
Lab_SQLite_02

❑ CHUẨN BỊ

- ĐÃ TẠO DATABASE, TABLE

❑ NỘI DUNG

- XÂY DỰNG PHƯƠNG THỨC KẾT NỐI SQLite
- XÂY DỰNG CÁC PHƯƠNG THỨC LẤY DỮ LIỆU
- XÂY DỰNG CÁC PHƯƠNG THỨC THÊM DỮ LIỆU
- XÂY DỰNG CÁC PHƯƠNG THỨC SỬA DỮ LIỆU
- XÂY DỰNG CÁC PHƯƠNG THỨC XÓA DỮ LIỆU

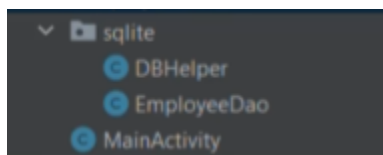
1. MỤC TIÊU:

- Tạo kết nối database
- Xây dựng các phương thức lấy, thêm, xóa, sửa dữ liệu

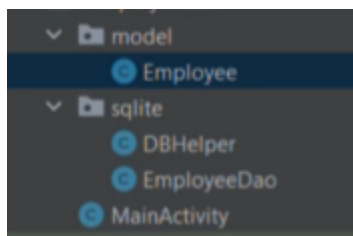
2. THỰC HIỆN

BUỐC 1: Tạo class EmployeeDao

Tại package sqlite đã tạo ta tạo mới class EmployeeDao



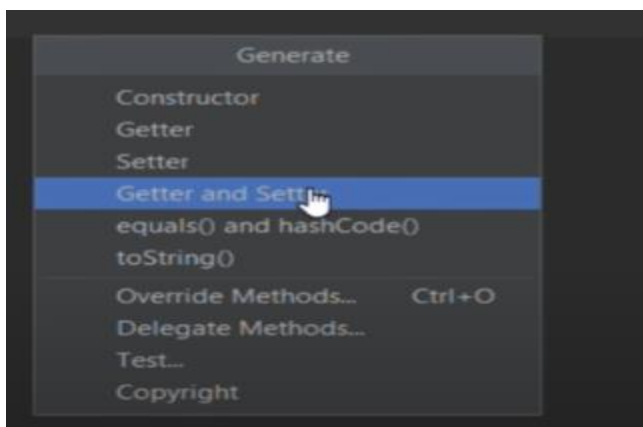
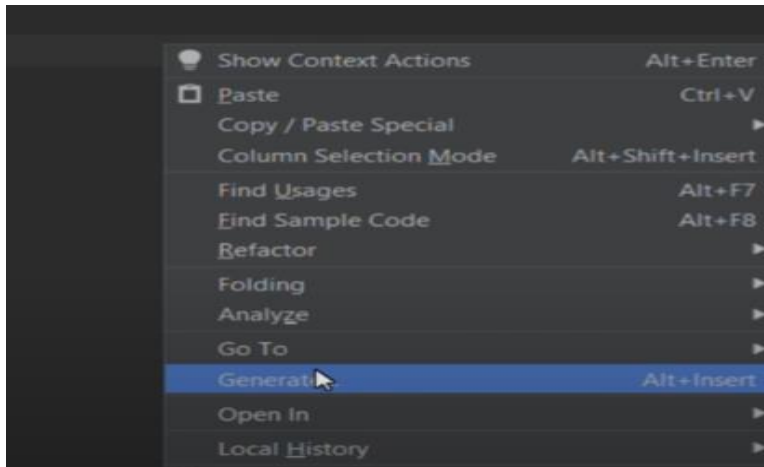
BUỐC 2: Tạo mới package model và class Employee trong package model

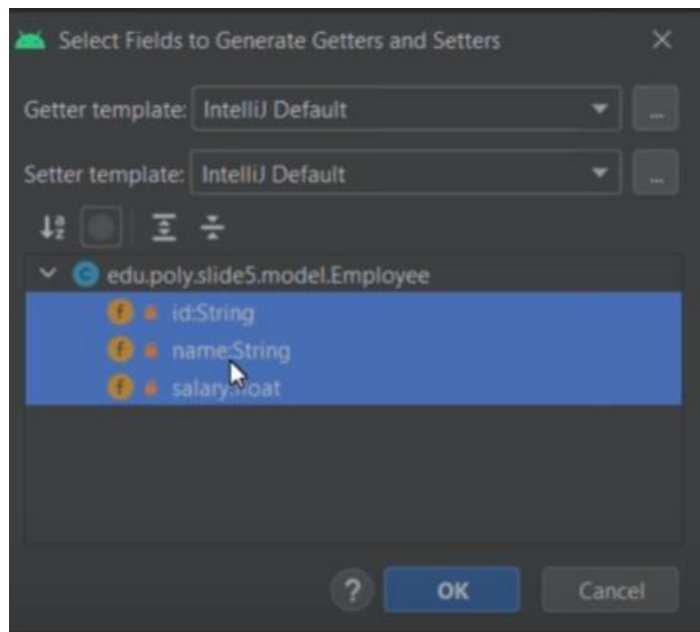


BUỐC 3: Định nghĩa các thông tin trong class model Employee

```
public class Employee {  
    private String id, name;  
    private float salary;  
  
    |  
}
```

Tiếp theo, tạo các phương thức Getter và Setter của các trường thông tin Id, name, salary

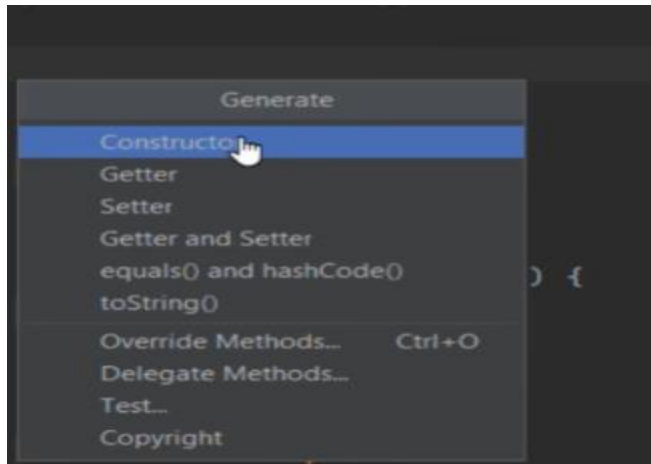




KẾT QUẢ:

```
public class Employee {  
    private String id, name;  
    private float salary;  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

BƯỚC 4: Tạo ra constructor ngầm định

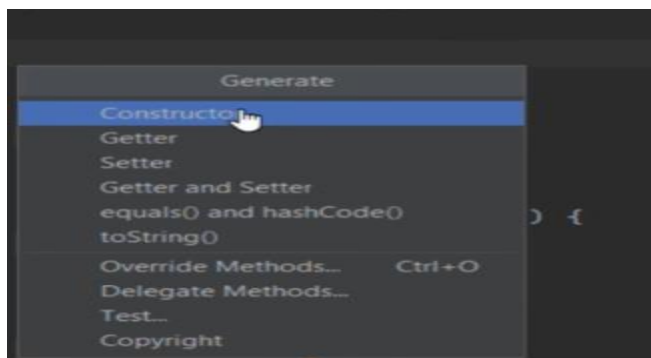


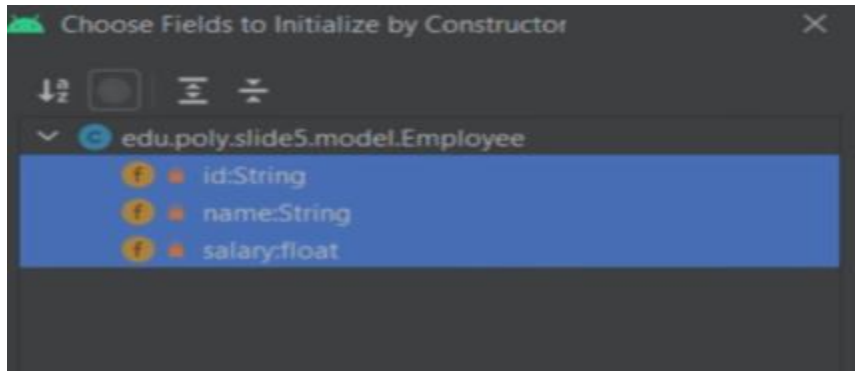
Bỏ chọn id, name, salary để sinh ra constructor ngầm định

Kết quả:

```
public Employee() {  
}
```

BƯỚC 5: Tạo ra constructor để khởi tạo giá trị cho 3 trường thông tin: id, name, salary khi khởi tạo đối tượng.





Kết quả:

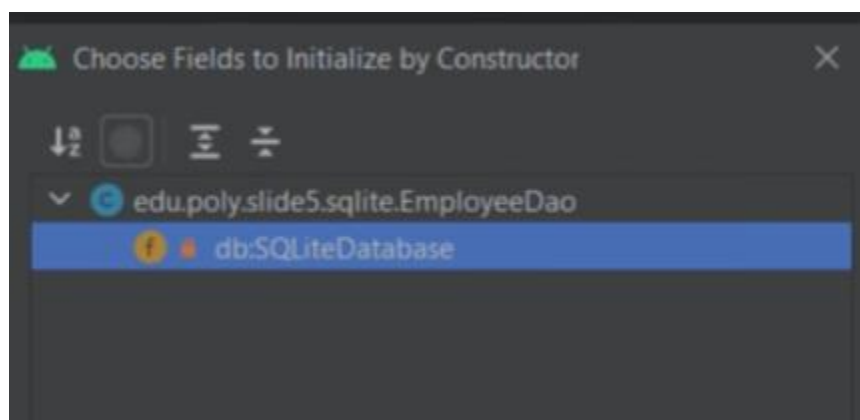
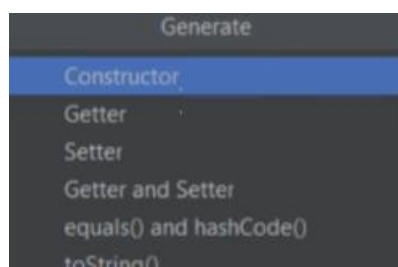
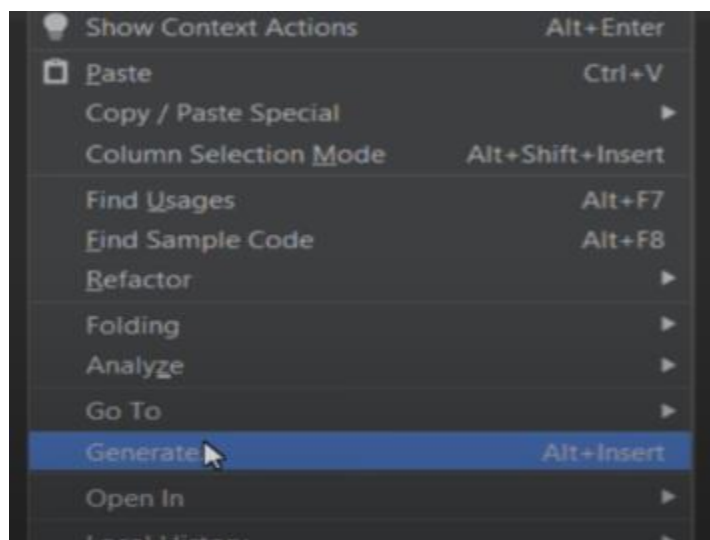
```
public Employee(String id, String name, float salary) {  
    this.id = id;  
    this.name = name;  
    this.salary = salary;  
}
```

BƯỚC 6: Tại class EmployeeDao

Khai báo trường dữ liệu db // để kết nối

```
import android.database.sqlite.SQLiteDatabase;  
  
public class EmployeeDao {  
    private SQLiteDatabase db;  
  
    |  
}
```

Tiếp theo, tạo constructor EmployeeDao



Kết quả:

```
public class EmployeeDao {  
    private SQLiteDatabase db;  
  
    public EmployeeDao(Context context) {  
    }  
}
```

BUỐC 7: Sử dụng đối tượng db để mở cơ sở dữ liệu của class DBHelper

```
public class EmployeeDao {  
    private SQLiteDatabase db;  
  
    public EmployeeDao(Context context) {  
        DBHelper helper = new DBHelper(context);  
  
        db = helper.getWritableDatabase();  
    }  
}
```

BUỐC 8: Xây dựng phương thức truy vấn lấy danh sách dữ liệu bằng câu truy vấn và đối số truyền vào

```
public EmployeeDao(Context context) {
    DBHelper helper = new DBHelper(context);

    db = helper.getWritableDatabase();
}

@SuppressWarnings("Range")
public List<Employee> get(String sql, String ...selectArgs){
    List<Employee> list = new ArrayList<>();
    Cursor cursor = db.rawQuery(sql, selectArgs);

    while (cursor.moveToNext()) {
        Employee emp = new Employee();
        emp.setId(cursor.getString(cursor.getColumnIndex(s "id")));
        emp.setName(cursor.getString(cursor.getColumnIndex(s "name")));
        emp.setSalary(cursor.getFloat(cursor.getColumnIndex(s "salary")));

        list.add(emp);
    }

    return list;
}
```

BUỐC 9: Xây dựng phương thức trả về **tất cả** nhân viên hiện có

Phía dưới phương thức get ta tạo thêm phương thức getAll như sau:

```
public List<Employee> getAll(){
    String sql = "SELECT * FROM nhanvien";

    return get(sql);
}
```

BƯỚC 10: Xây dựng phương thức trả về thông tin nhân viên theo id

```
public Employee getById(String id) {  
    String sql = "SELECT * FROM nhanvien WHERE id = ?";  
  
    List<Employee> list = get(sql, id);  
  
    return list.get(0);  
}
```

BƯỚC 11: Xây dựng phương thức thêm mới một nhân viên

```
public long insert(Employee emp) {  
    ContentValues values = new ContentValues();  
    values.put("id", emp.getId());  
    values.put("name", emp.getName());  
    values.put("salary", emp.getSalary());  
  
    return db.insert( table: "nhanvien", nullColumnHack: null, values);  
}
```

BƯỚC 12: Xây dựng phương thức CẬP NHẬT dữ liệu một nhân viên

```
public long update(Employee emp) {  
    ContentValues values = new ContentValues();  
    values.put("name", emp.getName());  
    values.put("salary", emp.getSalary());  
  
    return db.update( table: "nhanvien", values, whereClause: "id=?", new String[]{emp.getId()});  
}
```

BƯỚC 13: Xây dựng phương thức XÓA dữ liệu một nhân viên theo Id

```
public int delete(String id){  
    return db.delete( table: "nhanvien", whereClause: "id=?", new String[]{id});  
}
```