

**Ανοικτά Ακαδημαϊκά Μαθήματα**

**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας**

**Βάσεις Δεδομένων ΙΙ (Ε)**

**Ενότητα 12:** Εφαρμογή Android με χρήση Sqlite

Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης

Τμήμα Μηχανικών Πληροφορικής ΤΕ

|  |  |
| --- | --- |
| Το περιεχόμενο του μαθήματος διατίθεται με άδεια Creative Commons εκτός και αν αναφέρεται διαφορετικά | Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους. |

Περιεχόμενα

[1. Εφαρμογή Android με χρήση Sqlite 3](#_Toc401563938)

[Δημιουργία AVD 8](#_Toc401563939)

[Παράρτημα 10](#_Toc401563940)

Στόχος του εργαστηρίου είναι η εξοικείωση με τη δημιουργία εφαρμογών σε android και τη διαχείριση βάσεων δεδομένων σε sqlite. Μετά την επεξεργασία του εργαστηρίου ο ενδιαφερόμενος θα έχει κατανοήσει τα θέματα δημιουργίας μιας εφαρμογής σε android και σε sqlite σε περιβάλλον eclipse.

**Λέξεις κλειδιά**: Android, Sqlite, Eclipse.

# Εφαρμογή Android με χρήση Sqlite

Στις εφαρμογές Android χρησιμοποιείται SQLite database system, το οποίο είναι open-source, και το οποίο χρησιμοποιείται από μεγάλο πλήθος εφαρμογών όπως Mozilla Firefox, iPhone.

Προκειμένου να δουλέψουμε με android εφαρμογές θα χρησιμοποιήσουμε το Eclipse. Επίσης θα χρειαστούμε και το android-sdk-windows. ( <http://developer.android.com/sdk/eclipse-adt.html#installing> )

Δημιουργία Android Εφαρμογής, Android Development Tools (ADT)

1. **File** > **New** > **Project**.
2. **Android** > **Android Project**, and click **Next**.



1. **Finish**.

Εν συνεχεία ανοίγουμε το project που δημιουργήσαμε το οποίο περιέχει



src/

όλα τα Java αρχεία της εφαρμογής μας τα οποία εκτελούν κάποιες διαδικασίες( stub Activity) .

res/

Σε αυτό τον φάκελο περιέχονται τα resources της εφαρμογής όπως,drawable files, layout files,string values, κ.α.

Όποτε αρχικά στον φάκελο src>com.example υπάρχει το αρχείο Database.java

Παρατηρούμε ότι η κλάση κληρονομεί από την  [Activity](http://developer.android.com/reference/android/app/Activity.html) . Η Activity χρησιμοποιείται προκειμένου να εκτελεστούν ενέργειες. Η μέθοδος [onCreate()](http://developer.android.com/reference/android/app/Activity.html%22%20%5Cl%20%22onCreate%28android.os.Bundle%29)  καλείται όταν ξεκινήσει να εκτελείται η εφαρμογή, οπότε σε αυτό το σημείο εκτελούμε τις αρχικοποιήσεις και τις δηλώσεις των μεταβλητών που θα χρησιμοποιήσουμε.

Για τις ανάγκες της εφαρμογής θα έχουμε 2 αρχεία.

1. Το Database.java το οποίο θα καλούμε κάθε φορά για να τρέξουμε την εφαρμογή
2. Το HelpDb.java το οποίο θα περιέχει όλο τον αναγκαίο κώδικα προκειμένου να αλληλεπιδράσουμε με την βάση και τα δεδομένα της

Αρχικά θα δημιουργήσουμε την βάση «institute» και έναν πίνακα «students».

Students

|  |  |
| --- | --- |
| Id | Integer |
| Name | Text |
| Surname | Text |
| Email | Text |

Δημιουργούμε τα αντίστοιχα String

private static final String DATABASE\_NAME = "books";

 private static final String DATABASE\_TABLE = "titles";

 private static final int DATABASE\_VERSION = 1;

 private static final String DATABASE\_CREATE =

 "create table titles (\_id integer primary key autoincrement, "

 + "isbn text not null, title text not null, "

 + "publisher text not null);";

Εν συνεχεία δημιουργούμε τη μέθοδο DatabaseHelper η οποία κάνει extends την SQLiteOpenHelper και έχει την δυνατότητα να διαχειριστεί database creation and version management

 **private** **static** **class** DatabaseHelper **extends** SQLiteOpenHelper

 {

 DatabaseHelper(Context context)

 {

 **super**(context, *DATABASE\_NAME*, **null**, *DATABASE\_VERSION*);

 }

 @Override

 **public** **void** onCreate(SQLiteDatabase db)

 {

 db.execSQL(*DATABASE\_CREATE*);

 }

 @Override

 **public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion,

 **int** newVersion)

 {

 Log.*w*("HelpDb","Upgrading database from version"+ oldVersion

 + "to" + newVersion + ", which will destroy all old data");

 db.execSQL("DROP TABLE IF EXISTS "+*DATABASE\_TABLE*);

 onCreate(db);

 }

 }

Τέλος ακολουθούν οι μέθοδοι διαχείρισης των δεδομένων.

 //---opens the database---

 **public** HelpDb open() **throws** SQLException

 {

 db = DBHelper.getWritableDatabase();

 **return** **this**;

 }

 //---closes the database---

 **public** **void** close()

 {

 DBHelper.close();

 }

 //---insert a record into the database---

 **public** **long** insertTitle(String name, String surname, String

 email)

 {

 ContentValues initialValues = **new** ContentValues();

 initialValues.put("name", name);

 initialValues.put("surname", surname);

 initialValues.put("email", email);

 **return** db.insert(*DATABASE\_TABLE*, **null**, initialValues);

 }

 //---deletes a particular record ---

 **public** **boolean** deleteTitle(**long** rowId)

 {

 **return** db.delete(*DATABASE\_TABLE*, "id" +

 "=" + rowId, **null**) > 0;

 }

 //---retrieves all the records---

 **public** Cursor getAllTitles()

 {

 **return** db.query(*DATABASE\_TABLE*, **new** String[] {

 "id",

 "name",

 "surname",

 "email"},

 **null**,

 **null**,

 **null**,

 **null**,

 **null**);

 }

 //---retrieves a particular record ---

 **public** Cursor getTitle(**long** rowId) **throws** SQLException

 {

 Cursor mCursor =

 db.query(**true**, *DATABASE\_TABLE*, **new** String[] {

 "id",

 "name",

 "surname",

 "email"},

 "id" + "=" + rowId,

 **null**,

 **null**,

 **null**,

 **null**,

 **null**);

 **if** (mCursor != **null**) {

 mCursor.moveToFirst();

 }

 **return** mCursor;

 }

 //---updates a record ---

 **public** **boolean** updateTitle(**long** rowId, String name,

 String surname, String email)

 {

 ContentValues args = **new** ContentValues();

 args.put("name", name);

 args.put("surname", surname);

 args.put("email", email);

 **return** db.update(*DATABASE\_TABLE*, args,

 "id" + "=" + rowId, **null**) > 0;

 }

Αφού ολοκληρώσαμε όλες τις μεθόδους διαχείρισης της βάσης, θα συμπληρώσουμε και το Database.java

Όπως προείπαμε η μέθοδος η οποία είναι υπεύθυνη για την εμφάνιση των δεδομένων είναι η onCreate οπότε μέσα στην μέθοδο αυτή θα συμπληρώσουμε τις λειτουργίες που θέλουμε να εκτελεστούν.

HelpDb db = **new** HelpDb(**this**);

 //---add 2 titles---

 db.open();

 db.insertTitle(

 "CHRISTOS",

 "SKOURLAS",

 "CSKOURLAS@TEIATH.GR");

 db.insertTitle(

 "ANASTASIOS",

 "TSOLAKIDIS",

 "ATSOLAKID@TEIATH.GR");

 db.close();

 db.open();

 Cursor c = db.getTitle(2);

 **if** (c.moveToFirst())

 DisplayTitle(c);

 **else**

 Toast.*makeText*(**this**, "No title found",

 Toast.*LENGTH\_LONG*).show();

 db.close();

Και η DisplayTitle είναι εκείνη η μέθοδος η οποία καλείται προκειμένου να εμφανιστούν τα αποτελέσματα.

**public** **void** DisplayTitle(Cursor c)

 {

// A toast is a view containing a quick little message for the user.

 Toast.*makeText*(**this**,

 "id: " + c.getString(0) + "\n" +

 "NAME: " + c.getString(1) + "\n" +

 "SURNAME: " + c.getString(2) + "\n" +

 "EMAIL: " + c.getString(3),

 Toast.*LENGTH\_LONG*).show();

 }

Προκείμενο να «τρέψουμε» τις εφαρμογές θα χρησιμοποιήσουμε το simulator του android Android Virtual Device (AVD)

Δημιουργία AVD

Δημιουργία AVD στο Eclipse:

1. **Window > Android SDK and AVD Manager**,
2. Στο *Virtual Devices* panel, επιλέγουμε  **New** για να δημιουργήσουμε το καινούριο AVD.
3. Συμπληρώνουμε τα στοιχεία που μας ζητούνται



1. **Create AVD**.
2. **Εν συνεχεία επιλέγουμε το AVD που έχουμε δημιουργήσει και πατάμε start και μετά launch**

**Αφού έχουμε ολοκληρώσει την εφαρμογή και έχουμε δημιουργήσει το** AVD εν συνεχεία

1. Δεξί κλίκ στο project
2. Run as>Android Application

Και εμφανίζονται τα αποτελέσματα





Πατώντας DDMS πάνω και δεξιά στο παράθυρο(eclipse)



Μας εμφανίζονται όλες οι πληροφορίες σχετικά με τον emulator.

Ανοίγοντας τους φακέλους /data/data/YOUR\_PACKAGE/databases/ βλέπουμε την βάση που δημιουργήσαμε.

Εάν θέλουμε να δουλέψουμε με την βάση που μόλις δημιουργήσαμε τότε.

1. Τρέχουμε το command line(cmd) θέτοντας ως τρέχων φάκελο τον φάκελο tools του android-sdk.
2. Γράφουμε adb shell
3. Πηγαίνουμε στο φάκελο του βρίσκετε η βάση μας π.χ για το παράδειγμα που μόλις φτιάξαμε

C:\Program Files\android-sdk-windows\tools>adb shell

# cd data/data

# cd com.example

# ls

# cd databases

# sqlite3 institute

sqlite3 institute

SQLite version 3.6.22

Enter ".help" for instructions

Enter SQL statements terminated with a ";"

sqlite> .tables

android\_metadata students

sqlite> select \* from students;

1|CHRISTOS|SKOURLAS|CSKOURLAS@TEIATH.GR

2|ANASTASIOS|TSOLAKIDIS|ATSOLAKID@TEIATH.GR

3|CHRISTOS|SKOURLAS|CSKOURLAS@TEIATH.GR

4|ANASTASIOS|TSOLAKIDIS|ATSOLAKID@TEIATH.GR

sqlite>.quit

# exit

Παράρτημα

**Database.java**

**package** com.example;

**import** android.app.Activity;

**import** android.database.Cursor;

**import** android.os.Bundle;

**import** android.widget.Toast;

**public** **class** Database **extends** Activity {

 /\*\* Called when the activity is first created. \*/

 @Override

 **public** **void** onCreate(Bundle savedInstanceState)

 {

 **super**.onCreate(savedInstanceState);

 setContentView(R.layout.*main*);

 HelpDb db = **new** HelpDb(**this**);

 //---add 2 titles---

 db.open();

 db.insertTitle(

 "CHRISTOS",

 "SKOURLAS",

 "CSKOURLAS@TEIATH.GR");

 db.insertTitle(

 "ANASTASIOS",

 "TSOLAKIDIS",

 "ATSOLAKID@TEIATH.GR");

 db.close();

 db.open();

 Cursor c = db.getTitle(12);

 **if** (c.moveToFirst())

 DisplayTitle(c);

 **else**

 Toast.*makeText*(**this**, "No title found",

 Toast.*LENGTH\_LONG*).show();

 db.close();

 }

 **public** **void** DisplayTitle(Cursor c)

 {

 Toast.*makeText*(**this**,

 "id: " + c.getString(0) + "\n" +

 "NAME: " + c.getString(1) + "\n" +

 "SURNAME: " + c.getString(2) + "\n" +

 "EMAIL: " + c.getString(3),

 Toast.*LENGTH\_LONG*).show();

 }

}

**HelpDb.java**

**package** com.example;

**import** android.content.ContentValues;

**import** android.content.Context;

**import** android.database.Cursor;

**import** android.database.SQLException;

**import** android.database.sqlite.SQLiteDatabase;

**import** android.database.sqlite.SQLiteOpenHelper;

**import** android.util.Log;

**public** **class** HelpDb {

 **private** **static** **final** String *DATABASE\_NAME* = "institute";

 **private** **static** **final** String *DATABASE\_TABLE* = "students";

 **private** **static** **final** **int** *DATABASE\_VERSION* = 1;

 **private** **static** **final** String *DATABASE\_CREATE* =

 "create table students (id integer primary key autoincrement, "

 + "name text not null, surname text not null, "

 + "email text not null);";

 //Interface to global information about an application environment

 **private** **final** Context context;

 **private** DatabaseHelper DBHelper;

 **private** SQLiteDatabase db;

 **public** HelpDb(Context ctx)

 {

 **this**.context = ctx;

 DBHelper = **new** DatabaseHelper(context);

 }

 //H DatabaseHelper κάνει extends SQLiteOpenHelper και έχει την δυνατότητα

 //να διαχειριστεί database creation and version management

 **private** **static** **class** DatabaseHelper **extends** SQLiteOpenHelper

 {

 DatabaseHelper(Context context)

 {

 **super**(context, *DATABASE\_NAME*, **null**, *DATABASE\_VERSION*);

 }

 @Override

 **public** **void** onCreate(SQLiteDatabase db)

 {

 db.execSQL(*DATABASE\_CREATE*);

 }

 @Override

 **public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion,

 **int** newVersion)

 {

 Log.*w*("HelpDb", "Upgrading database from version " + oldVersion

 + " to "

 + newVersion + ", which will destroy all old data");

 db.execSQL("DROP TABLE IF EXISTS "+*DATABASE\_TABLE*);

 onCreate(db);

 }

 }

 //---opens the database---

 **public** HelpDb open() **throws** SQLException

 {

 db = DBHelper.getWritableDatabase();

 **return** **this**;

 }

 //---closes the database---

 **public** **void** close()

 {

 DBHelper.close();

 }

 //---insert a record into the database---

 **public** **long** insertTitle(String name, String surname, String email)

 {

 ContentValues initialValues = **new** ContentValues();

 initialValues.put("name", name);

 initialValues.put("surname", surname);

 initialValues.put("email", email);

 **return** db.insert(*DATABASE\_TABLE*, **null**, initialValues);

 }

 //---deletes a particular record ---

 **public** **boolean** deleteTitle(**long** rowId)

 {

 **return** db.delete(*DATABASE\_TABLE*, "id" +

 "=" + rowId, **null**) > 0;

 }

 //---retrieves all the records---

 **public** Cursor getAllTitles()

 {

 **return** db.query(*DATABASE\_TABLE*, **new** String[] {

 "id",

 "name",

 "surname",

 "email"},

 **null**,

 **null**,

 **null**,

 **null**,

 **null**);

 }

 //---retrieves a particular record ---

 **public** Cursor getTitle(**long** rowId) **throws** SQLException

 {

 Cursor mCursor =

 db.query(**true**, *DATABASE\_TABLE*, **new** String[] {

 "id",

 "name",

 "surname",

 "email"},

 "id" + "=" + rowId,

 **null**,

 **null**,

 **null**,

 **null**,

 **null**);

 **if** (mCursor != **null**) {

 mCursor.moveToFirst();

 }

 **return** mCursor;

 }

 //---updates a record ---

 **public** **boolean** updateTitle(**long** rowId, String name,

 String surname, String email)

 {

 ContentValues args = **new** ContentValues();

 args.put("name", name);

 args.put("surname", surname);

 args.put("email", email);

 **return** db.update(*DATABASE\_TABLE*, args,

 "id" + "=" + rowId, **null**) > 0;

 }

 }

|  |
| --- |
| **Ανοικτά Ακαδημαϊκά Μαθήματα****Τεχνολογικό Εκπαιδευτικό Ίδρυμα Αθήνας** |
| **Τέλος Ενότητας** |
| **Χρηματοδότηση*** Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
* Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Αθήνας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
* Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

 |

**Σημειώματα**

**Σημείωμα Αναφοράς**

Copyright ΤΕΙ Αθήνας, Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης 2014. Χ. Σκουρλάς, Α. Τσολακίδης, Δ. Βάσσης. «Βάσεις Δεδομένων ΙΙ (Ε). Ενότητα 12: Εφαρμογή Android με χρήση Sqlite». Έκδοση: 1.0. Αθήνα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: [ocp.teiath.gr](https://ocp.teiath.gr/).

**Σημείωμα Αδειοδότησης**

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό. Οι όροι χρήσης των έργων τρίτων επεξηγούνται στη διαφάνεια «Επεξήγηση όρων χρήσης έργων τρίτων».

Τα έργα για τα οποία έχει ζητηθεί άδεια αναφέρονται στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] http://creativecommons.org/licenses/by-nc-sa/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

* που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
* που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
* που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

**Επεξήγηση όρων χρήσης έργων τρίτων**

|  |  |
| --- | --- |
| © | Δεν επιτρέπεται η επαναχρησιμοποίηση του έργου, παρά μόνο εάν ζητηθεί εκ νέου άδεια από το δημιουργό. |
| διαθέσιμο με άδεια CC-BY | Επιτρέπεται η επαναχρησιμοποίηση του έργου και η δημιουργία παραγώγων αυτού με απλή αναφορά του δημιουργού. |
| διαθέσιμο με άδεια CC-BY-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού, και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. |
| διαθέσιμο με άδεια CC-BY-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η δημιουργία παραγώγων του έργου. |
| διαθέσιμο με άδεια CC-BY-NC | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-SA | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού και διάθεση του έργου ή του παράγωγου αυτού με την ίδια άδεια. Δεν επιτρέπεται η εμπορική χρήση του έργου. |
| διαθέσιμο με άδεια CC-BY-NC-ND | Επιτρέπεται η επαναχρησιμοποίηση του έργου με αναφορά του δημιουργού. Δεν επιτρέπεται η εμπορική χρήση του έργου και η δημιουργία παραγώγων του. |
| διαθέσιμο με άδεια CC0 Public Domain | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| διαθέσιμο ως κοινό κτήμα | Επιτρέπεται η επαναχρησιμοποίηση του έργου, η δημιουργία παραγώγων αυτού και η εμπορική του χρήση, χωρίς αναφορά του δημιουργού. |
| χωρίς σήμανση | Συνήθως δεν επιτρέπεται η επαναχρησιμοποίηση του έργου. |

**Διατήρηση Σημειωμάτων**

* Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:
* το Σημείωμα Αναφοράς
* το Σημείωμα Αδειοδότησης
* τη δήλωση Διατήρησης Σημειωμάτων
* το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει) μαζί με τους συνοδευόμενους υπερσυνδέσμους.