*Department of State*

**Object Definition and Naming Standard**

*Seventh Edition*

Updated: October 2010

Version: 7.0

*Prepared by IRM/OPS/SIO/APD/DM*



UNITED STATES DEPARTMENT OF STATE

BUREAU OF INFORMATION RESOURCE MANAGEMENT

SYSTEMS AND INTEGRATION OFFICE

APPLICATION Programming DIVISION– DATA MANAGEMENT BRANCH

**Revision History Summary**

|  |  |  |
| --- | --- | --- |
| **Release Number** | **Date** | **Revision Description** |
| First Edition | August 1997 | Document Object Definition and Naming Standard |
| Second Edition | February 1998 | Update Object Definition and Naming Standard |
| Third Edition | May 1998 | Update Object Definition and Naming Standard |
| Fourth Edition | January 2000 | Update Object Definition and Naming Standard |
| Fifth Edition | December 2002 | Comprehensive review of naming standard by Data Administration Working Group. Conformance with ISO 11179. See Change Summary for additional details. |
| Sixth Edition | February 2006 |  Changed named references to IRS/OPS/SIO/API/DM Branch from Data Administration to Data Management (DM)   Added Appendix B (Oracle and SQL Server - Database Naming Guidelines) and Appendix C (Document Revision History Detail)   Refer to Appendix C for detailed history of changes made to this document |
| Seventh Edition | October 2010 |  Changed named references from IRS/OPS/SIO/API/DM Branch to IRS/OPS/SIO/APD/DM   Changed DM Email address   Changed the requirement for general Naming rules   Added general naming Principal section   Deleted Acronyms and Abbreviations section of the document   Updated Naming and description section for the following: Entity, Attribute, Column, Table, View and relationship, Index and Triggers   Added New data Objects and naming standard for Constraints and Stored Procedure |
|  | January 25, 2011 | Reworded section 9 – Triggers (RMKajiru) |

**Obtaining Copies of This Manual or Submitting Comments**

**Submitting Comments**

If you have suggestions for this manual or know of any additions, deletions, or changes, please e-mail them to  [Data Management](mailto:Data%20Management) at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov) or send them to the following address:

Department of State

Data Management (IRM/OPS/SIO/APD/DM) SA-9, Room: NE 7046  
Washington, DC 20006

The memorandum should indicate the type of request (change, addition, deletion) and must specify clearly what is being requested and the reason the request should be granted. Please include the name, e-mail address, and telephone number of a contact that can provide further information if necessary.

The request will be reviewed and the originator will be notified, via e- mail or memorandum, of the action taken. IRM/OPS/SIO/APD/DM has 30 working days after receipt of a request for change, addition, or deletion to respond. If a final response cannot be provided within that time, an interim response will be issued.

If a request for change, addition, or deletion is approved, the revision will be incorporated into the next edition of this document.

**Obtaining Copies**

You can obtain a copy of this manual from DM’s web site <http://irm.m.state.sbu/sites/OPS/SIO/APD/dm/Pages/Home.aspx> or via e-mail to  [Data Management](mailto:Data%20Management) at  [[DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov)](mailto:DA@state.gov) or request additional copies in writing to:

Department of State

Data Management (IRM/OPS/SIO/APD/DM) SA-9, Room: NE 7046  
Washington, DC 20006

Table of Contents

[1. Introduction 6](#_Toc279565343)

[1.1 Purpose 7](#_Toc279565344)

[1.2 Scope 7](#_Toc279565345)

[1.3 Naming Conventions 8](#_Toc279565346)

[1.3.1 Naming Format Statements 8](#_Toc279565347)

[1.3.2 Naming Format Glossary 8](#_Toc279565348)

[1.3.3 Requirement Level Key Words 9](#_Toc279565349)

[1.3.4 Standard Business Terms and Abbreviations 9](#_Toc279565350)

[1.3.5 Business Name vs. Abbreviated Name vs. Synonym Name 10](#_Toc279565351)

[1.3.6 General principles and Naming Rules 11](#_Toc279565352)

[1.3.7 Language Specific Exceptions 11](#_Toc279565353)

[1.3.8 Invalid Name Components 12](#_Toc279565354)

[1.4 Standard Data Elements and the Enterprise Conceptual Data Model 12](#_Toc279565355)

[2. Entities 14](#_Toc279565356)

[2.1 Entity Types 14](#_Toc279565357)

[2.3 Naming Entities 14](#_Toc279565358)

[2.3.1 Entity Name Formats 15](#_Toc279565359)

[2.2 Describing Entities 15](#_Toc279565360)

[2.3.2 Prime Terms/Object Class Terms 16](#_Toc279565361)

[2.4 Entity Metadata Properties 16](#_Toc279565362)

[3. Attributes 17](#_Toc279565363)

[3.2 Naming Attributes 17](#_Toc279565364)

[3.2.1 Attribute Name Format 17](#_Toc279565365)

[3.1 Describing Attributes 17](#_Toc279565366)

[3.2.2 Class Words/Representation Terms 19](#_Toc279565367)

[3.2.3 Migrated Foreign Key Attributes 21](#_Toc279565368)

[3.3 Attribute Metadata Properties 22](#_Toc279565369)

[4. Relationships 23](#_Toc279565370)

[4.3 Naming Relationships 23](#_Toc279565371)

[4.2.1 Relationship Name Format 23](#_Toc279565372)

[4.2.2 Relationship Name Rules 23](#_Toc279565373)

[4.1 Describing Relationships 24](#_Toc279565374)

[4.3 Foreign Key Constraint on Relationship 24](#_Toc279565375)

[4.3.1 Foreign Key Constraints Name Format: 24](#_Toc279565376)

[5. Tables 25](#_Toc279565377)

[5.1 Naming Tables 25](#_Toc279565378)

[5.1.1 Name Format: 25](#_Toc279565379)

[5.2 Describing Tables 25](#_Toc279565380)

[5.3 Table Metadata Properties 25](#_Toc279565381)

[6. Columns 26](#_Toc279565382)

[6.1 Naming Columns 26](#_Toc279565383)

[6.1.1Name Format: 26](#_Toc279565384)

[6.2 Describing Columns 26](#_Toc279565385)

[6.3 Column Metadata Properties 26](#_Toc279565386)

[7. Views 27](#_Toc279565387)

[7.2 Naming Views 27](#_Toc279565388)

[7.2.1 View Name Format 27](#_Toc279565389)

[7.1 Describing Views 27](#_Toc279565390)

[7.3 View Metadata Properties 27](#_Toc279565391)

[8. Indexes 28](#_Toc279565392)

[8.1 Index Types 28](#_Toc279565393)

[8.2 Naming Indexes 28](#_Toc279565394)

[8.2.1 Index Name Formats 28](#_Toc279565395)

[8.3 Index Metadata Properties 28](#_Toc279565396)

[9. Triggers 29](#_Toc279565397)

[9.1 Naming Triggers 29](#_Toc279565398)

[9.1.1 Formatting Triggers 29](#_Toc279565399)

[9.2 Trigger Metadata 29](#_Toc279565400)

[10. Constraints 30](#_Toc279565401)

[10.1 Type of Constraints 30](#_Toc279565402)

[10.2 Naming Constraints 30](#_Toc279565403)

[10.2.1 Constraints Name Formats 30](#_Toc279565404)

[10.3 Constraint Metadata Properties 31](#_Toc279565405)

[11. Stored Procedure 32](#_Toc279565406)

[11.2 Naming Stored Procedure 32](#_Toc279565407)

[ Stored procedures performs a function, they are action oriented. Name MUST describe the function; 32](#_Toc279565408)

[ MUST use a verb as prefix to describe the work. 32](#_Toc279565409)

[ MUST use keyword of the object 32](#_Toc279565410)

[ SHOULD use Underscore to separate words 32](#_Toc279565411)

[11.2.1 Stored Procedure Name Formats 32](#_Toc279565412)

[11.3 Stored Procedure Metadata Properties 32](#_Toc279565413)

[12. Acronyms and Abbreviations 33](#_Toc279565414)

[12.1 Creating Abbreviations 33](#_Toc279565415)

[12.2 Creating Acronyms 34](#_Toc279565416)

[12.3 Candidate Term Submittals 34](#_Toc279565417)

[Appendix A. 35](#_Toc279565418)

[Invalid Entity and Attribute Name Components 35](#_Toc279565419)

[Appendix C. Document Revision 36](#_Toc279565420)

[History Detail 36](#_Toc279565421)

# 

# 1. Introduction

The Data Management (DM) Branch is responsible for establishing, maintaining, and administering the policies and procedures required to facilitate sharing data. To fulfill this function, DM must establish standards. *The Object Definition and Naming Standard* is a critical cornerstone to data and information sharing.

The definitions and naming standards detailed in this document were developed to:

* Facilitate data object sharing, data object consistency and communication among the Department’s organizations.
* Increase reliability of information stored, shared and managed by the repository tool set.
* Improves the accuracy of searches for a particular piece of data.
* Promote accessibility and understandability of information across systems.
* Improve the quality of data and application documentation.
* Assist the DM effort in eliminating data redundancy and inconsistency.
* Facilitate user access to object names and related documentation as used throughout the Department.
* Assist analysts in selecting names that are clear and represent rules of good grammar. Simplify recognition of synonyms.
* Standardize metadata collected for Standard Data Elements (SDEs)

The Department of State’s (DOS) naming standard complies with the *ISO/IEC 11179-5 Naming* *and Identification Principles for Data Elements* standard and uses terminology consistent with it.To clarify any aspect of this document, including examples, contact the Data Management Branch at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov).

The intended audience for this document includes:

* IT Project Managers who manage projects that involve the development of new systems and/or enhancements of existing IT systems.
* Data Architects/Analysts involved in developing high-level, technology-independent logical models such as data models, process models, and data-process interaction models.
* System Architects/Analysts, Application Developers, Database Administrators (DBAs), and others who wish to standardize physical data objects.
* Data Stewards responsible for managing particular classes of information enterprise-wide, and making decisions for the name, definition, and relationships of business data.

Department legacy systems will not be required to change implemented names to make those systems adhere to this standard. Data Management will support and maintain an Enterprise Metadata Repository (EMR) which will store metadata about the Department’s data resources. The EMR will provide a means to relate data resources from various structural platforms across the Department. The data structures contained in legacy databases will be populated in the EMR through the use of scanning tools that are part of the repository software. The names in legacy systems will be mapped to the Standard Data Elements (SDEs) that adhere to naming conventions outlined in this document. New systems will be developed in accordance with the naming standards outlined in this document which will facilitate re-use and integration.

## 1.1 Purpose

The purpose of this document is to provide users with a benchmark for defining objects and for creating standard object names and relate it to its business purpose.

## 1.2 Scope

This document provides standards for defining and naming logical and physical data objects and standardizing physical data objects. If a physical implementation is using tools that cannot support this standard, a deviation from the standard **may** be necessary.

Objects cited in this document do not represent the full set of objects available in all modeling, repository, or Computer-Aided Software Engineering (CASE) tools, or more generally, in the IT systems development environment. In some instances, a tool does not call an object type by the same name. If further guidance on naming is needed in these cases, contact Data Management at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov).

The data objects currently covered by this standard may not comprise all data objects in the Systems Development Life Cycle. As the use of other data objects becomes necessary, this standard will be revised to address them.

Further examples of the rules in this standard may be found in the *Standard Data Element* (SDE) publications of the Data Management Branch at the link <http://irm.m.state.sbu/sites/ops/SIO/APD/dm/Standards/Forms/AllItems.aspx>

Because the object definition is as important as the name, this document also provides the user with a set of rules for defining object types. Data object types other than those specifically covered in this document may be standardized using the rules in this standard. This will be especially appropriate for system developers and others wishing to standardize their physical data objects.

This standard provides the basis for defining and naming the following data objects, listed in the order in which they are discussed in the body of the standard:

|  |  |
| --- | --- |
| **DATA OBJECT** | **DEFINITION** |
| Entities | A thing or object of importance about which data must be captured. Each entity on a data model represents a person, place, thing, or concept about which the business stores formation |
| Attributes | An item of data, a fact, or a single piece of  information about an entity that quantifies,  Identifies, or describes an entity. |
| Relationships | A connection or association between entities that represent relevant information of value to the organization; represents a business rule |
| Tables | The physical representation of an entity, and containing rows and columns wherein data may be stored and retrieved |
| Columns | The physical representation of an attribute. It has a specified size and format. The smallest unit of data that has a meaning in describing information; the smallest unit of named data. |
| Views | A customized, usually limited, presentation of columns contained in one or more tables. Is a virtual table whose contents are defined by a query |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **DATA OBJECT** | | **DEFINITION** | | | | |
| Indexes | | A set of ordered pointers to specific rows in a table. | | | | |
| Triggers | | Stored code object that executes for specific events on tables | | | | |
| Constraints | | Restrictions on the contents of the database or on database operations | | | | |
| Stored Procedure | | A named program or routine stored in a database | | | | |
| 1.3 Naming Conventions | |  |  |  |

Several conventions for defining and naming objects are followed in this manual. This section describes those conventions.

### 1.3.1 Naming Format Statements

Each data object type has a naming standard defined in a format statement. The format statements concisely show how an object’s *Business Name* is formed. In addition to the format statement, names must follow several general rules outlined in each section. The format statements are composed of the following:

The greater than and less than symbols “< >” enclose each name component. The square bracket symbols “[]” enclose optional name components.

The term “(space)” represents a space character to be used between components.

### 1.3.2 Naming Format Glossary

The following name components are used in the naming format statements:

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Class Word | Describe the type of data; they indicate the domain of potential values from which the data item’s valid values are drawn.  e.g.: name, number, amount, percent |
| Data Element | A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes. It is also known as an Attribute, Column or Field. |
| Modifier | A word or words that help define and differentiate a name within the database; are used to add important business information to a business name. Usually a noun or noun phrase used to make the term it modifies more precise or accurate. (Qualifier Term).  e.g.: "CUSTOMER\_PHONE\_NUMBER", the word "PHONE" is a modifier ( it is being used to adequately describe the data object) |
| Object Class Term | | Highest level of qualification and the most important word in a business name. e.g. "employee".. |
| Prime Term | | Describe the major topic or subject area to which the data refers. It identifies the application area, major data category, table, or view, depending on the data object being named. e.g. ORGANIZATION, ACCOUNT |
| Property Term | | A component of the name of a data element that expresses a property of an object class or the category to which the data element belongs.  Property represents the distinguishing characteristic of the business entity. e.g. in the data element “EmployeeAddressText”, the component *Address* is the property term. |
| Qualifier Term | | A word or words that help define and differentiate a name within the database; usually a noun or noun phrase used to make the term it qualifies more precise or accurate. Qualifier terms may be attached to object class terms and property terms. e.g... In the data element “EmployeeMailingAddressText”, the component mailing is a qualifier term. (Modifier) |
| Representation Term | | A word that describes the form of the set of valid values for a data element. e.g... “amount, name” (Class Word) |
| Purpose Term | | A word or words that describe the function of a data object. |
| Role Name | | A noun or noun phrase that describes the function of a foreign key |
| Sequence | | A numeric component of a name that differentiates an object from another that would be identically named without it. |

### 1.3.3 Requirement Level Key Words

The standards in this document use key words to reflect varying levels of adherence that are required. When encountered in the standard, they will be capitalized and bolded. The table below defines the requirement level key words and their meaning.

|  |  |  |
| --- | --- | --- |
| **Label** | **Alternate Labels** | **Definition** |
| MUST BE | REQUIRED or SHALL | The definition is an absolute requirement of the specification |
| MUST NOT | SHALL NOT | The definition is an absolute prohibition of the specification |
| SHOULD | RECOMMENDED | There may be valid reasons when a particular item or behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing it. |
| SHOULD NOT | NOT RECOMMENDED | There may be valid reasons to ignore a particular item or behavior, but the full implications must be understood and carefully weighed before choosing a different course. |
| MAY | OPTIONAL | An item is truly optional. |

### 1.3.4 Standard Business Terms and Abbreviations

For the development of new systems and databases, all *prime terms, object class terms, property* *terms, modifiers, qualifier terms* and *representation terms* MUSTbe derived from theDepartment’s repository of standard business terms wherever possible. If a desired term is not present, it MUST be registered in the Enterprise Metadata Repository (EMR) and made available for re-use. Legacy systems that are in use in the Department will go through a procedure of registration of data elements. The legacy systems metadata will be scanned into the EMR and mapped to existing Standard Data Element (SDE) names, as applicable. Candidate standard data element names will be proposed where SDEs have not already been identified. The candidate elements will be created in accordance with DM’s standard naming convention, and be prepared to be approved by the Enterprise Data Administrative Group (E-DAWG) to become SDEs.

Candidate standard business terms may be submitted via e-mail to Data Management at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov) by sending a request citing the business term, a definition, and proposed abbreviation.

### 1.3.5 Business Name vs. Abbreviated Name vs. Synonym Name

The *Business Name* is the name of an object in a business context. Business name contains information pertinent to the organization. By combining *Standard Business* *Terms* according to the formats described in this document *Business Name* can be formed. *Abbreviated Name* is formed by replacing terms from the Business Name with standard abbreviations and using underscore in place of space. *Synonym Name* can be created by using the acronym formed by the first letter or letters of the business terms in a *Business Name* and omitting any underscores. *Synonym Names SHOULD* be incorporated into *Abbreviated Names* to represent *object class terms* or *prime terms* in physical data objects that require them (for example, *Columns, Indexes, Foreign Keys*). When *Synonym* *Names* are used, theyMUSTbe registered with Data Management.

Candidate *Synonym Names* may be submitted via e-mail to Data Management at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov) by sending a request citing the prime term/object class term and the proposed synonym.

Example of Business Name vs. Abbreviated Name vs. Synonym name :

An entity that represented a bank account held by a customer for the purpose of borrowing money may be named a “loan account” a have the following names:

Business Name: LOAN ACCOUNT

Abbreviated Name: LOAN\_ACCT – The standard abbreviation for LOAN is LOAN and for ACCOUNT it is ACCT.

Synonym Name: LNACCT – To further shorten the name LOAN is reduced to LN and combined with ACCT.

In data models and databases, the *Business Names, Abbreviated Names,* and *Synonym Names* are used as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Object** | **Business Name** | **Abbreviated Name** | **Synonym Name** |
| Entity | MUST be used | MUST NOT be used | MUST NOT be used |
| Attribute | MUST be used | MUST NOT be used | MUST NOT be used |
| Relationship | MUST be used | MUST NOT be used | MUST NOT be used |
| Table | SHOULD be used | MAY be used | MUST NOT be used |
| Column |  |  |  |
| Object Class  Term/Prime Term | MUST NOT be used | MAY be used | SHOULD be used |
| Qualifier Term  /Modifier/Property Term | MAY be used | SHOULD be used | MUST NOT be used |
| Representation  Term/Class Word | MUST NOT be used | MUST be used | MUST NOT be used |
| Foreign Key |  |  |  |
| Table Name | MAY be used | MAY be used | SHOULD be used |
| Purpose/Role Name | MAY be used | May be used | MUST NOT be used |
| Index |  |  |  |
| Table Name | MAY be used | MAY be used | **S** SHOULD be used |
| Purpose/Role Name | MAY be used | May be used | MUST NOT be used |
| Trigger |  |  |  |
| Table Name | MAY be used | MAY be used | SHOULD be used |
| Purpose | MAY be used | May be used | MUST NOT be used |
| View | SHOULD be used | MAY be used | MAY be used |

### 1.3.6 General principles and Naming Rules

General Principles:

* + Each database object must be uniquely identified.
  + Data Object names should be meaningful
  + Data Object names should describe what the object represents
  + Names should be independent of the application and independent of hardware and software used
  + Reserved words should not be used. *Reserved words are keywords that the DBMS employs for their exclusive use*
  + Data Object names should not include meaning that can change over the life of the object
* Acronyms MUST be Capitalized

Naming Guidelines for Logical Structure:

* The characters used in names MUST be upper case A-Z, 0-9 and space character
* Punctuation marks and special characters, including the slash (/) and the hyphen or dash (-) MUST NOT be used.
* Underscore MUST NOT be used in names
* The first character in a name MUST be an alphabetic.
* Entity names MUST be singular nouns
* Relationships MUST be verbs or verb phrases
* Possessive nouns MUST not be used in names
* Standard abbreviations ( from glossary of approved standard term) MUST be used where they exist

Naming Guidelines Physical Data Model:

* + Characters used in names MUST be upper case A-Z, 0-9
  + CamelCase MAY BE used in names
  + Standard abbreviations ( from glossary of approved standard term) MUST be used where they exist
  + Underscore MUST be used in place of space in naming
  + Acronyms from glossary of approved standard terms MUST be used in naming when they exist
  + The terms used in names SHOULD be plural nouns and MUST NOT be verbs (except in *Relationships*).
  + Possessive MUST not be used in names

### 1.3.7 Language Specific Exceptions

This standard has been developed to accommodate most popular Relational Database Management Systems (RDBMS) in use at the Department. It is understood, however, that physical database limitations may require shorter names than what would typically be derived using this standard.

### 1.3.8 Invalid Name Components

Synonyms are not to be used in place of a business term already in use to describe an object. Any word selected for use in an object name must be used consistently throughout the set of all objects. For example, if *APPROVAL* is used as a modifier for *DATE*, then *APPROVAL* must always be used whenever that concept must be captured. Synonyms like CONSENT, PERMISSION, and ENDORSEMENT must not then be used in place of *APPROVAL* to describe another term conveying the same concept as the first.

[Appendix A.](#page37) provides a list of invalid and reserved name components, which are not to be used in formulating names.

## 1.4 Standard Data Elements and the Enterprise Conceptual Data Model

Standard Data Elements (SDEs) are data attributes that have been standardized for usage across the Department. As such, the naming conventions for SDEs are the same as those outlined in this document.

The Enterprise Conceptual Data Model (ECDM) provides a specification of the key data entities that support Department of State’s (DOS) business process. The purpose of the Enterprise Conceptual Data Model (ECDM) is to provide a conceptual view the key data entities and their relationships that support DOS’s mission. DOS requires such a model in order to provide an organizing framework for further enterprise data architecture efforts. The ECDM acts as a high-level taxonomy organizing all of DOS’s data assets into groups and rules. This high-level presentation will allow DOS management and stakeholders to effectively understand the current state of the data architecture and to plan for a future state data architecture that will enhance DOS. It is composed of entities, relationships, general definitions and attributes. The ECDM defines the major data domains of information maintained to conduct Department business and consists of business objects (Entities) in the Department. These entities represent the highest-level view, the most essential data categories that define the boundaries and the nature of the Department’s business and distinguish it from any other government enterprise. ECDM is currently a work in progress model detail is being added by analysis and discussion that identify additional data requirements. Entities are added, and as facts are identified for each Entity, they become attributes of that Entity. These attributes can become *Candidate Standard Data Elements* and ultimately be approved as *Standard Data Elements.* For more information on SDEs and the Enterprise conceptual data Model contact Data Management at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov).

# 

# 2. Entities

An *Entity* is a set of real or abstract things (a person, place, thing, resource, concept, or event) that have common *Attributes* or characteristics about which a business retains information. In data modeling, it is a logical object whose physical counterpart is usually a *Table*.

## 2.1 Entity Types

There are five types of entities.

**Fundamental** – An entity that is independent of any other Entity for its existence. Alsoknown as an Identifier-Independent Entity. This may also be known as a Parent entity if it exists in a relationship with an Attributive Entity

**Associative** – An Entity that represents a Relationship between two or more Entities. An*Associative Entity* does not exist independently from the related *Entities*. An *Associative Entity* resolves many-to-many relationships.

**Attributive** – An entity that describes another entity. It is dependent on the existence ofthe other entity. Also known as an Identifier-Dependent Entity. Attributes repeated within an entity are candidates for attributive entities.

**Supertype** – An Entity that represents a general class of business objects that may be broken down into a hierarchy of more specific classes. A *Supertype Entity*’s attributes apply to all of its *Subtype Entities*, and the *Subtypes* inherit its identifier. It is also known as a Generic entity.

**Subtype** – An Entity that identifies or represents an occurrence of another Entity with the same Primary Key but has a narrower definition, a subset of different Attributes, and/or different relationships. It inherits all the attributes of the *Supertype Entity*. It is also known as a *Category Entity.*

**Subtype** – An Entity that identifies or represents an occurrence of another Entity with the same Primary Key but has a narrower definition, a subset of different Attributes, and/or different relationships. It inherits all the attributes of the *Supertype Entity*. It is also known as a *Category Entity.*

## 2.3 Naming Entities

Legacy systems in use in the Department are not required to change existing entity names to adhere to this standard. Those entities will be mapped to standard names where possible.

* In data models the *Business Name* MUST be used for the entity. An exception to this is made if the modeling tool cannot accommodate a long name. Standard abbreviations (from glossary of approved standard term) MUST be used where they exist as part of *Business Name*.

The following general standards apply in creating an *Entity Business Name*:

* The characters used in names MUST be upper case A-Z, 0-9 and space character
* Punctuation marks and special characters, including the slash (/) and the hyphen or dash (-) MUST NOT be used.
* Underscore MUST NOT be used in names
* The first character in a name MUST be an alphabetic.
* The name MUST be composed of singular Nouns or noun phrases SHOULD be singular
* Possessive nouns and proper nouns MUST NOT be used in the name
* The name MUST be fully spelled out.
* The name MUST be 120 characters or less in length
* Verbs SHOULD NOT be used
* The name of a child *Entity* MAY include the name of its parent *Entity*.

A *Synonym Name* MUST be defined for use as the *prime term/object class term* in *Column* *Names*. Using an acronym or shorter abbreviation for the terms in the *Table Business Name* forms *Synonym Names*. They do not have to be formed from standard abbreviations. They should, however, be recorded in the set of full metadata for an Entity. *Synonym Names* MUST be registered with the Data Management Branch and made available for reuse.

Candidate *Synonym Names* may be submitted via e-mail to Data Management at [DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov) by sending a request citing the prime term/object class term and the proposed synonym.

### 2.3.1 Entity Name Formats

All Entities MUST be named according to one of the formats described below:

|  |  |
| --- | --- |
| **Entity Type** | **Format** |
| All entities MAY be named using the format | [<Prime Term> (space)] < modifier(s)>  [<qualifier term> (space)] <object class term> e.g: HUMAN RESOURCES PERSONAL DATA |
| Associative entities SHOULD use the format | <parent entity name> (space) <Modifier >  e.g. HUMAN RESOURCES PERSONAL ACCOUNT DATA |
| Attributive entities MAY use the format | <parent entity name> (space) <modifier(s)>  e.g. HUMAN RESOURCES PERSONAL TRANSACTION DATA |
| Subtype entities MAY use the formats | <parent entity name> (space) <modifier(s)>  <modifier(s)> (space) <parent entity name>  e.g. HUMAN RESOURCES PERSONAL ACCOUNT TYPE DATA |

## 2.2 Describing Entities

The following rules apply when describing an *Entity* of any type (*Fundamental, Associative,* *Attributive, Supertype* or *Subtype*).

* An *Entity* description MUST be a noun phrase
* The description MUST be broad enough that no instances of the Entity are omitted
* The description MUST be clear, concise, and unambiguous
* The description MUST be relevant to its business purpose and independent of technology and implementation
* The description MUST be stable over time. The following words or phrases are examples of time dependency or process orientation, and MUST NOT be used to describe an *Entity*:

|  |  |  |  |
| --- | --- | --- | --- |
| At this (point in) time | |  | Occasionally |
|  | Perhaps |  | But not always |
|  | Unless this happens |  | In certain circumstances |
|  | In this situation |  | However, under these circumstances |
|  | When this happens |  | Frequently |
|  | If this happens |  | Depending on |
|  | However |  | Sometimes |

* The description MUST NOT simply repeat the name of the *Entity* as a description.
* The description MUST be of an *Entity*, not of the data the Department records about the *Entity*, nor the functions, applications, or organizations that use or create the data. Thedescription MUST NOT pertain to:
* When, how, or where the data about the Entity are used
* Who uses the data
* How to edit or process the data
* The format the data stored in or other physical considerations
* What hardware or software systems use the data
* *Abbreviations* and *Acronyms* MUST NOTbe used in descriptions
* Two *Entities* MUST NOT be circular in description. (For example, a description of one *Entity* should not point to descriptions of another)
* The description MUST be:
* Grammatically correct
* Spelled correctly
* Complete and accurate, fully reflecting the meaning of the Entity.
* Written in active voice, where possible.

### 2.3.2 Prime Terms/Object Class Terms

An *Entity’s Business Name* is also known as *a prime term* or *object class term*, which may be composed of more than one word. The designation of a prime term/object class term is critical

to the successful implementation of establishing standard data elements. *Prime terms* reflect the subject area information used by various business areas in the Department. They are also a key component of *Attribute* names. For *Attribute Abbreviated Names*, the prime term/object class term may be expressed as the *Synonym Name* instead of the *Entity Abbreviated Name* to satisfy the 30 characters or less length requirement.

## 2.4 Entity Metadata Properties

The metadata properties listed in the following table are to be used to fully document an *Entity*.

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| Abbreviated Name | The short form of the Business Name. (Follow the abbreviation guidelines found in Section 12 of this document.) |
| Business Name | The unabbreviated form of the entity name. |
| Comment | Any remarks of significance to the understanding of the entity’s history |
| Description | The textual description of the entity. |
| Primary Key | The primary identifier that is used to uniquely identify a record instance, or other data grouping in the entity. It is composed of one or more attributes. |
| Business Rule(s) | The manner in which one or more business processes uses the entity. There may be many business rules that pertain directly to the entity or to its relationship to other entities |
| Non-Key Attributes | Identifies all attributes in the entity that are not part of the primary key. |
| *Synonym* Name | Typically an acronym formed by the first letter or letters of the business terms in a Business Name. The Synonym Name is typically 8 characters or less and is used specifically in the Abbreviated Names of Attributes and Columns |

# 3. Attributes

An *Attribute* is an item of data, a fact or piece of information about an *Entity*. An *Attribute* represents a characteristic or descriptive property of an *Entity*. In data modeling, it is a logical object whose physical counterpart is a *Column.*

## 3.2 Naming Attributes

Legacy systems in use in the Department are *not* required to change existing attribute names to adhere to this standard. Those attributes will be mapped to standard data element names where possible.

The following general standards apply in creating an *Attribute Business Name*:

* The name MUST be UPPERCASE
* The name MUST be composed of the characters A-Z, 0-9
* The name MUST be fully spelled out
* The name MUST be 120 characters or less in length
* Nouns SHOULD be singular except where the plural form is commonly used
* Verbs MUST NOT be used
* Possessive nouns and proper nouns MUST NOT be used in the name
* Underscore MUST NOT be used in names

### 3.2.1 Attribute Name Format

*Attribute Names* MUSTuse the format:

|  |
| --- |
| <prime term> (space) [<modifier(s)> (space)] <class word>  e.g..: Building Identification Number |

*In ISO/IEC 11179 terminology:*

|  |
| --- |
| <object class term> (space) [<qualifier term> (space)] <property term> (space) <representation term> |

## 3.1 Describing Attributes

The following rules apply when describing an *Attribute*:

* The description MUST be a noun phrase, complete and detailed
* The description MUST pertain to a single occurrence of the Attribute in the present tense
* The description MUST be precise and unambiguous. It MUST identify the Attribute and distinguish it from any other Attribute
* The description MUST be relevant to its business purpose and independent of technology and implementation
* The description MUST be stable over time
* The description MUST NOT simply repeat the name of the *Attribute* as the description
* Abbreviations and acronyms MUST NOT be used in descriptions
* The *Attribute*’s description MUST NOT contain the description of the *Attribute’s* prime term/object class term or class word/representation term, since they have been defined separately.
* The description MUST be stable over time. The following words or phrases are examples of time dependency or process orientation, and MUST NOT be used to describe an *Attribute*:

|  |  |  |  |
| --- | --- | --- | --- |
| At this (point in) time | |  | Occasionally |
|  | Perhaps |  | But not always |
|  | Unless this happens |  | In certain circumstances |
|  | In this situation |  | However, under these circumstances |
|  | When this happens |  | Frequently |
|  | If this happens |  | Depending on |
|  | However |  | Sometimes |

* The description MUST BE of an *Attribute*, not of the data the Department records about the *Attribute*, nor the functions, applications or organizations that use or create the data. The description MUST not pertain to:
* When, how, or where the data about the *Attribute* are used
* Who uses the data
* How to edit or process the data
* The format the data stored in or other physical considerations
* What hardware or software systems use the data
* Abbreviations and acronyms MUST NOT be used in descriptions.
* Two *Attributes* MUST NOT be circular in description (for example, two *Attributes* descriptions cannot exist where one description points to a second description; and the second *Attribute* description points back to the first description).
* The description MUST BE:
  + - Grammatically correct.
    - Spelled correctly.
    - Complete and accurate, fully reflecting the meaning of the *Attribute*.
    - Written in active voice.
  + The description MUST NOT include an example. Examples of an *Attribute’s* domain values MAY be written as a separate business rule.

### 3.2.2 Class Words/Representation Terms

A *class word*, or *representation term,* is a reserved word to be used as part of an *Attribute Name* so that the type of data it represents may group the *Attribute*. In most cases, the words reserved as class words MUST NOT be used as modifiers in the *Attribute Name*.

The approved list of standard class words appears below. Each approved CLASS WORD is

shown together with its standard *Abbreviation.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class Word** | **Abbr.** | **Data Type** | **Data Element**  **Categories** | **Definition** |
| AMOUNT | AMT | Numeric | Amount, Average, Balance, Cost, Price | A monetary value. |
| CODE | CD | Alphanumeric | Code, Category, Status, Type, Condition | A combination of one or more numbers, letters, or special characters substituted for a specific meaning. Represents finite, predetermined values. |
| DATE | DT | Numeric | Date, Day, Month, Year | The designation of a specific 24-hour period of time. A date, specified by month, day, and year (for example, July 4, 1976), but in YYYYMMDD format (for example, 19760704). |
| DATETIME | DTTM | Numeric | Date and Time | The designation of a specific  chronological point in time in  conjunction with a specific 24-hour period of time. |
| DESCRIPTION | DESC | Alphanumeric | Description | A character string used to tell the  facts, details or particulars of  something. |
| FILE | FIL | Alphanumeric | Binary Data that can’t be described as Sound, Video, or Picture but has a file  extension type associated | An attribute that holds data in a known file format that does not conform to another, more specific class word (i.e., SOUND, VIDEO, or PICTURE). For example: an XML or PDF file stored in a Binary Large Object (BLOB). |
| IDENTIFIER | ID | Alphanumeric | Identifier, Designator, Index, Key | A combination of one or more  numbers, letters, or special characters that designate a specific entity that have no readily definable meaning. |
| INDICATOR | IND | Alphanumeric | Binary Data that can’t be described as sound, Video, or Picture but has a file extension type Associated | An attribute that holds data in a known file format that does not conform to another, more specific class word (i.e., SOUND, VIDEO, or PICTURE). For example: an XML or PDF file stored in a Binary Large Object (BLOB). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class Word** | **Abbr.** | **Data Type** | **Data Element**  **Categories** | **Definition** |
| NAME | NM | Alphanumeric | Name, Title | A designation of an entity expressed in a word or phrase. |
| NUMBER | NUM | Numeric | Number, Count, Index | A non-monetary numeric value that is not a calculated unit or aggregated unit. |
| PICTURE | PIC | Binary | Picture | A picture, including graphics that can be stored in a binary large object (BLOB) and viewed on the screen. |
| QUANTITY | QTY | Numeric | Quantity, Average, Balance, Deviation, Mean, Median, Mode, Altitude, Depth, Diameter, Dimension, Elevation, Height, Length, Radius, Width, Magnitude, Percent | A non-monetary numeric value that does not have to be a whole number. It is a calculated or aggregated value. |
| SOUND | SND | Binary | Sound | Audio that can be stored in a binary large object (BLOB) and heard on system speakers. |
| TEXT | TXT | Alphanumeric | Text, Comments, Memo, Description, Definition | An unformatted character string (free-form narrative), frequently in the form of words with no length limitation. |
| TIME | TM | Numeric | Time, Quarter | A designation of a specified  chronological point designated as an occurrence (in the past, present, or future) within a period. |
| VIDEO | VID | Binary | Video | Dynamic pictures that can be stored in a binary large object (BLOB) and viewed at a workstation. |

For each class word, an *Attribute’s* description SHOULD begin as follows:

|  |  |
| --- | --- |
| **Class Word** | **Description** |
| AMOUNT | The <modifier> amount of… |
| CODE | The code that represents… |
| DATE | The date on which… |
| DATETIME | The date and time at which… |
| DESCRIPTION | The description of… |
| FILE | The file that… |
| IDENTIFIER | The identifier of… |
| INDICATOR | Indicates whether |
| NAME | The name of… |
| NUMBER | The <modifier> number that… |
| PICTURE | The picture that… |
| QUANTITY | The quantity of… |
| SOUND | The sound that… |
| TEXT | The text that describes… |
| TIME | The time at which… |
| VIDEO | The video that… |

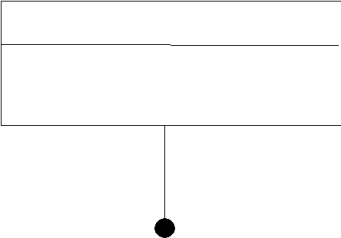
### 

### 3.2.3 Migrated Foreign Key Attributes

Attributes that exist because of a relationship to another Entity, relationship between the same Entity, multiple relationships with the same entity, or from a Categorization/Supertype Entity have additional rules that must also be followed. These rules impact which object class term/prime term is used in naming the Attribute:

* *Attributes* migrated through non-recursive relationshipsMUSTmaintain the name of theobject class term/prime term (the *Entity Name*) from which it migrated.
* *Attributes* migrated through recursive relationshipsMUSTprefix the object classterm/prime term with a *Role Name* for all but the primary *Attributes’* prime term/object class term. The primary attribute MAY also prefix the object class term/prime term with a Role Name.

Migrated foreign key attribute names are illustrated in  [Figure 2. Migrated Foreign Key Attribut](#page23)e  [Names](#page23) below. Note that the migrated attributes in the CUSTOMER ACCOUNT associative entity maintain the object class term/prime term of their originating entities: CUSTOMER and ACCOUNT rather than take on the prime term of their actual entity CUSTOMER ACCOUNT.



CUSTOMER

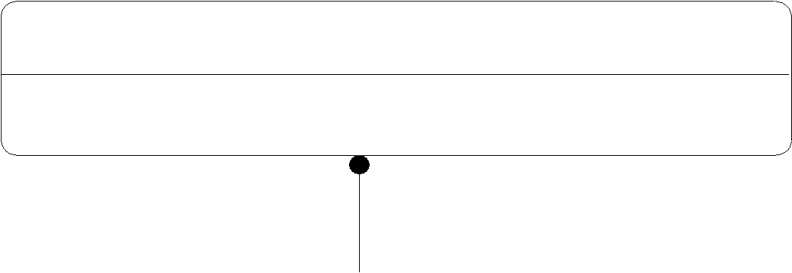
CUTOMER IDENTIFIER

CUSTOMER LAST NAME

CUSTOMER FIRST NAME

is authorized to use

CUSTOMER ACCOUNT



CUTOMER IDENTIFIER (FK)

ACCOUNT IDENTIFIER (FK)

CUSTOMER ACCOUNT PERSONAL IDENTIFICATION NUMBER

CUSTOMER ACCOUNT PRIMARY HOLDER INDICATOR

may be accessed by

ACCOUNT

ACCOUNT IDENTIFIER

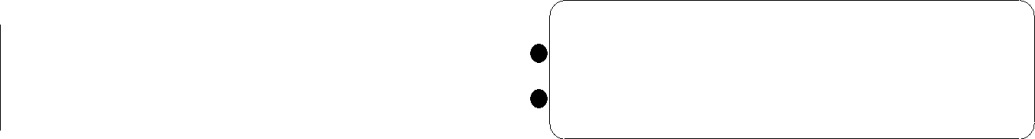
ACCOUNT TYPE CODE (FK)

ACCOUNT BALANCE AMOUNT

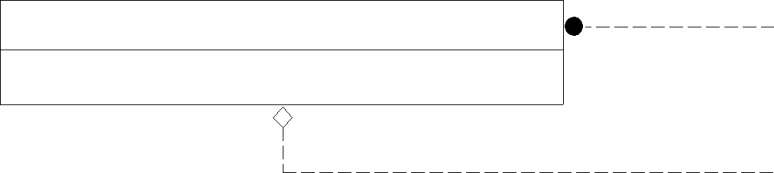
*Figure 2. Migrated Foreign Key Attribute Names*

Role Names by which migrated foreign keys are illustrated in Figure 3 (Role Names on Migrated Foreign Keys), where RELATED is the role given to the migrated foreign key PERSON IDENTIFIER. In the second example PARENT is the role given to attribute migrated through the recursive relationship of ORGANIZATION with itself

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PERSON | |  |  | PERSON RELATIONSHIP |  |
| is related to |  | PERSON IDENTIFIER (FK) |  |
| PERSON IDENTIFIER |  |  |  |
|  |  | RELATED PERSON IDENTIFIER (FK) |  |
|  | is related to |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |  |



ORGANIZATIO N



ORGANIZATIO N IDENTIFIER

PARENT ORGANIZATION IDENTIFIER (FK)

is parent to

*Figure 3. Role Names on Migrated Foreign Keys*

## 3.3 Attribute Metadata Properties

The metadata properties listed in the following *Table* are necessary to fully document an *Attribute*. The name of the property is the name that appears in the current repository schema.

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| Allowed Values | Entries permitted for an instance of an attribute |
| Business Name | The attribute name in its unabbreviated form |
| Business Rule(s) | The manner in which one or more business processes uses the attribute. There may be many business rules that pertain directly to the attribute or to its relationship to other attributes |
| Case Sensitivity | Indicates whether or not the data is to be upper case, lower case or mixed |
| Comment | Any remarks of significance to the understanding of the attribute’s history. |
| Data Length | The maximum allowable length for the attribute. |
| Data Type | The allowed data format for the attribute (for example, alphabetic, binary, and so on). |
| Default Value | The domain value that is automatically assigned when no other value is specifically identified. |
| Derivation Rule | The algorithm used to determine how the attribute is derived. |
| Description | The textual definition of the attribute. |
| Domain Definition | The general description of the applicable domain for the attribute. |
| Domain Detail | Additional detail pertaining to the Allowed Values. |
| Mandatory Requirement | Identifies whether or not the attribute is required for an instance of the entity to have meaning. |
| Originating Entity | The entity in which the attribute is initially defined. |
| Originating Organization | The organization(s) that is the source of the attribute’s definition and maintenance. |
| Other Security | Handling restriction, such as Sensitive, under Freedom of Information Act (FOIA). |
| Precision | The number of places after the decimal point. |
| Range Maximum | The upper bound of the range of acceptable data values. |
| Range Minimum | The lower bound of the range of acceptable data values. |
| Reference Documentation | Information pertaining to the source material for the attribute’s definition or a statement regarding the current source organization for the domain definition. If the source is from Data Management, a complete a set of domain values will be identified. If the source is other than Data Management, a sample of the domain will be defined in the Allowed Values property. |
| Security Classification | Level of national security protection required for the attribute. |

# 4. Relationships

A *Relationship* is an association between two or more *Entities* (a non-recursive relationship) or between occurrences of the same *Entity* (a recursive relationship) that represents a business rule. *Relationships* are used in Entity Relationship Diagrams (ERDs) to convey information as to how *Entities* correspond to one another. In data modeling, it is a logical object whose physicalcounterpart is a *Foreign Key Constraint*.

## 4.1 Naming Relationships

The following rules apply when determining a *Relationship Business Name*:

* + The *Entity Names* MUST be UPPERCASE
* The *verb phrase* MUST be lower case.
  + The *Relationship Name* MUST be an active voice *verb phrase*
  + The *Relationship* names MUST be clear and precise. Ideally, the names SHOULD fully describe the *Relationship* so that no further description is needed. Avoid weak verbs and imprecise clauses. For example, avoid the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Has |  | Does |
|  | Can |  | Could |
|  | Is related to |  | Might |
|  | Relates to |  | Has a relationship to |

Is a kind of

* Characters used for a *verb phrase* MUST be lowercase a-z and the space character. Punctuation marks or special characters, including the slash (/) and the hyphen (-) and numbers MUST NOT be used. Avoid using special characters like "/?!@#$ %^&\*() +-=‘," or numbers.
* Verbs occurring in a *Relationship* name MUST be singular unless the sense requires the plural.
* *Abbreviations* and *Acronyms* MUST NOTbe used in *Relationship* names.

### 4.1.1 Relationship Name Format

*Relationships* MUSTuse the following format:

|  |
| --- |
| *<(parent) entity name> (space) <active voice verb phrase> (space)* |

|  |
| --- |
| *<(child) entity name>* |

## 4.2 Describing Relationships

All of the information the analyst needs to precisely describe a *Relationship* is often provided when the following items are determined:

***The nature of the relationship between the entities,*** For example, the *Entities**EMPLOYEE* and *WORKSITE* may have the following *Relationship*:

* + *EMPLOYEE works at WORKSITE*
  + *WORKSITE is work location of EMPLOYEE*

The two *Entities*, together with the nature of the *Relationship* between two *Entities*, are known collectively as the *Relationship Name*. The *Relationship* is established by the business rules of the enterprise. Each business rule MAY be stated in either an active or passive voice, resulting in two *Relationship Names* for the resulting *Relationship*. For purposes of clarity in the data model, only the *Relationship Names* with the active voice MUST be described. The name with the passive voice is optional, and is only used where it adds significant value without detracting from the clarity of the model.

***The optionality between the entities****, Relationship* ***optionality*** indicates whether a

*Relationship* is optional or required. Frequently, a *Relationship* can be optional whenviewed from one *Entity* and required when viewed from the other. The following illustrates the concept of optionality.

* + - A *CUSTOMER may place ORDER* (optional)
    - An *ORDER* must be placed by a *CUSTOMER* (required)

***The cardinality between the entities***,*Relationship*cardinality indicates how many of one*Entity* is related to how many of another *Entity*. *Relationships* between two *Entities* maybe one-to-one (1:1), one-to-many (1: M), or many-to-many (M: M). the modeler should be wary of including 1:1 *Relationships* in a finished data model. A 1:1 *Relationship* normally indicates that two *Entities* can be combined into one *Entity*. An M: M *Relationship* MUSTbe represented by two 1: M *Relationships* and an *Associative Entity*.When recording the *Relationship Name*, optionality and cardinality, enough information is usually conveyed so that *Relationship* descriptions are not required.

## 4.3 Foreign Key Constraint on Relationship

In some Relational Database Management Systems (RDBMS), *Foreign Key Constraints* may be used to physically enforce a business rule defined by a *Relationship*. The *Foreign Key* constraint name in the physical data model does not affect the verb phrase on the *Relationship* in the logical data model.

### 4.3.1 Foreign Key Constraints Name Format:

*Foreign Key Constraint Names* MUSTus the following format:

|  |
| --- |
| FK\_*<sequence number>*\_*<table name>\_<referential table name>*[\_*<purpose* *or role name>*] |

The *Table Names* used in *Foreign Key Constraint Names* SHOULD be the *Table Abbreviated* *Names* butMAYalso be the *Table Business Names* or *Table Synonym Names*.

The *purpose, role name or sequence* is typically only used when more than one constraint exists between two tables. The *purpose* should describe the functional role of the constraint and MUST be composed of *Standard Business Terms*. The *role* may represent the role name of the column that is being constrained. The *sequence* simply differentiates between the two constraints but does not provide any additional information and is therefore the least desirable.

# 

# 5. Tables

A *Table* is the physical manifestation of an *Entity*, containing rows and columns wherein data may be stored and retrieved and represents a person, place, thing, resource, concept, or event about which a business retains information. These rules also apply for the representation of a *Table* in a physical data model. *Tables* follow the naming rules of their corresponding *Entities*,but with spaces replaced by underscores. See Section 2 of this document for a complete list of rules for describing and naming Tables, as well as their appropriate metadata properties.

## 5.1 Naming Tables

The same naming rules for naming an entity apply to naming tables:

Exceptions:

* + Acronyms and abbreviation from glossary of standard term MUST BE used if exist
  + CamelCaseMAY be used
  + Table names (nouns) must be pluralized
  + The names of tables that implement a star schema MUST have prefixes that specify the table’s role in the star
    - schema model:

“DIM” for dimension and “FACT” for fact.

* + Underscore MUST be used in place of space

### 5.1.1 Name Format:

[<Prime Term> (space)] < modifier(s)>

[<qualifier term> (space)] <Object class term> e.g.: *HR\_ PERSONAL \_DATA*

## 5.2 Describing Tables

Tables MUST be described in the same fashion as Entities.

## 5.3 Table Metadata Properties

The metadata properties used to fully document a *Table* are the same as those for an *Entity*.

# 6. Columns

A *Column* is a vertical segment in a *Table* and the physical manifestation of an *Attribute*. These rules also apply for the representation of a *Column* in a physical data model. See Section 2 of this document for a complete list of rules for describing and naming Columns, as well as their appropriate metadata properties.

## 6.1 Naming Columns

*Columns* MUSTbe named in the same way as their corresponding *Attributes*. Whereas theattribute name uses all business names for its components, the column name should be constructed as follows:

* CamelCaseMAY be used
* Underscore MUST be used in place of space
* Columns MUST be named with a class word, or it's abbreviation
* Acronyms and abbreviation from glossary of standard term MUST BE used if exist

### 6.1.1Name Format:

[<modifier(s)> (\_)] <noun>

[<qualifier term> (\_)] <object class term> e.g..: long\_term\_plan\_dsc

## 6.2 Describing Columns

*Columns* MUSTbe described in the same fashion as *Attributes*.

## 6.3 Column Metadata Properties

The metadata properties used to fully document a *Column* are the same as those for an *Attribute*.

# 7. Views

A *View* is a specific physical data object that provides access to all or a portion of one or more *Tables*. In the case of multiple *Tables* in a *View*, the *Tables* must be capable of being joined. Aprinciple reason for creating a *View* is to give a user somewhat limited access to the columns in the *View’s* tables. See Section 2 of this document for a complete list of rules for describing and naming *Views*, as well as their appropriate metadata properties.

## 7.2 Naming Views

The same general rules for naming an entity also apply to naming Views.

*Exception:*

* + View name MUST have a prefix “VW\_”

### 7.2.1 View Name Format

*View Names* SHOULDuse the format:

[<Prime Term> (space)] < modifier(s)>

[<qualifier term> (space)] <object class term> e.g.: VW\_*HR\_ PERSONAL DATA*

## 7.1 Describing Views

*Views* MUSTbe described in the same fashion as *Entities*.

## 7.3 View Metadata Properties

The metadata properties listed in the following table are to be used to fully document a view.

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| View Name | The identifier of the view, including the abbreviated Functional Name. |
|  |  |
| Functional Name | The name of the business function supported by the view. |
|  |  |
| Description | The textual description of the *View.* |
|  |  |
| Tables | The tables included in the *View.* |
|  |  |
| Columns | The columns included in the *View.* |

# 

# 8. Indexes

An *Index* is a set of ordered pointers to data contained in a *Table*, and it can be created with one or more columns contained in the *Table*. Three *Indexes* will be defined in this section: *Primary* *Key Index, Foreign Key Index, and Alternate Key Index*.

## 8.1 Index Types

**Primary Key Index** – An index placed on the column or columns that make up the Primary Key ofa Table. Primary Key Indexes MUST be defined as “UNIQUE” if RDBMS supports this feature

**Alternate Key Index** – An index placed on a column or columns that could be used to uniquelyidentify a row in the table, but are not the Primary Key. Alternate Key Indexes MUST be defined as “UNIQUE” if your RDBMS supports this feature

**Foreign Key Index** – An index placed on the column or columns that represent a foreign keyconstraint to another table

**Non-key Index** – An index placed on a column that does not represent a key of any of the typesdescribed above

## 8.2 Naming Indexes

* Index MUST be named according to their function and Table Name
* The Index MUST not exceed 30 characters in length

### 8.2.1 Index Name Formats

Indexes MUST be named according to one of the following formats:

|  |  |
| --- | --- |
| Index Type | Format |
| Primary Key Index: | PK\_<table name> |
| Alternate Key Index: | AK\_<table name>\_<purpose, role name or sequence> |
| Foreign Key Index | XFK\_<table name>\_<referential table name>\_<purpose, role name or sequence> |
| *Non\_Key\_Index* | <Table name>\_<purpose, role name or sequence> |

## 8.3 Index Metadata Properties

The metadata properties listed in the following *Table* are necessary to fully document an *Index*.

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| Index Name | The name of the index |
| Index Type | Whether the index defines a Primary Key, Foreign Key, Alternate Key, or Non-Key |
| Index Expanded Name | In the same format as the Index Name, but with all business terms spelled out |
| Table Name | The name of the table in which the index is defined. |
| Column(s) | The name(s) of the column(s) contained in the index |

# 9. Triggers

Some RDBMS support *Triggers* that are stored code objects that execute for specific events on *Tables*. These events are defined as before or after an insert, update or delete of a row orstatement on a specific table.

## 9.1 Naming Triggers

*Trigger Names* are based on when they fire, whether they fire for every row in a table or once forthe event statement, and the event that must occur for them to fire as described below:

* Trigger names MUST be prefixed with the letters “Trgr\_”
* The second portion of the trigger name is composed of three letters and followed by an underscore character.
  + The first letter MUST be either "b" or "a" indicating whether the trigger fires before or after the action, respectively.
  + The second letter MUST be one of the following letters: "i", "u", or "d" indicating whether the trigger fires upon an insert, update, or delete, respectively.
  + The third letter MUST be either an “r” or an “s” indicating whether the trigger fires for each row or statement, respectively.
* On the very seldom occasion that more than one trigger of the same name combination is required (for example, when more than one trigger is needed to update different sets of columns on a table), a number can be appended after the table name.
* The last portion of the trigger name MUST be the table name on which it fires an activity on

\_

### 9.1.1 Formatting Triggers

*Triggers* MUSTuse the following format:

*Trgr\_<B or A> <I or U or D> <R or S>\_Table Name*

The above formatted name can optionally be followed by *<\_1>*if more than one trigger with the exact same name is created.

B or A – Before or After

R or S – Row or Statement

I, U, D – Insert, Update, Delete

## 9.2 Trigger Metadata

*Metadata Properties:*

|  |  |
| --- | --- |
|  |  |
| Description | The textual description of the triggers and Purpose |
|  |  |
| Comment | Any remarks of significance to the understanding of the triggers history. |
|  |  |

# 10. Constraints

Database constraints are restrictions on the contents of the database or on database operations. Database constraints provide a way to guarantee that:

* rows in a table have valid primary or unique key values
* rows in a dependent table have valid foreign key values that reference rows in a parent table
* individual column values are valid

## 10.1 Type of Constraints

1) Primary Key (PK) **-** Serves as the unique identifier for rows in the table

2) A unique constraint (UNQ) - is similar to a primary key constraint but doesn't have to

be defined with Not Null.

3) Foreign key constraint (FK) **-** The relationship between rows in two tables is expressed by a foreign key in the dependent table. A foreign key is one or more columns that contain a value identical to a primary key (or unique key) value in some row in the parent table (i.e., the referenced table).

4)Check Constraints (CHK) - Used to enforce the validity of column values

## 10.2 Naming Constraints

* MUST including Keyword stating the database object
* MUST include suffix with abbreviation for the type of Constraints
* Separate each word with an underscore
* If more than one constraint is required exist within a type of constraint, MUST add a numeric suffix of 1 through 9.

### 10.2.1 Constraints Name Formats

Constraints MUST be named according to one of the following formats:

<Noun (keyword)>, <\_>, <abbreviation for type of constraints

|  |  |  |
| --- | --- | --- |
| **Type of Constraints** | **Naming Rule** | **Example** |
| Primary Key | Primary key type constraints MUST be named after the table name plus a suffix of "\_pk". | * proj\_et\_pk * prcl\_cnty\_pk * srfc\_wtr\_pmp\_pk |
| A unique constraint | The unique constraint MUST be named after the table plus a suffix of "\_unq". If more than one unique constraint is required, add a numeric suffix of 1 through 9. | * proj\_et\_unq * prcl\_cnty\_unq1 * prcl\_cnty\_unq2 |
| Foreign key constraint | A foreign key constraint MUST be named after the table plus a suffix of "\_fk". If more than on foreign key constraint is required, add a numeric suffix of 1 through 9. | * proj\_et\_fk * bdgt\_acct\_fk1 * bdgt\_acct\_fk2 |

|  |  |  |
| --- | --- | --- |
| Check Constraints | If the check condition references only one column, name the constraint after the column name plus a suffix of "\_chk".  If more than one check condition is required per table or column, add a numeric suffix of 1 through 9 | * proj\_id\_chk * prcl\_cnty\_chk1 * prcl\_cnty\_chk2 |

## 10.3 Constraint Metadata Properties

The metadata properties listed in the following table are necessary to fully document Constraints

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| Constraint Name | The Name of the Constraints |
| Constraint Type | Type of Constraint such as Primary key, Foreign Key, Unique constrain or Check Constrain |
| Table Name | The name of the table in which the constrain is defined |
| Column(s) | The name(s) of the column(s) contained in the constrain |

# 11. Stored Procedure

An action oriented named program or routine stored in a database. Stored procedures are precompiled database queries that improve the security, efficiency and usability of database client/server applications. Developers specify a stored procedure in terms of input and output variables

## 11.2 Naming Stored Procedure

### Stored procedures performs a function, they are action oriented. Name MUST describe the function;

### MUST use a verb as prefix to describe the work

### MUST use keyword of the object

### SHOULD use underscore to separate words

### 11.2.1 Stored Procedure Name Formats

Stored Procedure MUST be named according to one of the following formats:

Prefix (verb) <\_>, <Prime term>, <qualifier > e.g..: Get\_Customer\_Details Insert\_Customer\_Info

## 11.3 Stored Procedure Metadata Properties

The metadata properties listed in the following table are necessary to fully document Constraints

|  |  |
| --- | --- |
| **Metadata Property** | **Documentation Requirement** |
| Stored Procedure Name | The Name of the Stored Procedure |
| Description | Describe the function of the stored procedure. Schedule of when the procedure is applied. |
| Table Name | The name of the table in which the stored procedure is defined |
| Column(s) | The name(s) of the column(s) contained in the procedure |

# 

# 12. Acronyms and Abbreviations

*Acronyms* and *Abbreviation*s are necessary due to some physical tool constraints. They eliminate objectname redundancy and inconsistency and improve the quality of model descriptions and application documentation by using clear and commonly used words. Acronym and *Abbreviation* standards enable analysts to select *Acronyms* and abbreviations that are as clear and commonly used as possible. The standards also require a consistent use of the *Acronyms* and *Abbreviations*, regardless of the length of the name. The *Abbreviated Name* MUST have each word in the name abbreviated in accordance with this section. A Business Name MUST use Standard abbreviations (from glossary of approved standard term) MUST be used where they exist.

For the purposes of this document, the definition of *acronym* is “a word formed from the first (or first few) letters of a series of words.” An *Abbreviation* is defined as “a shortened *Form* of a word or phrase by contraction, or by omission of letters.” A *candidate* is defined as an acronym or *Abbreviation* which an organization wants to use that does not yet exist in the repository.”

## 12.1 Creating Abbreviations

Create a candidate *Abbreviation* by using the following rules:

1. Check the following sources for a common *Abbreviation* for the term in question:
   * A commonly accepted American dictionary
   * Any commonly accepted *Abbreviation* (de facto standard)

An *Abbreviation* for the term in question **may** be found through these sources, or follow *Abbreviation* rules 2 through 20. If a readily acceptable *Abbreviation* is found, use it, identify itssource, and go to rule 2; then skip rules 3 through 20. If the *Abbreviation* is not readily acceptable due to possible conflicts or duplications, or it does not adequately represent the word it replaces, continue with the *Abbreviation* rules. Apply common sense.

1. Ensure that each *Abbreviation* is unique, not only with regard to other *Abbreviations*, but also with respect to *Acronyms*.
2. Ensure that each term has only one *Abbreviation*.
3. *Abbreviations* MAYconsist of alphabetic characters only.
4. Only the singular *Form* of the business term should be used.
5. An *Abbreviation* should be recognizable; that is, looking at the *Abbreviation*, one should be able to visualize the word.
6. Generally, do not abbreviate words that are five or fewer characters, except for class words. Exceptions MAY be made for size considerations.
7. Preserve the first letter of the term, whether it is a vowel or consonant.
8. Always treat “y” as a consonant.
9. Delete unnecessary vowels; however, not all vowels need to be eliminated to have a valid *Abbreviation*. Keep those that are necessary to make the *Abbreviation* understandable*.*
10. Generally, delete one consonant of a double consonant. Exceptions MAY be made for clarity.
11. If the removal of a vowel causes a double consonant then keep the vowel.
12. If the term has a leading double vowel (e.g., “au” or “ou”), keep both vowels. For example,

*AUTHORIZATION* would be abbreviated *AUTHZN*.

1. If the abbreviation already exists for another word, for example, *FCLTY* for *FACILITY*, then it is necessary to either keep one of the vowels for the new *Abbreviation*, or use a commonly accepted *Abbreviation* that is sufficiently different. For example, using FAC for *FACULTY* might be usedin lieu of *FCLTY*, which would otherwise be one result of following the rules.

A root word and its derivatives SHOULD have the same ‘root’ *Abbreviation*. For example, the

*Abbreviation* for *EXEMPT* is *EXMPT*, and for *EXEMPTION* is *EXMPTN*. The ‘root *Abbreviation*’ in both cases is *EXMPT*.

Always eliminate the vowels in a suffix. Use *G* as the *Abbreviation* for the *ING* suffix, and *MT* for the *MENT* suffix. Using this rule, the *Abbreviation* for *PRINTING* is *PRINTG*, and an acceptable *Abbreviation* for *EMPLOYMENT* is *EMPMT*.

1. If a root word is five or fewer characters and is not abbreviated, its derivatives may have the root portion spelled out or abbreviated, but, if abbreviated, all derivatives must have the same *Abbreviation* of the root portion of the word. For example, *PRINT* is not abbreviated since it isfive characters. *PRINTING* can be abbreviated *PRINTG*, and all other derivatives would also contain the root *PRINT*. On the other hand, *CLEAR* is not abbreviated, but *CLEARANCE* MAY be abbreviated *CLRNC*. In this case *CLR* MUST be used as the root for all derivatives of

*CLEAR*.

1. *Abbreviations* must not spell an expletive.

## 12.2 Creating Acronyms

The rule for creating a candidate acronym is a simple. An acronym is formed from the first or first few, letters of a series of words. Examples include “FICA” in place of “F ederal I nsurance Contributions Act,” and “radar” in place of “RAdio Detecting and Ranging.” *Acronyms* are not to be used in Business Name s; nor are they to be used in descriptions unless they are first spelled out. However, some *Acronyms*, such as radar and sonar, become so common that they are accepted as words, and are notcapitalized in normal use. *Acronyms* must not spell an expletive.

## 12.3 Candidate Term Submittals

All candidate *Acronyms, Abbreviations*, and *Synonym Names* submitted to Data Management will go through a review and approval cycle. Once DM accepts candidates, they will be reviewed in a timely manner. Candidate *Acronyms* and *Abbreviations* that an organization desires to use, but do not exist in the *Standard Acronyms* and *Abbreviations List* for Common Business Terms, will be measured against the rules in this section.

Candidate terms may be submitted via e-mail to Data Management at  [[DataMgmtSupport@state.gov](mailto:DataMgmtSupport@state.gov),](mailto:DA@state.gov) citing the candidate term, the definition, and the proposed synonym.

# Appendix A: Invalid Entity and Attribute Name Components

Neither *Entity* nor *Attribute Names* MUST contain conjunctions, prepositions, certain adverbs, or phrases listed below unless it is necessary to meaningfully name the object.

Invalid *Entity* and *Attribute* name components:

|  |  |  |  |
| --- | --- | --- | --- |
|  | A |  | Non |
|  | An |  | Nor |
|  | After |  | Occasionally |
|  | Always |  | Off |
|  | And |  | Often |
|  | Because |  | On |
|  | But |  | Or |
|  | Do |  | Sometimes |
|  | Else |  | The |
|  | For |  | Then |
|  | Frequently |  | Through |
|  | How |  | Thru |
|  | However |  | To |
|  | If |  | Too |
|  | In |  | When |
|  |  |  | While |

The word ***KEY*** is reserved for use in naming key fields.

# 

# Appendix C: Document Revision

# History Detail

1. **Changes from the Fourth (4th) Edition** 
   1. The standard was brought into conformance to ISO/IEC 11179-5 – Information Technology – Specification and Standardization of Data Elements – Naming and Identification Principles for Data Elements from earlier conformance to Federal Information Processing Standards (FIPS) 156 Information Resources Dictionary System (IRDS).
   2. Acceptable Business Name, Abbreviated Name and Synonym Name usage has been changed as follows:
   3. Business Names must now be used in data models for all logical objects including Entities, Attributes and Relationships.
   4. For Tables, the Business Name is now preferred.
   5. Synonym Names are now preferred for the Prime Term/Object Class Term/Table Name component of Column, Foreign Key Index, and Trigger names.
   6. Business Terms are preferred in View names.
   7. Added DESCRIPTION, FILE, and INDICTOR to class word/representation terms.
   8. Foreign Key Constraint name prefixes were changed from R\_ to FK\_.
   9. Foreign Key Index prefixes were changed from FK\_ to XFK\_.
   10. Trigger prefixes were modified to include more accurate indicators as to when the trigger fires.
   11. The sequence number name component was dropped from View names to encourage more accurate functional names.
   12. The standard for Form objects was dropped.
2. **Changes from the Fifth (5th) Edition** 
   1. Changed IRM/OPS/SIO/API/DM organization name from Data Administration to Data Management (DM), including references to DM e-mail address.
   2. Updated term repository to specify Metadata Repository (MDR).
   3. Updated intended audience section.
   4. Added website reference for FIPS 184:  [http://www.itl.nist.gov/fipspubs/idef1x.do](http://www.itl.nist.gov/fipspubs/idef1x.doc)c.
   5. Deleted references to page numbers used to cross-reference applicable sections within the document. Table of contents will be sufficient for cross referencing and will lessen effort to synchronize re-pagination in future document updates.
3. Added items to Acronyms and Abbreviations List of document (DAWG, DM, EDM, and IDE1X).
4. Re-worded sentences for better clarity and completeness, without changing the context of the information being conveyed.
5. Added additional rule to Section 1.4.6, General Naming Rules, as follows: “For logical objects, the underscore character MAY be used in place of spaces.”
6. Added Data Modeling Syntax Legend in Section 2.0, Entity Types, to show graphical notations used in data models related to entities, super types/subtypes, and relationships.
7. Revised Section 4.3.1, Foreign Key Constraints Name Format to show:

FK\_*<sequence number>*\_ *<table name>\_<referential table* *name>*[\_*<purpose or role name>*]

1. Added Appendix B to this document to provide additional guidelines (template format) in naming database components related to ORACLE and MS SQL Server DBMS.
2. Added Appendix C to provide detailed account of the revisions made to this document. Transferred Section 1.3, “Changes from the Fourth Edition,” to Appendix C of this document (6th Edition).
3. **Changes from the Seventh (7th) Edition**
4. Changed all the named references from IRS/OPS/SIO/API/DM Branch to IRS/OPS/SIO/APD/DM
5. Changed DM Email address
6. Deleted all the reference to Old DM site
7. Changed the requirement for general Naming rules
8. Added General naming Principal section (section 1.3.6)
9. Add two new data objects Constraints and Stored Procedure at section (1.2)
10. Updated section 1.4 *Standard Data Elements and the Enterprise Conceptual Data Model* by replaced the Scope Model with ECDM
11. Deleted *Acronyms and Abbreviations* section of the (list under section 1.4). This section is not necessary since Acronyms and abbreviations are defined when mention at first.
12. Updated Section *2.2 Describing Entity Section* Re-worded sentences for better clarity and completeness, without changing the context of the information being conveyed. Deleted redundant rules
13. Updated section 2.1 *Entity Types* removed entity type picture illustration. It is tools specific illustration. Didn’t add much value to the document
14. Added Naming Guidelines for Logical and physical Structure
15. Added additional rule to Section 2.2 *Naming Entities,* deleted redundant rules
16. Deleted website reference for FIPS 184:  [http://www.itl.nist.gov/fipspubs/idef1x.do](http://www.itl.nist.gov/fipspubs/idef1x.doc)c (Out-of -date document)
17. Modified sections for naming, description and metadata properties for the following with new rules for the following object: Entity, Tables, Column, Triggers, views, relationship and index.
18. Added section 10. *Constraints* withType of Constraints, Constraints Name Formats, Constraints metadata properties.
19. Added section 11. *Stored Procedure* withType of Stored Procedure, Stored Procedure Name Formats, and Stored Procedure metadata properties.