

```

public class MyDbOpenHelper extends SQLiteOpenHelper{
    //データベース
    static final String DB_NAME = "sqlite_sample.db";
    static final int DB_VERSION = 3;
    //テーブル名, カラム名
    static final String TABLE_NAME = "mytable";
    static final String COL_NAME = "name";
    static final String COL_AGE = "age";
    static final String COL_DATE = "date";
    static final SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    static final String[] TABLE_COLS = new String[] {"_id", COL_NAME, COL_AGE, COL_DATE };
    //SQL文
    static final String CREATE_TABLE = "CREATE TABLE " + TABLE_NAME
        + " ( _id INTEGER PRIMARY KEY AUTOINCREMENT, "
        + COL_NAME + " TEXT, "
        + COL_AGE + " INTEGER, "
        + COL_DATE + " TEXT ); ";
    static final String DROP_TABLE = "DROP TABLE IF EXISTS " + TABLE_NAME;

    public MyDbOpenHelper(Context c) {
        super(c, DB_NAME, null, DB_VERSION);
    }

    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TABLE); //テーブル作成
        // 必要なら初期値も insert する
        ContentValues cv = new ContentValues();
        cv.put(COL_NAME, "近大太郎");
        cv.put(COL_AGE, 35);
        cv.put( COL_DATE, dateFormat.format(new Date()) );
        db.insert(TABLE_NAME, null, cv);
    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // テーブルを削除
        db.execSQL(DROP_TABLE);
        onCreate(db);
    }
}

```

mytable

_id	name	age	date

```

public class MainActivity extends Activity implements OnClickListener{

    //データベース
    SQLiteDatabase myDB;
    String myDbTable = MyDbOpenHelper.TABLE_NAME;
    String[] tabelCols = MyDbOpenHelper.TABLE_COLS; //テーブルのカラム名
    SimpleDateFormat dateFormat = MyDbOpenHelper.dateFormat; //日付の書式 yyyy-MM-dd HH:mm:ss
    ArrayAdapter<String> adapter; //データベースの検索結果表示用
    //Views
    EditText nameET, ageET;
}

```

```

Button saveBtn;
ListView listView; //データベースの検索結果表示

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Viewの設定, リスナーの登録
    findViews(); setListener();
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1);
    //helper
    MyDbOpenHelper helper = new MyDbOpenHelper(this);
    myDB = helper.getReadableDatabase();
    //テーブルから全データ抽出し, 表示
        getAllDataAndShow();
}

void findViews(){
    setContentView(R.layout.activity_main);
    nameET = (EditText)findViewById(R.id.nameEditText);
    ageET = (EditText)findViewById(R.id.ageEditText);
    saveBtn =(Button)findViewById(R.id.saveButton);
    listView = (ListView)findViewById(R.id.listView);
}

void setListener(){
    saveBtn.setOnClickListener(this);
    //listView -----
    listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        //itemのLongClickでデータ削除
            @Override
            public boolean onItemClick(AdapterView<?> parent, View view, int position, long id) {
                // TODO Auto-generated method stub
                ListView lv = (ListView)parent;
                String itemString = (String)lv.getItemAtPosition(position);//クリックされたitemの文字列
                //例itemString = "32:佐藤太郎(25)2015-01-20 12:15:30" 先頭の数字がデータベースのプライマリーキー_id の値
                int i = itemString.indexOf(":"); String _id= itemString.substring(0, i);
                //Log.d("ItemClicled _id", _id);
                deleteRecord(_id);
                showData();
                return false;
            }
    });
}

public void onClick(View v){
    if(v == saveBtn){
        //Toast.makeText( this, "saveBtn clicked", Toast.LENGTH_LONG ).show();
        String name = nameET.getText().toString();
        int age = Integer.parseInt( ageET.getText().toString() );
        String date = dateFormat.format(new Date());
        insertData( name, age, date );
    }
}

```

```

        showData();
    }
}
//---追加メソッド-----
//テーブルから全データ抽出
public void getAllDataAndShow(){
    Cursor c = myDB.query( myDbTable, tabelCols, null, null, null, null, null);
//後ろ5つの引数 String[] selectionArgs, String groupBy, String having, String orderBy, String limit

    boolean isNext = c.moveToFirst();
    while (isNext) {
        // DBからadapterに文字列を追加
        /* _id          name          age          date      */
        String s = c.getInt(0) + " : " + c.getString(1) +"(" + c.getInt(2) +")"+ c.getString(3);
        adapter.add(s);          isNext = c.moveToNext();
    }
    c.close();          listView.setAdapter(adapter);
}

// adaptorをクリアしてから、データベースの検索結果再表示
public void showData(){
    adapter.clear();
    Cursor c = myDB.query( myDbTable, tabelCols, null, null, null, null, null);
//後ろ5つの引数 String[] selectionArgs, String groupBy, String having, String orderBy, String limit
    boolean isNext = c.moveToFirst();
    while (isNext) {
        /* adapterにDBから文字列を追加 _id          name          age          date      */
        String s = c.getInt(0) + " : " + c.getString(1) +"(" + c.getInt(2) +")"+ c.getString(3);
        adapter.add(s);          isNext = c.moveToNext();
    }
    c.close();          adapter.notifyDataSetChanged();
}

// 1レコード挿入のメソッド
public void insertData( String name, int age, String date ) {
    ContentValues cv = new ContentValues();
    cv.put(MyDbOpenHelper.COL_NAME, name );
    cv.put(MyDbOpenHelper.COL_AGE, age);
    cv.put(MyDbOpenHelper.COL_DATE, date);
    myDB.insert( myDbTable, null, cv );
}

// 1レコード削除のメソッド
public void deleteRecord(String id){
    String s = "_id=" + id;
    myDB.delete(myDbTable, s, null);
}
}
-----

```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="match_parent">
    <EditText
      android:id="@+id/editText1"
      android:layout_width="60dp"
      android:layout_height="wrap_content"
      android:layout_weight="0.2"
      android:ems="10"
      android:text="氏名" />
    <EditText
      android:id="@+id/nameEditText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:inputType="text" >
      <requestFocus/>
    </EditText>
    <EditText
      android:id="@+id/editText2"
      android:layout_width="60dp"
      android:layout_height="wrap_content"
      android:layout_weight="0.2"
      android:ems="10"
      android:text="年齢" />
    <EditText
      android:id="@+id/ageEditText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_weight="1"
      android:inputType="text" >
    </EditText>
    <Button
      android:text="Save"
      android:id="@+id/saveButton"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"/>
  </LinearLayout>
  <ListView
    android:layout_height="0dp"
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_weight="1"/>
</LinearLayout>
```