--------------------------------DBHelper-----------------------------------------------------

*public class* DBHelper *extends* SQLiteOpenHelper {  
 *public* DBHelper(*@Nullable* Context context) {  
 *super*(context, "provider3", *null*, 1);  
 }  
  
 *@Override  
 public void* onCreate(SQLiteDatabase sqLiteDatabase) {  
 sqLiteDatabase.execSQL("create table PhongBans(" +  
 "id integer primary key," +  
 "name text)");  
 sqLiteDatabase.execSQL("create table NhanViens(" +  
 "id integer primary key," +  
 "name text," +  
 "address text," +  
 "id\_PhongBan integer not null constraint id references " +  
 "PhongBans(id) on delete cascade on update cascade )");  
 }  
  
 *@Override  
 public void* onUpgrade(SQLiteDatabase sqLiteDatabase, *int* i, *int* i1) {  
 sqLiteDatabase.execSQL("drop table if exists NhanViens");  
 sqLiteDatabase.execSQL("drop table if exists PhongBans");  
 onCreate(sqLiteDatabase);  
 }  
  
 *@Override  
 public void* onConfigure(SQLiteDatabase db) {  
 db.setForeignKeyConstraintsEnabled(*true*);  
 *super*.onConfigure(db);  
 }

--------------------------------------------provider------------------------------------------------------

*public class* MyContentProvider *extends* ContentProvider {  
  
 *public static* String AUTHORITY ="vn.com.linh.appa\_provider3";  
  
 *public static* String TABLE\_PHONGBAN="PhongBans";  
 *public static* String TABLE\_NHANVIEN="NhanViens";  
  
 *public static* String URL\_PHONGBAN="content://"+AUTHORITY+"/"+TABLE\_PHONGBAN;  
 *public static* String URL\_NHANVIEN="content://"+AUTHORITY+"/"+TABLE\_NHANVIEN;  
  
 *public static final* Uri CONTENT\_URI\_PHONGBAN=Uri.parse(URL\_PHONGBAN);  
 *public static final* Uri CONTENT\_URI\_NHANVIEN=Uri.parse(URL\_NHANVIEN);  
  
 *public static* UriMatcher uriMatcher;  
  
 *public static final int* PHONGBAN\_ONE=1;  
 *public static final int* PHONGBAN\_ALL=2;  
  
 *public static final int* NHANVIEN\_ONE=3;  
 *public static final int* NHANVIEN\_ALL=4;  
  
 *static* {  
 uriMatcher= *new* UriMatcher(UriMatcher.NO\_MATCH);  
  
 uriMatcher.addURI(AUTHORITY, TABLE\_PHONGBAN, PHONGBAN\_ALL);  
 uriMatcher.addURI(AUTHORITY, TABLE\_PHONGBAN+"/#", PHONGBAN\_ONE);  
  
 uriMatcher.addURI(AUTHORITY, TABLE\_NHANVIEN, NHANVIEN\_ALL);  
 uriMatcher.addURI(AUTHORITY, TABLE\_NHANVIEN+"/#", NHANVIEN\_ONE);  
 }  
 *private* SQLiteDatabase db;  
 *private static* HashMap<String,String> PROJECTION\_MAP;  
 *public* MyContentProvider() {  
 }  
  
 *@Override  
 public int* delete(Uri uri, String selection, String[] selectionArgs) {  
 *int* count=0;  
 *int* uri\_matcher=uriMatcher.match(uri);  
 *switch* (uri\_matcher){  
 *case* PHONGBAN\_ALL:  
 count=db.delete(TABLE\_PHONGBAN, selection,selectionArgs);  
 *break*;  
 *case* PHONGBAN\_ONE:  
 String id\_phongBan=uri.getPathSegments().get(1);  
 count=db.delete(TABLE\_PHONGBAN,"id= "+id\_phongBan,selectionArgs);  
 *break*;  
 *case* NHANVIEN\_ALL:  
 count=db.delete(TABLE\_NHANVIEN, selection,selectionArgs);  
 *break*;  
 *case* NHANVIEN\_ONE:  
 String id\_nhanVien=uri.getPathSegments().get(1);  
 count=db.delete(TABLE\_NHANVIEN,"id= "+id\_nhanVien,selectionArgs);  
 *break*;  
 }  
 getContext().getContentResolver().notifyChange(uri,*null*);  
 *return* count;  
 }  
  
 *@Override  
 public* String getType(Uri uri) {  
 *// TODO: Implement this to handle requests for the MIME type of the data  
 // at the given URI.  
 throw new* UnsupportedOperationException("Not yet implemented");  
 }  
  
 *@Override  
 public* Uri insert(Uri uri, ContentValues values) {  
 *//Phong Ban  
 long* row=db.insert(TABLE\_PHONGBAN, *null*, values);  
 *if* (row>0){  
 Uri uri1= ContentUris.withAppendedId(CONTENT\_URI\_PHONGBAN, row);  
 getContext().getContentResolver().notifyChange(uri1,*null*);  
 *return* uri1;  
 }  
 *//NhanVien  
 long* row2=db.insert(TABLE\_NHANVIEN, *null*, values);  
 *if* (row2>0){  
 Uri uri1= ContentUris.withAppendedId(CONTENT\_URI\_NHANVIEN, row);  
 getContext().getContentResolver().notifyChange(uri1,*null*);  
 *return* uri1;  
 }  
 *throw new* SQLException("Them khong thanh cong");  
  
 }  
  
 *@Override  
 public boolean* onCreate() {  
 *// TODO: Implement this to initialize your content provider on startup.* Context context= getContext();  
 DBHelper dbHelper= *new* DBHelper(context);  
 db=dbHelper.getWritableDatabase();  
 *if* (db==*null*)  
 *return false*;  
 *return true*;  
 }  
  
 *@Override  
 public* Cursor query(Uri uri, String[] projection, String selection,  
 String[] selectionArgs, String sortOrder) {  
 SQLiteQueryBuilder sqLiteQueryBuilder=*null*;  
 *int* uri\_matcher=uriMatcher.match(uri);  
  
 *switch* (uri\_matcher){  
 *case* PHONGBAN\_ALL:  
 sqLiteQueryBuilder=*new* SQLiteQueryBuilder();  
 sqLiteQueryBuilder.setTables(TABLE\_PHONGBAN);  
 sqLiteQueryBuilder.setProjectionMap(PROJECTION\_MAP);  
 *break*;  
 *case* PHONGBAN\_ONE:  
 sqLiteQueryBuilder=*new* SQLiteQueryBuilder();  
 sqLiteQueryBuilder.setTables(TABLE\_PHONGBAN);  
 sqLiteQueryBuilder.appendWhere("id= "+uri.getPathSegments().get(1));  
 *break*;  
 *case* NHANVIEN\_ALL:  
 sqLiteQueryBuilder=*new* SQLiteQueryBuilder();  
 sqLiteQueryBuilder.setTables(TABLE\_NHANVIEN);  
 sqLiteQueryBuilder.setProjectionMap(PROJECTION\_MAP);  
 *break*;  
 *case* NHANVIEN\_ONE:  
 sqLiteQueryBuilder=*new* SQLiteQueryBuilder();  
 sqLiteQueryBuilder.setTables(TABLE\_NHANVIEN);  
 sqLiteQueryBuilder.appendWhere("id= "+uri.getPathSegments().get(1));  
 *break*;  
 }  
 *if* (sortOrder==*null* || sortOrder==""){  
 sortOrder="name";  
 }  
 Cursor cursor=sqLiteQueryBuilder.query(db, projection,selection,selectionArgs,*null*,*null*,sortOrder);  
 cursor.setNotificationUri(getContext().getContentResolver(),uri);  
 *return* cursor;  
 }  
  
 *@Override  
 public int* update(Uri uri, ContentValues values, String selection,  
 String[] selectionArgs) {  
 *int* count=0;  
 *int* uri\_matcher=uriMatcher.match(uri);  
 *switch* (uri\_matcher){  
 *case* PHONGBAN\_ALL:  
 count=db.update(TABLE\_PHONGBAN, values,selection,selectionArgs);  
 *break*;  
 *case* PHONGBAN\_ONE:  
 String id\_phongBan=uri.getPathSegments().get(1);  
 count=db.update(TABLE\_PHONGBAN,values, "id= "+id\_phongBan, selectionArgs);  
 *break*;  
 *case* NHANVIEN\_ALL:  
 count=db.update(TABLE\_NHANVIEN, values,selection,selectionArgs);  
 *break*;  
 *case* NHANVIEN\_ONE:  
 String id\_nhanVien=uri.getPathSegments().get(1);  
 count=db.update(TABLE\_NHANVIEN,values,"id= "+id\_nhanVien,selectionArgs);  
 *break*;  
 }  
 getContext().getContentResolver().notifyChange(uri,*null*);  
 *return* count;  
 }

-------------------------------------------------------------NhanVienActivity--------------------------------------------------------------------------

*public class* NhanVienActivity *extends* AppCompatActivity {  
 EditText ed\_id, ed\_name, ed\_diaChi;  
 Spinner spinner;  
 GridView gridView;  
 Button btn\_them ,btn\_xoa, btn\_sua, btn\_tim, btn\_thoat;  
 *static final* String URL\_NHANVIEN="content://vn.com.linh.appa\_provider3/NhanViens";  
 *static final* String URL\_PHONGBAN="content://vn.com.linh.appa\_provider3/PhongBans";  
 List<NhanVien> nhanVienList=*new* ArrayList<>();  
 List<PhongBan> phongBanList=*new* ArrayList<>();  
 ArrayAdapter<String> adapter;  
 *int* id\_PhongBan;  
 *int* id\_NhanVien;  
 *@Override  
 protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_nhan\_vien);  
  
 ed\_id=findViewById(R.id.ed\_id);  
 ed\_name=findViewById(R.id.ed\_name);  
 ed\_diaChi=findViewById(R.id.ed\_diachi);  
 spinner=findViewById(R.id.spinner\_PhongBan);  
 gridView=findViewById(R.id.gridViewNhanVien);  
 load();  
 select();  
 gridView.setOnItemClickListener(*new* AdapterView.OnItemClickListener() {  
 *@Override  
 public void* onItemClick(AdapterView<?> adapterView, View view, *int* i, *long* l) {  
 NhanVien nhanVien= nhanVienList.get(i);  
 id\_NhanVien=nhanVien.getId();  
 ed\_id.setText(""+id\_NhanVien);  
 ed\_name.setText(nhanVien.getName());  
 ed\_diaChi.setText(nhanVien.getAddress());  
 }  
 });  
 btn\_them=findViewById(R.id.btn\_Them);  
 btn\_them.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View view) {  
 ContentValues contentValues=setContentValuesForNhanVien();  
 Uri nhanVienUri=Uri.parse(URL\_NHANVIEN);  
 Uri isInsert=getContentResolver().insert(nhanVienUri,contentValues);  
 *if* (isInsert!=*null*){  
 nhanVienList.clear();  
 select();  
 lamrong();  
 Toast.makeText(getApplicationContext(), "Them thanh cong", Toast.LENGTH\_SHORT).show();  
 }  
 *else* Toast.makeText(getApplicationContext(), "them that bai", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 btn\_sua=findViewById(R.id.btn\_CapNhat);  
 btn\_sua.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View view) {  
 String nhanVienIdUrl=URL\_NHANVIEN+"/"+id\_NhanVien;  
 ContentValues contentValues=setContentValuesForNhanVien();  
  
 *int* isUpdate=getContentResolver().update(Uri.parse(nhanVienIdUrl),contentValues,*null*,*null*);  
 *if* (isUpdate>0){  
 nhanVienList.clear();  
 select();  
 lamrong();  
 Toast.makeText(getApplicationContext(), "Cap nhat thanh cong", Toast.LENGTH\_SHORT).show();  
 }  
 *else* Toast.makeText(getApplicationContext(), "Cap nhat that bai", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 btn\_xoa=findViewById(R.id.btn\_Xoa);  
 btn\_xoa.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View view) {  
 String nhanVienIdUrl=URL\_NHANVIEN+"/"+id\_NhanVien;  
 Uri nhanVienUri=Uri.parse(nhanVienIdUrl);  
  
 *int* isDelete=getContentResolver().delete(nhanVienUri,*null*,*null*);  
 *if* (isDelete>0){  
 nhanVienList.clear();  
 select();  
 lamrong();  
 Toast.makeText(getApplicationContext(), "xoa thanh cong", Toast.LENGTH\_SHORT).show();  
 }  
 *else* Toast.makeText(getApplicationContext(), "xoa that bai", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 btn\_tim=findViewById(R.id.btn\_Tim);  
 btn\_tim.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View view) {  
 nhanVienList.clear();  
 String nhanVienIdUrl=URL\_NHANVIEN+"/"+ed\_id.getText().toString().trim();  
 Uri nhanVienIdUri=Uri.parse(nhanVienIdUrl);  
  
 Cursor cursor=getContentResolver().query(nhanVienIdUri, *null*, *null*, *null*, *null*);  
 *if* (cursor!=*null* && cursor.moveToFirst()){  
 *while* (!cursor.isAfterLast()){  
 *int* id=cursor.getInt(0);  
 String name=cursor.getString(1);  
 String diaChi=cursor.getString(2);  
 *int* id\_phongBan=cursor.getInt(3);  
 *for* (PhongBan phongBan:phongBanList){  
 *if* (phongBan.getId()==id\_phongBan){  
 NhanVien nhanVien= *new* NhanVien(id, name,diaChi,phongBan);  
 nhanVienList.add(nhanVien);  
 }  
 }  
 cursor.moveToNext();  
 }  
 cursor.close();  
 }  
 List<String> list= *new* ArrayList<>();  
 *for* (NhanVien nhanVien:nhanVienList){  
 list.add("id: "+nhanVien.getId()+" -ten: "+nhanVien.getName()+" -Dia chi: "+nhanVien.getAddress()+ "-Phong ban: "+nhanVien.getPhongBan().getName());  
  
 }  
 adapter= *new* ArrayAdapter<>(NhanVienActivity.*this*, android.R.layout.simple\_list\_item\_1,list);  
 gridView.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
 }  
 });  
 btn\_thoat=findViewById(R.id.btn\_Thoat);  
 btn\_thoat.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View view) {  
 finish();  
 }  
 });  
 }  
  
 *private void* select(){  
 Uri nhanVienUri=Uri.parse(URL\_NHANVIEN);  
 Cursor cursor=getContentResolver().query(nhanVienUri, *null*, *null*, *null*, *null*);  
 *if* (cursor!=*null* && cursor.moveToFirst()){  
 *while* (!cursor.isAfterLast()){  
 *int* id=cursor.getInt(0);  
 String name=cursor.getString(1);  
 String diaChi=cursor.getString(2);  
 *int* id\_phongBan=cursor.getInt(3);  
 *for* (PhongBan phongBan:phongBanList){  
 *if* (phongBan.getId()==id\_phongBan){  
 NhanVien nhanVien= *new* NhanVien(id, name,diaChi,phongBan);  
 nhanVienList.add(nhanVien);  
 }  
 }  
 cursor.moveToNext();  
 }  
 cursor.close();  
 }  
 List<String> list= *new* ArrayList<>();  
 *for* (NhanVien nhanVien:nhanVienList){  
 list.add("id: "+nhanVien.getId()+" -ten: "+nhanVien.getName()+" -Dia chi: "+nhanVien.getAddress()+ "-Phong ban: "+nhanVien.getPhongBan().getName());  
  
 }  
 adapter= *new* ArrayAdapter<>(NhanVienActivity.*this*, android.R.layout.simple\_list\_item\_1,list);  
 gridView.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
 }  
 *private void* lamrong(){  
 ed\_name.setText("");  
 ed\_id.setText("");  
 ed\_diaChi.setText("");  
 }  
 *private void* load(){  
 Cursor cursor=getContentResolver().query(Uri.parse(URL\_PHONGBAN),*null*,*null*, *null*,*null*);  
 *if* (cursor!=*null* && cursor.moveToFirst()){  
 *while* (!cursor.isAfterLast()){  
 *int* id=cursor.getInt(0);  
 String name=cursor.getString(1);  
 PhongBan phongBan= *new* PhongBan(id,name);  
 phongBanList.add(phongBan);  
 cursor.moveToNext();  
 }  
 cursor.close();  
 }  
 List<String> list= *new* ArrayList<>();  
 *for* (PhongBan phongBan:phongBanList){  
 list.add(phongBan.getName());  
 }  
 adapter= *new* ArrayAdapter<>(NhanVienActivity.*this*, android.R.layout.simple\_list\_item\_1,list);  
 spinner.setAdapter(adapter);  
 adapter.notifyDataSetChanged();  
 }  
 *private* ContentValues setContentValuesForNhanVien(){  
 ContentValues contentValues= *new* ContentValues();  
 contentValues.put("id", Integer.parseInt(ed\_id.getText().toString().trim()));  
 contentValues.put("name", ed\_name.getText().toString().trim());  
 contentValues.put("address", ed\_diaChi.getText().toString().trim());  
 String temPhongBan=spinner.getSelectedItem().toString();  
  
 *for* (PhongBan phongBan:phongBanList){  
 *if* (phongBan.getName()==temPhongBan)  
 id\_PhongBan=phongBan.getId();  
 }  
 contentValues.put("id\_PhongBan",id\_PhongBan);  
 *return* contentValues;  
 }  
  
}

-----------------------------------------------------------cấp quyền-------------------------------------------------------------------------

<uses-permission android:name="vn.com.linh.appa\_provider3.WRITE\_DATABASE"/>  
<uses-permission android:name="vn.com.linh.appa\_provider3.READ\_DATABASE"/>