Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

MainActivity.xml

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.a
    ndroid.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="#F2F2F2"
8.      android:padding="10dp"
9.      tools:context=".MainActivity">
10.
11.         <TextView
12.             android:id="@+id/appName"
13.             android:layout_width="wrap_content"
14.             android:layout_height="wrap_content"
15.             android:layout_marginTop="16dp"
16.             android:fontFamily="@font/faster_one"
17.             android:text="@string/appName"
18.             android:textSize="24sp"
19.             app:layout_constraintEnd_toEndOf="parent"
20.             app:layout_constraintHorizontal_bias="0.5"
21.             app:layout_constraintStart_toStartOf="parent"
22.             app:layout_constraintTop_toTopOf="parent" />
23.
24.         <EditText
25.             android:id="@+id/et_empID"
26.             android:layout_width="match_parent"
27.             android:layout_height="wrap_content"
28.             android:layout_marginTop="24dp"
29.             android:ems="10"
30.             android:fontFamily="@font/roboto"
31.             android:hint="@string/idHint"
32.             android:inputType="textPersonName"
33.             android:textSize="16sp"
34.             app:layout_constraintEnd_toEndOf="parent"
35.             app:layout_constraintHorizontal_bias="0.5"
36.             app:layout_constraintStart_toStartOf="parent"
37.             app:layout_constraintTop_toBottomOf="@+id/appName" />
38.
39.         <EditText
40.             android:id="@+id/et_empName"
41.             android:layout_width="match_parent"
42.             android:layout_height="wrap_content"
43.             android:layout_marginTop="8dp"
44.             android:ems="10"
45.             android:fontFamily="@font/roboto"
46.             android:hint="@string/nameHint"
47.             android:inputType="textPersonName"
48.             android:textSize="16sp"
49.             app:layout_constraintEnd_toEndOf="parent"
50.             app:layout_constraintHorizontal_bias="0.5"
51.             app:layout_constraintStart_toStartOf="parent"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
            app:layout_constraintTop_toBottomOf="@+id/et_empID" />

        <EditText
            android:id="@+id/et_empDpt"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:ems="10"
            android:fontFamily="@font/roboto"
            android:hint="@string/dptHint"
            android:inputType="textPersonName"
            android:textSize="16sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/et_empName" />

        <EditText
            android:id="@+id/et_empYOJ"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:ems="10"
            android:fontFamily="@font/roboto"
            android:hint="@string/yojHint"
            android:inputType="textPersonName"
            android:textSize="16sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/et_empDpt" />

        <EditText
            android:id="@+id/et_empContact"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:ems="10"
            android:fontFamily="@font/roboto"
            android:hint="@string/contactHint"
            android:inputType="textPersonName"
            android:textSize="16sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/et_empYOJ" />

        <Button
                    android:id="@+id/btn_register"
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_marginStart="8dp"
                    android:layout_marginTop="8dp"
                    android:layout_marginEnd="8dp"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
                    android:fontFamily="@font/dosis"
                    android:text="@string/btnTxt"
                    android:textSize="16sp"
                    android:textStyle="bold"
                    app:layout_constraintEnd_toStartOf="@+id/btn_viewAll"
                    app:layout_constraintHorizontal_bias="0.5"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/et_empContact" />

                <Button
                    android:id="@+id/btn_viewAll"
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_marginEnd="8dp"
                    android:text="@string/btnViewAll"
                    app:layout_constraintBaseline_toBaselineOf="@+id/btn_register"

                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintHorizontal_bias="0.5"
                    app:layout_constraintStart_toEndOf="@+id/btn_register" />

                <ListView
                    android:id="@+id/lv_empList"
                    android:layout_width="0dp"
                    android:layout_height="0dp"
                    android:layout_marginTop="8dp"
                    app:layout_constraintBottom_toBottomOf="parent"
                    app:layout_constraintEnd_toEndOf="parent"
                    app:layout_constraintHorizontal_bias="0.5"
                    app:layout_constraintStart_toStartOf="parent"
                    app:layout_constraintTop_toBottomOf="@+id/btn_register" />

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

activity_all_employees_list.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="match_parent"
    tools:context=".AllEmployeesList">

    <TextView
        android:id="@+id/tv_orgName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:fontFamily="@font/faster_one"
        android:text="@string/orgName"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ListView
        android:id="@+id/lv_allEmps"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_orgName" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Readable_item_style.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">

    <TextView
        android:id="@+id/tv_ID"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:fontFamily="@font/faster_one"
        android:paddingStart="5dp"
        android:text="Text View"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tv_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
        android:text="TextView"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_ID" />

    <TextView
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
        android:id="@+id/tv_dpt"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
        android:text="TextView"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_name" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

style_list_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="match_parent">


    <TextView
        android:id="@+id/tv_empID"
        android:layout_width="match_parent"
        android:layout_height="23dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:fontFamily="@font/faster_one"
        android:paddingStart="5dp"
        android:text="TextView"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />


    <TextView
        android:id="@+id/tv_empName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
        android:text="TextView"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_empID" />
    <!-- et Name hidden, when visible replaces tv Name -->
    <EditText
        android:id="@+id/up_empName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_empID"
        android:visibility="gone"/>

    <TextView
        android:id="@+id/tv_empDpt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
        android:text="TextView"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tv_empName" />
    <!-- et Dpt hidden, when visible replaces tv Dpt -->
    <EditText
        android:id="@+id/up_empDpt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:fontFamily="@font/dosis"
        android:paddingStart="5dp"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/up_empName"
        android:visibility="gone"/>


    <TextView
        android:id="@+id/tv_empYOJ"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
              android:layout_marginTop="8dp"
              android:layout_marginEnd="8dp"
              android:fontFamily="@font/dosis"
              android:paddingStart="5dp"
              android:text="TextView"
              android:textSize="16sp"
              android:textStyle="bold"
              app:layout_constraintEnd_toEndOf="parent"
              app:layout_constraintHorizontal_bias="0.5"
              app:layout_constraintStart_toStartOf="parent"
              app:layout_constraintTop_toBottomOf="@+id/tv_empDpt" />
          <!-- et yoj hidden, when visible replaces tv yoj -->
          <EditText
              android:id="@+id/up_empYOJ"
              android:layout_width="match_parent"
              android:layout_height="wrap_content"
              android:layout_marginStart="8dp"
              android:layout_marginTop="8dp"
              android:layout_marginEnd="8dp"
              android:fontFamily="@font/dosis"
              android:paddingStart="5dp"
              android:textSize="16sp"
              android:textStyle="bold"
              app:layout_constraintEnd_toEndOf="parent"
              app:layout_constraintHorizontal_bias="0.5"
              app:layout_constraintStart_toStartOf="parent"
              app:layout_constraintTop_toBottomOf="@+id/up_empDpt"
              android:visibility="gone"/>

          <TextView
              android:id="@+id/tv_empContact"
              android:layout_width="match_parent"
              android:layout_height="wrap_content"
              android:layout_marginStart="8dp"
              android:layout_marginTop="8dp"
              android:layout_marginEnd="8dp"
              android:fontFamily="@font/dosis"
              android:paddingStart="5dp"
              android:text="TextView"
              android:textSize="16sp"
              android:textStyle="bold"
              app:layout_constraintEnd_toEndOf="parent"
              app:layout_constraintHorizontal_bias="0.5"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
143.                app:layout_constraintStart_toStartOf="parent"
144.                app:layout_constraintTop_toBottomOf="@+id/tv_empYOJ" />
145.            <!-- et yoj hidden, when visible replaces tv yoj -->
146.            <EditText
147.                android:id="@+id/up_empContact"
148.                android:layout_width="match_parent"
149.                android:layout_height="wrap_content"
150.                android:layout_marginStart="8dp"
151.                android:layout_marginTop="8dp"
152.                android:layout_marginEnd="8dp"
153.                android:fontFamily="@font/dosis"
154.                android:paddingStart="5dp"
155.                android:textSize="16sp"
156.                android:textStyle="bold"
157.                app:layout_constraintEnd_toEndOf="parent"
158.                app:layout_constraintHorizontal_bias="0.5"
159.                app:layout_constraintStart_toStartOf="parent"
160.                app:layout_constraintTop_toBottomOf="@+id/up_empYOJ"
161.                android:visibility="gone"/>
162.
163.
164.            <Button
165.                android:id="@+id/btn_dlt"
166.                android:layout_width="0dp"
167.                android:layout_height="wrap_content"
168.                android:layout_marginStart="8dp"
169.                android:layout_marginTop="8dp"
170.                android:layout_marginEnd="8dp"
171.                android:fontFamily="@font/dosis"
172.                android:text="@string/dltBtn_txt"
173.                android:textSize="16sp"
174.                android:textStyle="bold"
175.                app:layout_constraintEnd_toStartOf="@+id/btn_edit"
176.                app:layout_constraintHorizontal_bias="0.5"
177.                app:layout_constraintHorizontal_chainStyle="packed"
178.                app:layout_constraintStart_toStartOf="parent"
179.                app:layout_constraintTop_toBottomOf="@+id/tv_empContact" />
180.
181.            <!-- Cancel btn hidden when visible replaces Delete btn -->
182.            <Button
183.                android:id="@+id/btn_cancel"
184.                android:layout_width="0dp"
185.                android:layout_height="wrap_content"
186.                android:layout_marginStart="8dp"
187.                android:layout_marginTop="8dp"
188.                android:layout_marginEnd="8dp"
189.                android:fontFamily="@font/dosis"
190.                android:text="@string/cancelBtn"
191.                android:textSize="16sp"
192.                android:textStyle="bold"
193.                app:layout_constraintEnd_toStartOf="@+id/btn_update"
194.                app:layout_constraintHorizontal_bias="0.5"
195.                app:layout_constraintHorizontal_chainStyle="packed"
196.                app:layout_constraintStart_toStartOf="parent"
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
            app:layout_constraintTop_toBottomOf="@+id/up_empContact"
            android:visibility="gone"/>

        <Button
            android:id="@+id/btn_edit"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginEnd="8dp"
            android:fontFamily="@font/dosis"
            android:text="@string/btnEdit"
            android:textSize="16sp"
            android:textStyle="bold"
            app:layout_constraintBaseline_toBaselineOf="@+id/btn_dlt"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toEndOf="@+id/btn_dlt" />

        <!-- Update btn hidden when visible replaces Edit btn -->
        <Button
            android:id="@+id/btn_update"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:layout_marginEnd="8dp"
            android:fontFamily="@font/dosis"
            android:text="@string/updateBtn"
            android:textSize="16sp"
            android:textStyle="bold"
            app:layout_constraintBaseline_toBaselineOf="@+id/btn_cancel"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.5"
            app:layout_constraintStart_toEndOf="@+id/btn_cancel"
            android:visibility="gone"/>

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

MainActivity.java

```
package com.example.employeedb_app;

import androidx.appcompat.app.AppCompatActivity;

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

```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
14. public class MainActivity extends AppCompatActivity {
15.
16.     EditText emp_ID, emp_name, emp_dpt, emp_yoj, emp_contact;
17.     Button btn_register, btn_viewAll;
18.     ListView emp_list;
19.     LayoutAdapter2 customAdapter;
20.     DbHelper db_help;
21.
22.     @Override
23.     protected void onCreate(Bundle savedInstanceState) {
24.         super.onCreate(savedInstanceState);
25.         setContentView(R.layout.activity_main);
26.
27.         //initializing db with context to our main activity
28.         db_help = new DbHelper(MainActivity.this);
29.
30.         emp_ID = findViewById(R.id.et_empID);
31.         emp_name = findViewById(R.id.et_empName);
32.         emp_dpt = findViewById(R.id.et_empDpt);
33.         emp_yoj = findViewById(R.id.et_empYOJ);
34.         emp_contact = findViewById(R.id.et_empContact);
35.         emp_list = findViewById(R.id.lv_empList);
36.         btn_register = findViewById(R.id.btn_register);
37.         btn_viewAll = findViewById(R.id.btn_viewAll);
38.
39.
40.         ShowAllEmployees();
41.
42.         //view all emps
43.         btn_viewAll.setOnClickListener(new View.OnClickListener() {
44.             @Override
45.             public void onClick(View v) {
46.                 Intent allEmps = new Intent(getApplicationContext(),AllEmployeesLi
    st.class);
47.                 startActivity(allEmps);
48.             }
49.         });
50.
51.         //registering emp on clicking btn
52.         btn_register.setOnClickListener(new View.OnClickListener() {
53.             @Override
54.             public void onClick(View v) {
55.                 if(emp_ID.length() != 0 && emp_name.length() != 0 && emp_dpt.lengt
    h() != 0 && emp_yoj.length() != 0 && emp_contact.length() != 0){
56.
57.                     try {
58.                         //retrieve all info
59.                         EmployeeModel emp = new EmployeeModel(-
    1,emp_ID.getText().toString().trim(),emp_name.getText().toString().trim(),emp_dpt.
    getText().toString().trim(),emp_yoj.getText().toString().trim(),emp_contact.getTex
    t().toString().trim());
60.                         boolean registerStatus = db_help.addEmployeeDetails(emp);
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
61.                         Toast.makeText(MainActivity.this,"Employee Registered Succ
   essfully? : "+registerStatus,Toast.LENGTH_LONG).show();
62.                         ShowAllEmployees();
63.                         clearAllFields();
64.                     } catch (Exception e) {
65.                         Toast.makeText(MainActivity.this, "Error Registering Emplo
   yee! error: "+e.getMessage(), Toast.LENGTH_LONG).show();
66.                     }
67.
68.                 }else
69.                     Toast.makeText(MainActivity.this,"All Fields are required!",To
   ast.LENGTH_LONG).show();
70.             }
71.         });
72.     }
73.
74.     private void clearAllFields() {
75.         emp_ID.setText("");
76.         emp_name.setText("");
77.         emp_dpt.setText("");
78.         emp_yoj.setText("");
79.         emp_contact.setText("");
80.     }
81.
82.     private void ShowAllEmployees() {
83.         //custom adapter
84.         customAdapter = new LayoutAdapter2(getApplicationContext(),R.layout.readab
   le_item_style,db_help.getAllEmployees());
85.         emp_list.setAdapter(customAdapter);
86.     }
87.
88.
89. }
```

AllEmployeesList.java

```
1.  package com.example.employeedb_app;
2.
3.  import androidx.appcompat.app.AppCompatActivity;
4.
5.  import android.content.Intent;
6.  import android.os.Bundle;
7.  import android.view.View;
8.  import android.widget.Button;
9.  import android.widget.ListView;
10.
11. public class AllEmployeesList extends AppCompatActivity {
12.
13.     ListView emps_list;
14.     LayoutAdapter customAdapter;
15.     DbHelper db_help;
16.     Button btnBack;
17.
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
18.     @Override
19.     protected void onCreate(Bundle savedInstanceState) {
20.         super.onCreate(savedInstanceState);
21.         setContentView(R.layout.activity_all_employees_list);
22.
23.         emps_list = findViewById(R.id.lv_allEmps);
24.         btnBack = findViewById(R.id.btnBack);
25.
26.         //initializing db with context to our main activity
27.         db_help = new DbHelper(AllEmployeesList.this);
28.         customAdapter = new LayoutAdapter(AllEmployeesList.this,R.layout.style_lis
    t_item,db_help.getAllEmployees());
29.         emps_list.setAdapter(customAdapter);
30.
31.         btnBack.setOnClickListener(new View.OnClickListener() {
32.             @Override
33.             public void onClick(View v) {
34.                 Intent home = new Intent(getApplicationContext(),MainActivity.clas
    s);
35.                 startActivity(home);
36.             }
37.         });
38.
39.     }
40. }
```

DbHepler.java(Class file)

```
1.  package com.example.employeedb_app;
2.
3.  import android.content.ContentValues;
4.  import android.content.Context;
5.  import android.database.Cursor;
6.  import android.database.sqlite.SQLiteDatabase;
7.  import android.database.sqlite.SQLiteOpenHelper;
8.
9.  import androidx.annotation.Nullable;
10.
11. import java.util.ArrayList;
12. import java.util.List;
13.
14. public class DbHelper extends SQLiteOpenHelper {
15.     public static final String EMP_TABLE = "EMP_TABLE";
16.     public static final String COL_EMP_SR= "Emp_SR";
17.     public static final String COL_EMP_ID = "Emp_ID";
18.     public static final String COL_EMP_NAME = "Emp_Name";
19.     public static final String COL_EMP_DPT= "Emp_Dpt";
20.     public static final String COL_EMP_YOJ = "Emp_YOJ";
21.     public static final String COL_EMP_CONTACT = "Emp_Contact";
22.
23.
24.     public DbHelper(@Nullable Context context) {
25.         super(context, "S7Employees.db", null, 1);
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
26.     }
27.
28.     @Override
29.     public void onCreate(SQLiteDatabase db) {
30.         String createTableStm = "CREATE TABLE " + EMP_TABLE+ "(" + COL_EMP_SR + "
    INTEGER PRIMARY KEY AUTOINCREMENT, " + COL_EMP_ID + " TEXT NOT NULL, " + COL_EMP_N
    AME + " TEXT NOT NULL, " + COL_EMP_DPT + " TEXT NOT NULL, "+ COL_EMP_YOJ + " TEXT
    NOT NULL, " + COL_EMP_CONTACT +" TEXT NOT NULL)";
31.         db.execSQL(createTableStm);
32.     }
33.
34.     @Override
35.     public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
36.         //not require as of now
37.         //this method is invoked when your DB's version is updated.
38.     }
39.
40.
41.     //INSERTING EMP DATA
42.     public boolean addEmployeeDetails(EmployeeModel emp){
43.
44.         //setting up vars
45.         SQLiteDatabase db = this.getWritableDatabase(); //accessing the DB in writ
    e mode
46.         ContentValues cv = new ContentValues(); //for storing information in key,v
    al pair
47.
48.         //initializing table columns value
49.         cv.put(COL_EMP_ID,emp.getEmp_ID());
50.         cv.put(COL_EMP_NAME,emp.getEmp_name());
51.         cv.put(COL_EMP_DPT,emp.getEmp_dpt());
52.         cv.put(COL_EMP_YOJ, emp.getEmp_yoj());
53.         cv.put(COL_EMP_CONTACT,emp.getEmp_contact());
54.
55.         //writing cv into the respective table
56.         final  long insertStatus = db.insert(EMP_TABLE,null,cv); //return +ve for
    true and -ve for false
57.         db.close();
58.         return insertStatus != -1; //returns bool value after comparision.
59.
60.     }
61.
62.
63.     //DELETING EMP FROM DB
64.     public void deleteEmployeeData(EmployeeModel emp){
65.         SQLiteDatabase db = this.getWritableDatabase();
66.         String qry = "DELETE FROM " + EMP_TABLE + " WHERE " + COL_EMP_SR + " = " +
     emp.getEmp_SR();
67.         Cursor cursor = db.rawQuery(qry, null);
68.
69.         cursor.moveToFirst();
70.         db.close();
71.     }
72.
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
73.     //UPDATING EMP
74.     public void  updateEmployeeData(EmployeeModel emp) {
75.         SQLiteDatabase db = this.getWritableDatabase();
76.         ContentValues cv = new ContentValues(); //for storing information in key,v
    al pair
77.
78.         //initializing table columns value
79.         cv.put(COL_EMP_NAME,emp.getEmp_name());
80.         cv.put(COL_EMP_DPT,emp.getEmp_dpt());
81.         cv.put(COL_EMP_YOJ, emp.getEmp_yoj());
82.         cv.put(COL_EMP_CONTACT,emp.getEmp_contact());
83.
84.         db.update(EMP_TABLE,cv,COL_EMP_ID+" = '"+emp.getEmp_ID()+"'",null);
85.         db.close();
86.     }
87.     //GETTING ALL EMP LIST
88.     public List<EmployeeModel> getAllEmployees(){
89.         //inserting all emp records
90.         List<EmployeeModel> emp_list = new ArrayList<>();
91.
92.         //retreiving data from the db
93.         String qry = "SELECT * FROM "+EMP_TABLE;
94.         SQLiteDatabase db = this.getReadableDatabase(); //for reading data form th
    e db
95.
96.         //we can execute the qry 2 ways:
97.         // 1. exeSql
98.         // 2. rawQuery returns cursor(set of results)
99.         Cursor resultSet = db.rawQuery(qry,null);
100.
101.                //checking if the result set is empty
102.                if(resultSet.moveToFirst()){
103.                    //resultSet.moveToFirst(): returns a bool value if the dataset
    contains atleast 1 value
104.
105.                    //loop through cursor list and store them in actual data list
106.                    do{
107.                        // column index referenced with table schema
108.                        int emp_sr = resultSet.getInt(0);
109.                        String emp_id = resultSet.getString(1);
110.                        String emp_name = resultSet.getString(2);
111.                        String emp_dpt = resultSet.getString(3);
112.                        String emp_yoj = resultSet.getString(4);
113.                        String emp_contact = resultSet.getString(5);
114.
115.                        EmployeeModel emp = new EmployeeModel(emp_sr, emp_id, emp_n
    ame,emp_dpt,emp_yoj,emp_contact);
116.                        emp_list.add(emp);
117.                    }while(resultSet.moveToNext());
118.
119.                }else{
120.                    //failure, nothing to return as a result
121.                }
122.                resultSet.close();
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
123.                db.close();
124.                return emp_list;
125.            }
126.
127.
128.        }
```

EmployeeModel.java (Class file)

```
1.  package com.example.employeedb_app;
2.
3.  public class EmployeeModel {
4.
5.      private String emp_name;
6.      private String emp_dpt;
7.      private String emp_contact;
8.      private String emp_yoj;
9.      private String emp_ID;
10.     private int emp_SR;
11.
12.
13.     public EmployeeModel(int emp_sr, String emp_ID, String emp_name, String emp_dp
    t, String emp_yoj, String emp_contact) {
14.         this.emp_name = emp_name;
15.         this.emp_dpt = emp_dpt;
16.         this.emp_contact = emp_contact;
17.         this.emp_yoj = emp_yoj;
18.         this.emp_ID = emp_ID;
19.         this.emp_SR = emp_sr;
20.     }
21.
22.     //Auto Inc row no. for the respective emp obj.
23.     public int getEmp_SR() {
24.         return emp_SR;
25.     }
26.     //ID
27.     public String getEmp_ID() {
28.         return emp_ID;
29.     }
30.     public void setEmp_ID(String emp_ID) {
31.         this.emp_ID = emp_ID;
32.     }
33.     //Name
34.     public String getEmp_name() {
35.         return emp_name;
36.     }
37.     public void setEmp_name(String emp_name) {
38.         this.emp_name = emp_name;
39.     }
40.     //Department
41.     public String getEmp_dpt() {
42.         return emp_dpt;
43.     }
44.     public void setEmp_dpt(String emp_dpt) {
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
45.         this.emp_dpt = emp_dpt;
46.     }
47.     //Contact
48.     public String getEmp_contact() {
49.         return emp_contact;
50.     }
51.     public void setEmp_contact(String emp_contact) {
52.         this.emp_contact = emp_contact;
53.     }
54.     //YOJ
55.     public String getEmp_yoj() {
56.         return emp_yoj;
57.     }
58.     public void setEmp_yoj(String emp_yoj) {
59.         this.emp_yoj = emp_yoj;
60.     }
61.
62.
63.     //toString() to print data of the class object
64.     @Override
65.     public String toString() {
66.         return "EmployeeModel{" +
67.                 "emp_name='" + emp_name + '\'' +
68.                 ", emp_dpt='" + emp_dpt + '\'' +
69.                 ", emp_contact='" + emp_contact + '\'' +
70.                 ", emp_yoj='" + emp_yoj + '\'' +
71.                 ", emp_ID=" + emp_ID +
72.                 '}';
73.     }
74. }
```

LayoutAdapter.java(Class file for updating emp record)

```
1.  package com.example.employeedb_app;
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.EditText;
10. import android.widget.TextView;
11. import android.widget.Toast;
12.
13. import androidx.annotation.NonNull;
14. import androidx.annotation.Nullable;
15. import androidx.appcompat.app.AppCompatActivity;
16.
17. import java.util.List;
18. import java.util.Objects;
19.
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
20. public class LayoutAdapter extends ArrayAdapter<EmployeeModel> {
21.
22.     private Context appContext;
23.     private int itemLayout;  //the xml layout we created to style each list item
24.     private List<EmployeeModel> allEmps;
25.     private DbHelper db_help;
26.
27.     public LayoutAdapter(@NonNull Context context, int layout, List<EmployeeModel>
    emps) {
28.         super(context, layout, emps);
29.         this.appContext = context;
30.         this.itemLayout = layout;
31.         this.allEmps = emps;
32.
33.     }
34.     /*
35.      *  getView() is responsible for rendering out each row
36.      * Here you define what information shows and where it sits within the ListVie
   w.
37.      * */
38.     @NonNull
39.     @Override
40.     public View getView(final int position, @Nullable View rowView, @NonNull ViewG
   roup generatedLayout) {
41.
42.         //initializing the inflater obj as per the context passed as an argument
43.         LayoutInflater layoutInflater = LayoutInflater.from(appContext);
44.         //db intialization
45.         db_help = new DbHelper(appContext);
46.
47.         //layout inflator gonna inflate each rowView as per the layout resource(in
    this case itemLayout) we pass a parameter.
48.         rowView = layoutInflater.inflate(itemLayout,generatedLayout,false);
49.         //generatedLayout:ViewGroup, is basically a view replica of 'itemLayout' f
   ile we have created to style each item in our list.
50.         //generatedLayout decides how our rowView gonna look like.
51.         //dynamically generated for each item/row.
52.
53.
54.         //Default layout
55.         final TextView empID = rowView.findViewById(R.id.tv_empID);
56.         final TextView empName = rowView.findViewById(R.id.tv_empName);
57.         final TextView empDpt = rowView.findViewById(R.id.tv_empDpt);
58.         final TextView empYOJ = rowView.findViewById(R.id.tv_empYOJ);
59.         final TextView empContact= rowView.findViewById(R.id.tv_empContact);
60.
61.         final Button btn_dlt = rowView.findViewById(R.id.btn_dlt);
62.         final Button btn_edit = rowView.findViewById(R.id.btn_edit);
63.
64.         //Update operation layout
65.         final Button btn_update = rowView.findViewById(R.id.btn_update);
66.         final Button btn_cancel = rowView.findViewById(R.id.btn_cancel);
67.
68.         final EditText up_empName = rowView.findViewById(R.id.up_empName);
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
69.         final EditText up_empDpt = rowView.findViewById(R.id.up_empDpt);
70.         final EditText up_empYOJ = rowView.findViewById(R.id.up_empYOJ);
71.         final EditText up_empContact = rowView.findViewById(R.id.up_empContact);
72.
73.         empID.setText("S7.dev/emp:"+Objects.requireNonNull(getItem(position)).getE
    mp_ID());
74.         empName.setText(Objects.requireNonNull(getItem(position)).getEmp_name());

75.         empDpt.setText(Objects.requireNonNull(getItem(position)).getEmp_dpt());
76.         empYOJ.setText(Objects.requireNonNull(getItem(position)).getEmp_yoj());
77.         empContact.setText(Objects.requireNonNull(getItem(position)).getEmp_contac
    t());
78.
79.         //deleting specific activity from the DB
80.         btn_dlt.setOnClickListener(new View.OnClickListener() {
81.             @Override
82.             public void onClick(View v) {
83.                 EmployeeModel targetEmp = (EmployeeModel) allEmps.get(position); /
    /got the activity obj
84.                 db_help.deleteEmployeeData(targetEmp);
85.                 allEmps.remove(position);
86.                 notifyDataSetChanged();
87.                 Toast.makeText(appContext,"Employee Data Deleted!",Toast.LENGTH_SH
    ORT).show();
88.             }
89.         });
90.
91.         //Just preparing the layout for the update operation
92.         btn_edit.setOnClickListener(new View.OnClickListener() {
93.             @Override
94.             public void onClick(View v) {
95.
96.                 //Replacing all btns
97.                 btn_edit.setVisibility(View.INVISIBLE);
98.                 btn_update.setVisibility(View.VISIBLE);
99.                 btn_dlt.setVisibility(View.INVISIBLE);
100.                    btn_cancel.setVisibility(View.VISIBLE);
101.
102.                    //replacing all tvs with ets
103.                    //Name
104.                    empName.setVisibility(View.INVISIBLE);
105.                    up_empName.setVisibility(View.VISIBLE);
106.                    up_empName.setText(empName.getText().toString());
107.                    //Dpt
108.                    empDpt.setVisibility(View.INVISIBLE);
109.                    up_empDpt.setVisibility(View.VISIBLE);
110.                    up_empDpt.setText(empDpt.getText().toString());
111.                    //YOJ
112.                    empYOJ.setVisibility(View.INVISIBLE);
113.                    up_empYOJ.setVisibility(View.VISIBLE);
114.                    up_empYOJ.setText(empYOJ.getText().toString());
115.                    //Contact
116.                    empContact.setVisibility(View.INVISIBLE);
117.                    up_empContact.setVisibility(View.VISIBLE);
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
                        up_empContact.setText(empContact.getText().toString());


                    }
                });

                //performing update operation
                btn_update.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                        EmployeeModel targetEmp = (EmployeeModel) allEmps.get(position); //got the activity obj


                        //getting updated values
                        if(up_empName.length() != 0 && up_empDpt.length() != 0 && up_empYOJ.length() != 0 && up_empContact.length() != 0){
                        targetEmp.setEmp_name(up_empName.getText().toString());
                        targetEmp.setEmp_dpt(up_empDpt.getText().toString());
                        targetEmp.setEmp_yoj(up_empYOJ.getText().toString());
                        targetEmp.setEmp_contact(up_empContact.getText().toString());
                            try {
                                db_help.updateEmployeeData(targetEmp);
                                notifyDataSetChanged();
                                Toast.makeText(appContext,"Successfully Updated!",Toast.LENGTH_LONG).show();

                            }catch (Exception e) {
                                Toast.makeText(appContext,"Updation Error! error: "+e.getMessage(),Toast.LENGTH_LONG).show();
                            }
                        }
                        else
                            Toast.makeText(appContext, "Fields can't be empty!", Toast.LENGTH_LONG).show();


                        //showing defaults btns: dlt & upt
                        btn_edit.setVisibility(View.VISIBLE);
                        btn_update.setVisibility(View.GONE);
                        btn_dlt.setVisibility(View.VISIBLE);
                        btn_cancel.setVisibility(View.GONE);
                        //replacing all tvs with ets
                        //Name
                        empName.setVisibility(View.VISIBLE);
                        up_empName.setVisibility(View.GONE);
                        //Dpt
                        empDpt.setVisibility(View.VISIBLE);
                        up_empDpt.setVisibility(View.GONE);
                        //YOJ
                        empYOJ.setVisibility(View.VISIBLE);
                        up_empYOJ.setVisibility(View.GONE);
```

Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
                        //Contact
                        empContact.setVisibility(View.VISIBLE);
                        up_empContact.setVisibility(View.GONE);

                    }
                });

                //perform cancel update operation
                btn_cancel.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {


                        //showing defaults btns: dlt & upt
                        btn_edit.setVisibility(View.VISIBLE);
                        btn_update.setVisibility(View.GONE);
                        btn_dlt.setVisibility(View.VISIBLE);
                        btn_cancel.setVisibility(View.GONE);
                        //replacing all tvs with ets
                        //Name
                        empName.setVisibility(View.VISIBLE);
                        up_empName.setVisibility(View.GONE);
                        //Dpt
                        empDpt.setVisibility(View.VISIBLE);
                        up_empDpt.setVisibility(View.GONE);
                        //YOJ
                        empYOJ.setVisibility(View.VISIBLE);
                        up_empYOJ.setVisibility(View.GONE);
                        //Contact
                        empContact.setVisibility(View.VISIBLE);
                        up_empContact.setVisibility(View.GONE);

                    }
                });

                return rowView;
            }

        }
```

LayoutAdpater2.java(simple list for emp records)

```
package com.example.employeedb_app;

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

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

```

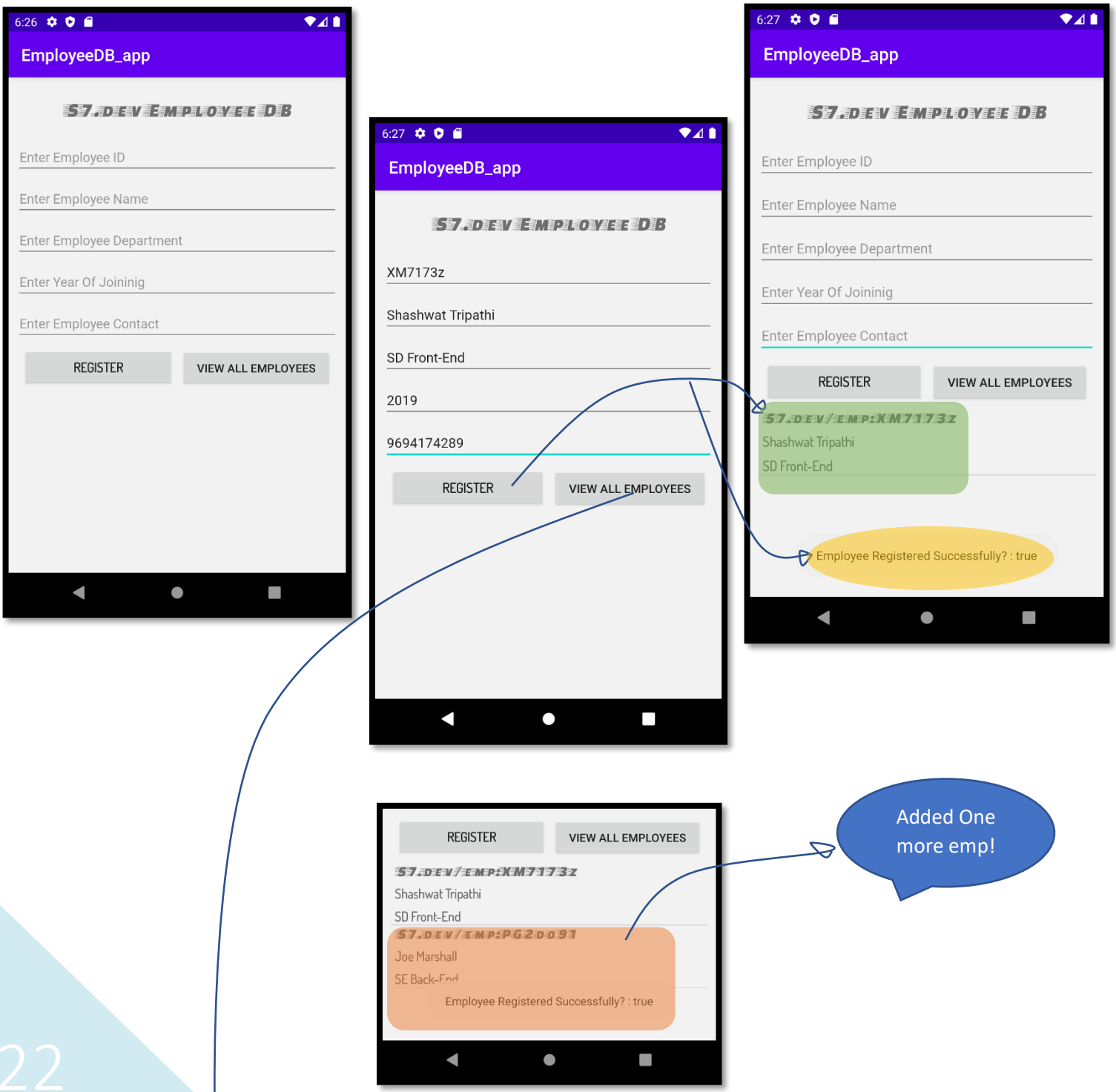
Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

```
13. import org.w3c.dom.Text;
14.
15. import java.util.List;
16. import java.util.Objects;
17.
18. public class LayoutAdapter2  extends ArrayAdapter<EmployeeModel> {
19.
20.     private Context appContext;
21.     private int itemLayout;  //the xml layout we created to style each list item
22.     private List<EmployeeModel> allEmps;
23.     private DbHelper db_help;
24.
25.     public LayoutAdapter2(@NonNull Context context, int layout, List<EmployeeModel
    > emps) {
26.         super(context, layout, emps);
27.         this.appContext = context;
28.         this.itemLayout = layout;
29.         this.allEmps = emps;
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 ListVie
    w.
35.      * */
36.     @NonNull
37.     @Override
38.     public View getView(final int position, @Nullable View rowView, @NonNull ViewG
    roup 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_help = new DbHelper(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' f
    ile we 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.         //getting views
52.         TextView ID = rowView.findViewById(R.id.tv_ID);
53.         TextView Name = rowView.findViewById(R.id.tv_name);
54.         TextView Dpt = rowView.findViewById(R.id.tv_dpt);
55.
56.         //setting views
57.         ID.setText("S7.dev/emp:"+ Objects.requireNonNull(getItem(position)).getEmp
    _ID());
58.         Name.setText(Objects.requireNonNull(getItem(position)).getEmp_name());
59.         Dpt.setText(Objects.requireNonNull(getItem(position)).getEmp_dpt());
60.
```

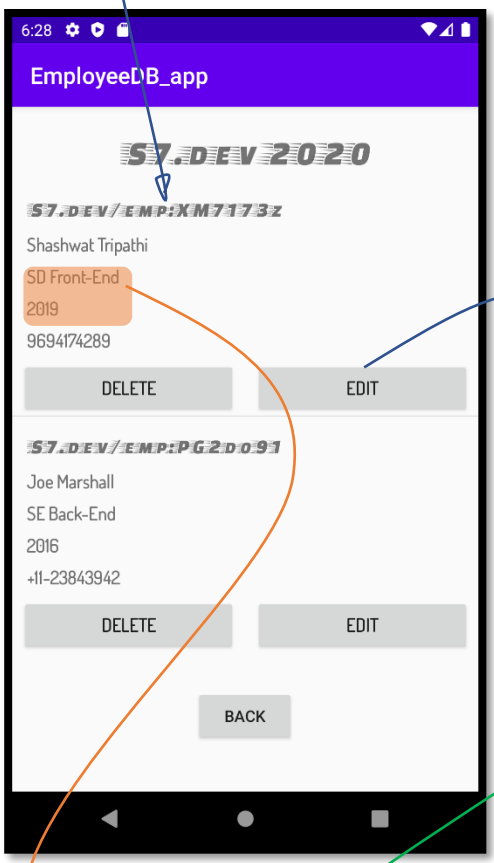
Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.

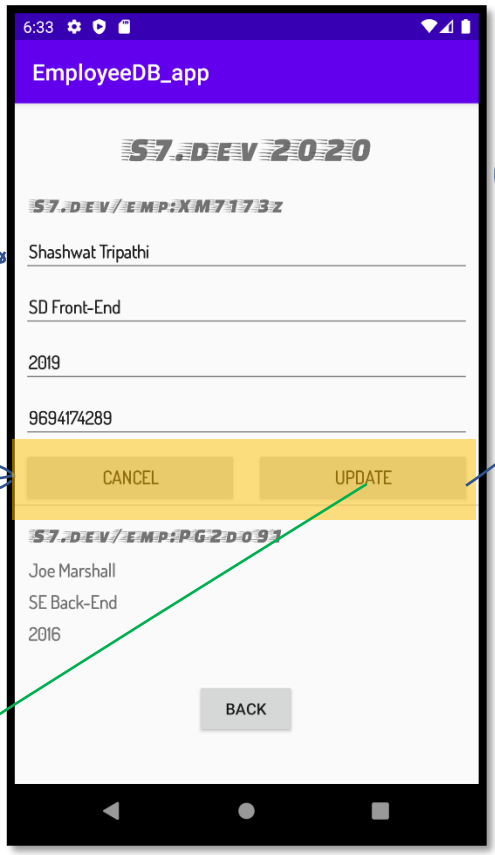
```
61.         return rowView;
62.     }
63. }
```

Snapshots:



Assignment 3: Develop an app for the demonstration of basic queries (create table, insert, select update, delete) in SQLite. Taking up a case of a company wishing to maintain records of their employees.





Buttons Changed