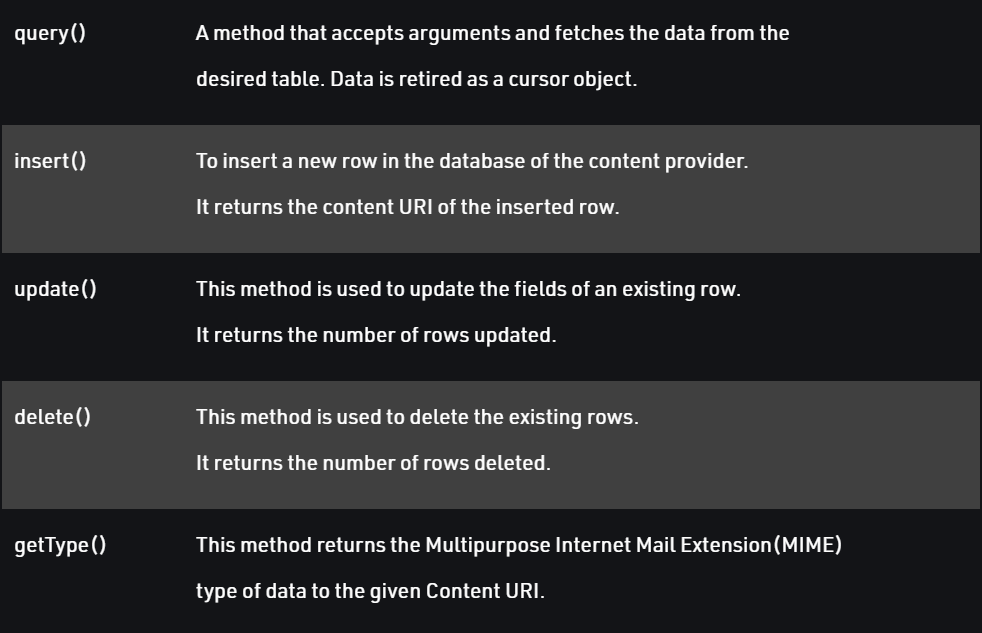
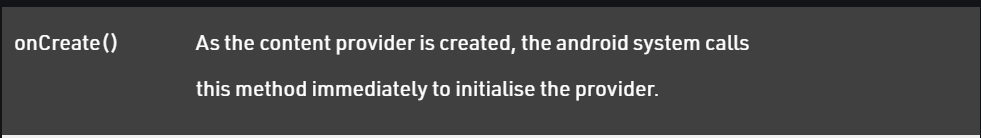
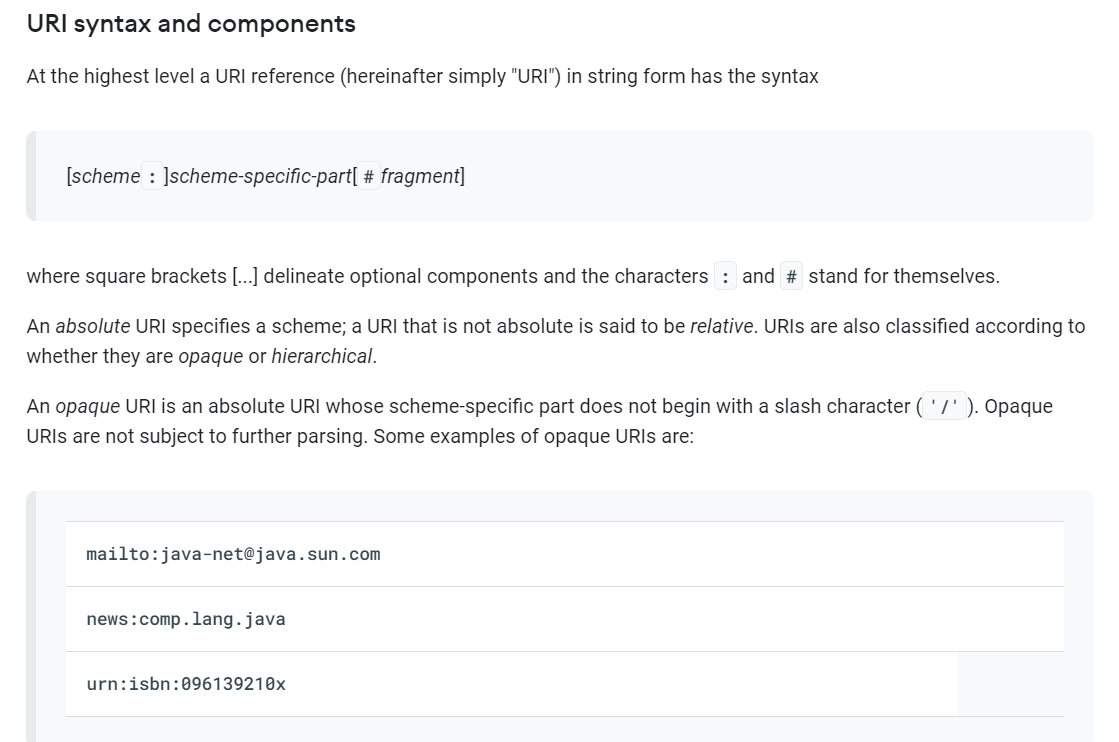
**Practical Related Questions**

1. **Write in detail which methods are needed to implement Content Provider class.**

Following are the six abstract methods and their description which are essential to override as the part of ContenProvider class:



1. **Explain different parts of an URI in android application. Also write the format of URI.**



1. **Write steps to create a content provider in android applications.**
2. Create a class that extends ContentProvider.
3. Create a contract class.
4. Create the UriMatcher definition.
5. Implement the onCreate() method.
6. Implement the getType() method.
7. Implement the CRUD methods.
8. Add the content provider to your AndroidManifest. xml.

**Exercise**

**MainActivity.java**

package com.example.contentprovidereg;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.view.inputmethod.InputMethodManager;  
import android.widget.CursorAdapter;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
  
 @Override  
 public boolean onTouchEvent(MotionEvent event){  
 InputMethodManager imm = (InputMethodManager)getSystemService(Context.*INPUT\_METHOD\_SERVICE*);  
 imm.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), 0);  
 return true;  
 }  
  
 public void onClickAddDetails(View view){  
 ContentValues values = new ContentValues();  
  
 values.put(MyContentProvider.*name*, ((EditText) findViewById(R.id.*textName*)).getText().toString());  
  
 getContentResolver().insert(MyContentProvider.*CONTENT\_URI*, values);  
  
 Toast.*makeText*(getBaseContext(), "New Record Inserted", Toast.*LENGTH\_LONG*).show();  
 }  
  
 public void onClickShowDetails(View view){  
 TextView result = (TextView) findViewById(R.id.*res*);  
  
 Cursor c = getContentResolver().query(Uri.*parse*("content://com.demo.user.provider/users"), null, null, null, null);  
  
  
 if(c.moveToFirst()){  
 StringBuilder str = new StringBuilder();  
 while(!c.isAfterLast()){  
 str.append("\n" + c.getString(c.getColumnIndex("id")) + ":- " + c.getString(c.getColumnIndex("name")));  
 c.moveToNext();  
 }  
 result.setText(str);  
 }  
 else{  
 result.setText("No Records Found");  
 }  
 }  
}

**MyContentProvider.java**

package com.example.contentprovidereg;

import android.content.ContentProvider;

import android.content.ContentUris;

import android.content.ContentValues;

import android.content.Context;

import android.content.UriMatcher;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteException;

import android.database.sqlite.SQLiteOpenHelper;

import android.database.sqlite.SQLiteQueryBuilder;

import android.net.Uri;

import java.util.HashMap;

public class MyContentProvider extends ContentProvider {

    public MyContentProvider() {

    }

    static final String PROVIDER\_NAME = "com.demo.user.provider";

    static final String URL = "content://" + PROVIDER\_NAME + "/users";

    static final Uri CONTENT\_URI = Uri.parse(URL);

    static final String id = "id";

    static final String name = "name";

    static final int uriCode = 1;

    static final UriMatcher uriMatcher;

    private static HashMap<String, String> values;

    static {

*// to match the content URI*

*// every time user access table under content provider*

        uriMatcher = new UriMatcher(UriMatcher.NO\_MATCH);

*// to access whole table*

        uriMatcher.addURI(PROVIDER\_NAME, "users", uriCode);

*// to access a particular row*

*// of the table*

        uriMatcher.addURI(PROVIDER\_NAME, "users/\*", uriCode);

    }

    @Override

    public int delete(Uri uri, String selection, String[] selectionArgs) {

*// Implement this to handle requests to delete one or more rows.*

        throw new UnsupportedOperationException("Not yet implemented");

    }

    @Override

    public String getType(Uri uri) {

        switch(uriMatcher.match(uri)){

            case uriCode:

                return "vnd.android.cursor.dir/users";

            default:

                throw new IllegalArgumentException("Unsupported URI: " + uri);

        }

    }

    @Override

    public Uri insert(Uri uri, ContentValues values) {

        long rowID = db.insert(TABLE\_NAME, "", values);

        if(rowID > 0){

            Uri \_uri = ContentUris.withAppendedId(CONTENT\_URI, rowID);

            getContext().getContentResolver().notifyChange(\_uri, null);

            return \_uri;

        }

        throw new SQLiteException("Failed to add a record into " + uri);

    }

    @Override

    public boolean onCreate() {

        Context context = getContext();

        DatabaseHelper dbHelper = new DatabaseHelper(context);

        db = dbHelper.getWritableDatabase();

        if(db != null){

            return true;

        }

        return false;

    }

    @Override

    public Cursor query(Uri uri, String[] projection, String selection,

                        String[] selectionArgs, String sortOrder) {

        SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

        qb.setTables(TABLE\_NAME);

        switch (uriMatcher.match(uri)){

            case uriCode:

                qb.setProjectionMap(values);

                break;

            default:

                throw new IllegalArgumentException("Unknown URL: " + uri);

        }

        if(sortOrder == null || sortOrder == ""){

            sortOrder = id;

        }

        Cursor c = qb.query(db, projection, selection, selectionArgs, null, null, sortOrder);

        c.setNotificationUri(getContext().getContentResolver(), uri);

        return c;

    }

    @Override

    public int update(Uri uri, ContentValues values, String selection,

                      String[] selectionArgs) {

*// TODO: Implement this to handle requests to update one or more rows.*

        throw new UnsupportedOperationException("Not yet implemented");

    }

*// creating object of database*

*// to perform query*

    private SQLiteDatabase db;

*// declaring name of the database*

    static final String DATABASE\_NAME = "UserDB";

*// declaring table name of the database*

    static final String TABLE\_NAME = "Users";

*// declaring version of the database*

    static final int DATABASE\_VERSION = 1;

*// sql query to create the table*

    static final String CREATE\_DB\_TABLE = " CREATE TABLE " + TABLE\_NAME

            + " (id INTEGER PRIMARY KEY AUTOINCREMENT, "

            + " name TEXT NOT NULL);";

    private static class DatabaseHelper extends SQLiteOpenHelper {

        DatabaseHelper(Context context){

*super*(context, DATABASE\_NAME, null, DATABASE\_VERSION);

        }

        @Override

        public void onCreate(SQLiteDatabase db){

            db.execSQL(CREATE\_DB\_TABLE);

        }

        @Override

        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){

            db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

            onCreate(db);

        }

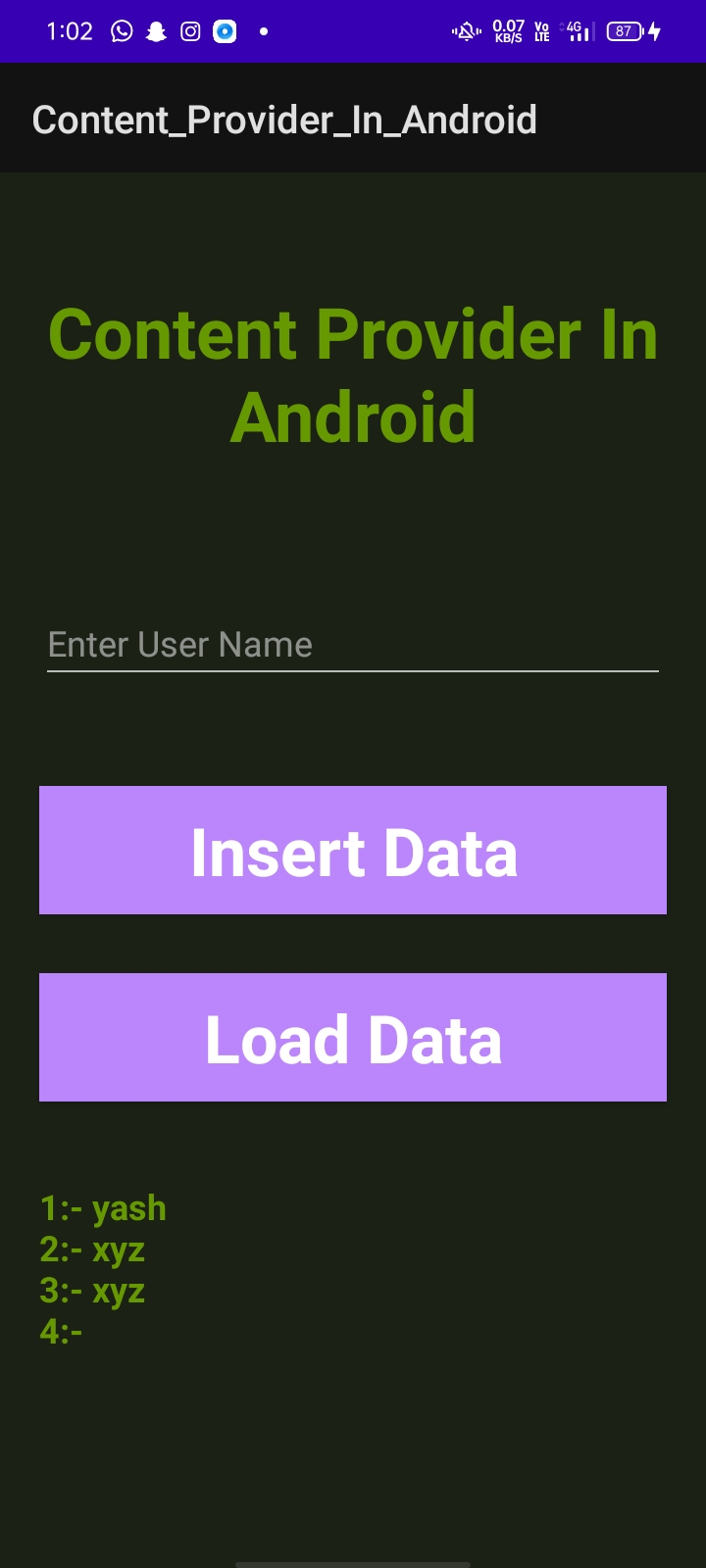
    }

}

**activity\_main.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"  
 android:background="#168BC34A"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:id="@+id/linearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerVertical="true"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toTopOf="@+id/imageView"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.13"  
 tools:ignore="MissingConstraints">  
  
 <TextView  
 android:id="@+id/textView1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="40dp"  
 android:layout\_marginBottom="70dp"  
 android:text="@string/heading"  
 android:textAlignment="center"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textColor="@android:color/holo\_green\_dark"  
 android:textSize="36sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/textName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="40dp"  
 android:hint="@string/hintText" />  
  
 <Button  
 android:id="@+id/insertButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickAddDetails"  
 android:text="@string/insertButtontext"  
 android:textAlignment="center"  
 android:textAppearance="@style/TextAppearance.AppCompat.Display1"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
  
 <Button  
 android:id="@+id/loadButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:background="#4CAF50"  
 android:onClick="onClickShowDetails"  
 android:text="@string/loadButtonText"  
 android:textAlignment="center"  
 android:textAppearance="@style/TextAppearance.AppCompat.Display1"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/res"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:clickable="false"  
 android:ems="10"  
 android:textColor="@android:color/holo\_green\_dark"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <ListView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/liv">  
 </ListView>  
  
 </LinearLayout>  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Output**

****