	Name: Raghan Mahashnovi
	Roll ND: 53
	Panel: A
	Loub Assignment - 11 (JP)
	Android SQLite Connectivity CRUD of countions
<i>3</i>	To understand different CRUD operations. To learn Android application and dutubouse convertising.
	Regular Statement: Carato Simple database using Salite Jar ain Anderoid application and porjorum Thro Caracte, update, Read and solute) dotabase operations.
6	2010 Decorations on Doutabase:
	July William Land
	About Salite:
0	Solite is an open soroce son database that stoles date to took jie on a device Android comes with built - in

Spriteinplementation. Sprite supposite and substituted dutabase features. Main penhage is analysised dutabase.

Squite:

Squite database caraction:

In order to create a database you just march to and

The ocides to create a database, you just never to call the method ofon are Create Database with your dutabase.

name and mode as a parameter. It returns an instance of Soffite database which you have to receive in your own object. Syntax as Johans:

Stylite Database mydatubase = open or Create octabase ("your database - name", more PRIVATE, mill);

Solite dutabase Insection:

The con create table on insort data into table using over SBI mothod defined in SQLite Doctabour class. Syntax as form my destabling. esc SQL ("Create Table If NOT EXISTS);

STUDENTS (Userume VARCHAB, Passioned VARCHAR);

my database. exer SQL ("Insort Into STUDENTS VALUES

('admin', 'admin');");

Solite database. Reading:
we can retorierse anything form database using an
Object of Reverse class. we will call a method of
this class called resurgency and it will return
resulted with any pointing to table we can more
cursor jornand and reterieve data. Syntax as follows

Cheron Justy = mydiatabase - sun Query ("Splent Juan)
STODENTC" CHILL. STODENTS", mull); Stering wormanne = subulthot. got storing (0); Stering parropord = subulthot. get storing (1); scelet lot, more To First (); · Functions of Cureon Class This mothod returns total more columns of teable:

This mothod returns total more columns alone;

This mothod returns total more columns alone; This method acetions Index number of a columnin got Column Name (int column Index) This mothed sectuling name of column by specifying indep of column. in) got column Names U: This method rectures acrowy of all of column. y) got Count (): This method sectures total mos of score in cureson ui) got Position (): This method reducent current position of current table. Uii) in Closed (): This method sections Joure if aurior is closed and sietuems false otherwise. · Platform: Windows . Conclusion: Thus, we have teaunt to develop andereit application with connection to splite database.

I would a given to update Jolds of Salite database by taking input gram uses. And let's say, we have a table named Company which his Jibbs 10, Name, Ago, Added and paday. Thou by using following away we can ydate Addeds Jield you a customor whose ID is 6. UPDATE COMPANY SET Added - 'Puro' LIHERS FD=6 * Weile discourace blo Salite and MySal. And Salite Mysalice the Salite and MySal. - resulted by Dicherch Hip in - Developed by Oceasle in Maylage August 2000. - It was developed andy in C - It was developed in Cand language. - Deel not supposed xml yournot - Supposed xml Journat. - Deel not supposed xml yournot - Supposed a developed in Cand to sum, howe it is resurreable juntificing, how it forward to sum, howe it is resurreable juntificing, how if forward client / Resurrey conditionary conditionary conditionary. 3. White page to counter database applications in Andrewide August a Renaire Template, helect empty withing and click most.	11	
taking input Japan wear. And let's say, we have a table named Company which has yield 10. Name, Age, Adobe and palary. Then hy using following quosey we can yolate. Address field your customer whose ID is 6. UPDATE company SET Address - 'proo' LIHERS FOR=6 2. Weills disposeure blo SQLite and mysol. Am Solite Mysolite and mysol. -> reveloped by phichers Hip in -> Developed by accorde in Maylage. August 2000. -> It was developed only in C -> It was developed in Cand language. (tt. -> Are not suppost xml sounds -> Supposts xml Journat. -> Does not sequise a source -> Progrise a service Jour to sum, hours it is secureable functioning, hours it forward chient / source; to such the true of sum, hours it is secureable functioning, hours it forward. 3. white steps to create dolabous applications in Android. And -> I write a new servicet in Android studio with suitable range. -> Labert a servicet Template. Lete to entry activity and		FAG
has jells 10, Name, Age, Address and relating. Then by using pollowing growing we can update Address yield Joera customory whose ID is 6. UPDATE COMPANY SET Address - 'Pune' WHERE FRO=6 Name of Solite and my SQL. And Solite my SQL. - newlood by D. Licheus Hip in - Developed by Ociarle in Mayleans August 2000. - It was developed andy in C -> It was developed in Camp Language. - Deel not support xm Joemat -> Supports xm Journat. - Deel not support xm Joemat -> Supports xm Journat. - Deel not support xm Joemat -> Supports xm Journat. - Deel not support xm Joemat -> Supports xm Journat. - Deel not support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Journat. - Such support xm Joemat -> Supports xm Joe	1,	white a growy to update fields of SQLite database by taking input Jerom uses.
Am Solite -> reveloped by D. Richard Hip in -> Devoloped by Ocearle in May 1995 -> The was developed only in C -> It was developed in Court Language. -> Does not support xm Jormat> Supports xm Jormat. -> Does not support xm Jormat> Requires a servery for to sum, home it is servered by functioning, home it forward client / Resurses conditioning 3. White steps to create database applications in Androide Am -> Create or new Praject in Android Studio with puitable rame. -> Gelect a Rusied Template. Delect and activity and	Ans	by using following grovey we can indate Address field Joera customer whose ID is 6.
3. White steps to create database applications in Androide Ans > whate ar new period in Android Studio with suitable mane. Solect a persied Templote. Select empty activity and	Aws	SQLite Neveloped by D. Lichard Hip in -Developed by Daracle in May 1945. August 2000. It was developed any in C -> It was developed in Cand Language. Ctt. Does not supposit xml Journal -> Supposits xml Journal. Does not sequise a sauroy -> Requires a server Jour
	3. Aus	white steps to create database applications in Anderside Suretaine or new Revagest in Andersid Studio with suitable mane.

Juduse Chen opposations in young Java class by impositing such unit parkages and motheds.

Doeign young layout of app by making supperspeciate changes to young Activity. Xml file.

Justice young simulatory doubte and more you consum your capp on visitual mentions.

1) MainActivity.java Class

```
package com.example.exno5;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends Activity implements OnClickListener
  EditText Rollno, Name, Marks;
  Button Insert, Delete, Update, View, View All;
  SQLiteDatabase db;
  /** Called when the activity is first created. */
  @Override
  public void onCreate(Bundle savedInstanceState)
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity main);
    Rollno=(EditText)findViewById(R.id.Rollno);
    Name=(EditText)findViewById(R.id.Name);
    Marks=(EditText)findViewById(R.id.Marks);
    Insert=(Button)findViewById(R.id.Insert);
    Delete=(Button)findViewById(R.id.Delete);
    Update=(Button)findViewById(R.id.Update);
    View=(Button)findViewById(R.id.View);
    ViewAll=(Button)findViewById(R.id.ViewAll);
    Insert.setOnClickListener(this);
```

```
Delete.setOnClickListener(this);
    Update.setOnClickListener(this);
    View.setOnClickListener(this);
    ViewAll.setOnClickListener(this);
    // Creating database and table
    db=openOrCreateDatabase("StudentDB", Context.MODE PRIVATE,
null);
    db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno
VARCHAR, name VARCHAR, marks VARCHAR);");
  public void onClick(View view)
    // Inserting a record to the Student table
    if(view==Insert)
      // Checking for empty fields
      if(Rollno.getText().toString().trim().length()==0||
           Name.getText().toString().trim().length()==0||
           Marks.getText().toString().trim().length()==0)
      {
        showMessage("Error", "Please enter all values");
         return;
      }
      db.execSQL("INSERT INTO student
VALUES(""+Rollno.getText()+"",""+Name.getText()+
           "','"+Marks.getText()+"');");
      showMessage("Success", "Record added");
      clearText();
    }
    // Deleting a record from the Student table
    if(view==Delete)
    {
      // Checking for empty roll number
      if(Rollno.getText().toString().trim().length()==0)
      {
         showMessage("Error", "Please enter Rollno");
```

```
return;
      }
      Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
      if(c.moveToFirst())
        db.execSQL("DELETE FROM student WHERE
rollno='"+Rollno.getText()+"'");
        showMessage("Success", "Record Deleted");
      }
      else
        showMessage("Error", "Invalid Rollno");
      clearText();
    }
    // Updating a record in the Student table
    if(view==Update)
    {
      // Checking for empty roll number
      if(Rollno.getText().toString().trim().length()==0)
        showMessage("Error", "Please enter Rollno");
        return;
      }
      Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
      if(c.moveToFirst()) {
        db.execSQL("UPDATE student SET name="" + Name.getText() +
"',marks="" + Marks.getText() +
             "' WHERE rollno='"+Rollno.getText()+"");
        showMessage("Success", "Record Modified");
      }
      else {
        showMessage("Error", "Invalid Rollno");
      clearText();
```

```
}
    // Display a record from the Student table
    if(view==View)
      // Checking for empty roll number
      if(Rollno.getText().toString().trim().length()==0)
        showMessage("Error", "Please enter Rollno");
        return;
      Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno=""+Rollno.getText()+""", null);
      if(c.moveToFirst())
      {
        Name.setText(c.getString(1));
        Marks.setText(c.getString(2));
      }
      else
        showMessage("Error", "Invalid Rollno");
        clearText();
      }
    // Displaying all the records
    if(view==ViewAll)
    {
      Cursor c=db.rawQuery("SELECT * FROM student", null);
      if(c.getCount()==0)
        showMessage("Error", "No records found");
        return;
      }
      StringBuffer buffer=new StringBuffer();
      while(c.moveToNext())
        buffer.append("Rollno: "+c.getString(0)+"\n");
        buffer.append("Name: "+c.getString(1)+"\n");
```

```
buffer.append("Marks: "+c.getString(2)+"\n\n");
      }
      showMessage("Student Details", buffer.toString());
    }
  }
  public void showMessage(String title,String message)
    Builder builder=new Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
  }
  public void clearText()
  {
    Rollno.setText("");
    Name.setText("");
    Marks.setText("");
    Rollno.requestFocus();
  }
}
```

2) Activity.xml file code:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="50dp"
      android:layout_y="20dp"
      android:text="Student Details"
      android:textSize="30sp" />
```

```
<TextView
 android:layout width="wrap content"
 android:layout_height="wrap_content"
 android:layout x="20dp"
 android:layout y="110dp"
 android:text="Enter Rollno:"
 android:textSize="20sp" />
<EditText
 android:id="@+id/Rollno"
 android:layout width="150dp"
 android:layout height="wrap content"
 android:layout x="175dp"
 android:layout y="100dp"
 android:inputType="number"
 android:textSize="20sp" />
<TextView
 android:layout width="wrap content"
 android:layout_height="wrap_content"
 android:layout x="20dp"
 android:layout_y="160dp"
 android:text="Enter Name:"
 android:textSize="20sp" />
<EditText
 android:id="@+id/Name"
 android:layout width="150dp"
 android:layout height="wrap content"
 android:layout x="175dp"
 android:layout y="150dp"
 android:inputType="text"
 android:textSize="20sp" />
<TextView
 android:layout width="wrap content"
```

```
android:layout_height="wrap_content"
 android:layout x="20dp"
 android:layout y="210dp"
 android:text="Enter Marks:"
 android:textSize="20sp" />
<EditText
 android:id="@+id/Marks"
 android:layout width="150dp"
 android:layout height="wrap content"
 android:layout x="175dp"
 android:layout y="200dp"
 android:inputType="number"
 android:textSize="20sp" />
<Button
 android:id="@+id/Insert"
 android:layout width="150dp"
 android:layout height="wrap content"
 android:layout x="25dp"
 android:layout y="300dp"
 android:text="Insert"
 android:textSize="30dp" />
<Button
 android:id="@+id/Delete"
 android:layout width="150dp"
 android:layout height="wrap content"
 android:layout x="200dp"
 android:layout y="300dp"
 android:text="Delete"
 android:textSize="30dp" />
<Button
 android:id="@+id/Update"
 android:layout_width="150dp"
 android:layout height="wrap content"
```

```
android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />
  <Button
    android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout x="200dp"
    android:layout_y="400dp"
    android:text="View"
    android:textSize="30dp" />
  <Button
    android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout height="wrap content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="View All"
    android:textSize="30dp" />
</AbsoluteLayout>
```

Output:



















