How to Update Data to SQLite Database in Android?

We have seen How to Create and Add Data to SQLite Database in Android as well as How to Read Data from SQLite Database in Android. We have performed different SQL queries for reading and writing our data to SQLite database. In this article, we will take a look at **updating data to SQLite database in Android**.

What we are going to build in this article?

We will be building a simple application in which we were already adding as well as reading the data. Now we will simply update our data in a new activity and we can get to see the updated data. Note that we are going to implement this project using the **Java** language.

Step by Step Implementation

Step 1: **Creating a new activity for updating our course**

As we want to update our course, so for this process we will be creating a new activity where we will be able to update our courses in the SQLite database. To create a new Activity we have to navigate to the **app > java > your app's package name > Right click on package name > New > Empty Activity** and name your activity as **UpdateCourseActivity** and create new Activity. Make sure to select the **empty activity**.

Step 2: Working with the activity_update_course.xml file

Navigate to the **app > res > Layout > activity_update_course.xml** and add the below code to it.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".UpdateCourseActivity">

    <!--Edit text to enter course name-->
    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter course Name" />

    <!--edit text to enter course duration-->
    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Duration" />

    <!--edit text to display course tracks-->
    <EditText
        android:id="@+id/idEdtCourseTracks"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Tracks" />

    <!--edit text for course description-->
    <EditText
        android:id="@+id/idEdtCourseDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Description" />

    <!--button for updating our course-->
    <Button
        android:id="@+id/idBtnUpdateCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Update Course"
        android:textAllCaps="false" />

</LinearLayout>
```

Step 3: Updating our DBHandler class

Navigate to the **app > java > your app's package name > DBHandler** and add the below code to it. In this, we simply have to create a new method to update our course.

```
// below is the method for updating our courses
public void updateCourse(String originalCourseName, String courseName, String
courseDescription,
                             String courseTracks, String courseDuration) {

        // calling a method to get writable database.
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();

        // on below line we are passing all values
        // along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);

        // on below line we are calling a update method to update our database and
passing our values.
        // and we are comparing it with name of our course which is stored in original
name variable.
        db.update(TABLE_NAME, values, "name=?", new String[]{originalCourseName});
        db.close();
}
```

Below is the updated code for the **DBHandler.java** file after adding the above code snippet.

```
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;

public class DBHandler extends SQLiteOpenHelper {

    // creating a constant variables for our database.
    // below variable is for our database name.
    private static final String DB_NAME = "coursedb";

    // below int is our database version
    private static final int DB_VERSION = 1;

    // below variable is for our table name.
    private static final String TABLE_NAME = "mycourses";

    // below variable is for our id column.
    private static final String ID_COL = "id";

    // below variable is for our course name column
    private static final String NAME_COL = "name";

    // below variable id for our course duration column.
    private static final String DURATION_COL = "duration";

    // below variable for our course description column.
    private static final String DESCRIPTION_COL = "description";

    // below variable is for our course tracks column.
    private static final String TRACKS_COL = "tracks";

    // creating a constructor for our database handler.
    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    // below method is for creating a database by running a sqlite query
    @Override
    public void onCreate(SQLiteDatabase db) {
        // on below line we are creating
        // an sqlite query and we are
        // setting our column names
        // along with their data types.
        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + NAME_COL + " TEXT,"
                + DURATION_COL + " TEXT,"
                + DESCRIPTION_COL + " TEXT,"
                + TRACKS_COL + " TEXT)";

        // at last we are calling a exec sql
        // method to execute above sql query
        db.execSQL(query);
    }

    // this method is use to add new course to our sqlite database.
    public void addNewCourse(String courseName, String courseDuration, String
courseDescription, String courseTracks) {

        // on below line we are creating a variable for
        // our sqlite database and calling writable method
        // as we are writing data in our database.
        SQLiteDatabase db = this.getWritableDatabase();

        // on below line we are creating a
        // variable for content values.
        ContentValues values = new ContentValues();

        // on below line we are passing all values
        // along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);

        // after adding all values we are passing
        // content values to our table.
        db.insert(TABLE_NAME, null, values);

        // at last we are closing our
        // database after adding database.
        db.close();
    }

    // we have created a new method for reading all the courses.
    public ArrayList<CourseModal> readCourses() {
        // on below line we are creating a
        // database for reading our database.
        SQLiteDatabase db = this.getReadableDatabase();

        // on below line we are creating a cursor with query to read data from database.
        Cursor cursorCourses = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);

        // on below line we are creating a new array list.
        ArrayList<CourseModal> courseModalArrayList = new ArrayList<>();

        // moving our cursor to first position.
        if (cursorCourses.moveToFirst()) {
            do {
                // on below line we are adding the data from cursor to our array list.
                courseModalArrayList.add(new CourseModal(cursorCourses.getString(1),
                        cursorCourses.getString(4),
                        cursorCourses.getString(2),
                        cursorCourses.getString(3)));
            } while (cursorCourses.moveToNext());
            // moving our cursor to next.
        }
        // at last closing our cursor
        // and returning our array list.
        cursorCourses.close();
        return courseModalArrayList;
    }

    // below is the method for updating our courses
    public void updateCourse(String originalCourseName, String courseName, String
courseDescription,
                             String courseTracks, String courseDuration) {

        // calling a method to get writable database.
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        // on below line we are passing all values
        // along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);

        // on below line we are calling a update method to update our database and 
passing our values.
        // and we are comparing it with name of our course which is stored in original 
name variable.
        db.update(TABLE_NAME, values, "name=?", new String[]{originalCourseName});
        db.close();
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // this method is called to check if the table exists already.
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

Step 4: Updating our CourseRVAdapter.java class

As we will be opening a new activity to update our course. We have to add on click listener for the items of our **RecycleView**. For adding **onClickListener()** to our recycler view items navigate to the **app > java > your app's package name > CourseRVAdapter** class and simply add an **onClickListener()** for our RecyclerView item. Add the below code to your adapter class.

```
// below line is to add on click listener for our recycler view item.
holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // on below line we are calling an intent.
                Intent i = new Intent(context, UpdateCourseActivity.class);

                // below we are passing all our values.
                i.putExtra("name", modal.getCourseName());
                i.putExtra("description", modal.getCourseDescription());
                i.putExtra("duration", modal.getCourseDuration());
                i.putExtra("tracks", modal.getCourseTracks());

                // starting our activity.
                context.startActivity(i);
        }
});
```

Below is the updated code for the **CourseRVAdapter.java** file after adding the above code snippet.

```
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;

public class CourseRVAdapter extends RecyclerView.Adapter<CourseRVAdapter.ViewHolder> {

    // variable for our array list and context
    private ArrayList<CourseModal> courseModalArrayList;
    private Context context;

    // constructor
    public CourseRVAdapter(ArrayList<CourseModal> courseModalArrayList, Context context)
{
        this.courseModalArrayList = courseModalArrayList;
        this.context = context;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        // on below line we are inflating our layout
        // file for our recycler view items.
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.course_rv_item, parent, false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        // on below line we are setting data
        // to our views of recycler view item.
        CourseModal modal = courseModalArrayList.get(position);
        holder.courseNameTV.setText(modal.getCourseName());
        holder.courseDescTV.setText(modal.getCourseDescription());
        holder.courseDurationTV.setText(modal.getCourseDuration());
        holder.courseTracksTV.setText(modal.getCourseTracks());

        // below line is to add on click listener for our recycler view item.
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // on below line we are calling an intent.
                Intent i = new Intent(context, UpdateCourseActivity.class);

                // below we are passing all our values.
                i.putExtra("name", modal.getCourseName());
                i.putExtra("description", modal.getCourseDescription());
                i.putExtra("duration", modal.getCourseDuration());
                i.putExtra("tracks", modal.getCourseTracks());

                // starting our activity.
                context.startActivity(i);
            }
        });
    }

    @Override
    public int getItemCount() {
        // returning the size of our array list
        return courseModalArrayList.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {

        // creating variables for our text views.
        private TextView courseNameTV, courseDescTV, courseDurationTV, courseTracksTV;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            // initializing our text views
            courseNameTV = itemView.findViewById(R.id.idTVCourseName);
            courseDescTV = itemView.findViewById(R.id.idTVCourseDescription);
            courseDurationTV = itemView.findViewById(R.id.idTVCourseDuration);
            courseTracksTV = itemView.findViewById(R.id.idTVCourseTracks);
        }
    }
}
```

Step 5: Working with the UpdateCourseActivity.java file

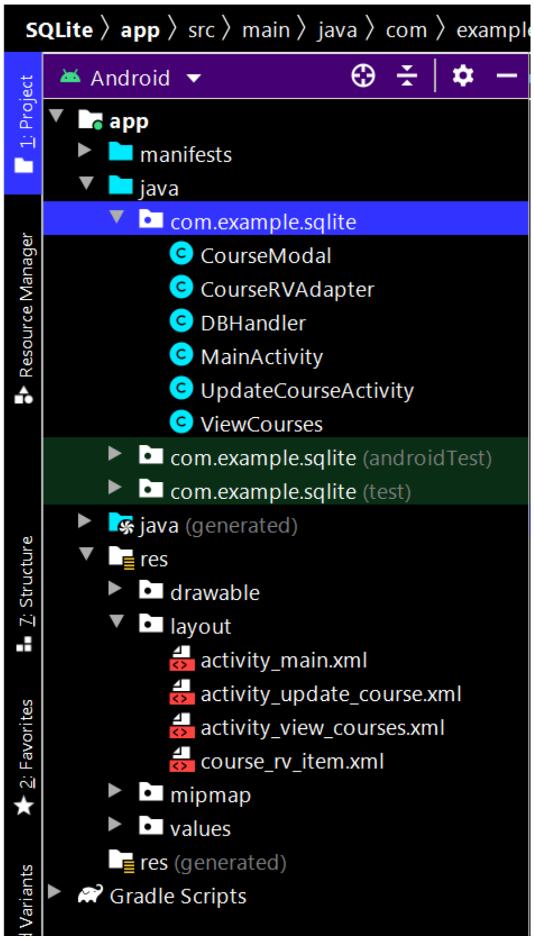
Navigate to the **app > java > your app's package name > UpdateCourseActivity.java** file and add the below code to it. Comments are added inside the code to understand the code in more detail.

```
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class UpdateCourseActivity extends AppCompatActivity {

    // variables for our edit text, button, strings and dbhandler class.
    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt,
courseDescriptionEdt;
    private Button updateCourseBtn;
    private DBHandler dbHandler;
    String courseName, courseDesc, courseDuration, courseTracks;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_course);

        // initializing all our variables.
        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        updateCourseBtn = findViewById(R.id.idBtnUpdateCourse);

        // on below line we are initialing our dbhandler class.
        dbHandler = new DBHandler(UpdateCourseActivity.this);

        // on below lines we are getting data which
        // we passed in our adapter class.
        courseName = getIntent().getStringExtra("name");
        courseDesc = getIntent().getStringExtra("description");
        courseDuration = getIntent().getStringExtra("duration");
        courseTracks = getIntent().getStringExtra("tracks");

        // setting data to edit text
        // of our update activity.
        courseNameEdt.setText(courseName);
        courseDescriptionEdt.setText(courseDesc);
        courseTracksEdt.setText(courseTracks);
        courseDurationEdt.setText(courseDuration);

        // adding on click listener to our update course button.
        updateCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // inside this method we are calling an update course
                // method and passing all our edit text values.
                dbHandler.updateCourse(courseName, courseNameEdt.getText().toString(),
courseDescriptionEdt.getText().toString(), courseTracksEdt.getText().toString(),
courseDurationEdt.getText().toString());

                // displaying a toast message that our course has been updated.
                Toast.makeText(UpdateCourseActivity.this, "Course Updated..",
Toast.LENGTH_SHORT).show();

                // launching our main activity.
                Intent i = new Intent(UpdateCourseActivity.this, MainActivity.class);
                startActivity(i);
            }
        });
    }
}
```

Now run your app and see the output of the app. Make sure to add data to the SQLite database before updating it.

Output:

Below is the complete project file structure after performing the update operation:



```
▼ app
    ▶ manifests
    ▼ java
        ▼ com.example.sqlite
              CourseModal
              CourseRVAdapter
              DBHandler
              MainActivity
              UpdateCourseActivity
              ViewCourses
        ▶ com.example.sqlite (androidTest)
        ▶ com.example.sqlite (test)
    ▶ java (generated)
    ▼ res
        ▶ drawable
        ▼ layout
              activity_main.xml
              activity_update_course.xml
              activity_view_courses.xml
              course_rv_item.xml
        ▶ mipmap
        ▶ values
      res (generated)
▶ Gradle Scripts
```