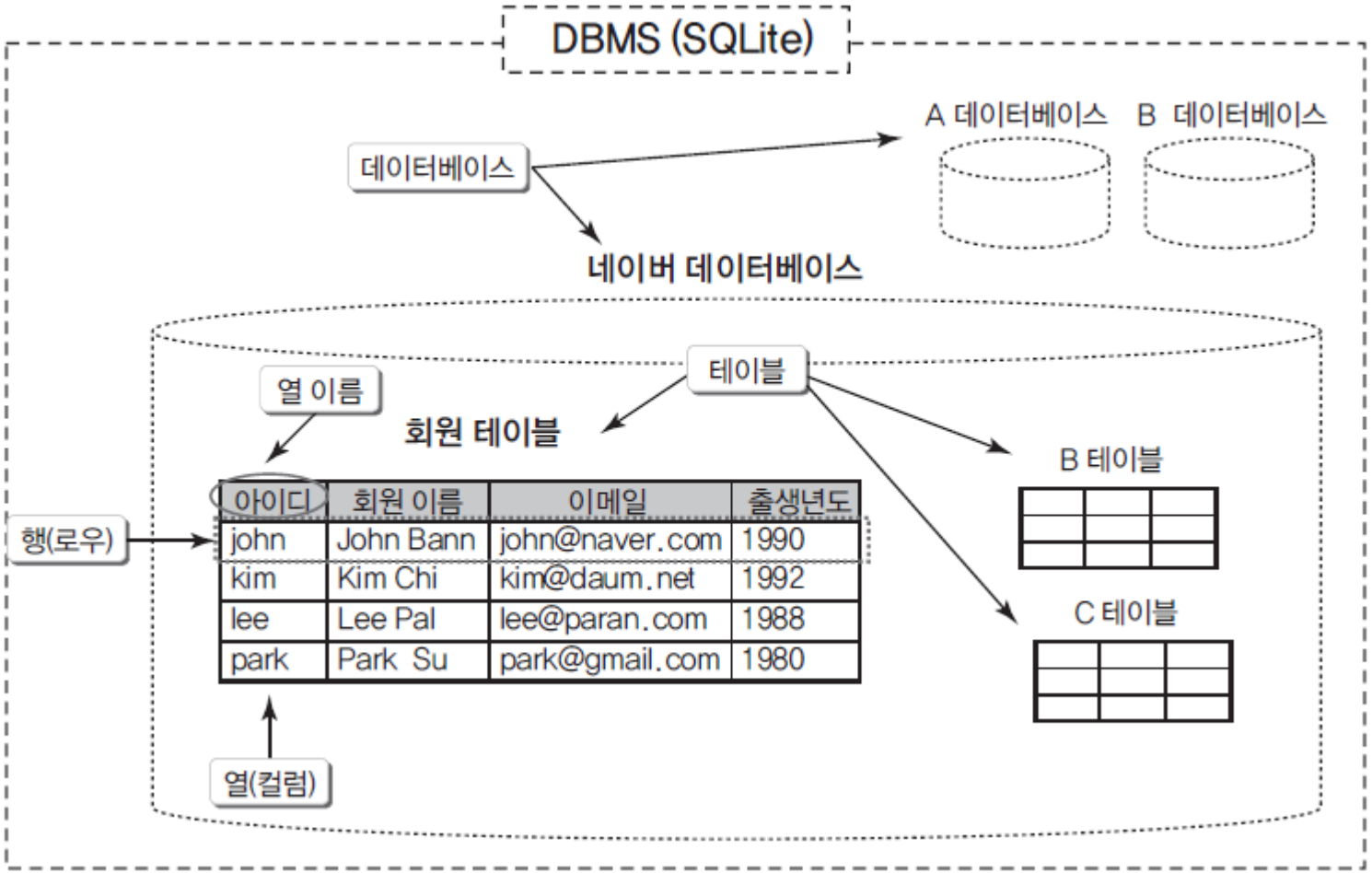
데이터 저장과 관리

Instructor: Park, JoonSeok (pjs50@pusan.ac.kr)

SQLite

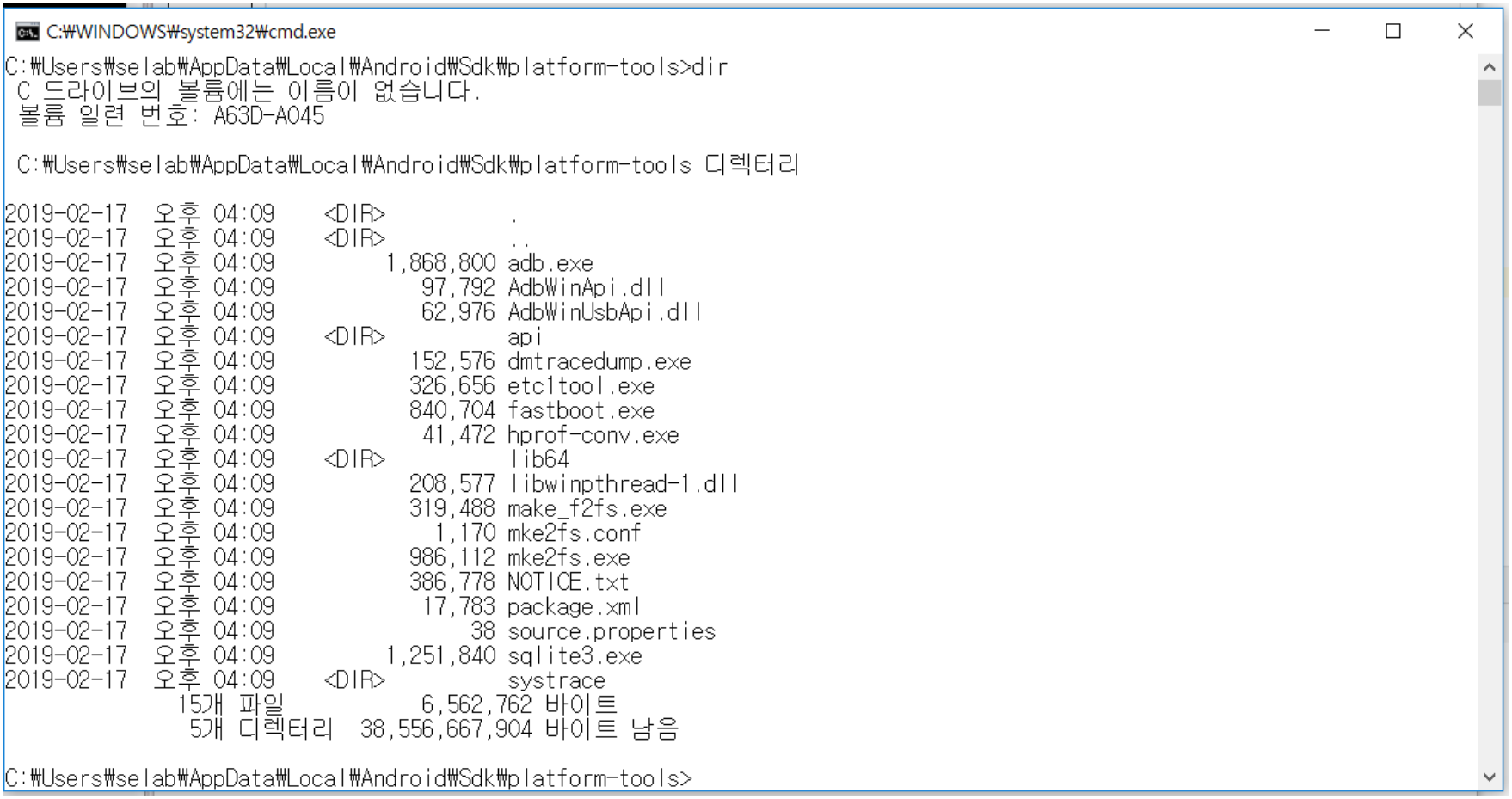
- SQLite는 모바일환경에 최적화된 데이터베이스로, 기본적인 개념과 동작은 PC용 데이터베이스와 거의 동일
- 모바일 환경에서 동작하는 SQLite는 관계형 DBMS의 단점을 극복하기 위해 동시접근이나 대용량 데이터관리에 제한이 있지만, 관계형 DBMS의 표준 SQL이나 개념 등은 동일하게 적용됨

SQLite


DBMS (SQLite)
A 데이터베이스
B 데이터베이스
데이터베이스
네이버 데이터베이스
테이블
열 이름
회원 테이블
B 테이블
C 테이블
행(로우)
아이디
회원 이름
이메일
출생년도
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
열(컬럼)

SQLite

- 데이터베이스 구축 – AVD 실행 후



SQLite

- 자주 사용하는 명령어
  - .tabel: 현재 데이터베이스의 테이블 목록을 보여줌
  - .schema: 테이블 이름, 테이블의 열, 데이터 형식 등의 정보를 보여줌
  - .header on: Select 문으로 출력할 때 헤더를 보여줌
  - .mode column: Select 문으로 출력할 때 컬럼 모드를 출력함
  - .exit: SQLite를 종료

애뮬레이터 실행 후

C:₩WINDOWS₩system32₩cmd.exe - adb shell

```
C:₩Users₩selab₩AppData₩Local₩Android₩Sdk₩platform-tools>adb root

C:₩Users₩selab₩AppData₩Local₩Android₩Sdk₩platform-tools>adb shell
generic_x86:/ #

generic_x86:/ # ls
acct       cache   data         init                init.usb.rc      mnt  product sys
adb_keys   charger default.prop init.environ.rc     init.zygote32.rc odm  sbin    system
bin        config  dev          init.rc             lost+found       oem  sdcard  ueventd.rc
bugreports d       etc          init.usb.configfs.rc metadata         proc storage vendor
generic_x86:/ #
```

```
generic_x86:/ # cd data
generic_x86:/data # ls
adb           app-lib       cache         lost+found misc_de     property       system_de  vendor_ce
anr           app-private   dalvik-cache  media      nativetest  resource-cache tombstones vendor_de
app           backup        data          mediadrm   nfc         ss             user
app-asec      benchmarktest drm           misc       ota         system         user_de
app-ephemeral bootchart     local         misc_ce    ota_package system_ce       vendor
generic_x86:/data # cd data
generic_x86:/data/data #
```

```
generic_x86:/data/data # cd com.example.selab.db_test
generic_x86:/data/data/com.example.selab.db_test # ls -l
total 8
drwxrws--x 2 u0_a85 u0_a85_cache 4096 2019-02-24 11:20 cache
drwxrws--x 2 u0_a85 u0_a85_cache 4096 2019-02-24 11:20 code_cache
generic_x86:/data/data/com.example.selab.db_test #
```

```
generic_x86:/data/data/com.example.selab.db_test # mkdir databases
generic_x86:/data/data/com.example.selab.db_test # ls -la
total 24
drwx------   5 u0_a85 u0_a85       4096 2019-02-24 11:32 .
drwxrwx--x 111 system system       4096 2019-02-24 11:20 ..
drwxrws--x   2 u0_a85 u0_a85_cache 4096 2019-02-24 11:20 cache
drwxrws--x   2 u0_a85 u0_a85_cache 4096 2019-02-24 11:20 code_cache
drwxrwxrwx   2 root   root         4096 2019-02-24 11:32 databases
generic_x86:/data/data/com.example.selab.db_test #
```

- 데이터베이스 생성

```
generic_x86:/data/data/com.example.selab.db_test # sqlite3 myDB
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite>
```

테이블 생성

```
sqlite> CREATE TABLE userTable(id char(4), username char(15), email char(15), birthyear int);
sqlite>
```

```
sqlite> .table
userTable
sqlite> .schema userTable
CREATE TABLE userTable(id char(4), username char(15), email char(15), birthyear int);
sqlite>
```

데이터 입력

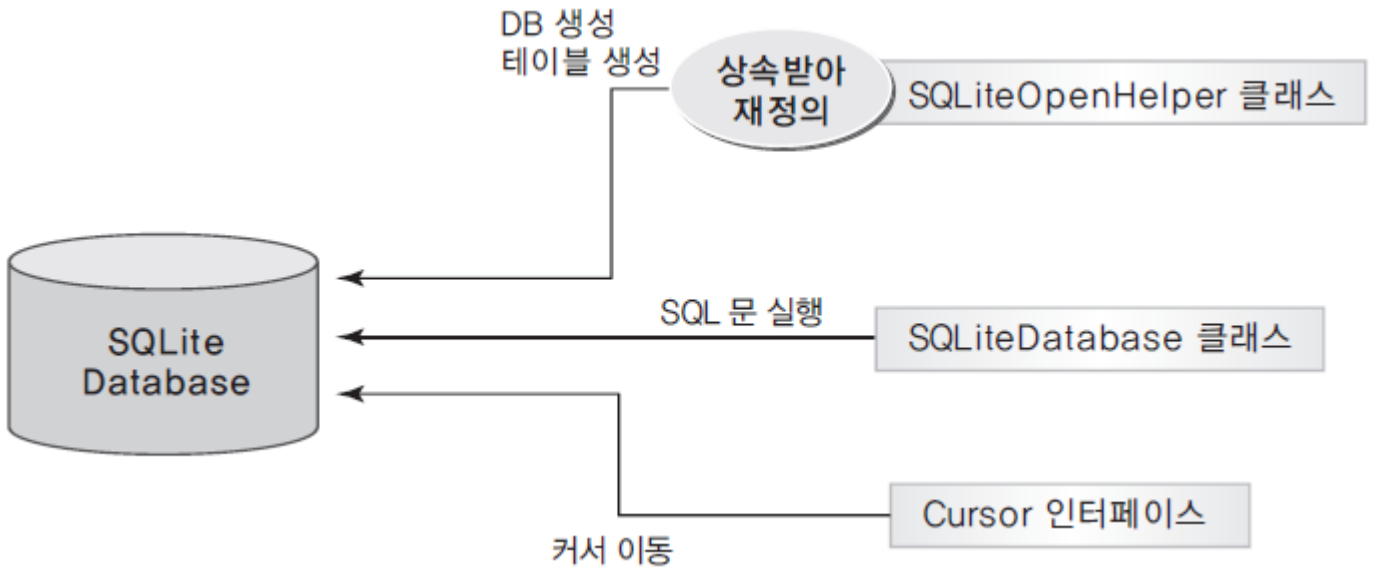
```
sqlite> insert into userTable values('kim', 'hong', 'hong@test.com', 2019);
```

```
sqlite> INSERT INTO userTable VALUES('a', 'a b', 'a@naver.com', 1990);
INSERT INTO userTable VALUES('a', 'a b', 'a@naver.com', 1990);
sqlite> INSERT INTO userTable VALUES('b', 'c c', 'b@naver.com', 2004);
INSERT INTO userTable VALUES('b', 'c c', 'b@naver.com', 2004);
sqlite> INSERT INTO userTable VALUES('d', 'c d', 'c@naver.com', 2013);
INSERT INTO userTable VALUES('d', 'c d', 'c@naver.com', 2013);
sqlite> INSERT INTO userTable VALUES('e', 'r d', 'd@naver.com', 2014);
INSERT INTO userTable VALUES('e', 'r d', 'd@naver.com', 2014);
```

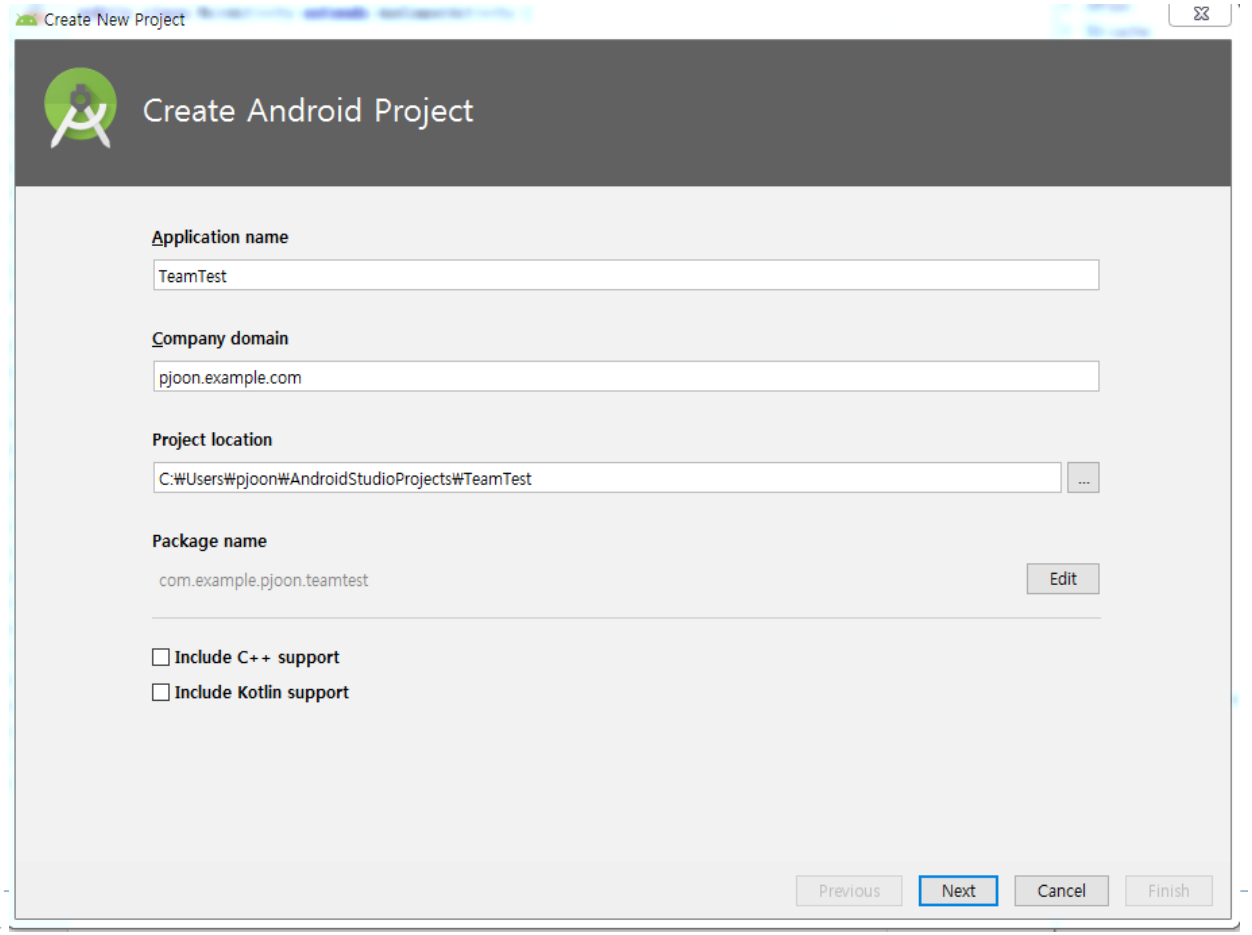
데이터 조회, 활용

```
sqlite> .header on
.header on
sqlite> .mode column
.mode column
sqlite> Select * from userTable;
Select * from userTable;
id          userName    email         birthYear
----------  ----------  ------------  ----------
a           a b         a@naver.com   1990
b           c c         b@naver.com   2004
d           c d         c@naver.com   2013
e           r d         d@naver.com   2014

sqlite> select id, birthYear  from userTable where birthyear > 2004;
select id, birthYear  from userTable where birthyear > 2004;
id          birthYear
----------  ----------
d           2013
e           2014
sqlite> select * from userTable where id ='a';
select * from userTable where id ='a';
id          userName    email         birthYear
----------  ----------  ------------  ----------
a           a b         a@naver.com   1990
sqlite>
```

- SQLiteOpenHelper 클래스, SQLiteDatabase 클래스, Cursor 인터페이스를 활용



- 팀의 이름과 인원을 데이터베이스에 입력하고 조회하는 응용 프로그램을 작성



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/a
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:orientation="horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="이름 : "
            android:textSize="20dp" />

        <EditText
            android:id="@+id/edtName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:orientation="horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="인원 : "
            android:textSize="20dp" />

        <EditText
            android:id="@+id/edtNumber"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1" >
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="0dip"
        android:layout_weight="1"
        android:orientation="horizontal" >

        <Button
            android:id="@+id/btnInit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="초기화" />

        <Button
            android:id="@+id/btnInsert"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="입력" />

        <Button
            android:id="@+id/btnSelect"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="조회" />
    </LinearLayout>
```

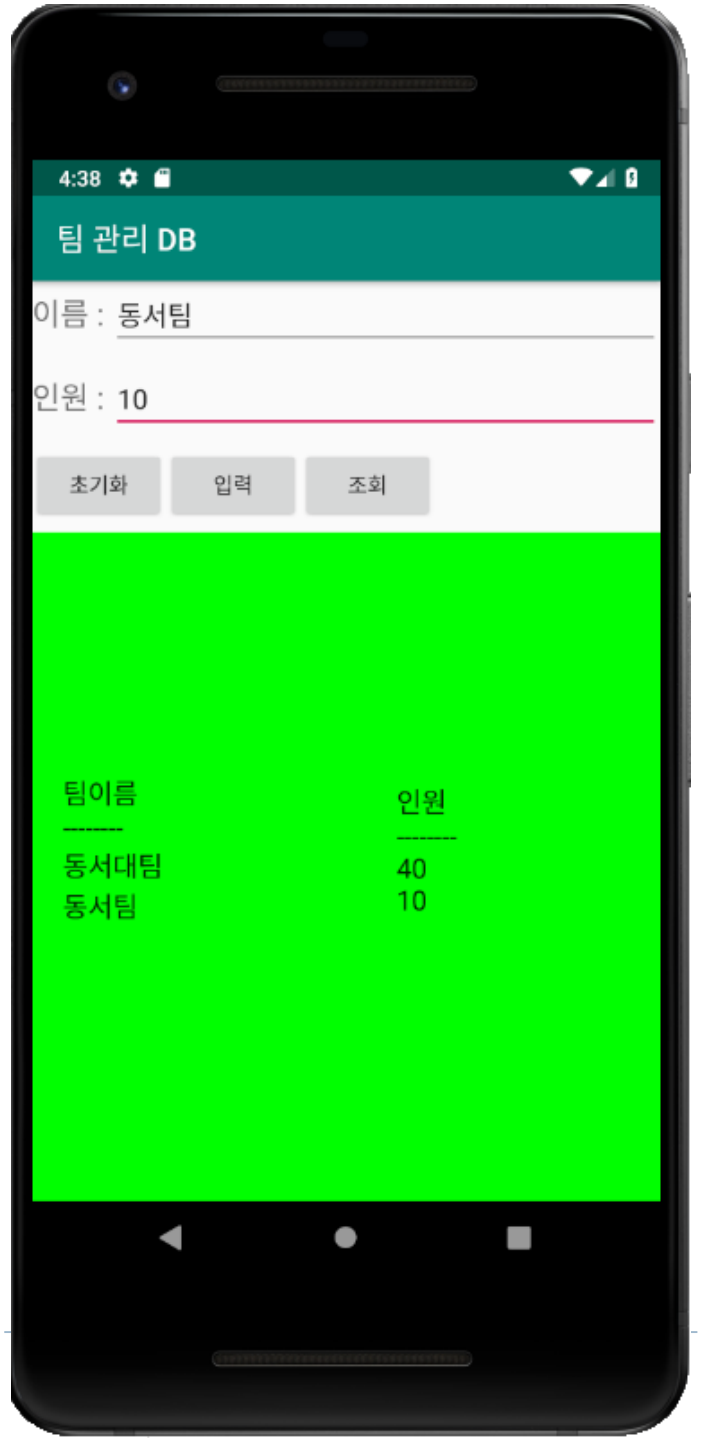
```
71
72      <LinearLayout
73          android:layout_width="fill_parent"
74          android:layout_height="0dip"
75          android:layout_weight="8"
76          android:orientation="horizontal" >
77
78          <EditText
79              android:id="@+id/edtNameResult"
80              android:layout_width="wrap_content"
81              android:layout_height="match_parent"
82              android:layout_weight="1"
83              android:background="#00FF00"
84              android:padding="20dp" />
85
86          <EditText
87              android:id="@+id/edtNumberResult"
88              android:layout_width="wrap_content"
89              android:layout_height="match_parent"
90              android:layout_weight="1"
91              android:background="#00FF00"
92              android:padding="20dp" />
93      </LinearLayout>
94
95  </LinearLayout>
```

```java
public class MainActivity extends AppCompatActivity {

    myDBHelper myHelper;
    EditText edtName, edtNumber, edtNameResult, edtNumberResult;
    Button btnInit, btnInsert, btnSelect;
    SQLiteDatabase sqlDB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        setTitle("팀 관리 DB");

        edtName = (EditText) findViewById(R.id.edtName);
        edtNumber = (EditText) findViewById(R.id.edtNumber);
        edtNameResult = (EditText) findViewById(R.id.edtNameResult);
        edtNumberResult = (EditText) findViewById(R.id.edtNumberResult);

        btnInit = (Button) findViewById(R.id.btnInit);
        btnInsert = (Button) findViewById(R.id.btnInsert);
        btnSelect = (Button) findViewById(R.id.btnSelect);

    }
```

```
34
35      public class myDBHelper extends SQLiteOpenHelper {
36          public myDBHelper(Context context) {
37              super(context, name: "groupDB", factory: null, version: 1);
38          }
39
40          @Override
41          public void onCreate(SQLiteDatabase db) {
42              db.execSQL("CREATE TABLE  groupTBL ( gName CHAR(20) PRIMARY KEY, gNumber INTEGER);");
43          }
44
45          @Override
46          public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
47
48              db.execSQL("DROP TABLE IF EXISTS groupTBL");
49              onCreate(db);
50
51          }
52      }
53
```

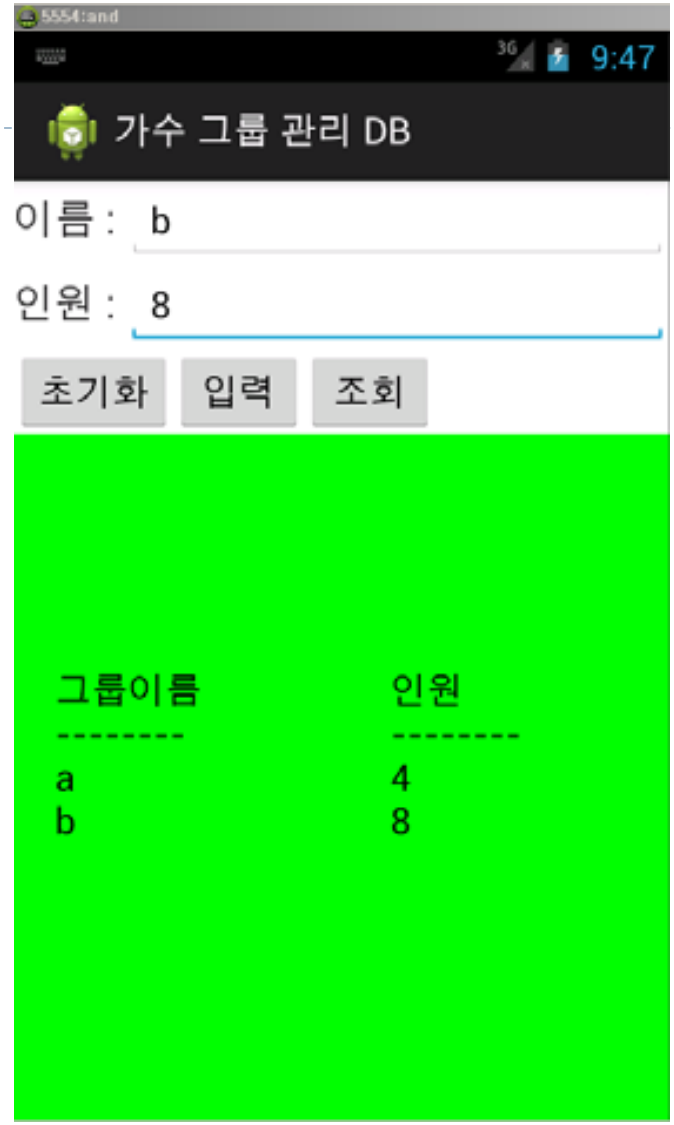
```
22  protected void onCreate(Bundle savedInstanceState) {
23      super.onCreate(savedInstanceState);
24      setContentView(R.layout.activity_main);
25      setTitle("팀 관리 DB");
26
27      edtName = (EditText) findViewById(R.id.edtName);
28      edtNumber = (EditText) findViewById(R.id.edtNumber);
29      edtNameResult = (EditText) findViewById(R.id.edtNameResult);
30      edtNumberResult = (EditText) findViewById(R.id.edtNumberResult);
31
32      btnInit = (Button) findViewById(R.id.btnInit);
33      btnInsert = (Button) findViewById(R.id.btnInsert);
34      btnSelect = (Button) findViewById(R.id.btnSelect);
35
36      myHelper = new myDBHelper( context: this);
37      btnInit.setOnClickListener(new View.OnClickListener() {
38          public void onClick(View v) {
39              sqlDB = myHelper.getWritableDatabase();
40              myHelper.onUpgrade(sqlDB, oldVersion: 1, newVersion: 2); // 인수는 아무거나 입력하면 됨.
41              sqlDB.close();
42          }
43      });
```

```
45      btnInsert.setOnClickListener(new View.OnClickListener() {
46          public void onClick(View v) {
47              sqlDB = myHelper.getWritableDatabase();
48              sqlDB.execSQL("INSERT INTO groupTBL VALUES ( '"
49                      + edtName.getText().toString() + "' , "
50                      + edtNumber.getText().toString() + ");");
51              sqlDB.close();
52              Toast.makeText(getApplicationContext(), text: "입력됨",
53                      Toast.LENGTH_SHORT).show();
54          }
55      });
56
57      btnSelect.setOnClickListener(new View.OnClickListener() {
58          public void onClick(View v) {
59              sqlDB = myHelper.getReadableDatabase();
60              Cursor cursor;
61              cursor = sqlDB.rawQuery( sql: "SELECT * FROM groupTBL;", selectionArgs: null);
62
63              String strNames = "팀이름" + "₩r₩n" + "--------" + "₩r₩n";
64              String strNumbers = "인원" + "₩r₩n" + "--------" + "₩r₩n";
65
66              while (cursor.moveToNext()) {
67                  strNames += cursor.getString( columnIndex: 0) + "₩r₩n";
68                  strNumbers += cursor.getString( columnIndex: 1) + "₩r₩n";
69              }
70
71              edtNameResult.setText(strNames);
72              edtNumberResult.setText(strNumbers);
73
74              cursor.close();
75              sqlDB.close();
76          }
77      });
78
```


4:38
팀 관리 DB
이름 : 동서팀
인원 : 10
초기화
입력
조회
팀이름

동서대팀
동서팀
인원

40
10


5554:and
9:47
가수 그룹 관리 DB
이름 : b
인원 : 8
초기화
입력
조회
그룹이름
인원
a
4
b
8

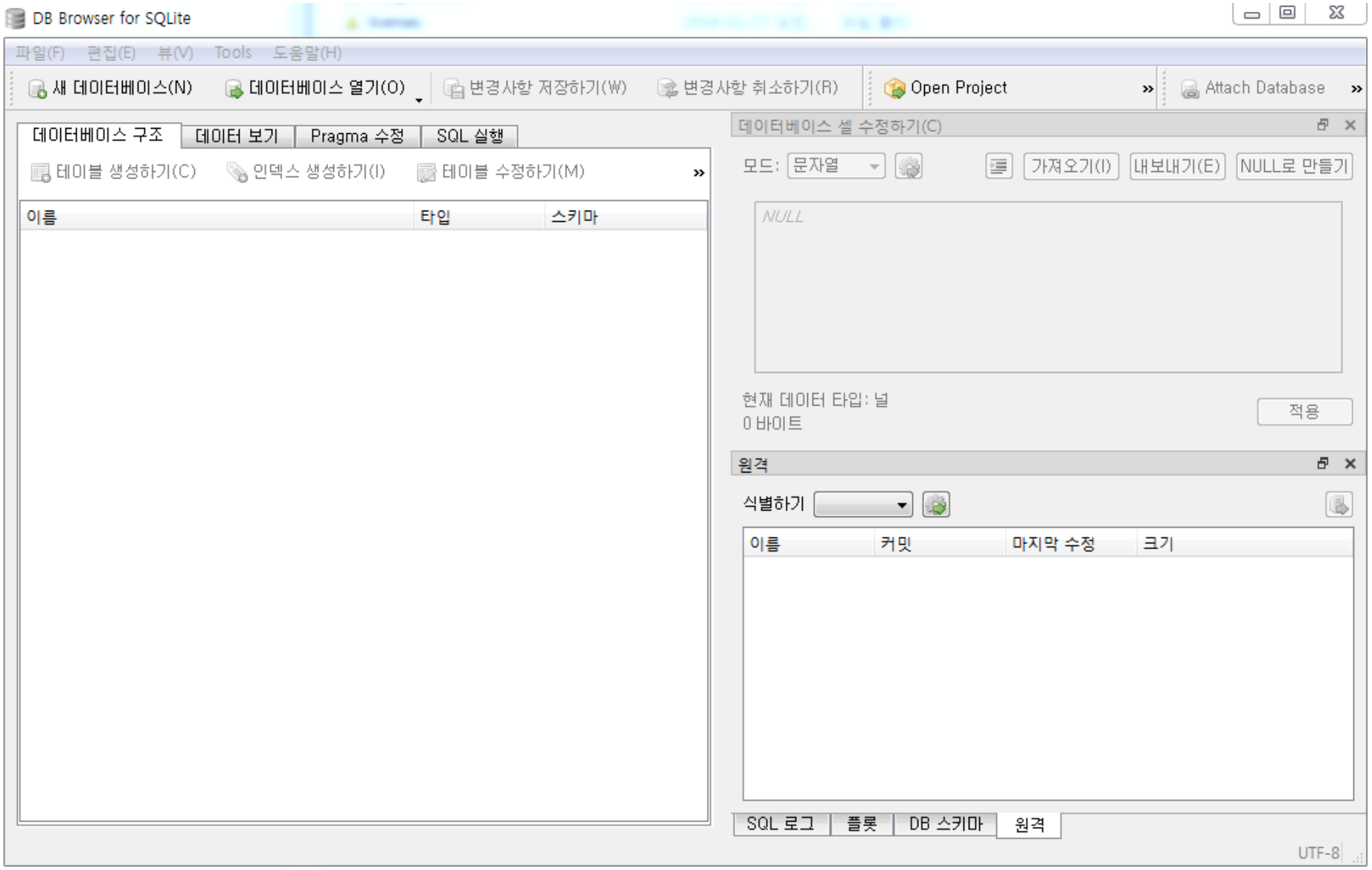
```
C:\Users\pjoon\AppData\Local\Android\Sdk\platform-tools>adb root

C:\Users\pjoon\AppData\Local\Android\Sdk\platform-tools>adb shell
generic_x86:/ # ls
acct        d                 init.rc                odm      sys
adb_keys    data              init.usb.configfs.rc   oem      system
bin         default.prop      init.usb.rc            proc     ueventd.rc
bugreports  dev               init.zygote32.rc       product  vendor
cache       etc               lost+found             sbin
charger     init              metadata               sdcard
config      init.environ.rc   mnt                    storage
generic_x86:/ # cd data/data
generic_x86:/data/data #
```

```
generic_x86:/data/data # cd com.example.pjoon.teamtest
generic_x86:/data/data/com.example.pjoon.teamtest # ls
cache code_cache databases
generic_x86:/data/data/com.example.pjoon.teamtest # cd databases
generic_x86:/data/data/com.example.pjoon.teamtest/databases # ls
groupDB
generic_x86:/data/data/com.example.pjoon.teamtest/databases # sqlite3 groupDB
SQLite version 3.22.0 2018-01-22 18:45:57
Enter ".help" for usage hints.
sqlite> .table
android_metadata  groupTBL
sqlite>
```

```
sqlite> .header on
sqlite> .mode column
sqlite> select * from groupTBL
   ...> ;
gName       gNumber
----------  ----------
동서대팀        40
동서팀         10
sqlite>
```

SQLite – GUI 툴 활용

- 다운로드
  - https://github.com/sqlitebrowser/sqlitebrowser/releases


DB Browser for SQLite
파일(F) 편집(E) 뷰(V) Tools 도움말(H)
새 데이터베이스(N)
데이터베이스 열기(O)
변경사항 저장하기(W)
변경사항 취소하기(R)
Open Project
Attach Database
데이터베이스 구조
데이터 보기
Pragma 수정
SQL 실행
테이블 생성하기(C)
인덱스 생성하기(I)
테이블 수정하기(M)
이름
타입
스키마
데이터베이스 셀 수정하기(C)
모드: 문자열
가져오기(I)
내보내기(E)
NULL로 만들기
NULL
현재 데이터 타입: 널
0 바이트
적용
원격
식별하기
이름
커밋
마지막 수정
크기
SQL 로그
플롯
DB 스키마
원격
UTF-8