

Experiment using Databases in Android (SQLite)

Mobile Application Development Lab

Design an Activity



Student Grade Details

Roll

Name of student

Average

Grade

INSERT

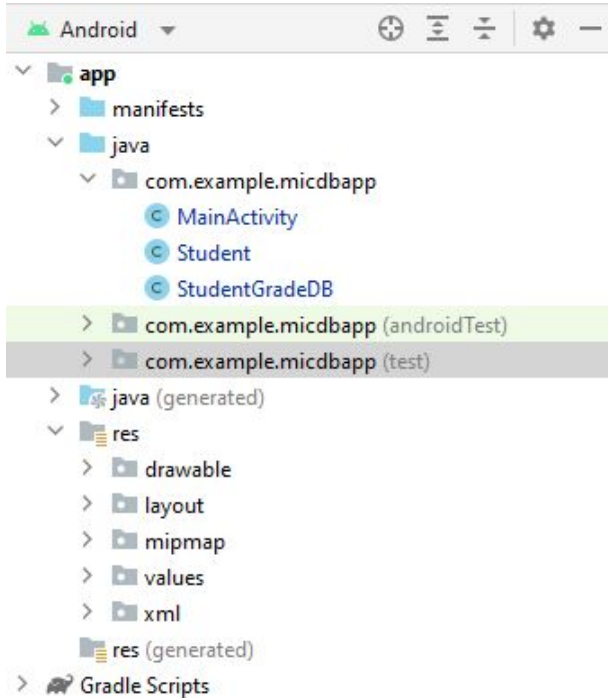
DELETE

SELECT

UPDATE

Student Grade Details	
Roll	editstt_roll
Name of student	editstt_name
Average	editstt_avg
Grade	editstt_grade
INSERT	
DELETE	
SELECT	
UPDATE	

MainActivity.java



```
package com.example.micdbapp;

import ...

public class MainActivity extends AppCompatActivity
{
    StudentGradeDB sgdb;
    Student s;
    EditText edttxtv_roll, edttxtv_sname, edttxtv_avg, edttxtv_grade;
    int roll;
    String sname, grade;
    float avg;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sgdb=new StudentGradeDB(getApplicationContext());

        edttxtv_roll=findViewById(R.id.edttxtv_roll);
        edttxtv_sname=findViewById(R.id.edttxtv_sname);
        edttxtv_avg=findViewById(R.id.edttxtv_avg);
        edttxtv_grade=findViewById(R.id.edttxtv_grade);
    }
}
```

MainActivity.java - insert student when insert button is clicked

```
public void insertStudent(View v)
{

    roll=Integer.parseInt(edtxtv_roll.getText().toString());
    sname=edtxtv_sname.getText().toString();
    avg=Float.parseFloat(edtxtv_avg.getText().toString());
    grade=edtxtv_grade.getText().toString();
    s=new Student(roll,sname,avg,grade);

    sgdb.addStudent(s);
    Toast.makeText( context: this, text: "Insertion Successful", Toast.LENGTH_SHORT).show();
}
```

MainActivity.java - delete student when delete button is clicked

```
public void delete_student(View v)
{
    try {
        roll = Integer.parseInt(edtxtv_roll.getText().toString());
        sgdb.deleteStudent(roll);
        //Toast.makeText(this, "Deletion is successful", Toast.LENGTH_SHORT).show();
        Log.d( tag: "MICHAEL good",  msg: "delete_student: Successful");
    }
    catch (Exception ex)
    {
        Log.d( tag: "PROBLEM MICHAEL:",  msg: "delete_student: "+ex.getMessage());
    }
}
```

MainActivity.java - retrieve student when select button is clicked

```
public void get_student(View v)
{
    try {
        roll = Integer.parseInt(edtxttv_roll.getText().toString());
        String c= sgdb.getStudent(roll);
        Toast.makeText( context: this, c, Toast.LENGTH_SHORT).show();
        Log.d( tag: "MICHAEL good", msg: "select_student: Successful"+c);
    }
    catch (Exception ex)
    {
        Log.d( tag: "PROBLEM MICHAEL:", msg: "select_student: "+ex.getMessage());
    }
}
```

MainActivity.java - retrieve student when select button is clicked

```
public void updateStudent(View v)
{
    try {
        roll = Integer.parseInt(edtxtv_roll.getText().toString());
        sname = edtxtv_sname.getText().toString();
        avg = Float.parseFloat(edtxtv_avg.getText().toString());
        grade = edtxtv_grade.getText().toString();
        s = new Student(roll, sname, avg, grade);

        sgdb.updateStudent(s);
        Toast.makeText(context: this, text: "updatation Successful", Toast.LENGTH_SHORT).show();
        Log.d(tag: "MICHAEL good", msg: "update_student: Successful");
    }
    catch (Exception ex)
    {
        Log.d(tag: "MICHAEL problem", msg: "update_student:"+ex.getMessage());
    }
}
```


SQLiteOpenHelper implementation - create a new java file

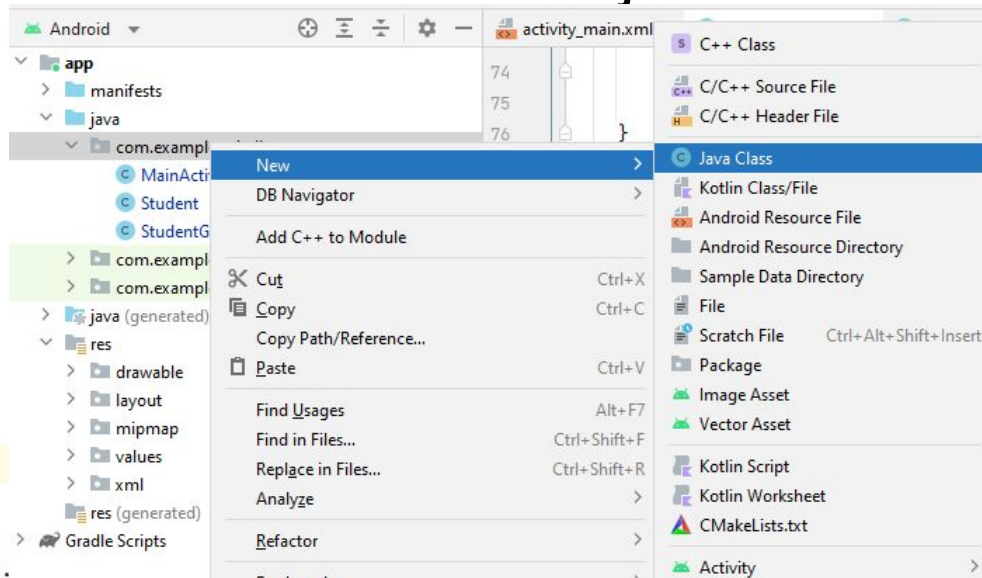
```
package com.example.micdbapp;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteCursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import com.example.micdbapp.Student;
import java.util.List;

public class StudentGradeDB extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "students";
    private static final String TABLE_NAME = "StudentGrade";
    private static final String KEY_ID = "Roll";
    private static final String KEY_NAME = "sname";
    private static final String KEY_AVG = "average";
    private static final String KEY_GRADE = "grade";

    public StudentGradeDB(Context context) {
        super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
        //3rd argument to be passed is CursorFactory instance
    }
}
```



SQLiteOpenHelper - onCreate(), onUpgrade()

```
// Creating Tables
@Override
public void onCreate(SQLiteDatabase db) {
    String CREATE_StudentS_TABLE = "CREATE TABLE " + TABLE_NAME + "("
        + KEY_ID + " INTEGER PRIMARY KEY,"
        + KEY_NAME + " TEXT,"
        + KEY_AVG + " FLOAT,"
        + KEY_GRADE + " TEXT" + ")";
    db.execSQL(CREATE_StudentS_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_NAME);

    // Create tables again
    onCreate(db);
}
```

SQLiteOpenHelper - Insertion

```
// code to add the new Student
void addStudent(Student student) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_NAME, student.getSname()); // Student Name
    values.put(KEY_ID, student.getRoll()); // Student Roll
    values.put(KEY_GRADE, student.getGrade());
    values.put(KEY_AVG, student.getAverage());

    // Inserting Row
    db.insert(TABLE_NAME, nullColumnHack: null, values);
    //2nd argument is String containing nullColumnHack
    db.close(); // Closing database connection
}
```

SQLiteOpenHelper - Selection

```
String getStudent(int roll)
{
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor c=db.query( table: "StudentGrade",new String[]{"Roll","sname","average","grade"},
        selection: "Roll=?",new String[]{Integer.toString(roll)},groupBy: null, having: null, orderBy: null);
    if(c.moveToFirst()) {
        int r = c.getInt( 0);
        String sname = c.getString( 1);
        float avg = c.getFloat( 2);
        String grade = c.getString( 3);
        String print = "Roll=" + r + "\nName=" + sname + "\nAverage=" + avg + "\nGrade=" + grade;
        return print;
    }
    else
    {
        return "No record available";
    }
}
```

SQLiteOpenHelper - Deletion & Updation

```
void deleteStudent(int roll)
{
    SQLiteDatabase db = this.getWritableDatabase();
    //db.rawQuery("delete from StudentGrade where roll="+roll,null);
    db.delete( table: "StudentGrade", whereClause: "Roll=?",new String[]{Integer.toString(roll)});
    db.close();
}

void updateStudent(Student student) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_NAME, student.getSname()); // Student Name
    values.put(KEY_ID, student.getRoll()); // Student Roll
    values.put(KEY_GRADE, student.getGrade());
    values.put(KEY_AVG, student.getAverage());

    // Inserting Row
    db.update(TABLE_NAME, values, whereClause: "Roll=?",new String[]{Integer.toString(student.getRoll())});
    //2nd argument is String containing nullColumnHack
    db.close(); // Closing database connection
}
```


Student class

```
package com.example.micdbapp;
public class Student
{
    private int roll;
    private String sname;
    private float average;
    private String grade;
    public Student(int roll,String sname,float average,String grade)
    {
        this.sname=sname;
        this.roll=roll;
        this.average=average;
        this.grade=grade;
    }
    public int getRoll() { return roll; }
    public String getSname() { return sname; }
    public float getAverage() { return average; }
    public String getGrade() { return grade; }
    public void setRoll(int roll) { this.roll = roll; }
    public void setSname(String sname) { this.sname = sname; }
    public void setGrade(String grade) { this.grade = grade; }
    public void setAverage(float average) { this.average = average; }
}
```

Using Database Inspector under App Inspection

The screenshot displays an IDE interface with the following components:

- Code Editor:** Shows `MainActivity.java` with the following code:

```
86 sqdb.updateStudent(s);
87 Toast.makeText(context: this, text: "updation Successful", Toast.LENGTH_SHORT).show();
88 Log.d( tag: "MICHAEL good", msg: "update_student: Successful");
89 }
90 catch (Exception ex)
91 {
92     Log.d( tag: "MICHAEL problem", msg: "update_student:"+ex.getMessage());
93 }
94 }
95 }
```
- App Inspection Panel:** Shows the selected application as `Pixel 2 API 30 > com.example.micdbapp`. It includes tabs for `Database Inspector`, `Network Inspector`, and `Background Task Inspector`.
- Database Inspector:** Displays a table named `StudentGrade` with the following data:

	Roll	sname	average	grade
1	1203	chandu	59.0	NULL
2	1204	Das	56.70000076293945	NULL
3	1205	Eli	78.0	NULL
4	1210	Illa	56.0	NULL
- Status Bar:** Indicates `App Inspection is running in the background. // You can either stop it, or leave it running and resume your session later. (a minute ago)`.

Select student result in logcat

Project

Commit

Pull Requests

Resource Manager

DB Browser

Structure

Bookmarks

File

Edit

View

Navigate

Code

Refactor

Build

Run

Tools

Git

Window

DB Navigator

Help

MicDBApp

app

src

main

java

com

example

micdbapp

MainActivity

Emulator: Pixel 2 API 30

app

Pixel 2 API 30

Git

85

86

87

88

89

90

91

92

93

94

95

xml

MainActivity.java

sgad.U

Toast.

Log.d(

}

catch (Excepti

{

Log.d(tag:

}

}

Device Manager

Device File Explorer

Emulator

Logcat

Logcat

+

Pixel 2 API 30 (emulator-5554) Android 11, API 30

Press Control + Space to see available keys

Q- 1201

0 results

↑

↓

⌂

+

-

⌕

⌕

⌕

2022-11-18 11:50:17.759 29901-29901 BOONDBROKERSVC pid-29901 U ononbind: Intent t act=com.google.android.gms.common.BIND_SH

2022-11-18 11:56:18.690 14986-15037 system_server pid-14986 I oneway function results will be dropped but finished with st

2022-11-18 11:56:18.692 32320-32320 MICHAEL good com.example.micdbapp D select_student: SuccessfulRoll=1203

2022-11-18 11:56:18 827 14986-15886 system_server nid-14986 I oneway function results will be dropped but finished with st

App Inspection is running in the background. // You can either stop it, or leave it running and resume your session later. (a minute ago)

4482:111 (435 chars, 3 line breaks) LF UTF-8 4 spaces master

