Personal Name Data Standard

Draft Proposal

For Discussion Purposes Only

Version 0.03

"NEWORG"

2012 01 02

Contents

01 Introduction

01.01 Purpose

This data standard covers Personal Name, hereinafter referred to as PersonName. Although this data standard attempts to encompass all cultures and languages throughout the world, it is not intended to be exhaustive, i.e., address every level and range of complexity that exists. Rather, it provides solutions to the most common cultural usages in a flexible generic extensible manner. It is targeted specifically for genealogical use, but it may be applicable for many other purposes. In addition, it is intended to provide useful input into the "NEWORG"[1] data standards.

A Personal Names Data Standard is needed to improve the quality, reliability and use of Personal Name data for genealogical purposes as:

- no comprehensive data standard exists that addresses all cultures and languages
- no standard has been approved by any genealogy organization

01.02 Scope

The scope of this data standard (see Figure 01 Logical Data Model after Chapter 03) includes:

- PersonName primary table
  - entity
  - attributes
  - derived values
- PersonName attribute's domain tables
  - entity
  - attributes
- PersonName attribute's link tables
  - entity
  - attributes
- PersonName external tables
  - entity
  - attributes

This data standard does not include any details on its external tables that are linked to PersonName, i.e. Person, Place, LocaleParameter, Note and Citation tables, as they are or will be specified in other Data Standards from "NEWORG".

This data standard primarily addresses the logical data model of the PersonName entity. It is based on the objective of obtaining maximum data structure stability by using normalized data entities represented in tables. Any relaxation of the principle of normalization, if necessary, is deferred until subsequent development of the physical data model. This design approach is appropriate regardless of the implementation technology used.

01.03 Internationalization

This standard is designed to support internationalization through Locale Parameters (County and Language Codes). Predefined values in domains would have to be defined for every Locale supported by the installation.

01.04 Design Strategy

This Data Standard is designed to incorporate and use other international (e.g. International Standards Organization (ISO) and The Unicode Consortium) or national standard that are effective and have achieved wide spread acceptance in the international market place so as to leverage our efforts. Therefor this standard references and utilizes:

- The Unicode Consortium Unicode Standard V6.0.0[2]
- ISO Information technology - Syntactic metalanguage Extended BNF[3]
- ISO Country Codes 2 Alpha[4]
- ISO Language Codes 2 Alpha[5]

01.05 Authority

This standard is being developed by the "NEWORG" Project Team. Currently "NEWORG" is not working with any genealogical or technological organization on this data standard. "NEWORG" would welcome the opportunity to do so.

01.06 Status

The status of this standard is a draft proposal. Constructive critiques and suggested improvements are welcome and can be sent by E-Mail to Neil Parker or "NEWORG".

01.07 Related Documents

This document is based on "Personal Name Data Standard Rationale" by "NEWORG" that contains the background, name conventions and objectives. In addition, for each major design issue, it contains problem, requirements, alternatives, analysis, conclusions and recommendations for the Personal Name Data Standard.

The physical model is briefly addressed in a separate document, "Personal Name Implementation Guidelines" by "NEWORG". It contains implementation guidelines for software developers to facilitate increased data structure stability, data extensibility, data integrity, data accuracy and program maintainability.

01.08 <Editorial Notes

Editorial notes are shown in enclosed angle brackets and are meant to indicate:

- a comment by the writer or reviewers
- an outstanding issue raise by the writer or a reviewer

Editorial notes will be deleted in subsequent versions.>

01.09 Copyright

<Boilerplate to be provided by "NEWORG">

02 Data Representation Grammar

This document uses a proper subset of ISO 14977 Extended Backus-Naur Form (referred to as EBNF) grammar. EBNF is a metasyntax notation used for expressing context-free grammars: that is a formal way to describe computer programming languages and other formal languages. An EBNF consist of terminal symbols and non-terminal syntax rules which are restrictions governing how terminal symbols can be combined into a legal sequence.Examples of terminal symbols include alphanumeric characters, punctuation marks, and white space characters. The EBNF defines productions rules where sequences of symbols are respectively assigned to a non-terminal.

02.01 Grammar

EBNF uses the following constructs for notation and usage:

Symbol = Terminal | Non-terminal ;

Terminal = "literal" ; (* a terminal cannot be further subdivided*)

Non-terminal = (* defined by a production rule of symbols *)

Expression = (* a finite combination of symbols including the operators defined below *)

Notation	**Usage**	**Example**
'… '	terminal representation	','
"… "	terminal representation	""
Non-terminal	non-terminal representation	FamilyName or Family_Name (Family-Name is not recommended)
=	definition	Comma = ',' ;
|	alternation	Letter = CapitalLetter | NonCapitalLetter ;
Symbol Symbol	sequence	A sequence of two Symbols separated by a Space
,	concatenation	Initial = CapitalLetter , Period ;
[…]	optionality (zero or once)	Abbreviation = Word [Period] ;
{ … }	Repetition (zero or more)	Number = Digit { , Digit } ;
(…)	grouping of expressions	CompoundName = (SimpleName | PrefixedName) , (Hyphen-Minus | Space | Null) , (SimpleName | PrefixedName) ;
;	termination of a production	SimpleName = Word ;
? … ?	for difficult definitions	Letter = ? All Unicode characters designated as letters ? ;
-	Exception	Consonant = Letter - Vowel ;
(*…*)	Comment	(* Not all abbreviations end in a period *)

The order of each element in a production rule is significant and must be preserved.

The following is an example of a syntax rule:

CompoundName = (SimpleName | PrefixedName) , (Hyphen-Minus | Space | Null) , (SimpleName | PrefixedName) ;

This production is read as:

A CompoundName is defined as either a SimpleName or a PrefixedName concatenated with either a Hyphen-Minus, a Space or a Null concatenated with either a SimpleName or a PrefixedName.

02.02 Terminology Convention

Non-terminals are written in Camel Case, i.e. a simple word is capitalized, an acronym is capitals and a compound word is joined with each part capitalized, e.g. Title, TypeID and FamilyName. The EBNF requirement that multiword terminals and non-terminals to be joined by a hyphen-minus is unfortunate. Many programming languages reserve the hyphen-minus for negation or subtraction and cannot accept a hyphen-minus in a name. The two solutions to this problem are to use camel case or to use the underscore to concatenate each part of a compound name, both alternatives allow a compound word to appear and to be treated as one word. This is very helpful in programming and using the same form in documentation facilitates clear understanding. This document uses camel case, as it is shorter and less unaesthetic.

Contrary to proper English use, variable names are not pluralized when used in text that would normal require it, e.g. the rule:

GivenNames = GivenName , { GivenName } ; is written

GivenNames is formed by GivenName then zero or more GivenName.

02.03 Term Congruency

The Personal Name Data Standard attempts to use the same term style as would be used in data modeling or programming and to be as congruent with GEDCOM 5.5 as is practical unless the GEDCOM 5.5 term is:

- misleading
- not self-descriptive
- contain an implied order bias where order may not be uniformly standardized
- are unnecessarily lengthy

Note: Terms within a table are not prefixed with their table name as this name qualification is implied.

03 General Terms

03.01 Terminals

03.01.01 Characters

This Data Standard incorporates and utilizes a subset (see Chapter 02) of ISO's EBNF for its definition and the Unicode Specifications for specification and designation of Terminals, i.e.:

- Character
- ControlCharacter
- Letter
- CapitalLetter
- Non-CapitalLetter
- Digit
- Logogram
- PunctuationMark
- Delimiter

These terms will not be defined further in this standard.

03.01.02 Strings

LineTerminator = CarriageReturn | NewLine | CarriageRetun NewLine | NewLine CarriageReturn ;

Null = '' ; (* No character *)

Illegible = '…' ; (* one or more not legible characters *)

String = [Letter | Digit | Logogram | Punctuation | Delimiter | Illegible] { , (Letter | Digit | Logogram | Punctuation | Delimiter | Illegible) } ;

Word = [Illegible] , Letter { [, Illegible] , Letter } [, Apostrophe [, Illegible] , Letter]
{ [, Illegible] , Letter , [Illegible] [Apostrophy] } ;

A word is a sequence of one or more Letters; a non-letter Apostrophe may be used once in the middle or end of a word.

One or more IIllegible may occur anywhere in a word except together, i.e. at the start, several times in the middle or at the end.

NameWord = Word ; (* a Word used as a name *)

NonNameWord = Word ; (* a Word not used as a name, e.g. de, d', , de le and van der *)

Abbreviation = Word [, Period] ; (* modern usage may drop the traditional use of a period ending*)

LogogramWord = ? Any Unicode character designated as a Logogram word ? ;

LogogramName = LogogramWord ; (* a LogogramWord used as a name *)

RomanNumeral = 'I' | 'II' | 'III' | 'IV' | 'V' | 'VI' ; (* Greater than six not used in a GenerationalTitle *)

03.01.03 Numbers and Keys

UnsignedInteger = Digit { , Digit } ;

An UnsignedInteger is a sequence one or more digits.

Key = ? A sequential unique non-reusable UnsignedInteger (not Null) for every record created in a database ? ;

PrimaryKey = Key ; (* identifies the row in this table where the PrimaryKey is declared*)

ForeignKey = Key ; (* references and identifies the row in another table *)

03.01.04 Order

Order = UnsignedInteger ;

Order is a sequential UnsignedInteger representing the order that records are presented.

Records should be ordered in a rational manner: natural (e.g. GivenName), chronological, surety, pervasiveness etc.

Order is assigned to records in a domain with the same PersonNameID starting at one for the first record and increasing by one for each subsequent record.

Order is recalculated for records of this domain that have the same PersonNameID using this method every time a record is created, deleted or reordered.

03.02 NameParts

NamePart = SimpleName | SingleInitialName | PrefixedName | CompoundName ;

A NamePart is a SimpleName, SingleInitialName, PrefixedName or CompoundName.

SimpleName = NameWord ;

A SimpleName is comprised of a NameWord.

SimpleNames may be capitalized or mixed case.

SimpleNames, except a single letter SimpleName, should not be all capitals.

<Capitalizing all letters looks unaesthetic and may lose valuable information, e.g. deVries and MacDonald.>

SingleInitialName = CapitalLetter , Period ;

A SingleInitialName is a single CapitalLetter concatenated with a Period.

A SingleInitialName is allowed instead of a full NamePart in GivenName.

A SingleInitialName is used only if the full NamePart is not known as otherwise its initial is obvious.

If a person uses his one or more initials instead of one or more name, these initial are stored as FamiliarNames and will a be presented in enclosed QuotationMark.

PrefixedNamePrefix = NonNameWord , Comma { NonNameWord , Comma } ;

A PrefixedNamePrefix consist of a NonNameWord concatenated with a comma and with zero or more NonNameWord concatenated with a Comma.

PrefixedName = PrefixedNamePrefix SimpleName ;

A PrefixedName consist of a PrefixedNamePrefix then a SimpleName, e.g., de, Vries and van, der, Zam.

<This is a simiiar the same approach taken in GEDCOM 5.5 except GEDCOM 5.5 does not use of the Comma after the PrefixedNamePrefix, introducing the possibility of ambiguity if a PrefixedName is used in a CompoundName (see below) Using a comma as a terminator for each word in a PrefixedNamePrefix allows each word to unambiguously be identified as a prefix.>

CompoundName = (SimpleName | PrefixedName) [, (Hyphen-Minus | Space)] (SimpleName | PrefixedName) , { (Hyphen-Minus | Null | Space) , (SimpleName | PrefixedName) } ;

A compound name consists of either a (SimpleName or PrefixedName), concatenated by a (Hyphen-Minus, Null, or Space) to a (SimpleName or PrefixedName) concatenated to zero or more (Hyphen-Minus, Null or Space) concatenated to a (SimpleName or PrefixedName), e.g. van, der, Zam-Smith; van, der, ZamSmith; and van, der, Zam Smith.

A SingleInitialName cannot be part of a CompoundName.

<Without the use all the Comma Space separators in the PrefixedName, a reader may not know what each NamePart is in van der Zam MacDonald or van der ZamMacDonald. Even if we write this last name as van, der, ZamMacDonald there is still ambiguity; we don't know whether the person two NameParts were van, der, ZamMac and Donald or van, der, Zam and MacDoanld. One way to address this issue would be to replace either the Null or the Space in CompoundName with the Underscore or Hyphen-Minus.>

If two NameParts of a CompoundName are concatenated by a Space or Null, the Space or Null should be replaced by an Underscore or Hyphen-Minus, e.g. Smith Jones or SmithJones would be changed to Smith_Jones or Smith-Jones.

03.03 NamePieces

NamePiece = Title | GivenName | FamiliarName | FamilyName | GenerationalTitle ;

A NamePiece is a Title, GivenName, FamiliarName, FamilyName or GenerationalTitle.

<Title and FamiliarName are not assigned nor registered at birth, marriages or death. Nevertheless all NamePiece can be useful in identifying a Person.>

03.04 Special NamePieces

03.04.01 Mandatory Fields

Only PersonNameType is mandatory.

<First GivenName might appear to be a mandatory field but if it is unknown (as indicated by its PersonNameType = Unknown), then even it is not mandatory.>

03.04.02 Exclusive Fields

Mononymic is mutually exclusive of all other NamePiece attributes by definition, i.e., if the PersonNameType is mononymic, then the first GivenName is the mononymic; all other NamePiece fields are Null.

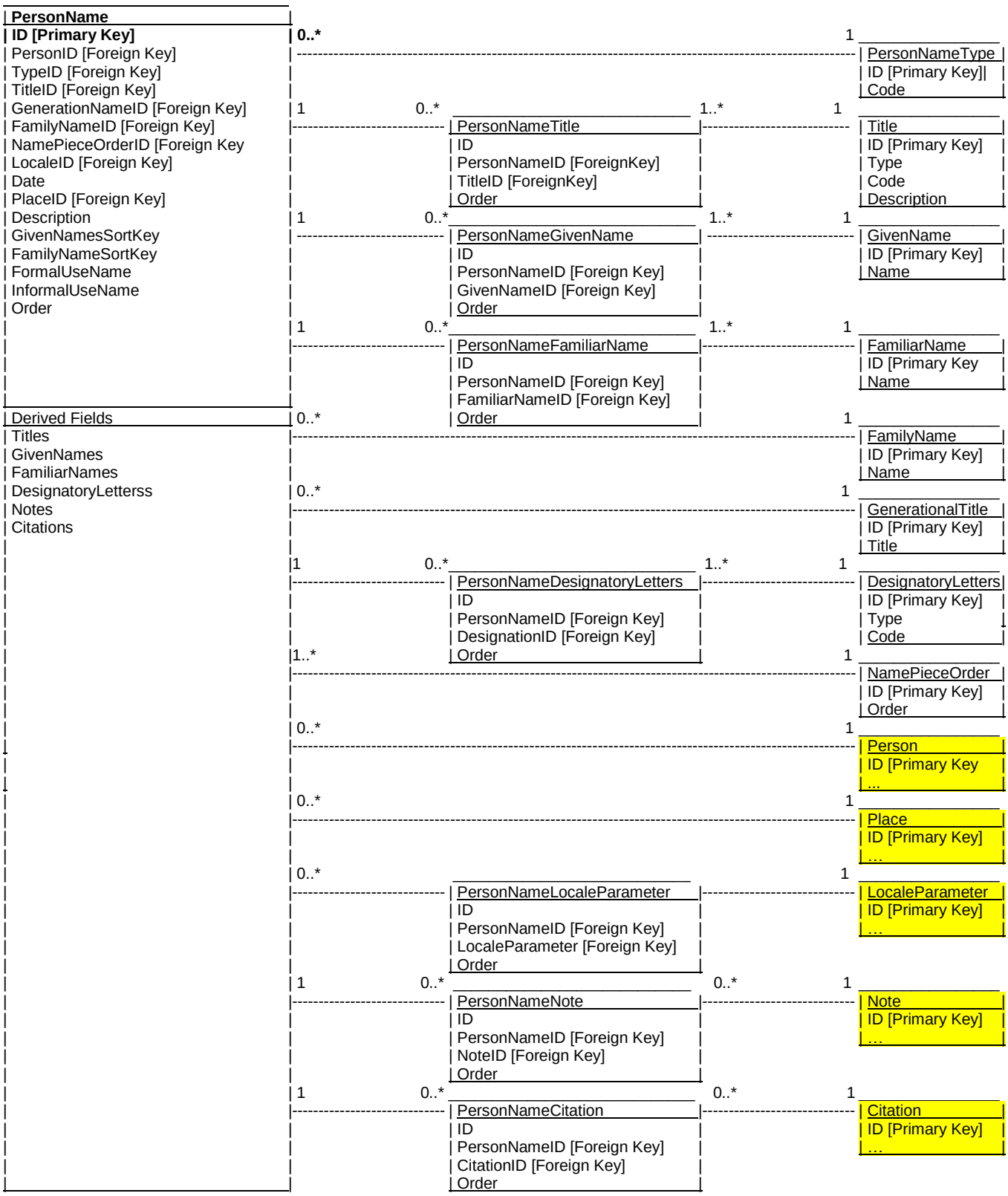
03.04.03 Included Fields

Patronymics and matronymic are a special form of one of the NamePiece fields in GivenNames (usually the last) or one of the NamePiece fields in FamilyName.

They have no special need to be specifically identified and therefore are not contained in a dedicated field.

GenerationName is an optional name used in China to identify siblings in the same family. When used, it is considered and treated as a first GivenName and therefore does not require a dedicated field.

Figure 1 - Logical Data Model



Legend: Shaded entities are defined in detail in other "NEWORG" Data Standards.

04 Primary Entity Table

See Figure 1 Logical Data Model contains the PersonName Entity - Relationship Diagram. This model gives a summary picture of the PersonName entity and its relationships to its attributes' domain tables. Many to many relationships have been converted to link tables for added clarity. All table attributes are shown in their table. Key derived fields are also shown in the PersonName table. Tables that are external to the PersonName entity are shaded and defined in other "NEWORG" data standards. They are not shown in any detail other than their table name and Primary Key.

04.01 PersonName

04.01.01 Entity

The PersonName entity is the primary entity and consists of the following attributes:

- ID
- PersonID
- PersonNameType
- FamilyNameID
- GenerationalTitleID
- NamePieceOrder
- LocaleParameterID
- Date
- Place
- Description
- GivenNamesSortKey
- FamilyNameSortKey
- FormalUseName
- InformalUseName
- Order

04.01.02 Attributes

ID = PrimaryKey ;

PersonID = ForeignKey ;

PersonNameTypeID = ForeignKey ;

FamilyNameID = ForeignKey ;

GenerationalTitleID = ForeignKey ;

NamePieceOrderID = ForeignKey ;

Date, or more likely a date range, indicates when this name was used.

A Date must comply with the "NEWORG" Date Data Standard.

Place ID = ForeignKey ;

Description = String ;

Description is a short free-format narrative describing date, place or other information about PersonName.

GivenNamesSortKey = GivenName { GivenName } ;

GivenNamesSortKey defaults to the derived GivenNames; unless overridden by a user-entered value.

FamilyNameSortKey = FamilyName ;

FamilyNameSortKey defaults to the FamilyName, unless overridden by a user-entered value.

FormalUseName = Title { , Comma Title } GivenName { GivenName } FamilyName [GenerationalTitle]
[DesignatoryLetters { , Comma DesignatoryLetters }] ;

DesignatoryLetters may be excluded in FormalUseName based on the customs of a given Locale, the occasion or individual preference.

FormalUseName can be generated automatically (perhaps based on a LocaleParameter) which becomes the initial value of this field, unless overridden by a user-entered value.

InformalUseName = GivenName FamilyName;

InformalUseName consists of the first GivenName then the FamilyName.

InformalUseName can be generated automatically (perhaps based on a LocaleParameter) which becomes the initial value of this field - unless overridden by a user-entered value.

NoteID= ForeignKey ;

CitationID = ForeignKey ;

Order = UnsignedInteger ;

04.01.03 Derived Values

Titles = [Title { Title }] ;

Titles is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

Titles consist of zero or more Title, each separated by a Space.

Titles is implemented using PersonNameTitle link table and the Title domain table to provide an ordered sequence of Titles.

GivenNames = [GivenName { GivenName }] ;

GivenNames is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

GivenNames consist of zero or more GivenName, each separated by a Space.

GivenNames is implemented using PersonNameGivenName link table and the GivenName domain table to provide an ordered sequence of GivenNames.

FamiliarNames = [QuotationMark , FamiliarName , QuotationMark { QuotationMark , FamiliarName , QuotationMark }] ;

FamiliarNames consist of zero or more FamiliarNames, each enclosed in quotation marks ("") and separated by a space.

FamiliarNames is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

FamiliarNames is implemented using PersonNameFamiliarName link table and the FamiliarName domain table to provide an ordered sequence of FamiliarNames.

DesignatoryLetterss = [DesignatoryLetters { , Comma DesignatoryLetters }] ;

DesignatoryLetterss is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

DesignatoryLetterss consist of zero or more DesignatoryLetters fields, each separated by a Space.

DesignatoryLetterss is implemented using PersonNameDesignatoryLetters link table and the DesignatoryLetters domain table to provide an ordered sequence of DesignatoryLetterss.

Notes = Note { LineTerminator , Note } ;

Notes is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

Notes consist of a Note concatenated to zero or more LineTerminator concatenated to Note.

Notes are implemented using PersonNameNote link table and the Note table to provide an ordered sequence of Notes.

Citations = Citation { LineTerminator , Citation} ;

Citations is a derived field, it is not directly contained in the PersonName record as it can occur zero to many times.

Citations consist of a Citation concatenated to zero or more LineTerminator concatenated to Citation.

Citations are implemented using PersonNameCitation link table and the Citation table to provide an ordered sequence of Citations.

05 Domain Tables

The following PersonName fields should be constrained by a corresponding domain table, e.g.:

- PersonNameType
- Title
- GivenName
- FamiliarName
- FamilyName
- GenerationalTitle
- DesignatoryLetters
- NamePartOrder
- Place
- LocaleParameter

05.01 PersonNameType

05.01.01 Entity

The PersonNameType domain table consists of the following attributes:

- ID
- Code

The PersonNameType domain table is user extensible.

05.01.02 Attributes

ID = PrimaryKey ;

Code = String ;

Code identifies the type of PersonName.

Predefined Codes are:

- Legal Name
- Birth Name
- Current Name
- Previous Name
- Married Name
- Divorced Name
- Adoptive Name
- Alias
- Also Known As
- Ecclesiastic Name
- Pen Name
- Stage Name
- Signature Name
- Familiar Name
- Formal Use Name
- Informal Use Name
- Unknown

05.02 Title

05.02.01 Entity

The Title domain table consists of the following attributes:

- ID
- Type
- Code
- Description

The Title domain table is initially empty and is user extensible.

05.02.02 Attributes

ID = PrimaryKey ;

Type = Word { Word] ;

Types consist of the following: Formal Social, Academic, Professional, Ecclesiastic, Devotional, Heads of State, Honorary, Executive, Judicial, Legislative, and Military

A Type is used primarily to help the user find his code by providing a convenient grouping of Codes but it may also be useful in determining Code presentation Order.

Code = Word | Abbreviation ;

Code is a means of identifying a person in the FormalUseName of a person.

Codes are typically abbreviations (to conserve print space) but many be written in full if unclear.

Codes consist of one or more words or abbreviations, e.g. Herr Professor Doctor; multiple words are treated as single Code, not multiple Codes and therefore are non-divisible.

Codes in the Formal Social type generally are not shown on genealogy displays, reports or charts whereas all others are.

Description = Word { Word } ;

Description is the full Text for the Code (abbreviation).

05.03 GivenName

05.03.01 Entity

The GivenName domain table consists of the following attributes:

- ID
- Name

The GivenName domain table is initially empty and is user extensible.

05.03.02 Attributes

ID = PrimaryKey ;

Name = NamePart ;

Name is a NamePart.

05.04 FamiliarName

05.04.01 Entity

The FamiliarName domain table consists of the following attributes:
- ID
- Name

The FamiliarName domain table is initially empty and is user extensible.

05.04.02 Attributes

ID = PrimaryKey ;

FamiliarName = [NamePiece] ;

A FamiliarName is an optional NamePiece.

A Familiar Name may be a derived name, preferred name, initial, nickname or a combination of the preceding.

05.05 FamilyName

05.05.01 Entity

The FamilyName domain table consists of the following attributes:
- ID
- Name

The FamilyName domain table is initially empty and is user extensible.

05.05.02 Attributes

ID = PrimaryKey ;

Name = SimpleName | PrefixedName | CompoundName] ;

Name may be either a SimpleName, PrefixedName or a CompoundName.

Name cannot be a SingleInitialName.

05.06 GenerationalTitle

05.06.01 Entity

The GenerationalTitle domain table consists of the following attributes:
- ID
- Code

A PersonName may have zero or one GenerationalTitle.

The GenerationalTitle domain table is predefined and is user extensible.

05.06.02 Attributes

ID = PrimaryKey ;

Code = ['Jr.' | 'Sr.' | RomanNumeral] ;

05.07 DesignatoryLetters

05.07.01 Entity

The DesignatoryLetters domain table contains of the following attributes:

- ID
- Type
- Code

The DesignatoryLetters is initially empty and is user extensible.

05.07.02 Attributes

ID = PrimaryKey ;

Type = 'Academic Qualifications' | 'Honorarias' | 'Esquire' | 'Professional Qualifications' | 'ReligiousOrders' ;

A Type is used primarily to help the user find his code by providing a convenient grouping of Codes and in determining Code presentation Order.

Code = CapitalLetter [, Period] { , Letter [, Period] } ;

Code uses only letters, usually all capitals without periods, e.g. MBA, PhD, CA and MD.

05.08 NamePieceOrder

05.08.01 Entity

The Name Piece Sequence domain table consists of:

- ID
- Sequence

The NamePieceOrder domain table predefined and user extensible.

05.08.02 Attributes

ID PrimaryKey ;

Sequence = 'Title, GivenNames, FamiliarNames, FamilyName, GenerationalTitle, DesignatoryLetterss' |
'FamilyName, GivenNames, FamiliarNames, GenerationalTitle, Title, DesignatoryLetterss' |
'Title, FamilyName, GivenNames, FamiliarNames, GenerationalTitle, DesignatoryLetterss' ;

NamePieceOrder represents the sequence that each NamePiece field occurs in each PersonName that is to be displayed, printed, or used on verbal or written communication.

NamePieceOrder consist of six sequenced NameParts literals:

- Title
- GivenNames
- FamiliarNames
- FamilyName
- GenerationalTitle
- DesignatoryLetterss

05.09 Place

05.09.01 Entity

The Place domain table consists of:

- ID
- …

The Place domain table is initially loaded with predefined values and is user extensible.

05.09.02 Attributes

ID PrimaryKey ;

… is defined in and must conform to the "NEWORG" Place data standards.

A Place is the place where the name was used during that date range.

<A Place is defined by zero or more prevailing government, religious or other authorities at the time and the Place names are those used at the Date in the PersonName record.

Place domain table records are hierarchical and temporal and hence sensitive to Date.>

05.10 LocaleParameter

05.10.01 Entity

The LocaleParameter entity domain table consists of:

- ID
- …

05.10.02 Attributes

ID = PrimaryKey ;

… is defined in "NEWORG" LocaleParameter Data Standard.

<LocaleParameter domain table could consist of CountryCode, LanguageCode, ParameterName, DataType, ParameterValue and Order. If data type is always a character literal, it can be implied and is therefore not needed in the table.

Locale could contain configuration attributes for each Locale (Language-Country) for the following ParameterName such as:

- WritingDirection
- CharactersAllowed
- LetterSortOrder
- Non-LetterCharactersAllowedInName
- LogogramsAllowed
- PrefixedNamesAllowed
- ValidPrefixedNamePrefixs
- CompoundNamesAllowed
- MaximumNumberOfNamePartsInCompoundName
- CompoundNameJoinerCharactersAllowed
- NamePartOrder
- FormalUseNameFormat
- InformalUseNameFormat

This facilitates the Personal Name Data Standard's ability to support Internationalization more flexibly.>

06 Link Tables

06.01 PersonNameTitle

Title can be in a many to many relationship with PersonName.

06.01.01 Entity

The PersonNameTitle link table consists of the following attributes:

- ID
- PersonNameID
- TitleID
- Order

06.01.02 Attributes

ID = PrimaryKey ;

PersonNameID = ForeignKey ;

TitleID = ForeignKey ;

Order = UnsignedInteger ;

The order of each Title is significant and must be preserved.

Order contains the presentation order of Titles.

06.02 PersonNameGivenName

GivenName can be in a many to many relationship with PersonName.

06.02.01 Entity

The PersonNameGivenName link table consists of the following attributes:

- ID
- PersonNameID
- GivenNameID
- Order

06.02.02 Attributes

ID = PrimaryKey ;

PersonNameID = ForeignKey ;

GivenNameID = ForeignKey ;

Order = UnsignedInteger ;

The order of each GivenName is significant and must be preserved.

Order contains the presentation order of GivenNames.

06.03 PersonNameFamiliarName

FamiliarName can be in a many to many relationship with PersonName.

06.03.01 Entity

The PersonNameFamiliarName link table consists of the following attributes:

- ID
- PersonNameID
- FamiliarNameID
- Order

06.03.02 Attributes

ID = PrimaryKey :

PersonNameID = ForeignKey ;

FamiliarNameID = ForeignKey ;

Order = UnsignedInteger ;

The order of each FamiliarName is significant and must be preserved.

Order contains the presentation order of FamiliarNames.

06.04 PersonNameDesignation

Designation can be in a many to many relationship with PersonName.

06.04.01 Entity

The PersonNameDesignation link table consists of the following attributes:

- ID
- PersonNameID
- DesignationID
- Order

06.04.02 Attributes

ID = PrimaryKey ;

PersonNameID = ForeignKey ;

DesignationID = ForeignKey ;

Order = UnsignedInteger ;

The order of each Designation is significant and must be preserved.

Order contains the presentation order of Designations.

06.05 PersonNameLocaleParameter

LocalParameter can be in a many to many relationship with PersonName.

06.05.01.01 Entity

The PersonNameLocaleParameter link table consists of the following attributes:

- ID
- PersonNameID
- LocaleParameterID
- Order

06.05.02 Attributes

ID = PrimaryKey ;

PersonNameID = ForeignKey ;

LocaleParameterID = ForeignKey ;

Order = UnsignedInteger ;

The order of each LocaleParameter is significant and must be preserved.

Order contains the presentation order of LocalParameter fields.

06.06 PersonNameNote

Notes can be in a many to many relationship with PersonName.

06.06.01 Entity

The PersonNameNote consists of the following attributes:

- ID
- PersonNameID
- NoteID
- Order

06.06.02 Attributes

ID = PrimaryKey ;

PersonNameID = ForeignKey ;

NoteID = ForeignKey ;

Order = UnsignedInteger ;

The order of each Note is significant and must be preserved.

Order contains the presentation order of Notes.

06.07 Citation

Citations can be in a many to many relationship with PersonName.

06.07.01 Entity

The PersonNameCitation consists of the following attributes:

- ID
- PersonNameID
- CitationID
- Order

06.07.02 Attributes

ID = PrimaryKey

PersonNameID = ForeignKey ;

CitationID = ForeignKey ;

Order = UnsignedInteger ;

The order of each Citation is significant and must be preserved.

Order contains the presentation order of Citations.

07 External Entity Tables

07.01 Person

Each PersonName entity is used by one Person entity and one Person entity may have zero or more PersonName Entities.

07.01.01 Entity

The Person external table consists of:

- ID
- …

07.01.02 Attributes

ID = PrimaryKey ;

… is defined in and must comply with "NEWORG" Person Data Standard.

07.02 LocaleParameter

LocaleParameter can be in a many to many relationship with PersonName.

07.02.01 Entity

LocaleParameter external table consists of:

- ID
- …

07.02.02 Attributes

ID = PrimaryKey ;

…. … is defined in and must comply with "NEWORG" Locale Data Standard

07.03 Place

07.03.01 Entity

Place external table consists of:

- ID ;
- …

07.03.02 Attributes

ID = PrimaryKey ;

…. … is defined in and must comply with "NEWORG" Place Data Standard

07.04 Note

Notes can be in a many to many relationship with PersonName.

Note is a free-form narrative with more extensive elaboration of Date, Place or Description about PersonName.

07.04.01 Entity

The Note external table consists of:

- ID
- …

07.04.02 Attributes

ID = PrimaryKey ;

… is defined in and must comply with "NEWORG" Note Data Standard

07.05 Citation

Citations can be in a many to many relationship with PersonName.

07.05.01 Entity

The Citation external table consists of:

- ID
- …

07.05.02 Attributes

ID = PrimaryKey ;

… Citations are specified in and must comply with the "NEWORG" Citation Data Standard.

The Citation data Standard includes Citations, Sources, and Repositories.

Glossary

This glossary is presented in non-alphabetical order to facilitate understanding of each term, as certain terms are dependent on previously defined terms.

Genealogy Name Terms

PersonName: is a data entity consisting of the following attributes: PersonID, PersonNameType, identification attributes and non-identification attributes. PersonID references the Person who uses this PersonName. Identification attributes are an ordered sequence of an optional Title, zero or more GivenName, zero or more FamiliarName, an optional FamilyName, an optional GenerationalTitle and zero or more DesignatoryLetters used to identify a person (not necessarily uniquely). Historically, and in everyday use, not all of these components are always used, even on legal documents. Some NamePiece may be used to honor a person's father, mother, grandfather, grandmother or another ancestor. Non-identification attributes consist of following: a NamePieceOrder, optional Date, an optional Place, an optional Description, a GivenNamesSortKey, a FamilyNameSortKey, a FormalUseName and an InformalUseName. These are considered non-identification attributes, as they are not essential to identify a person. However, they are useful for other genealogical purposes such as containers for other pertinent genealogical details, indexing a person's records by GivenNames or FamilyName; and use of a PersonName in formally or informally addressing him/her.

PersonNameType: PersonNameType typically indicates the usage of the PersonName. In addition, Type may give clues to the PersonName origin or other characteristics. Frequently used PersonNameType are: Legal Name, Birth Name, Current Name, Previous Name, Married Name, Divorced Name, Adoptive Name, Alias, Also Known As, Ecclesiastic Name, Pen Name, Stage Name, Signature Name, Familiar Name, Formal Use Name, Informal Use Name and Unknown.

NamePart: A NamePart may be a SimpleName, SingleInitialName, PrefixedName or CompoundName.

SimpleName: A SimpleName is a single word or logogram used as a label for a part of a PersonName.

SingleInitialName: A SingleInitialName is a single capital letter ending in a period used as a label for one of the GivenName of a PersonName.

PrefixedName: PrefixedName is a special form of a NamePart in which the SimpleName is prefaced by a PrefixedNamePrefix consisting of one or more words each separated by a space or nothing (e.g. de Vries, van der Zam, vander Zam, or d'Hunt). A PrefixedNamePrefix is explicitly identified because, depending on culture, customary use or personal preference; these prefixes may or may not be significant in sorting FamilyNames and GivenNames.

CompoundName: A CompoundName is a special form of a NamePart when each NamePart in a FamilyName (or a GivenName) is comprised of two or more SimpleName or PrefixedName, each joined by a Hyphen-Minus, Space or Null, e.g. Smith-Jones, Smith Jones and SmithJones.

A Mononymic: A mononymic is the use of a single name (e.g. Geronimo) as a GivenName instead of GivenNames and a FamilyName.

Patronymic: A patronymic is a name based on the father's (or grandfather's) first GivenName, usually with a prefix or postfix indicating son of or daughter of. In some cultures, patronymics are often used as a last GivenName qualifier where the individuals would otherwise have the same name.

Matronymic: A matronymic is a name based on the mother's (or grandmother's) first GivenName, usually with a prefix or postfix indicating son of or daughter of. In some cultures, matronymics are often used as a last GivenName qualifier where the individuals would otherwise have the same name.

GenerationName: A GenerationName is an optional field indicating the generation of the person. China uses it primarily to indicate each generation of a family. It is considered part of the GivenNames, not part of the FamilyName.

NamePiece: A NamePiece is a Title, GenerationName, each GivenName, each FamiliarName, FamilyName, GenerationalTitle or each DesignatoryLetters used to identify a person.

Title: A Title is used to add formality to a name to include in FormalUseName of a person. A Title can be can be grouped in several types, i.e. Formal Social, Academic, Professional, Ecclesiastic, Devotional, Heads of State, Judicial, Legislative and Military. A Title can occur zero or more times, e.g. Herr Professor Doctor Schmidt and usually abbreviations are used.

GivenNames: GivenNames are the one or more NamePieces given to a person at or shortly after birth by the parents to identify the person within the family. Some cultures allow a single letter to be used in any GivenName in which case it does not end in a period. First GivenName is the first NamePiece in the GivenNames. Middle GivenNames are the second and subsequent, if present, NamePieces in GivenNames. A middle NamePiece often is in honor of an ancestor or relative.

First Name and Forename: A first name or forename term is often used instead of a first GivenName. Unfortunately, first name and forename are misleading in cultures that use "FamilyName then GivenNames" order and therefore are not used further in this document.

Initial: An initial is the first letter of a name, followed by a mandatory period (so as to not confuse it with a single letter name). Some cultures often use an initial for each of the middle GivenNames and, in some cases, for the first GivenName, therefore initials with a period are allowed for each GivenName.

FamiliarName: FamiliarNames are used in a situation where the person normally is known by other than the person's first GivenName. FamiliarNames are always enclosed in double quotation marks. It is possible to have more than one Familiar Name in which case each is enclosed in double quote marks and separated by a space. A FamiliarName may be a preferred name, a derived name, an initial, a nickname, or a combination of the preceding forms and are included in a PersonName as additional GivenName(s), e.g.

- Preferred - uses a middle GivenNames in preference to the first GivenName, e.g. William James "James" Smith
- Derived - uses a short-form for one of the GivenNames, e.g., William James "Bill" Smith
- Initials - uses initials instead of Name, e.g. William James "W. J." Smith
- Nickname - uses a name not related to or derived from any GivenName, e.g. William James "Tex" Smith
- Combination - preferred and derived from Given Name, e.g., William James "Jim" Smith

FamilyName: The NamePiece of a PersonName that is used to describe family, clan, tribal group or marital association. A FamilyName is the NamePiece inherited from the FamilyName of the father and/or the mother. Some cultures today or other cultures (prior to 1200 to 1800) never used FamilyName and hence this field is not mandatory. Numerous cultures (e.g. Spanish speaking and Portuguese speaking) typically have two or four names, respectively, as a CompoundName in their FamilyName to honor both or all four sides of their family. Today in Western countries, more married women are keeping their birth FamilyName for professional or other reasons and married couples are adopting a CompoundName to honor equally both or all four sides of their family.

Surname: A Surname is sometimes used as a synonym for the FamilyName, but is not as self-defining as FamilyName (especially in other languages or cultures) and is not used further in this document.

GenerationalTitle: A GenerationalTitle allows the use of a name qualifier such as Jr., Sr., I, II, III, etc. to distinguish between members of the same family that would have otherwise have identical names, i.e. each GivenName and FamilyName.

DesignatoryLetters: DesignatoryLetters can be classified by type; and are zero or more:

- Academic Qualifications (e.g., BA, MBA and PhD)
- Honorarias (e.g. Honorary LLD)

- Esquire (i.e. Esq)
- Professional Qualifications (e.g., MD, PEng, CPA and Atty)
- ReligiousOrders (e.g. OFM and SJ)

DesignatoryLetters can only be used when sanctioned by an appropriate authorizing organization.

NamePieceOrder: NamePieceOrder specifies the order in which all NamePieces are normally presented or spoken for this person in this person's culture. If the GivenNames are stated first and the FamilyName is stated last, this is called the Western Order. If the FamilyName is stated first and the GivenNames are state last, this is called the Eastern Order and is used in Hungary, China, Japan, Korean, Vietnam and other Eastern Asian countries. Because the order of NameParts is important and culture-sensitive, their order is determined by NamePieceOrder for purposes of presentation and communication.

Date: Date is an optional field and is usually stated as a date range in which this PersonName was used by the person.

Place: Place is an optional field where this PersonName was used by this person.

Description: Description is an optional short free-format narrative comment on the Date or Place that complements it, especially if they are not used.

GivenNamesSortKey: GivenNamesSortKey is optional and is used as an override if the GivenNames field does not provide the correct automatically generated sort key, e.g. allows a GivenNames of William James "James" Smith to be sorted as James William Smith yet still be presented as William James Smith or William James "James" Smith.

FamilyNameSortKey: FamilyNameSortKey is optional and is used as an override if the FamilyName field does not provide the correct automatically generated sort key, e.g. allows a FamilyName of "van der Zam to be sorted as "Zam" yet still be presented as van der Zam.

FormalUseName: FormalUseName often uses Title, GenerationName, all GivenNames, FamilyName, GenerationalTitle and Designations.

InformalUseName: InformalUseName usually uses only first GivenName and Family Name.

Notes: Notes are an optional free format narrative comment on Date, Place or other important explanation of a Name.

Citations: Citations are the zero or more Citations from which this information (evidence) was extracted. Citations reference sources that reference Repositories.

Order: Order defines the preferred order of PersonNames, specifically which PersonName record is considered the main record; all other PersonName records are considered alternative records.

Technology Terms

Entity Table: A primary entity table is the main table that other records in domain tables or link table reference and is the focus of a specific data sub-model.

Domain Table: A domain table is a set of all distinct permissible values of a field and is typically used to contain the value pointed to by a field reference in a primary entity such as PersonName. In addition, each row of the table may contain other values such as a description and definition of the value. Domains also facilitate the elimination of illogical data structures, the easy implementation of referential Integrity (elimination or orphan references) and can conserve storage and improve data accuracy and data integrity. Domains are dynamic, stored like other data tables and therefore easy for the user to create, retrieve, update or delete records. Predefined values are supplied by the software manufacturer and are usually immutable to the user. User-

extensible means that the user can create, read, update and delete non-predefined values. Any domain table having predefined values should be locale-sensitive to support Internationalization.

Master List: A master list is a term often used in Genealogy software in place of the information technology term domain. Unfortunately, a master list can have many meanings and therefore is not used further in this document.

Link Table: Link Tables are used to store the foreign key of each entity that is references in a many to many relationship. In addition, they may store other attributes such as Order.

Endnotes

[1] "NEWORG" is a nonprofit organization dedicated to developing data and other standards for genealogy and educating
people on their use.

[2] The Unicode Consortium Unicode 6.0.0

[3] International Organization for Standardization ISO 14977:1966 Extended Backus-Naur Form Standard

[4] ISO 3166-1 Country Codes (2 Alpha)

[5] ISO 639-1:1988 Language Codes (2 Alpha)