05 - Database Handling

Ex. No. : 5 **Roll No. :**

Date : **Reg. No. :**

Aim

Develop an application that makes use of Database.

Objective

In this chapter we will learn to use a database to save and access data.

We will use the SQLiteDatabase and SQLiteOpenHelper classes to implement database handling.

Procedure

Start a New Project “Internal Database”

Delete the "Hello World” TextView.

Creating database handler class

To handle a database we need to create a database handler class.

So, right click on the package directory (inside java folder) and select New then Java Class.

Name it for example DatabaseHelper.

```
public class DatabaseHelper {
}
```

Now we need to extend the SQLiteOpenHelper class.

```
public class DatabaseHelper extends SQLiteOpenHelper{
}
```

This gives an error. So go ahead and press [ALT]+[ENTER].

A pop-up appears and select Implement methods.

Select onCreate(db:SQLiteDatabase):void and onUpgrade (db:SQLiteDatabase, oldVersion:int, newVersion:int):void and click OK.

Now these methods are automatically generated for you.

```
@Override
public void onCreate(SQLiteDatabase db) {

}

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

}
```

Add the following code above the onCreate method.

```
db.execSQL("create table employee (eid integer primary key, ename
text)");
```

Now we need to create a constructor for our class.

Press [ALT] + [Insert] and select Constructor select the first constructor with the arguments SQLiteOpenHelper(context:Context, name:String, factory:CursorFactory, version:int).

```
public DatabaseHelper(Context context, String name,
SQLiteDatabase.CursorFactory factory, int version) {
    super(context, name, factory, version);
}
```

Make some changes:

```
public DatabaseHelper(Context context) {
    super(context, "REC", null, 1);
}
```

Creating table:

Now we need to create a table in database.

Inside the onCreate function add the following code:

```
db.execSQL("create table employee (eid integer primary key, ename
varchar)");
```

Do not forget the spaces and commas (,) after/before the concatenations.

Now inside the onUpgrade function add:

```
db.execSQL("drop table if exists employee");
onCreate(db);
```

Inserting and deleting items (rows) – database:

Add a new method addEmployee().

```
    public void addEmployee(int eid, String ename) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("insert into employee values (" + eid + ", '" +
           ename + "')");
        db.close();
    }
```

This method inserts a new row into the table.

Add a new method viewEmployee().

```
    public Cursor viewEmployee(int eid) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor c = db.rawQuery("select * from employee where
           eid = " + eid, null);
        return c;
    }
```

Add a new method modifyEmployee().

```
    public void modifyEmployee(int eid, String ename) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("update employee set ename = '" + ename + "'
           where eid = " + eid);
        db.close();
    }
```

Add a new method deleteEmployee().

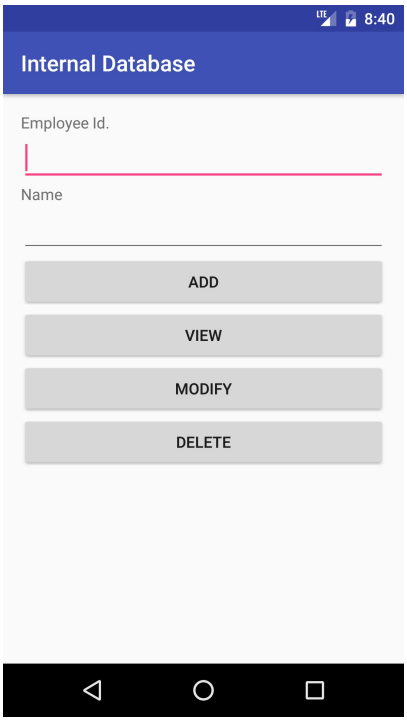
```
    public void deleteEmployee(int eid) {
        SQLiteDatabase db = this.getWritableDatabase();
        db.execSQL("delete from employee where eid = " + eid);
        db.close();
    }
```

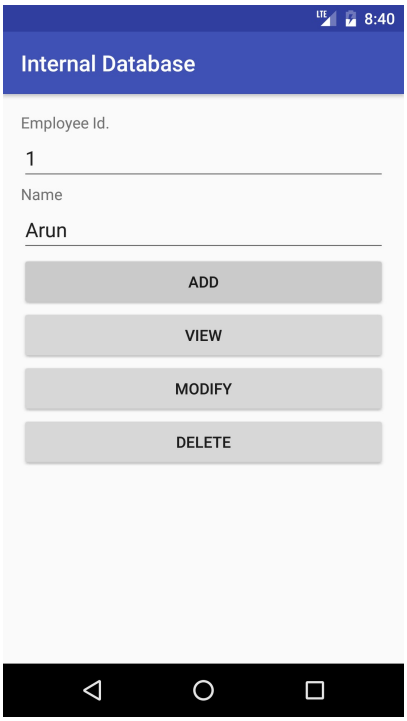
Insert the following code in MainActivity.java.

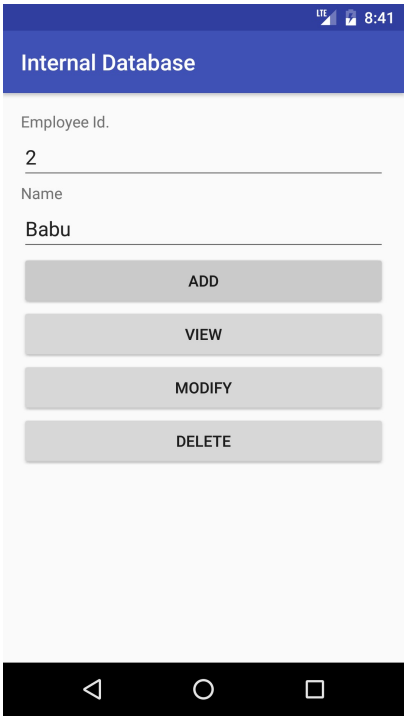
```
EditText etId, etName;
Button btnAdd, btnView, btnModify, btnDelete;
DatabaseHelper db;

etId = (EditText) findViewById(R.id.etId);
etName = (EditText) findViewById(R.id.etName);
btnAdd = (Button) findViewById(R.id.btnAdd);
btnView = (Button) findViewById(R.id.btnView);
btnModify = (Button) findViewById(R.id.btnModify);
```

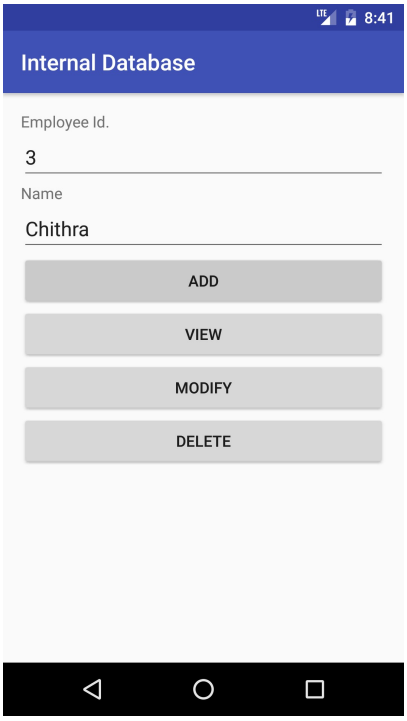
```
btnDelete = (Button) findViewById(R.id.btnDelete);
db = new DatabaseHelper(this);

btnAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int eid = Integer.parseInt(etId.getText().toString());
        String ename = etName.getText().toString();
        db.addEmployee(eid, ename);
    }
});

btnView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int eid = Integer.parseInt(etId.getText().toString());
        Cursor rs = db.viewEmployee(eid);
        if(rs.moveToNext())
            etName.setText(rs.getString(1));
    }
});

btnModify.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int eid = Integer.parseInt(etId.getText().toString());
        String ename = etName.getText().toString();
        db.modifyEmployee(eid, ename);
    }
});

btnDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int eid = Integer.parseInt(etId.getText().toString());
        db.deleteEmployee(eid);
    }
});
```

Output


8:40
Internal Database
Employee Id.
Name
ADD
VIEW
MODIFY
DELETE


8:40
Internal Database
Employee Id.
1
Name
Arun
ADD
VIEW
MODIFY
DELETE


8:41
Internal Database
Employee Id.
2
Name
Babu
ADD
VIEW
MODIFY
DELETE


8:41
Internal Database
Employee Id.
3
Name
Chithra
ADD
VIEW
MODIFY
DELETE