

Реляционные базы данных и SQLite

Чарльз Северанс



Пайтон для всех
www.py4e.com/lectures3/



DB Browser for SQLite

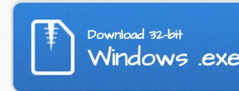
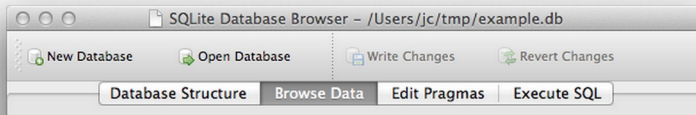
The Official home of the DB Browser for
SQLite



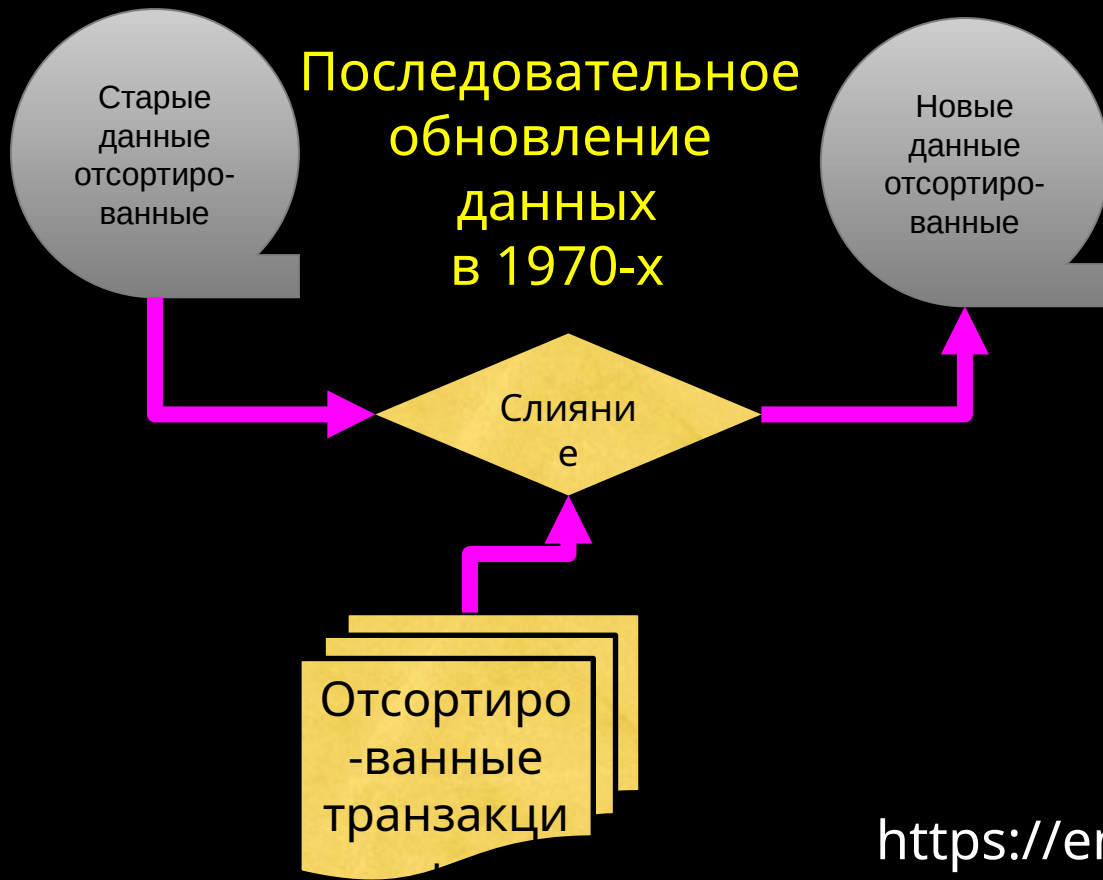
// News

- 2015-07-07 - Added PortableApp version of 3.7.0. Thanks John. :)
- 2015-06-14 - Version 3.7.0 released. :)
- 2015-05-09 - Added PortableApp version of 3.6.0v3.

// Screenshot



<http://sqlitebrowser.org/>



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

Произвольный доступ

- Имея произвольный доступ к данным...
- как сделать отображение данных наиболее эффективным?
- Сортировка может быть не лучшей идеей



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

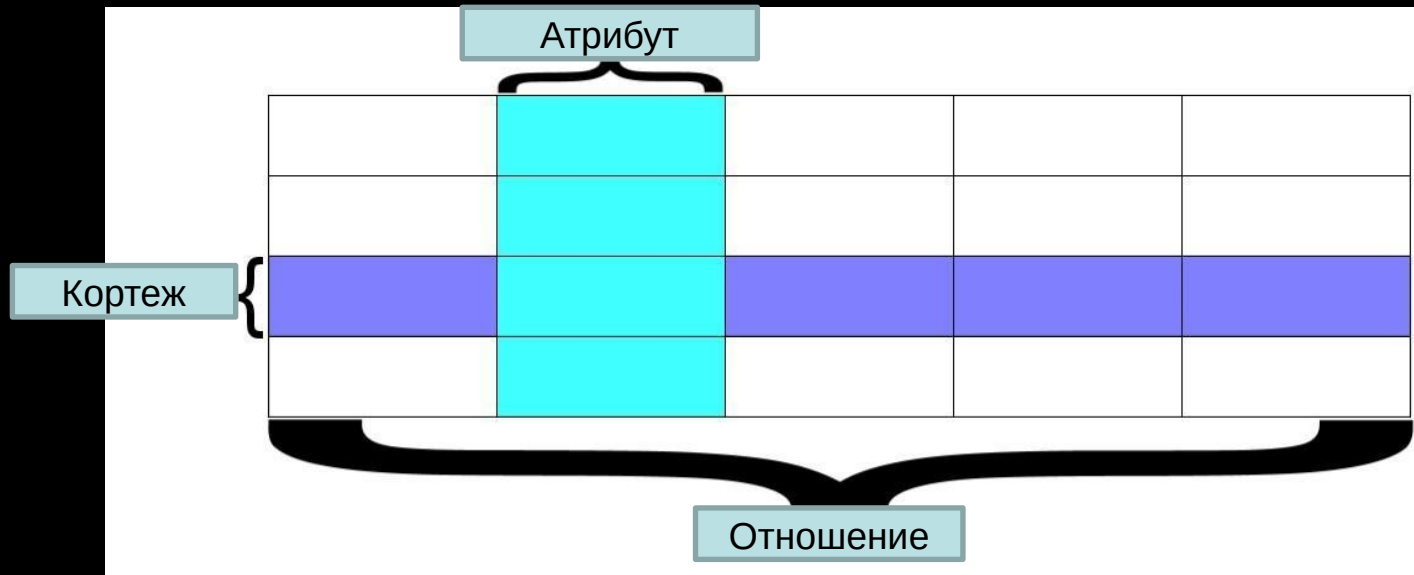
Реляционные базы данных

Реляционные базы данных организуют хранение данных в виде таблиц со строками и столбцами. Мощь реляционной базы данных заключается в способности эффективно извлекать данные из таблиц, особенно в случаях, когда выборка производится из нескольких взаимосвязанных таблиц

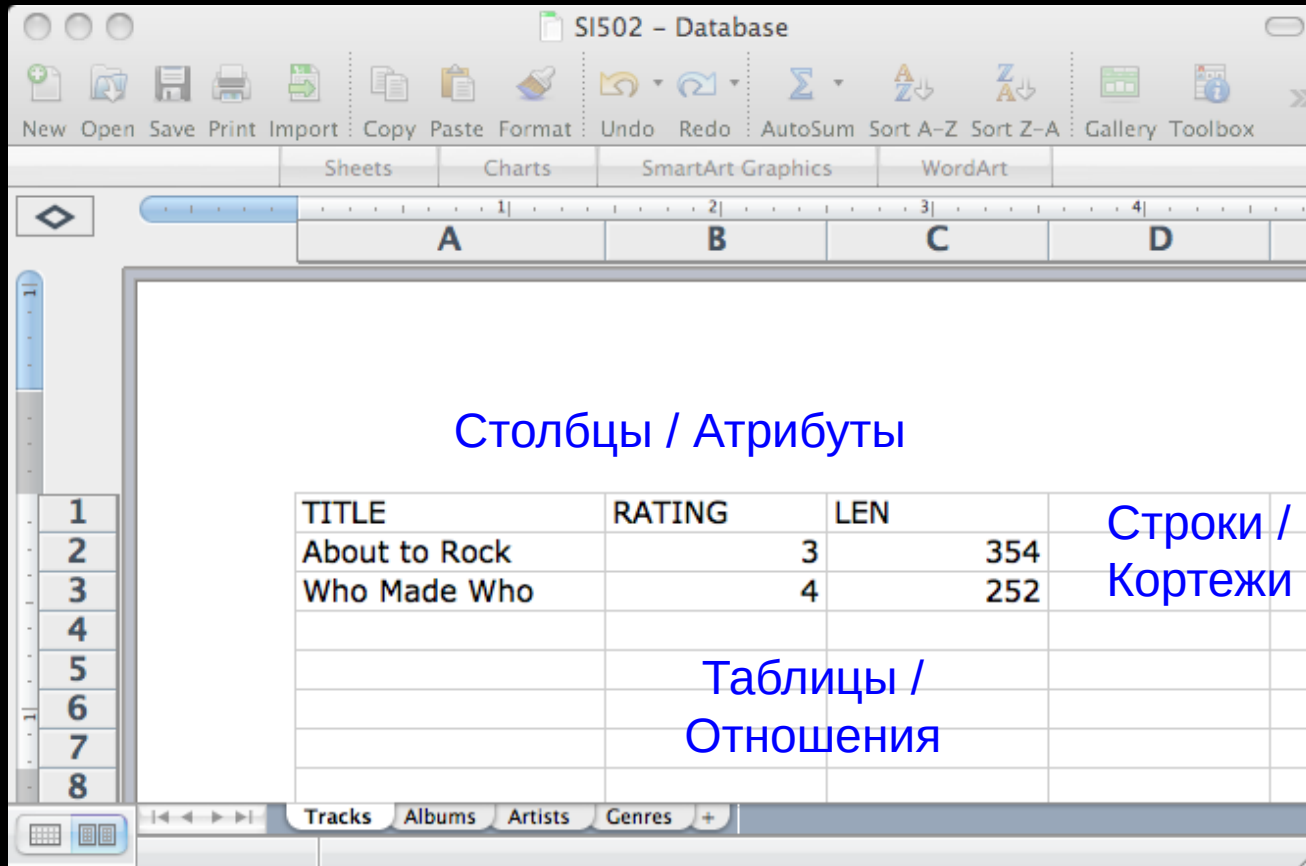
https://ru.wikipedia.org/wiki/Реляционная_база_данных

Терминология

- **База данных** содержит множество таблиц
- **Отношение (или таблица)** содержит кортежи и атрибуты
- **Кортеж (или ряд)** — набор полей, обычно представляющих «объект», например, человека или музыкальную композицию
- **Атрибут (также колонка или поле)** — один из множества возможных элементов данных, относящихся к объекту представленному строкой



Отношение представляет из себя набор кортежей с одинаковыми атрибутами. Кортеж обычно представляет объект и информацию об объекте. Объекты обычно представляют собой физические объекты или концепции. Отношение обычно описывается как таблица, организованная в виде строк и столбцов. Все данные, на которые ссылается атрибут, находятся в одном домене и подчиняются одним и тем же ограничениям. (Википедия)

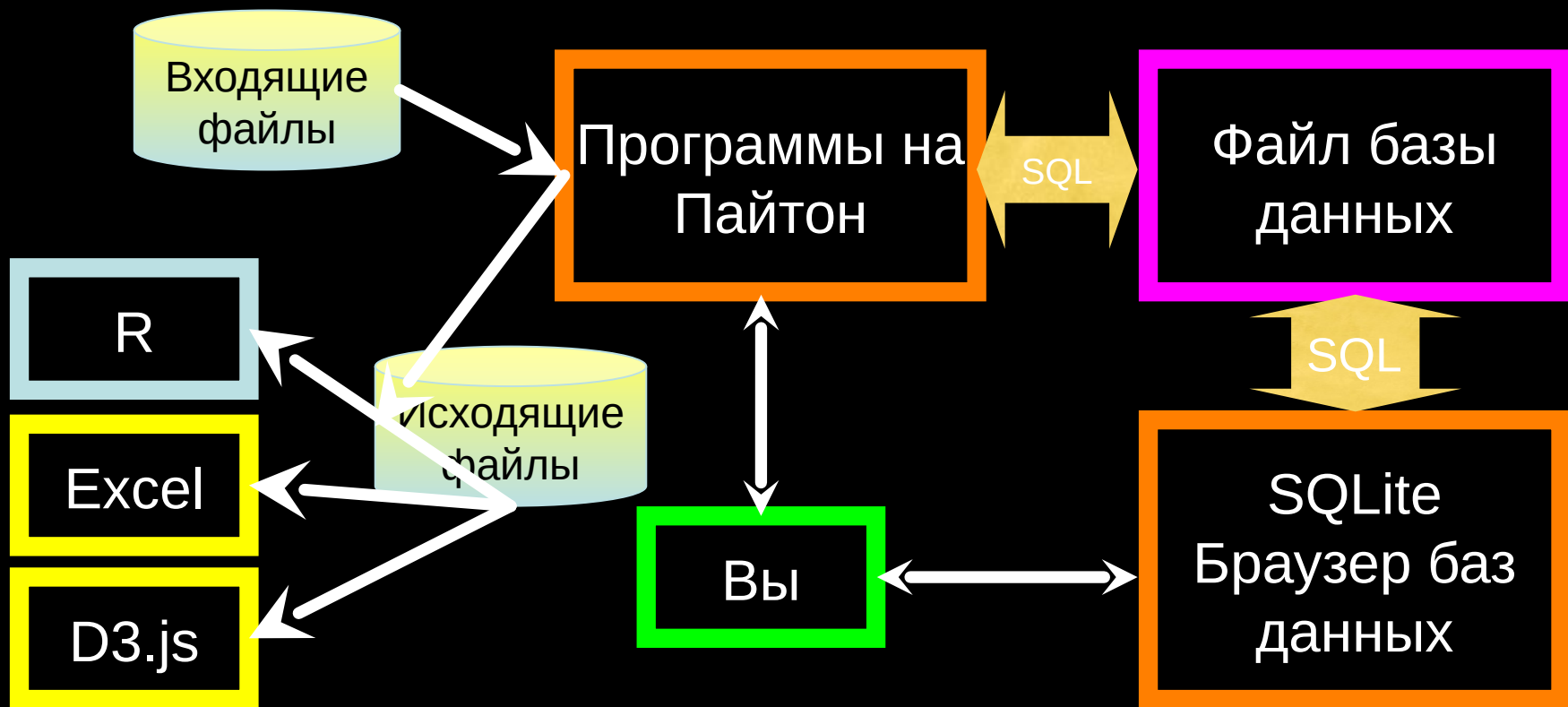


SQL

Язык структурированных запросов — язык команд, при помощи которых мы работаем с базой данных.

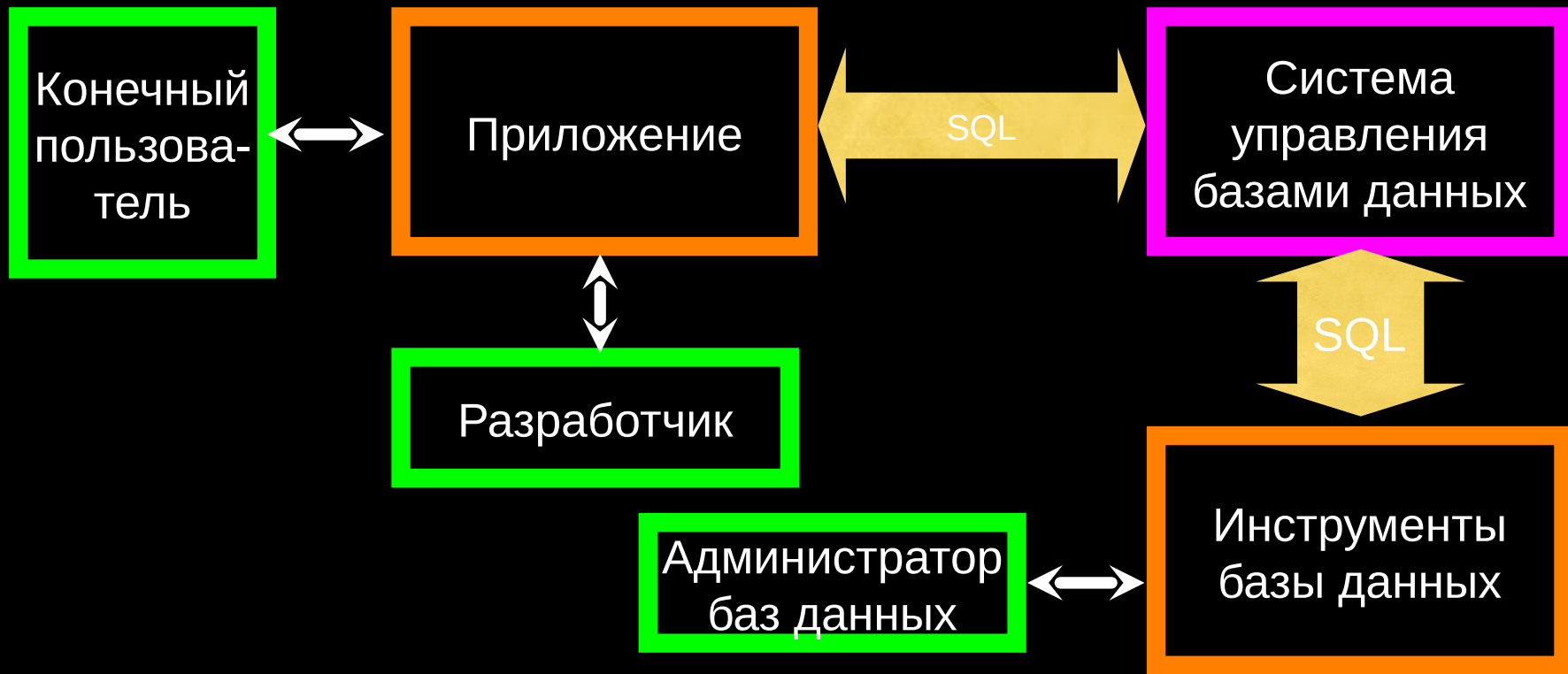
- Создать данные (также Вставить)
- Выбрать данные
- Обновить данные
- Удалить данные

<https://ru.wikipedia.org/wiki/SQL>



Веб-приложения и Базы данных

- **Разработчик приложения** разрабатывает логику и внешний вид приложения, отслеживает работу приложения
- **Администратор баз данных** следит за работой базы данных и настраивает ее в процессе эксплуатации программы
- Часто оба специалиста участвуют в построении «Модели данных»



Администратор баз данных

Администратор баз данных (АБД) — специалист, отвечающий за проектирование, внедрение, обслуживание и ремонт баз данных организации. Эта роль включает разработку и проектирование стратегий размещения баз данных, мониторинг и улучшение производительности баз данных, а также мониторинг свободного места и планирование расширения баз данных в будущем.

Также администратор баз данных может планировать, координировать и реализовывать меры безопасности для защиты баз данных.

https://ru.wikipedia.org/wiki/Администратор_баз_данных

Модель базы данных

Модель базы данных или схема базы данных — структура или формат базы данных, описанный на формальном языке, поддерживаемом системой управления базами данных (СУБД).

Другими словами, «модель базы данных» — это реализация модели данных в системе управления базами данных.

https://ru.wikipedia.org/wiki/Модель_базы_данных

Распространенные СУБД

- 3 самых широко используемых СУБД:
 - **Oracle** — крупная, масштаба предприятия, гибко настраиваемая. Коммерческое ПО с открытым исходным кодом;
 - **MySql** — попроще, но очень быстрая и масштабируемая, с открытым исходным кодом;
 - **SqlServer** — удобный продукт от Майкрософт (также Access)
- Множество небольших, бесплатных СУБД, с открытым исходным кодом:
 - HSQL, **SQLite**, Postgres, ...

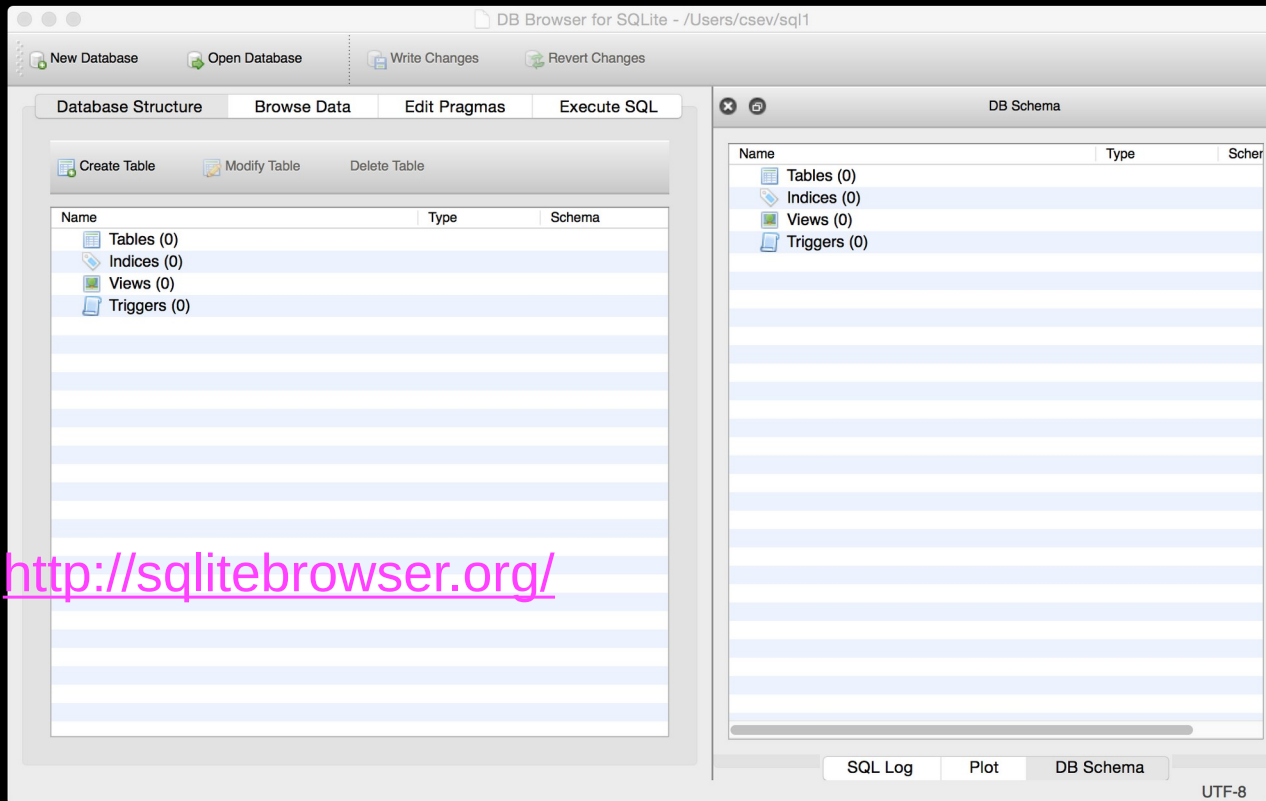
SQLite используется множеством приложений

The logo for Symbian, featuring the word "symbian" in a lowercase, sans-serif font with a blue dot above the 'i'.The Python logo, consisting of two interlocking snakes (one blue, one yellow) to the left of the word "python" in a lowercase, sans-serif font with a trademark symbol.The Skype logo, featuring the word "skype" in a lowercase, rounded, blue font with a trademark symbol.The Microsoft logo, featuring the word "Microsoft" in a bold, black, sans-serif font.The McAfee logo, featuring the word "McAfee" in a bold, red, sans-serif font.The Adobe logo, featuring a stylized red "A" with a white outline, above the word "Adobe" in a black, sans-serif font.The PHP logo, featuring the letters "php" in a lowercase, blue, sans-serif font inside a blue oval.The Google logo, featuring the word "Google" in its multi-colored, sans-serif font.The Toshiba logo, featuring the word "TOSHIBA" in a bold, red, sans-serif font.The Sun Microsystems logo, featuring the word "Sun" in a blue, sans-serif font with "microsystems" in a smaller font below it, and a stylized blue and white logo to the left.

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

Браузер SQLite

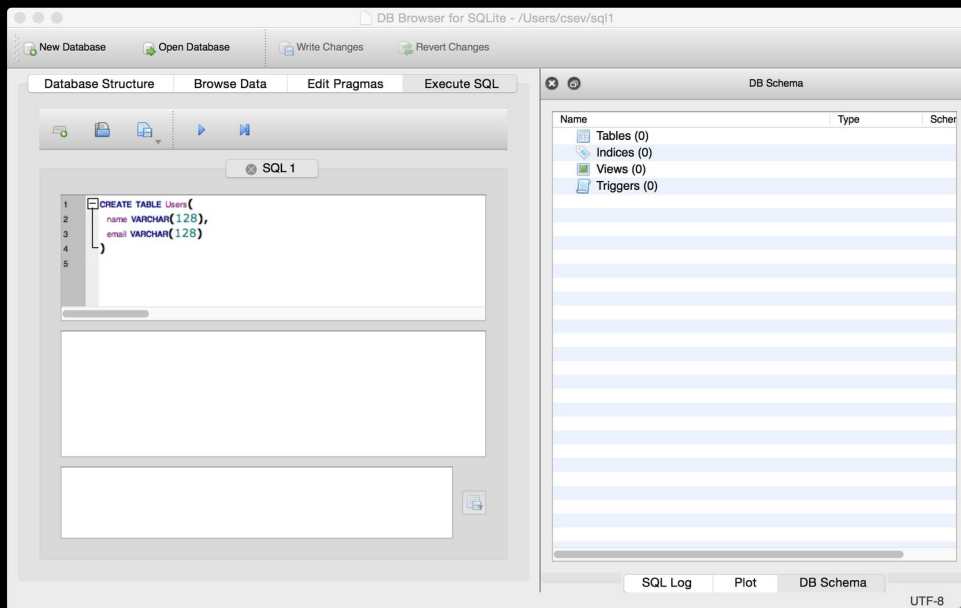
- SQLite — очень популярная база данных. Маленькая, быстрая и бесплатная
- Браузер SQLite позволяет напрямую управлять файлами SQLite
 - <http://sqlitebrowser.org/>
- SQLite встроена в Пайтон и в ряд других языков



Создадим базу данных

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

Начнем с простого: создадим таблицу



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

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

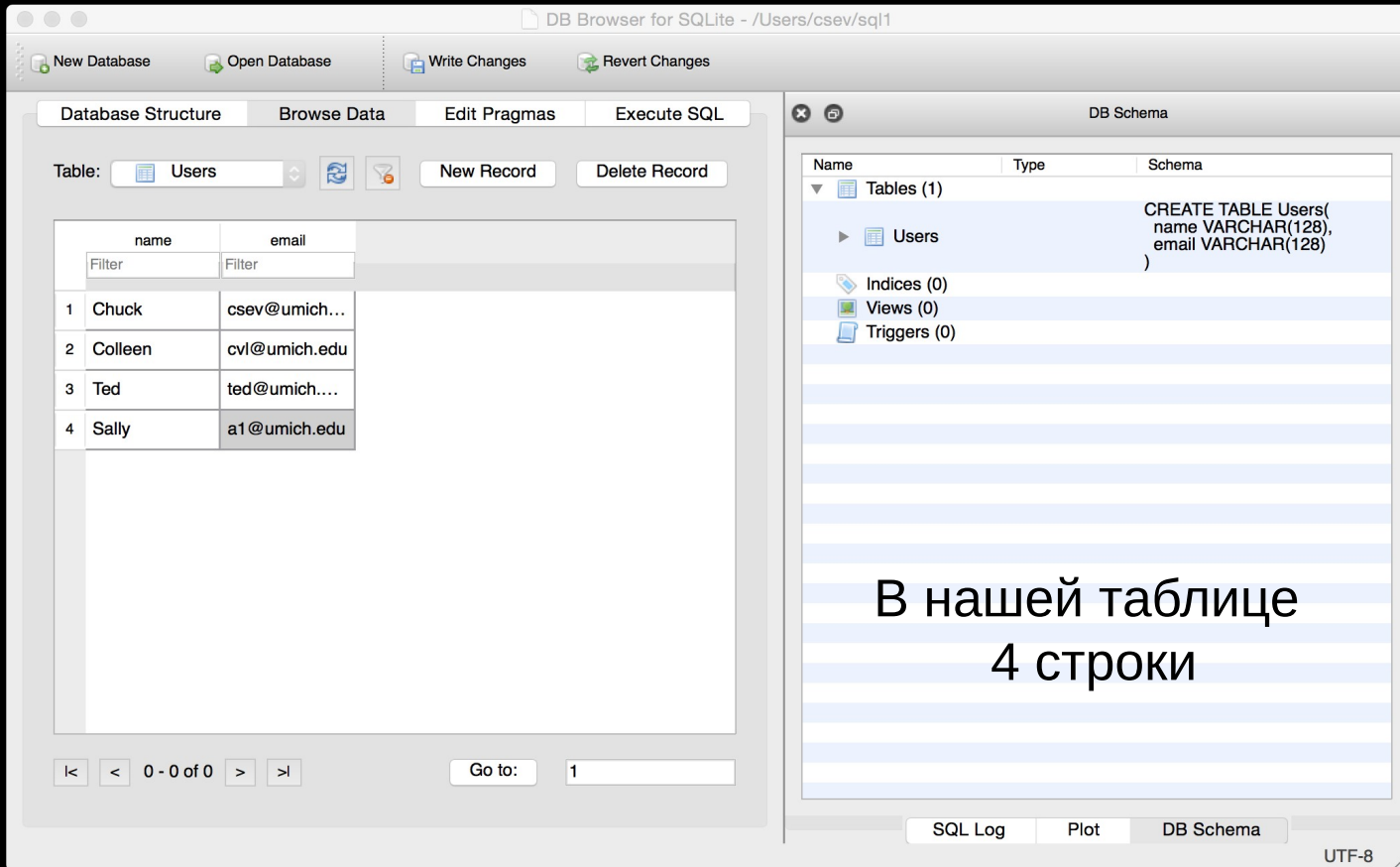
Create Table Modify Table Delete Table

Name	Type	Schema
▼ Tables (1)		
▶ Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

DB Schema

Name	Type	Schema
▼ Tables (1)		
▶ Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema



SQL

Язык структурированных запросов — язык команд, при помощи которых мы работаем с базой данных.

- Создать данные (также Вставить)
- Выбрать данные
- Обновить данные
- Удалить данные

<https://ru.wikipedia.org/wiki/SQL>

SQL: Insert

Оператор Insert вставляет строку в таблицу

```
INSERT INTO Users (name, email) VALUES ('Kristin', 'kf@umich.edu')
```

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 INSERT INTO Users (name, email) VALUES ('Kristin', 'kf@umich.edu')
2
```

Query executed successfully: CREATE TABLE Users(
 name VARCHAR(128),
 email VARCHAR(128)
) (took 0ms)

DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema UTF-8

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

```

SQL 1
1 INSERT INTO Users (name, email) VALUES ('Kristin', 'kf@umich.edu')
2

```

```

Query executed successfully: CREATE TABLE Users(
  name VARCHAR(128),
  email VARCHAR(128)
) (took 0ms)

```

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: Users New Record Delete Record

	name	email
	Filter	Filter
1	Chuck	csev@umich...
2	Colleen	cvi@umich.edu
3	Ted	ted@umich...
4	Sally	a1@umich.edu
5	Kristin	kf@umich.edu



< < 1 - 5 of 5 > >

Go to: 1

DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema

SQL: Delete

Удаляет строку из таблицы на основе критерия поиска

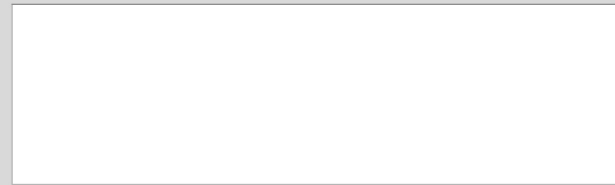
```
DELETE FROM Users WHERE email='ted@umich.edu'
```

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL



```
1 DELETE FROM Users WHERE email='ted@umich.edu'
2
```



Query executed successfully: DELETE FROM Users WHERE email='ted@umich.edu' (took 0ms)

DB Schema

Name	Type	Schema
▼ Tables (1)		
▶ Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1

```
1 DELETE FROM Users WHERE email='ted@umich.edu';
2
```

Query executed successfully: DELETE FROM Users WHERE email='ted@umich.edu' (took 0ms)

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: Users New Record Delete Record

	name	email
Filter	Filter	
1	Chuck	csev@umich...
2	Colleen	cvl@umich.edu
3	Sally	a1@umich.edu
4	Kristin	kf@umich.edu

Go to: 1

SQL Log Plot DB Schema

UTF-8

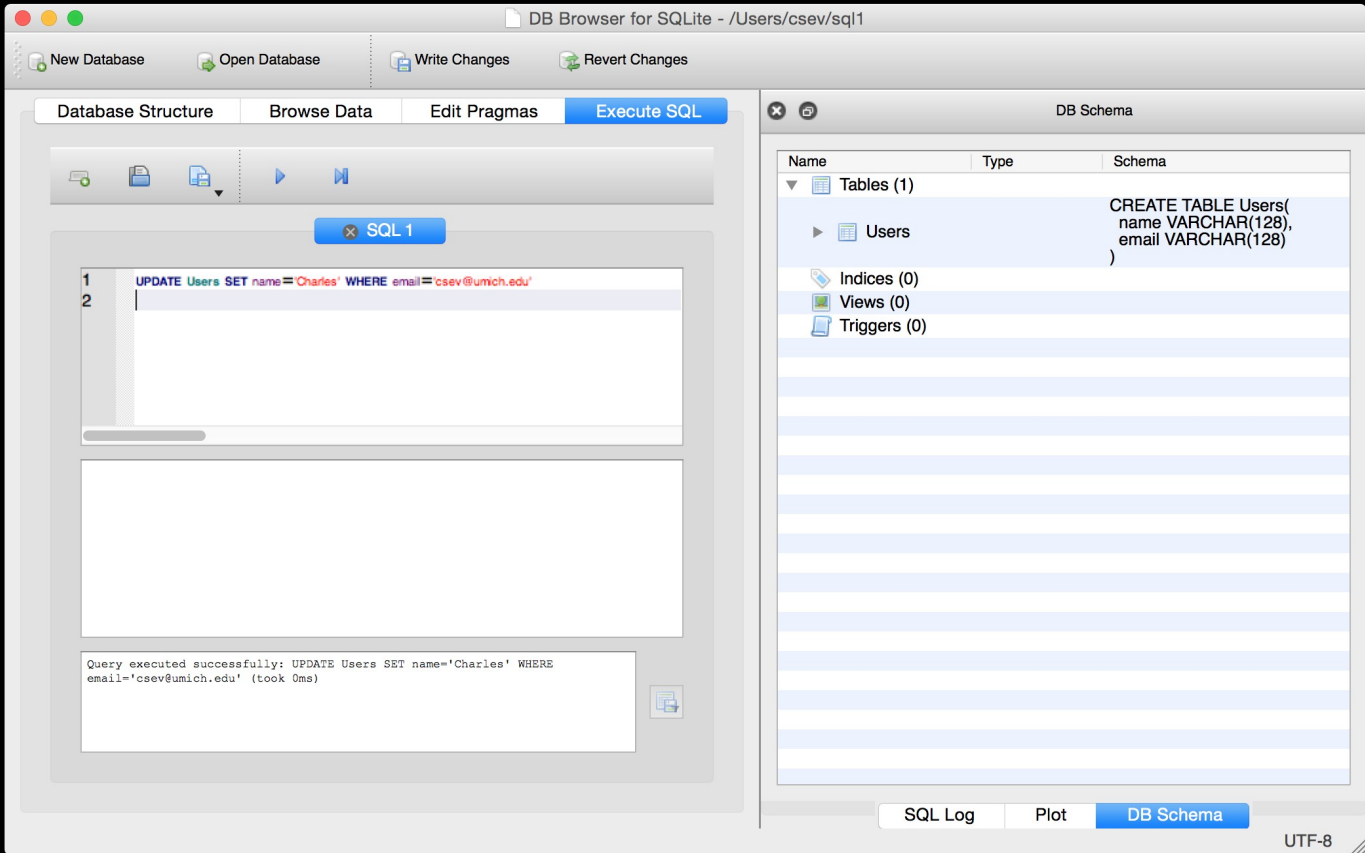
DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL: Update

Позволяет обновить значение столбца таблицы для строк, определенных условием в конструкции «where»

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



DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas **Execute SQL**

SQL 1

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

```
2
```

Query executed successfully: UPDATE Users SET name='Charles' WHERE email='csev@umich.edu' (took 0ms)

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure **Browse Data** Edit Pragmas Execute SQL

Table: Users New Record Delete Record

	name	email
Filter		Filter
1	Charles	csev@umich...
2	Colleen	cvl@umich.edu
3	Sally	a1@umich.edu
4	Kristin	kf@umich.edu

Go to: 1

DB Schema

Tables (1)

- Users

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

Indices (0)
Views (0)
Triggers (0)

SQL Log Plot **DB Schema**

UTF-8

Выборка данных: *Select*

Оператор `select` извлекает группу записей. Вы можете выбрать все записи или подмножество записей, используя условие «WHERE»

```
SELECT * FROM Users
```

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

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL



SQL 1

```
1 SELECT * FROM Users
2
```

	name	email
1	Charles	csev@umich.edu
2	Colleen	cvi@umich.edu
3	Sally	a1@umich.edu
4	Kristin	kf@umich.edu

4 Rows returned from: SELECT * FROM Users (took 0ms)

DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL



SQL 1

```
1 SELECT * FROM Users WHERE email='csev@umich.edu';
2
```

	name	email
1	Charles	csev@umich.edu

1 Rows returned from: SELECT * FROM Users WHERE email='csev@umich.edu' (took 0ms)

DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot DB Schema

Сортировка с помощью ORDER BY

Вы можете добавить конструкцию **ORDER BY** в оператор **SELECT**, чтобы получить результаты отсортированными по возрастанию или убыванию

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

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

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas **Execute SQL**

SQL 1

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

	name	email
1	Sally	a1@umich.edu
2	Charles	csev@umich.edu
3	Colleen	cvl@umich.edu
4	Kristin	kf@umich.edu

4 Rows returned from: SELECT * FROM Users ORDER BY email (took 0ms)

DB Schema

Name	Type	Schema
Tables (1)		
Users		CREATE TABLE Users(name VARCHAR(128), email VARCHAR(128))
Indices (0)		
Views (0)		
Triggers (0)		

SQL Log Plot **DB Schema**

UTF-8

SQL: краткое резюме

```
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
```

Не слишком увлекательно (пока)

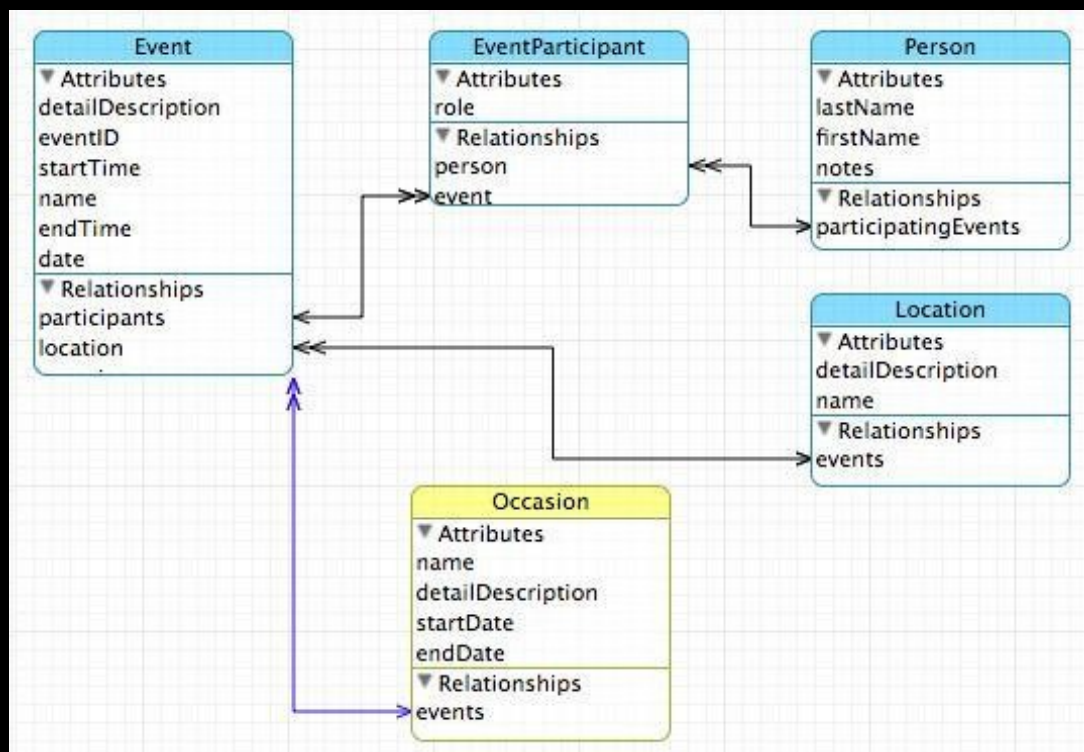
- Таблицы очень похожи на большие, быстрые, программируемые электронные таблицы со строками, столбцами и командами
- Почувствовать мощь таблиц можно, когда таблиц несколько, и мы можем использовать связи и отношения между ними

Сложные модели данных и отношения в них

https://ru.wikipedia.org/wiki/Реляционная_модель_данных

Проектирование баз данных

- Проектирование баз данных — своего рода **искусство**, требующее определенных навыков и опыта
- Наша цель — избегать серьезных ошибок и проектировать ясные и легко понятные базы данных
- Другие люди смогут позже настроить производительность
- Проектирование баз данных начинается со схемы...



Построение модели данных

- Рисуем схему с объектами данных для нашего приложения, а затем выясняем, как представить объекты и связи между ними
- Основное правило: не вводите одинаковые данные дважды, вместо этого используйте связи между таблицами
- Одному объекту «реального мира» соответствует одно описание в базе данных

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

Для каждой единицы информации...

- столбец сам является объектом или атрибутом другого объекта?
- Как только мы определим объекты, необходимо определить и отношения между ними

Длина Альбом

Жанр

Исполнитель

Рейтинг

Композиция

Счет

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tip Man	2:20	America	Greatest Hits	Easy Listen...	★★★★★	22

Композици

я

Альбом

Исполнител

ь

Жанр

Рейтинг

Длина

Счет

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tip Man	2:20	America	Greatest Hits	Easy Listen...	★★★★★	22

Композиция
Альбом
Исполнитель
Жанр
Рейтинг
Продолжительность
Счет

Исполнитель

относится к

Альбом

относится к

Жанр

Композиция

Рейтинг
Длина
Счет

относится к

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tip Man	2:20	America	Greatest Hits	Easy Listen...	★★★★★	22

Исполнитель

ОТНОСИТСЯ
К

Альбом

Композиция
Рейтинг
Длина
Счет

ОТНОСИТСЯ К

Жанр

ОТНОСИТСЯ К

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tip Man	2:20	America	Greatest Hits	Easy Listen...	★★★★★	22

Представление отношений в базе данных

Нормализация базы данных (3НФ)

- Теория баз данных объемна, невозможно понять ее без понимания исчисления предикатов
- **Не дублируйте данные**, ссылайтесь на данные или делайте на них указатели
- Используйте **целые числа для первичных и внешних ключей**
- Добавляйте специальный «**ключевой**» столбец в каждую таблицу, на которую мы будем ссылаться. Обычно программисты именуют его «**id**».

https://ru.wikipedia.org/wiki/Нормальная_форма

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...)	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tie Man	3:30	America	Greatest Hits	Easy Listen...	★★★★★	22

Мы хотим отслеживать, какая группа является создателем каждой музыкальной композиции

К какому альбому относится данная композиция?

К какому альбому относится эта песня?

Использование целочисленных ссылок

Мы используем целые числа для ссылки на строки в другой таблице

id	name
Filter	Filter
1	Led Zepplin
2	AC/DC

Исполнитель

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

Альбом

Три вида ключей

- **Первичный ключ** — как правило, целочисленное поле с авто-инкрементом
- **Логический ключ** — используется для поиска снаружи
- **Внешний ключ** — обычно целочисленный ключ, указывающий на строку в другой таблице



Правила для ключей

Лучшие практики:

- Никогда не используйте **логический ключ** в качестве **первичного ключа**
- **Логические ключи** могут меняться и меняются, хотя и медленно
- **Отношения**, основанные на сопоставлении строковых полей, менее эффективны, чем те, что используют целые числа

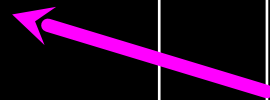
Пользователь
id
логин
пароль
имя
email
создан_дата
изменен_дата
логин_дата

Внешние ключи

- **Внешний ключ** — ситуация, когда в таблице есть столбец, содержащий ключ, который указывает на **первичный ключ** другой таблицы.

Исполнитель
id
ИМЯ
...

Альбом
id
название
artist_id
...



- Хорошо, когда первичные и внешние ключи являются целочисленными.

Установление отношений (в таблицах)

Исполнитель

ОТНОСИТСЯ К

Альбом

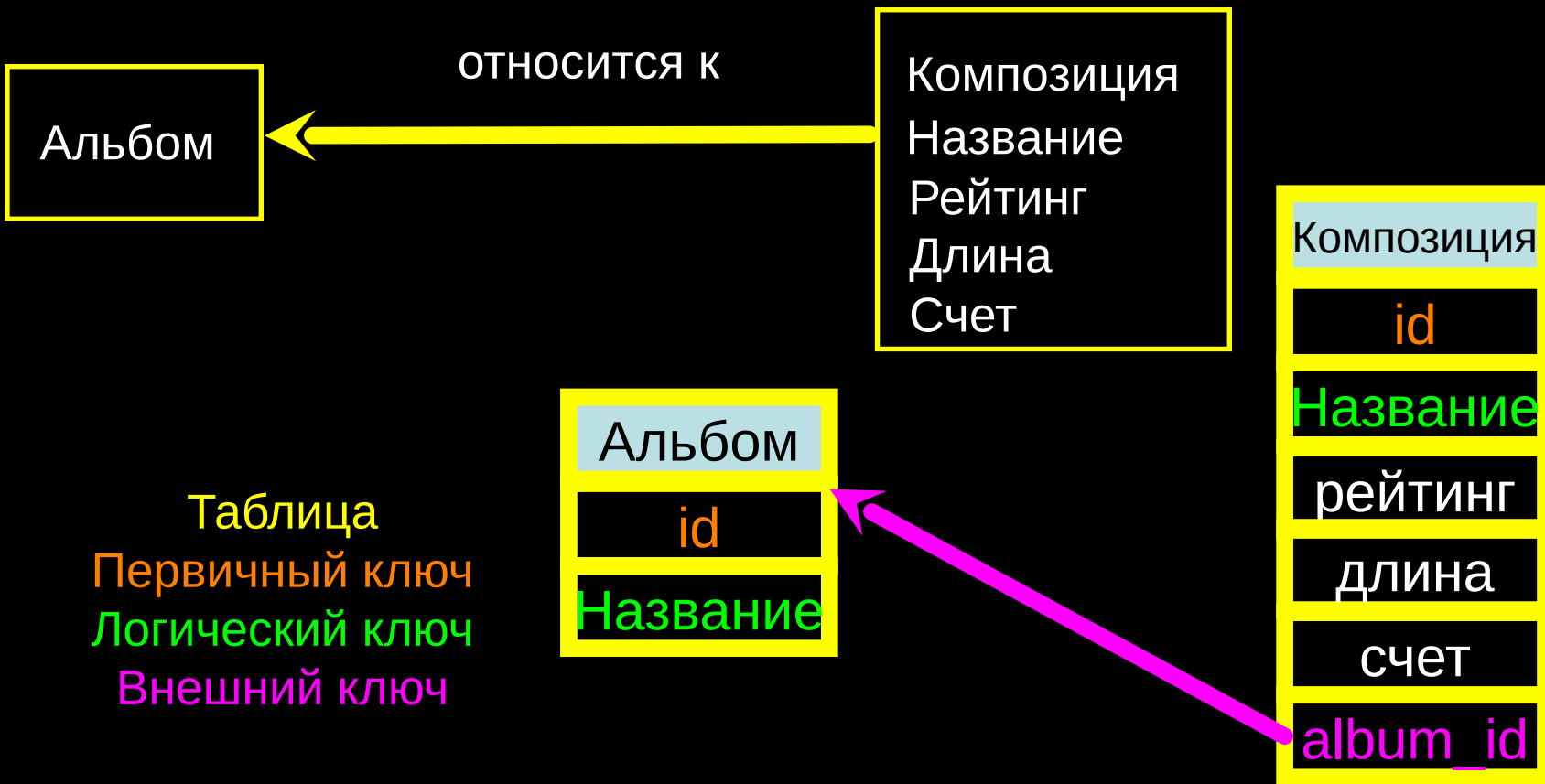
Композиция
Рейтинг
Длина
Счет

ОТНОСИТСЯ К

Жанр

ОТНОСИТСЯ К

<input checked="" type="checkbox"/>	Hells Bells	5:13	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Shake Your Foundations	3:54	AC/DC	Who Made Who	Rock	★★★★★	70
<input checked="" type="checkbox"/>	Chase the Ace	3:01	AC/DC	Who Made Who	Rock		56
<input checked="" type="checkbox"/>	For Those About To Rock (We ...	5:54	AC/DC	Who Made Who	Rock	★★★★★	61
<input checked="" type="checkbox"/>	Dúlamán	3:43	Altan	Natural Wonders M...	New Age		31
<input checked="" type="checkbox"/>	Rode Across the Desert	4:10	America	Greatest Hits	Easy Listen...	★★★★★	23
<input checked="" type="checkbox"/>	Now You Are Gone	3:08	America	Greatest Hits	Easy Listen...	★★★★★	18
<input checked="" type="checkbox"/>	Tip Man	7:30	America	Greatest Hits	Easy Listen...	★★★★★	22



Исполнитель
id
ИМЯ

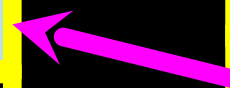
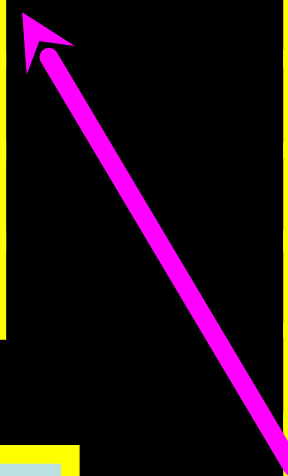
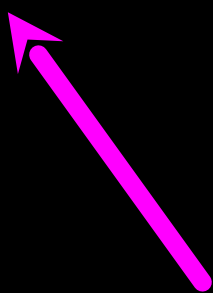
Album
id
название
artist_id

Композиция
id
название
рейтинг
длина
счет
album_id
genre_id

Жанр
id
ИМЯ

Таблица
Первичный ключ
Логический ключ
Внешний ключ

Именование внешнего ключа подобным способом (artist_id) — распространенная практика



Edit table definition

Table

Artist

▼ Advance

Fields

Add field Remove field Move field up Move field down

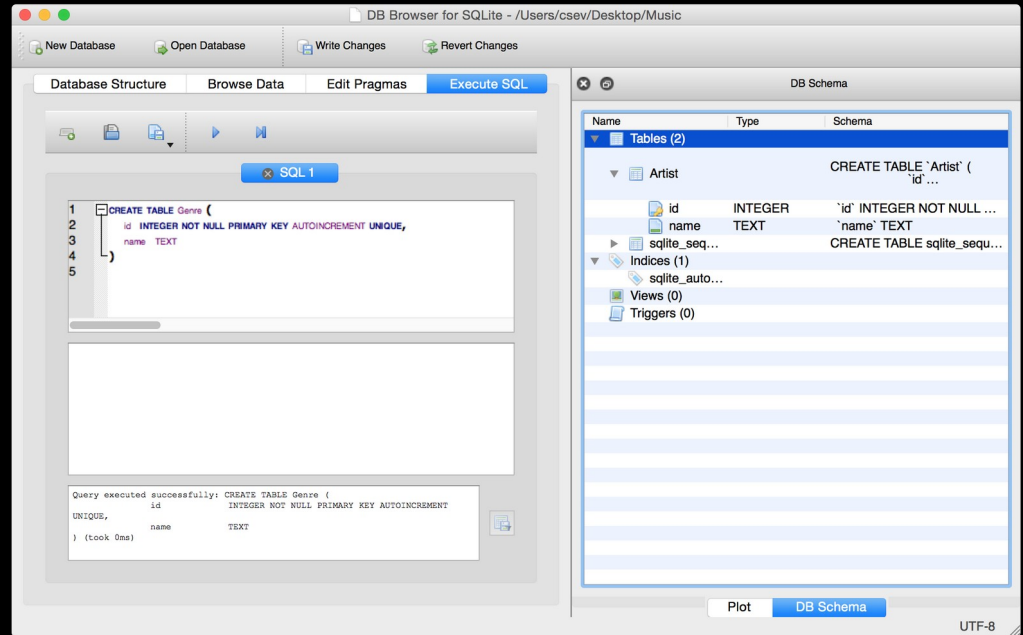
Name	Type	No	PK	AI	U	Default	Check
id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
name	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

```
1 CREATE TABLE Artist (  
2   id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
3   name TEXT  
4 );
```

Cancel OK

Создаем таблицу «Жанр»

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

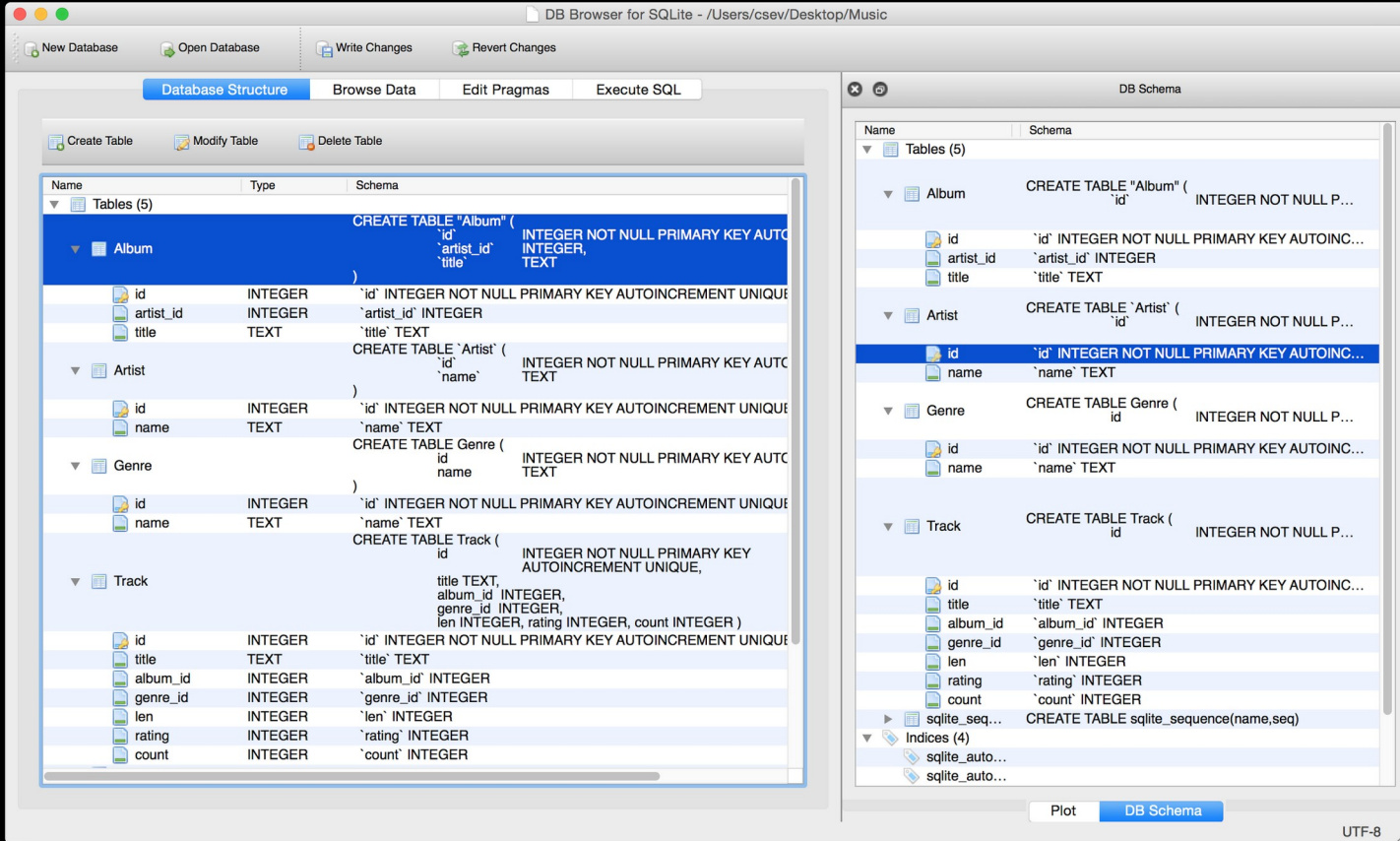


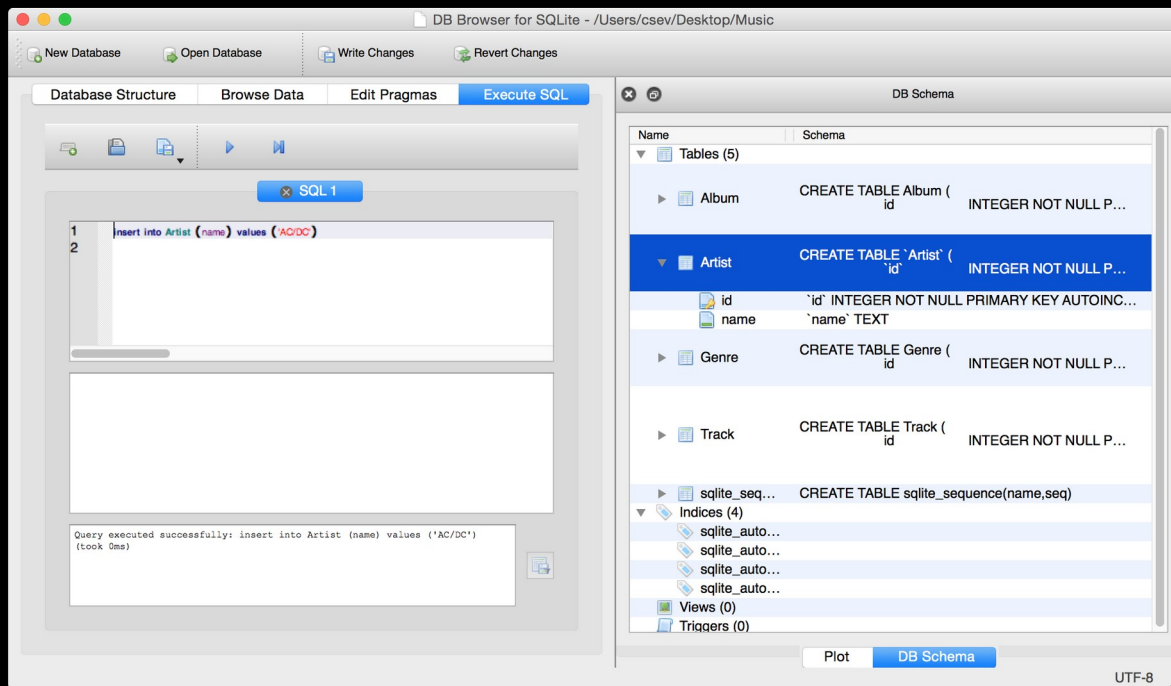
Создаем таблицу «Альбом»

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

Создаем таблицу «Композиция»

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





Вставка в таблицу «Исполнитель» в столбец «Имя» значений: «Led Zepplin» и «AC/DC»

insert into Artist (name) values ('Led Zepplin')
insert into Artist (name) values ('AC/DC')

DB Browser for SQLite - /Users/csev/Desktop/Music

Database Structure | Browse Data | Edit Pragmas | Execute SQL

SQL 1

```
1 insert into Artist (name) values ('AC/DC')
```

Query executed successfully: insert into Artist (name) values ('AC/DC') (took 0ms)

DB Browser for SQLite - /Users/csev/Desktop/Music

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: Artist

id	name
1	Led Zeppelin
2	AC/DC

DB Browser for SQLite - /Users/csev/Desktop/Music

DB Schema

Name | Schema

- Album CREATE TABLE Album (id INTEGER NOT NULL PRI...
- Artist CREATE TABLE 'Artist' (id INTEGER NOT NULL PRI...
- Genre CREATE TABLE Genre (id INTEGER NOT NULL PRI...
- Track CREATE TABLE Track (id INTEGER NOT NULL PRI...
- sqlite_seq... CREATE TABLE sqlite_sequence(name,seq)
- Indices (4) sqlite_auto..., sqlite_auto..., sqlite_auto..., sqlite_auto...
- Views (0)
- Triggers (0)

Plot | DB Schema

UTF-8

Вставка в таблицу
«Исполнитель» в
столбец «ИМЯ»
значений: «Led Zeppelin»
и «AC/DC»

insert into Artist (name) values ('Led Zeppelin')
insert into Artist (name) values ('AC/DC')

The screenshot shows the DB Browser for SQLite interface. The main window displays the 'Genre' table with two records:

id	name
1	Rock
2	Metal

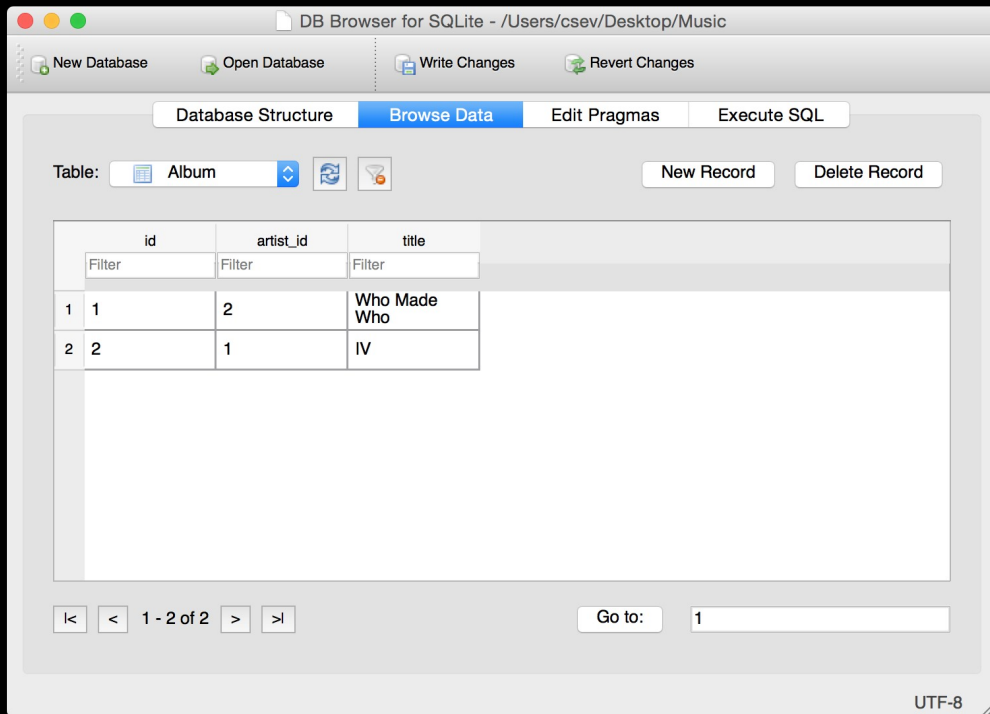
The right-hand pane shows the 'DB Schema' with the following tables and their definitions:

- Album: CREATE TABLE Album (id INTEGER NOT NULL PRI...
- Artist: CREATE TABLE `Artist` (id INTEGER NOT NULL PRI...
- Genre: CREATE TABLE Genre (id INTEGER NOT NULL PRI...
- Track: CREATE TABLE Track (id INTEGER NOT NULL PRI...
- sqlite_seq...: CREATE TABLE sqlite_sequence(name,seq)
- Indices (4): sqlite_auto..., sqlite_auto..., sqlite_auto..., sqlite_auto...
- Views (0)
- Triggers (0)

Navigation controls at the bottom of the table view include: < < 1 - 2 of 2 > > and a 'Go to:' field with the value '1'.

Вставка в таблицу «Жанр» в столбец «имя» значений: «Rock» и «Metal»

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



Вставка в таблицу «Альбом»
в столбцы «title» и «artist_id»
значений: «Who Made Who,
2» и «IV, 1» соответственно

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)

```

Вставка значений в таблицу «Композиция» в столбцы «title», «rating», «len», «count», «album_id», «genre_id»
(Название/Рейтинг/Длина/Счет/ album_id/ genre_id)

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

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

Композиция

Альбом

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

Исполнитель

id	name
Filter	Filter
1	Led Zepplin
2	AC/DC

id	name
Filter	Filter
1	Rock
2	Metal

Жанр

Использование оператора Join между таблицами

[https://ru.wikipedia.org/wiki/Join_\(SQL\)](https://ru.wikipedia.org/wiki/Join_(SQL))

Мощь реляционных баз данных

- Удаляя повторяющиеся данные и заменяя их ссылкой на единственную копию оригинальных данных, мы создаем «**сеть**» информации, которую реляционная база данных может прочитать очень быстро, даже если речь идет об очень большом объеме данных
- Когда нам нужны данные, часто они поступают из нескольких таблиц, связанных между собой **внешними ключами**

Операция JOIN (соединение)

- Являясь частью запроса, операция JOIN **связывает данные из нескольких таблиц**
- Необходимо указать в операторе JOIN **как использовать ключи**, которые соединяют таблицы, используя **предложение ON**

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

Альбом

Исполнитель

	title	name
1	Who Made Who	AC/DC
2	IV	Led Zeppelin

id	name
Filter	Filter
1	Led Zeppelin
2	AC/DC

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

Данные, которые мы
хотим увидеть

Таблицы,
содержащие
эти данные

Как связаны
таблицы

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

id	name
Filter	Filter
1	Led Zeppelin
2	AC/DC

	title	artist_id	id	name
1	Who Made Who	2	2	AC/DC
2	IV	1	1	Led Zeppelin

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

	title	genre_id	id	name
1	Black Dog	1	1	Rock
2	Black Dog	1	2	Metal
3	Stairway	1	1	Rock
4	Stairway	1	2	Metal
5	About to Rock	2	1	Rock
6	About to Rock	2	2	Metal
7	Who Made Who	2	1	Rock
8	Who Made Who	2	2	Metal

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

Соединение двух таблиц
без предложения **ON**
выводит все возможные
комбинации строк

	title	name
1	Black Dog	Rock
2	Stairway	Rock
3	About to Rock	Metal
4	Who Made Who	Metal

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

id	name
Filter	Filter
1	Rock
2	Metal

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

Данные, которые мы
хотим увидеть

Таблицы,
содержащие
эти данные

Как связаны
таблицы

```
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
1	Black Dog	Led Zeppelin	IV	Rock
2	Stairway	Led Zeppelin	IV	Rock
3	About to Rock	AC/DC	Who Made Who	Metal
4	Who Made Who	AC/DC	Who Made Who	Metal

Данные, которые
мы хотим увидеть

Таблицы,
содержащие
эти данные

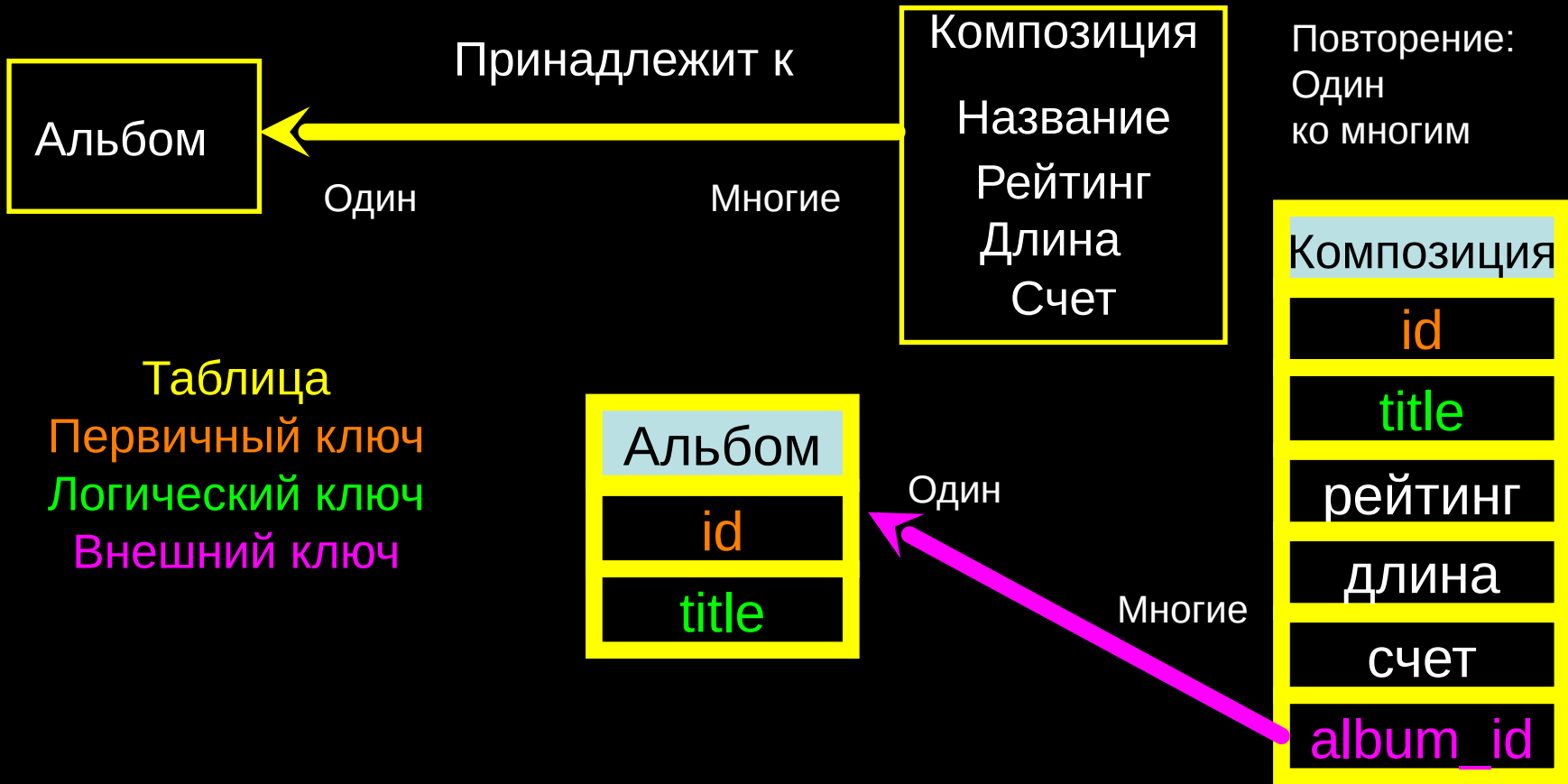
Как связаны
таблицы

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

	title	name	title	name
1	Black Dog	Led Zeppelin	IV	Rock
2	Stairway	Led Zeppelin	IV	Rock
3	About to Rock	AC/DC	Who Made Who	Metal
4	Who Made Who	AC/DC	Who Made Who	Metal

Отношения: многие ко многим

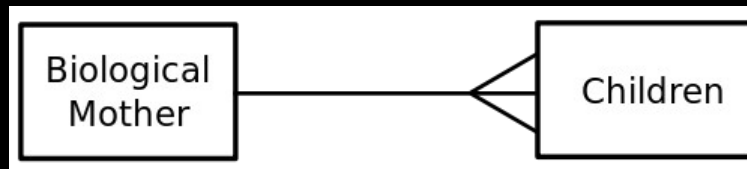
[https://en.wikipedia.org/wiki/Many-to-many_\(data_model\)](https://en.wikipedia.org/wiki/Many-to-many_(data_model))



[https://en.wikipedia.org/wiki/One-to-many_\(data_model\)](https://en.wikipedia.org/wiki/One-to-many_(data_model))

id	name
Filter	Filter
1	Rock
2	Metal

Один



Один

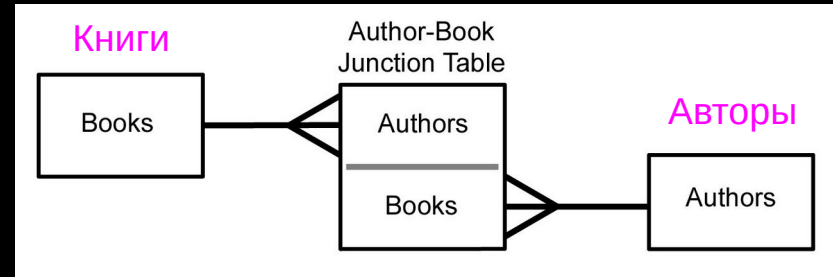
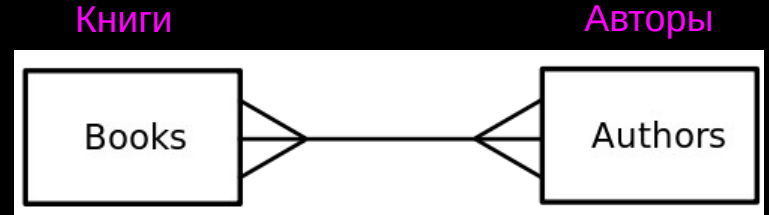
Многие

Многие

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

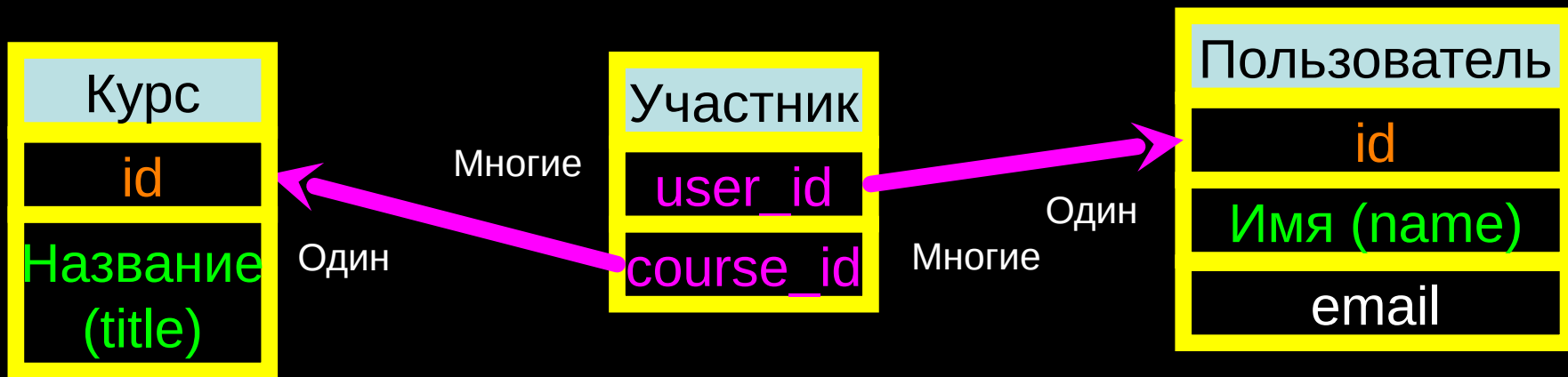
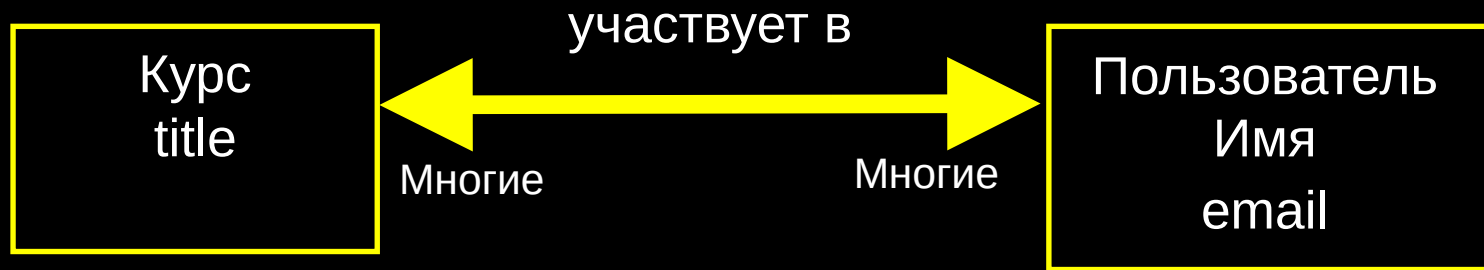
Многие ко многим

- Иногда нам бывает необходимо смоделировать отношения многие ко многим
- Нам нужно добавить «соединяющую» таблицу, содержащую два внешних ключа
- Обычно в таком случае не бывает отдельного первичного ключа



Таблица, связывающая
таблицы «Книги» и «Автор»

[https://en.wikipedia.org/wiki/Many-to-many_\(data_model\)](https://en.wikipedia.org/wiki/Many-to-many_(data_model))



Создаем таблицу «Пользователь»

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

Создаем таблицу «Курс»

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

Создаем таблицу «Участник»

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

Начните с
создания пустой
базы данных

New Database

Open Database

Write Changes

Revert Changes

Database Structure

Browse Data

Edit Pragmas

Execute SQL

Create Table

Modify Table

Delete Table

Name	Type	Schema
▼ Tables (4)		
▶ Course		CREATE TABLE Course (id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, title TEXT)
▶ Member		CREATE TABLE Member (user_id INTEGER, course_id INTEGER, PRIMARY KEY (user_id, course_id))
▶ User		CREATE TABLE User (id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE, name TEXT, email TEXT)
▶ sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
▼ Indices (3)		
sqlite_autoindex_Course_1		
sqlite_autoindex_Member_1		
sqlite_autoindex_User_1		
Views (0)		
Triggers (0)		

Добавление Пользователей и Курсов

```
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');
```

DB Browser for SQLite - /Users/csev/Desktop/si502_database

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma

Table: Course

	id	title
Filter	Filter	
1	1	Python
2	2	SQL
3	3	PHP

< < 1 - 3 of 3 > >

Go to

DB Browser for SQLite - /Users/csev/Desktop/si502_database

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragma Execute SQL

Table: User

	id	name	email
Filter	Filter	Filter	Filter
1	1	Jane	jane@tsugi.org
2	2	Ed	ed@tsugi.org
3	3	Sue	sue@tsugi.org

New Record Delete Record

< < 1 - 3 of 3 > >

Go to: 1

id	name	email
Filter	Filter	Filter
1	Jane	jane@tsugi.org
2	Ed	ed@tsugi.org
3	Sue	sue@tsugi.org

id	title
Filter	Filter
1	Python
2	SQL
3	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);
```

DB Browser for SQLite - /Users/csev/Desktop/si502_database

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Table: Member   New Record Delete Record

	user_id	course_id	role
	Filter	Filter	Filter
1	1	1	1
2	2	1	0
3	3	1	0
4	1	2	0
5	2	2	1
6	2	3	1
7	3	3	0

< < 1 - 7 of 7 > > Go to:

UTF-8

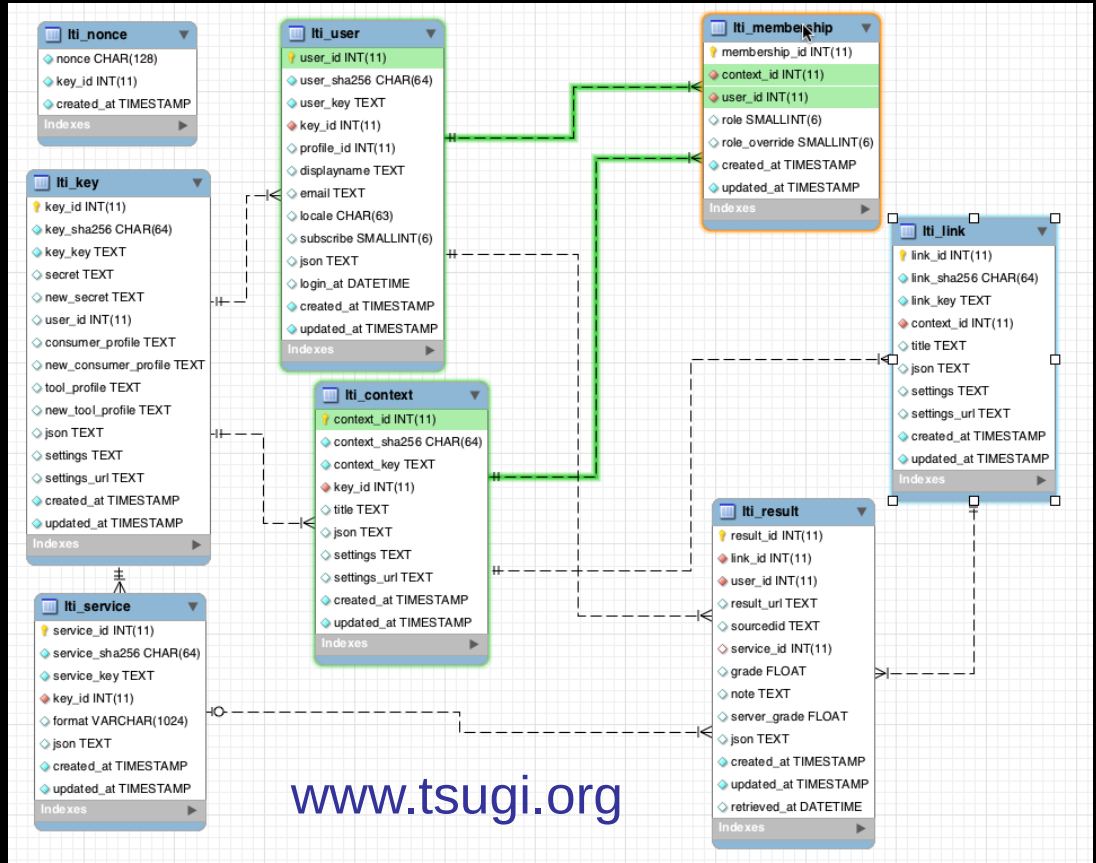
id	name	email
Filter	Filter	Filter
1	Jane	jane@tsugi.org
2	Ed	ed@tsugi.org
3	Sue	sue@tsugi.org

user_id	course_id	role
Filter	Filter	Filter
1	1	1
2	1	0
3	1	0
1	2	0
2	2	1
2	3	1
3	3	0

id	title
Filter	Filter
1	Python
2	SQL
3	PHP

	name	role	title
2	Sue	0	PHP
3	Jane	1	Python
4	Ed	0	Python
5	Sue	0	Python
6	Ed	1	SQL

```
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
```



www.tsugi.org

Сложность обеспечивает скорость

- Сложность делает высокую скорость возможной и позволяет быстро получать результаты по мере роста объема данных
- За счет **нормализации данных и связывания их при помощи целочисленных ключей** общий **объем данных**, который приходится **сканировать** реляционной базе данных, становится гораздо меньше, по сравнению с ситуацией, когда данные хранятся в одной «плоской» таблице
- Своеобразный **компромисс**: потратьте время на проектирование вашей базы данных, чтобы она продолжала работать быстро, когда ваше приложение станет успешным

Дополнительные темы по SQL

- **Индексы** повышают производительность для таких вещей, как строковые поля
- **Ограничения** целостности (не может быть NULL, и т.д.)
- **Транзакции** позволяют группировать SQL-операции и выполнять их как единую логическую единицу

Резюме

- Реляционные базы данных позволяют **масштабировать** базы данных до очень большого объема
- Ключевая идея — хранить только **одну копию каждого элемента данных** и использовать связи и соединения для того, чтобы ссылаться на данные из разных мест
- Это позволяет значительно **сократить объем данных, которые необходимо прочитать** при выполнении сложных операций с большими объемами данных
- Проектирование баз данных и написание SQL-запросов — своего рода **искусство**



Авторы / Благодарности



... Insert new Contributors and Translations here

Авторские права на эти слайды принадлежат Чарльзу Р. Северансу (www.dr-chuck.com), 2010 г., Школе Информации Мичиганского Университета и open.umich.edu, и доступны по лицензии Creative Commons Attribution 4.0 License.

Пожалуйста, сохраняйте этот слайд во всех копиях этого документа, в соответствии с требованиями Лицензии. Если вы внесли изменения, добавьте свое имя или организацию в список участников на этой странице.

Исходная разработка: Чарльз Северанс, Школа Информации Мичиганского Университета.

Перевод выполнила Фомкина Виолетта.

... Добавьте сюда новых авторов и переводчиков