



## Composer

- Composer is a subset of Person
- Problems are created by a Composer, not any Person.

## Problem

- The notion of an Independent Problem (one that exists without a Composer to create it) is absurd
- The use of surrogates on initially-perceived entities cripples the modelling process: Problem is not an Independent fact
- Thus Problem is **Identified** by its ComposerId, and a sequence of ProblemNo within that
- The Alternate Key Name ensures that each Problem is unique.

## Composer\_2

- Somewhat after the discussion, the model was completed to fulfil the full original requirement. Since the second Composer is "rare", an Associative table for Composers (plus a Constraint that calls a Function that limits it to two Composers) is not reasonable.
- A simple CHECK Constraint is required to ensure the Composers are not the same.

## Solution

- The ComposerId and the SolverId now appear in each row
- Thus a simple CHECK Constraint can be used, and the Function (required for checking other rows) is eliminated

## Entity Type

- Identifying Entity
- Transaction
- TransactionDetail

 IDEFIX Notation  
DDL

# Bridge



This page is required to overcome a document production bug