

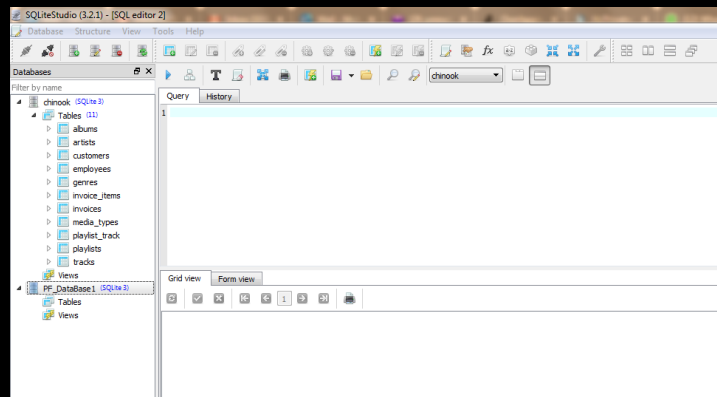
Object Oriented Programming, Namespaces and Classes

- Modules as Namespaces
- Classes as Namespaces
- Databases and SQL
- Python Database Programming

Acknowledgement

- Slides in this presentation are taken from online available slides of:
Charles Severance
 - <http://www.pythonlearn.com/>

SQLite Studio



<https://sqlitestudio.pl/index.rvt>

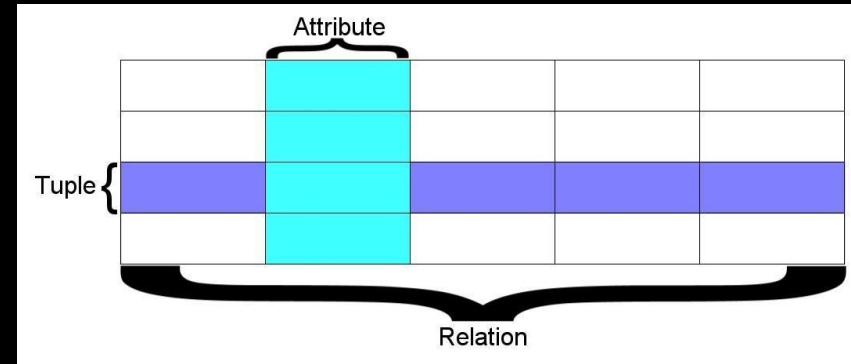
Relational Databases

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)** - a set of fields that 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)

Grid view Form view

Filter data Total rows loaded

	AlbumId	Title	ArtistId
1	1	For Those About To Rock We Salute You	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cellos	7
10	10	Audioslave	8
11	11	Out Of Exile	8
12	12	BackBeat Soundtrack	9
13	13	The Best Of Billy Cobham	10

Columns / Attributes

Rows / Tuples

Tables /
Relations

Structured Query Language

Database files are not read from or written to by an application program using the usual file input/output interface. They typically are also not accessed directly. Instead, the application program usually sends commands to a special type of server program called a *database engine* or a *database management system* that manages the database; that program will access the database file on the application's behalf. The commands accepted by database engines are statements written in a query language, the most popular of which is called *Structured Query Language*, typically referred to as *SQL*.

SQL

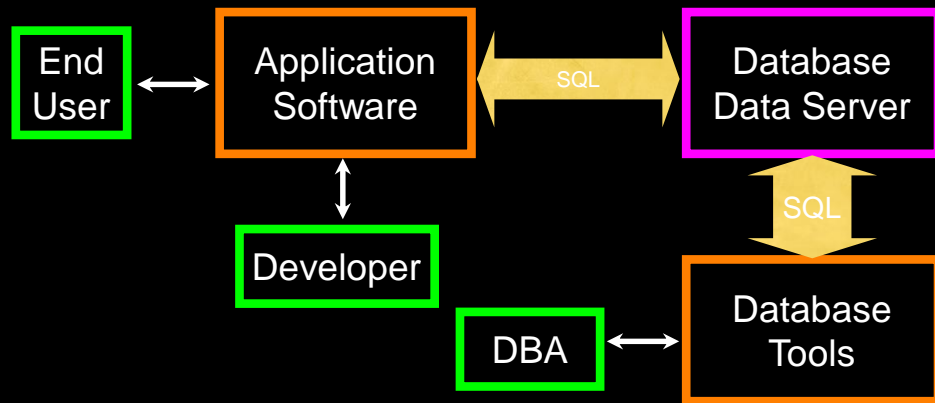
- **Structured Query Language** is the language we use to issue commands to the database
 - Create a table
 - Retrieve some data
 - Insert data
 - Delete data

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

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"

Large Project Structure

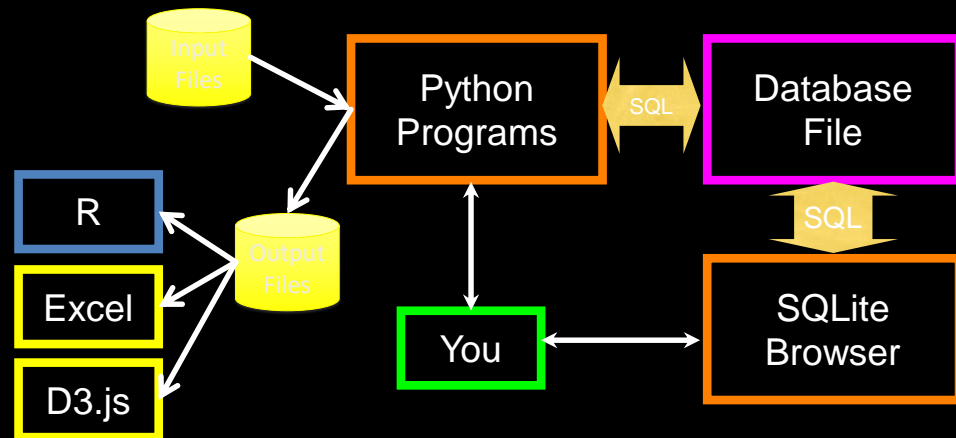


Database Administrator (DBA)

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, coordinate, and implement security measures to safeguard the database.

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

Data Analysis Structure



Database Model

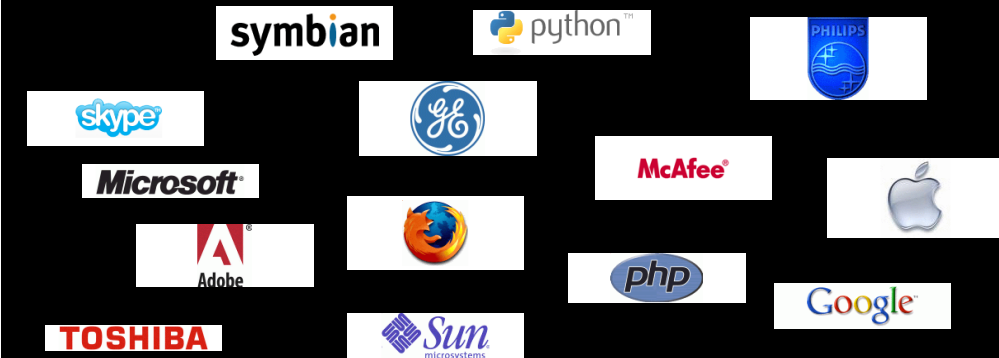
A **database model** or **database schema** is the **structure or format of a database**, described in a formal language supported by the database management system. In other words, a “database model” is the application of a data model when used in conjunction with a database management system.

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

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 is in lots of software...

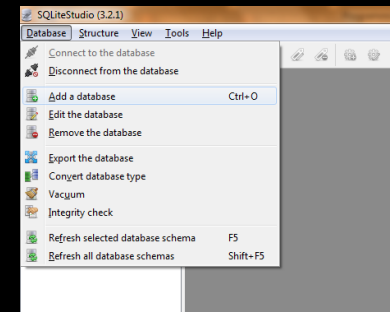


<http://www.sqlite.org/famous.html>

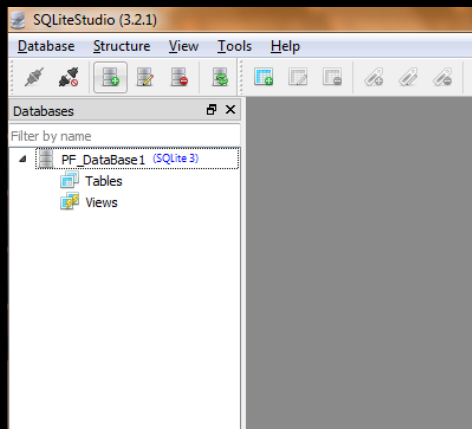
SQLite Studio

- SQLite is a very popular database - it is free and fast and small
- SQLite Studio allows us to directly manipulate SQLite files
- <https://sqlitestudio.pl/>
- SQLite is embedded in Python and a number of other languages

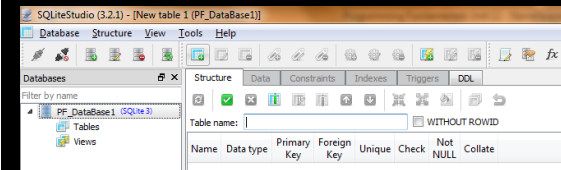
Open/Create a Database



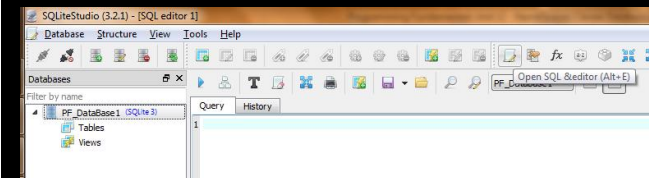
```
>>> import sqlite3
>>> con = sqlite3.connect('PF_Database1.db')
>>> cur = con.cursor()
```

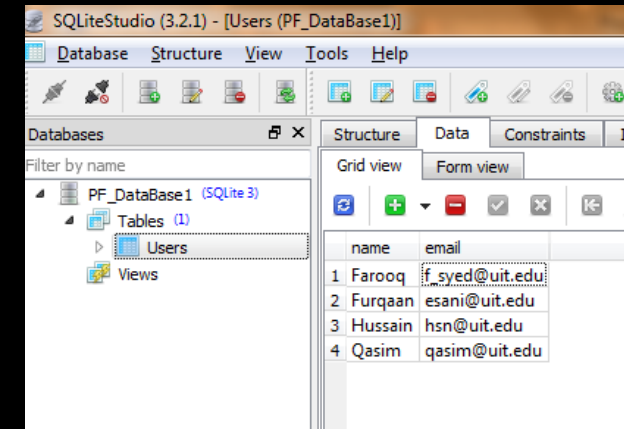
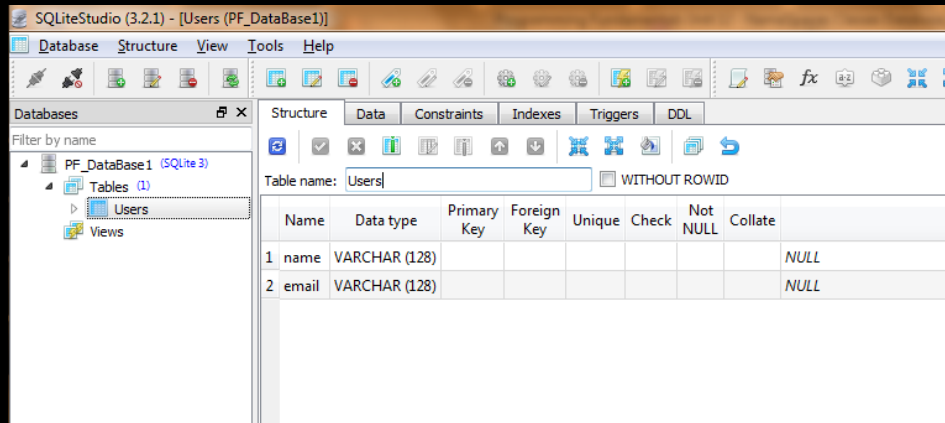


Start Simple - A Single Table



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





SQL

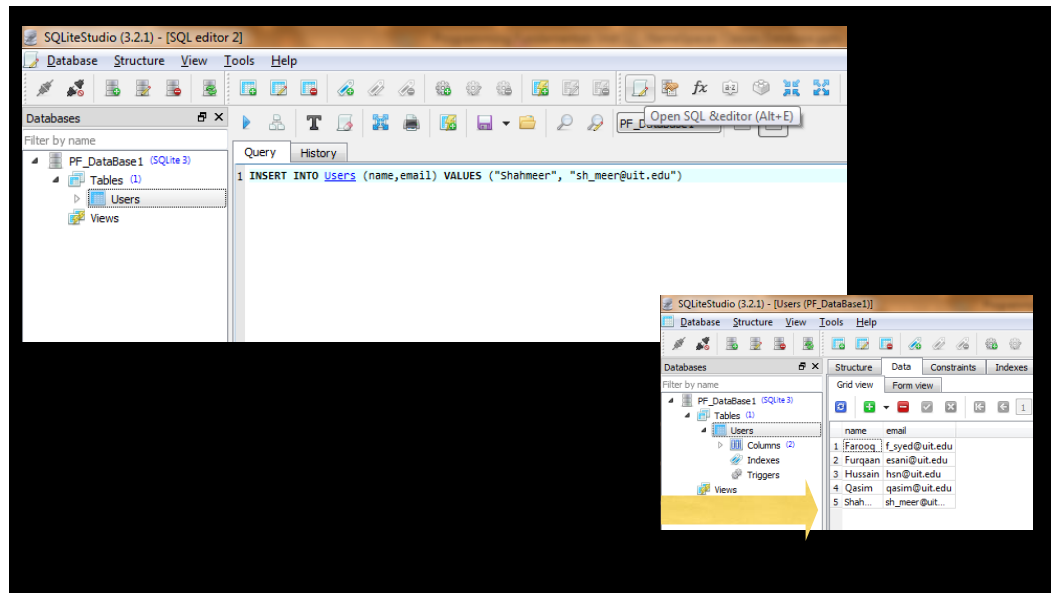
- **Structured Query Language** is the language we use to issue commands to the database
 - Create a table
 - Retrieve some data
 - Insert data
 - Delete data

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

SQL Insert

- The Insert statement inserts a row into a table

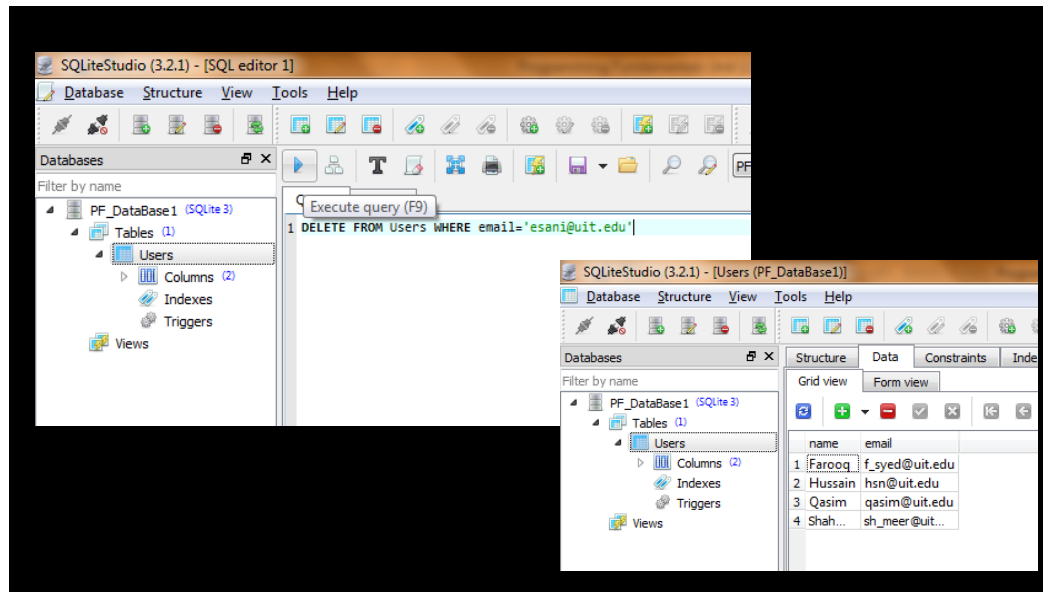
```
INSERT INTO Users (name, email) VALUES ('Shahmeer', 'sh_meer@uit.edu')
```



SQL Delete

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

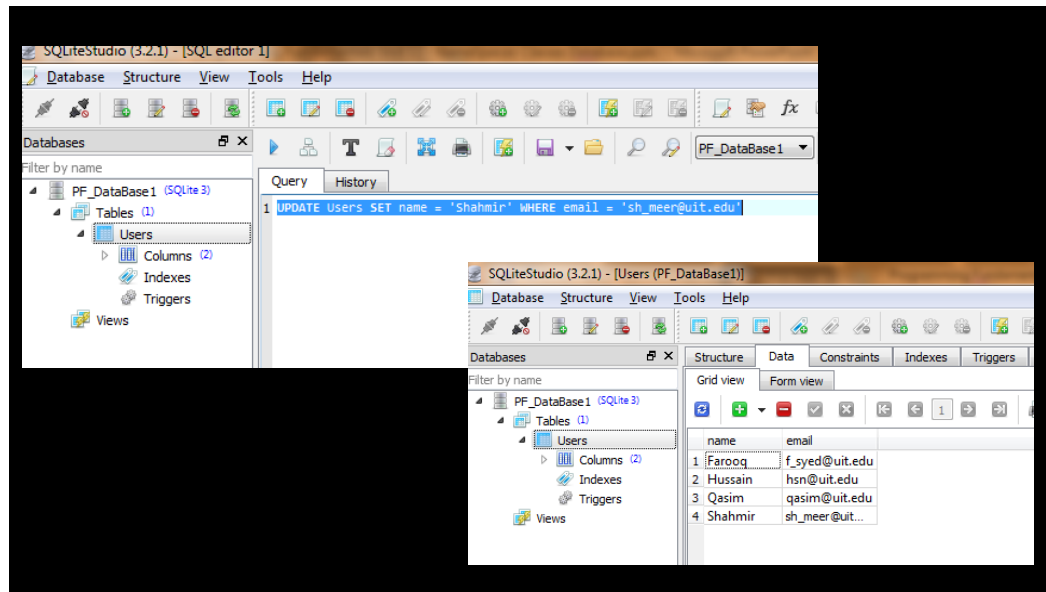
DELETE FROM Users WHERE email='esani@uit.edu'



SQL: Update

- Allows the updating of a field with a where clause

UPDATE Users SET name='Shahmir' WHERE email='sh_meer@uit.edu'



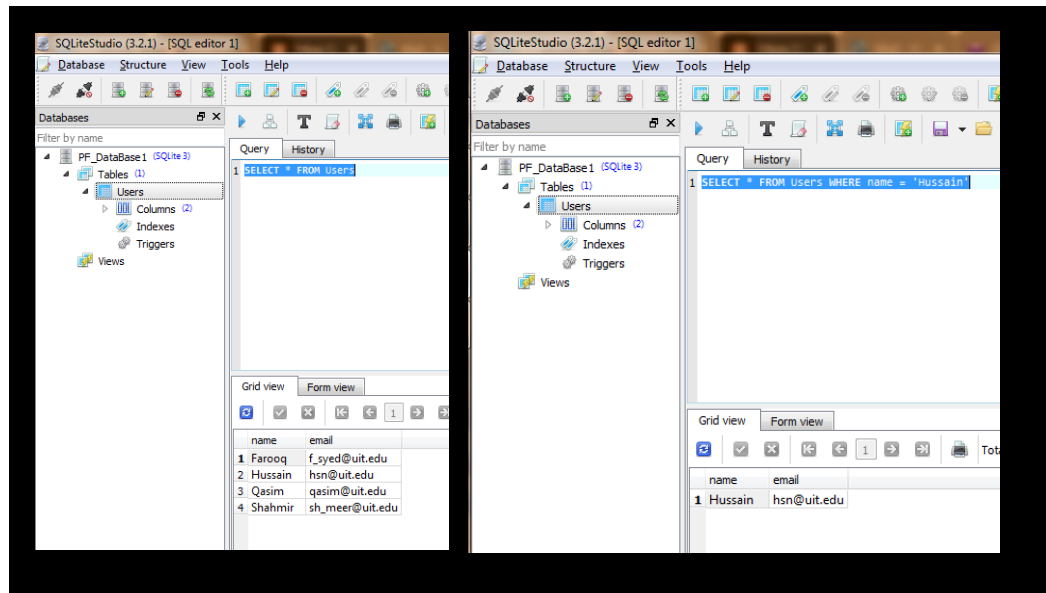
Retrieving Records: Select

- 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='f_syed@uit.edu'

Additionally for Python: `cur.fetchall()`

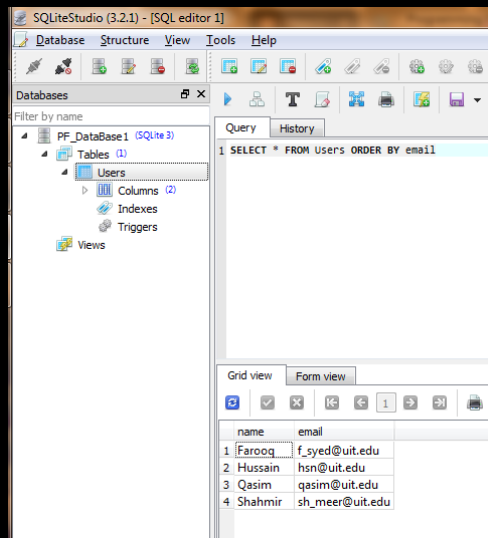


Sorting with ORDER BY

- 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



SQL Summary

```
INSERT INTO Users (name, email) VALUES ('Kristin', '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
```

This is not too exciting (so far)

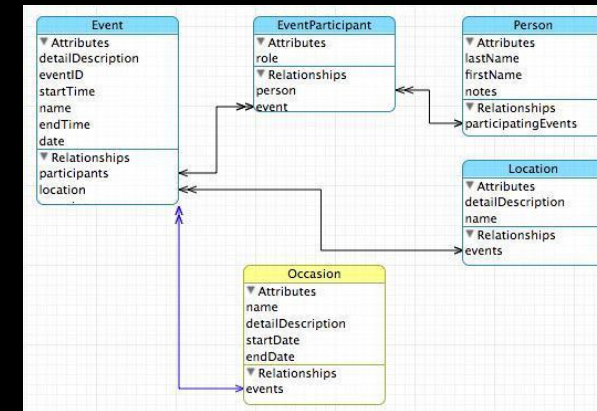
- Tables pretty much look like big fast programmable spreadsheets with rows, columns, and commands
- The power comes when we have more than one table and we can exploit the relationships between the tables

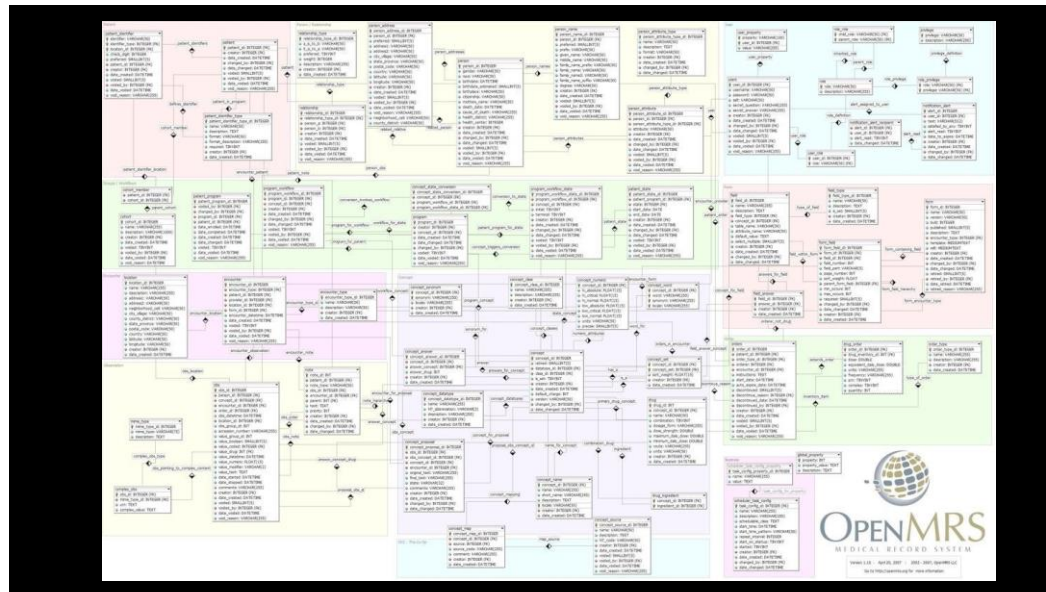
Complex Data Models and Relationships

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

Database Design

- Database design is an **art form** of its own with particular skills and experience
- Goal is to avoid the really bad mistakes and design clean and easily understood databases
- Database design starts with a picture...





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
☑ 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
☑ 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 an object or an attribute of another object?
- Once we define objects, we need to define the relationships between objects.

Len Album
Genre
Artist Rating
Track 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
☑ Tin Man	3:30	America	Greatest Hits	Easy Listen...	★★★★★	23



Representing Relationships in a Database

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

✓ 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
✓ The Man	3:20	America	Greatest Hits	Easy Listen...	★★★★★	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?

Integer Reference Pattern

We use integers to reference rows in another table

id	name
Filter	Filter
1	Led Zeppelin
2	AC/DC

Artist

id	artist_id	title
Filter	Filter	Filter
1	2	Who Made Who
2	1	IV

Album

Key Terminology

Finding our way around....

Three Kinds of Keys

- **Primary key** - generally an integer auto-increment field
- **Logical key** - What the outside world uses for lookup
- **Foreign key** - generally an integer key pointing to a row in another table



Primary Key Rules

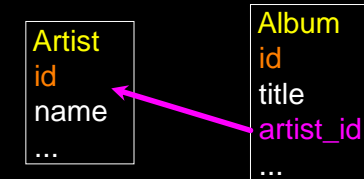
Best practices

- 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 less efficient than integers

User
id
login
password
name
email
created_at
modified_at
login_at

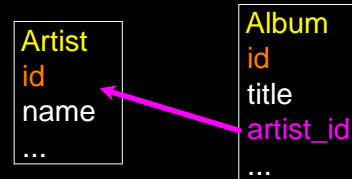
Foreign Keys

- A **foreign key** is when a table has a column that contains a key which points to the **primary key** of another table.
- When all primary keys are integers, then all foreign keys are integers - this is good - very good



Foreign Keys

- Foreign Key references the primary key of another Table! It helps connect your Tables. A foreign key can have a different name from its primary key
- It ensures rows in one table have corresponding rows in another
- Unlike the Primary key, they do not have to be unique. Most often they aren't
- Foreign keys can be null even though primary keys can not



Relationship Building (in tables)

