Policy: 8075-1- Database Design and Modeling Standard
Section: Database Standards
Office/Department: Office of IT Applications
Reports To: Division of Information Tech
Contact: 404-631-1000

PURPOSE

The purpose of this document is to establish department wide standards for data definition and modeling practices; promote their effective use for IT development projects and facilitate common, interoperable representations and descriptions of data.

SCOPE

- This policy applies to all GDOT IT personnel (including consultants, contractors, vendors and other third parties) actively engaged in database design, development, or modification; and to any outside entities engaged in developing IT for GDOT.
- Exceptions to the scope of this standard must be approved by the Administrator of IT Application Support & Development.
- The scope of this document does not extend to COTS products whose implementation within GDOT does not require significant customization or modification to the product.

RESPONSIBILITY

1. The Administrator of the Office of IT Application Support & Development is responsible for compliance with the standard, updates to the standard, and enforcing the standard.
2. IT Team Leaders are responsible for compliance with the standard and for reporting concerns to the IT Application Support & Development Administrator.

DEFINITIONS

COTS "Commercial off-the-shelf" Product - Any third-party software product that intended to be used "as-is".

STANDARDS

The main objectives of this standard are to:

- Promote and support the use of data modeling for system development.
- Increase data sharing opportunities across GDOT, reduce data redundancy, and improve application interoperability.
- Increase opportunities for consolidation of like data and business processes.
- Establish data modeling standards that best meet current and future GDOT requirements, considering traditional, legacy, and web-based application development efforts.
- Leverage and reuse existing data sub-models where appropriate.
- Build a foundation to help with the integration of information silos.

1. In accordance with the business and implementation requirements, GDOT IT may complete, and subsequently update the Conceptual Data Model throughout the development cycle of the IT project. A conceptual data model is typically used to identify and document business (domain) concepts with project stakeholders. It is one layer more abstract than a logical data model that describes a particular computer-based system. A conceptual data model contains subject areas, classes, and relationships, and generally models a project-specific domain. It is also the most viable level at which to integrate different data models because object representational differences are excluded.
2. GDOT IT or contracted vendor shall complete, and subsequently update a Logical Data Model throughout the development cycle of the IT project. This Data Model illustrating the logical structure of data objects shall be represented via an Entity Relationship Diagram (ERD). A logical data model contains subject areas, normalized classes, atomic attributes, relationships, and candidate/primary keys, and usually serves as a model for an enterprise-specific design of a project-specific domain.
3. GDOT IT shall or contracted vendor shall complete, and subsequently update a Physical Data Model throughout the development cycle of the IT project. This Data Model illustrating the physical structure of data objects shall be represented via an Entity Relationship Diagram (ERD). A physical data model contains tables, attributes, relationships, and candidate/primary keys. This model usually serves as an enterprise-specific implementation of a project-specific domain.
4. The GDOT Enterprise Data Warehouse (EDW) will be used as the primary destination for data consumption by GDOT applications. Data to be exchanged between multiple enterprise database source systems for purposes of completing/updating a data set for completing business processes is a valid exception.
5. Any and all documentation generated during the design and modeling process shall be stored in the GDOT IT ClearCase code repository and shall be subject to the terms and conditions governing GDOT's implementation and established policies for the Clear Case product.
6. The current preferred, enterprise ETL tool used by GDOT IT is SAP's Data Integrator. Additional tools or methods can be employed with the approval of the Administrator of IT Application Support & Development (e.g. SQL Server Integration Services).
7. For all COTS products, it is expected that the vendor will provide the ERD as needed to facilitate GDOT IT implementation, support, development, or future modification. Constraints and business rules will be included.
8. Enterprise modeling tools used for the purposes of database development by GDOT IT shall adhere to the following requirements:
    - Data modeling tools used in conjunction with relational database technologies shall be Extensible Markup Language (XML) compliant.
    - Data modeling tools used in conjunction with object-oriented database technologies shall be Unified Modeling Language (UML™) and Extensible Markup Language (XML) compliant.
9. The Database Design and Modeling process and procedures shall be subject to any limitations, specifications or restrictions that are adopted by GDOT as part of additional standards documentation.
10. All data models and designs developed by a vendor must be provided to and subsequently approved by IT management in the Office of IT Application Support & Development prior to code development.

DATA NAMING STANDARDS

- Microsoft Access/SQL Reserved Key Words will not be used as per: http://support.microsoft.com/kb/286335
- Oracle and PL/SQL Reserved Key Words will not be used as per: http://download.oracle.com/docs/cd/B28359_01/appdev.111/b31231/appb.htm
- Programmer's Guide to the Oracle Pre-compilers: https://docs.oracle.com/en/database/oracle/oracle-database/21/zzpre/Oracle-reserved-words-keywords-namespaces.html

- Hyphens, dashes, or spaces cannot be used in names
- Names cannot start with a number
- Do not use "\ / : * ? " < > | # { } % & <TAB>" ~ + ." within field names, coded domains, or valid values. Special characters may create constraints when working with other applications/systems.
- If in doubt about an alias, acronym, or definition, the Bureau of Transportation Statistics Dictionary will be used as the authoritative source.
- GDOT IT shall maintain a list of standard abbreviations that can be used in column/field naming.

ORACLE NAMING STANDARDS

Object and Dataset Names

This section discusses the GDOT standard naming conventions for Oracle objects and datasets. These conventions were designed to meet the following objectives:

- Guarantee uniqueness of Oracle object names within an Oracle database;
- Provide a uniform naming structure for Oracle objects of similar types;
- Simplify physical database design decisions regarding naming strategies;

Usage

The following naming standards apply to any Oracle object (table, view, PL/SQL routine, etc.) used to hold or maintain user data. It will be used in conjunction with Oracle as part of standard application development and system operation procedures. All oracle object names must begin with an alphabetic character. All Oracle objects will be named according to the standard formats listed in the table below.

Oracle Object Type	**Object Rule**	**Example**
Table	• Maximum 128-character name • Must be descriptive as it applies to the application • Must start with TBL_ identifier • Cannot contain reserved words or special characters • If the name is over 128-characters; abbreviate the object name.	TBL_EMPLOYEES
Index	• Maximum 30-character name • Must start with IDX_ identifier • Cannot contain reserved words or special characters • Must be unique within database; • Must be the table name, suffixed by the type of index: • A primary key index should be suffixed with a _pk## identifier (## = sequence number); • A foreign key index should be suffixed with a _fk## identifier (## = sequence number);	IDX_EMPL_PK01 IDX_ EMPL_PK02 IDX_ EMPL_UK01 IDX_ EMPL_AK01 IDX_ EMPL_FK01

	• A unique key index should be suffixed with a _uk## identifier (## = sequence number); • An alternate key index should be suffixed with a _ak## identifier (## = sequence number). • If the name is over 30-characters; abbreviate the object name.	SEE ABOVE EXAMPLES
View	• Maximum 128-character name • Must be descriptive as it applies to the application • Must start with VW_ identifier • Cannot contain reserved words or special characters • If providing view to another application that requires specific data; must have the SOURCE application acronym and TARGET application acronym. • If the name is over 30-characters; abbreviate the object name.	VW_EMPLOYEES VW_HCM_ELMS_EMPLOYEE_DATA
Synonym	• Maximum 128-character name • If the name is over 128-characters; abbreviate the object name.	EMPLOYEE_DATA
Column	• Maximum 30-character name • Unique within the corresponding table or view • Should be derived from the business name identified during the business/data analysis process • Each word within the column name must be separated by an underscore (_) • Column name may not contain reserved words or special characters • If the name is over 30-characters; abbreviate the object name.	EMPLOYEE_ID
Column Constraints	• Maximum 30-character name • Column check constraints must be prefixed by **CK_**, followed by an application identifier and underscore, followed by the column name. • If the name is over 30-characters; abbreviate the object name.	CK_ELMS_EMPLOYEE_ID
Primary Key	• Maximum 30-character name • Primary key name must be the table name prefixed by PK_ • Unique within the corresponding table • If the name is over 30-characters; abbreviate the object name.	PK_EMPLOYEES
Foreign Key	• Maximum 30-character name • Must have the table name and column name prefixed by FK##_ (## = sequence number). • Unique within the corresponding table	FK01_EMPLOYEES_COUNTY FK02_EMPLOYEES_CITY

	• If the name is over 30-characters; abbreviate the object name.	
Procedure, Function, Package	• Maximum 128-character name • Prefixed by PROC_ (procedure) or FUNC_ (function) or PKG_ (package) and Application Identifier, followed by a description; • If the name is over 128-characters; abbreviate the object name.	PROC_ELMS_GET_EMPLOYEES PKG_ELMS_GET_EMPLOYEES FUNC_ELMS_CONCAT_DATA
Trigger	• Maximum 128-character name • Must start with TRGR_ Identifier, followed by a description; • Cannot contain reserved words or special characters. • If the name is over ~~30~~ 128-characters; abbreviate the object name.	TRGR_ELMS_UPDT_COURSES
Materialized View	• Maximum 128-character name • Must be descriptive as it applies to the application • Must start with MVW_ identifier • Cannot contain reserved words or special characters. • If the name is over 128-characters; abbreviate the object name.	MVW_EMPLOYEE_DATA

GEODATABASE NAMING STANDARDS

- All Feature Datasets and Feature Classes will be uniquely named using the following structure:
  - Feature or Product Name (e.g. GIS.TRAFFIC_COUNTER)
  - Year if produced as separate products over multiple years (e.g. GIS.TRAFFIC_COUNTER_2009)
  - Format of feature if produced in multiple types (e.g. GIS.TRAFFIC_COUNTER_2009_POINT vs. GIS.TRAFFIC_COUNTER_2009_LINE or GIS.DOQQ_1999_BW vs. GIS.DOQQ_1999_CIR)
- All Feature Datasets and Feature Class names will be capitalized
- All Feature Datasets and Feature Classes will be expressed as singular, unless the proper name of the dataset is plural *(e.g.*, GIS.NATIONAL_WETLANDS_INVENTORY_2011). Aliases can be used to express plurality.
- All field (column) names will be expressed in singular.
- All field (column) names will be capitalized
- All field (column) aliases (comments) will be in title case
- Aliases, General Properties of the Feature Class, and published services can be expressed in plural forms to end users.
- Each word within the field (column) name must be separated by an underscore
- Keep all field names as distinct, concise and meaningful as possible in the fewest number of characters.
- Abbreviation is to be avoided, provided that all table and field (column) names are under 30 characters. If the naming cannot meet 30 characters, then it will be shortened by:
  - First, using the standard abbreviations below or the Bureau of Transportation Statistics Dictionary abbreviations
  - Second, by using the root of the word (e.g. TRANS instead of TRANSITORY). If could be considered and abbreviation, it should be added to the standard abbreviations.

    - Third, by a common language abbreviation (e.g. ELEM for Elementary instead of ELMT) or by shortening words by removing vowels (e.g. SCND instead of Secondary).
- When using abbreviations, first abbreviate the last word in the name and progressively toward the beginning if necessary.
- Suffixes of "_point," "_line," "_polygon" will be used only when multiple representations apply to the same feature class, *e.g.*, project_point, project_line, project_polygon, project_annotation
- For scripted feature classes, dynamically generated datasets, or feature classes that are refreshed or change on a regular basis do not include a year in the feature class name.
- Feature Datasets are to be used for the following (listed in order of importance to our workflow):
    - Spatially or thematically integrating related feature classes — Organize related feature classes into a common dataset for building a topology, a network dataset, a terrain dataset, or a geometric network.
    - Organizing data access based on database privileges — Administratively organize data access privileges using feature datasets, implementation is advantageous because all feature classes contained within a feature dataset have the same access privileges.
    - Organizing thematically related feature classes — Organize a collection of feature classes for a common theme into a single feature dataset.
- Multiple representations of a singular feature class should be governed by topology rules within a feature dataset.
- Preserve source name as much as possible.
- Keep naming of Feature Datasets, Feature Classes, and fields (columns) consistent within the geodatabase.
- If you have fields of the same name or similar name within the geodatabase (and they represent different things), rename them to distinguish between the fields across the geodatabase.
- In bringing over county and city FIPS codes, include the 13 for the state of Georgia as a separate field called STATE_CODE. Call it COUNTY_CODE instead of COUNTY_FIPS to provide GDOT with the flexibility to modify the code as it changes over time without having to reconfigure dependent IT systems.
- Clarify the measurement of the data such as BEGINNING_MILEPOINT or BEGINNING_MILEPOST, instead of BEGINNING_MEASURE.

Aliases and Abbreviations

- Keep aliases consistent as much as possible across geodatabases.
- Keep aliases consistent as much as possible across Feature Datasets, Feature Class, and fields (columns) within a geodatabase.
- Title case aliases
- Fully spell out all aliases. No abbreviations are allowed in aliases.
- Aliases need to be distinct, concise, and meaningful. Aliases are a short description. Aliases are meant to be more descriptive, intuitive, or understandable than a field name. Aliases do not have any defined size limitations; however, aliases longer than 50 characters should be reviewed to see if they could be made more concise. Aliases should not be a lengthy description or paragraph definition of what the field means. More complete definitions of a field should be incorporated into the Entity/Attribute sections of the metadata.
- When using abbreviations, the full word will be spelled out in the alias.

References:

- Microsoft Access/SQL Reserved Key Words per: http://support.microsoft.com/kb/286335

- Programmer’s Guide to the Oracle Pre-compilers: https://docs.oracle.com/en/database/oracle/oracle-database/21/zzpre/Oracle-reserved-words-keywords-namespaces.html
- Bureau of Transportation Statistics Dictionary abbreviations

History:

annual review: 06/22/23;
new policy effective: 12/18/09