**Practical-10 Create an application that use SQLite to locally store data on database and display the data on screen.**

public class Contact {

    int \_id;

    String \_name;

    String \_phone\_number;

    public Contact(){   }

    public Contact(int id, String name, String \_phone\_number){

        this.\_id = id;

        this.\_name = name;

        this.\_phone\_number = \_phone\_number;

    }

    public Contact(String name, String \_phone\_number){

        this.\_name = name;

        this.\_phone\_number = \_phone\_number;

    }

    public int getID(){

        return this.\_id;

    }

    public void setID(int id){

        this.\_id = id;

    }

    public String getName(){

        return this.\_name;

    }

    public void setName(String name){

        this.\_name = name;

    }

    public String getPhoneNumber(){

        return this.\_phone\_number;

    }

    public void setPhoneNumber(String phone\_number){

        this.\_phone\_number = phone\_number;

    }

}

**File: DatabaseHandler.java**

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;

public class DatabaseHandler extends SQLiteOpenHelper {

    private static final int DATABASE\_VERSION = 1;

    private static final String DATABASE\_NAME = "contactsManager";

    private static final String TABLE\_CONTACTS = "contacts";

    private static final String KEY\_ID = "id";

    private static final String KEY\_NAME = "name";

    private static final String KEY\_PH\_NO = "phone\_number";

    public DatabaseHandler(Context context) {

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

        //3rd argument to be passed is CursorFactory instance

    }

    // Creating Tables

    @Override

    public void onCreate(SQLiteDatabase db) {

        String CREATE\_CONTACTS\_TABLE = "CREATE TABLE " + TABLE\_CONTACTS + "("

                + KEY\_ID + " INTEGER PRIMARY KEY," + KEY\_NAME + " TEXT,"

                + KEY\_PH\_NO + " TEXT" + ")";

        db.execSQL(CREATE\_CONTACTS\_TABLE);

    }

    // Upgrading database

    @Override

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

        // Drop older table if existed

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

        // Create tables again

        onCreate(db);

    }

    // code to add the new contact

    void addContact(Contact contact) {

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(KEY\_NAME, contact.getName()); // Contact Name

        values.put(KEY\_PH\_NO, contact.getPhoneNumber()); // Contact Phone

        // Inserting Row

        db.insert(TABLE\_CONTACTS, null, values);

        //2nd argument is String containing nullColumnHack

        db.close(); // Closing database connection

    }

    // code to get the single contact

    Contact getContact(int id) {

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.query(TABLE\_CONTACTS, new String[] { KEY\_ID,

                        KEY\_NAME, KEY\_PH\_NO }, KEY\_ID + "=?",

                new String[] { String.valueOf(id) }, null, null, null, null);

        if (cursor != null)

            cursor.moveToFirst();

        Contact contact = new Contact(Integer.parseInt(cursor.getString(0)),

                cursor.getString(1), cursor.getString(2));

        // return contact

        return contact;

    }

    // code to get all contacts in a list view

    public List<Contact> getAllContacts() {

        List<Contact> contactList = new ArrayList<Contact>();

        // Select All Query

        String selectQuery = "SELECT  \* FROM " + TABLE\_CONTACTS;

        SQLiteDatabase db = this.getWritableDatabase();

        Cursor cursor = db.rawQuery(selectQuery, null);

        // looping through all rows and adding to list

        if (cursor.moveToFirst()) {

            do {

                Contact contact = new Contact();

                contact.setID(Integer.parseInt(cursor.getString(0)));

                contact.setName(cursor.getString(1));

                contact.setPhoneNumber(cursor.getString(2));

                // Adding contact to list

                contactList.add(contact);

            } while (cursor.moveToNext());

        }

        // return contact list

        return contactList;

    }

    // code to update the single contact

    public int updateContact(Contact contact) {

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(KEY\_NAME, contact.getName());

        values.put(KEY\_PH\_NO, contact.getPhoneNumber());

        // updating row

        return db.update(TABLE\_CONTACTS, values, KEY\_ID + " = ?",

                new String[] { String.valueOf(contact.getID()) });

    }

    // Deleting single contact

    public void deleteContact(Contact contact) {

        SQLiteDatabase db = this.getWritableDatabase();

        db.delete(TABLE\_CONTACTS, KEY\_ID + " = ?",

                new String[] { String.valueOf(contact.getID()) });

        db.close();

    }

    // Getting contacts Count

    public int getContactsCount() {

        String countQuery = "SELECT  \* FROM " + TABLE\_CONTACTS;

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.rawQuery(countQuery, null);

        cursor.close();

        // return count

        return cursor.getCount();

    }

}

**MainActivity.java**

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.util.Log;

import java.util.List;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        DatabaseHandler db = new DatabaseHandler(this);

        // Inserting Contacts

        Log.d("Insert: ", "Inserting ..");

        db.addContact(new Contact("Ravi", "9100000000"));

        db.addContact(new Contact("Srinivas", "9199999999"));

        db.addContact(new Contact("Tommy", "9522222222"));

        db.addContact(new Contact("Karthik", "9533333333"));

        // Reading all contacts

        Log.d("Reading: ", "Reading all contacts..");

        List<Contact> contacts = db.getAllContacts();

        for (Contact cn : contacts) {

            String log = "Id: " + cn.getID() + " ,Name: " + cn.getName() + " ,Phone: " +

                    cn.getPhoneNumber();

            // Writing Contacts to log

            Log.d("Name: ", log);

        }

    }

}

