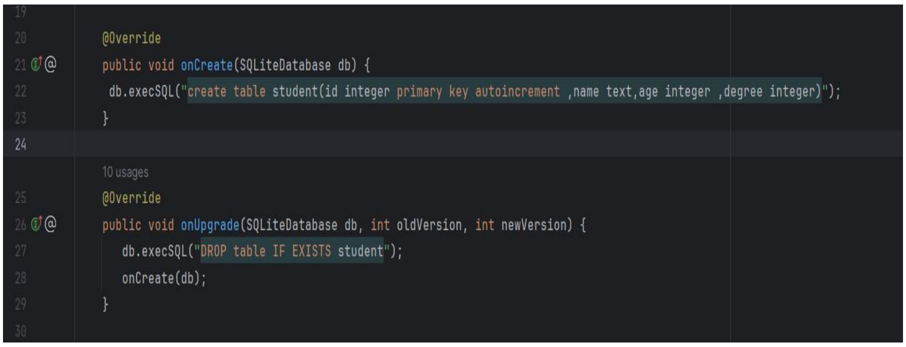
SQL Lite

This document will talk about how to create a database in your mobile app by using SQL lite.

1- First we have to create a class EXTENDS from SQLiteOpenHelper

- SQLiteOpenHelper : is an abstract class used to manage database creation and version management , this class takes care of opening the database if it exists, creating it if it does not, and upgrading it as necessary and You create a subclass implementing onCreate, onUpgrade
  such as :



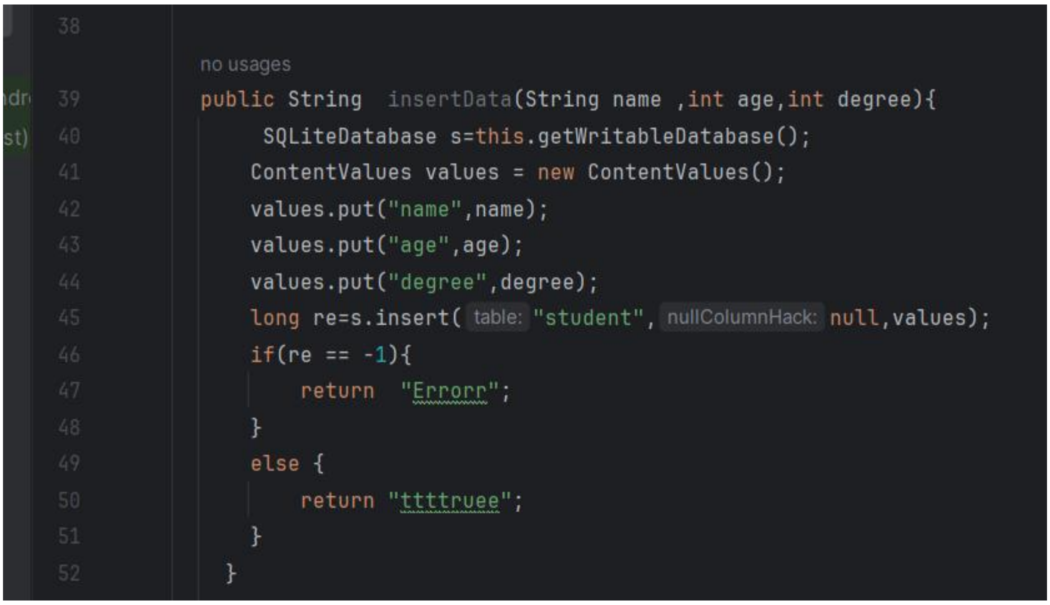
```java
@Override
public void onCreate(SQLiteDatabase db) {
 db.execSQL("create table student(id integer primary key autoincrement ,name text,age integer ,degree integer)");
}

10 usages
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
   db.execSQL("DROP table IF EXISTS student");
   onCreate(db);
}
```

2- Creating to important methods

1- Method to inserting data:

First we define a variable from SQLiteDatabase class it’s very important because it has methods to create, delete, execute SQL commands, and perform other common database management tasks .

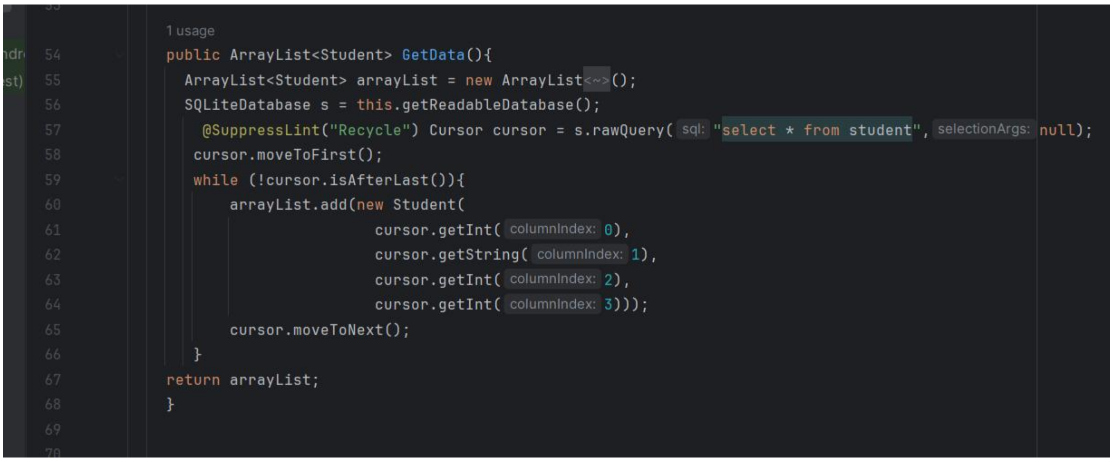
And the ContentValues class is used to store a set of values.



```java
no usages
public String  insertData(String name ,int age,int degree){
     SQLiteDatabase s=this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("name",name);
    values.put("age",age);
    values.put("degree",degree);
    long re=s.insert( table: "student", nullColumnHack: null,values);
    if(re == -1){
        return  "Errorr";
    }
    else {
        return "ttttruee";
    }
  }
```

* Hint (using of getWritableDatabase() function to writing data into Database).

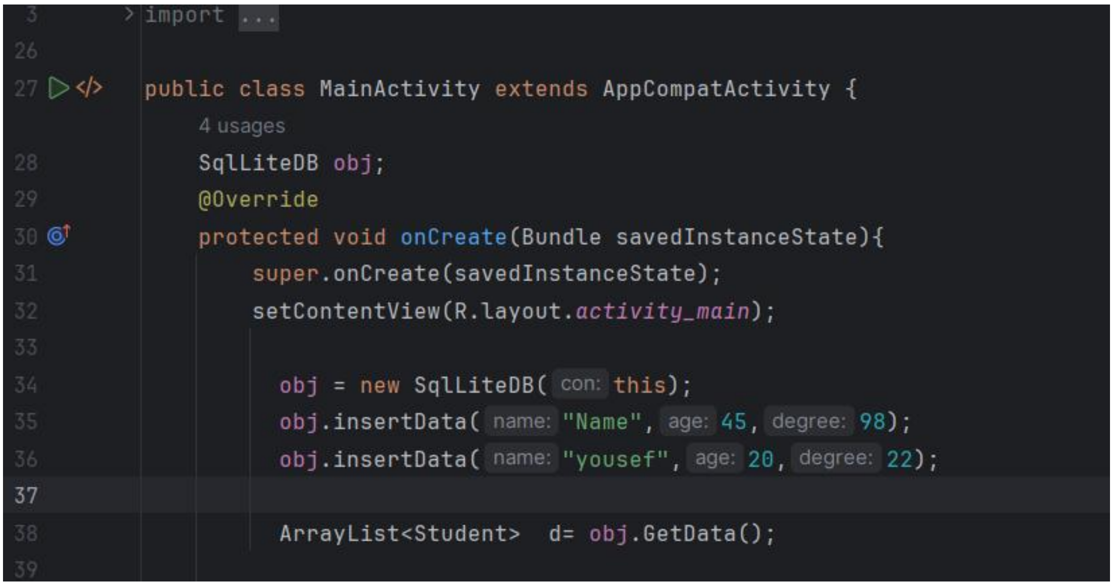
2- Method to get all data :



```
1 usage
public ArrayList<Student> GetData(){
  ArrayList<Student> arrayList = new ArrayList<~>();
  SQLiteDatabase s = this.getReadableDatabase();
   @SuppressLint("Recycle") Cursor cursor = s.rawQuery( sql: "select * from student", selectionArgs: null);
   cursor.moveToFirst();
   while (!cursor.isAfterLast()){
       arrayList.add(new Student(
                       cursor.getInt( columnIndex: 0),
                       cursor.getString( columnIndex: 1),
                       cursor.getInt( columnIndex: 2),
                       cursor.getInt( columnIndex: 3)));
       cursor.moveToNext();
   }
return arrayList;
}
```

* Cursor is a interface provides random read-write access to the result set returned by a database query , Cursor implementations are not required to be synchronized so code using a Cursor from multiple threads should perform its own synchronization when using the Cursor.

*And finally to insert and get data from mainActivity look for this code :



```
> import ...

public class MainActivity extends AppCompatActivity {
    4 usages
    SqlLiteDB obj;
    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

          obj = new SqlLiteDB( con: this);
          obj.insertData( name: "Name", age: 45, degree: 98);
          obj.insertData( name: "yousef", age: 20, degree: 22);

          ArrayList<Student>  d= obj.GetData();

```