|  |
| --- |
| Florida Department of Transportation |
| Data Modeling |
| Standards |
|  |
| **Data Administration** |
| **2/1/2013** |

|  |
| --- |
| Document the data model validation rules used by the Florida Department of Transportation (FDOT) for enterprise application development. |

Table of Contents

[Purpose 3](#_Toc347405357)

[References 3](#_Toc347405358)

[Definitions 3](#_Toc347405359)

[Background 4](#_Toc347405360)

[Procedure 4](#_Toc347405361)

[Model 4](#_Toc347405362)

[Database 5](#_Toc347405363)

[Entity / Table 5](#_Toc347405364)

[Entity / Table Constraint 6](#_Toc347405365)

[Attribute / Column 6](#_Toc347405366)

[Attribute / Column Constraint 7](#_Toc347405367)

[Key / Index 7](#_Toc347405368)

[Relationship 9](#_Toc347405369)

[Stored Procedure 10](#_Toc347405370)

[Tablespace 10](#_Toc347405371)

[View 10](#_Toc347405372)

[Miscellaneous 10](#_Toc347405373)

[Change History 11](#_Toc347405374)

# Purpose

Define and document the data model validation rules used by the Florida Department of Transportation (FDOT) for enterprise application development.

# References

Naming Conventions – Attributes, Columns and Elements

Naming Conventions – Physical Object Naming

# Definitions

**Abbreviation –** A shortened form of a word or phrase created by leaving out or substituting letters.

**Attribute** – A characteristic or property associated with a business entity (people, places, events, and so on). An attribute is the logical equivalent to a column.

**Column –** The physical implementation of an attribute in a database.

**Design Layering –** an approach to modeling in which separate models are created for each phase in the model development process

**Element** – Also known as “data element,” this is the smallest unit of information that can be perceived or understood by a computer application user. Element is an abstract item that forms the basis for attributes and columns. An attribute is a relationship between an element and an entity. A column is a relationship between an element and a table.

**Erwin** – The CA-Erwin data modeling tool used by FDOT to define and document data models and entity relationship diagrams.

**Gen** – The CA-Gen case tool used to define, produce and deliver application systems.

**Glossary** – A list of words or phrases with definitions and abbreviations.

**Key Unification** – the practice of combining like foreign key columns resulting from multiple relationships from a common origin to a single table.

**Repository** – the CA Repository, a software application designed to store, organize and make available electronically FDOT’s enterprise metadata.

**Repository Glossary** – A list of business terms, words, and phrases, with their abbreviations, stored in the Repository.

# Background

This procedure was developed to document the validation rules used for data models developed for the Florida Department of Transportation.

# Procedure

This procedure documents the validation rules that are applied to various model object types.

## Model

1. The model properties must contain:
   1. The model name which includes name components for:
      1. The application for which this is a model.
      2. Database platform, if determined.
      3. Type of model, e.g. conceptual, logical, physical, or combined logical and physical.
   2. Author or company preparing the model.
   3. Definition or description of the model, including the purpose for creating this model.
2. The model shall use the Information Engineering notation (crows' feet) option for the logical entity relationship diagram or model view.
3. The model shall use the Information Engineering or Dimensional Modeling notation option for the physical entity relationship diagram or model view.
4. Object names shall be unique within the scope of the object. Duplicated names are not allowed for entities, tables, and views in a model.
5. The model shall comply with the FDOT naming standards. Use of a naming standards file is optional.
6. All data structure object names shall comply with FDOT naming standards.
7. All object and object properties shall conform to platform limits for the chosen database platform.

## Database

1. Each database within the data model shall contain at least one table.

## Entity / Table

1. Attribute order in the entity shall match the column order in the associated table(s). If design layering is used then the attribute order and the column order shall be the same for all tables generated from the entity.
2. Key unification shall not occur in a physical schema and shall be resolved in the physical design.
3. Data sensitivity properties shall be included in the model for tables.
   1. Basis for protection, degree of protection, and data sensitivity date shall be completed.
   2. Data sensitivity explanation or reference shall be completed when the value for basis for protection or the value for degree of protection is not “PUB” for public.
4. All entities shall have a definition.
5. Initial, maximum, and monthly “grow by” table row numbers shall be included for all tables in a model.
6. Entity name length shall not exceed 32 characters.
7. Table name length shall not exceed 18 characters.
8. All tablespaces shall have an allocation.
9. All tables shall be associated to a tablespace.
10. All entities shall have at least one attribute.
11. All tables shall have at least one column.
12. All entities and tables should contain primary key attributes/columns and at least one non-primary key attribute/column.
13. An entity or table shall have at least one inbound or outbound relationship. Exceptions to this rule shall be approved by FDOT.
14. Volumetrics: initial rows count shall not be greater than maximum rows count.

## Entity / Table Constraint

Constraints shall use SQL92 format. FDOT does not use Field Procedures in the DB2 environments.

## Attribute / Column

1. The child data type shall not be different from the related parent data type on the relationship.
2. Varchar and varchar2 column names shall not exceed 17 characters in length. All other column names shall not exceed 18 characters in length. (This is a z/OS COBOL copybook limitation.)
3. Attribute and column data types shall be valid for the class word or class abbreviation.
4. Attribute and column data types shall be valid for the designated model platform.
5. Attribute and column data types shall be valid for selected domain.
6. All attributes shall be assigned to a valid domain as defined in the Class Word/Domain/Data Type crosswalk.
7. Columns shall not be typed as identity columns.

FDOT does not support identity columns in our DB2 environment. Use a DB2 Sequence object.

1. The class word or abbreviation name shall conform to FDOT naming standards.
2. The class abbreviation in column name shall match to the class word or abbreviation in attribute name.
3. The logical data type shall be compatible with the physical data type.
4. All attributes shall have a definition.
5. Attribute and columns name tokens shall be found in the glossary. Exception: If the name length will exceed the name length limits, a non-standard abbreviation may be acceptable. Non-standard abbreviations shall be approved by FDOT.
6. Columns shall not contain names for the null value constraint name.

Oracle platform only. DDL execution errors will occur in the FDOT Oracle environment when null value constraint names are provided.

1. Data sensitivity properties shall be included in the model for columns.
   1. Basis for protection, degree of protection and data sensitivity date shall be completed.
   2. Data sensitivity explanation or reference shall be completed when the value for basis for protection or the value for degree of protection is not “PUB” for public.
2. The child attribute or column name representing a foreign key should be the same as the parent primary key attribute or column name that corresponds on the relationship.
3. The attribute and column data types shall be allowable data types for selected domain.
4. The attribute and column data types should be the preferred data type for selected domain.
5. Attributes and columns should have the same data type definition for all occurrences of the attribute or column in the model. An explanation shall be required for the differences.
6. Attributes and columns should be associated to the same domain for all occurrences of the attributes and columns in the model. An explanation shall be required for the differences.
7. Attributes and columns shall not be associated to a deprecated domain.

## Attribute / Column Constraint

1. Error Level Rules
   1. Constraints shall use SQL92 format. FDOT does not use Field Procedures in the DB2 environments.
   2. The table number in constraint name shall be the same as the table number in table name.

## Key / Index

1. All key members shall be attributes in the model.
2. All index columns shall be columns in the model.
3. The key member order shall match the order of the attributes in the entity.
4. The index column order shall match the order of the columns in the table.
5. All keys shall contain at least one attribute.
6. All indexes shall contain at least one column.
7. All entities shall contain a primary key.
8. All tables shall contain a primary key index.
9. All indexes shall be associated to a tablespace.

## Relationship

1. Identifying relationships shall not allow nulls in the child foreign key column(s).
2. All relationships shall have a valid parent entity and/or table name.
3. All relationships shall have valid parent and child entities and table names.
4. Many-to-many relationships shall not be included in either a logical model or a physical model.
5. Each relationship shall have a parent to child phrase that represents an assertion about the relationship.
6. Each relationship shall have a child to parent phrase that represents an assertion about the relationship.
7. The parent or child phrase length shall not exceed 32 characters in length.
8. Relationships should not use “CASCADE” for any integrity rules.

## Stored Procedure

Stored procedures shall be associated to a table or view. This is valid as a warning only if the stored procedure is associated to another database object type or to the database itself.

## Tablespace

1. All physical designs shall contain at least one tablespace object.
2. Each Oracle tablespace shall be associated to an Oracle physical storage object.
3. Each DB2 tablespace shall be associated to a single DB2 table.
4. Each tablespace shall contain allocation values.
5. All Oracle tablespaces must specify an extent size value.

## View

1. Missing data sensitivity property

Basis for Protection, Degree of Protection and Sensitivity Designation Date are required. Sensitivity Explanation or Reference is required for all non-public data.

1. A view shall have SQL to define it whether it is generated by the modeling tool or is supplied as user defined SQL.
2. Data sensitivity properties shall be included in the model for views and materialized views.
   1. Basis for protection, degree of protection and data sensitivity date shall be completed.
   2. Data sensitivity explanation or reference shall be completed when the value for basis for protection or the value for degree of protection is not “PUB” for public.

## Miscellaneous

1. All model objects shall be shown on a diagram or contained in a list of model objects.

# Change History

|  |  |
| --- | --- |
| **Effective Date:** January 30, 2012 |  |
| **Create Date**: February 1, 2012 | **Created By:** Sarah Close |

|  |  |  |
| --- | --- | --- |
| **Last Update Date** | **Last Updated By** | **Reason for Change** |
| December 27, 2012 | Sarah Close | Updates to validation rules and minor typographical corrections. |
| January 31, 2013 | Sarah Close | Revise into requirements statement format. |
| January 31, 2013 | Morgan Bunch | Final revision for RFQ. |
| February 1, 2013 | Morgan Bunch | Spell/grammar check. |