

## Python for Everybody Database Handout

<https://www.py4e.com/lectures3/Pythonlearn-15-Database-Handout.txt>

Download and Install: <http://sqlitebrowser.org/>

### Single Table SQL

```
CREATE TABLE "Users" ("name" TEXT, "email" TEXT)
```

```
INSERT INTO Users (name, email) VALUES ('Chuck', 'csev@umich.edu')
INSERT INTO Users (name, email) VALUES ('Colleen', 'cvl@umich.edu')
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
INSERT INTO Users (name, email) VALUES ('Sally', 'al@umich.edu')
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
INSERT INTO Users (name, email) VALUES ('Kristen', 'kf@umich.edu')
```

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

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

```
SELECT * FROM Users
```

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

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

```
SELECT * FROM Users ORDER BY name DESC
```

### Multi-Table SQL:

```
CREATE TABLE "Artist" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE,
    "name" TEXT)
```

```
CREATE TABLE "Album" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE,
    artist_id INTEGER,
    "title" TEXT)
```

```
CREATE TABLE "Genre" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE,
    "name" TEXT)
```

```
CREATE TABLE "Track" (
    "id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL UNIQUE,
    album_id INTEGER, genre_id INTEGER, len INTEGER, rating INTEGER,
    "title" TEXT, "count" INTEGER)
```

```
INSERT INTO Artist (name) VALUES ('Led Zepplin')
INSERT INTO Artist (name) VALUES ('AC/DC')

INSERT INTO Genre (name) VALUES ('Rock') ;
INSERT INTO Genre (name) VALUES ('Metal');

INSERT INTO Album (title, artist_id) VALUES ('Who Made Who', 2);
INSERT INTO Album (title, artist_id) VALUES ('IV', 1);

INSERT INTO Track (title, rating, len, count, album_id, genre_id)
VALUES ('Black Dog', 5, 297, 0, 2, 1) ;
INSERT INTO Track (title, rating, len, count, album_id, genre_id)
VALUES ('Stairway', 5, 482, 0, 2, 1) ;
INSERT INTO Track (title, rating, len, count, album_id, genre_id)
VALUES ('About to Rock', 5, 313, 0, 1, 2) ;
INSERT INTO Track (title, rating, len, count, album_id, genre_id)
VALUES ('Who Made Who', 5, 207, 0, 1, 2) ;

SELECT Album.title, Artist.name FROM Album JOIN Artist
ON Album.artist_id = Artist.id

SELECT Album.title, Album.artist_id, Artist.id, Artist.name
FROM Album JOIN Artist ON Album.artist_id = Artist.id

SELECT Track.title, Track.genre_id, Genre.id, Genre.name
FROM Track JOIN Genre

SELECT Track.title, Genre.name FROM Track JOIN Genre
ON Track.genre_id = Genre.id

SELECT Track.title, Artist.name, Album.title, Genre.name
FROM Track JOIN Genre JOIN Album JOIN Artist
ON Track.genre_id = Genre.id AND Track.album_id = Album.id
AND Album.artist_id = Artist.id
```

### Many-Many Relationship

```
CREATE TABLE User (
    id      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    name    TEXT UNIQUE,
    email   TEXT
) ;

CREATE TABLE Course (
    id      INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,
    title   TEXT UNIQUE
) ;
```

```
CREATE TABLE Member (  
    user_id    INTEGER,  
    course_id  INTEGER,  
    role       INTEGER,  
    PRIMARY KEY (user_id, course_id)  
);
```

```
INSERT INTO User (name, email) VALUES ('Jane', 'jane@tsugi.org');  
INSERT INTO User (name, email) VALUES ('Ed', 'ed@tsugi.org');  
INSERT INTO User (name, email) VALUES ('Sue', 'sue@tsugi.org');
```

```
INSERT INTO Course (title) VALUES ('Python');  
INSERT INTO Course (title) VALUES ('SQL');  
INSERT INTO Course (title) VALUES ('PHP');
```

```
INSERT INTO Member (user_id, course_id, role) VALUES (1, 1, 1);  
INSERT INTO Member (user_id, course_id, role) VALUES (2, 1, 0);  
INSERT INTO Member (user_id, course_id, role) VALUES (3, 1, 0);
```

```
INSERT INTO Member (user_id, course_id, role) VALUES (1, 2, 0);  
INSERT INTO Member (user_id, course_id, role) VALUES (2, 2, 1);
```

```
INSERT INTO Member (user_id, course_id, role) VALUES (2, 3, 1);  
INSERT INTO Member (user_id, course_id, role) VALUES (3, 3, 0);
```

```
SELECT User.name, Member.role, Course.title  
FROM User JOIN Member JOIN Course  
ON Member.user_id = User.id AND Member.course_id = Course.id  
ORDER BY Course.title, Member.role DESC, User.name
```