MySQL Workshop

Scott D. Anderson

Workshop Plan

- Part 1: Simple Queries
  - Database concepts
  - getting started with Cloud 9
  - practical skills using MySQL
  - simple queries using SQL
- Part 2: Creating a database, inserting, updating and deleting data
- Part 3: Joining tables
- Part 4: complex queries with groups, subqueries and sorting
- Reference: https://cs.wellesley.edu/~cs304/mysql-workshop/

Basic Database Concepts

- Lots of things store data. e.g a MS Word file. That's not what we mean
- A *database* stores particular data *efficiently*
  - fast to look up data, particularly using certain ***keys***
  - fast to update/delete data
  - frugal with space
- A Database Management System (DBMS) allows you to create and manage different databases for different purposes
- MySQL is a *Relational* DBMS (RDBMS)

Relational Databases

- data is stored as *rows* in a *table* (relation)
- each row comprises *columns* with different kinds of info
- every row has the same columns

Name	**BID**	**house**	**class year**
Hermione	B123123123	Gryffindor	1998
Ginny	B234234234	Gryffindor	1999

Aside on Representation

Under the hood, databases have an Engine that are a *data structure* that allows fast access where most of the data (rows, AKA records) is on disk, rather than in memory. Examples:

- B-trees
- Hash Tables

Both of these are really cool. You probably already know hash tables. If we have time, we'll talk about B-trees. For next time, read about btrees

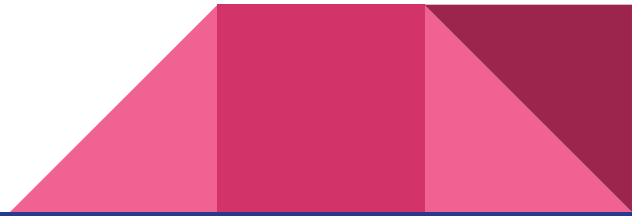
Relational Database properties

- columns are rarely added/removed. rows routinely are.
- each column cell is a single piece of information, not a list.
- So, there's no "list of courses" column for Hermione saying that she's taking ["arithmancy", "potions", "transfiguration", …]
- There can be multiple tables. We'll talk about Hermione's list of courses in part 3

Course	**Prof**	**location**
Potions	Snape	dungeon
Divination	Trelawney	astronomy tower

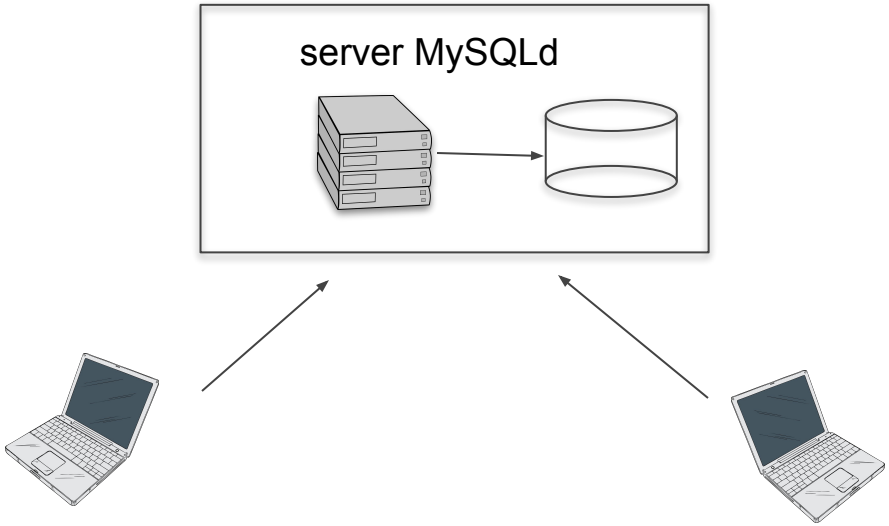
Other Databases

- MySQL is not the only DBMS.
- Microsoft Office has MS Access, which is a desktop RDBMS suitable for a single user.
- There are also non-relational DBMS systems (so-called NoSQL systems) such as MongoDB.
- MySQL is designed to handle *multiple concurrent* users, unlike MS Access



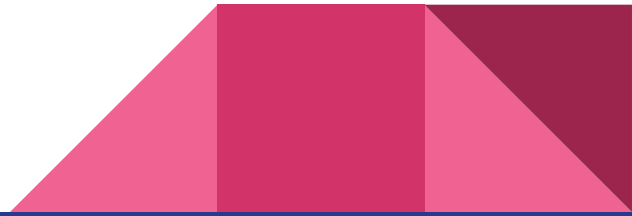
Client-Server Software

- MySQL has a *server* daemon that manages the data and as many *clients* as desired.



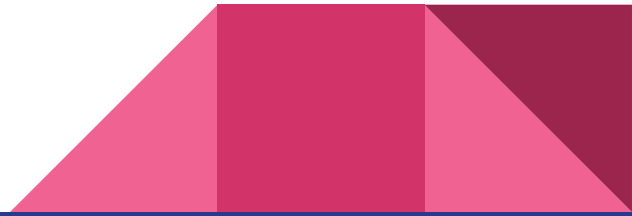
Cloud 9

- Cloud 9 is (now) owned and operated by Amazon (AWS)
- Gives you a (free) virtual server (computer) in the cloud (AWS physical servers) with 1GB of memory and 5GB of disk space
- Logging in gives you a web-based GUI and a command-line
- We'll use the command-line a lot
- For us, both the MySQL server and MySQL clients are in the C9 workspace
- Go ahead and login now….



A Cloud 9 Workspace

- Your C9 account gives you 1 private workspace and as many public workspaces as you want. Each is a virtual server.
- Click the big + to create a workspace.
- Name it something like "mysql-workshop"
- Make it a "Python" workspace (though many others would work)
- Confirm by clicking "create workspace"
- It takes a minute or so to start.

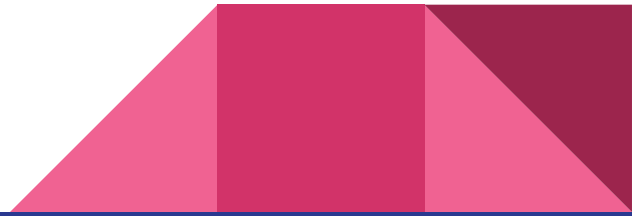


Using Cloud 9

- Notice:
  - file browser
  - menus
  - editor
  - terminals
- In a terminal (AKA the shell), try:

```
$ ls
$ ls ex50
```

Don't type the $. That's the **prompt**



Be a power user of the Shell

- A GUI (Graphical User Interface) is nicer than a shell, but
- A CLI (Command Line Interface) can be more powerful.
- There are many tricks to using it more effectively:

```
$ touch hermione-granger.txt harry-potter.txt ron-weasley.txt
$ ls -l *.txt
$ ls -l h*
$ ls -l he<tab>
$ ls -l h<tab><tab>
$ rm *.txt
```

`^p` or <up-arrow> for prior commands

Starting and stopping MySQL

In a terminal, follow me as I do the following (don't type the prompts):

```
$ mysql-ctl start
$ mysql-ctl cli
mysql> help
mysql> select user();
mysql> quit
$ mysql-ctl stop
```

Don't type the `mysql>` ; that's the **prompt**

Getting some data: Curl

The C9 servers have **`curl`**, which is a program that will get a file from a web address. (There's also **`wget`**, which does almost the same thing.) Use capital Oh, not a zero, as the command-line switch

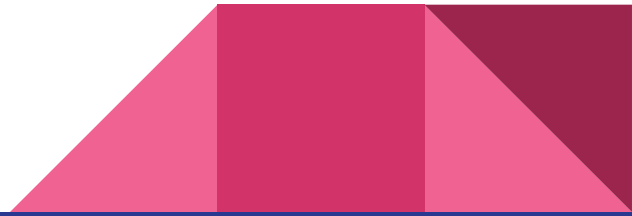
```
$ curl -O https://cs.wellesley.edu/~cs304/downloads/wmdb.sql
$ mysql-ctl cli
mysql> show databases;
mysql> source wmdb.sql;
mysql> show databases;
mysql> use wmdb;
mysql> show tables;
mysql> quit;
```

The Wellesley Movie Database

- The database we will be playing with is based on my favorite database, namely the Internet Movie Database (IMDB)
- There are pages for actors, pages for movies, and links between them.
- The WMDB has been built by student contributions over many offerings of CS304. The data is not guaranteed.
- Here are the IMDB pages for George Clooney and Gravity

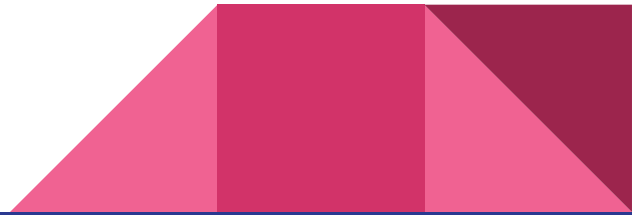
Practical use of the MySQL client

- commands often end in a semi-colon
- commands that don't finish can be continued on the next line
- if you messed up a previous line, just end the command with a semi-colon, ignore the error, and go on
- You can re-use a prior line by hitting up-arrow or control-p. You can do this in the shell, too. You can edit the prior command before submitting it.
- In a pinch, you can do control-c which will kill the client (not the server) and you can start over
- You can quit the client with control-d
- Try these!



SQL versus a GUI

- Data is pulled out of a database using a *query*
- SQL is the Structured Query Language. It's an industry standard, but with vendor variants.
- We'll learn SQL, rather than a GUI, for our queries. If you want a GUI for your queries, MS Access has a pretty good one, called Query By Example (QBE)
- There are many online tutorials. Feel free to use one to supplement this workshop.
- Excellent reference: https://dev.mysql.com/doc/refman/5.5/en/tutorial.html



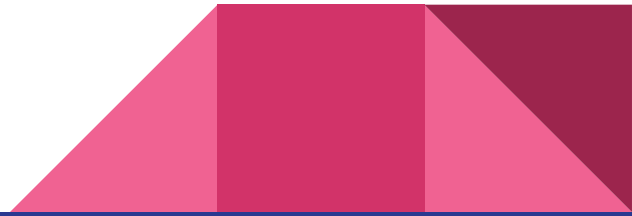
Why not a GUI?

I like GUIs as much as most people. GUIs have many advantages:

- menus remind you of your options: no memorization
- typos are impossible
- syntax is guided; much harder to make mistakes

So, why not a GUI?

- Can't be written down and automated
- Can't be programmed



First Queries

Try the following:

```
mysql> use wmdb;
mysql> select title from movie limit 10;
mysql> select name from person limit 10;
mysql> select name,birthdate from person limit 10;
mysql> select * from person limit 10;
```

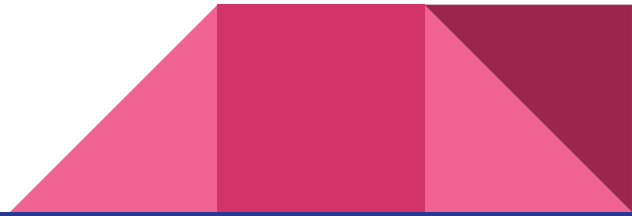
The NM is the ID of the person, just like IMDB: George Clooney

Queries: the SELECT statement

```
SELECT col1, col2, … or *
FROM table
WHERE boolean expression using cols, functions and constants
LIMIT number of rows;
```

Last two clauses optional, but statements always end with a semi-colon.

Refer to the workshop webpage for a link to the main reference



Example queries

Do this with me:

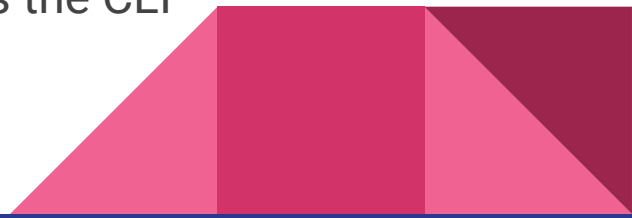
```
$ curl -O https://cs.wellesley.edu/~cs304/mysql-workshop/part1.tar
$ tar xf part1.tar
$ cd part1
$ ls
$ more wmdb1-all.sql
mysql> source wmdb1-all.sql;
$ mysql-ctl cli < wmdb1-all.sql
```

Batch files

- the `wmdb1-all.sql` file records a particular query
- we can run it from the mysql client by using the `source` command
- we can run it from the terminal command line by using a Unix trick of running a command and re-directing its input to be from a file instead of from the terminal. That's what the angle bracket does:

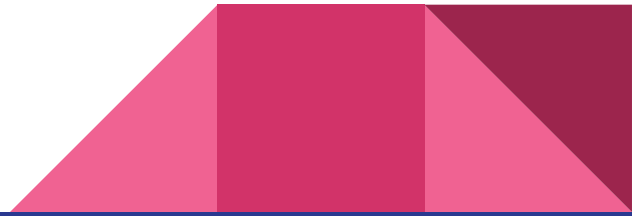
```
$ mysql-ctl cli < wmdb1-all.sql
```

- the `mysql` command w/o the input redirection starts the CLI



More batch files

- double-click on a batch file in the GUI to see it
- edit the file to put the columns in a different order or change the limit
- save the file,
- re-run it in the terminal. Use command history to avoid re-typing!

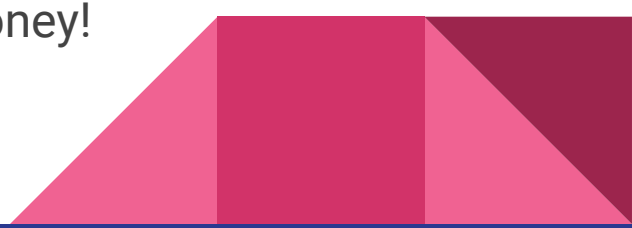


the WHERE clause

- the WHERE clause contains a *boolean* expression, which just means that the expression is either true or false, so:
  - x > 3 is boolean
  - x + 3 is not boolean
- only rows where the boolean expression is *true* are returned (printed)
- To print out all info on George Clooney, we do:

```
select * from person where name = 'George Clooney';
```

- Note that there might be more than one George Clooney!



People's names are not unique

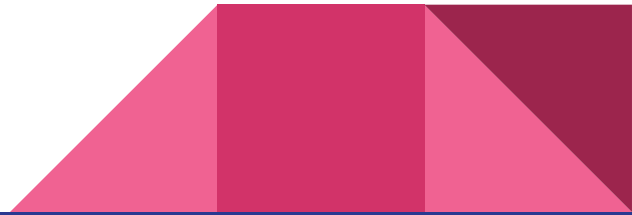
- I could tell you stories of my name …
- IMDB.com lists 3 people named "Mary Moore" (it adds roman numerals)
- Mary Tyler Moore added her middle name because "there were half a dozen other Mary Moores registered" with SAG
- If WMDB had multiple people named "George Clooney", all would be reported.
- We'll see examples of multiple matches very soon.
- That's why the IMDB and the WMDB use an ID for each person, which is the NM value. George Clooney is 123.

Examples of WHERE clauses

Look for these in wmdb[2-8]*.sql:

```
where name = 'George Clooney';
where nm = 123;
where birthdate = '1961-05-06';
where year(birthdate) = '1961';
where month(birthdate) = 5;
where dayofweek(birthdate) = 7;
where year(birthdate) = 1961 and month(birthdate) = 5;
```

Try your own variants. You can either modify the file or type directly to the MySQL CLI.



Boolean Expressions

- boolean expressions are true/false
- boolean expressions can be combined with AND, OR and NOT:

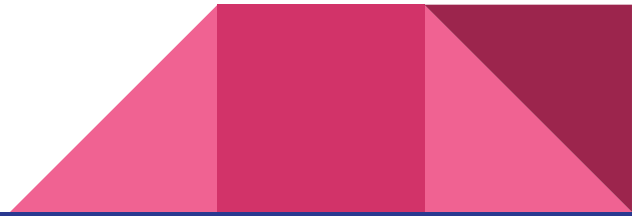
```
… where B1 and B2;
… where B2 or B2;
… where not B2;
```

This is fairly intuitive but is complex and can get out of hand.

Try it!

Try some complex boolean expressions:

- someone born in May of 1961
- someone born in 1961 or 1962
- someone born in May or June of 1961
- someone born in 1961 but not in December



Complex Boolean expressions

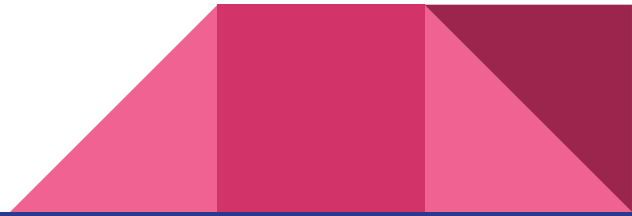
```
where year(birthdate) = 1961 and month(birthdate) = 5

where year(birthdate) = 1961 or year(birthdate) = 1962

where year(birthdate) = 1961 and (month(birthdate) = 5 or
month(birthdate) = 6)
```

Parentheses are crucial!

More Solutions

where year(birthdate) = 1961 and not month(birthdate) = 12

where year(birthdate) = 1961 and month(birthdate) <> 12



MySQL Date Functions

- in the previous examples, we used functions like month(), year() and dayofweek() to take apart a date.
- there are many date functions defined in MySQL, including:
  - ways to subtract two dates to get a time interval,
  - to add a time interval to a date to get a different date
  - format a date in a variety of ways
- See the online documentation for more, if you are interested

Summary of Part 1

- MySQL, like all relational databases, is built from **tables**.
- Each table has one or more **columns**.
- A **query** finds all the rows for which a boolean expression in the **WHERE** clause is true.
- A query can specify what columns are chosen
- A query can be typed
  - directly into the MySQL **shell** (**CLI**) for immediate execution or
  - saved in a **batch file** for debugging and execution when wanted.