Relational Databases

Charles Severance



Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/.

Copyright 2009- Charles Severance







SQLite Manager For Firefox

| | TENSIONS THEMES COLLECTIONS MORE | Q search for add-ons |
|----------|--|--|
| Extensio | ns » SQLite Manager | |
| | SQLite Manager 0.8.1 by lazierthanthou | ★★★★★ 44 user reviews 192,653 users ⋒í |
| | Manage any SQLite database on your computer. | |
| | + Add to Firefox Privacy Policy | Add to collection Share this Add-on |
| | Meet the Developer: lazierthanthou | |
| | Learn why SQLite Manager was created and find out what's next for this add-on. | |

https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/

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 and in particular where there are multiple tables and the relationships between those tables involved in the query.

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

Terminology

- Database Contains many tables
- Relation (or table) contains tuples and attributes
- Tuple (or row) is a set of fields it generally represents 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)

| 00 | 00 | | | | Ē | SI | 502 - Datab | ase | | | | | |
|-----|-------|------|----------|------------|---------------|------|-------------------|---------|---------|---------------|------------|-----------|-----|
| 2 | | | | | Ê 🔮 | | S • 🖸 • | Σ | • A | the At | | | >>> |
| New | Open | Save | Print In | port Copy | Paste Forma | at : | Undo Redo | AutoS | um Sort | t A-Z Sort Z- | A 🗄 Gallen | y Toolbox | (|
| | | | | Sheets | Charts | | SmartArt C | Graphic | s | WordArt | | | |
| \$ | | · · | 1.1.1 | | • • • 1 • | 1 1 | 1 · · · 2 · D | | | 3 | 4 | D | |
| | | | | | A | | D | | | C | | U | |
| . 1 | | | | TITLE | (| 20 | RATING | Attr | LEN | es | | Rows | |
| 2 | | | | About to | Rock | | | 3 | | 354 | | 110003 | |
| 4 | } | | | Who Ma | de Who | | | 4 | | 252 | | Tuple | S |
| . 5 | 5 | | | | | | | | | | | | |
| 8 | 3 | | | | | | | Tat | oles / | Relatio | ons | | |
| | | 14.4 | ⊨. ⊨L. | Tracks All | bums 🖉 Artist | ts | Genres + | | | | | | |
| | | | | | | | | | | | | | 1 |

Two Roles in Large Projects

- Application Developer Builds the logic for the application, the look and feel of the application monitors the application for problems
- Database Administrator Monitors and adjusts the database as the program runs in production
- Often both people participate in the building of the "Data model"

Application Structure



Database Administrator (dba)

A database administrator (DBA) is a person responsible for the design, implementation, maintenance and repair of an organization's database. The role includes the development and design of database strategies, monitoring and improving database performance and capacity, and planning for future expansion requirements. They may also plan, co-ordinate and implement security measures to safeguard the database.

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

Database Model

A database model or database schema is the structure or format of a database, described in a formal language supported by the database management system, In other words, a "database model" is the application of a data model when used in conjunction with a database management system.

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



- Structured Query Language is the language we use to issue commands to the database
 - Create a table
 - Retrieve some data
 - Insert data
 - Delete data

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

Common Database Systems

- Three Major Database Management Systems in wide use
 - Oracle Large, commercial, enterprise-scale, very 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, Postgress, ...

SQLite Database Manager

- SQLite is a very popular database it is free and fast and small
- We have a FireFox plugin to manipulate SQLite databases
 - https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/
- SQLite is embedded in Python and a number of other languages



http://www.sqlite.org/famous.html

| f(x) = f(x) | | A 64 . | | | | |
|--------------------|---|---|---|--|--|---|
| | | 1 📑 🗌 | Directory | (Select Profile Date | (abase) 🗘 Go | |
| | Structure | Browse 8 | Search Exer | cute SQL DB Se | ttings | |
| TABLE: sqlite_mast | ter | | | | | |
| No. of Records: 0 | No. | of Indexes: 0 | No. of T | riggers: 0 | | |
| Columns (5) | | | | | | |
| Column ID | Name | Туре | Not Null | Default Value | Primary Key | E. |
| 0 | type | text | 0 | null | 0 | |
| 1 | name | text | 0 | null | 0 | |
| 2 | tbl_name | text | 0 | null | 0 | |
| 3 | rootpage | integer | 0 | null | 0 | |
| 4 | sql | text | 0 | null | 0 | |
| | | | | | | |
| | TABLE: sqlite_mast Export More Info No. of Records: 0 Columns (5) Column ID 0 1 2 3 4 | Structure TABLE: sqlite_master Export More Info More Info No. of Records: 0 No. of Columns (5) Column ID Name 0 type 1 name 2 tbl_name 3 rootpage 4 sql | Structure Browse of an anti-anti-anti-anti-anti-anti-anti-anti- | Structure Browse & Search Exect TABLE: sqlite_master Export More Info No. of Records: 0 No. of Indexes: 0 No. of Tr Columns (5) Column ID Name Type Not Null 0 type text 0 1 name text 0 2 tbl_name text 0 3 rootpage integer 0 4 sql text 0 | Structure Browse & search Execute SQL DB search TABLE: sqlite_master Export More Info No. of Records: 0 No. of Indexes: 0 No. of Triggers: 0 Columns (5) Column ID Name Type Not Null Default Value 0 type text 0 null 1 name text 0 null 2 tbl_name text 0 null 3 rootpage integer 0 null 4 sql text 0 null | Structure Browse & Search Execute SQL Dis Settings TABLE: sqlite_master Export More Info No. of Records: 0 No. of Indexes: 0 No. of Triggers: 0 Column ID Name Type Not Null Default Value Primary Key 0 type text 0 null 0 1 name text 0 null 0 2 tbl_name text 0 null 0 3 rootpage integer 0 null 0 4 sql text 0 null 0 |

https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/

Application Structure



Start Simple - A Single Table

• Lets make a table of People - with a Name and an E-Mail

| 0 | 0 0 | | SQL | te Manager - | /Users/c | sev/sql1.s | qlite | | | |
|---------|------------------------------|----------------|------------------|--------------------|--------------|--------------|--------------------|---------------|------|-----|
| ¢ | | | Database: mai | n ‡] 1 | able Name | Users | | | | Go |
| >q ▼ | Master Table (Tables (0) | Define Columr | 15 | C Temporary | y table 🗌 I | f Not Exists | | | | |
| | Views (0) Indexes (0) | Column Name | Data Type | Primary Key? | Autoinc? | Allow Null? | Unique? | Default Value | | |
| • | Triggers (0) | email | TEXT | 🗌 Yes | O Yes | 🗹 Yes | 🗌 Yes | | • | |
| | | name | TEXT | 🗌 Yes | O Yes | 🗹 Yes | 🗌 Yes | | - | |
| | | | T | 🗌 Yes | O Yes | 🗹 Yes | 🗌 Yes | | • | |
| | | | • | 🗌 Yes | Yes | 🗹 Yes | 🗌 Yes | | • | |
| | | | | 🗌 Yes | Ves | 🗹 Yes | 🗌 Yes | | | E. |
| | | | | 🗌 Yes | O Yes | 🗹 Yes | 🗌 Yes | | | |
| | | | | 🗌 Yes | Ves | Ves | 🗌 Yes | | | |
| | | | _ | 🗌 Yes | 🗌 Yes | 🗹 Yes | 🗌 Yes | | | |
| | | | • | 🗌 Yes | Yes | 🗹 Yes | 🗌 Yes | | • | |
| | | | | Yes | 🗌 Yes | 🗹 Yes | 🗌 Yes | | • | |
| | | | | | | | (| Cancel O | ĸ | |
| | | | | Our | first | tabl | e w | ith two c | olur | nns |
| SQ | Lite 3.7.17 Ge | cko 24.0 0.7.7 | Shared Number of | f files in selecte | ed directory | : 13 | ter de la comercia | | | |

| 000 | | | SQLite Manager - /Users/csev/sql1.sqlit | te | |
|--------------|-----|-------------------------------------|---|--------------------------------|--------|
| 0 * | 🗅 🔁 | <i>⊈</i> f(x) ⊞ [#] | Directory | (Select Profile Database) 💲 Go | |
| | | | Structure Provise & Search Ex | acuta SOL DR Sattings | |
| sql1.sqlite | ÷ | | Structure Browse & Search Ex | ecute SQL DB Settings | |
| Master Table | (1) | | | | |
| ▼ Tables (1) | | TABLE Users | Search Show All Ar | dd Duplicate Edit | Delete |
| Users | | rowid | email | name | Ę |
| Views (0) | | 1 | csev@umich.edu | Chuck | |
| Indexes (0) | | 2 | olga@umich.edu | Olga | |
| Triggers (0) | | 3 | gab@umich.edu | Gab | |
| | | 4 | gaurav@umich.edu | Gaurav | |
| | | | Our table | e with four rows | |
| | | | | | |
| | | << | < 1 to 4 of 4 > | >> | |



- Structured Query Language is the language we use to issue commands to the database
 - Create a table
 - Retieve some data
 - Insert data
 - Delete data

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



• The Insert statement inserts a row into a table

insert into Users (name, email) values ('Ted', 'ted@umich.edu')

| 000 | | | | | SQLit | e Manag | ger – /Users/c | sev/so | l1.sqlite | | | | | | | | | | |
|---|---------|-------------------------------|-----------------|-----------------------|-----------|-------------|--|----------|---|-------|---------------|-----------|---|---|--|-------------|--|-------------|--------|
| ¢ 🕺 | | Ê | 3 | <i>f</i> (<i>x</i>) | | * | r 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | Dir | ectory | ► (Se | ect Profile I | Database) | | ¢ Go | | | | | |
| sql1.sqlite | \$ | _ | | | Structu | re | Browse & Sear | ch | Execute S | QL I | OB Setting | s | | | | | | | |
| Master Tabl Tables (1) | • le | | Enter SQ | L Users (na | me email) | values ('T | ed' 'ted⊚umich | edu') | | | | | | | | | | | |
| Views (0) Indexes (0) | | Run SQL Actions V Last Error: | | | | | | | | | | | | | | | | | |
| Triggers (0) |) | | - Hall SQ | | | Luster | 1000 | | | | SQ | Lite Mana | ager – / | /Users/cs | sev/sql1 | 1.sqlite | | | |
| | | | | | | | I 🕉 | | <i>i</i> 🖉 | f(x) | Ē | ı∰ | | r i | Direc | ctory 🕨 | (Select Profile | e Database) | ‡ Go |
| | | | | | | | sal1.salite | | | | Struc | ture | Brows | e & Searc | ch E | Execute SQL | DB Settin | igs | |
| | | | | | | | Master Ta Tables (1) Users Views (0) Indexes (0) Triggers (0) |)))) | TABLE rowid 1 2 3 4 5 | Users | | Sear | rch em cse olga gab gau tedr | Show A ail v@umich.ee a@umich.ee @umich.ed @umich.ed | All cdu cdu cdu du h.edu du du | Add | Duplicate name Chuck Olga Gab Gaurav Ted | Edit |) Dele |
| SQLite 3.7.17 | Gecl | ko 24 | 1.0 0.7. | .7 Shar | ed Numb | ber of file | 25 | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | < | < | < | 1 to | o 5 of | 5 | > | >> | | | |



• Deletes a row in a table based on a selection criteria

delete from Users where email='ted@umich.edu'

| Image: Second secon | Database) ‡ Go S S QLite Manager – /Users/csev/s D Ucture Browse & Search Search Show All email csev@umich.edu olga@umich.edu | sql1.sqlite Directory (Select Profile Database) |
|---|--|--|
| sql1.sqlite Structure Browse & Search Execute SQL DB Setting: Master Table Tables (1) Enter SQL delete from Users where email="ted@umich.edu" Views (0) Indexes (0) Run SQL Actions * Last Error: not an error Triggers (0) Image: Structure in the second in the se | s QLite Manager – /Users/csev/s Cucture Browse & Search Search Show All email csev@umich.edu olga@umich.edu | sql1.sqlite Directory (Select Profile Database) + Go Execute SQL DB Settings Add Duplicate Edit Dele Chuck |
| Master Table Tables (1) Users Views (0) Indexes (0) Triggers (0) Run SQL Actions Last Error: not an error Sql1.sqlite + Master Table Tables (1) Weaves (0) Indexes (0) Tables (1) Views (0) Views (0) Tables (1) Views (0) Views (0) Tables (1) Views (0) Tables (1) Tables (1) Views (0) Indexes (0) <pindexes (0)<="" p=""> Indexes (0) <pindexes (0)<="" p=""> Indexes (0) <pi< th=""><th>SQLite Manager – /Users/csev/s Uture Browse & Search Search Show All email csev@umich.edu olga@umich.edu</th><th>sql1.sqlite Directory (Select Profile Database)</th></pi<></pindexes></pindexes> | SQLite Manager – /Users/csev/s Uture Browse & Search Search Show All email csev@umich.edu olga@umich.edu | sql1.sqlite Directory (Select Profile Database) |
| Kun SQL Actions Last Error: not an error Triggers (0) Kun SQL Actions Last Error: not an error Sql1.sqlite Master Table TABLE Users Tables (1) Views (0) Views (0) Indexes (0) Triggers (0) | GLite Manager – /Users/csev/s Control Control | sql1.sqlite Directory (Select Profile Database) Co Execute SQL DB Settings Add Duplicate Edit Dele Chuck Olse |
| Image: second system | D ucture Browse & Search Search Show All email csev@umich.edu olga@umich.edu | Directory (Select Profile Database) Go Execute SQL DB Settings Add Duplicate Edit Dele Chuck Chuck Chuck |
| sql1.sqlite Str Master Table Tables (1) Users rowid Views (0) 1 Indexes (0) 2 Triggers (0) 3 | ucture Browse & Search Search Show All email csev@umich.edu olga@umich.edu | Execute SQL DB Settings Add Duplicate Edit Dele name Chuck Olse |
| Master Table Tables (1) Users Views (0) Indexes (0) Triggers (0) Table Users rowid 1 2 4 | Search Show All email csev@umich.edu olga@umich.edu | Add Duplicate Edit Dele |
| | gab@umich.edu gaurav@umich.edu | Gab Gaurav |
| SQLite 3.7.17 Gecko 24.0 0.7.7 Shared Number of Rows R | | |
| | 1 to 4 of 4 > | >> |



• Allows the updating of a field with a where clause

update Users set name='Charles' where email='csev@umich.edu'

| 000 | SQLite Manager – /Users/c | ev/sql1.sqlite | | | | | | | | |
|--|---|---|--|--|--|--|--|--|--|--|
| 🖉 💥 🗋 | 逆 🔏 f(x) 🛒 醋 📑 📑 | Directory (Select Profile Database) | | | | | | | | |
| sql1.sqlite 🛔 | Structure Browse & Sear | h Execute SQL DB Settings | | | | | | | | |
| Master Table Tables (1) Users | Enter SQL update Users set name='Charles' where email='csev@um | ch.edu' | | | | | | | | |
| Views (0) Indexes (0) Triggers (0) | Run SQL Actions V Last Error: not an error | not an error | | | | | | | | |
| P Higgers (0) | | SQLite Manager – /User | rs/csev/sql1.sqlite | | | | | | | |
| 4 | sall salite | Structure Browse & S | earch Execute SQL DB Settings | | | | | | | |
| **** | Master Tal Tables (1) Users Views (0) Indexes (0) Triggers (0) | Ie TABLE Users Search SI rowid email 1 csev@um 2 olga@um 3 gab@umi 4 gaurav@u | now All Add Duplicate fit Dele name ich.edu Charles ich.edu Olga ich.edu Gab umich.edu Gaurav | | | | | | | |
| SQLite 3.7.17 Geck | o 24.0 0.7.7 Shared Number of Row | • | | | | | | | | |
| | | <pre> << < 1 to 4 of 4</pre> | | | | | | | | |
| | SOLite 3.7.17 | Gecko 24.0 0.7.7 Shared Number of Rows Returned | : 0 ET: 0 ms | | | | | | | |

Retrieving Records: Select

 The select statement 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'

| OOO SQLite Manager - /Use | s/csev/sql1.sqlite | | | | | | |
|---|----------------------|----------------------------|--|------|--|--|--|
| 🖸 💥 🗋 🚔 🐔 f(x) 📑 醋 醋 | Directory (Select | Profile Database) 🗍 | \$ Go | | | | |
| sql1.sqlite + Structure Browse & | earch Execute SQL DB | Settings | | | | | |
| Master Table Tables (1) Users Liews (0) Enter SQL select * from Users | | | | | | | |
| ► Indexes (0) Run SQL Actions ▼ Last Error: not a | error | | | | | | |
| Triggers (0) | name | | THE SECOND SECONDO SECOND SECONDO SECOND SECOND SECONDO SECON | | | | |
| creav@umich.edu | Charles | | | | | | |
| a olga@umich.edu | Olga | | | | | | |
| gab@umich.edu | Gab | | | | | | |
| gaurav@umich.edu | Gaurav | | | | | | |
| | | | | | | | |
| | 000 | SQLite Ma | Manager – /Users/csev/sql1.sqlite | | | | |
| | 🖾 💥 🗋 📂 🔏 | f(x) 📑 📝 | Directory 🕨 (Select Profile Database) | ‡ Go | | | |
| | sql1.sqlite 🛊 | Structure | Browse & Search Execute SQL DB Settings | | | | |
| | Master Table | | | | | | |
| | Tables (1) | 2L | | | | | |
| | Users select * | rom Users where email='cse | csev@umich.edu' | | | | |
| | Indexes (0) Run S | QL Actions v Las | ast Error: not an error | | | | |
| SOLite 3 7 17 Cecko 24 0 0 7 7 Shared Number of Pows Peturner | F Triggers (0) | | | | | | |
| Source Sinit A decko 24.0 0.00 Shared Humber of Kows Recarded | email csev@up | ich edu | Charles | L¥ | | | |
| | 4 CSEV@001 | nemedu | Charles | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | 4 | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

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
select * from Users order by name

| 0 | 00 | | | | | SQ | Lite Ma | nager – | /Users/cs | ev/sql1.sqli | te | | | - |
|----------|-------------------------|-----|------|------------|-----------|----------|----------|----------|--------------|--------------|---------|--------------------------|-------|-----|
| Ø | \gg | | | ž 🏅 | f(x) | Ē | Ħ | Ē | T | Directory | / ► | (Select Profile Database |) ‡ | Go |
| sq | 1.sqlite | ÷ | | | | Strue | cture | Brow | se & Searcl | n Execu | ute SQL | DB Settings | | |
| • | Master Ta Tables (1) | ble | | Enter SC | ĮL | | | | | | | | | |
| | Users | | | select * f | rom Users | order by | email | | | | | | | |
| F | Indexes (0) | 0) | | Run S | | Actions | ▼ Last | t Error: | not an error | | | | | |
| ▶ : | Triggers (| (0) | | | | | | | | | | | | |
| | | | | email | | | | | | name | | | | EŞ. |
| | | | | csev@um | ich.edu | | | | | Charles | | | | |
| | | | | gab@umi | ch.edu | | | | | Gab | | | | |
| | | | | gaurav@u | mich.edu | | | | | Gaurav | | | | |
| | | | 4 | olga@um | ich.edu | | | | | Olga | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | 4 | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | |
| SQ | Lite 3.7.1 | 7 G | ecko | 24.0 0.7 | .7 Sha | red N | umber of | Rows Re | turned: 4 | | | | ET: 0 | ms |

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
 select * from Users where email='csev@umich.edu'
 select * from Users order by email

This is not too exciting (so far)

• Tables pretty much look like big fast programmable spreadsheet with rows, columns, and commands

• The power comes when we have more than one table and we can exploit the relationships between the tables

Complex Data Models and Relationships

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

Database Design

- Database design is an art form of its own with particular skills and experience
- Our goal is to avoid the really bad mistakes and design clean and easily understood databases
- Others may performance tune things later
- Database design starts with a picture...

Event EventParticipant Person Attributes Attributes Attributes detailDescription role lastName Relationships eventID firstName startTime person notes 6-6 Relationships >> event name > participatingEvents endTime date Relationships Location participants Attributes location ~ detailDescription name Relationships >events Occasion Attributes name detailDescription startDate endDate Relationships events


Building a Data Model

- Drawing a picture of the data objects for our application and then figuring out how to represent the objects and their relationships
- Basic Rule: Don't put the same string data in twice use a relationship instead
- When there is one thing in the "real world" there should be one copy of that thing in the database

| Track | Len | Artist | Album | Genre | Rating | Count |
|--------------------------------|------|---------------|-------------------|-------------|--------|-------|
| Hells Bells | 5:13 | AC/DC | Who Made Who | Rock | ***** | 61 |
| Shake Your Foundations | 3:54 | AC/DC | Who Made Who | Rock | ***** | 70 |
| Chase the Ace | 3:01 | AC/DC | Who Made Who | Rock | | 56 |
| For Those About To Rock (We | 5:54 | AC/DC | Who Made Who | Rock | ***** | 61 |
| 🛛 Dúlamán | 3:43 | Altan | Natural Wonders M | New Age | | 31 |
| Rode Across the Desert | 4:10 | America | Greatest Hits | Easy Listen | ***** | 23 |
| Now You Are Gone | 3:08 | America | Greatest Hits | Easy Listen | ***** | 18 |
| I Tin Man | 3:30 | America | Greatest Hits | Easy Listen | ***** | 23 |
| Sister Golden Hair | 3:22 | America | Greatest Hits | Easy Listen | ***** | 24 |
| Track 01 | 4:22 | Billy Price | Danger Zone | Blues/R&B | ***** | 26 |
| Track 02 | 2:45 | Billy Price | Danger Zone | Blues/R&B | ***** | 18 |
| Track 03 | 3:26 | Billy Price | Danger Zone | Blues/R&B | ***** | 22 |
| Track 04 | 4:17 | Billy Price | Danger Zone | Blues/R&B | ***** | 18 |
| Track 05 | 3:50 | Billy Price | Danger Zone | Blues/R&B | ***** | 21 |
| War Pigs/Luke's Wall | 7:58 | Black Sabbath | Paranoid | Metal | ***** | 25 |
| Paranoid | 2:53 | Black Sabbath | Paranoid | Metal | ***** | 22 |
| Planet Caravan | 4:35 | Black Sabbath | Paranoid | Metal | ***** | 25 |
| Iron Man | 5:59 | Black Sabbath | Paranoid | Metal | ***** | 26 |
| Electric Funeral | 4:53 | Black Sabbath | Paranoid | Metal | ***** | 22 |
| Hand of Doom | 7:10 | Black Sabbath | Paranoid | Metal | ***** | 23 |
| 🛛 Rat Salad | 2:30 | Black Sabbath | Paranoid | Metal | ***** | 31 |
| Jack the Stripper/Fairies Wear | 6:14 | Black Sabbath | Paranoid | Metal | ***** | 24 |
| Bomb Squad (TECH) | 3:28 | Brent | Brent's Album | | | 1 |
| clay techno | 4:36 | Brent | Brent's Album | | | 2 |
| Heavy | 3:08 | Brent | Brent's Album | | | 1 |
| I Hi metal man | 4:20 | Brent | Brent's Album | | | 1 |
| Mistro | 2:58 | Brent | Brent's Album | | | 1 |

For each "piece of info"...

| Is the column a attribute of and | Len | lbum | | | | |
|--|-------------------|-------------------------|-------------------|-------------|-------|--------|
| Once we define to define the rest | e obje elation | cts we ne ships bety | ed ween | Artist | t | Rating |
| objects. | | | | Track | Coun | t |
| ✓ Hells Bells | 5:13 | AC/DC | Who Made Who | Rock | ***** | 61 |
| Shake Your Foundations | 3:54 | AC/DC | Who Made Who | Rock | ***** | 70 |
| Chase the Ace | 3:01 | AC/DC | Who Made Who | Rock | | 56 |
| For Those About To Rock (We | 5:54 | AC/DC | Who Made Who | Rock | ***** | 61 |
| 🗹 Dúlamán | 3:43 | Altan | Natural Wonders M | New Age | | 31 |
| Rode Across the Desert | 4:10 | America | Greatest Hits | Easy Listen | ***** | 23 |
| Now You Are Gone | 3:08 | America | Greatest Hits | Easy Listen | ***** | 18 |
| | | | | | | |



| Art | ist | belo | ngs-to Album bel Gen | ongs-to re | Track Rating Len Count belongs-to | |
|-----------------------------|------|---------|----------------------------|---------------|---|----|
| ✓ Hells Bells | 5:13 | AC/DC | Who Made Who | Rock | **** | 61 |
| Shake Your Foundations | 3:54 | AC/DC | Who Made Who | Rock | **** | 70 |
| ☑ Chase the Ace | 3:01 | AC/DC | Who Made Who | Rock | | 56 |
| For Those About To Rock (We | 5:54 | AC/DC | Who Made Who | Rock | **** | 61 |
| 🗹 Dúlamán | 3:43 | Altan | Natural Wonders M | New Age | | 31 |
| Rode Across the Desert | 4:10 | America | Greatest Hits | Easy Listen | **** | 23 |
| Now You Are Gone | 3:08 | America | Greatest Hits | Easy Listen | **** | 18 |
| | 2.20 | A | Constant Illes | Frank Links | | |

Representing Relationships in a Database

| ✓ Hells Bells | 5:13 | AC/DC | Who Made Who | Rock | ***** | 61 |
|-----------------------------|------|---------|-------------------|-------------|-------|----|
| Shake Your Foundations | 3:54 | AC/DC | Who Made Who | Rock | ***** | 70 |
| Chase the Ace | 3:01 | AC/DC | Who Made Who | Rock | | 56 |
| For Those About To Rock (We | 5:54 | AC/DC | Who Made Who | Rock | ***** | 61 |
| 🗹 Dúlamán | 3:43 | Altan | Natural Wonders M | New Age | | 31 |
| Rode Across the Desert | 4.10 | America | Greatest Hits | Easy Listen | **** | 23 |
| Now You Are Gone | 3:08 | America | Greatest Hits | Easy Listen | ***** | 18 |
| E Tin Man | 2.20 | Amorica | Createst Hits | Encu Liston | | 22 |

We want to keep track of which band is the "creator" of each music track... What album does this song "belong to"??

Which album is this song related to?

Database Normalization (3NF)

- There is *tons* of database theory way too much to understand without excessive predicate calculus
 - Do not replicate data reference data point at data
 - Use integers for keys and for references
 - Add a special "key" column to each table which we will make references to. By convention many programmers call this column "id"

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





Finding our way around....

Three Kinds of Keys

- Primary key generally an integer autoinrcement field
- Logical key What the outside world uses for lookup
- Foreign key generally an integer key point to a row in another table

| Site | |
|---------|--|
| id | |
| title | |
| user_id | |
| ••• | |

Primary Key Rules

- Best practices
- Never use your logical key as the primary key
- Logical keys can and do change albeit slowly
- Relationships that are based on matching string fields are far less efficient than integers performance-wise

User id login password name email created_at modified_at login_at

Foreign Keys

- A foreign key is when a table has a column that contains a key which points the primary key of another table.
- When all primary keys are integers, then all foreign keys are integers this is good - very good
- If you use strings as foreign keys you show yourself to be an uncultured swine



Relationship Building (in tables)

| Art | ist | belo | ngs-to Album bel Gen | ongs-to re | Track Rating Len Count belongs-to | |
|-----------------------------|------|---------|----------------------------|---------------|---|----|
| ✓ Hells Bells | 5:13 | AC/DC | Who Made Who | Rock | **** | 61 |
| Shake Your Foundations | 3:54 | AC/DC | Who Made Who | Rock | **** | 70 |
| ☑ Chase the Ace | 3:01 | AC/DC | Who Made Who | Rock | | 56 |
| For Those About To Rock (We | 5:54 | AC/DC | Who Made Who | Rock | **** | 61 |
| 🗹 Dúlamán | 3:43 | Altan | Natural Wonders M | New Age | | 31 |
| Rode Across the Desert | 4:10 | America | Greatest Hits | Easy Listen | **** | 23 |
| Now You Are Gone | 3:08 | America | Greatest Hits | Easy Listen | **** | 18 |
| | 2.20 | A | Constant Illes | Frank Links | | |





| 0 | 00 | | | SQLite Manager - | - /Users/ | csev/sql1.s | qlite | | |
|----------|---|----------------|----------------------|------------------|------------|----------------|------------------|----------------------------------|----------|
| ¢ so | X al1.sqlite | | Database: | main 🛟 | Table Name | e: Artist | | | \$ Go |
| ► ▼ | Master Ta Tables (1 | Define Column | 16 | Temporar | y table 🗌 | If Not Exists | | | |
| b | sqlite, Views (0) Indexes (| Column Name | Data Type INTEGER | Primary Key? | Autoinc? | Allow Null? | Unique? ✓ Yes | Default Value | |
| | Triggers | name | TEXT | ▼ Yes | Yes | ☑ Yes ☑ Yes | Yes | ▼ | |
| | | | | Ves | Yes | Ves | Yes | ▼ | |
| | | | | ▼ □ Yes | O Yes | ✓ Yes | O Yes | | |
| | | | | Ves | Yes | ✓ Yes ✓ Yes | Yes | ▼ ▼ | |
| | | | | ▼ Yes | Yes | ☑ Yes ☑ Yes | Yes | ▼ ▼ | |
| | | | | | | | (| Cancel OK | |
| SC | Lite 3.7.17 | 7 Gecko 24.0 0 | .7.7 Shared | Number of Rows R | eturned: 0 | | | | ET: 0 ms |

| 000 | SQLite Manag | ger – /Users/csev/sql1.s | lite | |
|--|---|--------------------------|---|-----------------------|
| 🐼 🏂 🖻 🞽 💰 | f(x) 📑 👬 🚺 | 📑 📑 Directo | ry 🕨 (Select Profile D | atabase) 🗍 Go |
| sql1.sqlite | Structure | Browse & Search | Execute SQL DB Se | ettings |
| Master Table (1) Tables (5) Album Artist Genre Track sqlite_sequence Views (0) Indexes (4) Triggers (0) | TABLE: Genre Drop Employ Create statement CREATE TABLE "Genre" (* TEXT) More Info No. of Records: 0 Columns (2) | pty Rename | Reindex Copy JTOINCREMENT NOT NULL U No. of Triggers: 0 | Export NIQUE , "name" |
| | Column ID Name | Type Not Nul | Default Value | Primary Key 🛛 🖽 |
| | 0 id | INTEGER 1 | null 1 | L |
| | 1 name | TEXT 0 | null 0 |) |
| | Name | Type Not Nu | ll Default | |
| | | ▼ □ | Add Colum | n |
| | | | | |
| SQLite 3.7.17 Gecko 24.0 0. | 7.7 Shared Number of Rov | vs Returned: 0 | | ET: 0 ms |

| 000 | SQLite | Manager – /Users | /csev/sql1.sqlit | e | | |
|--|--|--|------------------|--|----------------------------------|------|
| 🖉 💥 🗋 🞽 🏄 | f(x) | * 💣 🕈 | Directory | (Select Pro | file Database) | ¢ Go |
| sql1.sqlite | St | ructure Brows | e & Search | Execute SQL | DB Settings | |
| Master Table (1) Tables (5) Album Artist Genre Track sqlite_sequence Views (0) Indexes (4) Triggers (0) | TABLE: Album Drop Create stateme CREATE TABLE INTEGER, "title" More Info No. of Records: | Empty nt Album" ("id" INTEGER FEXT) 0 No. of | Rename Re | OINCREMENT NOT NOT NOT NOT NOT NOT NOT NOT NOT N | Export ULL UNIQUE , artist_id | |
| | Column ID | Nama | Not Null | Default Value | Drimony Kay | |
| | Column ID | id INTEC | FR 1 | null | Primary Key | C\$ |
| | 1 | artist id INTEC | ER 0 | null | 0 | |
| | 2 | title TEXT | 0 | null | 0 | |
| | Name | Туре | Not Null | Default | | |
| | | | | Add | Column | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| 000 | SQLit | te Manager - | - /Users/cse | v/sql1.sqlit | e | | | |
|--|--|---|--------------|--------------|-----------------|-----------------|------|--|
| 🖾 💥 🗅 🚔 🖌 | f(x) | * | F | Directory | (Select Pro | ofile Database) | ‡ Go | |
| sql1.sqlite | | tructure | Browse & S | Search | Execute SQL | DB Settings | | |
| Master Table (1) Tables (5) Album Artist Genre Track sqlite_sequence Views (0) Indexes (4) Triggers (0) | TABLE: Track Drop Create statem CREATE TABLE INTEGER, genr | TABLE: Track Drop Empty Rename Reindex Copy Export Create statement 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) More Info | | | | | | |
| 4 | No. of Records | . 0 | No. of Index | kes: 1 | No. of Triggers | .: 0 | | |
| | Columns (7) | | | | 1.5.6.4.4.4 | | | |
| | Column ID | Name | Туре | Not Null | Default Value | Primary Key | E\$ | |
| | 1 | album id | INTEGER | 0 | null | 0 | | |
| | 2 | genre id | INTEGER | 0 | null | 0 | | |
| | 3 | len | INTEGER | 0 | null | 0 | | |
| | 4 | rating | INTEGER | 0 | null | 0 | | |
| | 5 | title | TEXT | 0 | null | 0 | | |
| | 6 | count | INTEGER | 0 | null | 0 | | |
| | Name | т | ype | Not Null | Default | | | |
| | | Г | | | Add | Column | | |
| | | L | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |



| 000 | SQLite Manager – /Users/csev/sql1.sqlite | | | | | | | | | | |
|--|--|---|----------|--|--|--|--|--|--|--|--|
| 🖾 💥 🗋 🖻 | £ | f(x) $f(x)$ Directory (Select Profile Database) | ‡ Go | | | | | | | | |
| sql1.sqlite 🛔 | | Structure Browse & Search Execute SQL DB Settings | | | | | | | | | |
| Master Table (1) Tables (5) | 1 | Enter SQL | | | | | | | | | |
| Album Artist | | select * from Genre | | | | | | | | | |
| Genre Track | | Run SQL Actions V Last Error: not an error | | | | | | | | | |
| sqlite_sequence | | id name | Ę | | | | | | | | |
| Views (0) Indexes (4) | | 1 Rock 2 Metal | | | | | | | | | |
| Triggers (0) | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| SQLite 3.7.17 Gecko 24.0 | 0.7.7 | Shared Number of Rows Returned: 2 | ET: 1 ms | | | | | | | | |

insert into Genre (name) values ('Rock') insert into Genre (name) values ('Metal')

| 000 | | SQL | ite Manager - | /Users/csev | /sql1.sq | qlite | | | | | |
|--|-------|--|-------------------|---------------------|------------|-------|---------|--------------------|-------------------|-------|----|
| 🖉 💥 🗋 📂 | £ | <i>f(x)</i> ■ | F | T | Director | ry | • | Select | Profile Database) | ÷ | Go |
| sql1.sqlite | | | Structure | Browse & Se | earch | Exe | cute SC | QL | DB Settings | | |
| Master Table (1) Tables (5) Album Artist Genre | | Enter SQL select * from Al Run SQL | bum; Actions 🔻 | Last Error: | not an err | or | | | | | |
| sqlite_sequence Views (0) Indexes (4) Triggers (0) | • | id 1 2 | | artist_id 2 1 | | | | title Who IV | a o Made Who | | Ð |
| | | | | | | | | | | | |
| SQLite 3.7.17 Gecko 24.0 | 0.7.7 | Shared Nu | mber of Rows Re | turned: 2 | | | | | | ET: 0 | ms |

insert into Album (title, artist_id) values ('Who Made Who', 2) insert into Album (title, artist_id) values ('IV', I) 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)

| id | album_id | genre_id | len | rating | title | count |
|----|----------|----------|-----|--------|---------------|-------|
| 1 | 2 | 1 | 297 | 5 | Black Dog | 0 |
| 2 | 2 | 1 | 482 | 5 | Stairway | 0 |
| 3 | 1 | 2 | 313 | 5 | About to Rock | 0 |
| 4 | 1 | 2 | 207 | 5 | Who Made W | 0 |
| | | | | | | |

We have relationships!

| | | | | | • | | |
|----|----------|---------|---------|--------|---------------|-------|-------|
| id | album_ic | d genre | _id len | rating | title | count | |
| 1 | 2 | 1 | 297 | 5 | Black Dog | 0 | |
| 2 | 2 | 1 | 482 | 5 | Stairway | 0 | |
| 3 | 1 | 2 | 313 | 5 | About to Rock | C 0 | |
| 4 | 1 | 2 | 207 | 5 | Who Made W | . 0 | Irack |



id Use 'control + ;' to execute the query. 1 Led Zepplin 2 AC/DC Artist

id name 1 Rock 2 Metal Genre

Using Join Across Tables

http://en.wikipedia.org/wiki/Join_(SQL)

Relational Power

- By removing the replicated data and replacing it with references to a single copy of each bit of data we build a "web" of information that the relational database can read through very quickly even for very large amounts of data
- Often when you want some data it comes from a number of tables linked by these foreign keys

The JOIN Operation

- The JOIN operation links across several tables as part of a select operation
- You must tell the JOIN how to use the keys that make the connection between the tables using an ON clause



select Album.title,Artist.name from Album join Artist on Album.artist_id = Artist.id



The tables which hold the data

How the tables are linked

| id | artist_id | title | id | Use 'control + ;' to execute the query. | |
|----|-----------|--------------|----|---|--|
| 1 | 2 | Who Made Who | 1 | | |
| 2 | 1 | IV | 2 | AC/DC | |
| | | | | | |
| | | | | | |

| Album.title | Album.artist_id / | Artist.id | Artist.name |
|--------------|-------------------|-----------|-------------|
| title | artist_id | id | name |
| Who Made Who | 2 | 2 | AC/DC |
| IV | 1 | 1 | Led Zepplin |

select Album.title, Album.artist_id, Artist.id, Artist.name
from Album join Artist on Album.artist_id = Artist.id

| | | | | | e | | |
|----|----------|----------|-----|--------|---------------|-------|--|
| id | album_id | genre_id | len | rating | title | count | |
| 1 | 2 | 1 | 297 | 5 | Black Dog | 0 | |
| 2 | 2 | 1 | 482 | 5 | Stairway | 0 | |
| 3 | 1 | 2 | 313 | 5 | About to Rock | 0 | |
| 4 | 1 | 2 | 207 | 5 | Who Made W | 0 | |
| | | | | | | | |

| title | name | | |
|---------------|-------|----|-------|
| Black Dog | Rock | id | name |
| Stairway | Rock | 1 | Rock |
| About to Rock | Metal | 2 | Metal |
| Who Made Who | Metal | | |
| | | | |

select Track.title, Genre.name from Track join Genre on Track.genre_id = Genre.id

What we want to see

The tables which hold the data

7

How the tables are linked

It can get complex...

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

| title | name | title | name |
|---------------|-------------|--------------|-------|
| Black Dog | Led Zepplin | IV | Rock |
| Stairway | Led Zepplin | IV | Rock |
| About to Rock | AC/DC | Who Made Who | Metal |
| Who Made Who | AC/DC | Who Made Who | Metal |
| | | | |

What we want to see

The tables which hold the data

How the tables are linked

| Hells Bells | | 5:13 | AC/DC | Who Made | Who | Rock | ***** | 61 | |
|--|--|----------|---------------|-------------|----------|---------------|-------|--------|--|
| Shake Your Foundation | ons | 3:54 | AC/DC | Who Made | Who | Rock | ***** | 70 | |
| Chase the Ace | | 3:01 | AC/DC | Who Made | Who | Rock | | 56 | |
| For Those About To | Rock (We | 5:54 | AC/DC | Who Made | Who | Rock | ***** | 61 | |
| Dúlamán | | 3:43 | Altan | Natural Wo | onders M | New Age | | 31 | |
| Rode Across the Dese | ert | 4:10 | America | Greatest H | its | Easy Listen | ***** | 23 | |
| Now You Are Gone | | 3:08 | America | Greatest H | its | Easy Listen | ***** | 18 | |
| 🗹 Tin Man | | 3:30 | America | Greatest H | its | Easy Listen | ***** | 23 | |
| Sister Golden Hair | | 3:22 | America | Greatest H | its | Easy Listen | ***** | 24 | |
| Track 01 | | 4:22 | Billy Price | Danger Zo | ne | Blues/R&B | ***** | 26 | |
| Track 02 | | 2:45 | Billy Price | Danger Zo | ne | Blues/R&B | ***** | 18 | |
| Track 03 | | 3:26 | Billy Price | Danger Zo | ne | Blues/R&B | ***** | 22 | |
| ☑ Track 04 | | 4:17 | Billy Price | Danger Zo | ne | Blues/R&B | ***** | 18 | |
| ✓ Track 05 | | 3:50 | Billy Price | Danger Zo | ne | Blues/R&B | ***** | 21 | |
| ☑ War Pigs/Luke's Wall | | 7:58 | Black Sabbath | n Paranoid | | Metal | **** | 25 | |
| Paranoid | | 2:53 | Black Sabbath | n Paranoid | | Metal | ***** | 22 | |
| Planet Caravan | title | | | - | +i+l. | | | 2.02.0 | |
| 🗹 Iron Man | uue | | 1 | lame | uu | e | 1 | lame | |
| Electric Funeral | Black D |)oa | L | ed Zepplin | IV | IV | | Rock | |
| Hand of Doom | Hand of Doom Rat Salad Jack the Stripper/Fai | | | | | | | | |
| Rat Salad | | | L | ed Zepplin | IV | | R | ock | |
| ✓ Jack the Stripper/Fai | | | Δ | C/DC | Who | Who Made Who | | letal | |
| ✓ Bomb Squad (TECH) ✓ clay techno ✓ Who Made Who | | ~ | (C/DC | who made | | ide who Metai | | | |
| | | Made Who | | AC/DC | | Who Made Who | | letal | |
| ✓ Heavy | | | | - | | | | | |
| ✓ Hi metal man | | 4:20 | Brent | Brent's Alb | um | | | 1 | |
| Mistro | | 2:58 | Brent | Brent's Alb | um | | | 1 | |

Complexity Enables Speed

- Complexity makes speed possible and allows you to get very fast results as the data size grows.
- By normalizing the data and linking it with integer keys, the overall amount of data which the relational database must *scan* is far lower than if the data were simply flattened out.
- It might seem like a tradeoff spend some time designing your database so it continues to be fast when your application is a success
Additional SQL Topics

- Indexes improve access performance for things like string fields
- Constraints on data (cannot be NULL, etc..)
- Transactions allow SQL operations to be grouped and done as a unit
- See SI664 Database Design (All Semesters)

Summary

- Relational databases allow us to scale to very large amounts of data
- The key is to have one copy of any data element and use relations and joins to link the data to multiple places
- This greatly reduces the amount of data which much be scanned when doing complex operations across large amounts of data
- Database and SQL design is a bit of an art-form