**SQLite with Spinner**

**MainActivity.java**

package com.example.sqlitewithspinner;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DatabaseHandler extends SQLiteOpenHelper {  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *DATABASE\_NAME* = "spinnerExample";  
 private static final String *TABLE\_NAME* = "labels";  
 private static final String *COLUMN\_ID* = "id";  
 private static final String *COLUMN\_NAME* = "name";  
  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 // Creating Tables  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 // Category table create query  
 String CREATE\_ITEM\_TABLE = "CREATE TABLE " + *TABLE\_NAME* + "("  
 + *COLUMN\_ID* + " INTEGER PRIMARY KEY," + *COLUMN\_NAME* + " TEXT)";  
 db.execSQL(CREATE\_ITEM\_TABLE);  
 }  
  
 // Upgrading database  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 // Drop older table if existed  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
  
 // Create tables again  
 onCreate(db);  
 }  
  
 */\*\*  
 \* Inserting new lable into lables table  
 \* \*/* public void insertLabel(String label){  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(*COLUMN\_NAME*, label);//column name, column value  
  
 // Inserting Row  
 db.insert(*TABLE\_NAME*, null, values);//tableName, nullColumnHack, CotentValues  
 db.close(); // Closing database connection  
 }  
  
 */\*\*  
 \* Getting all labels  
 \* returns list of labels  
 \* \*/* public List<String> getAllLabels(){  
 List<String> list = new ArrayList<String>();  
  
 // Select All Query  
 String selectQuery = "SELECT \* FROM " + *TABLE\_NAME*;  
  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);//selectQuery,selectedArguments  
  
 // looping through all rows and adding to list  
 if (cursor.moveToFirst()) {  
 do {  
 list.add(cursor.getString(1));//adding 2nd column data  
 } while (cursor.moveToNext());  
 }  
 // closing connection  
 cursor.close();  
 db.close();  
 // returning lables  
 return list;  
 }  
}

**DatabaseHandler.java**

package com.example.sqlitewithspinner;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import java.util.ArrayList;  
import java.util.List;  
  
public class DatabaseHandler extends SQLiteOpenHelper {  
 private static final int *DATABASE\_VERSION* = 1;  
 private static final String *DATABASE\_NAME* = "spinnerExample";  
 private static final String *TABLE\_NAME* = "labels";  
 private static final String *COLUMN\_ID* = "id";  
 private static final String *COLUMN\_NAME* = "name";  
  
 public DatabaseHandler(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
  
 // Creating Tables  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 // Category table create query  
 String CREATE\_ITEM\_TABLE = "CREATE TABLE " + *TABLE\_NAME* + "("  
 + *COLUMN\_ID* + " INTEGER PRIMARY KEY," + *COLUMN\_NAME* + " TEXT)";  
 db.execSQL(CREATE\_ITEM\_TABLE);  
 }  
  
 // Upgrading database  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 // Drop older table if existed  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
  
 // Create tables again  
 onCreate(db);  
 }  
  
 */\*\*  
 \* Inserting new lable into lables table  
 \* \*/* public void insertLabel(String label){  
 SQLiteDatabase db = this.getWritableDatabase();  
  
 ContentValues values = new ContentValues();  
 values.put(*COLUMN\_NAME*, label);//column name, column value  
  
 // Inserting Row  
 db.insert(*TABLE\_NAME*, null, values);//tableName, nullColumnHack, CotentValues  
 db.close(); // Closing database connection  
 }  
  
 */\*\*  
 \* Getting all labels  
 \* returns list of labels  
 \* \*/* public List<String> getAllLabels(){  
 List<String> list = new ArrayList<String>();  
  
 // Select All Query  
 String selectQuery = "SELECT \* FROM " + *TABLE\_NAME*;  
  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);//selectQuery,selectedArguments  
  
 // looping through all rows and adding to list  
 if (cursor.moveToFirst()) {  
 do {  
 list.add(cursor.getString(1));//adding 2nd column data  
 } while (cursor.moveToNext());  
 }  
 // closing connection  
 cursor.close();  
 db.close();  
 // returning lables  
 return list;  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
  
 <EditText  
 android:id="@+id/input\_label"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="46dp"  
 android:hint="Add item"  
 android:ems="10" />  
  
 <Button  
 android:id="@+id/btn\_add"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/input\_label"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="67dp"  
 android:text="Add item" />  
  
 <Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/btn\_add"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_marginTop="70dp" />  
</RelativeLayout>