МГТУ
им. Н. Э. Баумана
ТЕХНОПАРК
Mail.Ru Group

SQLite

Лекция №7

Дроздов Артём

SQLiteDatabase

Основные методы

1. insert
2. delete
3. update
4. query
5. execSQL
6. rawQuery
7. beginTransaction/endTransaction

SQLiteDatabase

Insert

```
ContentValues values = new ContentValues();
values.put(COLUMN_NAME_TITLE, title);
values.put(COLUMN_NAME_SUBTITLE, subtitle);

long newRowId = db.insert(TABLE_NAME, null, values);
```

SQLiteDatabase

Delete

```
String selection = COLUMN_NAME_TITLE + " LIKE ?";
String[] selectionArgs = { "MyTitle" };

db.delete(TABLE_NAME, selection, selectionArgs);
```

SQLiteDatabase

Update

```
ContentValues values = new ContentValues();
values.put(COLUMN_NAME_TITLE, title);

String selection = COLUMN_NAME_TITLE + " LIKE ?";
String[] selectionArgs = { "MyTitle" };

int count = db.update(
      TABLE_NAME,
      values,
      selection,
      selectionArgs);
```

SQLiteDatabase

Query

```
String[] projection = {
        COLUMN_NAME_TITLE,
        COLUMN_NAME_SUBTITLE
};

String selection = COLUMN_NAME_TITLE + " = ?";
String[] selectionArgs = { "My Title" };

String sortOrder = COLUMN_NAME_SUBTITLE + " DESC";

Cursor c = db.query(TABLE_NAME, projection, selection,
selectionArgs, null /*groupBy*/, null /*having*/, sortOrder);
```

SQLiteDatabase

execSQL

```
String sql = "INSERT INTO " + TABLE_NAME +
             " (" + COLUMN_NAME_TITLE + "," +
             COLUMN_NAME_SUBTITLE + ")" +
              " VALUES (?, ?)";
Object[] params = new Object[] {title, subtitle};

db.execSQL(sql, params);
```

SQLiteDatabase

rawQuery

```
String query = "SELECT " +
    COLUMN_NAME_TITLE + "," + COLUMN_NAME_SUBTITLE +
    " FROM " + TABLE_NAME + " WHERE " +
    COLUMN_NAME_TITLE + "=?";

String[] params = new String[] { title };

db.rawQuery(query, params);
```

SQLiteDatabase

Transactions

```
db.beginTransaction();
try {
    //select, insert, update, delete...
    db.setTransactionSuccessful();
} finally {
    db.endTransaction();
}
```

SQLiteOpenHelper

Helper class

```
public class DbHelper extends SQLiteOpenHelper {
  int DATABASE_VERSION = 1;
  String DATABASE_NAME = "DatabaseName.db";
  public DbHelper(Context context) {
     super(context, DATABASE_NAME, null, DATABASE_VERSION);
  }
  public void onCreate(SQLiteDatabase db) {
  }
  public void onUpgrade(SQLiteDatabase db, int oldVer, int newVer) {
  }
  public void onDowngrade(SQLiteDatabase db, int oldVer, int newVer) {
  }
}
```

SQLiteOpenHelper

Obtaining database

```
DbHelper dbHelper = new DbHelper();
//...
SQLiteDatabase db = dbHelper.getWritableDatabase();
```

Рекомендации

1. Используйте синглтон для хранения и доступа к SQLiteDatabase
2. Работайте в одном потоке (разумеется, не UI)
3. Выносите все запросы в отдельный класс
4. Используйте транзакции

ORM

OrmLite

Entity

```
@DatabaseTable(tableName = "user")
public class User {
  @DatabaseField(id = true) String username;
  @DatabaseField String password;
  public User() {
  }
  public User(String username, String password) {
     this.username = username;
     this.password = password;
  }
}
```

OrmLite

Dao

```
Dao<User, Integer> dao = helper.getUserDao();
List<User> list = dao.queryForAll();


User user = new User();
dao.create(user);


dao.delete(user);
```

Список ORM

1. OrmLite
2. ActiveAndroid
3. GreenDao
4. DBFlow
5. Ollie

МГТУ
им. Н. Э. Баумана
ТЕХНОПАРК
Mail.Ru Group

Конец!
Парам-парам-пам!

Спасибо за внимание!