

Experiment - 7

Name: Ansari Ushair

Roll no : 14DCO53

Class : BE.CO

Batch : 03

Aim : Develop an application that makes use of database

#Theory

➤ What is data? And database?

In simple words data can be facts related to any object in consideration. For example, our name, age, height, weight, etc are some data related to us. A picture, image, file, pdf etc can also be considered data.

Database is a systematic collection of data. Databases support storage and manipulation of data. Databases make data management easy.

A student record management would definitely use database to store data pertaining to students, phone numbers, other contact details, etc.

➤ SQL

Structured Query Language or SQL is a standard Database language which is used to create, maintain and retrieve the data from relational databases like MySQL, Oracle, SQL Server, PostgreSQL, etc. The recent ISO standard version of SQL is SQL:2019.

As the name suggests, it is used when we have structured data (in the form of tables). All databases that are not relational (or do not use fixed structure tables to store data) and therefore do not use SQL, are called NoSQL databases. Examples of NoSQL are MongoDB, DynamoDB, Cassandra, etc.

➤ SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

➤ Classes & API's

SQLiteOpenHelper class - It provides the functionality to use the SQLite database. It is used for database creation and version management. We provide the implementation of onCreate() and onUpgrade() methods of SQLiteOpenHelper class to perform any function.

Constructors:

Constructor	Description
SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version)	creates an object for creating, opening and managing the database
SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version, DatabaseErrorHandler errorHandler)	creates an object for creating, opening and managing the database. It specifies the error handler

Methods:

Method	Description
public abstract void onCreate(SQLiteDatabase db)	called only once when database is created for the first time.
public abstract void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)	called when database needs to be upgraded.
public synchronized void close ()	closes the database object.
public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion)	called when database needs to be downgraded.

SQLiteDatabase class - It contains methods to be performed on sqlite database such as create, update, delete, select etc.

Methods:

Method	Description
void execSQL(String sql)	executes the sql query not select query.
long insert(String table, String nullColumnHack, ContentValues values)	inserts a record on the database. The table specifies the table name, nullColumnHack doesn't allow completely null values. If second argument is null, android will store null values if values are empty. The third argument specifies the values to be stored.
int update(String table, ContentValues values, String whereClause, String[] whereArgs)	updates a row.
Cursor query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)	returns a cursor over the resultset.

Program:

MainActivity.java

```
package com.example.exp7;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    EditText editId,editName,editEmail,editClass;
    Button bAdd,bUpdate,bDelete;
    ListView lstPerson;

    List<Person> data = new ArrayList<>();
    DatabaseHelper db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        db = new DatabaseHelper(this);

        bAdd = findViewById(R.id.btnAdd);
        bUpdate= findViewById(R.id.btnUpdate);
        bDelete= findViewById(R.id.btnDelete);
        lstPerson =findViewById(R.id.list);

        editId= findViewById(R.id.editId);
        editName= findViewById(R.id.editName);
        editEmail= findViewById(R.id.editEmail);
        editClass= findViewById(R.id.editClass);

        refreshData();

        bAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Person person = new
                Person(Integer.parseInt(editId.getText().toString()),editName.getText().toString(),editEmail.getText().toString()
,editClass.getText().toString());
                db.addPerson(person);
                refreshData();
            }
        });
    }
}
```

```

        bUpdate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Person person = new
Person(Integer.parseInt(editId.getText().toString()),editName.getText().toString(),editEmail.getText().toString()
,editClass.getText().toString());
                db.updatePerson(person);
                refreshData();
            }
        });
        bDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Person person = new
Person(Integer.parseInt(editId.getText().toString()),editName.getText().toString(),editEmail.getText().toString()
,editClass.getText().toString());
                db.deletePerson(person);
                refreshData();
            }
        });
    }

    private void refreshData(){
        data=db.getAllPerson();
        PersonAdapter adapter = new PersonAdapter(MainActivity.this,data,editId,editName,editEmail,editClass);
        lstPerson.setAdapter(adapter);
    }
}

```

Person.java

```

package com.example.exp7;

public class Person {
    private int Id;
    private String name,email,sclass;

    public Person() {
    }

    public Person(int id, String name, String email, String sclass) {
        Id = id;
        this.name = name;
        this.email = email;
        this.sclass = sclass;
    }

    public int getId() {
        return Id;
    }

    public void setId(int id) {
        Id = id;
    }

    public String getName() {
        return name;
    }
}

```

```

    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getSclass() {
        return sclass;
    }

    public void setSclass(String sclass) {
        this.sclass = sclass;
    }
}

```

PersonAdapter.java

```
package com.example.exp7;
```

```

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.TextView;

```

```
import java.util.List;
```

```

public class PersonAdapter extends BaseAdapter {
    Activity activity;
    List<Person> lstPersons;
    LayoutInflater inflater;
    EditText editId,editName,editEmail,editClass;

```

```

    public PersonAdapter(Activity activity, List<Person> lstPersons, EditText editId, EditText editName,
    EditText editEmail, EditText editClass) {
        this.activity = activity;
        this.lstPersons = lstPersons;
        inflater =(LayoutInflater)activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        this.editId = editId;
        this.editName = editName;
        this.editEmail = editEmail;
        this.editClass = editClass;
    }

```

```
@Override
```

```

public int getCount() {
    return lstPersons.size();
}

```

```

    }

    @Override
    public Object getItem(int i) {
        return lstPersons.get(i);
    }

    @Override
    public long getItemId(int i) {
        return lstPersons.get(i).getId();
    }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        View rowView;
        rowView=inflater.inflate(R.layout.row,null);
        final TextView txtRowId,txtRowName,txtRowEmail,txtRowClass;
        txtRowId=(TextView)rowView.findViewById(R.id.rowId);
        txtRowName=(TextView)rowView.findViewById(R.id.txtRowName);
        txtRowEmail=(TextView)rowView.findViewById(R.id.txtEmail);
        txtRowClass=(TextView)rowView.findViewById(R.id.txtClass);

        txtRowId.setText(""+lstPersons.get(i).getId());
        txtRowName.setText(""+lstPersons.get(i).getName());
        txtRowEmail.setText(""+lstPersons.get(i).getEmail());
        txtRowClass.setText(""+lstPersons.get(i).getSclass());

        rowView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                editId.setText(""+txtRowId.getText());
                editName.setText(""+txtRowName.getText());
                editEmail.setText(""+txtRowEmail.getText());
                editClass.setText(""+txtRowClass.getText());
            }
        });
        return rowView;
    }
}

```

DatabaseHelper.java

```

package com.example.exp7;

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;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {

```

```

private static final int DATABASE_VER=1;
private static final String DATABASE_NAME="EDMTDEV";
//
private static final String TABLE_NAME="Students";
private static final String KEY_ID="Id";
private static final String KEY_NAME="Name";
private static final String KEY_EMAIL="Email";
private static final String KEY_CLASS="Class";

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VER);
}

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
    String CREATE_TABLE="CREATE TABLE "+TABLE_NAME+" ("
        + KEY_ID+" INTEGER PRIMARY KEY,"+KEY_NAME+" Text,"
        +KEY_EMAIL+" TEXT,"+KEY_CLASS+" TEXT"+" ) ";
    sqLiteDatabase.execSQL(CREATE_TABLE);
}

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    sqLiteDatabase.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
    onCreate(sqLiteDatabase);
}

public void addPerson(Person person)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NAME,person.getName());
    values.put(KEY_EMAIL,person.getEmail());
    values.put(KEY_CLASS,person.getSclass());
    db.insert(TABLE_NAME,null,values);
    db.close();
}

public int updatePerson(Person person)
{
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values= new ContentValues();
    values.put(KEY_NAME,person.getName());
    values.put(KEY_EMAIL,person.getEmail());
    values.put(KEY_CLASS,person.getSclass());

    return db.update(TABLE_NAME,values,KEY_ID+"=?",new String[] {String.valueOf(person.getId())});
}

public void deletePerson(Person person)
{
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_NAME,KEY_ID+"=?",new String[] {String.valueOf(person.getId())});
    db.close();
}

```

```

public Person getPerson(int id){
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE_NAME,new
String[] {KEY_ID,KEY_NAME,KEY_EMAIL,KEY_CLASS},KEY_ID+"=?",
    new String[] {String.valueOf(id)},null,null,null,null);
    if(cursor != null)
        cursor.moveToFirst();
    return new Person(cursor.getInt(0),cursor.getString(1),cursor.getString(2),cursor.getString(3));
}

public List<Person> getAllPerson() {
    List<Person> lstPersons = new ArrayList<>();
    String selectQuery = "SELECT * FROM " + TABLE_NAME;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    if (cursor.moveToFirst()) {
        do {
            Person person = new Person();
            person.setId(cursor.getInt(0));
            person.setName(cursor.getString(1));
            person.setEmail(cursor.getString(2));
            person.setSclass(cursor.getString(3));

            lstPersons.add(person);
        }
        while (cursor.moveToNext());
    }
    return lstPersons;
}
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Student"
        android:layout_gravity="center_horizontal"
        android:textSize="22sp"
        android:textStyle="bold"
        android:layout_marginTop="10dp"
        />

    <EditText
        android:id="@+id/editId"
        android:layout_width="match_parent"

```



```

        android:layout_height="wrap_content"
        android:hint="Id "
    />
    <EditText
        android:id="@+id/editName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name "
    />
    <EditText
        android:id="@+id/editEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email " />

    <EditText
        android:id="@+id/editClass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Class "
    />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/btnAdd"
            android:text="Add"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>

        <Button
            android:id="@+id/btnUpdate"
            android:text="Update"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>

        <Button
            android:id="@+id/btnDelete"
            android:text="Delete"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"/>

    </LinearLayout>
    <ListView
        android:id="@+id/list"
        android:choiceMode="singleChoice"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></ListView>
</LinearLayout>

```

row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/rowId"
        android:textStyle="bold"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="1"
        android:textSize="40sp"
        android:layout_marginLeft="10dp"
    />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/txtRowName"
            android:text="Name"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>

        <TextView
            android:id="@+id/txtEmail"
            android:text="Email"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>

        <TextView
            android:id="@+id/txtClass"
            android:text="Class"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"/>

    </LinearLayout>

</LinearLayout>
```

Output:

2:22

Exp7

Student

1

Ushair Ansari

ushairansari@gmail.com

BECO

ADDUPDATEDELETE

2:20

Exp7

Student

Id

Name

Email

Class

ADDUPDATEDELETE

1

Ushair Ansari

ushairansari@gmail.com

BECO

2

Danish Khan

danishkhan1812@gmail.com

BECO