INTERLIS Relations in QGIS

How INTERLIS associations and inheritances are handled as QGIS relations

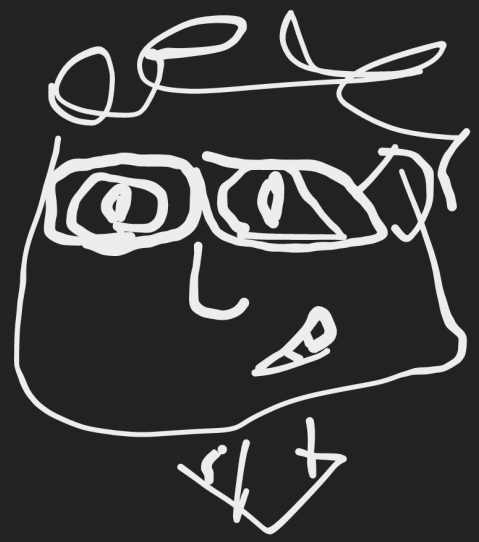
Andreas Neumann

Kanton Solothurn

QGIS PSC Member

Super QGIS Users

QGIS Model Baker Group



David Signer

OPENGIS.ch

Software Engineer

QGIS Core Comitter

QGIS Model Baker Coordinator

Workshop

- INTERLIS Classes
- INTERLIS Associations
- Check the INTERLIS Model
- Inheritancs in Physical Models
- Associations in QGIS

INTERLIS Classes

Classes

Syntax

```
ClassDef =  'CLASS' Class-Name '='
                { AttributeDef }
            'END' Class-Name ';'.
```

Example

```
CLASS Building =
    Name : TEXT*20;
    Nr_of_Floors : MANDATORY 1 .. 100;
END Building;
```

Structures

Syntax

```
StructureDef =  'STRUCTURE' Struct-Name '='
                    { AttributeDef }
                'END' Struct-Name ';'.
```

Example

```
STRUCTURE Address =
    StreetName : TEXT*40;
    Number : TEXT*12;
END Address;

CLASS Building =
    Position : Address;
END Building;
```

```
CLASS Building =
    Position : BAG {0..*] OF Address;
END Building;
```

Types of classes

- Concrete
- Abstract
- Final
- Derivate/Extended

Concrete Classes

```
CLASS Building =
END Building;

CLASS Office_Building
    EXTENDS Building =
END CLASS Office_Building;
```

Abstract Classes

```
CLASS Building (ABSTRACT) =
END Building;

CLASS Office_Building
    EXTENDS Building =
END CLASS Office_Building;
```

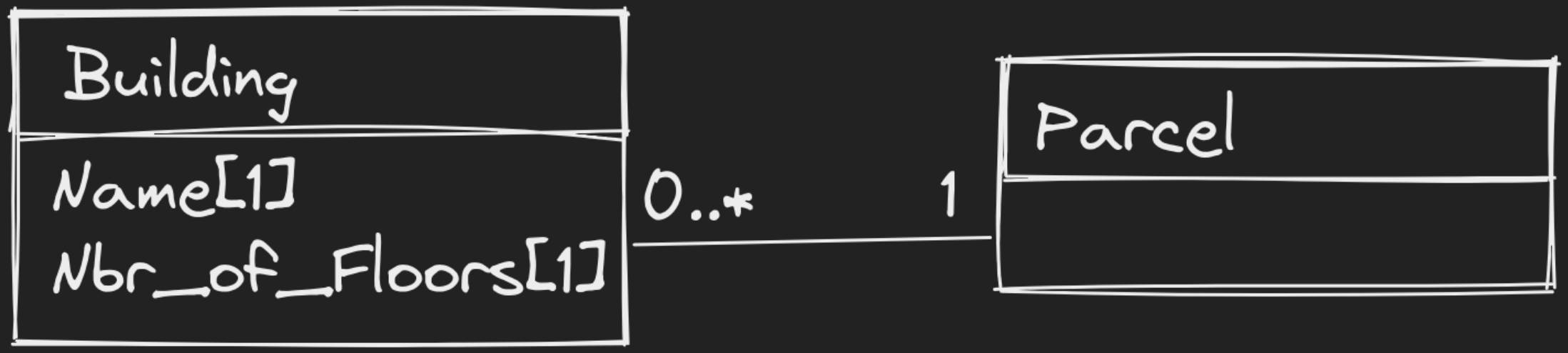
Final Classes

```
CLASS Building (FINAL) =
END Building;

CLASS Office_Building
    EXTENDS Building =
END CLASS Office_Building;
```

```
!! Error: Building cannot be extended
```

INTERLIS Associations



Cardinality

one-to-many

A building belongs to exactly one parcel. A parcel can have multiple buildings.

```
ASSOCIATION =
    local_buildings -- {0..*} Building;
    parcel -- {1} Parcel;
END;
```

many-to-many

A building belongs to exactly one parcel. A parcel can have multiple buildings.

```
ASSOCIATION Property =
    Person -- {1..*} Person;
    Parcel -- {0..*} Parcel;
END;
```

Strength

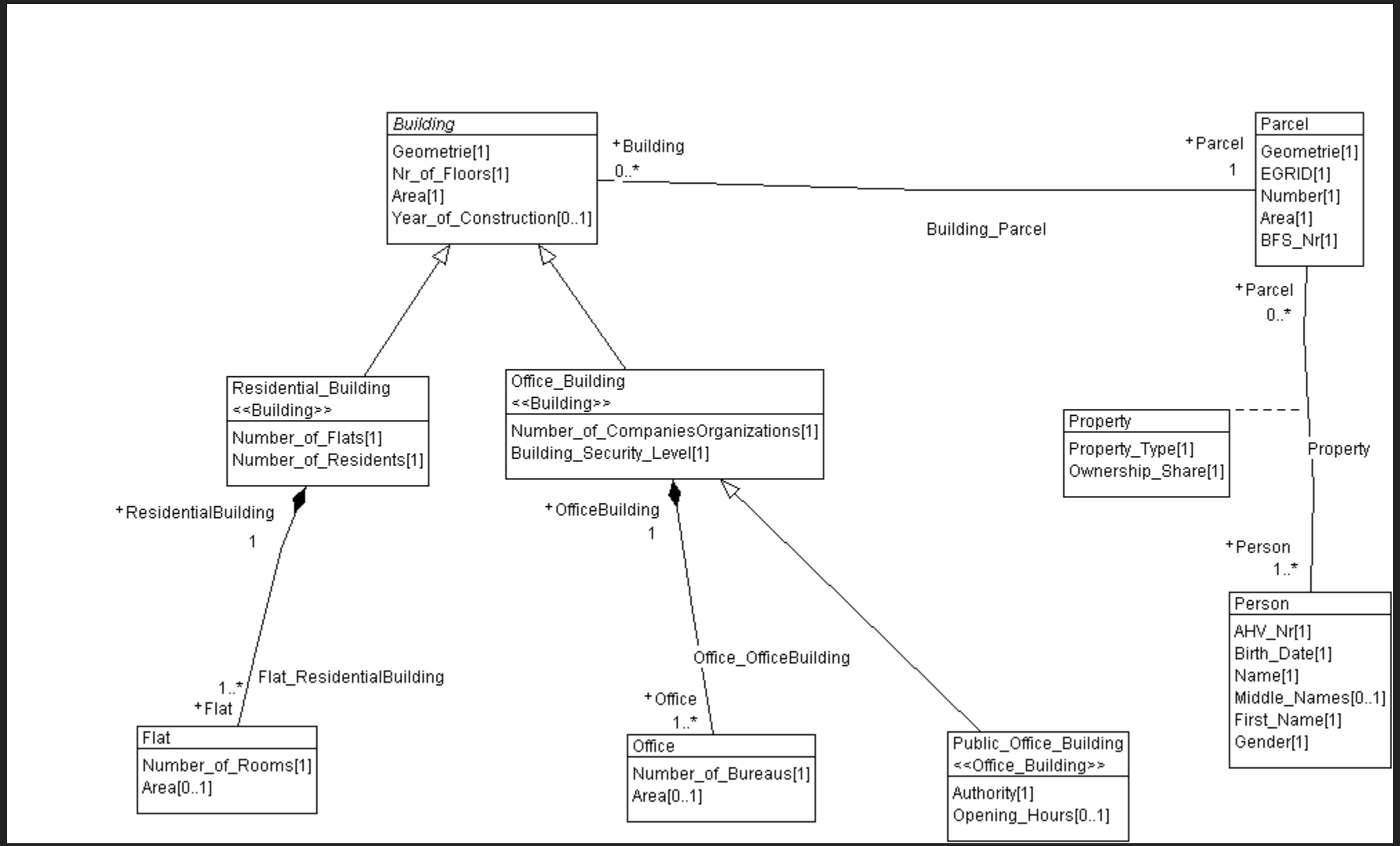
- Association `--`: Relationship between independent objects.
- Aggregation `-<>`: Relationship between parts and a whole. A part can be part of multiple wholes.
- Composition `-<#>`: Relationship between parts and a whole. A part can only be part of a single whole.

Attributes

An association can contain attributes as well.

```
ASSOCIATION Property =
    Person -- {1..*} Person;
    Parcel -- {0..*} Parcel;
    Ownership_Share : 0 .. 100;
END;
```

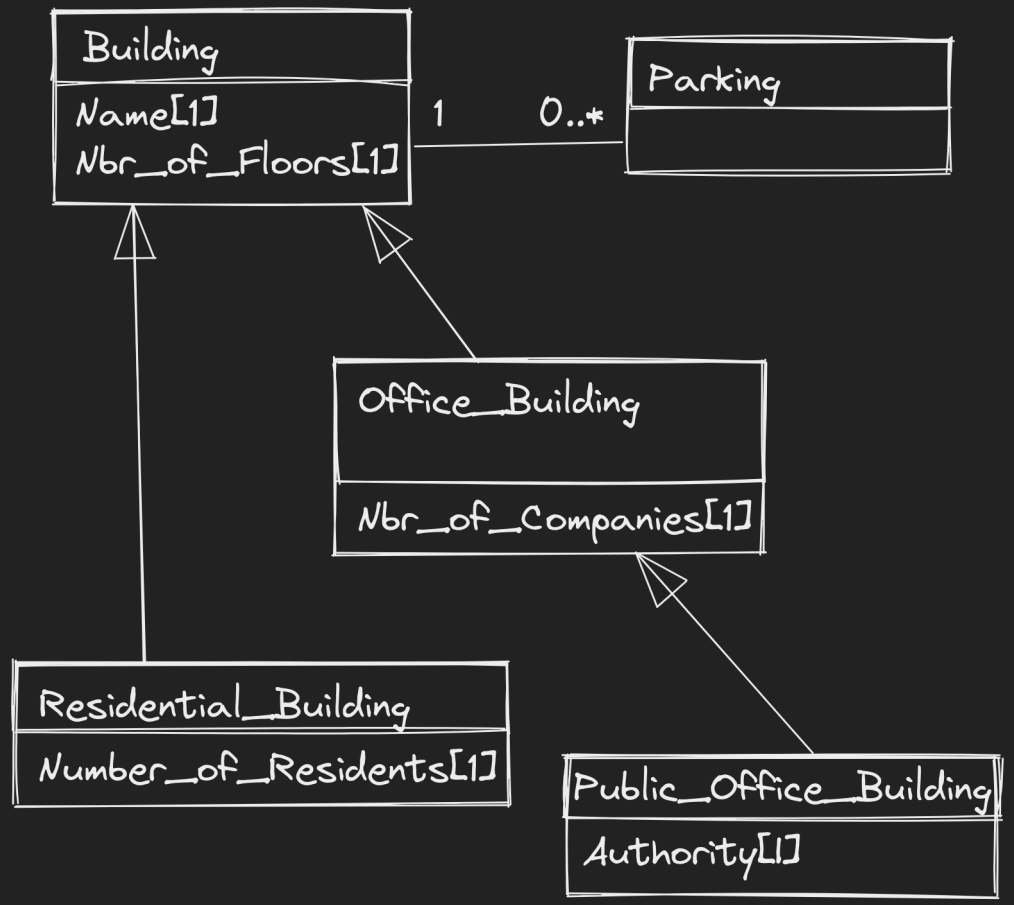
Example Model `Building_Parcel_Property`



Inharitance mapping

- New Class
- Super Class
- Sub Class
- New+Sub Class

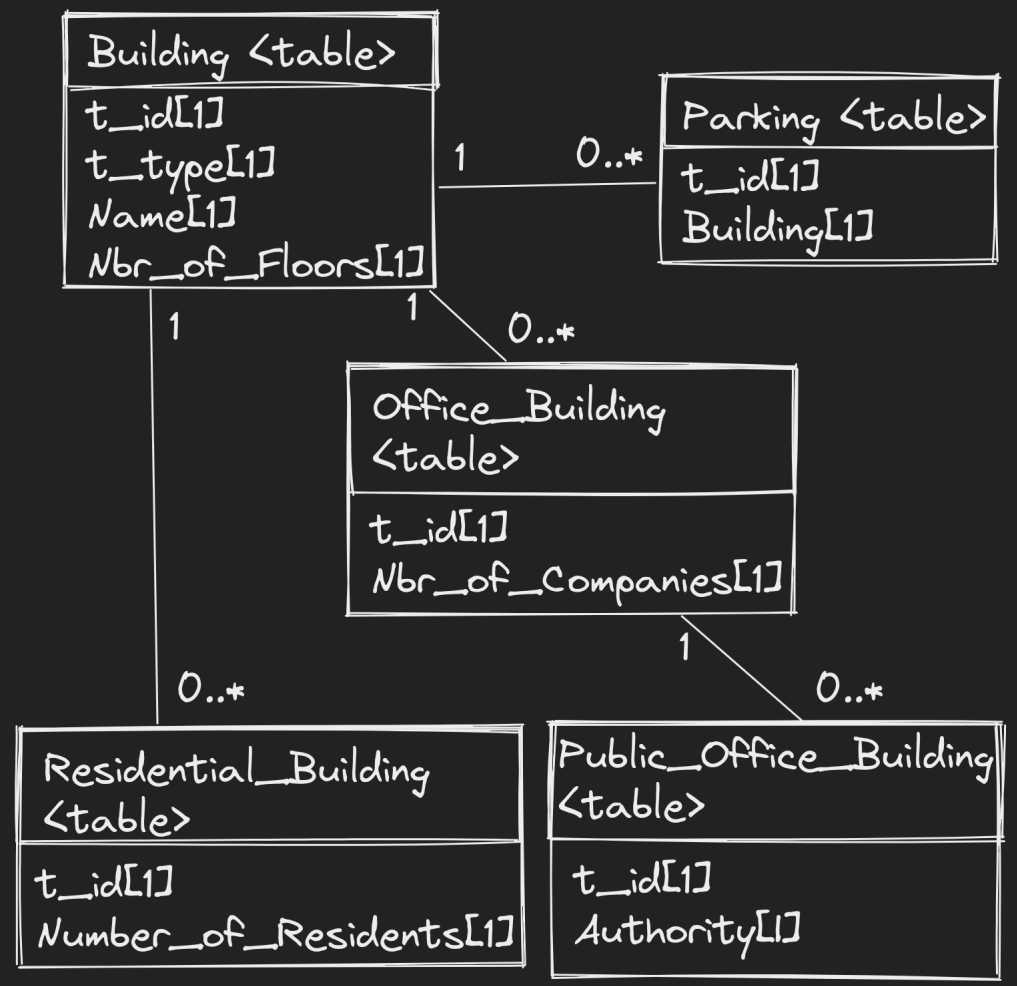
Sample Model



New Class

```
Building.t_type: (
    Residental_Building,
    Office_Building,
    Public_Office_Building
)
```

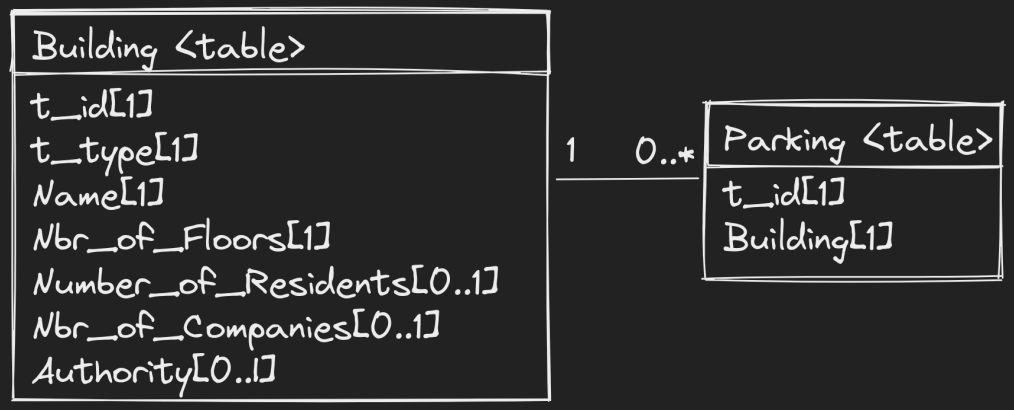
- Specializations are mapped as associations
- Multiple inserts and updates required per object
- Not null attributes can be setted
- Referential integrity is respected



Super Class

```
Building.t_type: (
    Residental_Building,
    Office_Building,
    Public_Office_Building
)
```

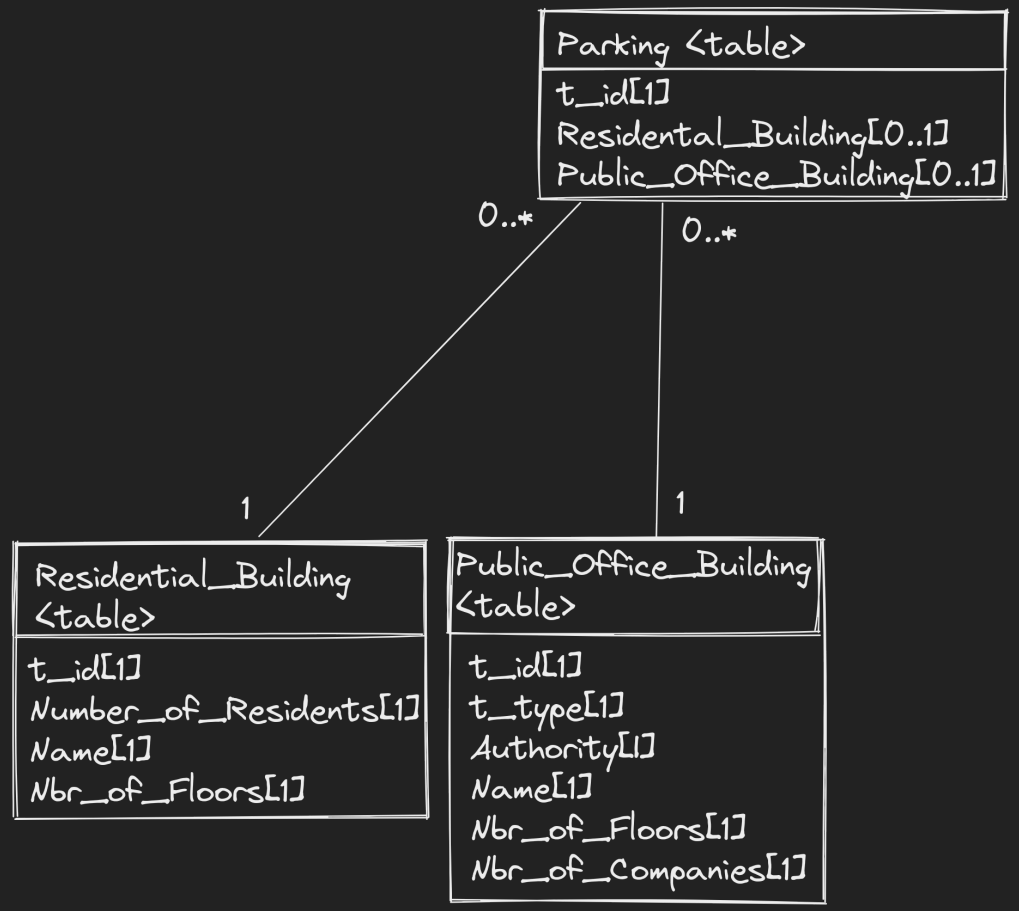
- Missing not null constraints
- Less tables and associations (easy to use)



Sub Class

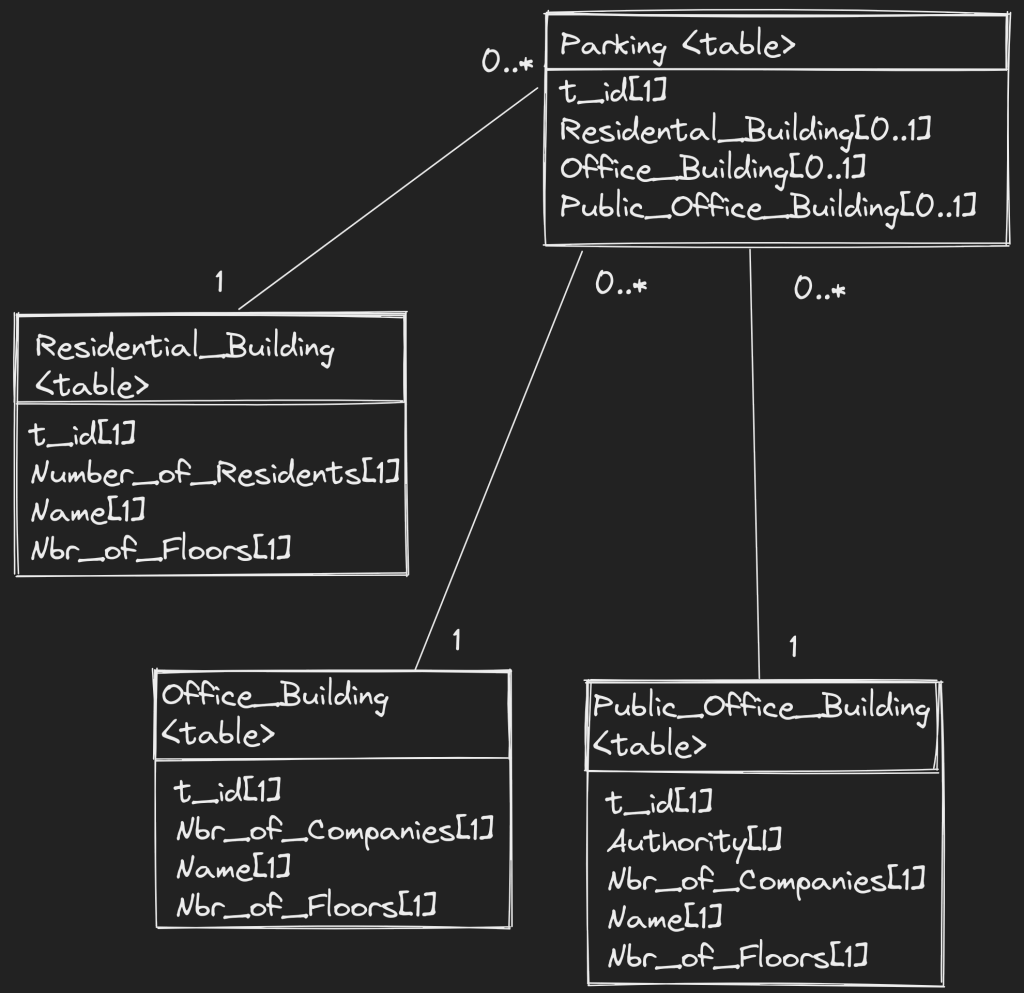
```
Public_Office_Building.t_type: (
    Office_Building,
    Public_Office_Building
)
```

- Missing not null constraints



New + Sub Class

- Missing not null constraints
- Referential integrity is respected



Smart Mapping in ili2db

noSmartMapping

- All classes are mapped using *New Class* strategy

smart1Inheritance

- Abstract classes without associations -> *Sub Class* strategy
- Abstract classes with associations and no concrete super class -> *New Class* strategy
- Concrete classes without concrete super class -> *New Class* strategy
- All other classes -> *Super Class* strategy

smart2Inheritance

- Abstract classes -> *Sub Class* strategy
- All concrete classes -> *New* + *Sub Class* strategy

Relations in QGIS