**Practical No. 13**

**Aim: Create an android app to save user data in a database and use of different queries.**

**MainActivity.java**

package com.example.dbase;  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.List;  
public class MainActivity extends AppCompatActivity {  
 private EditText editTextName, editTextEmail, editTextUserId;  
 private TextView textViewData;  
 private DatabaseHelper databaseHelper;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 editTextName = findViewById(R.id.*editTextName*);  
 editTextEmail = findViewById(R.id.*editTextEmail*);  
 editTextUserId = findViewById(R.id.*editTextUserId*);  
 Button btnInsert = findViewById(R.id.*btnInsert*);  
 Button btnRetrieveAll = findViewById(R.id.*btnRetrieveAll*);  
 Button btnRetrieveById = findViewById(R.id.*btnRetrieveById*);  
 textViewData = findViewById(R.id.*textViewData*);  
 databaseHelper = new DatabaseHelper(this);  
 btnInsert.setOnClickListener(v -> insertUserData());  
 btnRetrieveAll.setOnClickListener(v -> retrieveAllUsers());  
 btnRetrieveById.setOnClickListener(v -> retrieveUserById());  
 }  
 private void insertUserData() {  
 String name = editTextName.getText().toString().trim();  
 String email = editTextEmail.getText().toString().trim();  
 long userId = databaseHelper.insertUser(name, email);  
 Log.*d*("UserData", "User inserted with ID: " + userId);  
 }  
 private void retrieveAllUsers() {  
 List<User> allUsers = databaseHelper.getAllUsers();  
 StringBuilder userData = new StringBuilder("All Users:\n");  
 for (User user : allUsers) {  
 userData.append("ID: ").append(user.getId()).append(", Name: ").append(user.getName())  
 .append(", Email: ").append(user.getEmail()).append("\n");  
 }  
 textViewData.setText(userData.toString());  
 }  
 @SuppressLint("SetTextI18n")  
 private void retrieveUserById() {  
 String userIdString = editTextUserId.getText().toString().trim();  
 if (!userIdString.isEmpty()) {  
 long userId = Long.*parseLong*(userIdString);  
 User user = databaseHelper.getUserById(userId);  
 if (user != null) {  
 textViewData.setText("User found by ID:\n" + "ID: " + user.getId() + "\nName: " + user.getName() + "\nEmail: " + user.getEmail());  
 } else {  
 textViewData.setText("User not found by ID: " + userId);  
 }  
 } else {  
 textViewData.setText("Please enter a user ID.");  
 }  
 }  
}

**DatabaseHelper.java**

package com.example.dbase;  
import android.annotation.SuppressLint;  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
import java.util.ArrayList;  
import java.util.List;  
class DatabaseHelper extends SQLiteOpenHelper {  
 private static final String *DATABASE\_NAME* = "user\_data.db";  
 private static final int *DATABASE\_VERSION* = 1;  
 public static final String *TABLE\_USERS* = "users";  
 public static final String *COLUMN\_ID* = "id";  
 public static final String *COLUMN\_NAME* = "name";  
 public static final String *COLUMN\_EMAIL* = "email";  
 private static final String *TABLE\_CREATE* =  
 "CREATE TABLE " + *TABLE\_USERS* + " (" + *COLUMN\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " + *COLUMN\_NAME* + " TEXT, " + *COLUMN\_EMAIL* + " TEXT);";  
 public DatabaseHelper(Context context) {  
 super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);  
 }  
 @Override  
 public void onCreate(SQLiteDatabase db) {  
 db.execSQL(*TABLE\_CREATE*);  
 }  
 @Override  
 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE\_USERS*);  
 onCreate(db);  
 }  
 public long insertUser(String name, String email) {  
 SQLiteDatabase db = this.getWritableDatabase();  
 ContentValues values = new ContentValues();  
 values.put(*COLUMN\_NAME*, name);  
 values.put(*COLUMN\_EMAIL*, email);  
 long userId = db.insert(*TABLE\_USERS*, null, values);  
 db.close();  
 return userId;  
 }  
 @SuppressLint("Range")  
 public List<User> getAllUsers() {  
 List<User> users = new ArrayList<>();  
 String selectQuery = "SELECT \* FROM " + *TABLE\_USERS*;  
 SQLiteDatabase db = this.getWritableDatabase();  
 Cursor cursor = db.rawQuery(selectQuery, null);  
 if (cursor != null && cursor.moveToFirst()) {  
 do {  
 User user = new User();  
 user.setId(cursor.getLong(cursor.getColumnIndex(*COLUMN\_ID*)));  
 user.setName(cursor.getString(cursor.getColumnIndex(*COLUMN\_NAME*)));  
 user.setEmail(cursor.getString(cursor.getColumnIndex(*COLUMN\_EMAIL*)));  
 users.add(user);  
 } while (cursor.moveToNext());  
 cursor.close();  
 }  
 db.close();  
 return users;  
 }  
 @SuppressLint("Range")  
 public User getUserById(long userId) {  
 SQLiteDatabase db = this.getReadableDatabase();  
 Cursor cursor = db.query(  
 *TABLE\_USERS*,  
 new String[]{*COLUMN\_ID*, *COLUMN\_NAME*, *COLUMN\_EMAIL*},  
 *COLUMN\_ID* + "=?",  
 new String[]{String.*valueOf*(userId)},  
 null,  
 null,  
 null,  
 null  
 );  
 User user = null;  
 if (cursor != null && cursor.moveToFirst()) {  
 user = new User();  
 user.setId(cursor.getLong(cursor.getColumnIndex(*COLUMN\_ID*)));  
 user.setName(cursor.getString(cursor.getColumnIndex(*COLUMN\_NAME*)));  
 user.setEmail(cursor.getString(cursor.getColumnIndex(*COLUMN\_EMAIL*)));  
 cursor.close();  
 }  
 db.close();  
 return user;  
 }  
}

**User.java**

package com.example.dbase;  
public class User {  
 private long id;  
 private String name;  
 private String email;  
 public User() {  
 }  
 public User(String name, String email) {  
 this.name = name;  
 this.email = email;  
 }  
 public long getId() {  
 return id;  
 }  
 public void setId(long id) {  
 this.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;  
 }  
}

**activity\_main.xml**

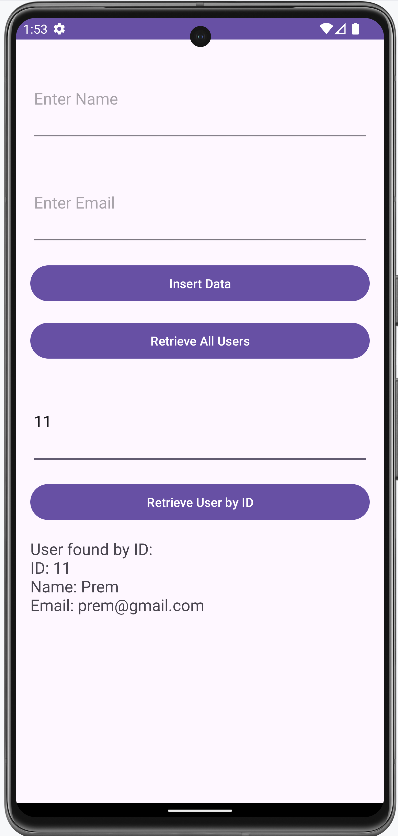
*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
 <EditText  
 android:id="@+id/editTextName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:hint="Enter Name"/>  
 <EditText  
 android:id="@+id/editTextEmail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:layout\_below="@id/editTextName"  
 android:layout\_marginTop="16dp"  
 android:inputType="textEmailAddress"  
 android:hint="Enter Email"/>  
 <Button  
 android:id="@+id/btnInsert"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editTextEmail"  
 android:layout\_marginTop="16dp"  
 android:text="Insert Data"/>  
 <Button  
 android:id="@+id/btnRetrieveAll"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btnInsert"  
 android:layout\_marginTop="16dp"  
 android:text="Retrieve All Users"/>  
 <EditText  
 android:id="@+id/editTextUserId"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:layout\_below="@id/btnRetrieveAll"  
 android:layout\_marginTop="16dp"  
 android:hint="Enter User ID"/>  
 <Button  
 android:id="@+id/btnRetrieveById"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editTextUserId"  
 android:layout\_marginTop="16dp"  
 android:text="Retrieve User by ID"/>  
 <TextView  
 android:id="@+id/textViewData"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/btnRetrieveById"  
 android:layout\_marginTop="16dp"  
 android:text="Data will be displayed here."  
 android:textSize="18sp"/>  
</RelativeLayout>

**AndroidManifest.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Dbase"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>

**Output:**

|  |  |
| --- | --- |
|  |  |

****

**Alarm manager.**

**MainActivity.java**

package com.example.alarmmanger;  
  
import android.annotation.SuppressLint;  
import android.os.Bundle;  
import androidx.appcompat.app.AppCompatActivity;  
import android.app.AlarmManager;  
import android.app.PendingIntent;  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
import android.content.IntentFilter;  
import android.os.Build;  
import android.os.SystemClock;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import androidx.annotation.RequiresApi;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.Locale;  
public class MainActivity extends AppCompatActivity {  
 private TextView alarmTextView;  
 private AlarmManager alarmManager;  
 private PendingIntent alarmIntent;  
 private static final long *INTERVAL\_TEN\_SECONDS* = 10000; *// 10 seconds in milliseconds* private BroadcastReceiver alarmReceiver = new BroadcastReceiver() {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 SimpleDateFormat sdf = new SimpleDateFormat("HH:mm:ss", Locale.*getDefault*());  
 String currentTime = sdf.format(new Date());  
 alarmTextView.setText("Alarm triggered at: " + currentTime);  
 }  
 };  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 alarmTextView = findViewById(R.id.*alarmTextView*);  
 Button setAlarmButton = findViewById(R.id.*setAlarmButton*);  
 alarmManager = (AlarmManager) getSystemService(Context.*ALARM\_SERVICE*);  
 Intent alarmReceiverIntent = new Intent(this, AlarmReceiver.class);  
 alarmIntent = PendingIntent.*getBroadcast*(this, 0, alarmReceiverIntent, PendingIntent.*FLAG\_IMMUTABLE*);  
 registerReceiver(alarmReceiver, new IntentFilter("ALARM\_TRIGGERED"));  
 setAlarmButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 setAlarm();  
 }  
 });  
 }  
 @SuppressLint("ScheduleExactAlarm")  
 @RequiresApi(api = Build.VERSION\_CODES.*M*)  
 private void setAlarm() {  
 long triggerTime = SystemClock.*elapsedRealtime*() + *INTERVAL\_TEN\_SECONDS*;  
 alarmManager.setExactAndAllowWhileIdle(AlarmManager.*ELAPSED\_REALTIME\_WAKEUP*, triggerTime, alarmIntent);  
 alarmTextView.setText("Alarm set to trigger after 10 seconds from current time.");  
 }  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 unregisterReceiver(alarmReceiver);  
 }  
}

**AlarmReceiver.java**

package com.example.alarmmanger;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
public class AlarmReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 Intent broadcastIntent = new Intent("ALARM\_TRIGGERED");  
 context.sendBroadcast(broadcastIntent);  
 }  
}

**activity\_main.xml**

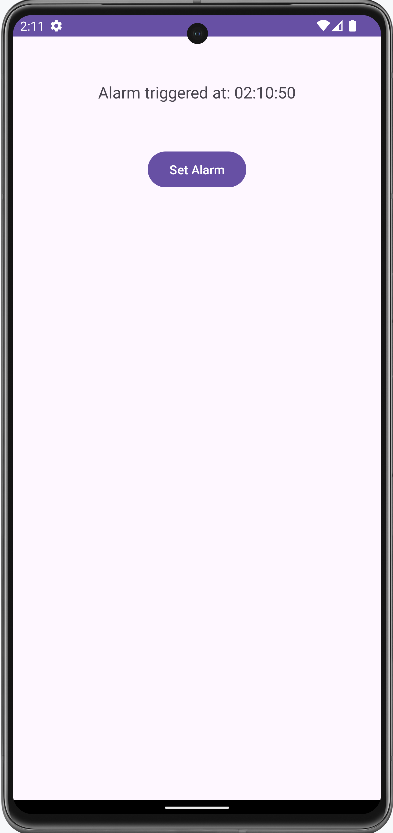
*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/alarmTextView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Alarm Status"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:textSize="18sp"/>  
 <Button  
 android:id="@+id/setAlarmButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/alarmTextView"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:text="Set Alarm"/>  
  
</RelativeLayout>

**AndroidManifest.xml**

*<?*xml version="1.0" encoding="utf-8"*?>*<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
 <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"/>  
 <uses-permission android:name="android.permission.SCHEDULE\_EXACT\_ALARM"/>  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Alarmmanger"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <receiver android:name=".AlarmReceiver"/>  
 </application>  
</manifest>

**Output:**

|  |  |
| --- | --- |
|  |  |

****