

COURSE OUTCOME - 5

PROGRAM NO-1

AIM:

Create a database using SQLite and perform INSERT and SELECT.

PROGRAM CODE:

Activity_main.xml	<pre><?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=".MainActivity"> <Button android:id="@+id/adduser" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="insertUser" android:text="Add User" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.526" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.455" /> <Button android:id="@+id/view" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="viewUser" android:text="View Users" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.528" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.787" /> <Button</pre>
-------------------	---

```
android:id="@+id/delete"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:onClick="deleteUser"
android:text="Delete User"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.528"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.664" />
```

<EditText

```
android:id="@+id/name"
android:layout_width="188dp"
android:layout_height="43dp"
android:ems="10"
android:hint="Name"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.578"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.174" />
```

<EditText

```
android:id="@+id/num"
android:layout_width="188dp"
android:layout_height="43dp"
android:ems="10"
android:hint="Mobile Number"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.578"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.354" />
```

<TextView

```
android:layout_width="319dp"
android:layout_height="27dp"
android:text="User Registration"
android:textAlignment="center"
android:textAllCaps="true"
```

	<pre> android:textStyle="bold" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintHorizontal_bias="0.629" app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.042" /> <EditText android:id="@+id/add" android:layout_width="188dp" android:layout_height="43dp" android:ems="10" android:hint="Address" android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.578" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.271" /> <Button android:id="@+id/update" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="updateUser" android:text="Update User Data" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.528" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.571" /> </androidx.constraintlayout.widget.ConstraintLayout> </pre>
MainActivity.java	<pre> package com.example.co5_q1insertread; import androidx.appcompat.app.AlertDialog; import androidx.appcompat.app.AppCompatActivity; import android.database.Cursor; import android.os.Bundle; import android.view.View; </pre>

```

import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText name, address, number;
    Button add, update, delete, viewuser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }
    public void insertUser(View view){
        name = findViewById(R.id.name);
        address = findViewById(R.id.add);
        number = findViewById(R.id.num);
        DBHelper DB = new DBHelper(this);
        String nam = name.getText().toString();
        String add = address.getText().toString();
        String num = number.getText().toString();
        Boolean checkinsert = DB.insertuserDB(nam,add,num);
        if(checkinsert){
            Toast.makeText(this, "User added Succesfully",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(this, "User not added",
Toast.LENGTH_SHORT).show();
        }
        name.setText("");
        address.setText("");
        number.setText("");
    }
    public void updateUser(View view){
        name = findViewById(R.id.name);
        address = findViewById(R.id.add);
        number = findViewById(R.id.num);
        DBHelper DB = new DBHelper(this);
        String nam = name.getText().toString();
        String add = address.getText().toString();
        String num = number.getText().toString();
        Boolean checkupdate = DB.UpdateuserDB(nam,add,num);
        if(checkupdate){
            Toast.makeText(this, "User data updated Succesfully",
Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

    }
    else{
        Toast.makeText(this, "User date not updated",
Toast.LENGTH_SHORT).show();
    }
    name.setText("");
    address.setText("");
    number.setText("");
}
public void deleteUser(View view){
    name = findViewById(R.id.name);
    DBHelper DB = new DBHelper(this);
    String nam = name.getText().toString();
    Boolean checkdelete= DB.DeleteuserDB(nam);
    if(checkdelete){
        Toast.makeText(this, "User deleted Succesfully",
Toast.LENGTH_SHORT).show();
    }
    else{
        Toast.makeText(this, "User not deleted",
Toast.LENGTH_SHORT).show();
    }
    name.setText("");
    address.setText("");
    number.setText("");
}
public void viewUser(View view){
    DBHelper DB = new DBHelper(this);
    Cursor res = DB.getuserDB();
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    if(res.getCount()==0){
        alert.setCancelable(true);
        alert.setTitle("Users Data");
        alert.setMessage("No user found");
        alert.show();
    }
    else {
        StringBuffer buffer = new StringBuffer();
        while (res.moveToNext()) {
            buffer.append("Name:" + res.getString(0) + "\n");
            buffer.append("Address:" + res.getString(1) + "\n");
            buffer.append("Mobile Number:" + res.getString(2) + "\n\n");
        }
        alert.setCancelable(true);
        alert.setTitle("Users Data");
        alert.setMessage(buffer.toString());
    }
}

```

	<pre> alert.show(); name.setText(""); address.setText(""); number.setText(""); } } } </pre>
DBHelper.java	<pre> package com.example.co5_q1insertread; 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; public class DBHelper extends SQLiteOpenHelper { public DBHelper(Context context) { super(context, "User.DB", null, 1); } @Override public void onCreate(SQLiteDatabase DB) { DB.execSQL("create table userdata(name Text primary key, address TEXT, number TEXT)"); } @Override public void onUpgrade(SQLiteDatabase DB, int i, int i1) { DB.execSQL("drop Table if exists userdata"); } public Boolean insertuserDB(String name,String address, String number){ SQLiteDatabase DB = this.getWritableDatabase(); ContentValues cv = new ContentValues(); cv.put("name",name); cv.put("address",address); cv.put("number",number); long result = DB.insert("userdata",null,cv); if(result==-1){ return false; } else{ return true; } } public Boolean UpdateuserDB(String name,String address, String </pre>

```

number){
    SQLiteDatabase DB = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("address",address);
    cv.put("number",number);
    Cursor cursor = DB.rawQuery("Select * from userdata where name
= ?",new String[]{name});
    if(cursor.getCount(>0){
        long result = DB.update("userdata",cv,"name=?",new
String[]{name});
        if(result==-1){
            return false;
        }
        else{
            return true;
        }
    }
    else{
        return false;
    }
}

public Boolean DeleteuserDB(String name){
    SQLiteDatabase DB = this.getWritableDatabase();
    Cursor cursor = DB.rawQuery("Select * from userdata where name
= ?",new String[]{name});
    if(cursor.getCount(>0){
        long result = DB.delete("userdata","name=?",new
String[]{name});
        if(result==-1){
            return false;
        }
        else{
            return true;
        }
    }
    else{
        return false;
    }
}

public Cursor getuserDB(){
    SQLiteDatabase DB = this.getWritableDatabase();
    Cursor cursor = DB.rawQuery("Select * from userdata",null);
    return cursor;
}
}

```


OUTPUT:

The screenshot shows a mobile application interface with a dark background. At the top, the title 'C05_Q1(INSERT&READ)' is displayed. Below it, the section 'USER REGISTRATION' is centered. There are four input fields: the first contains 'Rasika V V', the second contains 'rasi', and the third contains '13690063'. Below the input fields are four red buttons: 'ADD USER', 'UPDATE USER DATA', 'DELETE USER', and 'VIEW USERS'. The status bar at the top shows the time as 9:09 and battery level at 19%.

The screenshot shows a list of users under the heading 'Users Data'. Each user entry includes their Name, Address, and Mobile Number. The data is as follows:

Name	Address	Mobile Number
christy raj	Raj	1235689
Ananya B	ananya	4790542
Anushree	anu	096311
Fathima Novrin	novrii	1368900568
Anilamol Chacko	anilaa	3589521
Riya Vincent	riyaaa	524890
Rasika V V	rasi	13690063

The status bar at the top shows the time as 9:09 and battery level at 19%.

RESULT:

The program executed successfully and obtained the output.

PROGRAM NO - 2

AIM:

Perform UPDATE and DELETE on SQLite database.

PROGRAM CODE:

Activity_main.xml	<pre><?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=".MainActivity"> <Button android:id="@+id/adduser" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="insertUser" android:text="Add User" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.526" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.455" /> <Button android:id="@+id/view" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="viewUser" android:text="View Users" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.528" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.787" /></pre>
-------------------	---

```
<Button
    android:id="@+id/delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="deleteUser"
    android:text="Delete User"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.528"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.664" />
```

```
<EditText
    android:id="@+id/name"
    android:layout_width="188dp"
    android:layout_height="43dp"
    android:ems="10"
    android:hint="Name"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.578"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.174" />
```

```
<EditText
    android:id="@+id/num"
    android:layout_width="188dp"
    android:layout_height="43dp"
    android:ems="10"
    android:hint="Mobile Number"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.578"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.354" />
```

```
<TextView
    android:layout_width="319dp"
    android:layout_height="27dp"
    android:text="User Registration"
    android:textAlignment="center"
```

	<pre> android:textAllCaps="true" android:textStyle="bold" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintHorizontal_bias="0.629" app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.042" /> <EditText android:id="@+id/add" android:layout_width="188dp" android:layout_height="43dp" android:ems="10" android:hint="Address" android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.578" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.271" /> <Button android:id="@+id/update" android:layout_width="wrap_content" android:layout_height="wrap_content" android:onClick="updateUser" android:text="Update User Data" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.528" app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.571" /> </androidx.constraintlayout.widget.ConstraintLayout> </pre>
MainActivity.java	<pre> package com.example.co5_q2updatedelete; import androidx.appcompat.app.AlertDialog; import androidx.appcompat.app.AppCompatActivity; import android.database.Cursor; import android.os.Bundle; import android.view.View; import android.widget.Button; </pre>

```

import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    EditText name, address, number;
    Button add, update, delete, viewuser;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void insertUser(View view){
        name = findViewById(R.id.name);
        address = findViewById(R.id.add);
        number = findViewById(R.id.num);
        DBHelper DB = new DBHelper(this);
        String nam = name.getText().toString();
        String add = address.getText().toString();
        String num = number.getText().toString();
        Boolean checkinsert = DB.insertuserDB(nam,add,num);
        if(checkinsert){
            Toast.makeText(this, "User added Succesfully",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(this, "User not added",
Toast.LENGTH_SHORT).show();
        }
        name.setText("");
        address.setText("");
        number.setText("");
    }
    public void updateUser(View view){
        name = findViewById(R.id.name);
        address = findViewById(R.id.add);
        number = findViewById(R.id.num);
        DBHelper DB = new DBHelper(this);
        String nam = name.getText().toString();
        String add = address.getText().toString();
        String num = number.getText().toString();
        Boolean checkupdate = DB.UpdateuserDB(nam,add,num);
        if(checkupdate){
            Toast.makeText(this, "User data updated Succesfully",
Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(this, "User date not updated",

```

```

Toast.LENGTH_SHORT).show();
    }
    name.setText("");
    address.setText("");
    number.setText("");
}
public void deleteUser(View view){
    name = findViewById(R.id.name);
    DBHelper DB = new DBHelper(this);
    String nam = name.getText().toString();
    Boolean checkdelete= DB.DeleteuserDB(nam);
    if(checkdelete){
        Toast.makeText(this, "User deleted Succesfully",
Toast.LENGTH_SHORT).show();
    }
    else{
        Toast.makeText(this, "User not deleted",
Toast.LENGTH_SHORT).show();
    }
    name.setText("");
    address.setText("");
    number.setText("");
}
public void viewUser(View view){
    DBHelper DB = new DBHelper(this);
    Cursor res = DB.getuserDB();
    AlertDialog.Builder alert = new AlertDialog.Builder(this);
    if(res.getCount()==0){
        alert.setCancelable(true);
        alert.setTitle("Users Data");
        alert.setMessage("No user found");
        alert.show();
    }
    else {
        StringBuffer buffer = new StringBuffer();
        while (res.moveToNext()) {
            buffer.append("Name:" + res.getString(0) + "\n");
            buffer.append("Address:" + res.getString(1) + "\n");
            buffer.append("Mobile Number:" + res.getString(2) + "\n\n");
        }
        alert.setCancelable(true);
        alert.setTitle("Users Data");
        alert.setMessage(buffer.toString());
        alert.show();
        name.setText("");
        address.setText("");
    }
}

```

	<pre> number.setText(""); } } </pre>
DBHelper.java	<pre> package com.example.co5_q2updatedelete; 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; public class DBHelper extends SQLiteOpenHelper { public DBHelper(Context context) { super(context, "User.DB", null, 1); } @Override public void onCreate(SQLiteDatabase DB) { DB.execSQL("create table userdata(name Text primary key, address TEXT, number TEXT)"); } @Override public void onUpgrade(SQLiteDatabase DB, int i, int i1) { DB.execSQL("drop Table if exists userdata"); } public Boolean insertuserDB(String name,String address, String number){ SQLiteDatabase DB = this.getWritableDatabase(); ContentValues cv = new ContentValues(); cv.put("name",name); cv.put("address",address); cv.put("number",number); long result = DB.insert("userdata",null,cv); if(result==-1){ return false; } else{ return true; } } public Boolean UpdateuserDB(String name,String address, String number){ SQLiteDatabase DB = this.getWritableDatabase(); ContentValues cv = new ContentValues(); cv.put("address",address); </pre>

```

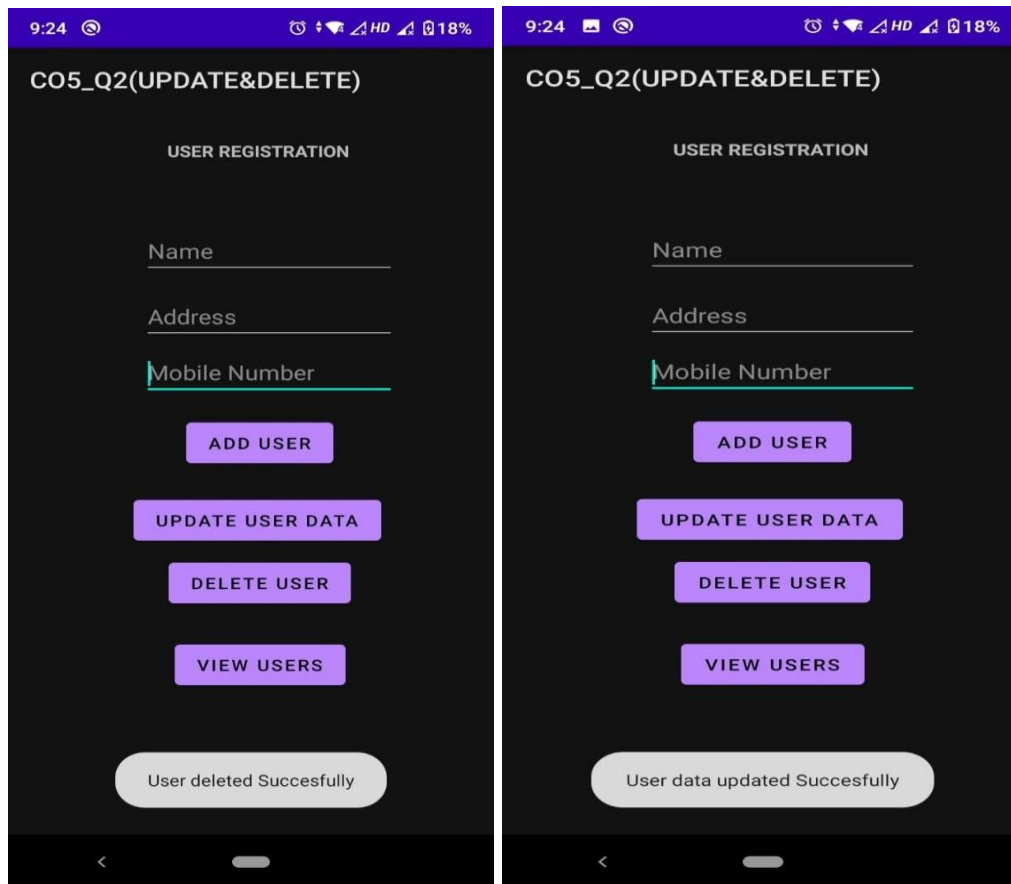
        cv.put("number",number);
        Cursor cursor = DB.rawQuery("Select * from userdata where name
= ?",new String[]{name});
        if(cursor.getCount()>0){
            long result = DB.update("userdata",cv,"name=?",new
String[]{name});
            if(result==-1){
                return false;
            }
            else{
                return true;
            }
        }
        else{
            return false;
        }
    }

    public Boolean DeleteuserDB(String name){
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor cursor = DB.rawQuery("Select * from userdata where name
= ?",new String[]{name});
        if(cursor.getCount()>0){
            long result = DB.delete("userdata","name=?",new
String[]{name});
            if(result==-1){
                return false;
            }
            else{
                return true;
            }
        }
        else{
            return false;
        }
    }

    public Cursor getuserDB(){
        SQLiteDatabase DB = this.getWritableDatabase();
        Cursor cursor = DB.rawQuery("Select * from userdata",null);
        return cursor;
    }
}

```


OUTPUT:



RESULT:

The program executed successfully and obtained the output.