

ОСНОВЫ синтаксиса SQL



Адаптировано из:
Charles Severance
Python for Everybody
<https://www.py4e.com/lessons/database>



Распространенные системы управления базами данных (СУБД)

- Наиболее широко используются:
 - **Oracle** – Большая, платная, промышленного масштаба, множество тонких настроек
 - **MySQL** – Попроще, но очень быстрая и масштабируемая - коммерческий open source
 - **SqlServer** – Очень удобная – разработка Microsoft (также Access)
- Множество других проектов поменьше, бесплатных и open source
 - HSQL, **SQLite**, Postgres, ...

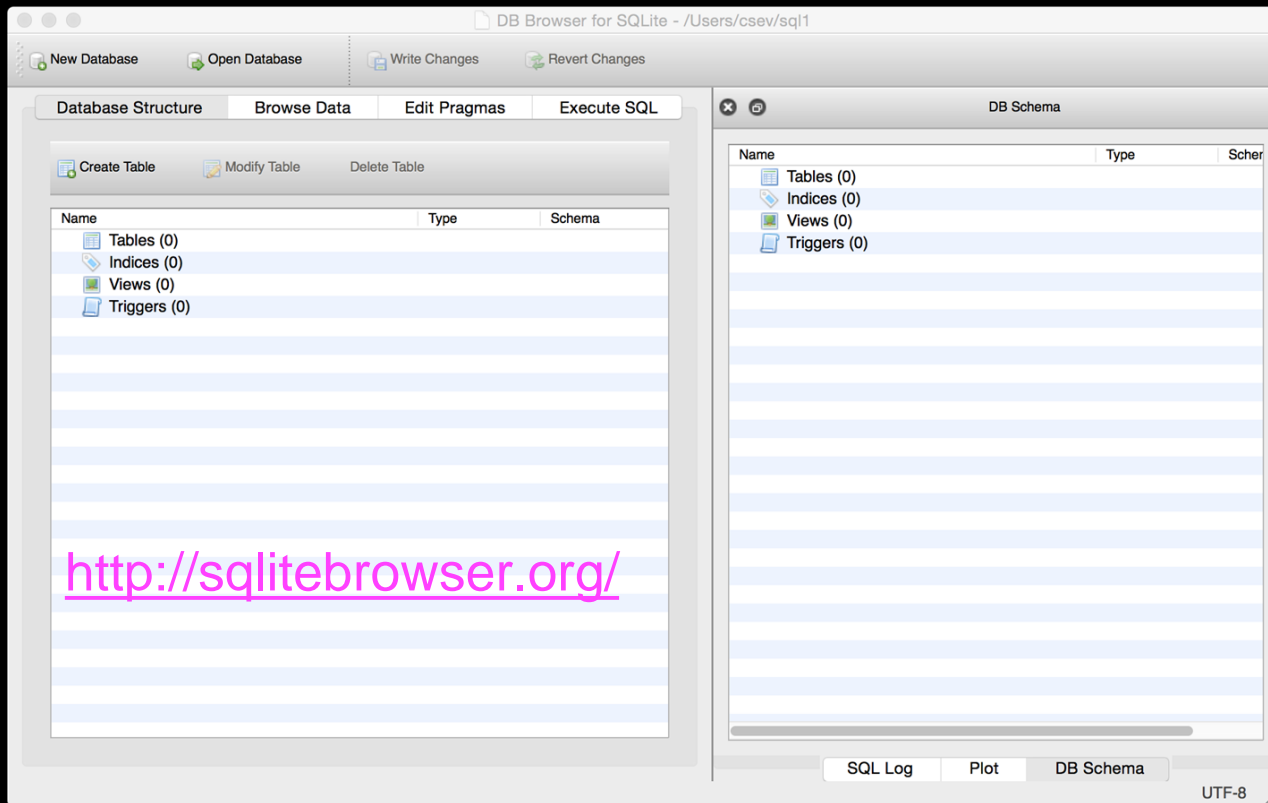
SQLite используется во многих программных продуктах...

The Symbian logo, featuring the word "symbian" in a lowercase, sans-serif font with a blue dot over the 'i'.The Python logo, consisting of two interlocking snakes (one blue, one yellow) followed by the word "python" in a lowercase, sans-serif font with a trademark symbol.The Skype logo, featuring the word "skype" in a lowercase, sans-serif font with a blue speech bubble icon above the 'y'.The Microsoft logo, featuring the word "Microsoft" in a bold, sans-serif font.The McAfee logo, featuring the word "McAfee" in a bold, sans-serif font with a registered trademark symbol.The Adobe logo, featuring a red stylized "A" with a registered trademark symbol, and the word "Adobe" in a sans-serif font below it.The PHP logo, featuring the letters "php" in a lowercase, 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 stylized, blue font with a registered trademark symbol, and the word "microsystems" in a smaller, sans-serif font below it.

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

Графический интерфейс для SQLite

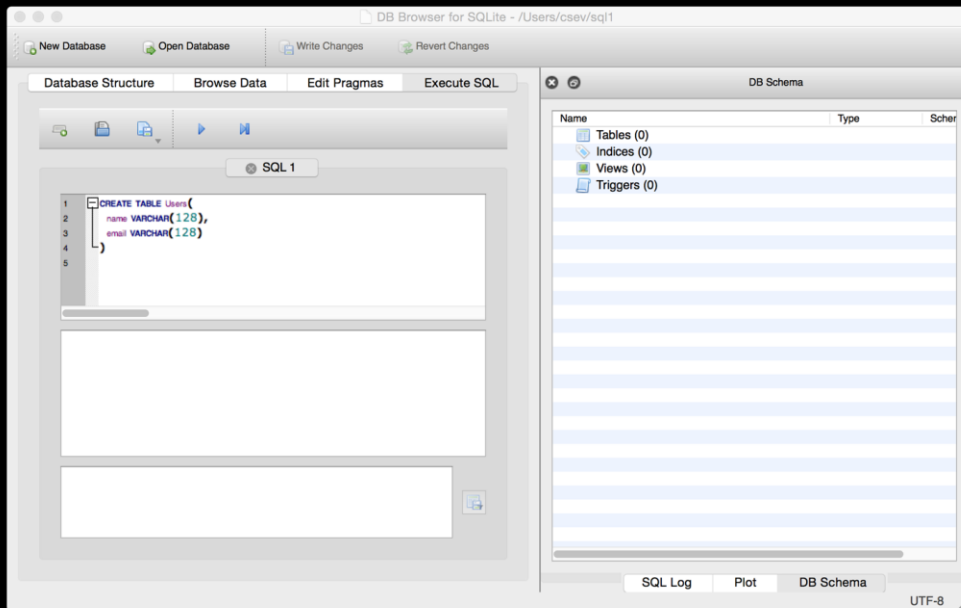
- SQLite – это весьма популярная СУБД. Она бесплатная, отличается быстродействием и компактностью.
- Графический интерфейс для SQLite позволяет напрямую выполнять операции с базой данных.
 - <http://sqlitebrowser.org/>
- SQLite интегрирована в Python и ряд других языков.



Создаем базу данных

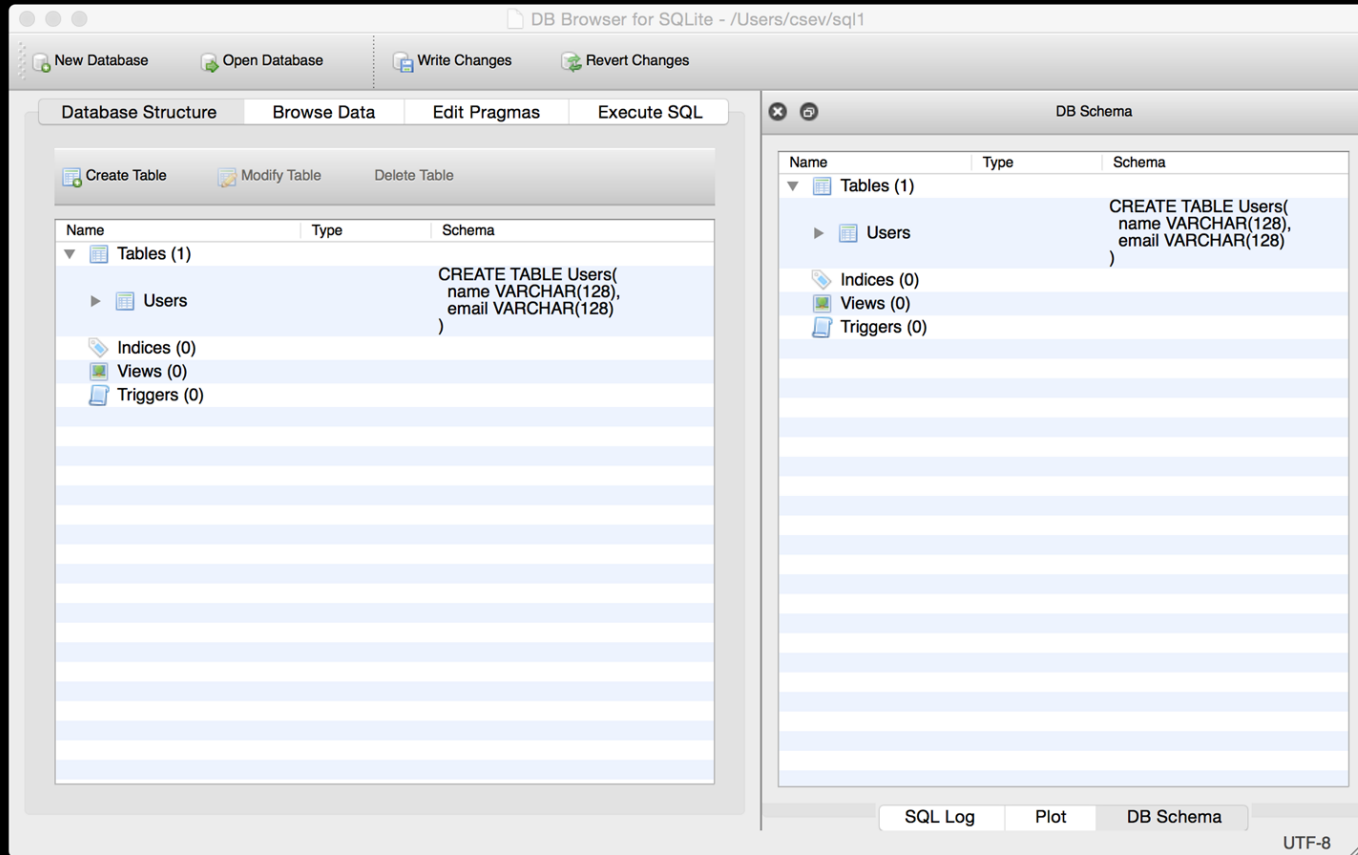
<https://www.py4e.com/lectures3/Pythonlearn-15-Database-Handout.txt>

Начинаем с азов – всего одна таблица



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

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



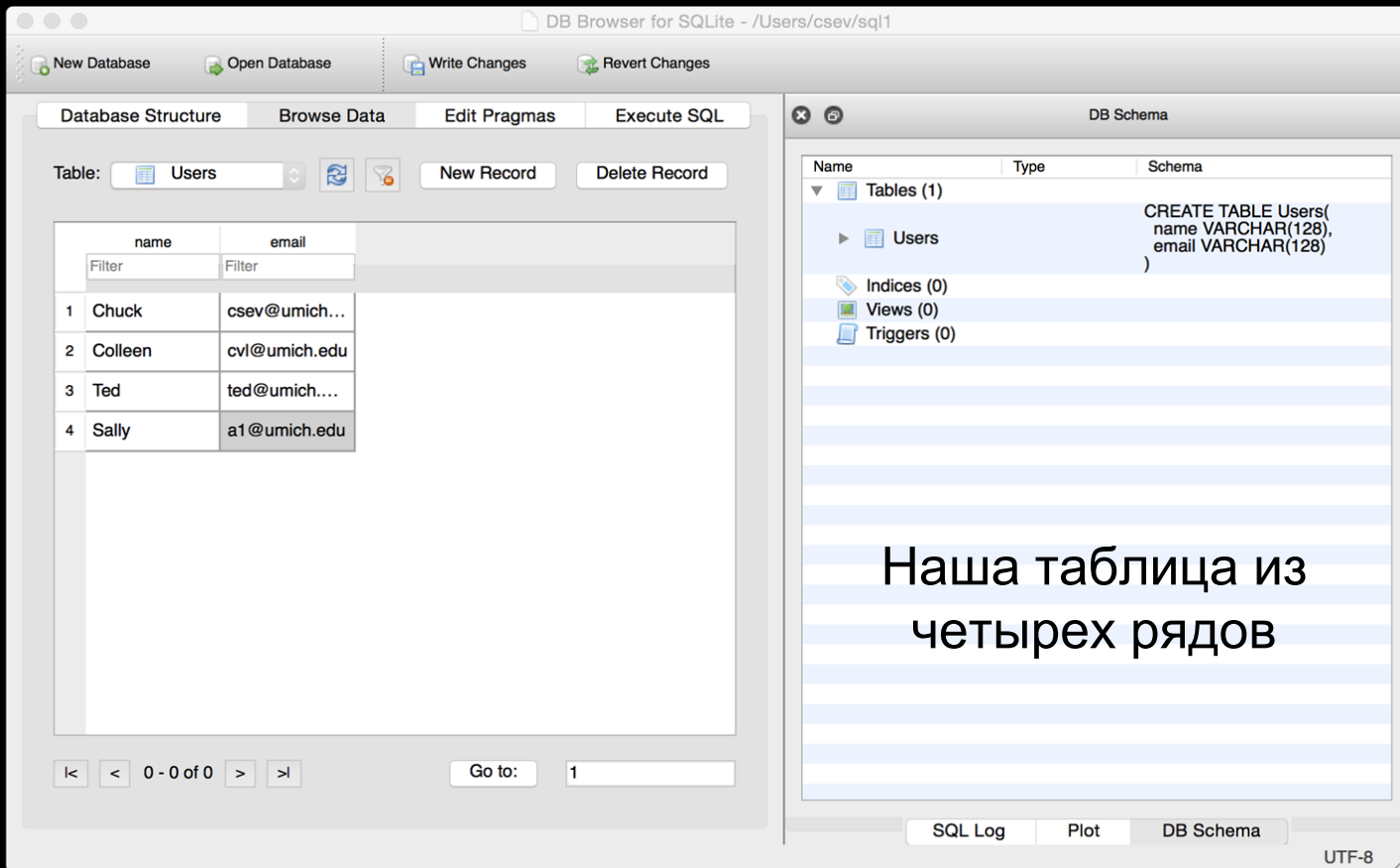


Table: Users

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

< < 0 - 0 of 0 > >

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)

Наша таблица из
четырёх рядов

SQL Log

Plot

DB Schema

UTF-8

SQL

Structured Query Language (язык структурированных запросов) – используется для передачи команд в базу данных

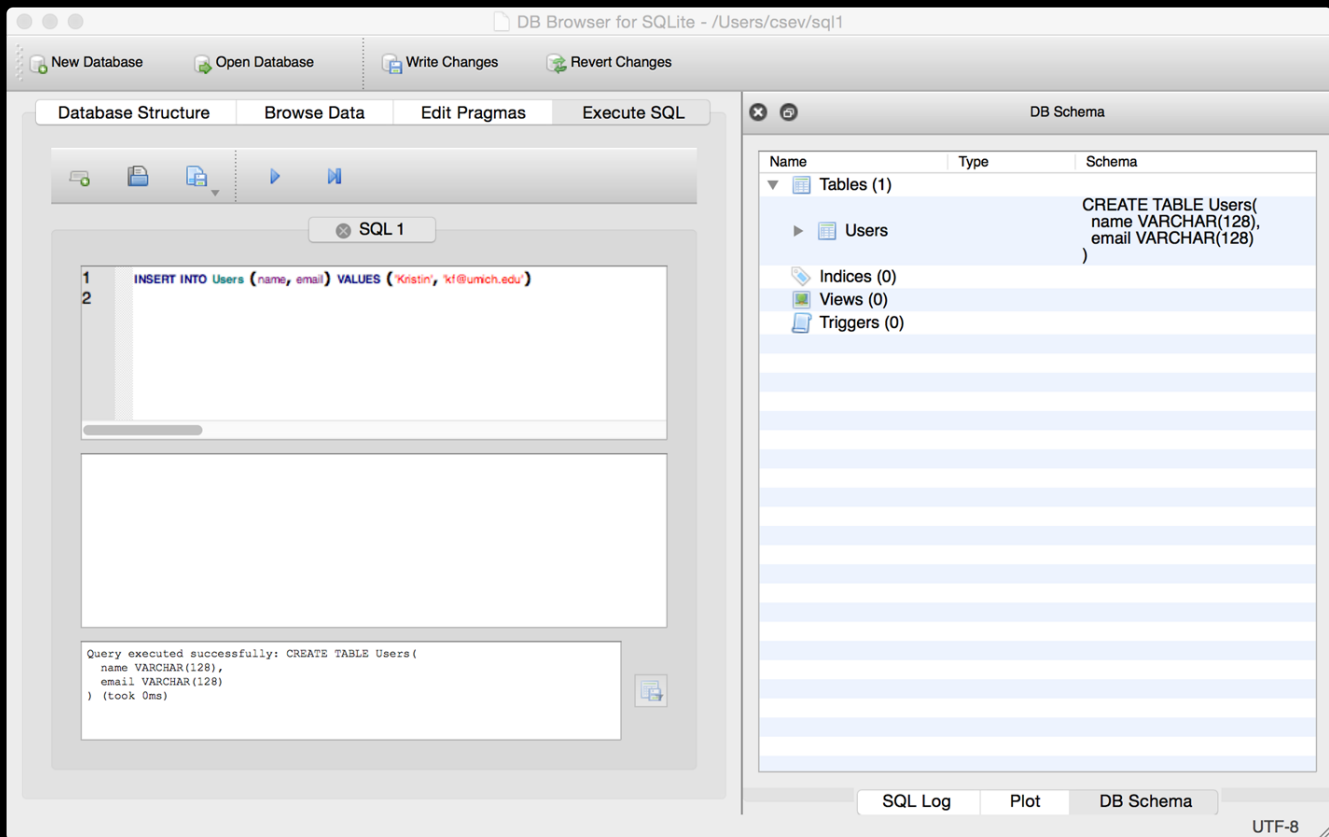
- Создание записей (иначе говоря – вставка, или Insert)
- Получение данных
- Изменение данных
- Удаление данных

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

SQL: Insert

Инструкция INSERT – служит для добавления ряда в таблицу:

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



DB Browser for SQLite - /Users/csev/sql1

New DatabaseOpen DatabaseWrite ChangesRevert Changes

Database StructureBrowse DataEdit PragmasExecute 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 Browser for SQLite - /Users/csev/sql1

New DatabaseOpen DatabaseWrite ChangesRevert Changes

Database StructureBrowse DataEdit PragmasExecute SQL

Table: Users

	name	email
	Filter	Filter
1	Chuck	csev@umich...
2	Colleen	cvl@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 LogPlotDB Schema

UTF-8

SQL: Delete

Удаляет один или несколько рядов из таблицы с учетом критериев выбора:

```
DELETE FROM Users WHERE email='ted@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

1DELETE FROM Users WHERE email=t@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 LogPlotDB Schema

UTF-8

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	cvi@umich.edu
3	Sally	a1@umich.edu
4	Kristin	kf@umich.edu

1 - 4 of 4

Go to: 1

DB Schema

Name Type Schema

Tables (1)

- Users

Indices (0)

Views (0)

Triggers (0)

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

SQL Log Plot DB Schema

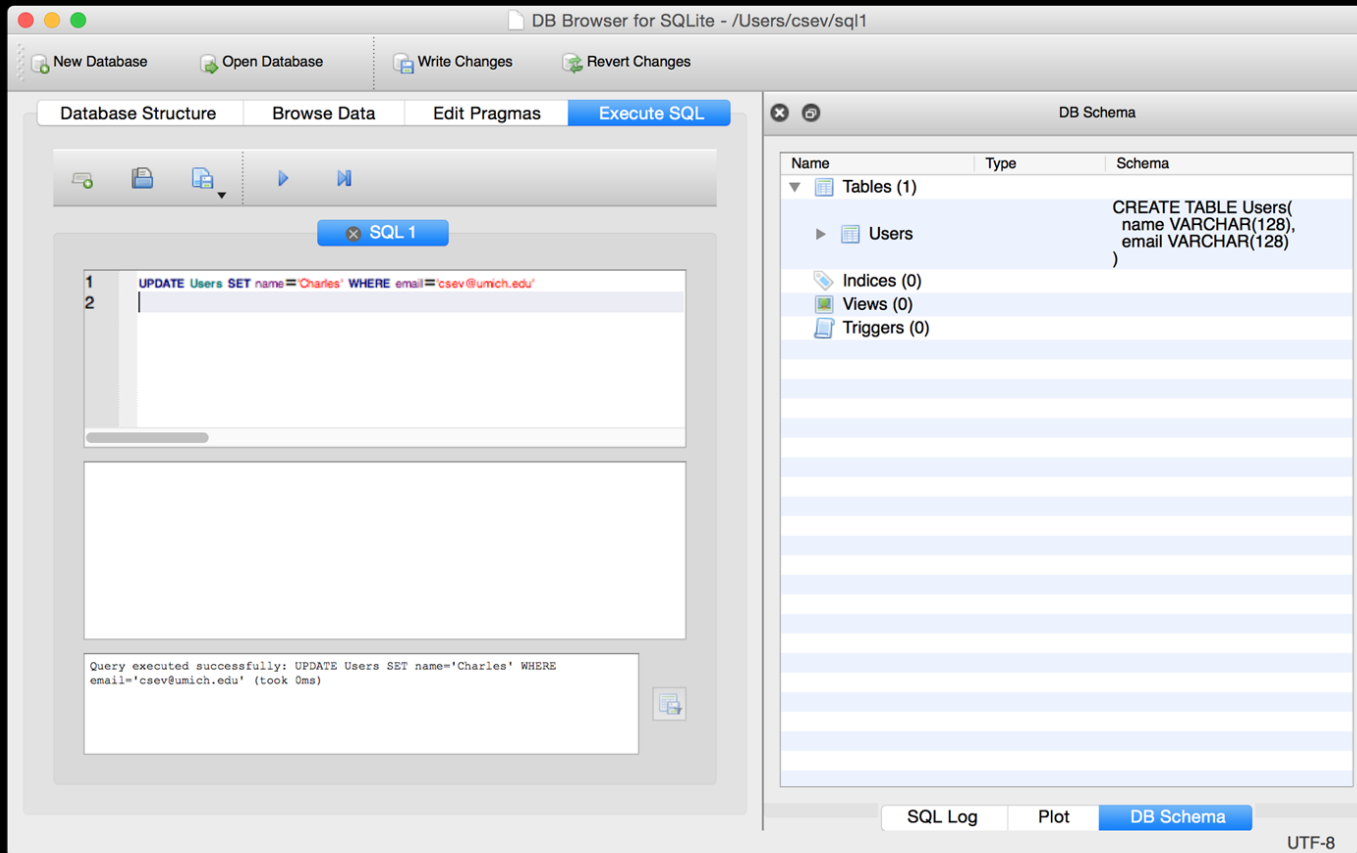
UTF-8

SQL: Update

Позволяет обновить (изменить) значение поля с использованием условия `where`

** без `where` новое значение будет присвоено указанному полю во всех рядах таблицы*

```
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

SQL Log Plot DB Schema

UTF-8

DB Schema

Tables (1)

- Users

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

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

Получение записей: SELECT

Инструкция SELECT используется для получения набора записей из таблицы

— всех :

```
SELECT * FROM Users;
```

— или некоторого подмножества по условию WHERE:

```
SELECT * FROM Users 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 Pragma 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 UTF-8

DB Browser for SQLite - /Users/csev/sql1

New Database Open Database Write Changes Revert Changes

Database Structure Browse Data Edit Pragas Execute SQL

SQL 1

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

	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

UTF-8

Сортировка с помощью ORDER BY

Конструкция **ORDER BY** может включаться в **SELECT**-инструкции для вывода результатов с сортировкой по возрастанию (`ASC`, по умолчанию) или по убыванию (`DESC`)

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

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

DB Browser for SQLite - /Users/csev/sql1

New DatabaseOpen DatabaseWrite ChangesRevert Changes

Database StructureBrowse DataEdit PragmasExecute SQL

SQL 1

1SELECT * 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 LogPlotDB Schema

UTF-8

Базовый синтаксис SQL

```
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;
```



Acknowledgements / Contributions



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

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

Перевод с английского, адаптация:
Максим Миславский out of nowhere