Practical no: 19

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:

query()	A method that accepts arguments and fetches the data from the desired table. Data is retired as a cursor object.
insert()	To insert a new row in the database of the content provider. It returns the content URI of the inserted row.
update()	This method is used to update the fields of an existing row. It returns the number of rows updated.
delete()	This method is used to delete the existing rows. It returns the number of rows deleted.
getType()	This method returns the Multipurpose Internet Mail Extension(MIME) type of data to the given Content URI.
onCreate()	As the content provider is created, the android system calls this method immediately to initialise the provider.

Practical no: 19

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

URI syntax and components

At the highest level a URI reference (hereinafter simply "URI") in string form has the syntax

```
[scheme :]scheme-specific-part[ # fragment]
```

where square brackets [...] delineate optional components and the characters `:` and `#` stand for themselves.

An *absolute* URI specifies a scheme; a URI that is not absolute is said to be *relative*. URIs are also classified according to whether they are *opaque* or *hierarchical*.

An *opaque* URI is an absolute URI whose scheme-specific part does not begin with a slash character (`'/'`). Opaque URIs are not subject to further parsing. Some examples of opaque URIs are:

```
mailto:java-net@java.sun.com
news:comp.lang.java
urn:isbn:096139210x
```

3. Write steps to create a content provider in android applications.

a) Create a class that extends ContentProvider.
b) Create a contract class.
c) Create the UriMatcher definition.
d) Implement the onCreate() method.
e) Implement the getType() method.
f) Implement the CRUD methods.
g) Add the content provider to your AndroidManifest. xml.

Practical no: 19

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);
```

Practical no: 19

```
        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);
```

Practical no: 19

```
    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);
```

Practical no: 19

```
    }

    @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
```

Practical no: 19

```
    // 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">
```

Practical no: 19

```
    <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"
```

Practical no: 19

```
            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>
```

Practical no: 19

Output