MySQL
Not your SQL, understand?

MySQL
- Relational Database Management System
  - RDBMS
- Stores stuff in Tables
- Tables have named columns
- Tables have multiple rows with the same columns for each row
- Tables can be related to each other

Connecting
- AWS VM
- From your command line:
  
  $ mysql
Databases

• `show databases;`
• Lists all the databases on this server
• `use <database>;`
• Select a database to send commands to

Looking At Tables

• `show tables;`
• Lists all tables in the database
• `describe <tablename>;`
• Print out the column structure of the given table

SQL

• SQL - Structured Query Language
• An english like syntax to interact with a databases
• Basic Verbs initiate Commands
  • SELECT
  • INSERT
  • UPDATE
  • DELETE
CREATE TABLE

• Make a new table to hold stuff
• Think about the columns you want to have in your table
• Data Modeling

CREATE TABLE `staff` (  
`id` int(11) NOT NULL auto_increment,  
`name` varchar(1024) default NULL,  
`phone` varchar(1024) default NULL,  
`email` varchar(1024) default NULL,  
PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

MySQL Datatypes


• Several ways to hold a string  
  • CHAR and VARCHAR  
  • Also BLOB and TEXT  
• Numbers
  • INT, SMALLINT, BIGINT etc  
  • DECIMAL, NUMERIC, FLOAT, DOUBLE, BIT  
• Dates & Times
  • DATE, TIME, TIMESTAMP, DATETIME

CRUD

• You’ll hear people mention CRUD in connection with databases
  • Create
  • Retrieve
  • Update
  • Delete
SQL

<table>
<thead>
<tr>
<th>CRUD</th>
<th>SQL Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>INSERT</td>
</tr>
<tr>
<td>Retrieve</td>
<td>SELECT</td>
</tr>
<tr>
<td>Update</td>
<td>UPDATE</td>
</tr>
<tr>
<td>Delete</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

Select

- Getting data out of tables

```
SELECT <fields> FROM <tables> [WHERE <conditions>];
```

- SQL is case insensitive
- These all work the same
- The Asterisk '* ' means “All the fields in the tables”
- Can select just specific fields by specifying which ones

```
select * from staff;
SELECT * FROM staff;
Select * From Staff;
```

```
mysql> select name, email from staff;
+----+---------+
<table>
<thead>
<tr>
<th>name</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mark</td>
<td><a href="mailto:mark@cs.berkeley.edu">mark@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>Matt</td>
<td><a href="mailto:matt@cs.berkeley.edu">matt@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>Jane</td>
<td><a href="mailto:jane@cs.berkeley.edu">jane@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>Tracy</td>
<td><a href="mailto:tracy@cs.berkeley.edu">tracy@cs.berkeley.edu</a></td>
</tr>
</tbody>
</table>
```

```
mysql> select * from staff;
+----+---------+------+
<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mark</td>
<td><a href="mailto:mark@cs.berkeley.edu">mark@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>2</td>
<td>Matt</td>
<td><a href="mailto:matt@cs.berkeley.edu">matt@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>3</td>
<td>Jane</td>
<td><a href="mailto:jane@cs.berkeley.edu">jane@cs.berkeley.edu</a></td>
</tr>
<tr>
<td>4</td>
<td>Tracy</td>
<td><a href="mailto:tracy@cs.berkeley.edu">tracy@cs.berkeley.edu</a></td>
</tr>
</tbody>
</table>
```
selecting specific things

• The WHERE clause for a SELECT statement allows us to limit the rows selected from a set of tables

```sql
SELECT * FROM staff WHERE name='Mark';
```

![SQL query output]

selecting specific things

• Doesn’t have to be an exact match – LIKE

• % is our wildcard match character for strings in SQL

```sql
SELECT * FROM staff WHERE name LIKE 'M%';
```

![SQL query output]

insert

• Adding new rows to a table

• Values must match positions with their field names

• Values must be correct for the datatype of the field

• Strings must be surrounded by single quotes – 'some string'

```sql
INSERT INTO <table> (field1, field2, ...) VALUES (value1, value2, ...);
```
Why didn't we specify the id field?

Where does the 8 come from?

**AUTO INCREMENT**

- When defining a table, you can specify a **PRIMARY KEY** field be **AUTO INCREMENT**
- This does pretty much what it sounds like
- Anytime a new row is inserted into the table, MySQL will automatically assign a new value, incrementing an internal counter
update
• Change a value for a field or set of fields.

\[
\text{UPDATE<table> SET field1=value1, field2=value2 WHERE [conditions];}
\]

• WATCH OUT!
• If you don’t specify any conditions, you will update EVERY ROW!

update

\[
\text{UPDATE staff SET phone='626-TECH' WHERE id=1;}
\]

delete
• Deletes rows from a table

\[
\text{DELETE FROM <table> WHERE [conditions];}
\]

• WATCH OUT!
• If you don’t specify any conditions, you will DELETE EVERY ROW!
Joins

- The Relational part of RDBMS
Joins

• You can SELECT from multiple tables in a single query

```sql
SELECT games.game_state,
games.game_id,
players_games.player_id
FROM players_games INNER JOIN games
ON players_games.game_id = games.game_id;
```

Joins

• When specifying fields to select from multiple tables, you prefix the field name by the table name

• `tablename.fieldname`

```sql
SELECT games.game_state,
games.game_id,
players_games.player_id
...  
```

Table Name  Field Name

```sql
mysql> SELECT games.game_state,
games.game_id,
players_games.player_id
FROM players_games INNER JOIN games
ON players_games.game_id = games.game_id;
```

3 rows in set (0.00 sec)

mysql>
Lots Of Other Stuff

- Lots of built-in functions
  - ABS, AVG, POW, RAND, SYSDATE, VARIANCE
- Standard Operators
  - + - / * = > etc
- Stored Procedures
  - Write your code directly in the database, then make SQL calls to the functions
  - Can store JSON natively now
- Transactions

Great!

Now go do all that from PHP!