

ПРИМЕНЕНИЕ ПРОГРАММЫ DB BROWSER FOR SQLITE ДЛЯ СОЗДАНИЯ БАЗ ДАННЫХ

USING DB BROWSER FOR SQLITE PROGRAM FOR CREATION OF DATABASES

**A. Suvorov
Z. Zotkin**

Summary. In this article we discuss capabilities of the program DB Browser for SQLite for creation of databases. An example of creation of a database that stores information about registration of students for the courses is given. Examples of generation of SQL queries for this database are discussed.

Keywords: database, SQL query, DB Browser for SQLite program, linking tables, registration of students.

Суворов Александр Павлович

*К.т.н., старший преподаватель, Московский
государственный строительный университет*
suvorovap@mgsu.ru

Зоткин Сергей Петрович

*К.т.н., доцент, Московский государственный
строительный университет*
zotkinsp@mgsu.ru

Аннотация. В данной статье рассматриваются возможности программы DB Browser for SQLite для создания баз данных. Рассмотрен пример создания простой базы данных, хранящей информацию о регистрации студентов на учебные курсы. Даны примеры формирования SQL запросов для этой базы данных.

Ключевые слова: база данных, SQL запрос, программа DB Browser for SQLite, связывание таблиц, регистрация студентов.

Введение

Базы данных могут применяться во многих областях знаний: экономика, электронная коммерция, медицина, статистика, наука, образование и т.д., то есть там, где возникает необходимость организации, структурирования данных и управления ими. Наиболее распространенной моделью реализации базы данных является реляционная модель. Эта модель основана на принципе выявления сущностей и связей между ними. Именно поэтому реляционная база данных состоит из нескольких таблиц, которые связаны между собой.

В данной статье приводится пример создания простой базы данных в программе DB Browser for SQLite. Эта программа является свободно-распространяемым программным продуктом и может быть легко использована в учебных целях, например, на занятиях по курсу «Информационные технологии». Созданная база данных может храниться на домашнем компьютере. Данная программа может служить хорошей альтернативой более известной программе Microsoft Access, так как недостатком последней программы является необходимость ее оплаты. Программа DB Browser for SQLite позволяет создавать таблицы в интерактивном режиме, позволяет связывать эти таблицы. Интерфейс программы переведен на русский язык. При формировании и заполнении таблиц знание языка SQL не требуется. Однако при фор-

мировании запросов пользователь должен знать язык SQL, но при наличии множества подсказок в этой программе запись запросов на языке SQL вряд ли будет являться большим затруднением для пользователя.

Рассмотрен пример формирования базы данных, в которой содержится информация о регистрации студентов университета на различные курсы. Эта регистрация предполагает, что студент не входит в какую-либо группу студентов, а является самостоятельным слушателем учебных курсов, как это обычно и происходит в западных университетах.

В первом разделе статьи мы рассмотрим создание таблиц, входящих в данную базу данных, а во втором разделе мы разберем примеры запросов к созданной базе данных.

Создание и заполнение таблиц

Рассмотрим пример базы данных, в которой будет храниться информация о регистрации (записи) студентов на те или иные курсы. Эта база данных будет состоять из трех таблиц. В первой таблице students будет храниться информация о студентах института. Во второй таблице courses мы запишем информацию о курсах, которые предлагаются для обучения. Наконец в третьей таблице

registration будет располагаться (наиболее важная) информация о регистрации или записи студентов на курсы.

Создание таблицы в программе DBBrowser for SQLite происходит по команде Создать таблицу, которая доступна на закладке Структура БД. При запуске команды Создать таблицу открывается диалоговое окно Редактирование определения таблицы. В этом диалоговом окне прежде всего мы задаем имя таблицы. Новое поле добавляется в структуру таблицы при нажатии кнопки Add (добавить) на закладке Поля. Для созданных полей таблицы мы задаем имя поля и тип данных. Чаще всего используется текстовое поле (TEXT) или поле целого типа (INTEGER). Также при создании полей таблицы можно указать характеристики поля: НП (поле непустое), ПК (первичный ключ), АИ (автоинкремент), У (уникальное поле). Характеристика НП означает, что это поле должно быть обязательно заполнено и не может быть пустым. Характеристика ПК означает, что это поле является первичным ключом в данной таблице. При активации характеристики У значение в этом поле не может повторяться для разных строк (записей) будущей таблицы.

Таблица students будет иметь следующие поля: SID (идентификатор студента, student ID), Lastname (фамилия студента), Firstname (имя студента), Dateofbirth (дата рождения студента). Возможно также добавление и других полей. Для каждого из этих полей назначается тип данных TEXT (текстовое поле). Поле SID будет являться первичным ключом таблицы, т.е. для этого поля надо установить флажок ПК.

Заполнение таблицы происходит на вкладке Данные. После того как мы зайдем на эту вкладку, мы увидим названия созданных нами полей — SID, Lastname, Firstname, Dateofbirth, но сама таблица пока остается незаполненной. Для того чтобы создать конкретную запись, мы должны справа найти небольшой значок », нажав на который мы увидим выпадающее меню с командой Добавить запись. Если мы выберем еще раз опцию Добавить запись, то новая и пока пустая запись будет пристроена к таблице снизу. Щелкая на ячейки (поля) этой новой записи и двигаясь по горизонтали, можно указать конкретные значения для каждого поля — SID, Lastname, Firstname, Dateofbirth и таким образом заполнить новую запись. Но вместо Добавить запись мы можем выбрать Вставить значения, и тогда мы увидим не всю таблицу, а лишь последнюю запись в виде формы с полями Имя — Тип — Значение. Теперь для данной записи мы можем указать значение для каждого из полей (SID, Lastname, Firstname, Dateofbirth), используя для этого столбец Значение данной формы. После заполнения формы следует ее сохранить. После заполнения таблицы следует нажать на кнопку Записать изменения, чтобы наши данные были сохранены на диске.

Приведем пример таблицы students.

SID	Lastname	Firstname	Dateofbirth
S1	Витько	Никита	23.02.2002
S2	Голованов	Андрей	14.03.2002
S3	Фетисова	Мария	11.11.2002
S4	Шабров	Тихон	05.08.2003

Таблица courses будет иметь следующие поля: CID (идентификатор курса), Subject (название изучаемого предмета), Description (описание курса), Day (день недели, в который читается курс). Как и прежде, мы создаем структуру таблицы, пользуясь командой Создать таблицу на закладке Структура БД. Тип данных для каждого из этих полей является текстовым (TEXT). Для поля CID устанавливаем флажок ПК, так как это поле будет являться первичным ключом таблицы.

Теперь, пользуясь вкладкой Данные, мы заполняем таблицу конкретной информацией. Как и прежде, находим значок » и далее из выпадающего меню выбираем команду Добавить запись → Добавить запись или Добавить запись → Вставить значения. После добавления записей следует использовать команду Записать изменения. Приводим пример таблицы courses

CID	Subject	Description	Day
MAT-1	Математика	векторная алгебра	понедельник
MAT-2	Математика	векторная алгебра	вторник
ФИЗ	Физика	механика	среда

Как мы видим из этой таблицы, курсы MAT-1 и MAT-2 являются одинаковыми по содержанию, но читаются в разные дни и возможно разными преподавателями.

Наконец переходим к главной таблице registration, в которой мы запишем данные о регистрации студентов на разные курсы. Эта таблица будет иметь следующие поля: RID (номер записи, числовой тип данных INTEGER), SID (идентификатор студента, текстовый тип данных TEXT), CID (идентификатор курса, текстовый тип данных TEXT), Grade (оценка, текстовый тип данных TEXT). Поле RID будет являться первичным ключом для этой таблицы. Когда мы создаем структуру таблицы registration, мы должны указать, как эта таблица связана с ранее созданными таблицами students и courses. Мы уже знаем, что после добавления того или иного поля таблицы, т.е. после нажатия на кнопку Add в диалоговом окне Редактирование определения таблицы, у нас есть возможность задать имя поля, тип данных и различные характеристики для этого поля, например, НП, ПК и т.д. Если мы сдвинем вправо горизонтальный ползунок, то увидим, что существует возможность задавать и другие свойства для созданных полей. Найдем свойство Внешний ключ. Именно в этом столбце мы задаем поле, которое будет

являться внешним ключом для таблицы registration. Щелкнем в этом столбце на соответствующее поле, например, поле SID. Далее в этом столбце мы можем выбрать таблицу и поле, которое будет являться первичным ключом по отношению к полю SID таблицы registration. Выберем таблицу students и поле SID. Таким образом мы установили связь между таблицами registration и students по полю SID:

Внешний ключ	Первичный ключ
Таблица registration	Таблица students
Поле SID	Поле SID

Аналогично устанавливаем связь между таблицами registration и courses по полю CID:

Внешний ключ	Первичный ключ
Таблица registration	Таблица courses
Поле CID	Поле CID

Для этого мы должны, как и прежде, использовать свойство Внешний ключ для поля CID таблицы registration. Внизу диалогового окна Редактирование определения таблицы будет видно, что наши действия по связыванию таблиц были аналогичны следующим командам SQL:

```
FOREIGN KEY("CID") REFERENCES "courses" ("CID"),
FOREIGN KEY("SID") REFERENCES "students" ("SID")
```

Теперь пользуясь вкладкой Данные, приступаем к заполнению таблицы registration. Приводим пример этой таблицы

RID	SID	CID	Grade
1	S1	MAT-1	
2	S2	MAT-1	
3	S3	MAT-2	
4	S4	MAT-2	
5	S1	ФИЗ	
6	S2	ФИЗ	
7	S3	ФИЗ	
8	S4	ФИЗ	

Как мы видим, половина студентов записалась на курс MAT-1, другая половина студентов записалась на курс MAT-2. Все студенты записались на курс ФИЗ. Поле Grade остается пока незаполненным, так как оценка неизвестна.

Запросы SQL

Теперь переходим к выполнению SQL запросов для нашей базы данных. Для выполнения запросов переходим на вкладку SQL, которая находится правее

вкладки Данные. При этом появляется поле, в котором можно записывать запрос. Рассмотрим простой запрос, какие студенты записаны на курс MAT-1. Используем для этого таблицу registration. Записываем такую команду

```
select * from registration where registration.CID=>MAT-1»
```

Команда SQL выполняется после нажатия F5 или после нажатия кнопки Пуск в виде синего треугольника. Результат очевиден — мы получим две записи из таблицы registration

RID	SID	CID	Grade
1	S1	MAT-1	
2	S2	MAT-1	

Увеличим информативность данного примера, добавив в результирующую таблицу данного запроса имена и фамилии студентов, записавшихся на курс MAT-1. Имена и фамилии студентов хранятся, как мы помним, в таблице students. Поэтому для выполнения этого запроса мы используем связь между таблицами registration и students, установленную ранее. Используем, например, такую команду SQL

```
select registration.SID, registration.CID, students.
Lastname, students.Firstname from registration, students
where registration.SID=students.SID AND registration.
CID="MAT-1"
```

В этом запросе мы явно указываем на условие, что значение поля, которое является внешним ключом, должно совпадать со значением поля, которое является первичным ключом (registration.SID=students.SID). Результат выполнения этого запроса таков

SID	CID	Lastname	Firstname
S1	MAT-1	Витько	Никита
S2	MAT-1	Голованов	Андрей

Теперь выполним другой запрос, а именно выясним, на какие курсы записался студент с идентификатором S1. (Предварительно можно выяснить по таблице students, что этот идентификатор имеет студент Витько Никита.) Для этого можно выполнить такой запрос

```
select registration.SID, registration.CID, courses.Subject,
courses.Description from registration, courses where
registration.CID=courses.CID AND registration.SID="S1"
```

В этом запросе мы использовали связь между таблицами registration и courses по полю CID. Результатом выполнения этого запроса будет следующая таблица

SID	CID	Subject	Description
S1	МАТ-1	Математика	векторная алгебра
S1	ФИЗ	Физика	механика

До сих пор в запросах мы использовали связь между двумя таблицами. Но если мы хотим ответить на вопрос, на какие курсы записался студент с фамилией Голованов, то нам придется использовать в запросе три таблицы одновременно. Вот как это будет выглядеть

```
select registration.SID, registration.CID, courses.Subject,
courses.Description from registration, courses, students
where registration.CID=courses.CID AND registration.
SID=students.SID AND students.Lastname="Голованов"
```

Результат будет такой

SID	CID	Subject	Description
S2	МАТ-1	Математика	векторная алгебра
S2	ФИЗ	Физика	механика

Выясним теперь, какие курсы читаются в среду, используя команду

```
select courses.Subject from courses where courses.
Day="среда"
```

Результат будет очевиден

Subject
Физика

ЗАКЛЮЧЕНИЕ

В данной статье был рассмотрен пример построения базы данных с использованием программы DB Browser for SQLite. Рассмотренная база данных хранит информацию о регистрации студентов на учебные курсы. База данных состоит из трех таблиц: students (студенты), courses (курсы) и registration (регистрация), которые связаны между собой.

Программа DB Browser for SQLite является легкой в изучении, бесплатной и в то же время имеет обширные возможности при создании и управлении базами данных. Авторы рекомендуют эту программу при изучении курсов «Информационные технологии» и «Информатика».

ЛИТЕРАТУРА

1. Осипов Д. Л. Технологии проектирования баз данных. — Москва: изд-во ДМК, 2019. — 498 с.
2. Владстон Феррейра Фило. Теоретический минимум по computer science. — Санкт-Петербург: изд-во Питер, 2018. — 224 с.
3. Гинис Л. А. Создание проблемно-ориентированных баз данных в СУБД MS Access 2010 по дисциплине «Информационные ресурсы в кадастровой деятельности»: методические указания к выполнению практических работ. — Ростов-на-Дону: изд-во Южного федерального университета, 2014. — 20 с.

© Зоткин Сергей Петрович (zotkinsp@mgsu.ru).

Журнал «Современная наука: актуальные проблемы теории и практики»