

Ch12-데이터 저장과 관리



2025-01-01(분반)

AI 오픈소스 전문프로젝트

Fri 09:00~ (S4-1-201)

학습목표

- 데이터베이스의 기본 개념을 배운다.
- SQLite 사용법을 익힌다.
- SQLite를 이용해 앱을 개발한다.
- SQLite GUI 툴 사용법을 익힌다.

목차

01 SQLite의 기본

02 SQLite의 활용

01

SQLite의 기본

1. 데이터베이스 기본 개념

■ 데이터베이스 정의

- 대용량의 데이터 집합을 체계적으로 구성해놓은 것

■ 데이터베이스 관리 시스템

- 데이터베이스는 여러 사용자나 시스템이 서로 공유할 수 있어야 함
- 데이터베이스 관리 시스템(DataBase Management System, DBMS)
 - 이러한 데이터베이스를 관리해주는 시스템 또는 소프트웨어
- DBMS의 유형
 - 계층형(hierarchical), 망형(network), 관계형(relational), 객체지향형(object-oriented), 객체관계형(object-relational)

5 / 41

1. 데이터베이스 기본 개념

■ 관계형 데이터베이스

- 계층형, 망형, 관계형, 객체지향형, 객체관계형 DBMS 등의 유형 중 실질적으로 가장 많이 사용됨
- SQLite도 관계형 DBMS 속함

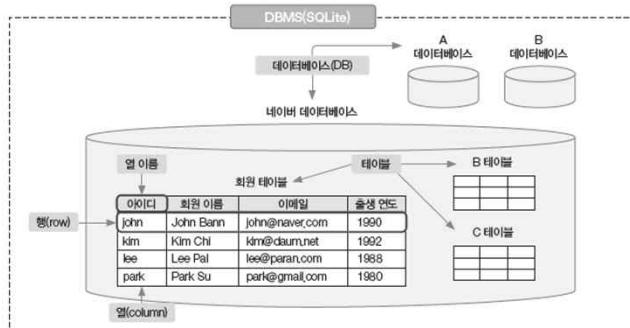
■ 관계형 데이터베이스의 장단점

- 장점
 - 업무가 변화할 경우에 다른 DBMS에 비해 변화에 쉽게 순응할 수 있는 구조
 - 유지 및 보수 측면에서도 편리
 - 대용량 데이터 관리와 데이터 무결성(Integration)을 잘 보장
- 단점
 - 시스템 자원을 많이 차지해서 시스템이 전반적으로 느려짐

6 / 41

1. 데이터베이스 기본 개념

■ 데이터베이스 관련 용어



7 / 41

1. 데이터베이스 기본 개념

■ 데이터베이스 관련 용어

- 데이터 : 하나하나의 단편적인 정보를 뜻함
- 테이블 : 회원 데이터가 표 형태로 표현된 것
- 데이터베이스(DB) : 테이블이 저장되는 장소로 주로 원통 모양으로 표현
 - 각 데이터베이스는 서로 다른 고유한 이름이 있어야 함
- DBMS : 데이터베이스를 관리하는 시스템 또는 소프트웨어를 말함
 - 안드로이드에 포함된 SQLite 소프트웨어가 이에 해당
- 열(칼럼 또는 필드) : 각 테이블은 1개 이상의 열로 구성됨
- 열 이름 : 각 열을 구분하는 이름, 열 이름은 각 테이블 안에서는 중복되지 않아야 함
- 데이터 형식 : 열의 데이터 형식을 뜻함
 - 테이블을 생성할 때 열 이름과 함께 지정해야 함
- 행(로우) : 실제 데이터
- SQL : 사용자와 DBMS가 소통하기 위한 언어

8 / 41

2. SQLite에서 데이터베이스 구축

■ SQLite에서 데이터베이스 구축

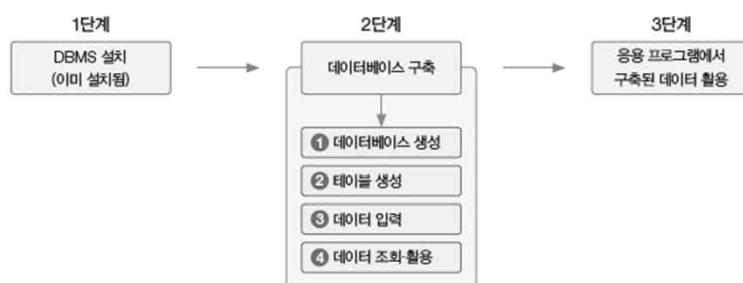


그림 12-2 데이터베이스 구축 및 운영 절차

9 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

- 0. 데이터베이스 생성 전 작업
 - (1) 프로젝트 이름 : Project12_1
 - (2) 패키지 이름 : com.cookandroid.project12_1
 - (3) 프로젝트를 실행해서 AVD 가동
 - (4) 명령 프롬프트를 실행한 후 adb.exe가 있는 폴더로 이동

C:\> CD C:\CookAndroid\SDK\platform-tools
C:\CookAndroid\SDK\platform-tools> DIR adb.exe
C 드라이브의 폴더에는 이름이 없습니다.
파일 검색 번호: F066-7FD4
C:\CookAndroid\SDK\platform-tools 디렉터리
2023-12-01 오후 03:39 [5,920,544 adb.exe]
 1개 파일 5,920,544 바이트를
 0개 디렉터리 29,988,532,224 바이트 남음
C:\CookAndroid\SDK\platform-tools>

그림 12-3 adb.exe 파일 확인

10 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

■ 0. 데이터베이스 생성 전 작업

- (5) 명령 프롬프트에서 다음 명령을 차례로 수행하면 SQLite에 접속할 준비가 된 것

샘플 이용한 데이터베이스 접속

```
1 adb root
2 adb shell
3 cd /data/data/com.cookandroid.project12_1
4 ls -l
5 mkdir databases
6 cd databases
7 pwd
```

C:\CookAndroid\SDK\platform-tools> adb root
adb is already running as root
C:\CookAndroid\SDK\platform-tools> adb shell
emu64t:/ #
emu64t:/ # cd /data/data/com.cookandroid.project12_1
emu64t:/data/data/com.cookandroid.project12_1 #
emu64t:/data/data/com.cookandroid.project12_1 # ls -l
total 16
drwxrws--x 2 ub_a297 ub_a297 cache 4896 2022-10-09 07:44 cache
drwxrws--x 2 ub_a297 ub_a297 code 4896 2022-10-09 07:49 code
drwxrws--x 2 ub_a297 ub_a297 lib 4896 2022-10-09 07:49 lib
emu64t:/data/data/com.cookandroid.project12_1 # mkdir databases
emu64t:/data/data/com.cookandroid.project12_1 # cd databases
emu64t:/data/data/com.cookandroid.project12_1# databases
emu64t:/data/data/com.cookandroid.project12_1# databases #
/data/data/com.cookandroid.project12_1# databases
emu64t:/data/data/com.cookandroid.project12_1# databases #|

그림 12-4 데이터베이스용 폴더 생성

11 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

■ 1. 데이터베이스 생성

SQLite 실행 및 데이터베이스 생성

```
1 sqlite3 naverDB
```

sqlite>

emu64t:/data/data/com.cookandroid.project12_1# sqlite3 naverDB
SQLite version 3.32.2 2021-07-12 15:00:17
Enter ".help" for usage hints.
sqlite> |

그림 12-5 데이터베이스 생성

12 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

■ 2. 테이블 생성

- 테이블을 생성하는 SQL문 형식

```
CREATE TABLE 테이블이름 (열이름1 데이터형식, 열이름2 데이터형식, ...);
```

테이블 생성 및 확인

```
1 CREATE TABLE userTable (id char(4), userName char(15),
                           email char(15), birthYear int);
2 .table
3 .schema userTable
```

```
C:\Users\Wondi\Downloads> + ~
Enter ".help" for usage hints.
sqlite>
sqlite> CREATE TABLE userTable ( id char(4), userName char(15), email char(15), birthYear int );
sqlite> .table
sqlite> userTable
sqlite> .schema userTable
CREATE TABLE userTable ( id char(4), userName char(15), email char(15), birthYear int );
sqlite> |
```

그림 12-6 테이블 생성 및 확인

13 / 41

2. SQLite에서 데이터베이스 구축

여기서 잠깐



SQL문의 규칙

SQL문은 대·소문자를 구분하지 않는다. 또한 모든 SQL문의 끝에는 세미콜론(:)을 붙여야 한다. 예외적으로 SQLite 자체 명령은 소문자로 써야 하고 시작에 마침표(.)를 붙이며 끝에 세미콜론을 붙이지 않아도 된다. 자주 사용하는 SQLite 명령은 다음과 같다.

- `.table`: 현재 데이터베이스의 테이블 목록을 보여준다.
- `.schema 테이블이름`: 테이블의 열, 데이터 형식 등의 정보를 보여준다.
- `.header on`: SELECT문으로 출력할 때 헤더를 보여준다.
- `.mode column`: SELECT문으로 출력할 때 칼럼 모드로 출력해준다.
- `.exit`: SQLite를 종료한다.

일반적으로 SELECT문을 사용하기 전에 `.header on`과 `.mode column`을 설정하면 결과 화면이 보기 좋게 출력된다.

```
sqlite> select * from userTable;
kim1 kim1 chikim1@test.com|2020
kim1 kim1 chikim1@test.com|2021
kim2 kim2 chikim2@test.com|2022
```



```
sqlite> select * from userTable;
id      username   email           birthyear
-----+-----+-----+-----+
kim1   kim1 chi   kim1@test.com 2020
kim1   kim1 chi   kim1@test.com 2021
kim2   kim2 chi   kim2@test.com 2022
```

14 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

■ 3. 데이터 입력

- 행 데이터를 입력하는 SQL문 형식

```
INSERT INTO 테이블이름 VALUES(값1, 값2, ...);
```

데이터 4건 입력

```
1 INSERT INTO userTable VALUES('john', 'John Bann', 'john@naver.com', 1990);
2 INSERT INTO userTable VALUES('kim', 'Kim Chi', 'kim@daum.net', 1992);
3 INSERT INTO userTable VALUES('lee', 'Lee Pal', 'lee@paran.com', 1988);
4 INSERT INTO userTable VALUES('park', 'Park Su', 'park@gmail.com', 1980);
```

```
sqlite> INSERT INTO userTable VALUES('john', 'John Bann', 'john@naver.com', 1990);
sqlite> INSERT INTO userTable VALUES('kim', 'Kim Chi', 'kim@daum.net', 1992);
sqlite> INSERT INTO userTable VALUES('lee', 'Lee Pal', 'lee@paran.com', 1988);
sqlite> INSERT INTO userTable VALUES('park', 'Park Su', 'park@gmail.com', 1980);
sqlite> |
```

그림 12-7 데이터 4건 입력

15 / 41

2. SQLite에서 데이터베이스 구축

■ 실습 12-1 네이버 데이터베이스 구축하기

■ 4. 데이터 조회 · 활용

- 데이터를 조회 · 활용하는 SQL문 형식

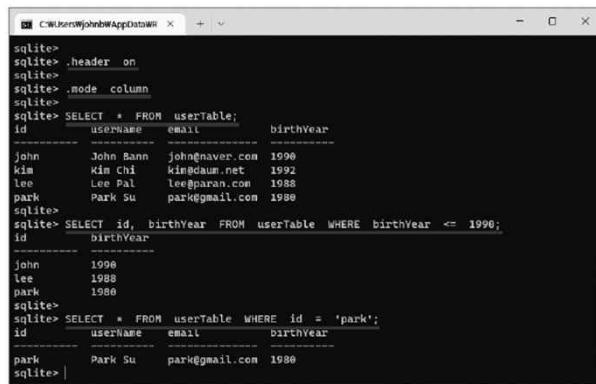
```
SELECT 열이름1, 열이름2, ... FROM 테이블이름 WHERE 조건;
```

데이터 조회의 예

```
1 .header on
2 .mode column
3 SELECT * FROM userTable;
4 SELECT id, birthYear FROM userTable WHERE birthYear <= 1990;
5 SELECT * FROM userTable WHERE id = 'park';
```

16 / 41

2. SQLite에서 데이터베이스 구축



```
sqlite> .header on
sqlite> .mode column
sqlite> SELECT * FROM userTable;
id      userName    email           birthYear
----- -----
john    John Bann  john@naver.com  1990
kim     Kim Chi    kim@daum.net   1992
lee     Lee Pal    lee@paran.com  1988
park    Park Su    park@gmail.com 1980
sqlite> SELECT id, birthYear FROM userTable WHERE birthYear <= 1990;
id      birthYear
----- -----
john    1990
lee     1988
park    1980
sqlite> SELECT * FROM userTable WHERE id = 'park';
id      userName    email           birthYear
----- -----
park    Park Su    park@gmail.com 1980
sqlite> |
```

그림 12-8 데이터 조회의 예

17 / 41

02

SQLite의 활용

1. SQLite 프로그래밍

■ 안드로이드 앱 개발을 위한 SQLite 동작 방식

- SQLiteOpenHelper 클래스, SQLiteDatabase 클래스, Cursor 인터페이스 활용

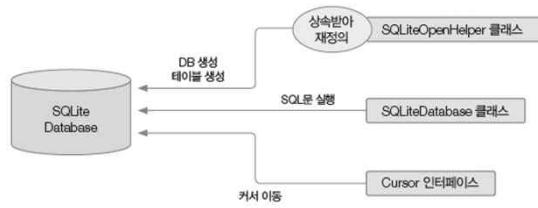


그림 12-9 SQLite 관련 클래스의 동작

19 / 41

1. SQLite 프로그래밍

■ 각 클래스에서 주로 사용되는 메소드

표 12-1 SQLite 관련 클래스 및 인터페이스와 메소드

클래스 또는 인터페이스	메소드	주요 용도
SQLiteOpenHelper 클래스	생성자	DB 생성
	onCreate()	테이블 생성
	onUpgrade()	테이블 삭제 후 다시 생성
	getReadableDatabase()	읽기 전용 DB 열기, SQLiteDatabase 반환
	getWritableDatabase()	읽고 쓰기용 DB 열기, SQLiteDatabase 반환
SQLiteDatabase 클래스	execSQL()	SQL문(Insert/Update/Delete) 실행
	close()	DB 닫기
	query(), rawQuery()	Select 실행 후 커서 반환
Cursor 인터페이스	moveToFirst()	커서의 첫 행으로 이동
	moveToLast()	커서의 마지막 행으로 이동
	moveToNext()	현재 커서의 다음 행으로 이동

20 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 1 안드로이드 프로젝트 생성
 - (1) 프로젝트 이름 : Project12_2
 - (2) 패키지 이름 : com.cookandroid.project12_2



그림 12-10 가수 그룹 관리 DB 앱 결과 화면

21 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 2 화면 디자인 및 편집
 - activity_main.xml의 바깥 레이아웃 안에 4개의 라이아웃(수평)을 만들고 화면을 다음과 같이 구성하기
 - 라이아웃1~3은 layout_weight를 1로 라이아웃4는 8로 설정
 - 라이아웃1 : 텍스트뷰 1개, 에디트텍스트(edtName) 1개
 - 라이아웃2 : 텍스트뷰 1개, 에디트텍스트(edtNumber) 1개
 - 라이아웃3 : 버튼 3개(btnInit, btnInsert, btnSelect)
 - 라이아웃4 : 에디트텍스트 2개 (edtNameResult, edtNumberResult)

22 / 41

1. SQLite 프로그래밍

예제 12-1 activity_main.xml

```
1 <LinearLayout>
2     <LinearLayout>
3         android:layout_width="fill_parent"
4         android:layout_height="60dp"
5         android:layout_weight="1"
6         android:orientation="horizontal" >
7             <TextView
8                 android:text="이름 : " />
9             <EditText
10                android:id="@+id/edtName"
11                android:layout_weight="1" />
12         </LinearLayout>
13         <LinearLayout>
14             ---- 생략(텍스트뷰, 에디트텍스트) ----
15         </LinearLayout>
16         <LinearLayout>
17             <Button
18                 android:id="@+id/btnInit"
19                 android:text="초기화" />
20             ---- 생략(버튼 2개) ----
21         </LinearLayout>
22         <LinearLayout>
23             <EditText
24                 android:id="@+id/edtNameResult"
25                 android:layout_weight="1" />
26             ---- 생략(에디트텍스트) ----
27         </LinearLayout>
28     </LinearLayout>
```

23 / 41

1. SQLite 프로그래밍

■ 실습 12-2 가수 그룹 관리 DB 만들기

■ 3 Java 코드 작성 및 수정

- (1) SQLiteOpenHelper 클래스에서 상속받은 클래스를 정의한 후 생성자 수정

예제 12-2 Java 코드 1

```
1    @Override
2    public void onCreate(Bundle savedInstanceState) {
3        super.onCreate(savedInstanceState);
4        setContentView(R.layout.activity_main);
5    }
6
7    public class myDBHelper extends SQLiteOpenHelper {
8        public myDBHelper(Context context) {
9            super(context, "groupDB", null, 1);
10       }
11    @Override
12    public void onCreate(SQLiteDatabase db) {
13
14    }
15    @Override
16    public void onUpgrade(SQLiteDatabase db, int oldVersion,
17                         int newVersion) {
18
19    }
20 }
```

24 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 3 Java 코드 작성 및 수정
 - (2) myDBHelper 클래스의 onCreate()와 onUpgrade() 메소드 코딩
 - onCreate() 메소드 : 테이블을 생성하는 기능을 코딩
 - onUpgrade() 메소드 : 테이블을 삭제한 후 다시 생성

예제 12-3 Java 코드 2

```
1 @Override
2 public void onCreate(SQLiteDatabase db) {
3     db.execSQL("CREATE TABLE groupTBL ( gName CHAR(20) PRIMARY KEY,
4                                     gNumber INTEGER);");
5 }
6 @Override
7 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
8     db.execSQL("DROP TABLE IF EXISTS groupTBL");
9     onCreate(db);
10 }
```

25 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 3 Java 코드 작성 및 수정
 - (3) 메인 액티비티 클래스에 다음 변수를 선언하고 onCreate()에서는 위젯 변수에 activity_main.xml의 7개 위젯을 대입
 - 새로 생성한 myDBHelper 클래스 변수
 - 에디트텍스트에 대응할 변수 4개
 - 버튼에 대응할 변수 3개
 - SQLiteDatabase 클래스 변수

26 / 41

1. SQLite 프로그래밍

예제 12-4 Java 코드 3

```
1 public class MainActivity extends AppCompatActivity {
2
3     myDBHelper myHelper;
4     EditText edtName, edtNumber, edtNameResult, edtNumberResult;
5     Button btnInit, btnInsert, btnSelect;
6     SQLiteDatabase sqlDB;
7
8     @Override
9     public void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12         setTitle("가수 그룹 관리 DB");
13
14         edtName = (EditText) findViewById(R.id.edtName);
15         edtNumber = (EditText) findViewById(R.id.edtNumber);
16         edtNameResult = (EditText) findViewById(R.id.edtNameResult);
17         edtNumberResult = (EditText) findViewById(R.id.edtNumberResult);
18         btnInit = (Button) findViewById(R.id.btnInit);
19         btnInsert = (Button) findViewById(R.id.btnInsert);
20         btnSelect = (Button) findViewById(R.id.btnSelect);
21     }
22     ~~~~ 생략 ~~~~
```

27 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 3 Java 코드 작성 및 수정
 - (4) <초기화>를 클릭했을 때 동작하는 리스너 코드

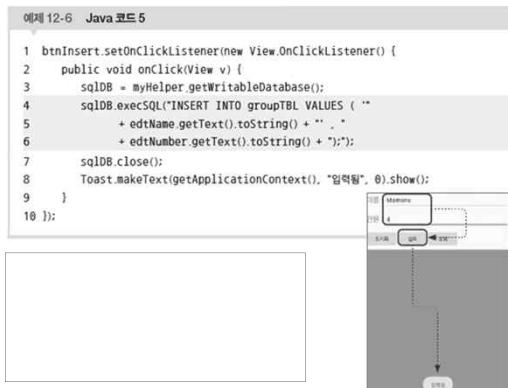
예제 12-5 Java 코드 4

```
1 myHelper = new myDBHelper(this);
2 btnInit.setOnClickListener(new View.OnClickListener() {
3     public void onClick(View v) {
4         sqlDB = myHelper.getWritableDatabase();
5         myHelper.onUpgrade(sqlDB, 1, 2);
6         sqlDB.close();
7     }
8 });
```

28 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 3 Java 코드 작성 및 수정
 - (5) <입력>을 클릭하면 에디트텍스트의 값이 입력되는 리스너 코딩
 - 실행하여 데이터를 몇 건 입력했을 때, 지금까지 코딩에 오류가 없다면 ‘입력됨’ 메시지가 나옴



29 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 3 Java 코드 작성 및 수정
 - (6) <조회>를 클릭할 때, 테이블에 입력된 내용이 모두 아래쪽 에디트텍스트에 출력되는 리스너 코딩

30 / 41

1. SQLite 프로그래밍

```
예제 12-7 Java 코드 6

1 btnSelect.setOnClickListener(new View.OnClickListener() {
2     public void onClick(View v) {
3         sqlDB = myHelper.getReadableDatabase();
4         Cursor cursor;
5         cursor = sqlDB.rawQuery("SELECT * FROM groupTBL;", null);
6
7         String strNames = "그룹 이름" + "\r\n" + "-----" + "\r\n";
8         String strNumbers = "인원" + "\r\n" + "-----" + "\r\n";
9
10        while (cursor.moveToNext()) {
11            strNames += cursor.getString(0) + "\r\n";
12            strNumbers += cursor.getString(1) + "\r\n";
13        }
14
15        edtNameResult.setText(strNames);
16        edtNumberResult.setText(strNumbers);
17
18        cursor.close();
19        sqlDB.close();
20    }
21});
```

31 / 41

1. SQLite 프로그래밍

- 실습 12-2 가수 그룹 관리 DB 만들기
- 4 프로젝트 실행 및 결과 확인
 - (1) 프로젝트를 실행한 후 데이터를 입력하고 조회
 - (2) 명령 프롬프트에서 확인

명령 프롬프트에서의 데이터베이스 접근

```
1 CD \CookAndroid\ sdk\platform-tools\
2 adb root
3 adb shell
4 # cd /data/data/com.cookandroid.project12_2
5 # cd databases
6 # sqlite3 groupDB
7 sqlite> .header on
8 sqlite> .mode column
9 sqlite> SELECT * FROM groupTBL;
```

```
C:\> CD \CookAndroid\ sdk\platform-tools\adb shell
C:\CookAndroid\SDK\platform-tools> adb root
adb is already running as root
C:\CookAndroid\SDK\platform-tools> cd /data/data/com.cookandroid.project12_2
C:\data\data\com.cookandroid.project12_2> cd databases
C:\data\data\com.cookandroid.project12_2\database> sqlite3 groupDB
SQLite version 3.32.2 2021-07-12 15:00:17
Enter ".help" for usage hints.
sqlite> .header on
sqlite> .mode column
sqlite> SELECT * FROM groupTBL;
  name    gnumber
  ----- 
  Manau      4
  Oh my Girl 7
  Black Pink 4
  Redie      9
sqlite> |
```

그림 12-11 명령 프롬프트에서의 데이터베이스 조회 화면

32 / 41

1. SQLite 프로그래밍

▶ 직접 풀어보기 12-2

- [실습 12-2]에 <수정>과 <삭제>를 추가하라.
- '이름'에 그룹 이름과 변경된 인원을 입력한 후 <수정>을 클릭하면 해당 그룹의 인원이 변경된다.
 - '이름'에 그룹 이름을 입력하고 <삭제>를 클릭하면 해당 그룹의 행이 삭제된다.
 - <입력>, <수정>, <삭제>를 클릭하면 그 결과가 즉시 화면에 출력된다.

▶ [문제]

- 수정 SQL: UPDATE groupTBL SET gNumber = 변경된_인원 WHERE gName = "변경할_그룹_이름";
- 삭제 SQL: DELETE FROM groupTBL WHERE gName = "삭제_그룹_이름";
- 입력/수정/삭제 후 즉시 결과가 보이게 하려면 btnSelect.onClick() 을 호출한다.

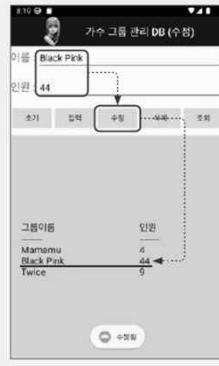


그림 12-12 수정된 가수 그룹 관리 DB 앱

33 / 41

2. SQLite GUI 툴 활용

■ DB Browser for SQLite

- SQLite에 접근할 때 명령 프롬프트를 이용했지만 DB Browser for SQLite라는 GUI 툴을 사용하면 좀 더 편리함

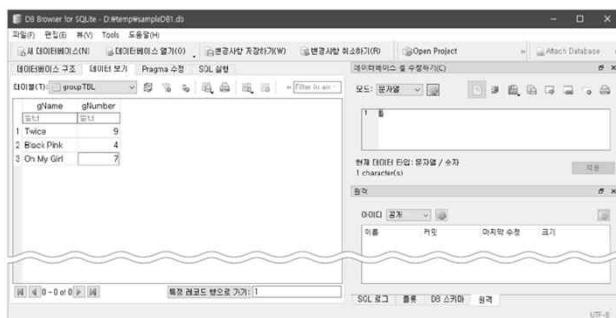


그림 12-13 DB Browser for SQLite 화면

34 / 41

2. SQLite GUI 툴 활용

■ DB Browser for SQLite에서 DB 및 테이블 생성

- [파일]-[새 데이터베이스]를 선택 - [저장하려는 파일명을 고르세요] 창에서 데이터베이스 파일이 저장될 경로와 파일명을 지정하고 <저장> 클릭

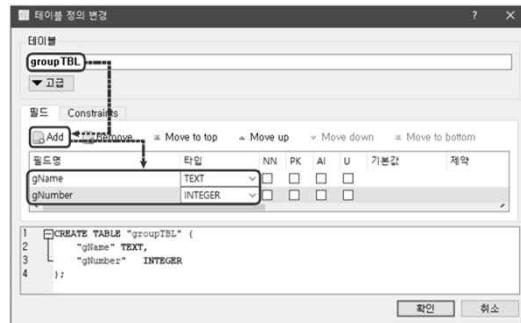


그림 12-14 DB Browser for SQLite에서의 데이터베이스 및 테이블 생성

35 / 41

2. SQLite GUI 툴 활용

■ DB Browser for SQLite에서 데이터 입력

- (1) [데이터 보기] 탭 클릭
- (2) 데이터 입력 후 메뉴의 [파일]-[변경사항 저장하기] 선택 - 변경 사항 저장
- (3) 생성한 데이터베이스 파일을 Device File Explorer를 통해 AVD에 넣어서 사용

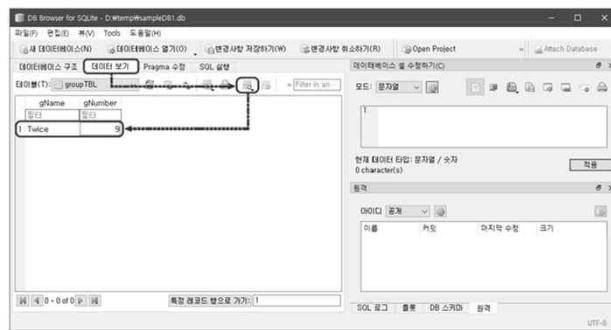


그림 12-15 DB Browser for SQLite에서의 행 데이터 입력

36 / 41

2. SQLite GUI 툴 활용

▶ 직접 풀이보기 12-3

Device Explorer를 이용하여 [실습 12-2]의 groupDB 데이터베이스 파일을 AVD에서 PC로 가져온 (Save As) 후, DB Browser for SQLite에서 가수 그룹 이름을 모두 한글로 고치고 가수 그룹을 몇 개 더 입력하라. 그런 다음 AVD에 다시 넣고(Upload) [실습 12-2]를 실행하여 데이터가 잘 조회되는지 확인 하라.



그림 12-16 SQLite를 활용한 가수 그룹 관리 앱

37 / 41

2. SQLite GUI 툴 활용

■ SQLite Developer

- DB Browser for SQLite와 마찬가지로 그래픽 화면에서 데이터베이스를 관리하기 위한 툴
- <http://www.sqlitedeveloper.com/download>에서 다운로드하여 설치
- [Database]-[Register Database]로 편집할 데이터베이스 파일을 선택 후 데이터 추가

38 / 41

2. SQLite GUI 를 활용

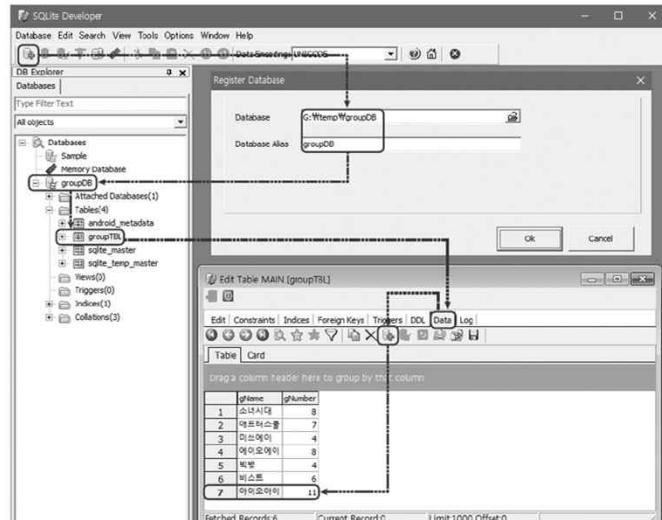


그림 12-17 SQLite Developer에서의 행 데이터 처리

39 / 41

감사합니다.

