

Lab_SQLite_03

❑ CHUẨN BỊ

- ĐÃ TẠO DATABASE, TABLE
- 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

❑ NỘI DUNG

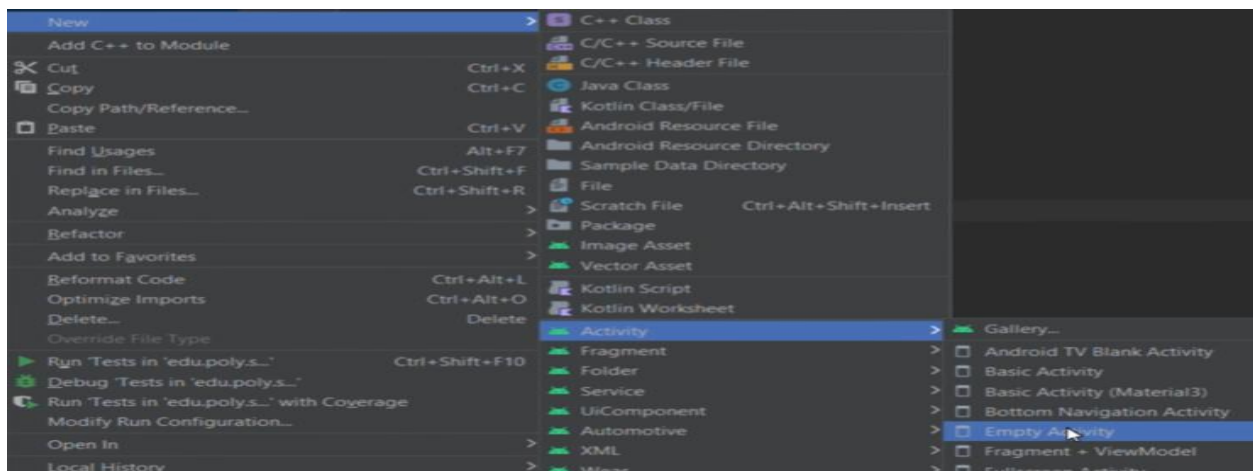
- SHOW DỮ LIỆU LÊN LISTVIEW
- XỬ LÝ CẬP NHẬT DỮ LIỆU
- XỬ LÝ XÓA DỮ LIỆU

1. MỤC TIÊU:

- Lấy dữ liệu trong SQLite lên layout XML, cụ thể là ListView
- Xây dựng các chức năng xử lý cập nhật dữ liệu, xóa dữ liệu, thêm dữ liệu trong activity

2. THỰC HIỆN

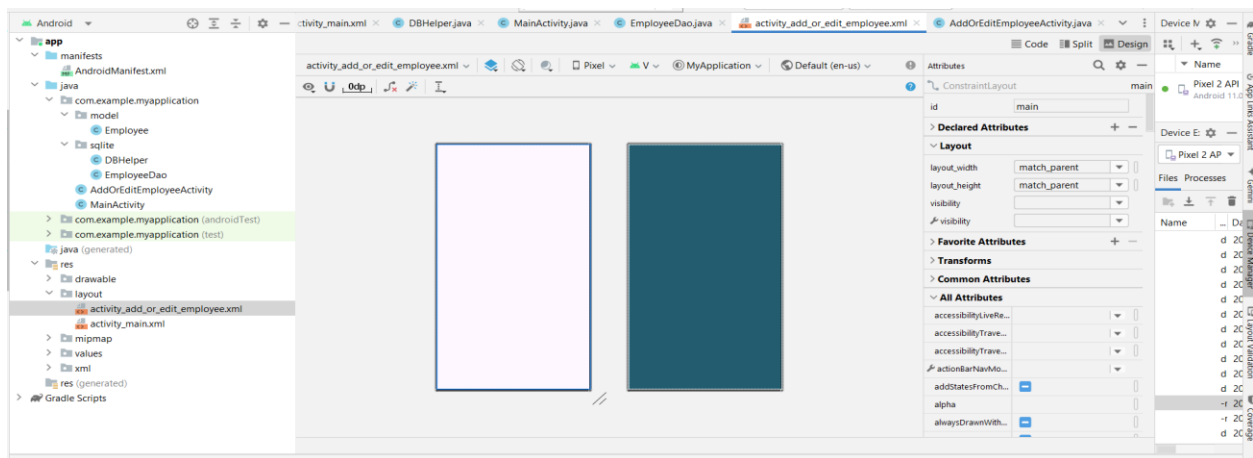
BƯỚC 1: TẠO mới activity đặt tên AddOrEditEmployeeActivity



Kết quả:

```
public class AddOrEditEmployeeActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_add_or_edit_employee);  
    }  
}
```

BƯỚC 2: TẠO LAYOUT tại `activity_add_or_edit_employee.xml`



Yêu cầu:

- Sử dụng `LinearLayout` tổng với các thuộc tính: `orientation = vertical`, `layout_height = wrap_content`

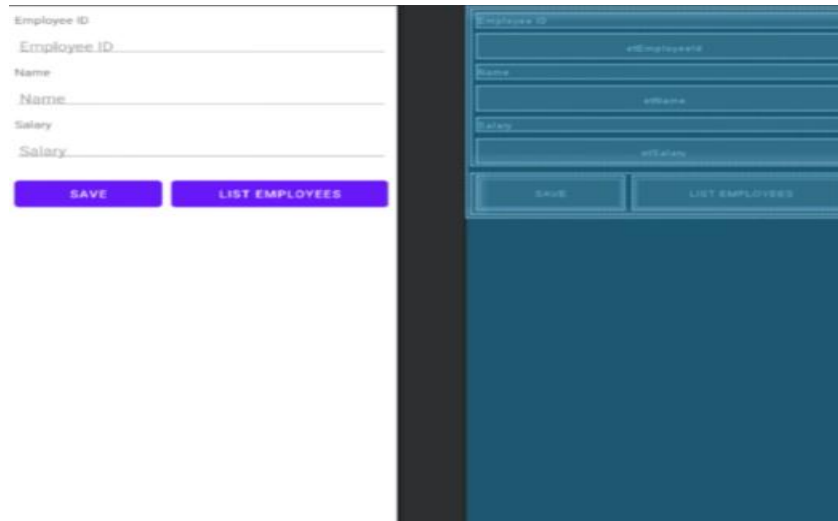
Chứa 2 `LinearLayout` con:

1 sử dụng `orientation = vertical`

và 1 sử dụng `orientation = horizontal`

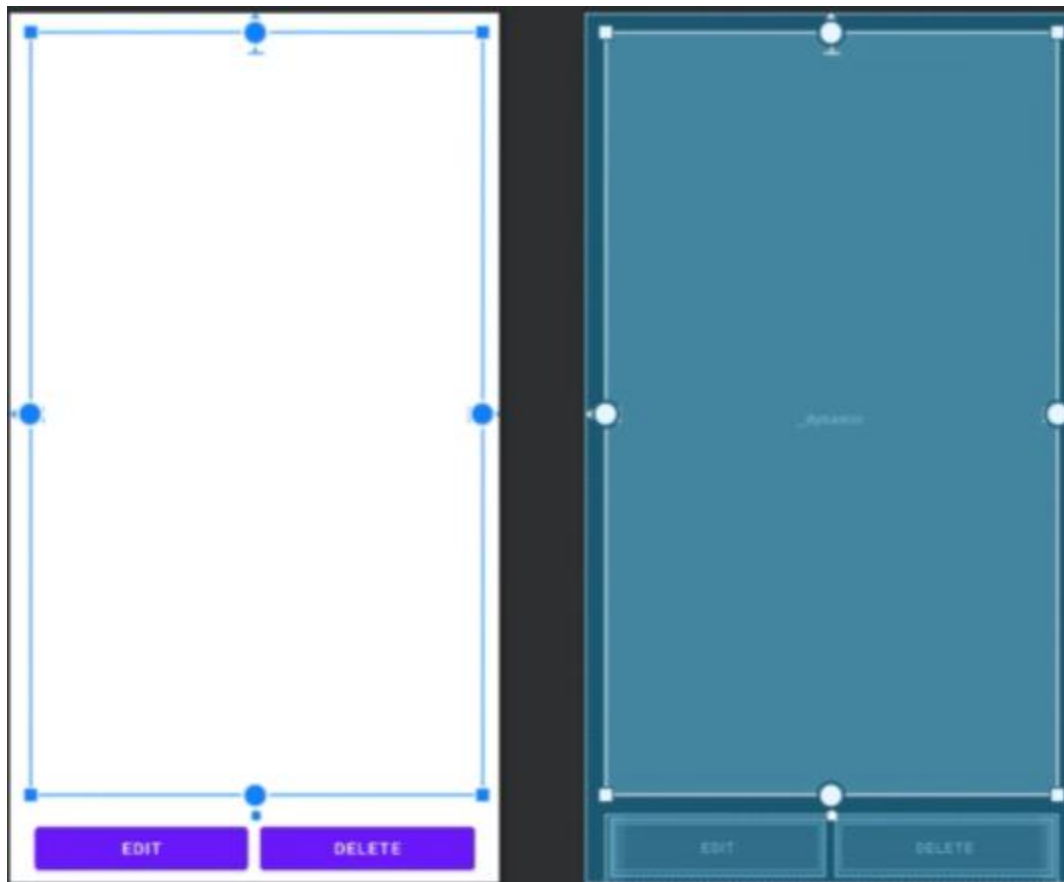
- `LinearLayout` con 1 (định dạng: `layout_margin = 5dp`): Tạo 2 `textView`, 1 **Number** đặt Id lần lượt: `etEmployeeId`, `etName`, **`etSalary`**.

- `LinearLayout` con 2 (định dạng: `layout_margin = 5dp`): Tạo 2 button đặt Id lần lượt: `btSave` và `btListEmployee`



BƯỚC 3: Tại activity_main.xml xây dựng layout như hình, có:

- 1 listview đặt Id: lvEmployees
- 2 button, đặt Id lần lượt: btnEdit, btnDelete



BUỚC 4: Tại MainActivity.java XỬ LÝ SỰ KIỆN KHI NGƯỜI DÙNG CLICK VÀO NÚT EDIT HOẶC DELETE

Tạo implements View.OnClickListener

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        DBHelper dbHelper = new DBHelper(context: this);  
        SQLiteDatabase database = dbHelper.getReadableDatabase();  
        database.close();  
    }  
}
```

BUỚC 5: Tạo sự kiện onClick

```
@Override  
public void onClick(View view) {  
    switch (view.getId()) {  
        case R.id.btnEdit:  
            Intent intent = new Intent(packageContext: this, AddOrEditEmployeeActivity.class);  
            startActivity(intent);  
            break;  
    }  
}
```

BUỚC 6: Tại onCreate bỏ chức năng kết nối database

Bắt sự sự onClick theo Id của btnEdit và btnDelete

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    DBHelper dbHelper = new DBHelper(this);  
    SQLiteDatabase database = dbHelper.getReadableDatabase();  
    database.close();  
    findViewById(R.id.btnEdit).setOnClickListener(this);  
    findViewById(R.id.btnDelete).setOnClickListener(this);  
}
```

SAU ĐÓ, RUN CHẠY KIỂM TRA.

Click vào NÚT EDIT VÀ DELETE sẽ thấy chuyển qua Activity
AddOrEditEmployeeActivity

**BUỚC 7: Tại AddOrEditEmployeeActivity.java thêm implements
View.OnClickListener để bắt sự kiện onClick**

```
public class AddOrEditEmployeeActivity extends AppCompatActivity
    implements View.OnClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_or_edit_employee);
    }

    @Override
    public void onClick(View view) {
    }
}
```

Tiếp theo, viết xử lý tại hàm onClick() khi Click vào btnSave và btListEmployee

Đầu tiên, khai báo các trường thông tin và gọi phương thức findViewById

```
public class AddOrEditEmployeeActivity extends AppCompatActivity
    implements View.OnClickListener {
    private EditText etEmployeeId, etName, etSalary;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_or_edit_employee);

        etEmployeeId = findViewById(R.id.etEmployeeId);
        etName = findViewById(R.id.etName);
        etSalary = findViewById(R.id.etSalary);
    }
}
```

Sau đó, viết hàm xử lý tại onClick() khi click vào nút Save và đưa ra thông báo

```
@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btnSave:
            EmployeeDao dao = new EmployeeDao(context, this);
            Employee emp = new Employee();

            emp.setId(etEmployeeId.getText().toString());
            emp.setName(etName.getText().toString());
            emp.setSalary(Float.parseFloat(etSalary.getText().toString()));

            dao.insert(emp);
            Toast.makeText(context, this, text: "New employee has been saved!",
                Toast.LENGTH_SHORT).show();
            break;
    }
}
```

Tại onCreate() bổ sung phương thức findViewById khi người dùng click vào nút Save hoặc nút ListEmployee

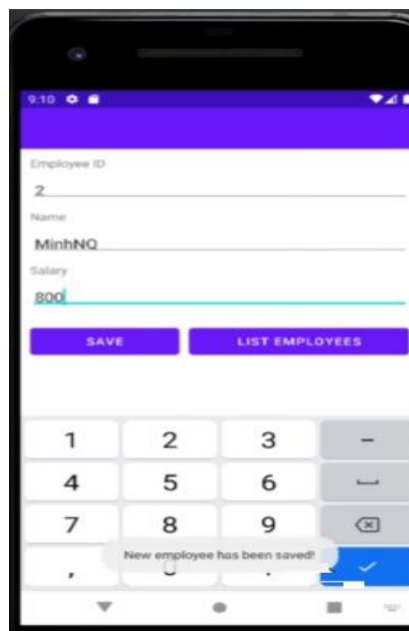
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_or_edit_employee);

    etEmployeeId = findViewById(R.id.etEmployeeId);
    etName = findViewById(R.id.etName);
    etSalary = findViewById(R.id.etSalary);

    findViewById(R.id.btnSave).setOnClickListener(this);
    findViewById(R.id.btnListEmployees).setOnClickListener(this);
}
```

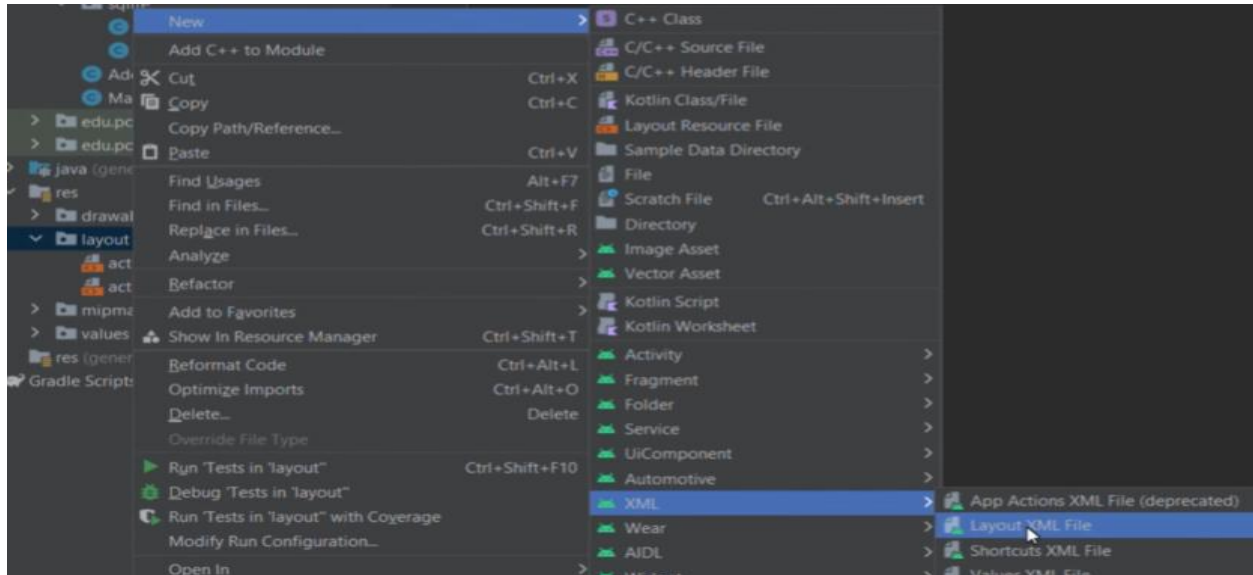
BƯỚC 8: RUN CHẠY APP

- Click vào nút Edit từ mainActivity sẽ chuyển qua activity AddOrEditEmployeeActivity
- Sau đó, nhập các thông tin và nhấn nút Save thì sẽ thấy lưu csdl vào SQLite và đưa ra thông báo.



BUỚC 9:

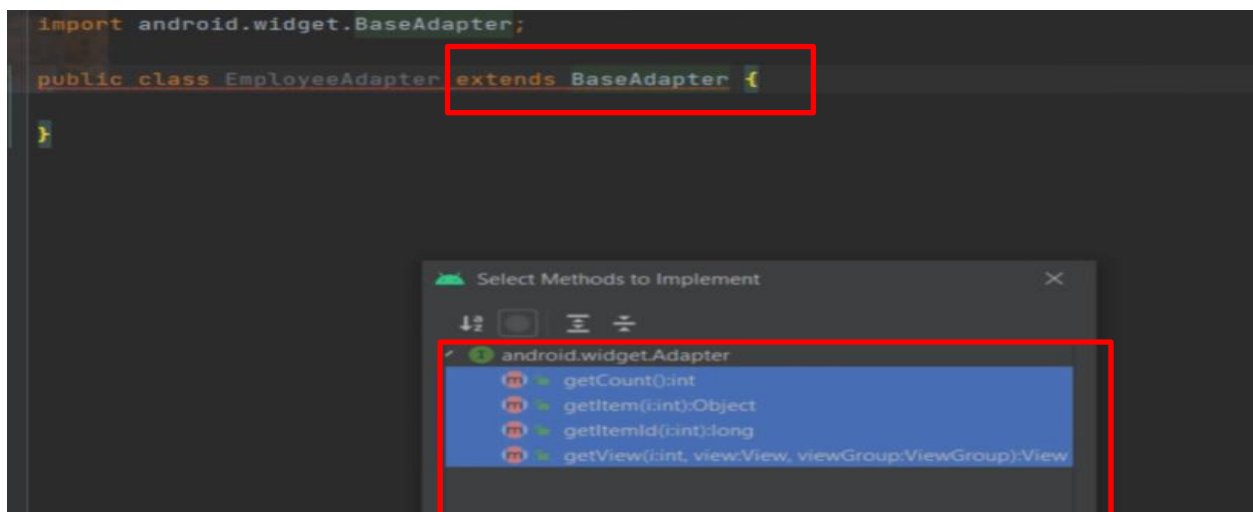
- Click vào Package gốc tạo mới package đặt tên adapter và trong package adapter tạo mới class đặt tên EmployeeAdapter
- Click vào thư mục layout, tạo mới layout và đặt tên: layout_employee_item



BUỚC 10: Thiết kế layout_employee_item để hiển thị Name và Salary của nhân viên

- Sử dụng 1 LinearLayout: vertical, wrap_content, margin: 5dp, id =
- Trong LinearLayout dùng 2 textView: margin: 3dp, đặt id lần lượt: tvName, tvSalary

BUỚC 11: Tại class EmployeeAdapter: thực hiện **extends BaseAdapter** và implement các phương thức có trong class BaseAdapter



Bổ sung các trường thông tin dữ liệu và tạo constructor dựa trên 2 thành phần dữ liệu là context và list

```
public class EmployeeAdapter extends BaseAdapter {  
    private Context context;  
    private List<Employee> list;  
  
    public EmployeeAdapter(Context context, List<Employee> list) {  
        this.context = context;  
        this.list = list;  
    }  
  
    @Override  
    public int getCount() {  
        return 0;  
    }  
  
    @Override  
    public Object getItem(int i) {  
        return null;  
    }  
}
```

Tiếp theo bổ sung thông tin cho các phương thức getCount(), getItem(), getItemId(), getView()

```
@Override  
public int getCount() {  
    return list.size();  
}  
  
@Override  
public Object getItem(int iPosition) {  
    return list.get(iPosition);  
}  
  
@Override  
public long getItemId(int i) {  
    return i;  
}
```

```

@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    if(view == null){
        view = LayoutInflater.from( context).inflate(R.layout.layout_employee_item, root: null);
    }

    TextView tvName = view.findViewById(R.id.tvName);
    TextView tvSalary = view.findViewById(R.id.tvSalary);

    Employee emp = list.get(i);
    tvName.setText(emp.getName());
    tvSalary.setText("" + emp.getSalary());

    return view;
}

```

BƯỚC 12: Tại MainActivity.jav thực hiện các khai báo adapter, listview và bổ sung phương thức findViewById

```

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private EmployeeAdapter employeeAdapter;
    private ListView lvEmployees;
}

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    DBHelper dbHelper = new DBHelper(this);
    SQLiteDatabase database = dbHelper.getReadableDatabase();
    database.close();
    findViewById(R.id.btnEdit).setOnClickListener(this);
    findViewById(R.id.btnDelete).setOnClickListener(this);

    lvEmployees = findViewById(R.id.lvEmployees);
    EmployeeDao dao = new EmployeeDao( context: this);
    List<Employee> list = dao.getAll();

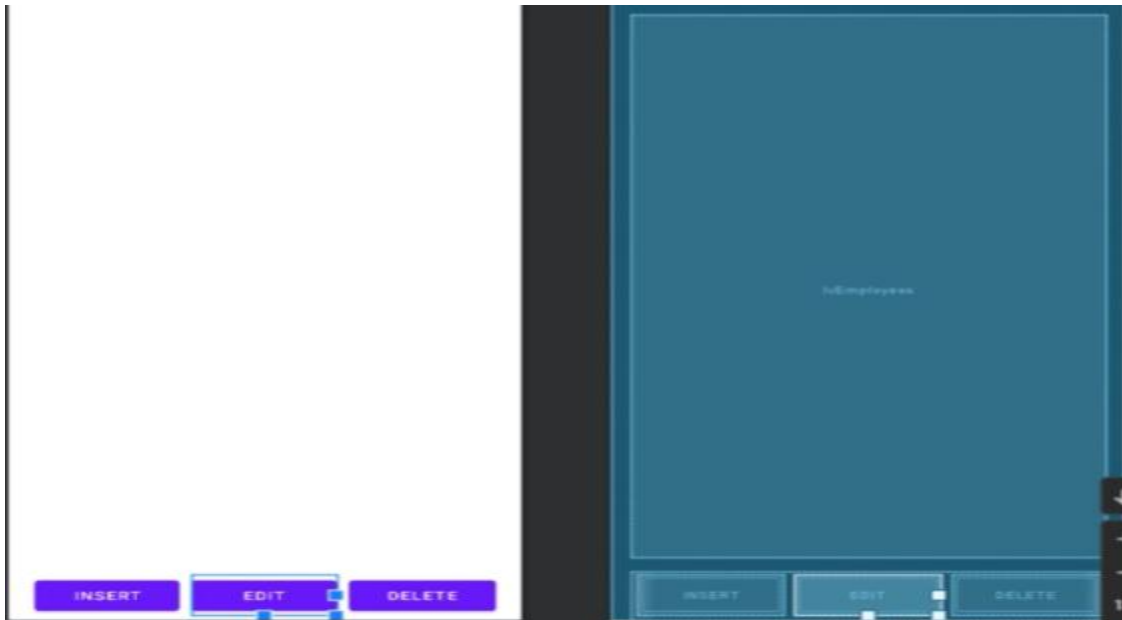
    employeeAdapter = new EmployeeAdapter( context: this, list);
    lvEmployees.setAdapter(employeeAdapter);
}

```

RUN CHẠY LẠI, SẼ THẤY DANH SÁCH HIỂN THỊ LÊN MÀN HÌNH

BƯỚC 13:

TẠI activity_main.xml BỔ SUNG THÊM NÚT INSERT để thêm mới dữ liệu, đặt id = btnInsert



Tại MainActivity.java: gán phương thức xử lý sự kiện khi người dùng click vào Insert

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    DBHelper dbHelper = new DBHelper(this);
    SQLiteDatabase database = dbHelper.getReadableDatabase();
    database.close();

    findViewById(R.id.btnEdit).setOnClickListener(this);
    findViewById(R.id.btnInsert).setOnClickListener(this);
    findViewById(R.id.btnDelete).setOnClickListener(this);

    lvEmployees = findViewById(R.id.lvEmployees);

    EmployeeDao dao = new EmployeeDao(context: this);
    List<Employee> list = dao.getAll();
    employeeAdapter = new EmployeeAdapter(context: this, list);
    lvEmployees.setAdapter(employeeAdapter);
}
```

Bổ sung gán sự kiện tại onClick(), tùy chỉnh lại đoạn code

```
@Override
public void onClick(View view) {
    Intent intent = new Intent( packageContext: this, AddOrEditEmployeeActivity.class);
    switch (view.getId()){
        case R.id.btnInsert:
            startActivity(intent);
            break;
        case R.id.btnEdit:
            Bundle bundle = new Bundle();
            bundle.putString("id", employeeId);
            intent.putExtra( name: "data", bundle);

            startActivity(intent);
            break;
    }
}
```

BƯỚC 14: Lấy thông tin nhân viên mà người dùng chọn để EDIT

Tại MainActivity.java: Khai báo biến bắt Id nhân viên

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private EmployeeAdapter employeeAdapter;
    private ListView lvEmployees;
    private String employeeId;
```

Sau đó, tại hàm onCreate() bổ sung phương thức setOnItemClickListener()

```
findViewById(R.id.btnEdit).setOnClickListener(this);
findViewById(R.id.btnInsert).setOnClickListener(this);
findViewById(R.id.btnDelete).setOnClickListener(this);

lvEmployees = findViewById(R.id.lvEmployees);

EmployeeDao dao = new EmployeeDao( context: this);
List<Employee> list = dao.getAll();
EmployeeAdapter = new EmployeeAdapter( context: this, list);
lvEmployees.setAdapter(employeeAdapter);
lvEmployees.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Employee emp = list.get(i);
        employeeId = emp.getId();
    }
});
```

Tiếp theo, Tại sự onClick() bổ sung ràng buộc khi người dùng chưa chọn Nhân viên muốn chỉnh sửa mà đã click vào EDIT thì hiện thông báo

```
@Override
public void onClick(View view) {
    Intent intent = new Intent( packageContext: this, AddOrEditEmployeeActivity.class);
    switch (view.getId()){
        case R.id.btnInsert:
            startActivity(intent);
            break;
        case R.id.btnEdit:
            if (employeeId == null){
                Toast.makeText( context: this, text: "Employee id must be selected", Toast.LENGTH_SHORT).show();
                return;
            }

            Bundle bundle = new Bundle();
            bundle.putString("id", employeeId);
            intent.putExtra( name: "data", bundle);
```

BƯỚC 15: Tại AddOrEditEmployeeActivity.java bổ sung phương thức **readEmployee()** để khi người dùng chọn nhân viên muốn SỬA và click vào EDIT thì sẽ **nạp thông tin nhân viên muốn SỬA lên activity**

```
        findViewById(R.id.btnSave).setOnClickListener(this);
        findViewById(R.id.btnListEmployees).setOnClickListener(this);
    }

    private void readEmployee(){
        Intent intent = getIntent();
        Bundle bundle = intent.getBundleExtra( name: "data");
        if (bundle == null){
            return;
        }
        String id = bundle.getString( key: "id");

        EmployeeDao dao = new EmployeeDao( context: this);
        Employee emp = dao.getById(id);

        etEmployeeId.setText(emp.getId());
        etName.setText(emp.getName());
        etSalary.setText(""+ emp.getSalary());
    }

    @Override
    public void onClick(View view) {
```

Tại hàm onCreate() bổ sung hàm gọi `readEmployee()`

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_or_edit_employee);

    etEmployeeId = findViewById(R.id.etEmployeeId);
    etName = findViewById(R.id.etName);
    etSalary = findViewById(R.id.etSalary);

    findViewById(R.id.btnSave).setOnClickListener(this);
    findViewById(R.id.btnListEmployees).setOnClickListener(this);
    readEmployee();
}

private void readEmployee(){
    Intent intent = getIntent();
    Bundle bundle = intent.getBundleExtra("data");
    if (bundle == null){
        return;
    }
}
```

RUN CHẠY LẠI APP

- Click vào nút INSERT sẽ xuất hiện activity thêm dữ liệu
- Click vào nút EDIT thì thông báo người dùng chưa chọn nhân viên muốn sửa
- Click chọn nhân viên muốn sửa, rồi Click vào nút EDIT thì hiển thị ra activity chứa thông tin người dùng muốn sửa.



TA THẤY VỚI CHỨC NĂNG INSERT VÀ EDIT ĐỀU CHUYỂN ĐẾN a AddOrEditEmployeeActivity. DO ĐÓ, TA CẦN ĐIỀU CHỈNH HIỂN THỊ NÚT SAVE CỦA AddOrEditEmployeeActivity LẠI CHO PHÙ HỢP: với INSERT thì hiển thị là SAVE còn với EDIT thì hiển thị là UPDATE => BƯỚC 16 SẼ XỬ LÝ VẤN ĐỀ NÀY

BƯỚC 16:

TẠI AddOrEditEmployeeActivity.java bổ sung đối tượng button dùng chung, đặt là btnSave

```
public class AddOrEditEmployeeActivity extends AppCompatActivity
    implements View.OnClickListener {
    private EditText etEmployeeId, etName, etSalary;
    private Button btnSave;
```

Tại onCreate() bổ sung

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_or_edit_employee);

    etEmployeeId = findViewById(R.id.etEmployeeId);
    etName = findViewById(R.id.etName);
    etSalary = findViewById(R.id.etSalary);

    btnSave = findViewById(R.id.btnSave);
    btnSave.setOnClickListener(this);

    findViewById(R.id.btnListEmployees).setOnClickListener(this);

    readEmployee();
}
```

Tại readEmployee() bổ sung

```
Bundle bundle = Intent.getBundleExtra("name: data");
if (bundle == null){
    return;
}
String id = bundle.getString("key: id");

EmployeeDao dao = new EmployeeDao(context: this);
Employee emp = dao.getById(id);

etEmployeeId.setText(emp.getId());
etName.setText(emp.getName());
etSalary.setText(" "+ emp.getSalary());

btnSave.setText("Update");
}
```

TẠI onClick() chỉnh sửa code để kiểm tra coi dùng đang dùng insert hay edit.

```
@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.btnSave:
            EmployeeDao dao = new EmployeeDao(context, this);
            Employee emp = new Employee();

            emp.setId(etEmployeeId.getText().toString());
            emp.setName(etName.getText().toString());
            emp.setSalary(Float.parseFloat(etSalary.getText().toString()));

            if (btnSave.getText().equals("Save")) {
                dao.insert(emp);
            }else{
                dao.update(emp);
            }

            Toast.makeText(context, this, text: "New employee has been saved!",
                Toast.LENGTH_SHORT).show();

            break;
    }
}
```

RUN CHẠY LẠI APP

- TA THẤY KHI CHỌN NHÂN VIÊN CHỈNH SỬA VÀ CLICK VÀO NÚT EDIT THÌ CHUYỂN ĐẾN AddOrEditEmployeeActivity VÀ HIỂN THỊ NÚT Update
- Chỉnh sửa trường thông tin nào đó của nhân viên và click vào Update để cập nhật dữ liệu (Nếu chưa xử lý cập nhật thì tắt app và chạy lại để kiểm tra)

BƯỚC 17: XỬ LÝ KHI NGƯỜI DÙNG CLICK VÀO NÚT DELETE

Tại MainActivity.java =>onClickListener() bổ sung thêm case delete

```
Bundle bundle = new Bundle();
bundle.putString("id", employeeId);
intent.putExtra("name", bundle);

startActivity(intent);
break:
case R.id.btnDelete:
    if (employeeId == null){
        Toast.makeText(context, "Employee id must be selected",
            Toast.LENGTH_SHORT).show();
        return;
    }

    EmployeeDao dao = new EmployeeDao(context);
    dao.delete(employeeId);
    employeeId = null;

    Toast.makeText(context, "Employee was deleted",
        Toast.LENGTH_SHORT).show();
    break;
```

RUN CHẠY LẠI APP ĐỂ KIỂM TRA CHỨC NĂNG DELETE .Nếu khi đã xóa rồi mà thông tin hiển thị vẫn còn thì ta tắt app và chạy lại thì sẽ thấy thông tin xóa đã được cập nhật. Điều này, dẫn đến ứng dụng xử lý chậm, không mượt mà. Do đó, bước 18 sẽ xử lý vấn đề này.

BƯỚC 18:

Tại MainActivity bổ sung trường thông tin **list** và Phương thức **onResume()** để xử lý

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    private EmployeeAdapter employeeAdapter;  
    private ListView lvEmployees;  
    private String employeeId;  
    private List<Employee> list;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {...}  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
    }  
}
```

Tại onCreate() điều chỉnh dùng trường list vừa tạo thay vì khai báo đối tượng list tại đây

```
SQLiteDatabase database = dbHelper.getReadableDatabase();  
database.close();  
findViewById(R.id.btnEdit).setOnClickListener(this);  
findViewById(R.id.btnInsert).setOnClickListener(this);  
findViewById(R.id.btnDelete).setOnClickListener(this);  
  
lvEmployees = findViewById(R.id.lvEmployees);  
  
EmployeeDao dao = new EmployeeDao(context, this);  
list = dao.getAll();  
employeeAdapter = new EmployeeAdapter(context, this, list);  
lvEmployees.setAdapter(employeeAdapter);  
lvEmployees.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
        Employee emp = list.get(i);  
        employeeId = emp.getId();  
    }  
});
```

Tiếp theo, điều chỉnh trong phương thức **onResume()**

```
@Override
protected void onResume() {
    super.onResume();

    EmployeeDao dao = new EmployeeDao( context: this);
    List<Employee> updatedList = dao.getAll();

    list.clear();
    updatedList.forEach(item->list.add(item));

    employeeAdapter.notifyDataSetChanged();
}
```

Tại onClick() gọi phương thức onResume() sau khi xóa dữ liệu

```
break;
case R.id.btnDelete:
    if (employeeId == null){
        Toast.makeText( context: this, text: "Employee id must be selected",
            Toast.LENGTH_SHORT).show();
        return;
    }

    EmployeeDao dao = new EmployeeDao( context: this);
    dao.delete(employeeId);
    employeeId = null;
    onResume();

    Toast.makeText( context: this, text: "Employee was deleted",
        Toast.LENGTH_SHORT).show();
    break;
}
```

RUN CHẠY LẠI APP: Click vào Delete nhân viên và Edit 1 nhân viên để kiểm tra

HẾT