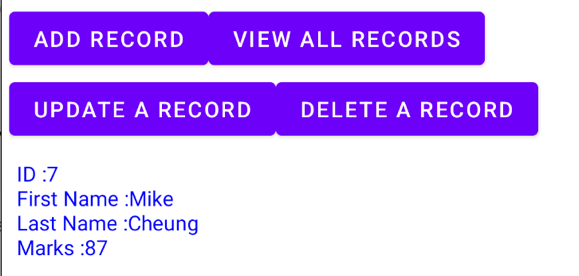
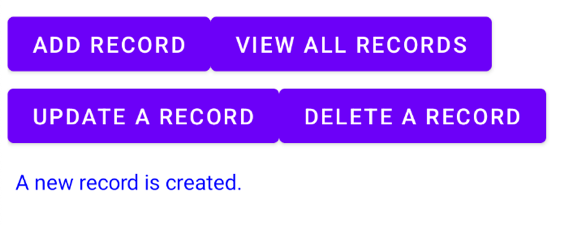
# Lab 5 part 2

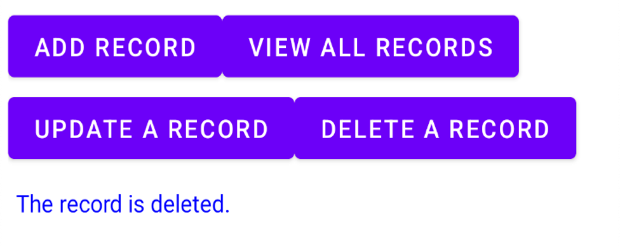
In this part, students will develop a mobile application for local database. The layout included will be shown below:

Here are the screen captures of the application:

一張含有 文字 的圖片

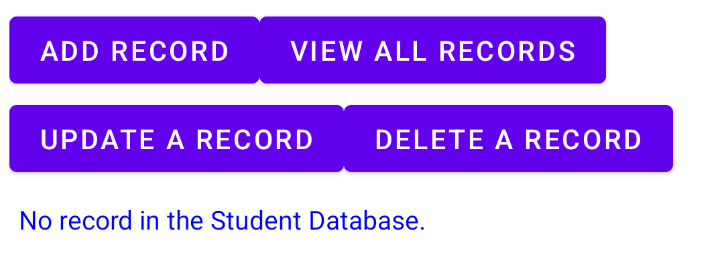
自動產生的描述





一張含有 文字 的圖片

自動產生的描述



In this lab, we will create a database called “Student.db” with a table called “student\_table”, here are the details of the table:

|  |  |
| --- | --- |
| student\_table | |
| **Attribute** | **Data Type** |
| ID | Integer |
| First Name | String |
| Last Name | String |
| Marks | Integer |

Step 1:

In Android Studio, create a new project named Lab 6 with following project setting:

* Choose your project : **Empty Activity**
* Application Name : **Lab 5 part 2**
* Project location : use the default setting
* Language : **Java**
* Minimum API level: **API 21: Android 5.0 (Lollipop)**
* Click **Finish**

Step 2:  
Open activity\_main.xml and insert the following code:

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingLeft="5dp"  
 android:paddingRight="5dp"  
 >  
  
 <LinearLayout android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:text="First Name:" />  
  
 <EditText android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/firstname"  
 />  
 </LinearLayout>

<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:text="Last Name:" />  
 <EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/lastname"  
 />  
</LinearLayout>  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:text="Marks:" />  
 <EditText  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/marks"  
 />  
</LinearLayout>

<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:text="ID:" />  
 <EditText  
 android:layout\_width="fill\_parent"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/ed\_id"  
 />  
</LinearLayout>  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:onClick="addRecord"  
 android:text="Add Record" />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:onClick="viewAllRecords"  
 android:text="View All Records" />  
 </LinearLayout>

<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <Button android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:onClick="updateRecord"  
 android:text="Update a Record" />  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="15sp"  
 android:onClick="deleteRecord"  
 android:text="Delete a Record" />  
 </LinearLayout>  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="fill\_parent">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingLeft="5dp"  
 android:paddingTop="10dp"  
 android:textColor="#0000FF"  
 android:id="@+id/tv\_results"  
 android:text=""  
 />  
 </ScrollView>  
</LinearLayout>

Step 3:

Right click com.example.lab5part2 -> New -> Java class, choose **Java** and type “DatabaseHelper”, then press Enter.

Step 4:

Open DatabaseHelper.java and insert the following code to define the basic structure of the database:

public class DatabaseHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "Student.db";  
 private static final String *TABLE\_NAME* = "student\_table";  
 private static final String *COL\_1* = "ID";  
 private static final String *COL\_2* = "FIRSTNAME";  
 private static final String *COL\_3* = "LASTNAME";  
 private static final String *COL\_4* = "MARKS";  
 private static final int *DATABASED\_VERSION* = 1;

}

Step 5:

To create the SQLite Database, SQLiteOpenHelper is necessary to be extended, DatabaseHelper(), onCreate() and onUpgrade() will appear after the extension. Replace with the following code:

public class DatabaseHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "Student.db";  
 private static final String *TABLE\_NAME* = "student\_table";  
 private static final String *COL\_1* = "ID";  
 private static final String *COL\_2* = "FIRSTNAME";  
 private static final String *COL\_3* = "LASTNAME";  
 private static final String *COL\_4* = "MARKS";  
 private static final int *DATABASED\_VERSION* = 1;  
 public DatabaseHelper(Context context) {  
 *// super(context, name, factory, version)* super(context, *DATABASE\_NAME*, null, *DATABASED\_VERSION*); }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL("create table " + *TABLE\_NAME* +"(ID INTEGER PRIMARY KEY AUTOINCREMENT, " + "FIRSTNAME TEXT, LASTNAME TEXT, MARKS INTEGER)");  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_NAME*);  
 onCreate(db);  
 }

}

Step 6:

ContentValues is mechanism of storage, it used to store a sample values like string, int etc. You need to provide the column name and the value that you want by using **put**. Insert the following code inside DatabaseHelper.java as a new method.

*// Method to insert a record to the database*public boolean insertData(String firstname, String lastname, String marks) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(*COL\_2*,firstname);  
 contentValues.put(*COL\_3*,lastname);  
 contentValues.put(*COL\_4*,marks);  
 long result = db.insert(*TABLE\_NAME*, null, contentValues);  
 if (result == -1)  
 return false;  
 else  
 return true;  
}

Step 7:

To view the data, we need to use Cursor as a “mouse” to get the data we need.

*// Method to show all the records*public Cursor getAllData() {  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor res = db.rawQuery("select \* from " + *TABLE\_NAME*, null);  
 return res;  
}

Step 8:

Insert the following code inside DatabaseHelper.java for updating record.

*// Method to update a record*public boolean updateData(String id, String firstname, String lastname, String marks)  
{  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues contentValues = new ContentValues();  
 contentValues.put(*COL\_1*,id);  
 contentValues.put(*COL\_2*,firstname);  
 contentValues.put(*COL\_3*,lastname);  
 contentValues.put(*COL\_4*,marks);  
 db.update(*TABLE\_NAME*, contentValues, "ID = ?", new String[] {id}); return true;  
}

Step 9:

*// Method to delete a record*public Integer deleteData (String id) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 return db.delete(*TABLE\_NAME*, "ID = ?", new String[] {id});   
}

Now, we finished the coding of DatabaseHelper.java. However, it is just a logic without visible action, we need to move on the MainActivity.java to finish the remaining part.

Step 10:

Create the related attributes in MainActivity.java

EditText firstName, lastName, marks, id;  
TextView results;  
DatabaseHelper studentDb;

Step 11:

Link up attributes with xml in onCreate()

firstName = findViewById(R.id.*firstname*);  
lastName = findViewById(R.id.*lastname*);  
marks = findViewById(R.id.*marks*);  
id = findViewById(R.id.*ed\_id*);  
results = findViewById(R.id.*tv\_results*);  
studentDb = new DatabaseHelper(this);

Step 12:  
Add the method for inserting data

public void addRecord(View view) {  
 boolean isInserted = studentDb.insertData(firstName.getText().toString(), lastName.getText().toString(), marks.getText().toString());  
 if (isInserted)  
 results.setText("A new record is created.");  
 else  
 results.setText("Data cannot be inserted.");  
}

Step 13:

Add the method for reviewing data

public void viewAllRecords(View view) {  
 Cursor res = studentDb.getAllData();  
 if (res.getCount() == 0) {  
 results.setText("No record in the Student Database.");  
 return; }  
 StringBuffer buffer = new StringBuffer(); while (res.moveToNext()) {  
 buffer.append("ID :"+ res.getString(0)+"\n");  
 buffer.append("First Name :"+ res.getString(1)+"\n");  
 buffer.append("Last Name :"+ res.getString(2)+"\n");  
 buffer.append("Marks :"+ res.getString(3)+"\n\n");  
 }  
 results.setText(buffer.toString());   
}

Step 14:

Add the method for updating data

public void updateRecord(View view) {  
 boolean isUpdate = studentDb.updateData(id.getText().toString(), firstName.getText().toString(), lastName.getText().toString(), marks.getText().toString());  
 if (isUpdate)  
 results.setText("The record is updated.");  
 else  
 results.setText("The record cannot be updated.");  
}

Step 15:

Add the method for deleting data

public void deleteRecord(View view) {  
 Integer deletedRows = studentDb.deleteData(id.getText().toString());  
 if (deletedRows > 0)  
 results.setText("The record is deleted.");  
 else results.setText("The record cannot be deleted.");  
}