**בסיסי נתונים – Android Studio**

**עבודה מדורגת בשלבים:**

**דוגמה לעבודה עם בסיס נתונים אשר מתאר ושומר נתונים אודות: משתמש וסיסמה**

1. פתיחת מחלקה חדשה: במקרה שלנו: **User**

יש לסמן את ה- MainActivityולבחור: **File**>**new**>**java** **class**, לעדכן שם המתחיל באות גדולה וללחוץ אישור.

1. במחלקה שלנו – User נעדכן את התכונות הרצויות: במקרה שלנו:

**public** String **Username**;  
**public** String **Password**;

**שימו לב!** מאפיין הגישה לתכונות הינו **public** והאות הראשונה גדולה בשם התכונה.

1. פתיחת מחלקה חדשה: בכל מקרה תיקרא: **DBHelper**. (עקוב אחר ההוראות בסעיף 1).
2. יצירת ירושה ממחלקת עזר לבסיסי נתונים:

**public class** DBHelper **extends** SQLiteOpenHelper{

1. יש ללחוץ על שורת כותרת המחלקה, ולהקיש Alt+Enter

יש ללחוץ על **אישור** בכדי להוסיף את **2 שיטות החובה של המחלקה**. (מתרחש אוטומטית)

1. יש ללחוץ שוב על שורת כותרת המחלקה, ולהקיש Alt+Enter

יש ללחוץ על **אישור** בכדי להוסיף את **בנאי המחלקה**. (מתרחש אוטומטית)

1. עדכון במחלקה **DBHelper** לאחר כותרת המחלקה, שני קבועים במאפיין גישה **private**:

(שם בסיס הנתונים וגירסה – תמיד 1)

**private static final** String ***DATABASE\_NAME*** = **"users.db"**;  
**private static final int *DATABASE\_VERSION*** = 1;

1. עדכון בהמשך המחלקה **DBHelper** את הקבוע: **שם הטבלה**

**public static final** String ***TABLE\_NAME*** = **"all\_users"**;

1. עדכון בהמשך המחלקה **DBHelper** את הקבועים אשר הינם התכונות במחלקה User:

**public static final** String ***USERNAME*** = **"Username"**;  
**public static final** String ***PASSWORD*** = **"Password"**;

1. תיקון הבנאי במחלקה **DBHelper** באופן הבא:

**public** DBHelper(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);  
}

1. כתיבת השיטה onCreate במחלקה DBHelper:

**public void** onCreate(SQLiteDatabase sqLiteDatabase) {  
 String sql\_create = **"CREATE TABLE "** + ***TABLE\_NAME*** + **" ("**;  
 sql\_create += ***USERNAME*** + **" TEXT , "**;  
 sql\_create += ***PASSWORD*** + **" TEXT); "**;  
 sqLiteDatabase.execSQL(sql\_create);  
}

**תיווצר המחרוזת: CREATE TABLE all\_users (Username TEXT , Password TEXT);**

1. במחלקה **MainActivity** נוסיף 2 שורות להגדרת עצמים על סמך בסיס הנתונים:

(מתחת לכותרת המחלקה)

**public class** MainActivity **extends** AppCompatActivity {  
  
 SQLiteDatabase **sqdb**;  
 DBHelper **my\_db**;

1. במחלקה **MainActivity** נוסיף **3 שורות** אשר יוצרות עצמים עבור יצירת בסיס הנתונים:

(שורות אלו מתווספות **לאחר** ה- setContentView).

@Override  
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **my\_db = new DBHelper(this);  
  
 sqdb = my\_db.getWritableDatabase();  
  
 sqdb.close();**

1. **הרצת הפרויקט:** אם מתקבלת שגיאת ריצה יש לחזור ולתקן כתיבה לא נכונה של הסעיפים הקודמים.
2. מזל טוב! נוצר לנו בסיס נתונים תקין. איך נוכל לראות אותו?

יש לפתוח את ה-ADM: **Tools**> **Android**> **Android** **Device** **Monitor**

בלשונית הנקראת File Explorer נבחר: databases **> users.db**< חבילת הפרויקט data > data >



נלחץ על כפתור **pull** – (מצד ימין למעלה) אשר נראה כך:

**נשמור את הקובץ במחשב.**



1. נפתח את התוכנה: sqliteopen browser
2. נלחץ על:  **Open Databases** ובסיס הנתונים שלנו יעלה לצורך הצגה ושינויים. (דרך browse data)
3. נסיים שינויים על בסיס הנתונים ונשמור אותו: **Write Changes**.

חשוב: לא להזין נתונים ברשומה הנקראת: **Filter**, אלא ללחוץ על: **New** **Record**

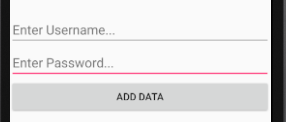


1. נפתח שוב את ה- **Android** **Device** **Monitor** והפעם נלחץ על כפתור **push** –

נבחר את בסיס הנתונים אשר שמרנו קודם לכן, והוא "יחזור" אל האמולטור מלא בנתונים.

(יש להקפיד לא לשנות את שם הקובץ של בסיס הנתונים)

1. נעצב ממשק פשוט של 2 תיבות עריכה (EditText) וכפתור אחד (Button) בכדי להוסיף נתונים:

יש להגדיר את **כל המשתנים באופן רגיל** ב- xml וב- java.

למשל, מתחת לכותרת המחלקה:

EditText etUser;

למטה, ב-onCreate לקשר עם xml:

etUser = (EditText) findViewById(R.id.***etUser***);

1. כעת, נכתוב את השיטה אשר עומדת מאחורי הכפתור ADD DATA: (דרך onClick)

השיטה תוסיף רשומה לטבלה בבסיס הנתונים שלנו.

**(שיטת חובה בפרויקט)**

**public void** addData(View view) {  
  
 String et1 = **etUser**.getText().toString();  
 String et2 = **etPass**.getText().toString();  
  
 ContentValues cv = **new** ContentValues();  
 cv.put(**my\_db**.***USERNAME*** , et1);  
 cv.put(**my\_db**.***PASSWORD*** , et2);  
  
 **sqdb** = **my\_db**.getWritableDatabase();  
 **sqdb**.insert(**my\_db**.***TABLE\_NAME*** , **null** , cv);  
 **sqdb**.close();  
  
 **etUser**.setText(**""**);  
 **etPass**.setText(**""**);  
 Toast.*makeText*(**this**, **"Data added to your db"**, Toast.***LENGTH\_LONG***).show();  
}

1. **שיטות לקריאת נתונים**1:

שיטה בסיסית הינה קריאת הרשומה הראשונה בטבלה באופן ויזואלי שאנחנו רואים מה-sqlite browser.

אמנם זוהי הרשומה הראשונה אך היא נכנסה אחרונה אל הטבלה.

רשומה אשר נכנסת אל הטבלה למעשה דוחפת את שאר הרשומות למטה.

כך, ששימוש בפקודה: c.**moveToLast**();  
יציב את הסמן על השורה הראשונה בטבלה (שורה אשר נכנסה אליה **אחרונה**).

השיטה עובדת ע"י לחיצה על כפתור דרך מאפיין onClick ב- xml:

**(שיטת חובה בפרויקט)**

**public void** readData(View view) {  
  
 **sqdb** = **my\_db**.getWritableDatabase();  
 Cursor c = **sqdb**.query(DBHelper.***TABLE\_NAME*** ,  
 **null** , **null** , **null** , **null** , **null** , **null**);  
  
 **int** col1 = c.getColumnIndex(DBHelper.***USERNAME***);  
 **int** col2 = c.getColumnIndex(DBHelper.***PASSWORD***);  
  
 c.**moveToLast**();  
  
 String s1 = c.getString(col1);  
 String s2 = c.getString(col2);  
 **etUser**.setText(s1);  
 **etPass**.setText(s2);  
  
 c.close();  
 **sqdb**.close();  
}

1. **שיטות לקריאת נתונים**2:

ניתן להזיז את הסמן ולקרוא רשומה מסוימת הקריאה מתחילה **מלמטה** ו**מהמספר 0**.

למשל, הקוד הבא יגיע אל הערך בבסיס הנתונים הבא:

c.moveToPosition(2);

Oren

Avi

Doron

Yossi

**Moti** 2

Yoram 1

Roee 0

1. **שיטות לקריאת נתונים3:**

השיטה בודקת האם משתמש מסוים קיים בבסיס הנתונים שלי או לא?

(את שמו אני מקליד בתיבת הטקסט ולוחץ על כפתור אשר מאחוריו כתובה השיטה דרך onClick)

**public void** readData(View view) {  
  
 **sqdb** = **my\_db**.getWritableDatabase();  
 Cursor c = **sqdb**.query(DBHelper.***TABLE\_NAME*** ,  
 **null** , **null** , **null** , **null** , **null** , **null**);  
  
 **int** col1 = c.getColumnIndex(DBHelper.***USERNAME***);  
 String ed1 = **etUser**.getText().toString();  
 **boolean** check = **false**;  
 c.moveToFirst();  
  
 **while**(!c.isAfterLast())  
 {  
 String s1 = c.getString(col1);  
 **if** (s1.equals(ed1))  
 check = **true**;  
  
 c.moveToNext();  
 }  
 **if** (check == **true**)  
 Toast.*makeText*(**this**, **"User Exist in db!"**, Toast.***LENGTH\_LONG***).show();  
 **else** Toast.*makeText*(**this**, **"User does not Exist in db!"**, Toast.***LENGTH\_LONG***).show();  
  
 c.close();  
 **sqdb**.close();  
}

1. **שיטות לקריאת נתונים4:**

השיטה הבאה חשיבותה גבוהה והיא תציג **רשימה עדכנית של בסיס הנתונים**.

לפני כתיבת השיטה יש לעבור על מערך המחרוזות (זה שישמור את הנתונים – רשומה בכל תא)

ולאתחל אותו – בכל תא נציב: "" בכדי להימנע משגיאות ריצה כאשר נעבירו לרשימה.

:set אחרי ה-

**sArr** = **new** String[25];  
 **for** (**int** i = 0 ; i<**sArr**.**length** ; i++)  
 **sArr**[i] = **""**;

**השיטה עצמה: (כמובן אחרי שהגדרנו רשימה כרגיל)**

**public void** readData(View view) {  
  
 **sqdb** = **my\_db**.getWritableDatabase();  
 Cursor c = **sqdb**.query(DBHelper.***TABLE\_NAME*** ,  
 **null** , **null** , **null** , **null** , **null** , **null**);  
  
 **int** col1 = c.getColumnIndex(DBHelper.***USERNAME***);  
 **int** col2 = c.getColumnIndex(DBHelper.***PASSWORD***);  
 **int** i = 0;  
  
 String ed1 = **etUser**.getText().toString();  
 String ed2 = **etPass**.getText().toString();  
  
 c.moveToFirst();  
  
 **while**(!c.isAfterLast())  
 {  
 String s1 = c.getString(col1);  
 String s2 = c.getString(col2);  
 **sArr**[i] = s1 + **" "** + s2;  
  
 c.moveToNext();  
 i++;  
 }  
 c.close();  
 **adapter** = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1*** , **sArr**);  
  
 **lv1**.setAdapter(**adapter**);  
  
 **sqdb**.close();  
}