in future projects and this week's lab
  - Quit MariaDB

```
MariaDB [first]> quit
```

  - Issue the proper command to create a text file with the extension .sql

Windows:

```
c:\xampp\mysql\bin\mysqldump -u root --databases first > first.sql
```

Mac, Unix, or Linux

```
/Applications/XAMPP/xamppfiles/bin/mysqldump  -u root --databases first > first.sql
```

  - Open the file `first.sql` to confirm (or print it to the console with cat)