Purpose
This is a case study of a database that is:
• a faithful implementation of the Relational Model
  (Dr E F Codd, not the pretenders)
• produced using the IDEF1X Methodology & Standard, plus our Extensions
• correctly Normalised to Fifth Normal Form: devoid of Update Anomalies & Nulls; all Functional, Multi-Valued & Join Dependencies have been resolved
• almost all Business Rules implemented as Declarative Constraints (no code dependencies)
• demonstrates the correct use of Relational Keys
• in a PDF document that is complete with cross-references (select an item for further detail)
• contains the minimum level of complexity for education re Relational Database Design in general, and the highlighted subjects in particular.

Domain
• Is shown to the extent possible in IDEF1X
• Coloured relationship lines further identify the Domains, the Relational Sets, within the limits of IDEF1X. Black is used for the unimportant sets.
• The Hierarchical Layout Extension is employed here.
Relational Key
- Natural Keys, Relational Integrity and power is retained
- Surrogates always constitute:
  - an additional column & index (the natural Key provides row uniqueness)
  - a breach of the Relational Model: Keys must be made up from the data
- Further, except when assigned at the top of a Data Hierarchy (eg. PartyTo), they (eg. AddressAsx) constitute:
  - a breach of the Relational Model: Access Path Independence, and
  - Relational Integrity and power is lost at that location.
- Only two surrogates are used: AddressAsx (on this page) and PartyTo.
- County and Suburb are optional (standard Null Substitute)
- These Reference tables are fully loaded from Post Office data, both data entry and spelling errors are eliminated
- Only AddressAsx needs to be added for new addresses, requiring a valid Street row
- Null values are not stored. Two techniques are given for optional columns:
  - Columnar Normalisation such that Nulls are eliminated ( AssetSerial)
  - County and Suburb (this page) using Null Substitutes
- Views are, by definition (a) derived from tables, thus (b) they are the flattened or ‘denormalised’ form:
  - as such, they contain Nullable columns.
- Developers and “theoreticians” typically implement views as “tables”
  - SG extensions: Hierarchical Layout; etc.
- Physical-only columns ( eg. TimeStamp, UserStamp) not shown

Context
This document was initially placed in the public domain, as a real example, that I could refer to, in order to facilitate robust discussions about the subject matter on the sites that I attend. Evidently, the document is being downloaded and used beyond that. While I welcome that, there is a problem with it: the context is absent. Nothing can be studied effectively when it is removed from its context. Therefore, additional notes are warranted.
- The Data Model was developed as a case study which is used in several different courses. There is a separate version of this model for each stage of Data Modelling/Normalisation
- This document is not the education, it is not the case study, it is only the data model of the case study of the course.
- This version/stage of the Data Model is Normalised correctly, up to SNF, allowing the remaining steps to proceed unhindered
- The level/stage of Normalisation is higher than that which most designers achieve, and therefore it was chosen for the said discussions.
- It is not Normalised completely: two steps remain, before full DKNF ( the full intention as per the Relational Model, not the deranged definition) is achieved.
- As per my custom, every version/stage of the model retains two mistakes, for participants to detect, and to resolve, in the next version/stage.
Order DM Advanced
Party & Object

PartyType
- PartyTypeCode
  - name

CreditRating
- CreditRatingCode
  - Name
  - Description

Corporation
- CorporationNo
  - CreditRatingCode
  - CreditLimit
  - MaxOverdue
  - IncorporationDate

Individual
- IndividualId
  - BirthDate

Object
- ObjectCode
  - ObjectTypeCode
  - Name

Part
- PartCode
  - UnitsPerQty
  - QtyInStock
  - QtyBackOrder
  - Price

Asset
- AssetCode
  - PurchasePrice
  - PurchaseDate
  - ItemNo
  - SerialNo

Assembly
- AssemblyCode
  - ComponentCode
  - Quantity

AssetItem
- AssetItemNo
  - Description
  - IsObsolete
  - IsInscribed

AssetSerial
- AssetSerialNo
  - SerialNo
  - IsInscribed

Object_V
- ObjectCode
  - ObjectType
  - Name
  - FullName
  - ItemNo
  - PurchaseDate
  - PurchasePrice
  - SerialNo
  - UnitsPerQty
  - QtyInStock
  - QtyBackOrder
  - Description
  - IsObsolete
  - IsAsset
  - IsInscribed
  - IsComponent
  - IsAssembly

Entity Type
- Reference
- Identifying/Major
- Identifying
- Transaction
- TransactionDetail
- History
- Audit

DEREK ASIRVADEM • 16 Sep 12

Copyright © 2012 Software Gems Pty Ltd

Order DM Advanced • 3 of 4
Relational Key
- Natural Keys, Relational Integrity and power is retained
- This provides Data Distribution & high Concurrency
- OrderSaleNo is sequential within CustomerNo
- OrderPurchaseNo is sequential within VendorNo
- In Transaction tables such Keys serve multiple purposes
- Keys are migrated and used as Roles as per the Relational Model and IDEF1X:

Domain
- Is shown to the extent possible in IDEF1X (ie. no Domain Extension)
- Coloured relationship lines further identify the Domains, the Relational Sets, within the limits of IDEF1X. Black is used for the unimportant sets.
- The Hierarchical Layout & Collapsed Entity Extensions are employed here

Select any collapsed table to open it