Experiment 11
Develop an android application to store data in the SQLite database.

Activity_main.xml

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.androi
    d.com/apk/res/android"
3.      xmlns:app="http://schemas.android.com/apk/res-auto"
4.      xmlns:tools="http://schemas.android.com/tools"
5.      android:layout_width="match_parent"
6.      android:layout_height="match_parent"
7.      android:background="#FFFFFF"
8.      android:padding="10dp"
9.      tools:context=".MainActivity">
10.
11.     <TextView
12.         android:id="@+id/appTitle"
13.         android:layout_width="0dp"
14.         android:layout_height="wrap_content"
15.         android:fontFamily="@font/roboto"
16.         android:text="@string/appTitle_txt"
17.         android:textAlignment="center"
18.         android:textSize="24sp"
19.         app:layout_constraintLeft_toLeftOf="parent"
20.         app:layout_constraintRight_toRightOf="parent"
21.         app:layout_constraintTop_toTopOf="parent" />
22.
23.     <TextView
24.         android:id="@+id/lb_activity_name"
25.         style="@style/heading"
26.         android:layout_width="80dp"
27.         android:layout_height="wrap_content"
28.         android:text="@string/lb_act_name_hint"
29.         app:layout_constraintBaseline_toBaselineOf="@+id/et_activity_name"
30.         app:layout_constraintEnd_toStartOf="@+id/et_activity_name"
31.         app:layout_constraintHorizontal_bias="0.5"
32.         app:layout_constraintHorizontal_chainStyle="packed"
33.         app:layout_constraintStart_toStartOf="parent" />
34.
35.     <TextView
36.         android:id="@+id/lb_activity_txt"
37.         style="@style/heading"
38.         android:layout_width="80dp"
39.         android:layout_height="wrap_content"
40.         android:text="@string/lb_actv_txt_hint"
41.         app:layout_constraintBaseline_toBaselineOf="@+id/et_activity_desp"
42.         app:layout_constraintEnd_toStartOf="@+id/et_activity_desp"
43.         app:layout_constraintHorizontal_bias="0.5"
44.         app:layout_constraintHorizontal_chainStyle="packed"
45.         app:layout_constraintStart_toStartOf="parent" />
46.
47.     <TextView
48.         android:id="@+id/lb_activity_deadline"
49.         style="@style/heading"
50.         android:layout_width="80dp"
51.         android:layout_height="wrap_content"
52.         android:text="@string/lb_act_time_text"
53.         app:layout_constraintBaseline_toBaselineOf="@+id/et_set_deadline"
54.         app:layout_constraintEnd_toStartOf="@+id/et_set_deadline"
```

Experiment 11

Develop an android application to store data in the SQLite database.

```
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintHorizontal_chainStyle="packed"
        app:layout_constraintStart_toStartOf="parent" />

    <EditText
        android:id="@+id/et_activity_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:fontFamily="@font/roboto"
        android:hint="@string/ed_act_name_hint"
        android:inputType="textPersonName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/lb_activity_name"
        app:layout_constraintTop_toBottomOf="@+id/appTitle" />

    <EditText
        android:id="@+id/et_activity_desp"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:ems="10"
        android:fontFamily="@font/roboto"
        android:gravity="start|top"
        android:hint="@string/ed_act_desp_hint"
        android:inputType="textMultiLine"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/lb_activity_txt"
        app:layout_constraintTop_toBottomOf="@+id/et_activity_name" />

    <EditText
        android:id="@+id/et_set_deadline"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:ems="10"
        android:fontFamily="@font/roboto"
        android:hint="@string/ed_act_date_hint"
        android:inputType="date"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toEndOf="@+id/lb_activity_deadline"
            app:layout_constraintTop_toBottomOf="@+id/et_activity_desp" />

       <Button
            android:id="@+id/btn_add_activity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="@font/roboto"
            android:text="@string/btn_add_act"
            app:layout_constraintBaseline_toBaselineOf="@+id/btn_view_activities"
            app:layout_constraintEnd_toStartOf="@+id/btn_view_activities"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent" />

       <Button
            android:id="@+id/btn_view_activities"
```

Experiment 11
Develop an android application to store data in the SQLite database.

```
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="16dp"
                android:text="@string/btn_view_act"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.5"
                app:layout_constraintStart_toEndOf="@+id/btn_add_activity"
                app:layout_constraintTop_toBottomOf="@+id/et_set_deadline" />

            <ListView
                android:id="@+id/lv_allActivities"
                android:layout_width="0dp"
                android:layout_height="0dp"
                android:layout_marginTop="16dp"
                android:cacheColorHint="#FFFFFF"
                android:divider="@android:color/background_light"
                android:dividerHeight="5dp"
                app:layout_constraintBottom_toBottomOf="parent"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintHorizontal_bias="0.533"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/btn_view_activities"
                app:layout_constraintVertical_bias="0.647" />

        </androidx.constraintlayout.widget.ConstraintLayout>
```

Acitivity_style_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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="wrap_content"
    android:background="#E8E1F6">

    <TextView
        android:id="@+id/tv_act_title"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:fontFamily="@font/roboto"
        android:hint="@string/lb_act_name_hint"
        android:paddingStart="5dp"
        android:textColor="#000000"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/tv_act_time"
        app:layout_constraintEnd_toEndOf="@+id/tv_act_time"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_chainStyle="packed" />

    <TextView
        android:id="@+id/tv_act_info"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:fontFamily="@font/roboto"
        android:hint="@string/lb_actv_txt_hint"
```

Experiment 11
Develop an android application to store data in the SQLite database.

```
        android:paddingStart="5dp"
        android:paddingEnd="2dp"
        android:textColor="#000000"
        android:textSize="15sp"
        app:layout_constraintEnd_toStartOf="@+id/btn_delete"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_act_time" />

    <TextView
        android:id="@+id/tv_act_time"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:fontFamily="@font/roboto"
        android:hint="@string/lb_act_time_text"
        android:paddingStart="5dp"
        android:textColor="#000000"
        android:textSize="16sp"
        android:textStyle="italic"
        app:layout_constraintBottom_toTopOf="@+id/tv_act_info"
        app:layout_constraintEnd_toEndOf="@+id/tv_act_info"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_act_title" />

    <Button
        android:id="@+id/btn_delete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="8dp"
        android:background="#3949AB"
        android:fontFamily="@font/roboto"
        android:text="@string/btn_delete_txt"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textStyle="bold|italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/tv_act_title"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.sqlitedb_basix;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

```

Experiment 11
Develop an android application to store data in the SQLite database.

```
12. public class MainActivity extends AppCompatActivity {
13.
14.     EditText et_act_name, et_act_info, et_act_date;
15.     Button btn_add_act, btn_view_acts;
16.     ListView activityList;
17.     LayoutAdapter customAdapter;
18.     DatabaseHelper db_helper;
19.
20.     @Override
21.     protected void onCreate(Bundle savedInstanceState) {
22.         super.onCreate(savedInstanceState);
23.         setContentView(R.layout.activity_main);
24.
25.         //db intialization
26.         db_helper = new DatabaseHelper(MainActivity.this);
27.
28.         //view binding
29.         et_act_name = findViewById(R.id.et_activity_name);
30.         et_act_info = findViewById(R.id.et_activity_desp);
31.         et_act_date = findViewById(R.id.et_set_deadline);
32.         btn_add_act = findViewById(R.id.btn_add_activity);
33.         activityList = findViewById(R.id.lv_allActivities);
34.         btn_view_acts = findViewById(R.id.btn_view_activities);
35.
36.         //list view intialization
37.         ShowAllActivities();
38.
39.         btn_add_act.setOnClickListener(new View.OnClickListener() {
40.             @Override
41.             public void onClick(View view) {
42.                 if(et_act_name.length() != 0 && et_act_info.length() != 0 && et_act_dat
    e.length() != 0) {
43.                     try {
44.                         UserActivityModel u1 = new UserActivityModel(-
    1, et_act_name.getText().toString(), et_act_info.getText().toString(), et_act_date.getT
    ext().toString());
45.                         boolean insertStatus = db_helper.addActivity(u1);
46.                         Toast.makeText(getApplicationContext(), "Insert Status: " + ins
    ertStatus, Toast.LENGTH_LONG).show();
47.                         ShowAllActivities();
48.                         clearAllFields();
49.
50.                     } catch (Exception e) {
51.                         Toast.makeText(getApplicationContext(), "Error creating activit
    y!", Toast.LENGTH_LONG).show();
52.                     }
53.                 }
54.                 else {
55.                     Toast.makeText(getApplicationContext(), "All fields are required!",
     Toast.LENGTH_LONG).show();
56.                 }
57.             }
58.
59.         });
60.
61.         btn_view_acts.setOnClickListener(new View.OnClickListener() {
62.             @Override
63.             public void onClick(View view) {
64.                 try{
65.                     ShowAllActivities();
```

Experiment 11
Develop an android application to store data in the SQLite database.

```
                }catch(Exception e){
                    Toast.makeText(getApplicationContext(), "Error fetching activity!", Toast.LENGTH_LONG).show();
                }
            }
        });

    }


    private void ShowAllActivities() {
        //custom adapter
         customAdapter = new LayoutAdapter(getApplicationContext(),R.layout.activity_style_item,db_helper.getAllActivities());
        activityList.setAdapter(customAdapter);

        //activities = new ArrayAdapter<UserActivityModel>(MainActivity.this,android.R.layout.simple_list_item_1 ,db_helper.getAllActivities());
        //activityList.setAdapter(activities);
    }
    private void clearAllFields(){
        et_act_name.setText("");
        et_act_info.setText("");
        et_act_date.setText("");
    }
}
```

DatabaseHelper.java(Class)

```
package com.example.sqlitedb_basix;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String ACTIVITY_TABLE = "ACTIVITY_TABLE";
    public static final String COL_ACTIVITY_NAME = "Activity_Name";
    public static final String COL_ACTIVITY_INFO = "Activity_Info";
    public static final String COL_ACTIVITY_DEADLINE = "Activity_Deadline";
    public static final String COL_ID = "ID";

    public DatabaseHelper(@Nullable Context context) {
        super(context, "ActivityTracker.db", null, 1);
    }

    //this method is automatically called when you try to access the
    // DB for the very frist time.
```

Experiment 11

Develop an android application to store data in the SQLite database.

```
    // this includes codes that generates the DB and table
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableStm = "CREATE TABLE " + ACTIVITY_TABLE+ "(" + COL_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + COL_ACTIVITY_NAME + " TEXT NOT NULL, " + COL_ACTIVITY_INFO + " TEXT NOT NULL, " + COL_ACTIVITY_DEADLINE + " TEXT NOT NULL)";
        db.execSQL(createTableStm);
    }

    //this method is invoked when your DB's version is updated.
    //Suppose your app is used by hundred of users and you wanted to change the DB's schema without crashing the existing app.
    // like adding new colums or adding multiple tables
    //this method comes to resucue and upgrades the schema of older app DB to the latest one.
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }

    public boolean addActivity(UserActivityModel user_act){
        SQLiteDatabase db = this.getWritableDatabase(); //for insert actions
        ContentValues cv = new ContentValues(); //stores data in value pairs, similar to 'put Extra' in Intent
        cv.put(COL_ACTIVITY_NAME,user_act.getActivity_name());
        cv.put(COL_ACTIVITY_INFO,user_act.getActivity_info());
        cv.put(COL_ACTIVITY_DEADLINE,user_act.getActivity_deadline());

        final long insertStatus = db.insert(ACTIVITY_TABLE, null, cv);//returns insert status  as long (0> true, 0< false)

        return insertStatus != -1; //imp;licit conversion long to bool
    }
    public void deleteActivity(UserActivityModel user_act){
        SQLiteDatabase db = this.getWritableDatabase();
        String qry = "DELETE FROM "+ACTIVITY_TABLE+" WHERE "+ COL_ID +" = " + user_act.getActivity_id();
        Cursor cursor = db.rawQuery(qry, null);

        cursor.moveToFirst();
        db.close();
    }
    public List<UserActivityModel> getAllActivities(){
        List<UserActivityModel> resultantList = new ArrayList<>();

        //retreiving data from the db
        String qry = "SELECT * FROM "+ACTIVITY_TABLE;
        SQLiteDatabase db = this.getReadableDatabase(); //for reading data form the db


        //we can execute the qry 2 ways:
        // 1. exeSql
        // 2. rawQuery returns cursor(set of results)
       Cursor resultSet = db.rawQuery(qry,null);

       //checking if the result set is empty
        if(resultSet.moveToFirst()){
            //resultSet.moveToFirst(): returns a bool value if the dataset contains atleast 1 value

```

Experiment 11
Develop an android application to store data in the SQLite database.

```
            //loop through cursor list and store them in actual data list
            do{                                     // column index referenced with table schema
                String act_name = resultSet.getString(1);
                String act_info = resultSet.getString(2);
                String act_deadline = resultSet.getString(3);
                int act_ID = resultSet.getInt(0);
                UserActivityModel activityObj = new UserActivityModel(act_ID,act_name,act_info,act_deadline);
                resultantList.add(activityObj);
            }while(resultSet.moveToNext());
        }else{
            //failure, nothing to return as a result
        }
        resultSet.close();
        db.close();
        return  resultantList;
    }
}
```

UserActivityModel.java(Class)

```
package com.example.sqlitedb_basix;


public class UserActivityModel {

    private String activity_name, activity_info,activity_deadline;
    private int activity_id;
    public UserActivityModel(int activity_id,String activity_name, String activity_info, String activity_deadline) {

        this.activity_id = activity_id;
        this.activity_name = activity_name;
        this.activity_info = activity_info;
        this.activity_deadline = activity_deadline;
    }

    //toString to print data of the class object
    @Override
    public String toString() {
        return "UserActivityModel{" +
                "activity_name='" + activity_name + '\'' +
                ", activity_info='" + activity_info + '\'' +
                ", activity_deadline='" + activity_deadline + '\'' +
                ", activity_id=" + activity_id +
                '}';
    }

    public int getActivity_id() {
        return activity_id;
    }

    public void setActivity_id(int activity_id) {
        this.activity_id = activity_id;
    }

```

Experiment 11
Develop an android application to store data in the SQLite database.

```
35.     public String getActivity_deadline() {
36.         return activity_deadline;
37.     }
38.
39.     public void setActivity_deadline(String activity_deadline) {
40.         this.activity_deadline = activity_deadline;
41.     }
42.
43.     public UserActivityModel() {
44.     }
45.
46.     public String getActivity_name() {
47.         return activity_name;
48.     }
49.
50.     public void setActivity_name(String activity_name) {
51.         this.activity_name = activity_name;
52.     }
53.
54.     public String getActivity_info() {
55.         return activity_info;
56.     }
57.
58.     public void setActivity_info(String activity_info) {
59.         this.activity_info = activity_info;
60.     }
61.
62.
63. }
```

LayoutAdapter.java

```
1.  package com.example.sqlitedb_basix;
2.
3.  import android.content.Context;
4.  import android.view.LayoutInflater;
5.  import android.view.View;
6.  import android.view.ViewGroup;
7.  import android.widget.ArrayAdapter;
8.  import android.widget.Button;
9.  import android.widget.TextView;
10. import android.widget.Toast;
11.
12. import androidx.annotation.NonNull;
13. import androidx.annotation.Nullable;
14.
15.
16. import java.util.List;
17. import java.util.Objects;
18.
19. public class LayoutAdapter extends ArrayAdapter<UserActivityModel> {
20.     private Context appContext;
21.     private int itemLayout;   //  the xml layout we created to style each list item
22.     private List<UserActivityModel> allActivities;
23.     private DatabaseHelper db_helper;
24.
25.     public LayoutAdapter(@NonNull Context context, int layout, @NonNull List<UserActivi
    tyModel> objects) {
26.         super(context, layout, objects);
```

Experiment 11
Develop an android application to store data in the SQLite database.

```
27.         this.appContext = context;
28.         this.itemLayout = layout;
29.         this.allActivities = objects;
30.     }
31.
32.     /*
33.      *  getView() is responsible for rendering out each row
34.      * Here you define what information shows and where it sits within the ListView.
35.      * */
36.     @NonNull
37.     @Override
38.     public View getView(final int position, @Nullable View rowView, @NonNull ViewGroup
    generatedLayout) {
39.
40.         //initializing the inflater obj as per the context passed as an argument
41.         LayoutInflater layoutInflater = LayoutInflater.from(appContext);
42.         //db intialization
43.         db_helper = new DatabaseHelper(appContext);
44.
45.         //layout inflator gonna inflate each rowView as per the layout resource(in this
     case itemLayout) we pass a parameter.
46.         rowView = layoutInflater.inflate(itemLayout,generatedLayout,false);
47.         //generatedLayout:ViewGroup, is basically a view replica of 'itemLayout' file w
    e have created to style each item in our list.
48.         //generatedLayout decides how our rowView gonna look like.
49.         //dynamically generated for each item/row.
50.
51.
52.         TextView act_title = rowView.findViewById(R.id.tv_act_title);
53.         TextView act_date = rowView.findViewById(R.id.tv_act_time);
54.         TextView act_info = rowView.findViewById(R.id.tv_act_info);
55.         Button btn_dlt = rowView.findViewById(R.id.btn_delete);
56.
57.         act_title.setText(Objects.requireNonNull(getItem(position)).getActivity_name())
    ;
58.         act_date.setText(Objects.requireNonNull(getItem(position)).getActivity_deadline
    ());
59.         act_info.setText(Objects.requireNonNull(getItem(position)).getActivity_info());

60.
61.         //deleting specific activity from the DB
62.         btn_dlt.setOnClickListener(new View.OnClickListener() {
63.             @Override
64.             public void onClick(View v) {
65.                 UserActivityModel clickedActivity = (UserActivityModel) allActivities.g
    et(position); //got the activity obj
66.                 db_helper.deleteActivity(clickedActivity);
67.                 allActivities.remove(position);
68.                 notifyDataSetChanged();
69.                 Toast.makeText(appContext,"Activity Deleted!",Toast.LENGTH_SHORT).show(
    );
70.             }
71.         });
72.
73.         return rowView;
74.     }
75.
76.
77. }
```

Experiment 11
Develop an android application to store data in the SQLite database.

Snapshots: