ICT Project Guidance

Design :   
Technical -- Item Grouping

Version:

0.1

Author:

Sky Sigal, Solution Architect

## Description

Grouping is a fundamental means of organising information, that has sufficient impact that it deserves consideration in of itself, separate from other design concerns.

## Synopsis

Items can be grouped by association or dynamic based attribute, permitting filtering, sorting, working on, etc.

## Contents

[Description 1](#_Toc153788863)

[Synopsis 1](#_Toc153788864)

[Contents 2](#_Toc153788865)

[Introduction 3](#_Toc153788866)

[Relationship Approaches 4](#_Toc153788867)

[Direct Linking 4](#_Toc153788868)

[Indirect Linking 4](#_Toc153788869)

[Dynamic Linking 5](#_Toc153788870)

[Appendices 7](#_Toc153788871)

[Appendix A - Document Information 7](#_Toc153788872)

[Versions 7](#_Toc153788873)

[Images 7](#_Toc153788874)

[Tables 7](#_Toc153788875)

[References 7](#_Toc153788876)

[Review Distribution 7](#_Toc153788877)

[Audience 7](#_Toc153788878)

[Structure 7](#_Toc153788879)

[Diagrams 7](#_Toc153788880)

[Terms 8](#_Toc153788881)

## Introduction

Most items in a database are grouped.

Common examples include:

* Folders are groups of zero or more Files or other Resources.
* Departments are groups of zero or more People,
* Classrooms are groups of
* People are groups of one or more Identities,
* Invoices are groups of zero or more-line items,
* Logs are groups of events,
* Role to Permissions
* Etc.

Other groupings include:

* Persons have zero or more contact Channels associated to them,
* Etc.

# Common Group Characteristics

## Recognisability

Similar to recommendations on State values, Groups should implement an interface similar to *IHasNameAndDescription*, such that it becomes trivial to Title the group, as well as provide it with a description of its purpose.

Note:  
Some other common interfaces to consider implementing would include ones similar to the following: IHasState, IHasNaturalDisplayOrder, etc.

## Inclusivity

It’s a common pattern to record in the database’s Title & Description attributes Resource keys (e.g.: ‘@somegroup\_title’, ‘@somegroup\_description’), such that the rendering layer of the system can use it to look up the current user’s language-region appropriate resource and replace it before it is rendered.

## Applicability

Many Groups may be formed before they are usable. An example might be a school administrator setting up next semester’s classrooms, assigning teachers and learners to them before the group is visible and accessible to others.

To support this flexibility of application, consider ensuring Groups implement an interface similar to *IHasEnabledFromToUTCDateTime*, that support a general Enabled Boolean flag, and two From/To attributes for UTC datetimes in the future.

Note:  
This approach often improves security by ensuring any forgotten group (e.g. the “Happy Potter Badminton Team”) is not left open ended to maybe become a security weakness, and is disabled unless it is extended.

# Relationship Approaches

Items can be made members of a group in one of several ways, each with a number of different advantages and disadvantages.

### Direct Linking

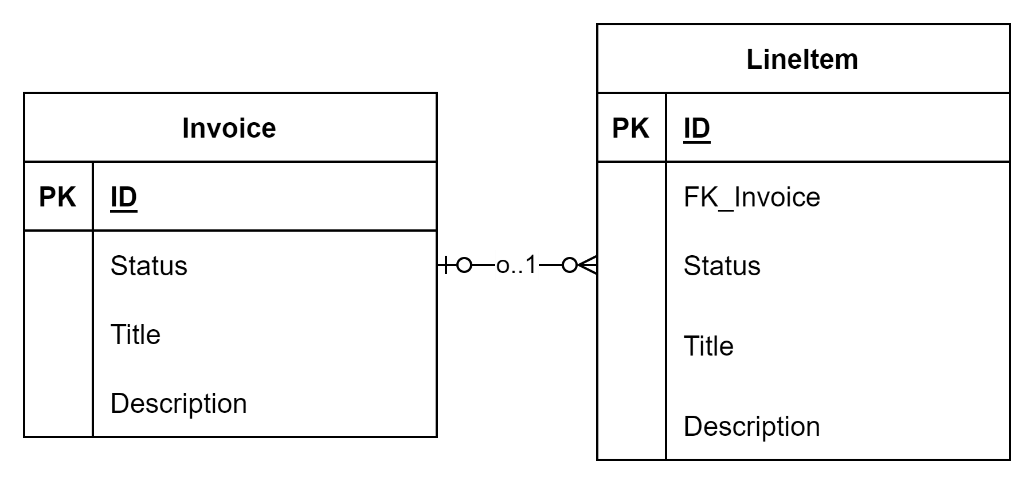


Figure : Direct Linking

An Item can be linked to a parent group by adding an FK property to the child item, pointing back to the Id of the parent Group object.

**Advantages**

This development approach is relatively simple.

Queries statements can be developed by ORMs to return the parent and child objects in one query.

**Considerations**

-

**Disadvantages**

It adds one or more properties (e.g., the FK property) to the child item that has nothing to do with the purpose of the item itself.

Note:  
the object will already have a visible storage Id that is not relevant to the business needs, so is this a real issue? Not really for internal objects which developers can be informed on to avoid – but its more of an issue for logical entities that are not mapped to data transfer objects that remove this property.

The child item can only belong to a single parent group.

The parent group ID is required at the point in time of saving the child item, implying that the parent item must be loaded into memory first.

This approach does not allow adding attributes about the relationship. For example, preferred order, enrolled from/to, etc.

The item will return child objects even if they have an attribute on them that marks them as non-relevant, requiring subsequent logic to be employed to exclude these values from being rendered.

Whereas items can be excluded from presentation, you don’t have control of remote systems, so excluded items will be sent, and one can only hope they implement logic correctly to exclude them.

### Indirect Linking

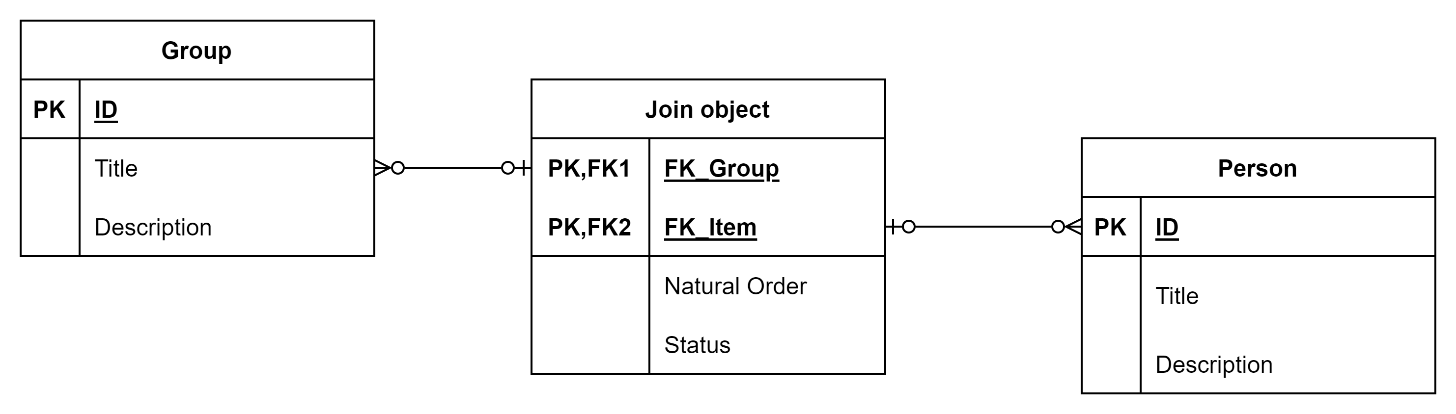


Figure : Indirect Linking

An item can be linked to a group via a intermediate joining object.

**Advantages**

Permits describing the relationship with secondary attributes. Examples include describing the state, natural order of items, from/to, or similar -- without having to pollute child items with attributes that are not part of the business purpose. This provides a lot of.

Permits associating items to multiple groups without having to change the shape or constraints of the child item.

Queries statements can be developed by ORMs to return the parent and child objects in one query.

**Considerations**

The development of a join table in a database is slightly more complex than a direct relationship.

**Disadvantages**

It introduces an object in between the parent object and the child object that can be hidden when displayed, but should be kept when transmitting in between systems, making it slightly more complex for downstream systems to understand how to import the information.

### Dynamic Linking

Grouping can be dynamic, assembling items based on rules.

A simple rule can be selecting items that have been tagged in a certain way. More complex rules could include items that are within certain time constraints, etc.

**Advantages**

State is dynamic and this approach allows items to be dynamically associated based on rules.

**Considerations**

The attributes, or tags, should not be part of the child item, to not pollute the object with properties that have nothing to do with its business purpose.

But nor is it best practice to link a child item to a descriptor directly (i.e. FK based).

It’s often best to do it indirectly (i.e., using an item descriptor that has a reference to the child item and well as uses join objects to descriptor properties).

**Disadvantages**

As there is no defined (therefore discoverable) relationship between the parent group and child items, getting a group and its items is a two-step operation.

Appendices

Appendix A - Document Information

### Versions

* 1. Initial Draft
  2. Change to Diagrams and other small changes.

### Images

[Figure 1: Direct Linking 5](#_Toc153790610)

[Figure 2: Indirect Linking 6](#_Toc153790611)

### Tables

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
| Amy Orr, Data Domain Architect |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Terms

Refer to the project’s Glossary.

##### IT

: acronym for Information, using Technology to automate and facilitate its management.

##### ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. IT is a subset of ICT.

##### Join Object

: the modelling of a relational join table.

##### Join Table

: a many to many table, containing at least 2 Foreign keys, one to each object. In many cases, the join table can cross relate 3 or tables (e.g.: User to Role to Resource).

##### ORM

: acronym for *Object Relational Mapping*.

##### Object Relational Mapping (ORM)

: a technique for converting data between a relational database and logic tier POCO entities.

##### POCO

: acronym for *Plain Old Class Objects* (see POJO).

##### POJO

: acronym for *Plain Old Java Objects*

##### Plain Old Class Objects (POCO)

*:* seen in .NET CLR based system designs, where standard objects, devoid of logic, are used to manage system resources.

Plain Old Java Objects (POJO*)*: seen in Java based system designs, with the same purpose as POCO.