Lecture 17

MySQL
Tables and Relations 101

* Course logo spider web photograph from Morguefile openstock photograph by Gabor Karpati, Hungary.
Review – Starting Servers

Keep in mind MySQL works through a client-server interface to a separate database server process/machine.

```
> /Applications/MAMP/Library/bin/mysql -u ross -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/Applications/MAMP/tmp/mysql/mysql.sock' (2)
```
Data - Tables

- Table is, more or less, a flat database.
- Contains “things”, objects.
- One thing per row.
- Columns store attributes.
- Primary index unique for each row.
- Relations kick in later
  - How we relate tables to one another.
Exercise Some Commands

- use
- CREATE
- INSERT
- SELECT
- UPDATE
- DROP

<table>
<thead>
<tr>
<th>cd_id</th>
<th>name</th>
<th>artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mudlark</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>2</td>
<td>My Feet Are Smiling</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>3</td>
<td>Lucky The Man</td>
<td>Wizz Jones</td>
</tr>
<tr>
<td>4</td>
<td>The Blue Idol</td>
<td>Altan</td>
</tr>
<tr>
<td>5</td>
<td>Return of the Repressed</td>
<td>John Fahey</td>
</tr>
</tbody>
</table>
Create Music Database

In general, new databases do not come and go often.
Example 1

```
use music;

DROP TABLE IF EXISTS cds;

CREATE TABLE cds (  
  cd_id int(5) NOT NULL,  
  name varchar(50),  
  artist varchar(50),  
  PRIMARY KEY (cd_id)  
);

INSERT INTO cds VALUES  
(1, 'Mudlark', 'Leo Kottke'),  
(2, 'My Feet Are Smiling', 'Leo Kottke'),  
(3, 'Lucky The Man', 'Wizz Jones'),  
(4, 'The Blue Idol', 'Altan');

SELECT name, artist FROM cds;

INSERT INTO cds VALUES  
(5, 'Return of the Repressed', 'Leo Kottke');

SELECT name, artist FROM cds WHERE cd_id = 5;

UPDATE cds SET artist = 'John Fahey' WHERE cd_id = 5;

SELECT name, artist FROM cds;
```

Terminal — mysql -u ross -p — 54×44

```
mysql> source makeMusic1.sql;
Database changed
Query OK, 0 rows affected, 1 warning (0.00 sec)
Query OK, 0 rows affected (0.02 sec)
Query OK, 4 rows affected (0.00 sec)
Records: 4 Duplicates: 0 Warnings: 0

<table>
<thead>
<tr>
<th>name</th>
<th>artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mudlark</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>My Feet Are Smiling</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>Lucky The Man</td>
<td>Wizz Jones</td>
</tr>
<tr>
<td>The Blue Idol</td>
<td>Altan</td>
</tr>
</tbody>
</table>

4 rows in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)

```

<table>
<thead>
<tr>
<th>name</th>
<th>artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return of the Repressed</td>
<td>Leo Kottke</td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```

<table>
<thead>
<tr>
<th>name</th>
<th>artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mudlark</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>My Feet Are Smiling</td>
<td>Leo Kottke</td>
</tr>
<tr>
<td>Lucky The Man</td>
<td>Wizz Jones</td>
</tr>
<tr>
<td>The Blue Idol</td>
<td>Altan</td>
</tr>
<tr>
<td>Return of the Repressed</td>
<td>John Fahey</td>
</tr>
</tbody>
</table>

5 rows in set (0.00 sec)

mysql>
```
Where’s the Beef, ah ... Data.

- Data from tables is stored in files.
- Case sensitivity comes from host OS.
- In case it is not obvious.
  - **NEVER MANIPULATE THESE FILES DIRECTLY.**
Leo Kottke appears twice.

- **Leo Kottke**
  - Mudlark

- **Wizz Jones**
  - Lucky The Man
  - The Blue Idol

- **Altan**

- **John Fahey**
  - Return of the Repressed
  - My Feet Are Smiling
Example 2

```sql
use music;
DROP TABLE IF EXISTS cd;
DROP TABLE IF EXISTS artist;

CREATE TABLE artist (  
    artist_id int(5) NOT NULL,  
    artist_name varchar(50),  
    PRIMARY KEY (artist_id)  
);

CREATE TABLE cd (  
    cd_id int(5) NOT NULL,  
    cd_name varchar(50),  
    artist_id int(5) NOT NULL,  
    PRIMARY KEY (cd_id),  
    FOREIGN KEY (artist_id) REFERENCES artist(artist_id)  
);

INSERT INTO artist VALUES (1, 'Leo Kottke');
INSERT INTO artist VALUES (2, 'Wizz Jones');
INSERT INTO artist VALUES (3, 'Altan');
INSERT INTO artist VALUES (4, 'John Fahey');

INSERT INTO cd VALUES (1, 'Mudlark', 1);
INSERT INTO cd VALUES (2, 'My Feet Are Smiling', 1);
INSERT INTO cd VALUES (3, 'Lucky The Man', 2);
INSERT INTO cd VALUES (4, 'The Blue Idol', 3);
INSERT INTO cd VALUES (5, 'Return of the Repressed', 4);

SELECT cd_name, artist_name FROM cd NATURAL JOIN artist;
```

Output:

```
+----------------+---------------+
| cd_name        | artist_name   |
|----------------+---------------+
| Mudlark        | Leo Kottke    |
| My Feet Are Smiling | Leo Kottke   |
| Lucky The Man  | Wizz Jones    |
| The Blue Idol  | Altan         |
| Return of the Repressed | John Fahey |

5 rows in set (0.00 sec)
```
### Keys: Primary and Foreign

**Primary Key Unique!**

Establishes Relation: Loosely speaking, a ‘pointer’ to the other object.

```sql
CREATE TABLE artist (
    artist_id int(5) NOT NULL,
    artist_name varchar(50),
    PRIMARY KEY (artist_id)
);

CREATE TABLE cd (
    cd_id int(5) NOT NULL,
    cd_name varchar(50),
    artist_id int(5) NOT NULL,
    PRIMARY KEY (cd_id),
    FOREIGN KEY (artist_id) REFERENCES artist(artist_id)
);

INSERT INTO artist VALUES (1, 'Leo Kottke');
INSERT INTO artist VALUES (2, 'Wizz Jones');
INSERT INTO artist VALUES (3, 'Altan');
INSERT INTO artist VALUES (4, 'John Fahey');

INSERT INTO cd VALUES (1, 'Mudlark', 1);
INSERT INTO cd VALUES (2, 'My Feet Are Smiling', 1);
INSERT INTO cd VALUES (3, 'Lucky The Map!', 2);
```

Two cds related to one artist.