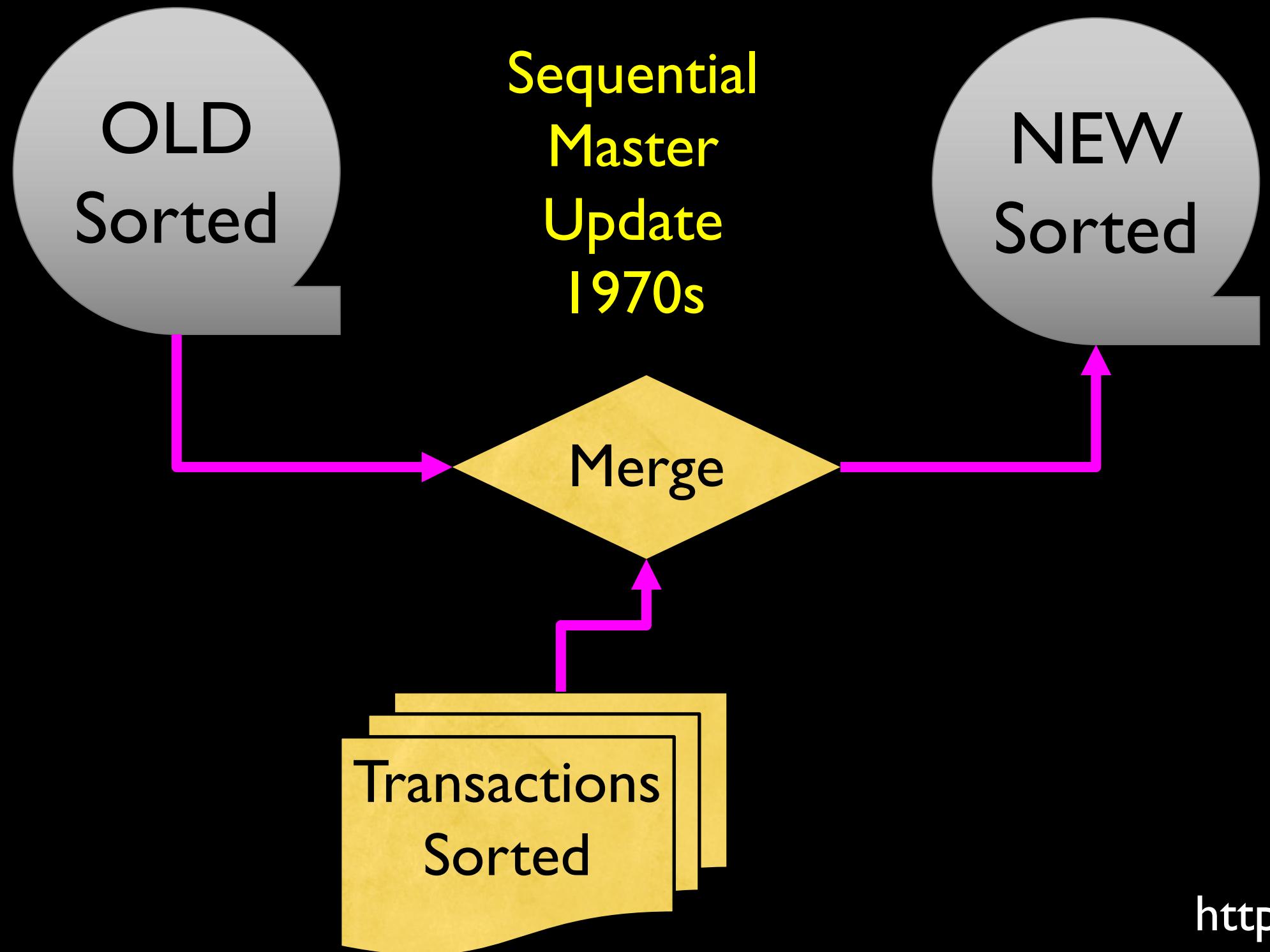


Relational Databases and MySQL

Charles Severance
www.wa4e.com





https://en.wikipedia.org/wiki/IBM_729

Random Access

- When you can randomly access data...
- How can you lay out data to be most efficient?
- Sorting might not be the best idea



https://en.wikipedia.org/wiki/Hard_disk_drive_platter

Relational Databases

Relational databases model data by storing rows and columns in tables. The power of the relational database lies in its ability to efficiently retrieve data from those tables - in particular, where the query involves multiple tables and the relationships between those tables.

http://en.wikipedia.org/wiki/Relational_database

Structured Query Language

- Structured Query Language (SQL) came out of a government / industry partnership
- National Institute of Standards and Technology (NIST)

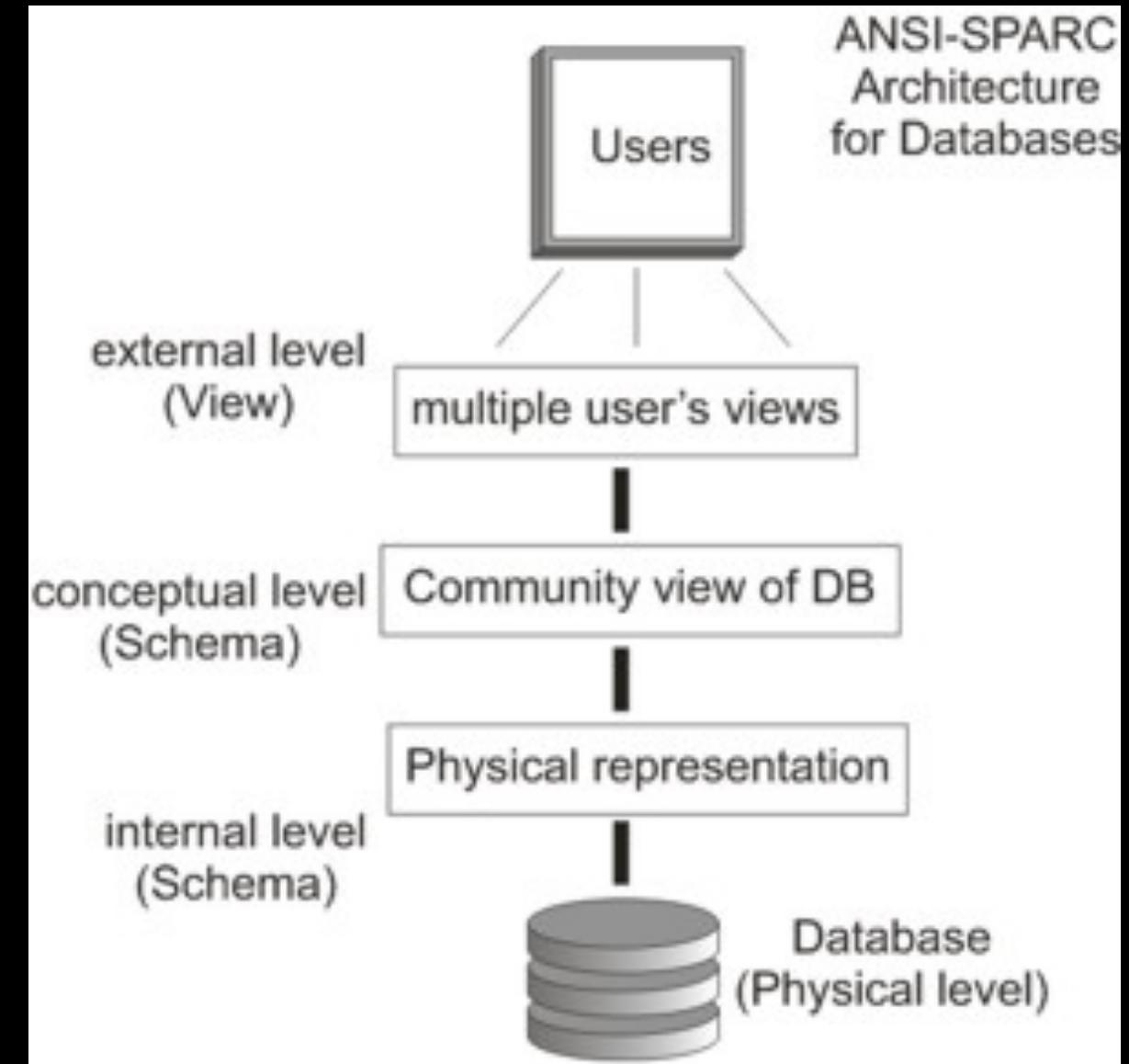


<https://youtu.be/rLUM3vst87g>

SQL

Structured Query Language is the language we use to issue commands to the database

- Create/Insert data
- Read>Select some data
- Update data
- Delete data

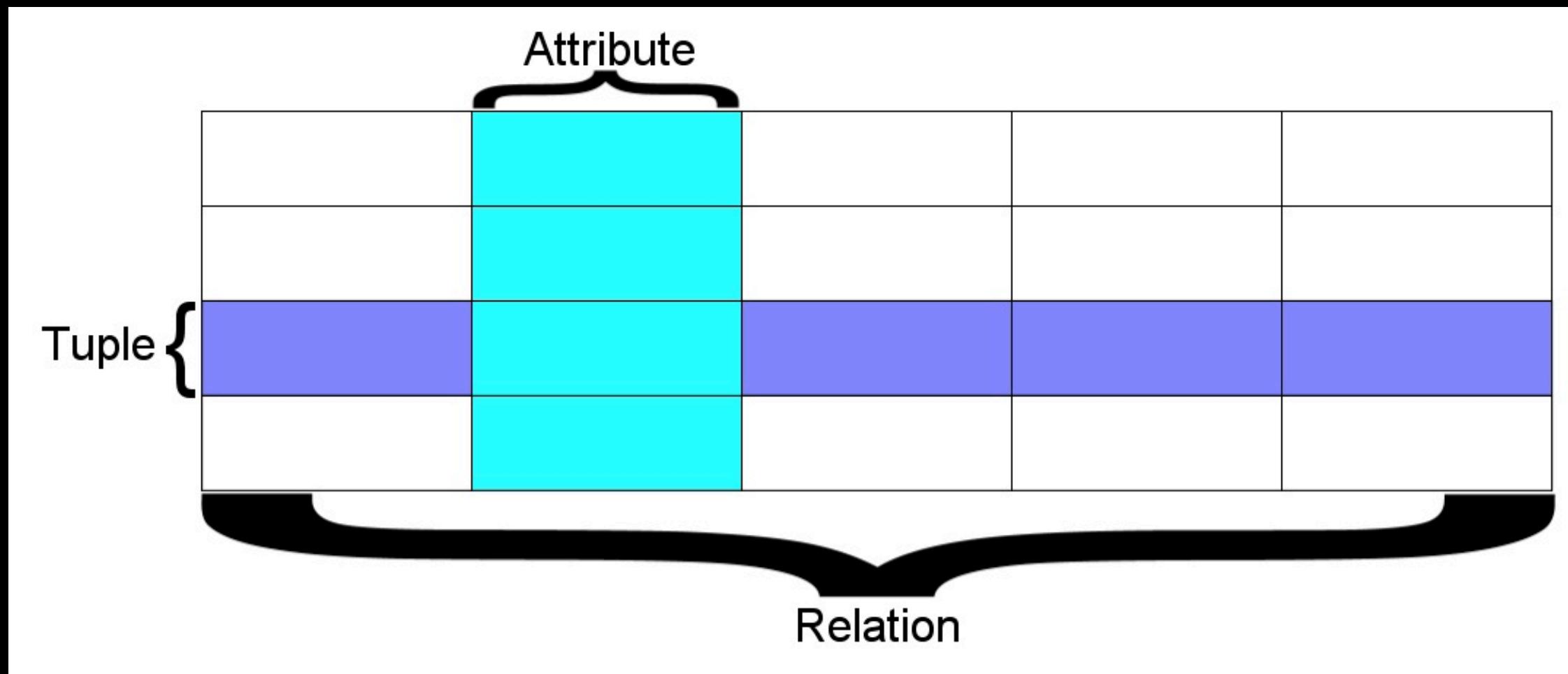


<http://en.wikipedia.org/wiki/SQL>

https://en.wikipedia.org/wiki/ANSI-SPARC_Architecture

Terminology

- **Database** - contains one or more tables
- **Relation (or table)** - contains tuples and attributes
- **Tuple (or row)** - a set of fields which generally represent an “object” like a person or a music track
- **Attribute (also column or field)** - one of possibly many elements of data corresponding to the object represented by the row



A **relation** is defined as a **set of tuples** that have the same **attributes**. A **tuple** usually represents an **object** and **information about that object**. **Objects** are typically **physical objects or concepts**. A **relation** is usually described as a **table**, which is **organized into rows and columns**. All the data referenced by an **attribute** are in the same **domain** and **conform to the same constraints**.

(wikipedia)

SI502 - Database

New Open Save Print Import : Copy Paste Format : Undo Redo : AutoSum Sort A-Z Sort Z-A : Gallery Toolbox

Sheets Charts SmartArt Graphics WordArt

A B C D

1 2 3 4 5 6 7 8

Columns / Attributes

	TITLE	RATING	LEN	
1	About to Rock		3	354
2	Who Made Who		4	252
3				
4				
5				
6				
7				
8				

Rows / Tuples

Tables / Relations

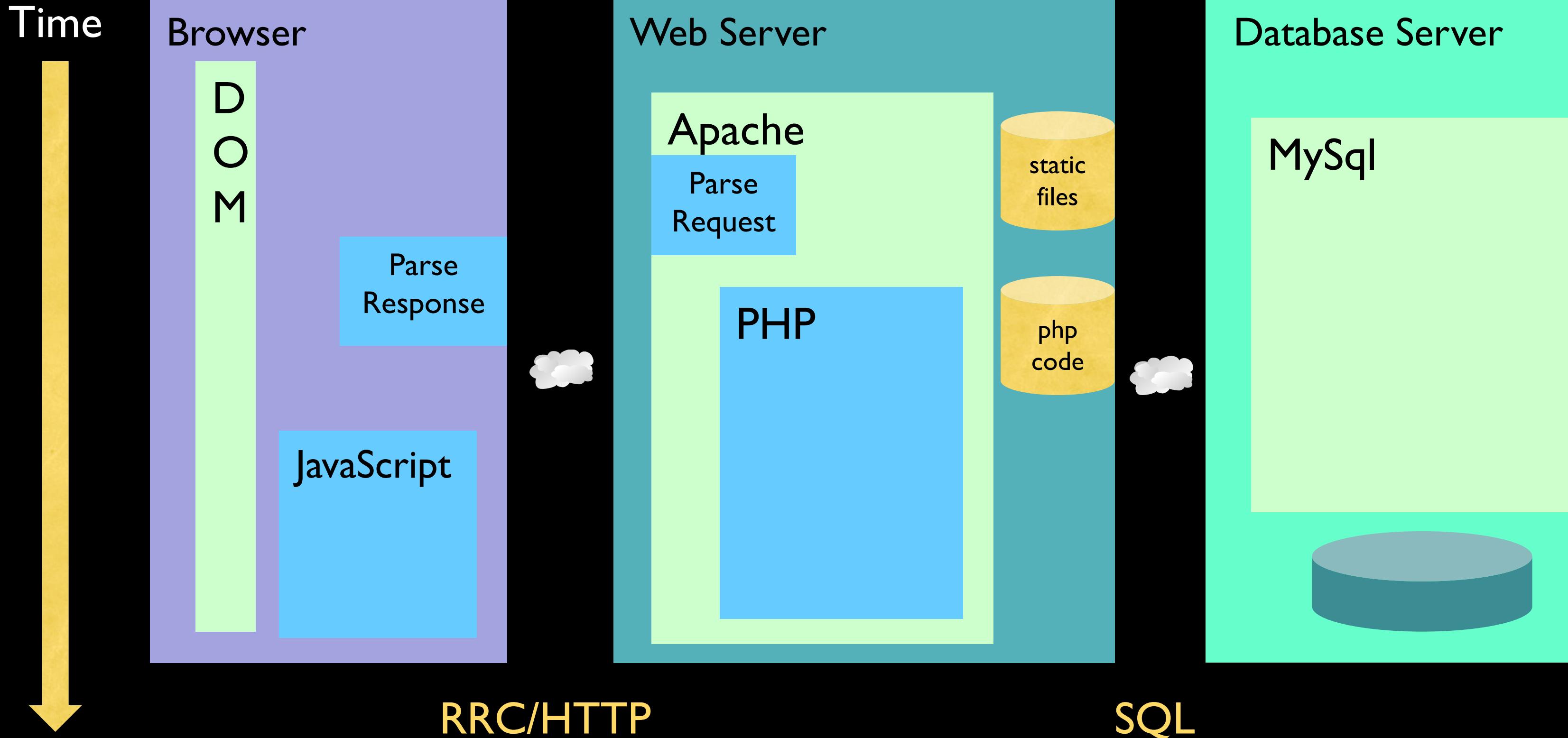
Tracks Albums Artists Genres +

Common Database Systems

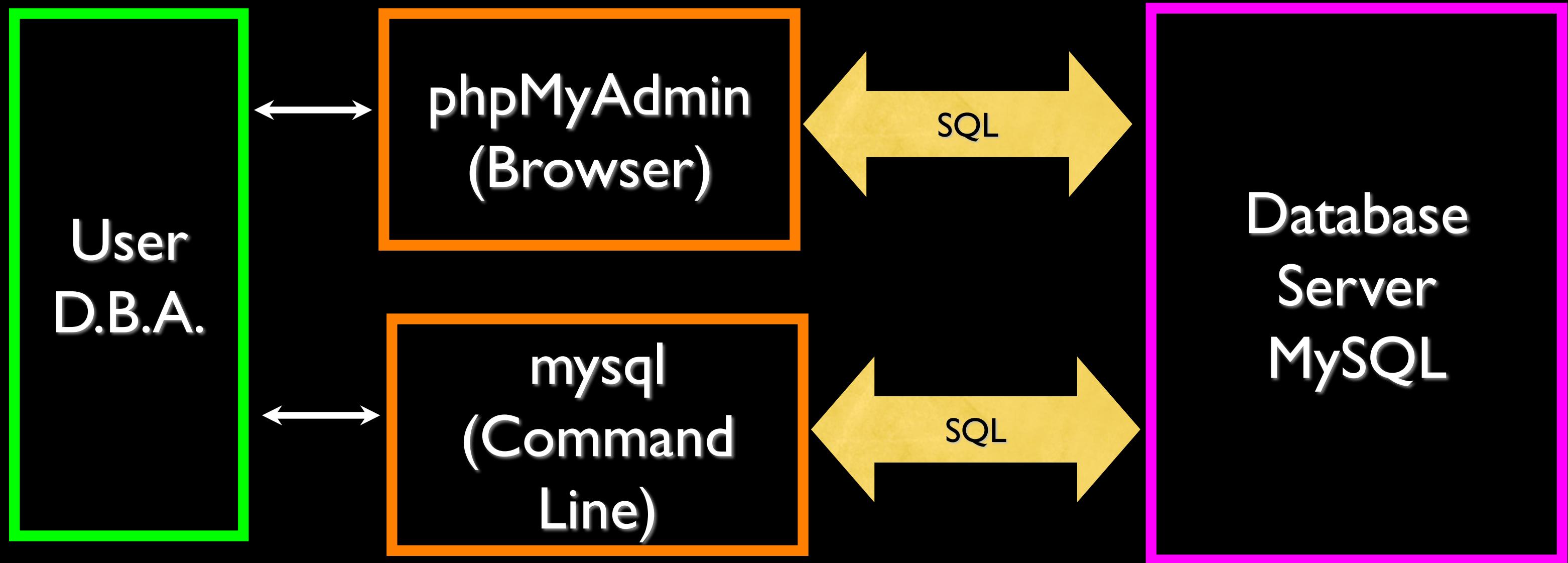
- Three major Database Management Systems in wide use
 - **Oracle** - Large, commercial, enterprise-scale, very tweakable
 - **MySQL** - Simpler but very fast and scalable - commercial open source
 - **SqlServer** - Very nice - from Microsoft (also Access)
- Many other smaller projects, free and open source
 - HSQL, SQLite, PostgreSQL ...



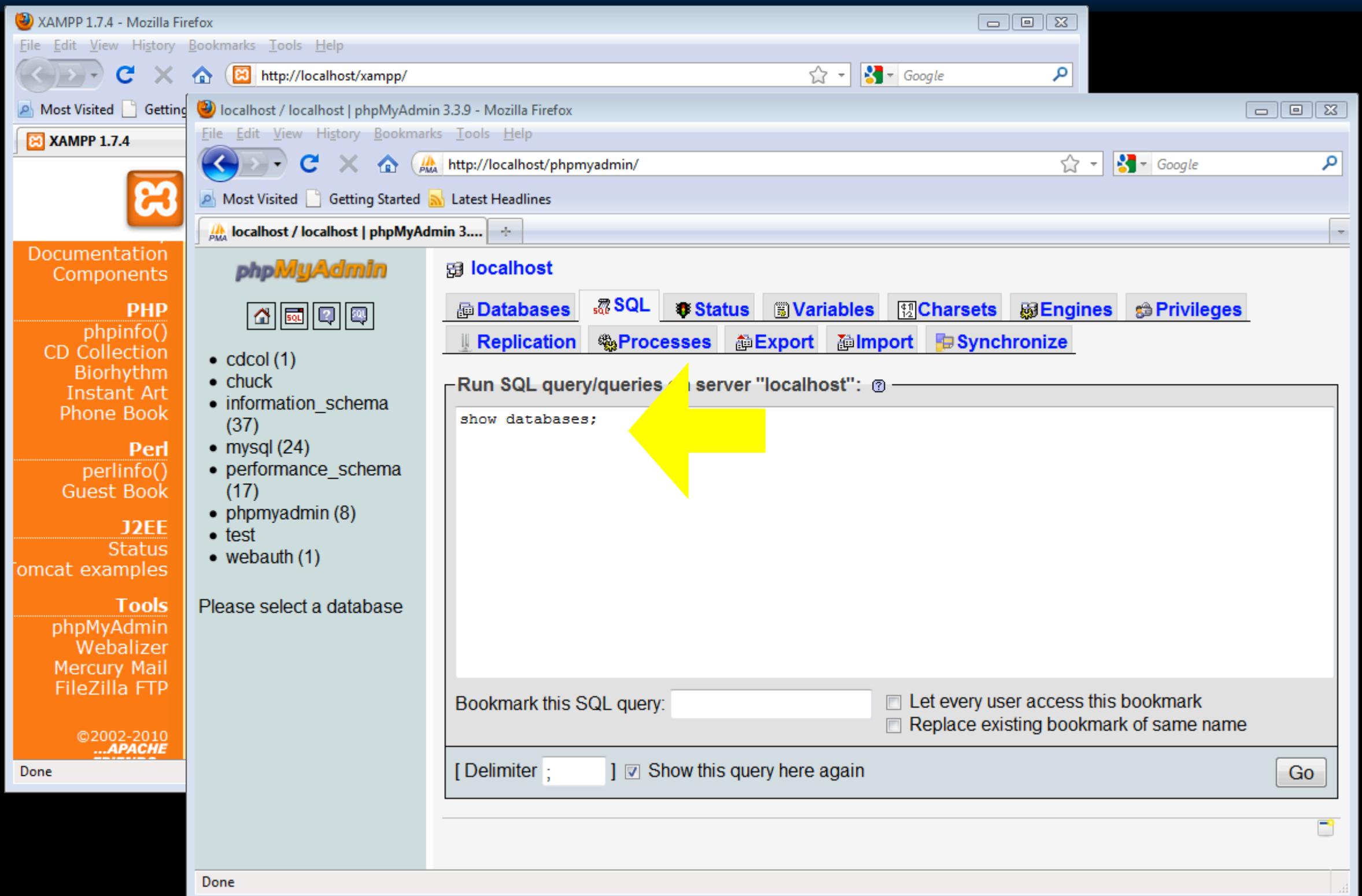
Basic SQL Operations

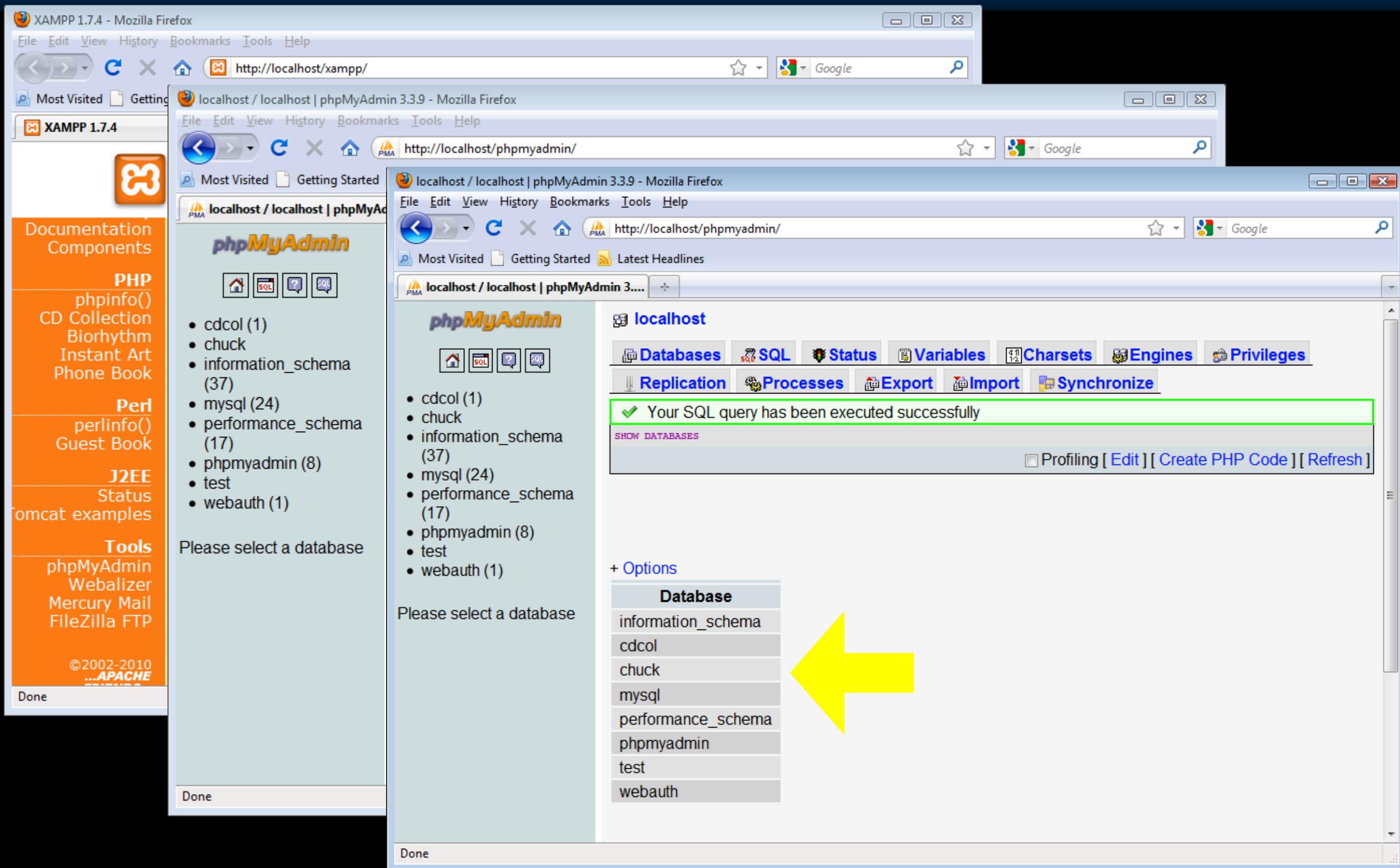


Using SQL









Command Line

After Control Panel is running...

- Macintosh

- `/Applications/MAMP/Library/bin/mysql --host=localhost -uroot -p`
- Enter "root" when prompted for the password

- Windows

- `c:\xampp\mysql\bin\mysql.exe -u root -p`
- Press enter when prompted for password

```
Administrator: C:\Windows\system32\cmd.exe - c:\xampp\mysql\bin\mysql.exe -u root
C:\Users\Administrator>c:\xampp\mysql\bin\mysql.exe -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.8 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
Administrator: C:\Windows\system32\cmd.exe - c:\xampp\mysql\bin\mysql.exe -u root
C:\Users\Administrator>c:\xampp\mysql\bin\mysql.exe -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.8 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

```
cseverancembp:~ csev$ /Applications/MAMP/Library/bin/mysql --host=localhost -uroot -proot
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.25 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```



Your First MySQL Command

show databases

Kind of like `print('hello world')`

```
mysql> show databases;
+-----+-----+
| Database |           |
+-----+-----+
| information_schema | ←
| cdcol | ←
| chuck | ←
| mysql |           |
| sakai |           |
| test |           |
+-----+
6 rows in set (0.06 sec)

mysql> 
```

If this does not work, stop and figure out why.

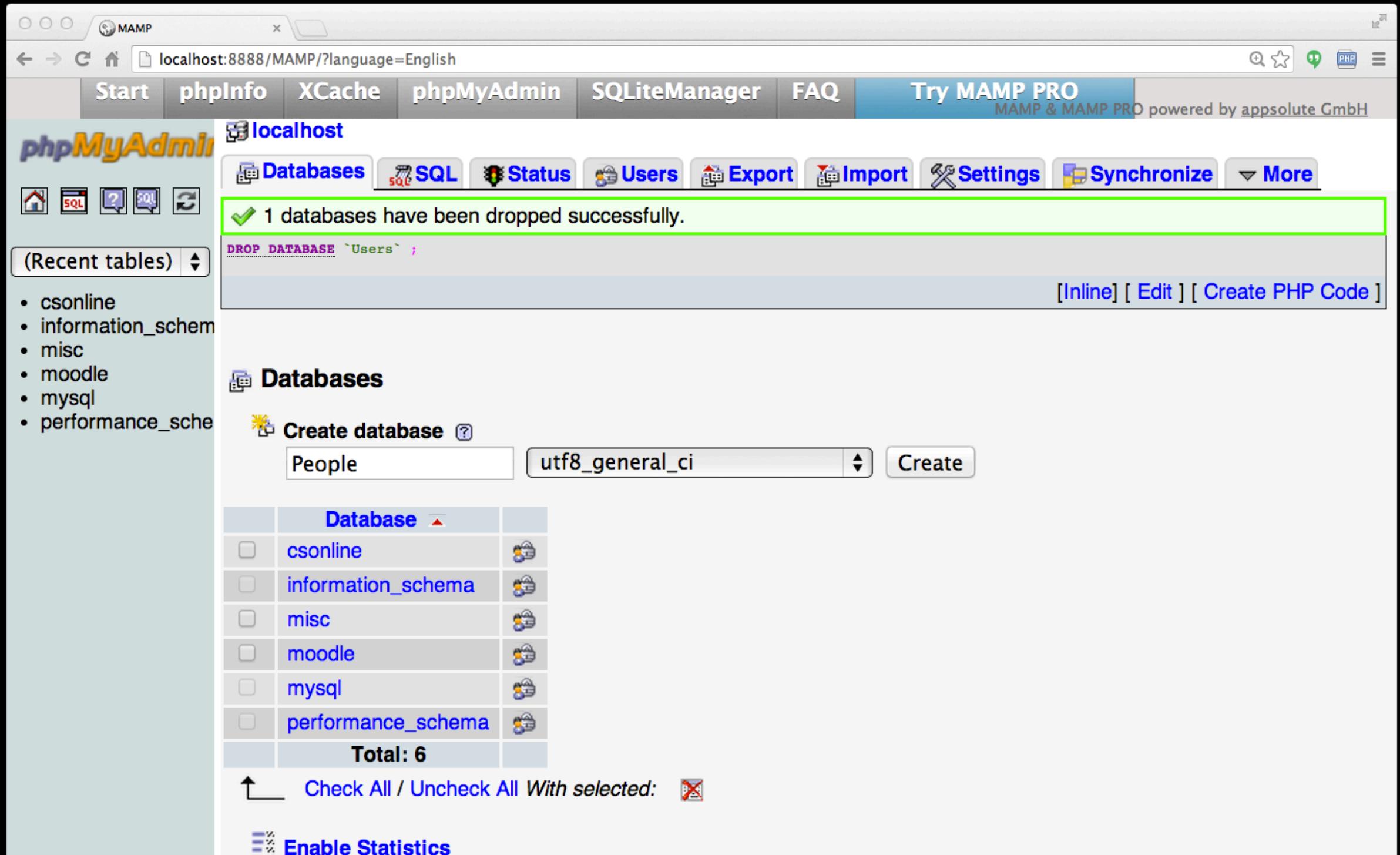
Some of these are part of MySQL and store internal data - don't mess with them.



Creating a Database

Command Line:

```
CREATE DATABASE People;  
USE People;
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The top navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The main menu bar for phpMyAdmin includes Database, SQL, Status, Users, Export, Import, Settings, Synchronize, and More.

A success message in a green box states: "1 databases have been dropped successfully." Below it is the SQL command: `DROP DATABASE `Users`;` with options to [Inline], [Edit], or [Create PHP Code].

The left sidebar lists recent tables: csonline, information_schema, misc, moodle, mysql, and performance_schema.

The main content area shows the "Databases" section. It includes a "Create database" form with "People" selected as the database name and "utf8_general_ci" as the character set. A "Create" button is present. A table lists existing databases: csonline, information_schema, misc, moodle, mysql, and performance_schema. The total count is displayed as "Total: 6".

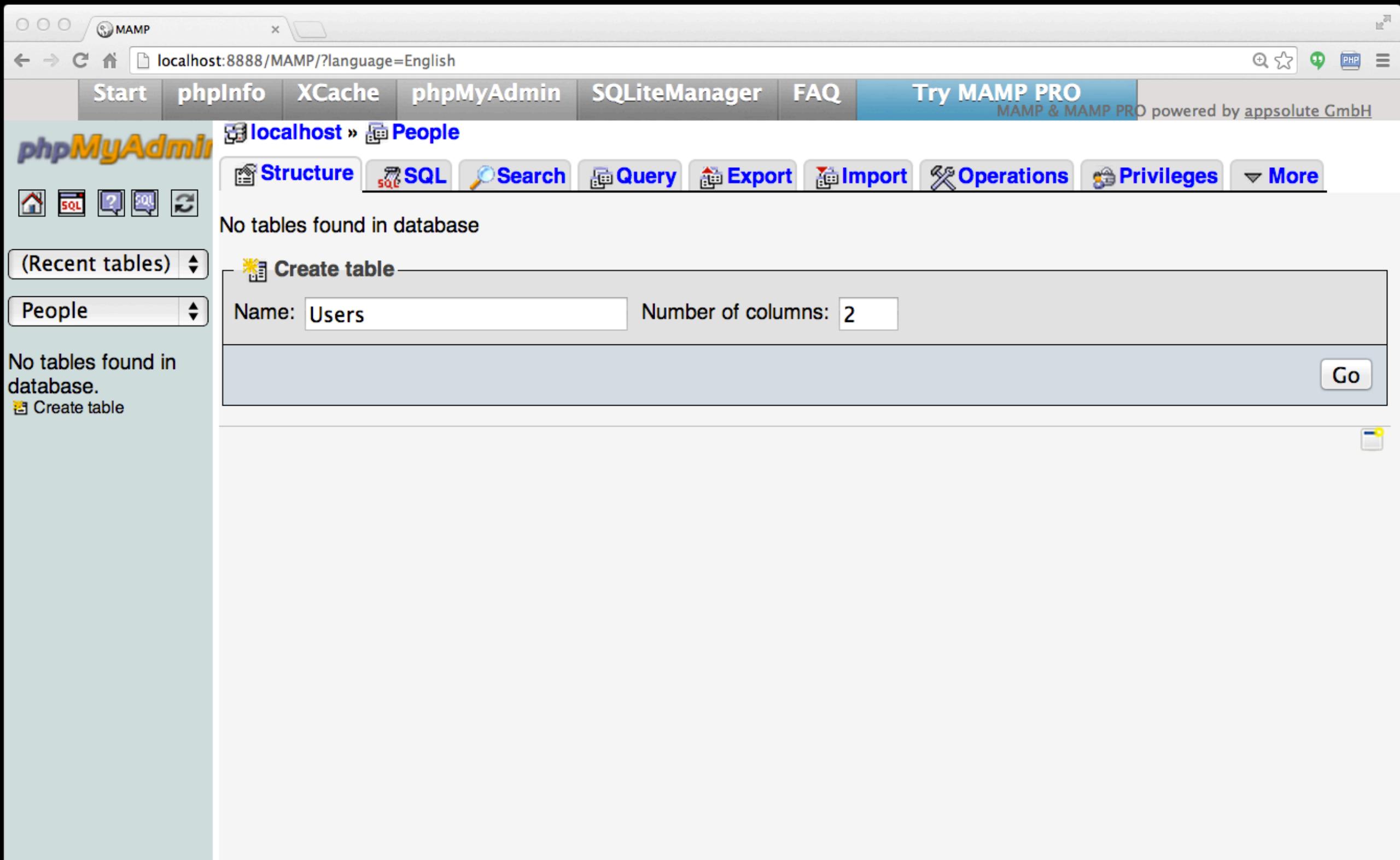
At the bottom, there are links for "Check All / Uncheck All With selected:" and "Enable Statistics".

Start Simple - A Single Table

- Let's make a table of Users in our People database
- Two columns - name and email

```
CREATE TABLE Users(  
    name VARCHAR(128),  
    email VARCHAR(128)  
);
```

```
DESCRIBE Users;
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The browser title bar reads "localhost:8888/MAMP/?language=English". The top navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and a prominent "Try MAMP PRO" button. Below the navigation is a breadcrumb trail: "localhost » People". The main menu bar offers options like Structure, SQL, Search, Query, Export, Import, Operations, Privileges, and More. A sub-menu for "Recent tables" lists "People", which is currently selected. The main content area displays the message "No tables found in database". A "Create table" section is visible, with "Name: Users" and "Number of columns: 2" entered. A "Go" button is located at the bottom right of this section. On the left sidebar, under "Recent databases", it also says "No tables found in database." and has a "Create table" link.

MAMP

localhost:8888/MAMP/?language=English

Start phpInfo XCache phpMyAdmin SQLiteManager FAQ Try MAMP PRO
MAMP & MAMP PRO powered by apposite GmbH

phpMyAdmin

Table name: Add column(s)

Structure

Name	Type	Length/Values	Default	Collation
name	VARCHAR	128	None	
email	VARCHAR	128	None	

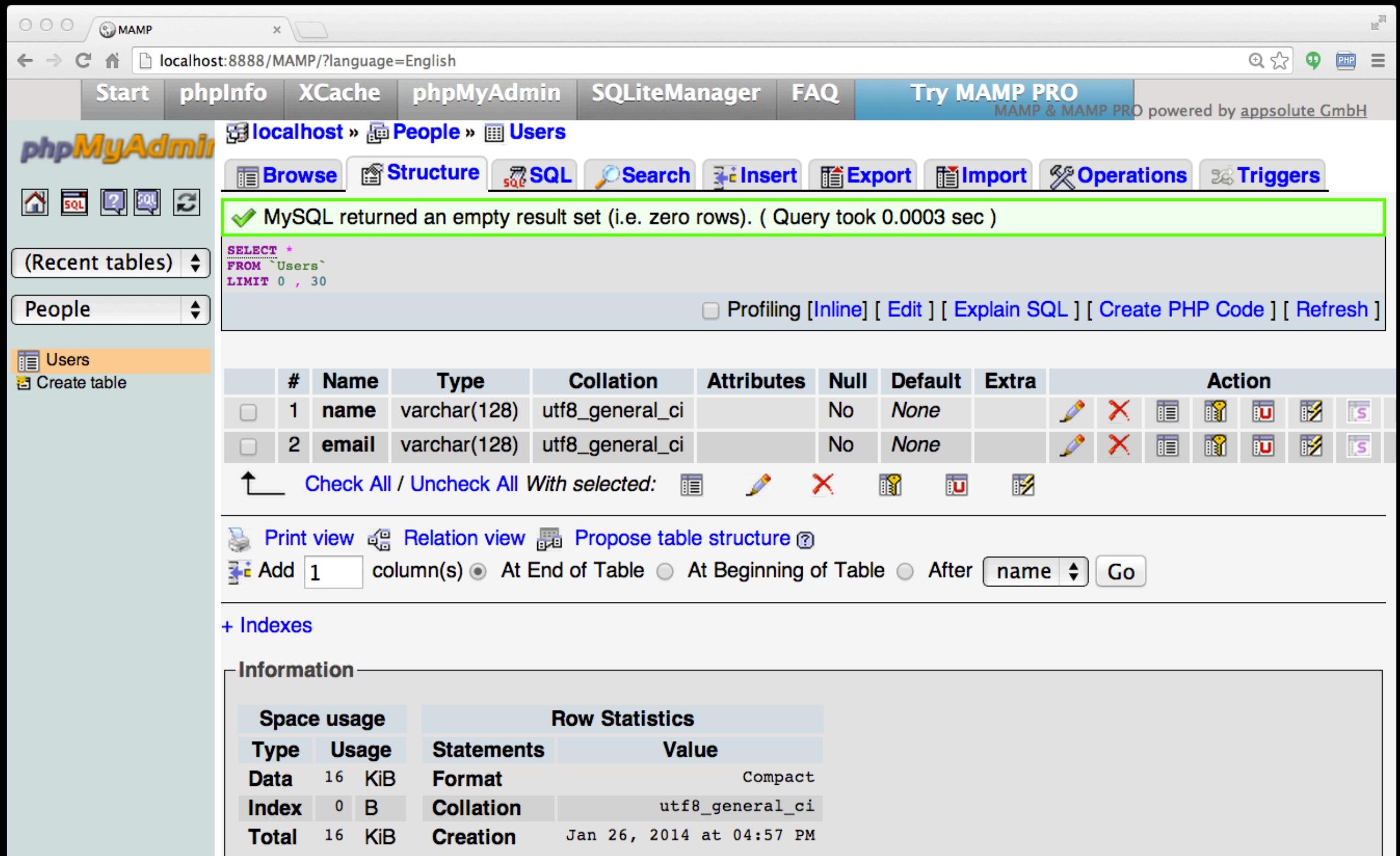
Table comments:

Storage Engine: InnoDB

Collation:

PARTITION definition:

(Recent tables)
No tables found in database.



The screenshot shows the phpMyAdmin interface for a MySQL database named 'People'. The current page is 'Users' under the 'People' category. The top navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. Below the navigation is a breadcrumb trail: localhost > People > Users. The main toolbar includes buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Operations, and Triggers.

A message box at the top states: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 sec)".

The SQL query displayed is:

```
SELECT *  
FROM `Users`  
LIMIT 0 , 30
```

Below the SQL area are buttons for Profiling, Inline, Edit, Explain SQL, Create PHP Code, and Refresh.

The 'Users' table structure is shown in a grid:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	name	varchar(128)	utf8_general_ci		No	None		     
2	email	varchar(128)	utf8_general_ci		No	None		     

Below the table, there is a link to 'Check All / Uncheck All With selected:' followed by icons for Edit, Delete, Check, Uncheck, Key, and Print.

At the bottom, there are buttons for Print view, Relation view, Propose table structure, Add (with a dropdown for 1 column(s)), and options for At End of Table, At Beginning of Table, After name, and Go.

Below the table, there is a section for Indexes and a detailed Information panel.

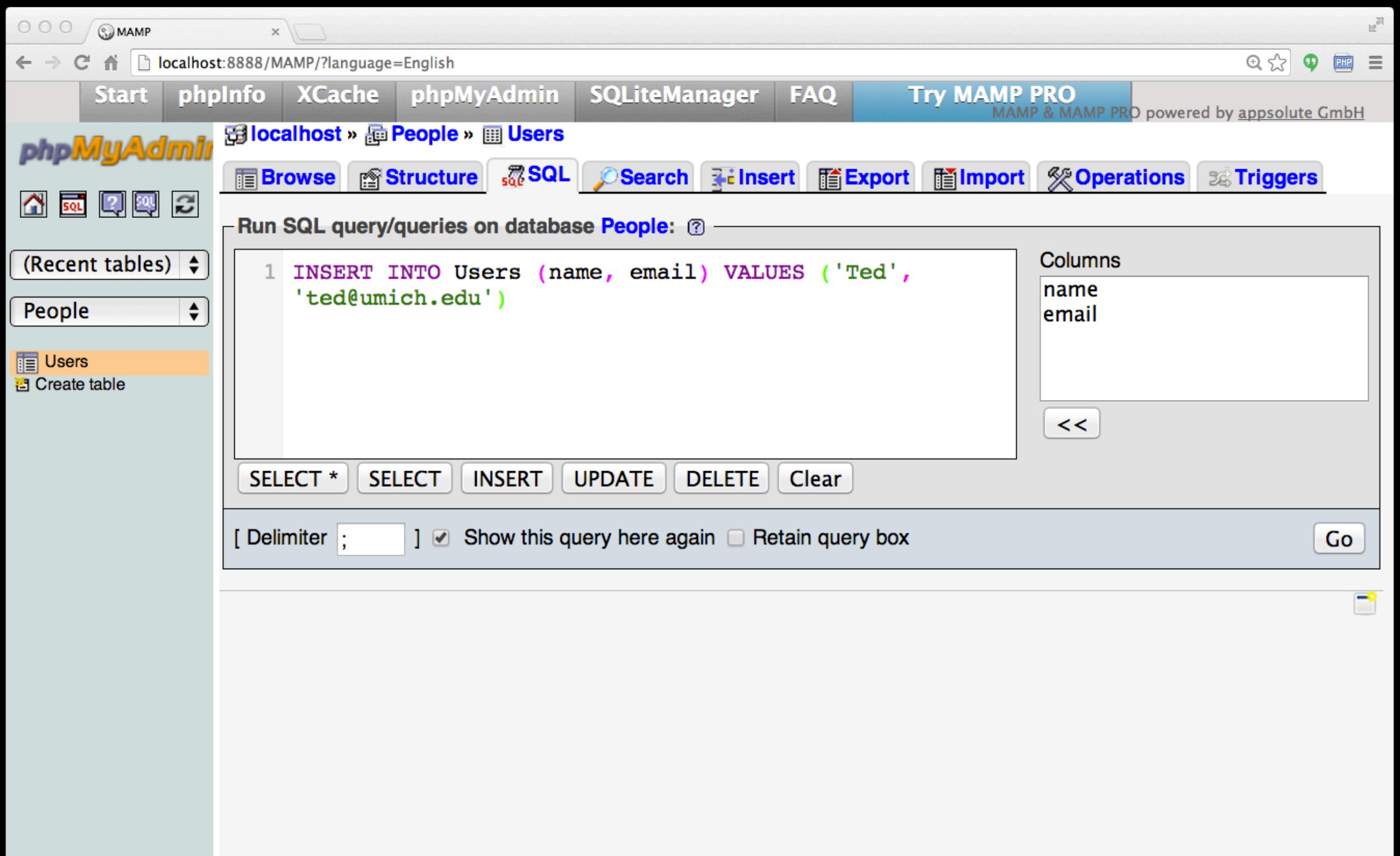
The Space usage and Row Statistics for the table are as follows:

Type	Usage	Statements	Value
Data	16 KiB	Format	Compact
Index	0 B	Collation	utf8_general_ci
Total	16 KiB	Creation	Jan 26, 2014 at 04:57 PM

SQL: Insert

The **INSERT** statement inserts a row into a table

```
INSERT INTO Users (name, email) VALUES ('Chuck', 'csev@umich.edu') ;
INSERT INTO Users (name, email) VALUES ('Somesh', 'somesh@umich.edu') ;
INSERT INTO Users (name, email) VALUES ('Caitlin', 'cait@umich.edu') ;
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu') ;
INSERT INTO Users (name, email) VALUES ('Sally', 'sally@umich.edu') ;
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The URL in the browser is `localhost:8888/MAMP/?language=English`. The main navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. Below the navigation is a breadcrumb trail: localhost > People > Users. The top menu bar contains several tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Operations, and Triggers. The SQL tab is currently selected. A large text area labeled "Run SQL query/queries on database People:" contains the following SQL code:

```
1 INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
```

To the right of the SQL area, there is a "Columns" panel listing the columns for the "Users" table: name and email. Below the SQL area are several buttons: SELECT *, SELECT, INSERT, UPDATE, DELETE, and Clear. At the bottom of the SQL input field, there are options for "Delimiter" (set to ;), "Show this query here again" (checked), "Retain query box" (unchecked), and a "Go" button.

MAMP

localhost:8888/MAMP/?language=English

Start phpInfo XCache phpMyAdmin SQLiteManager FAQ Try MAMP PRO
MAMP & MAMP PRO powered by appslute GmbH

localhost » People » Users

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 4 (~5 total , Query took 0.0004 sec)

```
SELECT *  
FROM `Users`  
LIMIT 0 , 30
```

Profiling [Inline] [Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

+ Options

	name	email
<input type="checkbox"/>	Chuck	csev@umich.edu
<input type="checkbox"/>	Sally	sally@umich.edu
<input type="checkbox"/>	Somesh	somesh@umich.edu
<input type="checkbox"/>	Caitlin	cait@umich.edu
<input type="checkbox"/>	Ted	ted@umich.edu

Check All / Uncheck All With selected:

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Query results operations

Print view Print view (with full texts) Export Display chart Create view

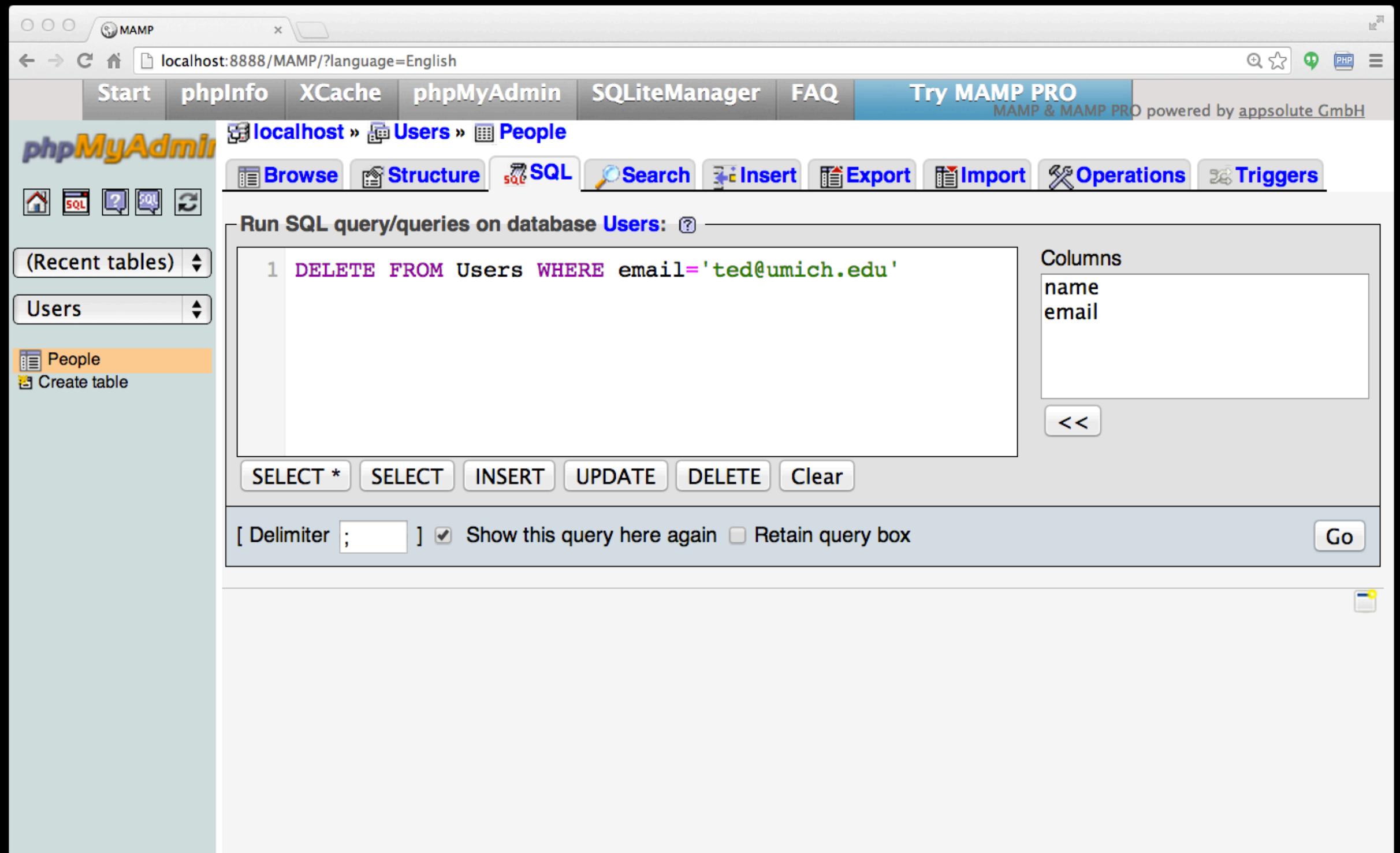




SQL: Delete

Deletes a row in a table based on selection criteria

```
DELETE FROM Users WHERE email='ted@umich.edu'
```



The screenshot shows the phpMyAdmin interface running on a local MAMP setup. The URL in the browser is `localhost:8888/MAMP/?language=English`. The main navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The current page is the phpMyAdmin dashboard for the 'Users' database, specifically the 'People' table.

The SQL query entered is:

```
1 DELETE FROM Users WHERE email='ted@umich.edu'
```

The results pane on the right shows the columns for the 'People' table:

name	email

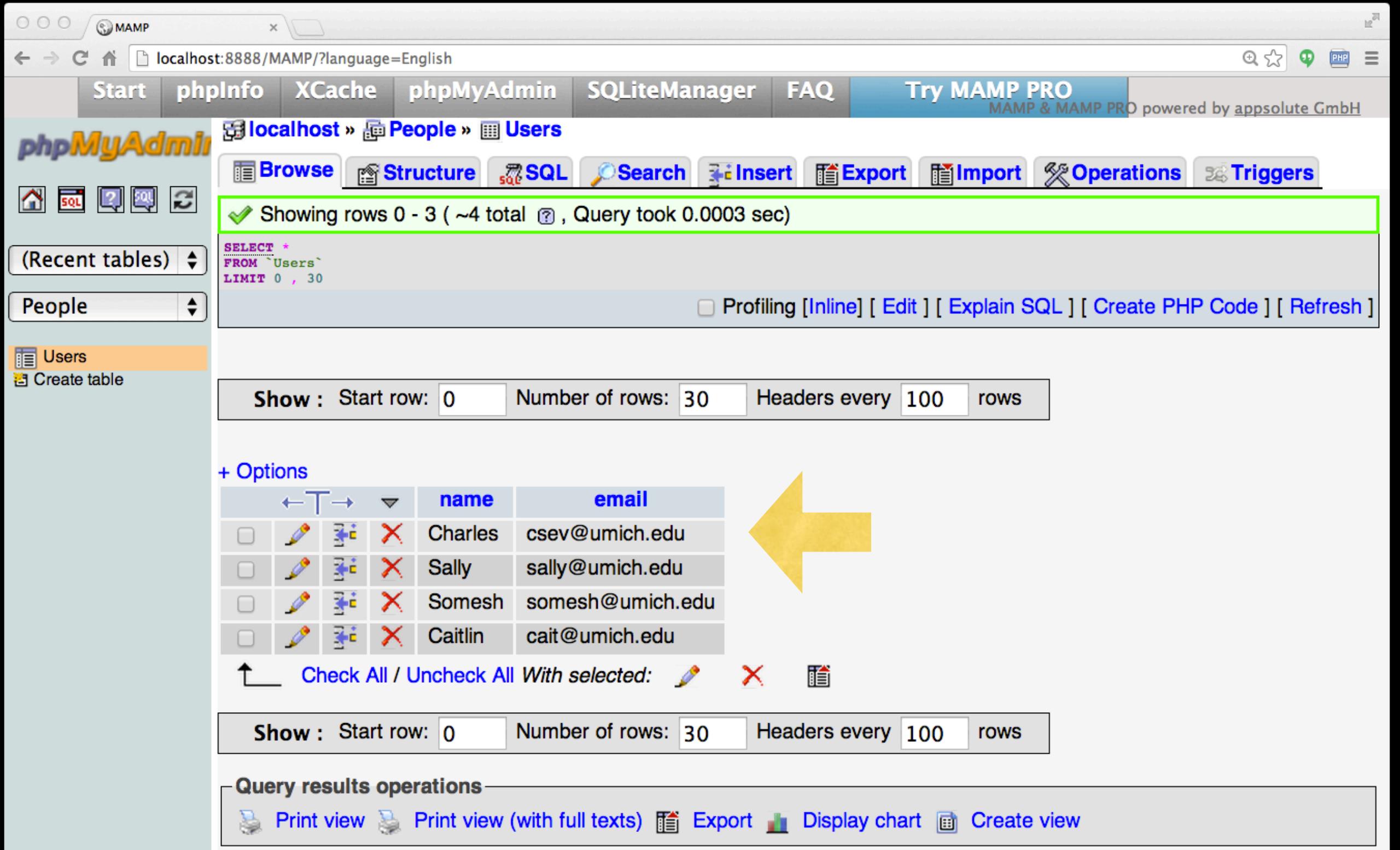
Below the query input area are several buttons: SELECT *, SELECT, INSERT, UPDATE, DELETE, Clear, and a Go button. There are also checkboxes for 'Delimiter' (set to ;) and 'Show this query here again' (checked), as well as a 'Retain query box' checkbox.



SQL: Update

Allows the updating of a field with a **WHERE** clause

```
UPDATE Users SET name='Charles' WHERE email='csev@umich.edu'
```



The screenshot shows the phpMyAdmin interface for a database named 'People'. The 'Users' table is selected. The table has four rows with the following data:

	name	email
<input type="checkbox"/>	Charles	csev@umich.edu
<input type="checkbox"/>	Sally	sally@umich.edu
<input type="checkbox"/>	Somesh	somesh@umich.edu
<input type="checkbox"/>	Caitlin	cait@umich.edu

Below the table, there is a section for 'Query results operations' with links for Print view, Print view (with full texts), Export, Display chart, and Create view.

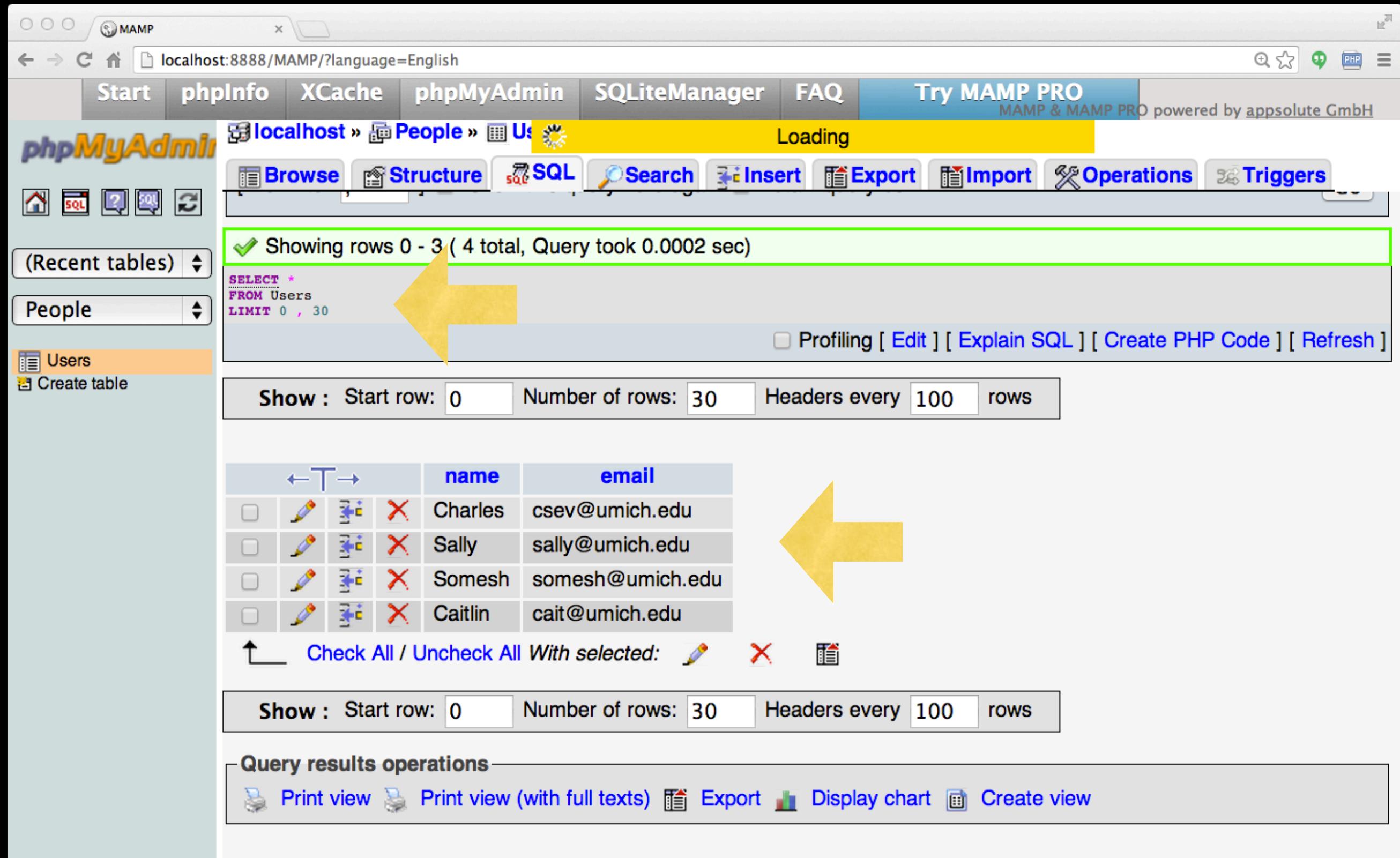


Retrieving Records: Select

Retrieves a group of records - you can either retrieve all the records or a subset of the records with a **WHERE** clause

```
SELECT * FROM Users
```

```
SELECT * FROM Users WHERE email='csev@umich.edu'
```



Showing rows 0 - 3 (4 total, Query took 0.0002 sec)

```
SELECT *  
FROM Users  
LIMIT 0 , 30
```

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

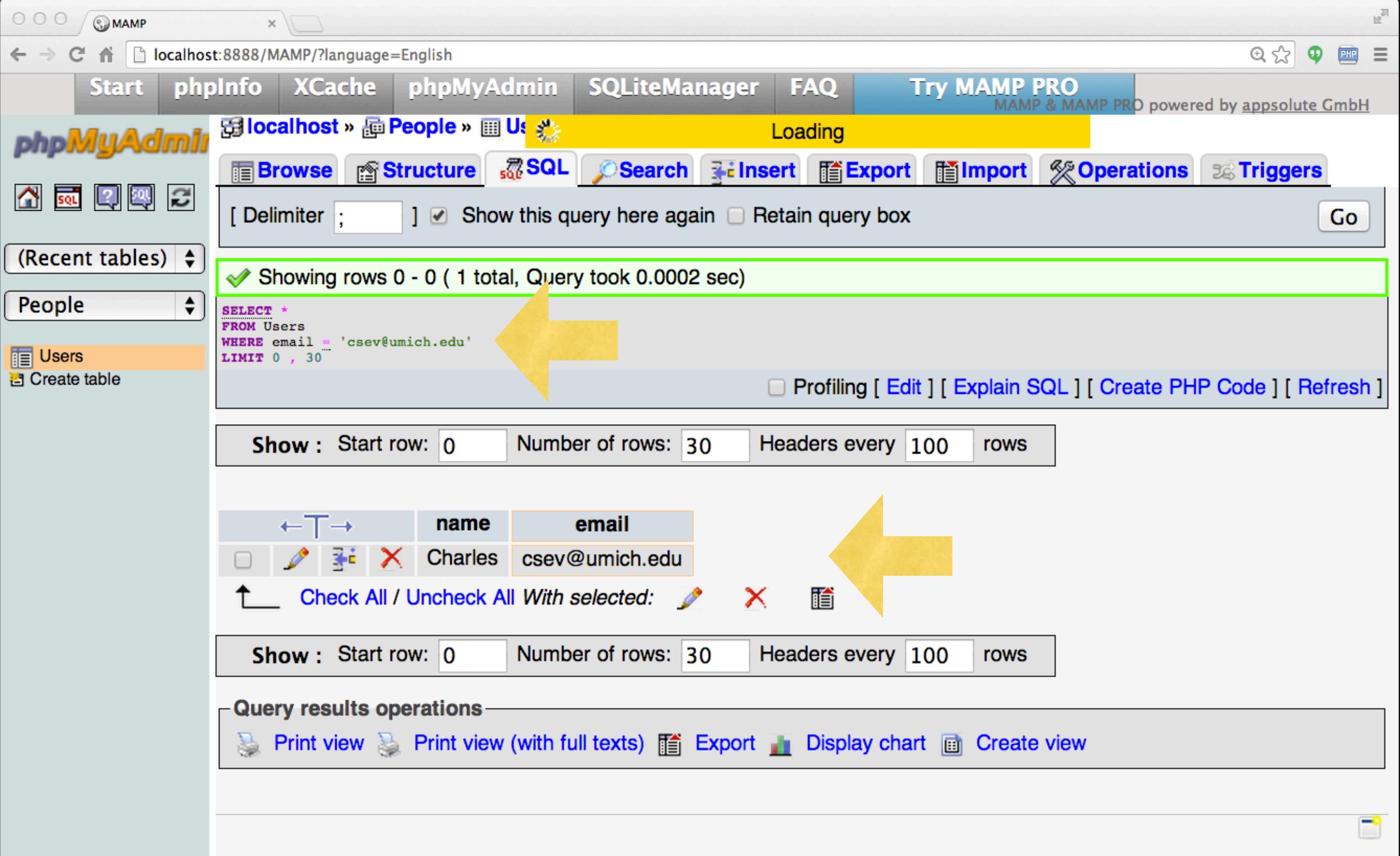
	name	email
<input type="checkbox"/>	Charles	csev@umich.edu
<input type="checkbox"/>	Sally	sally@umich.edu
<input type="checkbox"/>	Somesh	somesh@umich.edu
<input type="checkbox"/>	Caitlin	cait@umich.edu

Check All / Uncheck All With selected:

Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Query results operations

 Print view  Print view (with full texts)  Export  Display chart  Create view



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The URL in the browser is `localhost:8888/MAMP/?language=English`. The main menu bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The phpMyAdmin menu bar shows the current database as 'localhost' and the table as 'People'. Below the menu is a toolbar with various navigation and management buttons: Browse, Structure, SQL, Search, Insert, Export, Import, Operations, and Triggers.

The main content area displays a query result. The message says "Showing rows 0 - 0 (1 total, Query took 0.0002 sec)". The SQL query shown is:

```
SELECT *  
FROM Users  
WHERE email = 'csev@umich.edu'  
LIMIT 0 , 30
```

Below the query results, there is a "Show" configuration section with fields for "Start row: 0", "Number of rows: 30", and "Headers every 100 rows".

Underneath the query results, a specific row from the 'Users' table is selected for editing. The row contains the columns 'name' (Charles) and 'email' (csev@umich.edu). The row is highlighted with a yellow border. To the left of the row, there are edit icons: a pencil for edit, a delete icon, and a checkmark for selecting all. Below these icons is the text "Check All / Uncheck All With selected:". To the right of the row are three more icons: a pencil, a delete, and a clipboard.

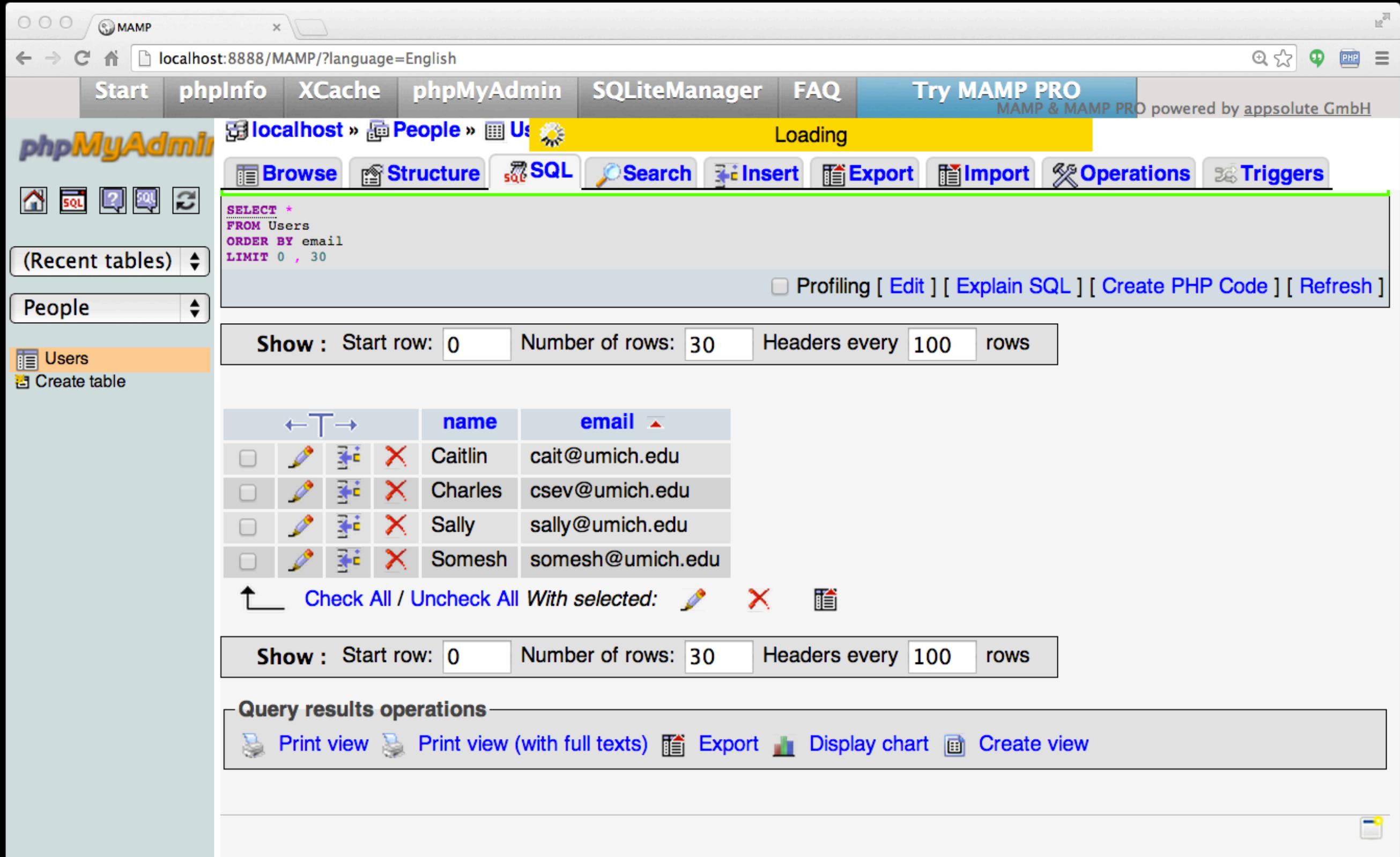
At the bottom of the page, there is another "Show" configuration section and a "Query results operations" section containing links for Print view, Print view (with full texts), Export, Display chart, and Create view.



Sorting with ORDER BY

You can add an **ORDER BY** clause to **SELECT** statements to get the results sorted in ascending or descending order

```
SELECT * FROM Users ORDER BY email
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The URL in the browser is `localhost:8888/MAMP/?language=English`. The main navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. Below the navigation is a breadcrumb trail: localhost > People > Users. A toolbar below the trail provides quick access to Browse, Structure, SQL, Search, Insert, Export, Import, Operations, and Triggers.

In the central area, there is a SQL query editor containing the following code:

```
SELECT *  
FROM Users  
ORDER BY email  
LIMIT 0 , 30
```

Below the query editor are filtering options: Show : Start row: 0, Number of rows: 30, Headers every 100 rows.

The main content area displays a table with the following data:

	name	email
<input type="checkbox"/>	Caitlin	cait@umich.edu
<input type="checkbox"/>	Charles	csev@umich.edu
<input type="checkbox"/>	Sally	sally@umich.edu
<input type="checkbox"/>	Somesh	somesh@umich.edu

Below the table are buttons for Check All / Uncheck All With selected: and a set of edit icons (pencil, delete, edit).

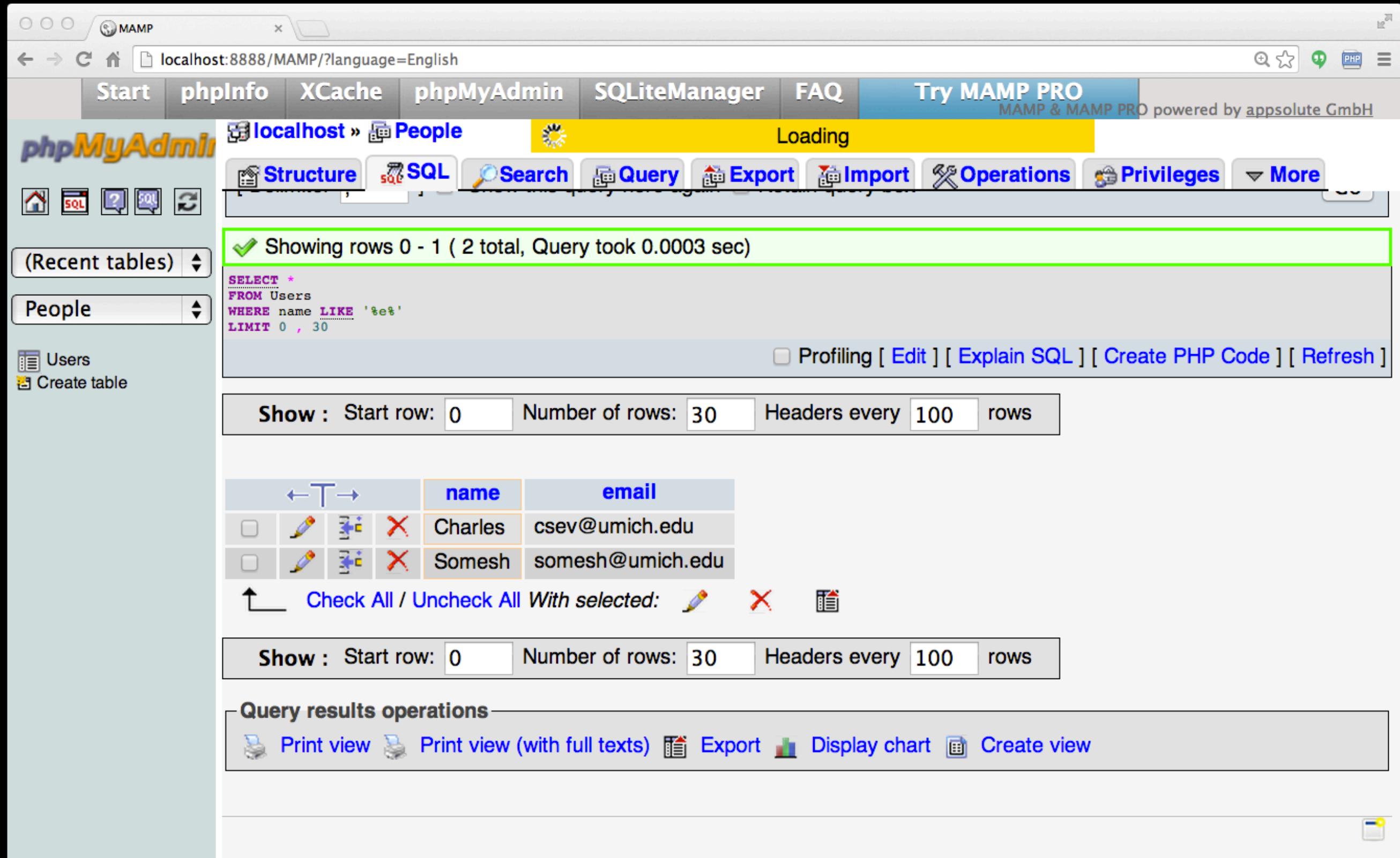
At the bottom, another set of filtering options: Show : Start row: 0, Number of rows: 30, Headers every 100 rows.

The footer section is titled "Query results operations" and includes links for Print view, Print view (with full texts), Export, Display chart, and Create view.

The LIKE Clause

We can do wildcard matching in a WHERE clause
using the LIKE operator

```
SELECT * FROM Users WHERE name LIKE '%e%'
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The URL in the browser is `localhost:8888/MAMP/?language=English`. The main navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The phpMyAdmin tab is active, showing the database structure for the 'People' table.

The 'Structure' tab is selected. A query has been run:

```
SELECT *  
FROM Users  
WHERE name LIKE '%se'  
LIMIT 0 , 30
```

The results show two rows:

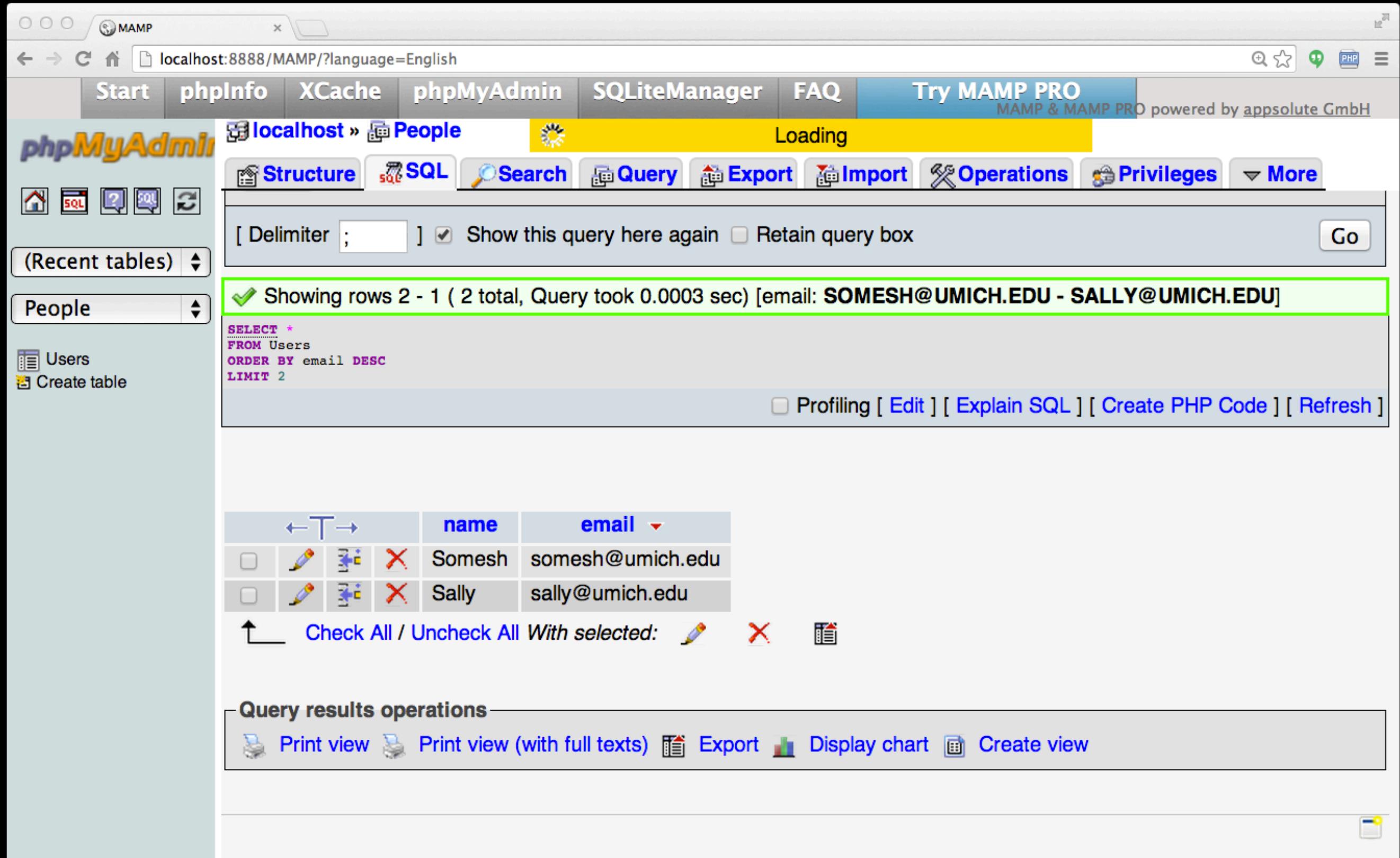
	name	email
<input type="checkbox"/>	Charles	csev@umich.edu
<input type="checkbox"/>	Somesh	somesh@umich.edu

Below the table, there are buttons for Check All / Uncheck All With selected, and links for Print view, Print view (with full texts), Export, Display chart, and Create view.

The LIMIT Clause

- The **LIMIT** clause can request the first "n" rows, or the first "n" rows after some starting row. Note: the first row is zero, not one.
- WHERE and ORDER BY clauses happen ***before*** the LIMIT is applied.
- The limit can be a count or a starting row and count (starts from 0).

```
SELECT * FROM Users ORDER BY email DESC LIMIT 2;  
SELECT * FROM Users ORDER BY email LIMIT 1,2;
```



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The top navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The main title is "localhost » People". A toolbar below the title offers options like Structure, SQL, Search, Query, Export, Import, Operations, Privileges, and More.

In the central query results area, a message indicates "Showing rows 2 - 1 (2 total, Query took 0.0003 sec) [email: SOMESH@UMICH.EDU - SALLY@UMICH.EDU]". The SQL query displayed is:

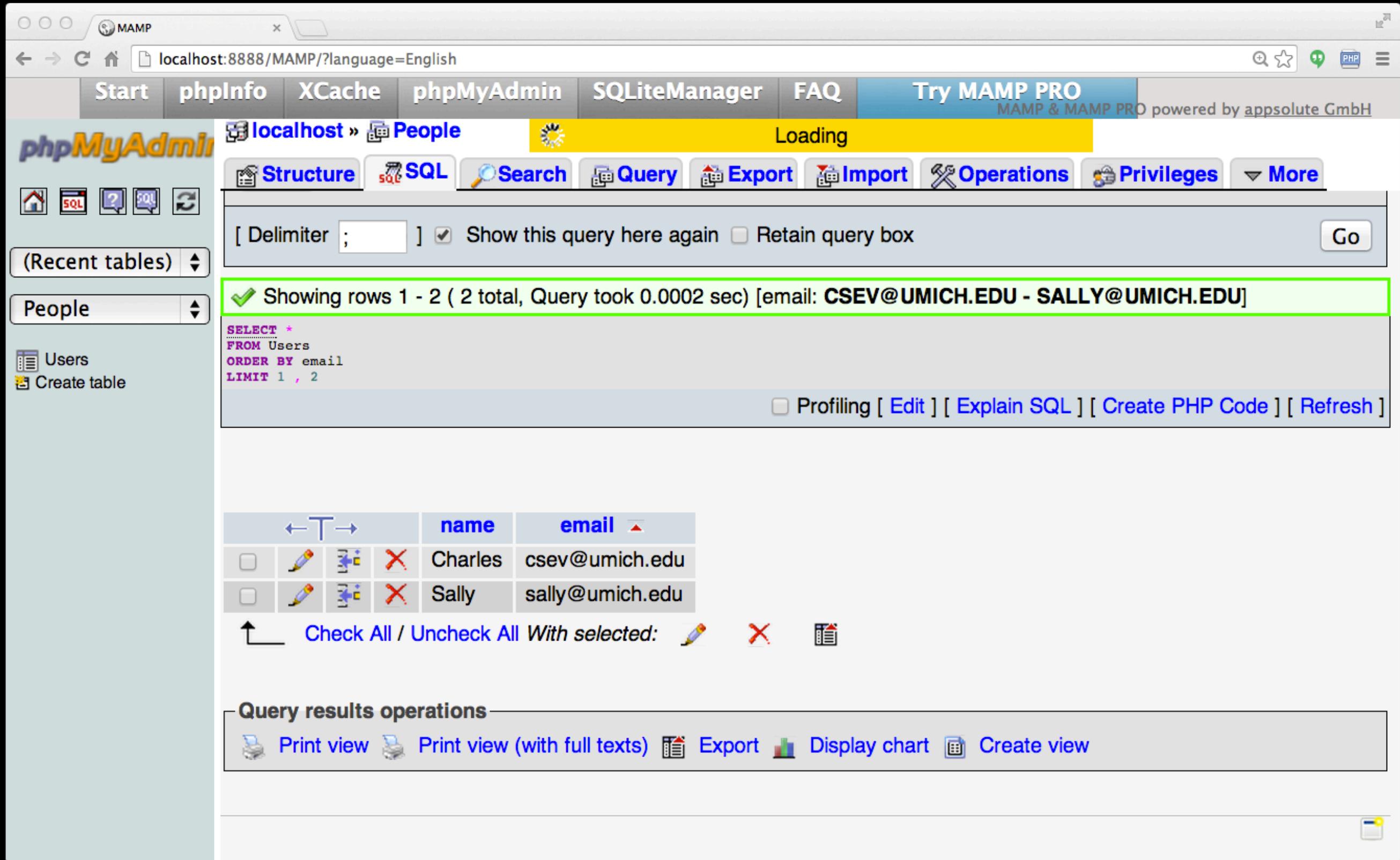
```
SELECT *  
FROM Users  
ORDER BY email DESC  
LIMIT 2
```

The results table displays two rows of data:

	name	email
<input type="checkbox"/>	Somesh	somesh@umich.edu
<input type="checkbox"/>	Sally	sally@umich.edu

Below the table, there are buttons for Check All / Uncheck All, and operations like Edit, Delete, and View.

At the bottom, under "Query results operations", are links for Print view, Print view (with full texts), Export, Display chart, and Create view.



The screenshot shows the phpMyAdmin interface running on a local MAMP server. The top navigation bar includes links for Start, phpInfo, XCache, phpMyAdmin, SQLiteManager, FAQ, and Try MAMP PRO. The main content area displays a query result for the 'Users' table:

```
SELECT *  
FROM Users  
ORDER BY email  
LIMIT 1 , 2
```

The result shows two rows:

	name	email
Charles	csev@umich.edu	
Sally	sally@umich.edu	

Below the table, there are buttons for Check All / Uncheck All With selected, and icons for Edit, Delete, and View.

At the bottom, there is a section for Query results operations with links for Print view, Print view (with full texts), Export, Display chart, and Create view.

Counting Rows with SELECT

You can request to receive the **count** of the rows that would be retrieved instead of the rows

```
SELECT COUNT(*) FROM Users;  
SELECT COUNT(*) FROM Users WHERE email='csev@umich.edu'
```

SQL Summary

```
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
```

```
DELETE FROM Users WHERE email='ted@umich.edu'
```

```
UPDATE Users SET name='Charles' WHERE email='csev@umich.edu'
```

```
SELECT * FROM Users WHERE email='csev@umich.edu'
```

```
SELECT * FROM Users ORDER BY email
```

```
SELECT * FROM Users WHERE name LIKE '%e%'
```

```
SELECT * FROM Users ORDER BY email LIMIT 1,2;
```

```
SELECT COUNT(*) FROM Users WHERE email='csev@umich.edu'
```

This is not too exciting (so far)

- Tables pretty much look like big, fast programmable spreadsheets with rows, columns, and commands.
- The power comes when we have more than one table and we can exploit the relationships between the tables.



Data Types in SQL



Looking at Data Types

- Text fields (small and large)
- Binary fields (small and large)
- Numeric fields
- AUTO_INCREMENT fields

String Fields

- Understand character sets and are indexable for searching
- **CHAR** allocates the entire space (faster for small strings where length is known)
- **VARCHAR** allocates a variable amount of space depending on the data length (less space)

Text Fields

- Have a character set - paragraphs or HTML pages
 - **TINYTEXT** up to 255 characters
 - **TEXT** up to 65K
 - **MEDIUMTEXT** up to 16M
 - **LONGTEXT** up to 4G
- Generally not used with indexing or sorting - and only then limited to a prefix

Binary Types (rarely used)

- Character = 8 - 32 bits of information depending on character set
- Byte = 8 bits of information
 - **BYTE(n)** up to 255 bytes
 - **VARBINARY(n)** up to 65K bytes
- Small Images - data
- Not indexed or sorted

Binary Large Object (BLOB)

- Large raw data, files, images, word documents, PDFs, movies, etc.
- No translation, indexing, or character set
 - **TINYBLOB(n)** - up to 255
 - **BLOB(n)** - up to 65K
 - **MEDIUMBLOB(n)** - up to 16M
 - **LONGBLOB(n)** - up to 4G



Integer Numbers

Integer numbers are very efficient, take little storage, and are easy to process because CPUs can often compare them with a single instruction.

- **TINYINT** (-128, 128)
- **SMALLINT** (-32768, +32768)
- **INT** or **INTEGER** (2 Billion)
- **BIGINT** - ($10^{**}18$ ish)

Floating Point Numbers

Floating point numbers can represent a wide range of values, but accuracy is limited.

- **FLOAT** (32-bit) 10^{**38} with seven digits of accuracy
- **DOUBLE** (64-bit) 10^{**308} with 14 digits of accuracy



Dates

- **TIMESTAMP** - 'YYYY-MM-DD HH:MM:SS' (1970, 2037)
- **DATETIME** - 'YYYY-MM-DD HH:MM:SS'
- **DATE** - 'YYYY-MM-DD'
- **TIME** - 'HH:MM:SS'
- Built-in MySQL function NOW()



Database Keys and Indexes

AUTO_INCREMENT

Often as we make multiple tables and need to JOIN them together we need an integer primary key for each row so we can efficiently add a reference to a row in some other table as a foreign key.

```
DROP TABLE Users;  
  
CREATE TABLE Users (  
    user_id INT UNSIGNED NOT NULL  
        AUTO_INCREMENT,  
    name VARCHAR(128),  
    email VARCHAR(128),  
    PRIMARY KEY(user_id),  
    INDEX(email)  
)
```



MySQL Functions

Many operations in MySQL need to use the built-in functions (like NOW() for dates).

- <http://dev.mysql.com/doc/refman/5.0/en/string-functions.html>
- <http://dev.mysql.com/doc/refman/5.0/en/date-and-time-functions.html>



Indexes

- As a table gets large (they always do), scanning all the data to find a single row becomes very costly
- When drchuck@gmail.com logs into FaceBook, they must find my password amongst 500 million users
- There are techniques to greatly shorten the scan as long as you create data structures and maintain those structures - like shortcuts
- Hashes or Trees

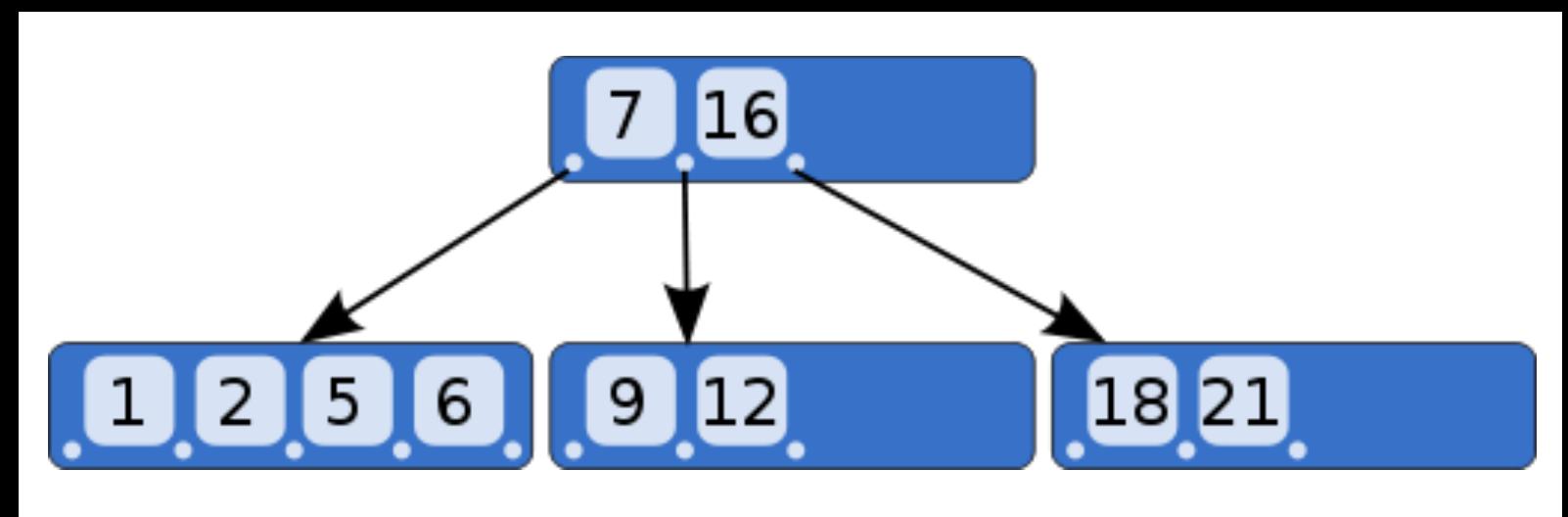


MySQL Index Types

- **PRIMARY KEY** - Very little space, exact match, requires no duplicates, extremely fast for integer fields
- **INDEX** - Good for individual row lookup and sorting / grouping results - works best with exact matches or prefix lookups - can suggest HASH or BTREE

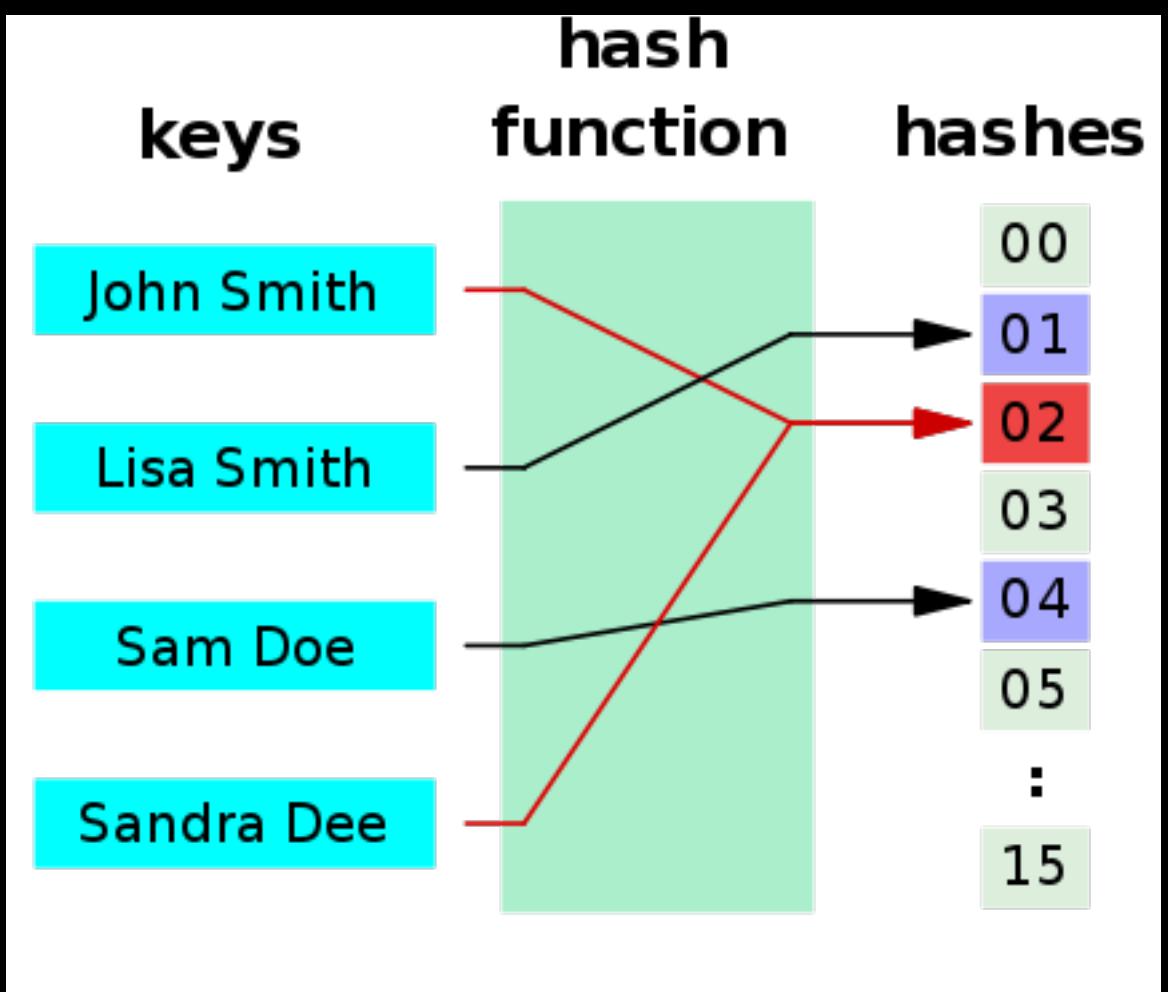
B-Trees

A *B-tree* is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic amortized time. The *B-tree* is optimized for systems that read and write large blocks of data. It is commonly used in databases and file systems.



Hashes

A *hash function* is any algorithm or subroutine that maps large data sets to smaller data sets, called *keys*. For example, a single integer can serve as an index to an array (cf. associative array). The values returned by a hash function are called *hash values*, *hash codes*, *hash sums*, *checksums*, or simply *hashes*. Hash functions are mostly used to accelerate table lookup or data comparison tasks such as finding items in a database...



Specifying Indexes

```
DROP TABLE Users;

CREATE TABLE Users (
    user_id INT UNSIGNED NOT NULL
        AUTO_INCREMENT,
    name VARCHAR(128),
    email VARCHAR(128),
    PRIMARY KEY(user_id),
    INDEX(email)
)
```

```
ALTER TABLE Users ADD INDEX ( email ) USING BTREE
```



Summary

- SQL allows us to describe the shape of data to be stored and give many hints to the database engine as to how we will be accessing or using the data.
- SQL is a language that provides us operations to Create, Read, Update, and Delete (CRUD) our data in a database.

Acknowledgements / Contributions

These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) as part of www.wa4e.com and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan
School of Information

Insert new Contributors and Translators here including names and dates

Continue new Contributors and Translators here

