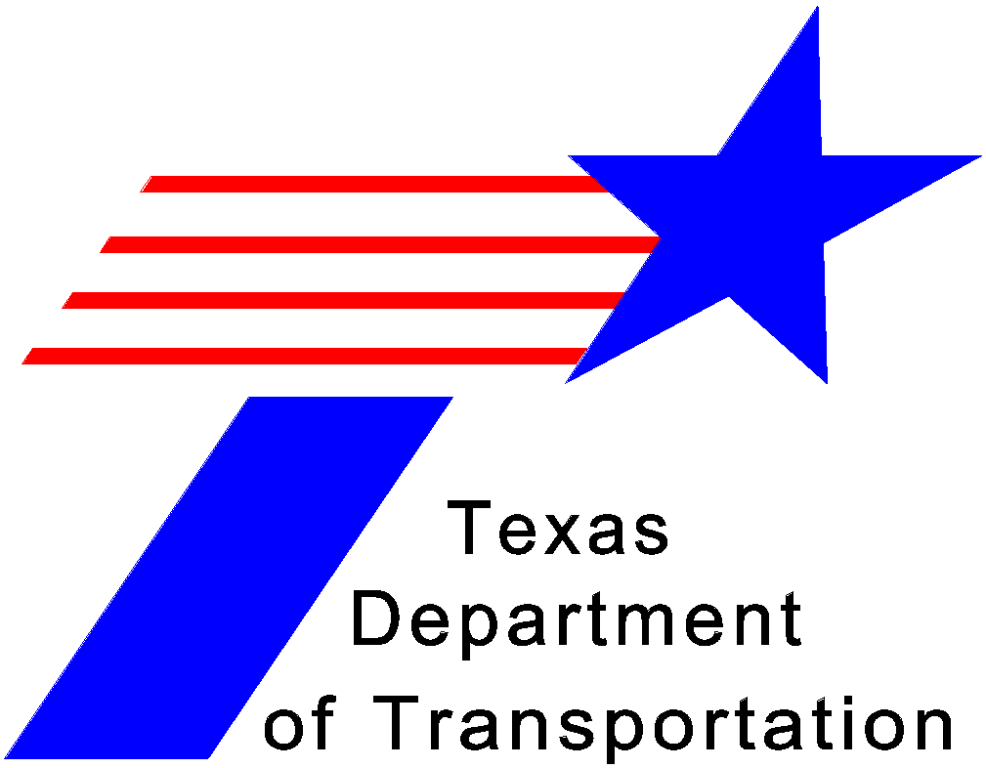
Data Architecture



Version 4.2
July 2010

Version History

Version	Date	Editor	Note(s)
4.1	2/2010	EH	Changes to match TSD font and format standards
4.2	7/2010	JP	Glossary links from words within body of text Chapter/Section bookmarks created (pdf version)

NOTE: Incidental corrections (font, spelling, grammar, and the like) that do not affect the meaning or intent of this document may be made but not noted.

Document Navigation

This document originates as a Microsoft Word file (.doc) and is intended to be viewed and navigated as an Adobe PDF file (.pdf).

Bookmarks have been provided to each Chapter and their subsequent Section(s). Click on the Bookmark icon to open the navigation tree to the Chapter level. Click the “+” sign beside the icon to expand one level. Click the “-“sign beside the icon to collapse back one level. If the Bookmark navigation panel is not open along the left edge of the window, right click in the document and choose "Show Navigation Panel Buttons".

Data Architecture Table of Contents

Data Architecture Table of Contents (continued)

Chapter 1

TxDOT's Data Architecture

Contents:

This page intentionally left blank.

Section 1

Overview

Data

More information was created worldwide in the last three years than in all of previous recorded history. However, according to a 1998 Business Objects Survey, 62% of all managers feel they do not get the right amount of information to make a decision. It is estimated that 80% of decision-making time is spent gathering the data while only 20% is spent actually making decisions (source: *Data Management Association International Facts*).

TxDOT Data

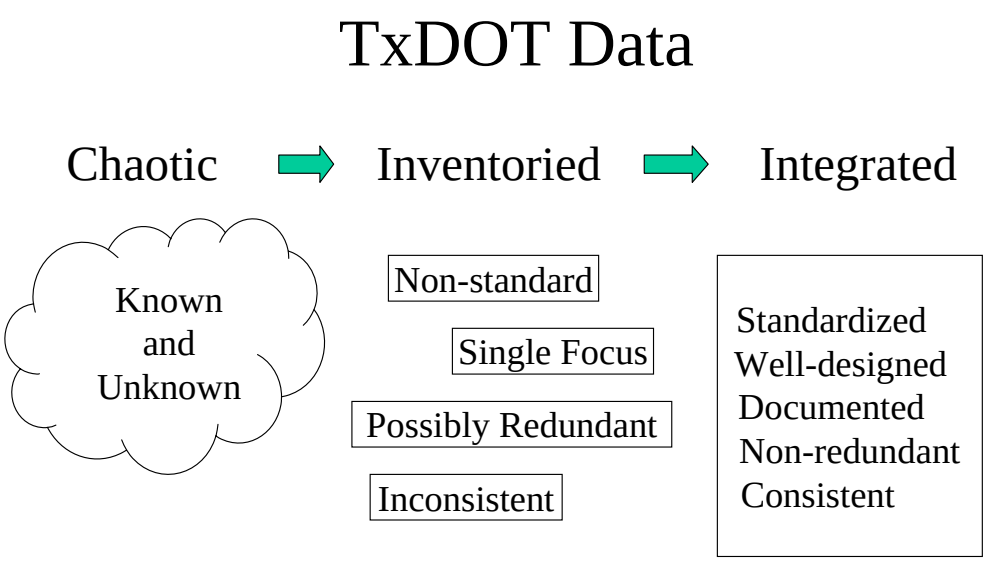
Over the years, data at the Texas Department of Transportation (TxDOT) has been developed in a somewhat "chaotic" manner. Many projects with similar data were developed independently for a single purpose, creating redundant data or incompatible data formats. Some projects followed existing data standards, while some did not. Consequently, TxDOT data now exists at various levels of integration.

Some data is documented and known to some extent. However, some data is not documented and may be unknown to most users. For instance, there are some older systems for which documents may not have been written or may have been lost. Some new systems using vended packages or contracted development may not have shared their documentation. In addition, some documentation is inconsistent.

Strategy

TxDOT data is used daily in making decisions and in allocating resources. In order to ensure that these decisions and allocations are made with the highest quality data, sound data management practices are essential.

By using sound data management practices, TxDOT data is gradually being moved from a somewhat "chaotic" state toward a more "integrated" state. This is a long-term process.



The first step to integrating TxDOT data is to create an inventory of the data, beginning with what is known. Redundancy and variability are identified and cross-referenced.

As existing data structures are modified, or when new data is gathered, the standards and guidelines in the TxDOT Data Architecture should be followed. Following the Data Architecture will ensure that, over time, TxDOT data will become standardized, non-redundant and consistent.

What is Architecture?

Architecture is the manner in which the components of a computer or computer system are organized and integrated.

What is Data Architecture?

Data Architecture is the manner in which data components are organized and integrated.

A data resource designed and constructed according to a data architecture provides a consistent foundation across organizational boundaries to provide easily identifiable, readily available, high quality data to support business activities.

Scope of the Data Architecture

The focus of the Data Architecture is to provide TxDOT with an enterprise-wide blueprint for data design. For software project development teams, working with existing databases or developing new databases, the Data Architecture is essential. Topics addressed in this document include:

- Data standards
- Data naming
- Data normalization
- GIS data standards.

For the software project manager modifying or creating a new database, Chapter 2 (Data Standards) and Appendix B (Data Normalization) will provide guidance through the data design process. Chapter 3 (Data Naming Class Words) and Chapter 4 (Data Naming Abbreviations) will provide technical guidance in data naming. For projects including components of a Geographical Information System (GIS) in their database, Chapter 5 (GIS Data Standards) will provide additional guidance. Chapter 6 will provide guidance when considering Commercial Off-the-Shelf (COTS) software with a database.

Chapters 2, 3, and 4, along with Appendix B should be given to and followed by contractors who are modifying or developing a database for TxDOT. If contractors are developing a GIS component, Chapter 5 should be followed as well.

Appendix C (Data Model Interpretation) may be useful to subject matter experts when reading, reviewing, or verifying data models.

Appendix D (TSID Interpretation) may also be used by subject matter experts for assistance in reading the TxDOT System Interface Diagrams (TSIDs).

Benefits of Data Architecture

Applying components of the TxDOT Data Architecture when creating or modifying data will:

- Provide a data environment that promotes better communication and more informed decision-making for both internal and external stakeholders
- Promote a common understanding of the data
- Promote the sharing of data across organizational boundaries
- Reduce data redundancy
- Reduce loss of knowledge due to turnover and retirement.

Applying the data architecture will produce benefits to TxDOT. Non-compliance, in favor of quick implementation, will almost certainly result in eventual increased costs.

This page intentionally left blank.

Section 2

Inventories

Overview

The TxDOT Data Architecture includes the current inventories and the integration methods used to move the data from a chaotic state to an integrated state.

Current TxDOT Inventory

The current inventory of information about data at TxDOT consists of the following:

- Data Subject Areas
- Glossaries
- Data Model Inventory
- Data and Application Inventory.

Data Subject Areas

Data subject areas are data categories that are used to group the data. It is a set of similar or closely related subjects or entity types. Data subject areas give a general indication of where data responsibilities reside. These subject areas are used by a new user to aid in searching for a data source or data.

At TxDOT, subject areas are typically aligned with divisions and offices. Currently identified subject areas are:

Data Subject Areas and Related Information

Subjects	Division or Office	Subject Area
Business Plans	Administration	Direction and Leadership
Internal Audits, External Audits, Internal Reviews	Audit Office	Audits
Airports, Airport Facilities, Aviation Projects, Aviation Funds	Aviation Division	Aviation
Bridges, Bridge Projects, Bridge Inspections, Bridge Conditions, Bridge Design, Bridge Maintenance, Bridge Construction, Historic Bridges, Bridge Funds	Bridge Division	Bridges

(continued...)

Current TxDOT Inventory *(continued)*

Data Subject Areas and Related Information

Subjects	Division Office	Subject Area
Construction Projects, Construction and Maintenance Contracts, Contractors, Contract Letting, Contract Bids, Contract Claims, DBE, SBE, HUB, Materials Testing, Research, Pavements	Construction Division	Transportation Systems Construction
Designs, Plans, Specifications, Consultants	Design Division	Transportation Systems Design
Historic Sites, Natural Resources Management, Environmental Compliance, Hazardous Materials	Environmental Affairs Division	Environment
Accounts, Budget, Funds, State Infrastructure Bank, Expenditures, Revenue, Payroll, Travel Expenses, Contractor Payments, Vouchers	Finance Division	Fiscal Resources
Open Records Requests, Opinions, Contracts, Contract Management Standards, Policies and Procedures	Office of General Counsel	General Counsel
Security, Equipment, Materials, Supplies, Needs, Procurements, Disposals, Plan Distribution, Warehouses, Recycling, Inventory	General Services Division	General Services
Federal Legislation, State Legislation, Internal Communications, Media Relations, Community Relations, International Activities, Border Transportation, Border Inspection Facilities	Government and Public Affairs Division	Legislative and Public Affairs International Relations
Employees, Benefits, Hiring, Management, Employee Classification, Employee Relations, Training	Human Resources Division	Human Resources
Information Technology Architectures, Hardware, Software, Data, Computer Applications, Telecommunications Networks, Help Desk, Document Management, Geographic Information Systems, Engineering Design, Databases, Security, Global Positioning Systems	Technology Services Division	Information Resources

(continued...)

Current TxDOT Inventory *(continued)*

Data Subject Areas and Related Information

Subjects	Division or Office	Subject Area
Maintenance Projects, Facilities, Vegetation, Ferries, Emergency Management, Highway Conditions, Rest Areas	Maintenance Division	Transportation Systems Maintenance
Motor Carriers, Over-Size/Over-Weight Permits, Licenses, Consumer Complaints, Compliance	Motor Carrier Division	Motor Carrier
Motor Vehicle Dealers, Licenses, Consumer Complaints	Motor Vehicle Division	Motor Vehicles
Employee Safety, Tort Claims	Occupational Safety Division	Occupational Safety
Equal Employment Opportunity (EEO) Categories, EEO Audits, EEO Complaints, Contractor EEO Compliance, Disadvantaged Business Enterprise (DBE), Historically Underutilized Business (HUB) and the Small Business Enterprise (SBE) programs	Office of Civil Rights	Civil Rights Business Opportunities
Public Transportation Systems, Transit Statistics	Public Transportation Division	Public Transportation
Technology Research Abstracts	Research and Technology Implementation Office	Research and Technology Implementation
Real Property, Highway Beautification, Permits, Utilities	Right of Way Division	Right of Way
Signs, Signals, RR Crossings, ITS, Pavement Markings, Roadway Illumination, Traffic Safety, Speed Zones, Hazardous Material Routing, Traffic Safety Grants	Traffic Operations Division	Traffic Operations
Transportation Needs, Transportation Planning, Rail Systems, Highway Performance Monitoring, Traffic Analysis	Transportation Planning and Programming Division	Transportation Systems Planning & Programming
Travel Information Centers, Highway Conditions, Wildflowers, Travel Literature, Travel Maps, Texas Highways Magazine, Litter Prevention	Travel Division	Travel
Turnpikes, Turnpike Projects	Turnpike Authority Division	Turnpikes
Motor Vehicle Titles, Motor Vehicle Registrations, Automobile Thefts, Theft Prevention Grants	Vehicle Titles and Registration Division	Vehicle Titles & Registration

Glossaries

A glossary is a collection of specialized terms with their meanings. Glossary definitions provide an understanding of data and the context in which it is used. At TxDOT, a business term can have different meanings depending on the division or function to which it applies. There are currently multiple glossaries throughout TxDOT, including one for this document.

A database has been developed to inventory the existing glossaries, glossary terms, and their meanings. The process to inventory TxDOT glossaries has begun. Contact the Data and Quality Management Services Branch (DAQ) of the Technology Services Division (TSD) for information regarding the glossaries that have been inventoried.

Data Model Inventory

A data model is a graphic representation of the data structure of a database, along with an associated data dictionary. Data models may represent data in varying degrees of scope and complexity. When they are available, data models of current development and production data are inventoried by the Data and Quality Management Services Branch (DAQ) of the Technology Services Division (TSD).

DAIS

DAIS is an application that contains information about data and databases as they are being developed. It also contains information about data and databases when they are finalized and placed in a production mode. Much of TxDOT's current development and production data is not documented, not documented thoroughly or consistently, or has inaccessible documentation.

Much of the existing documentation is on paper, in programs, or on servers or personal computers that are not accessible by most of TxDOT. When possible, information from existing data dictionaries and other sources is extracted from the source and integrated into DAIS. Additional information about the data is also maintained in DAIS as it becomes known.

Information about applications is also included in DAIS. Application descriptions, application status, Office of Primary Responsibility (OPR), TxDOT System Interface Diagrams (TSIDs), data usage, and other information about the applications are maintained in DAIS.

(continued…)

DAIS *(continued)*

Information currently in DAIS includes:

- TxDOT System Interface Diagrams (TSIDs)
- Test and Production ADABAS data dictionary information
- Computer application descriptions
- Application and data relationship information
- Standard data naming abbreviations
- TACS Table information

This page intentionally left blank.

Section 3

Integrated Methods of Data Design

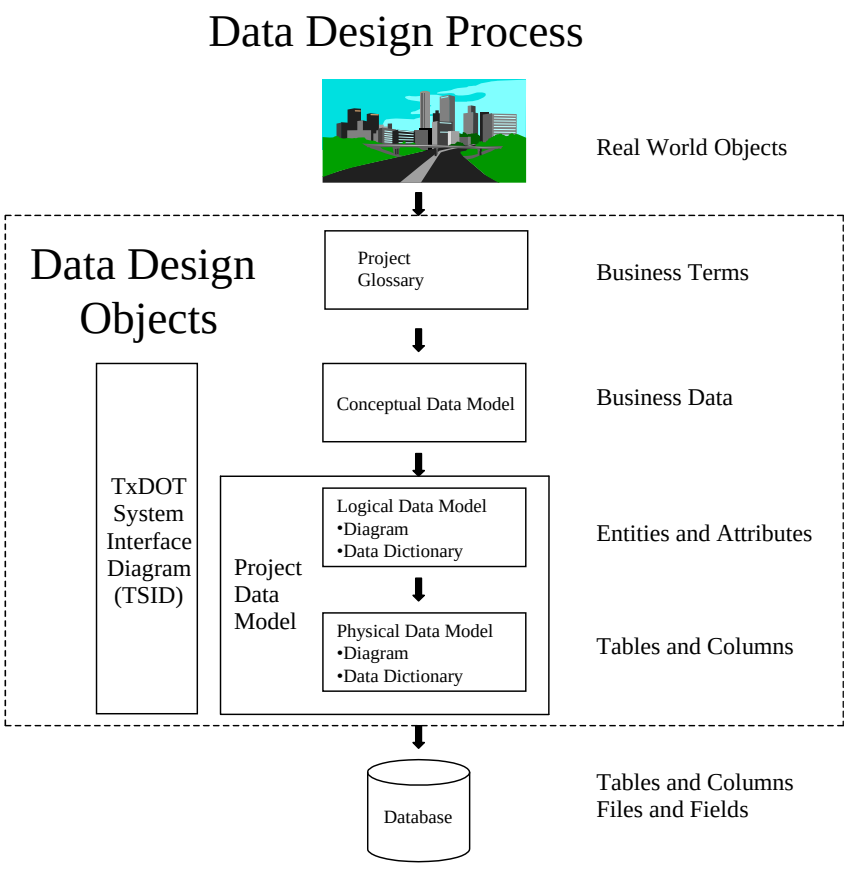
Overview

A sound design process must be followed in order to create integrated data. This section outlines the integrated methods to be followed in data design and the tools to be used.

The components needed to build integrated data at TxDOT are project glossaries, data models, and TxDOT System Interface Diagrams (TSID). Additionally, this section discusses the emerging technologies and considerations to provide a future perspective.

Data Design Process

The data design process documents information about real world objects. As projects move through the design process, the information documented becomes more and more detailed. It begins with broad user requirements and ends with detailed technical requirements for building a database. The resulting documents are data design objects.



(continued...)

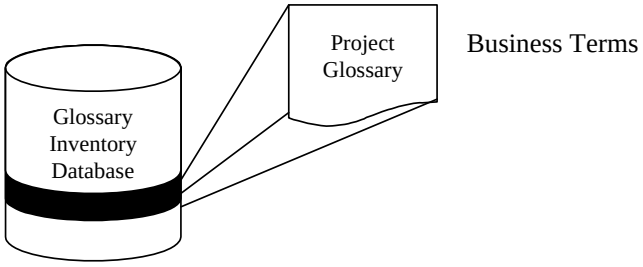
Data Design Process *(continued)*

As the data design objects are developed, they should be integrated with existing TxDOT data design objects in order to create integrated data. To begin integration, comparisons must be made. When a data design object is compared to other data design objects, variations or similarities may be discovered. These variations or similarities across projects could lead to data and application conflicts or redundancies.

Project Glossary

When a project starts, a project glossary should be developed so that all participants will have a common understanding of the terms used in the project. An extract from the Glossary Inventory Database is used to start the project glossary. Project specific terms, technology terms, or any term that may cause confusion should be included. Common terms, such as "program", "network", and "system" have different meanings to different people and should be defined for the project.

Glossary Integration



A glossary that defines and cross-references terms will help increase understanding and communication. When a project glossary is compared to other project glossaries, additional terms, meanings, or similarities may be discovered. These variations or similarities across projects could lead to data and application conflicts or redundancies if they are not clearly understood. The project glossary will be re-integrated into the project glossary database once it is completed. See Chapter 2 of this document for Glossary Guidelines.

Data Models

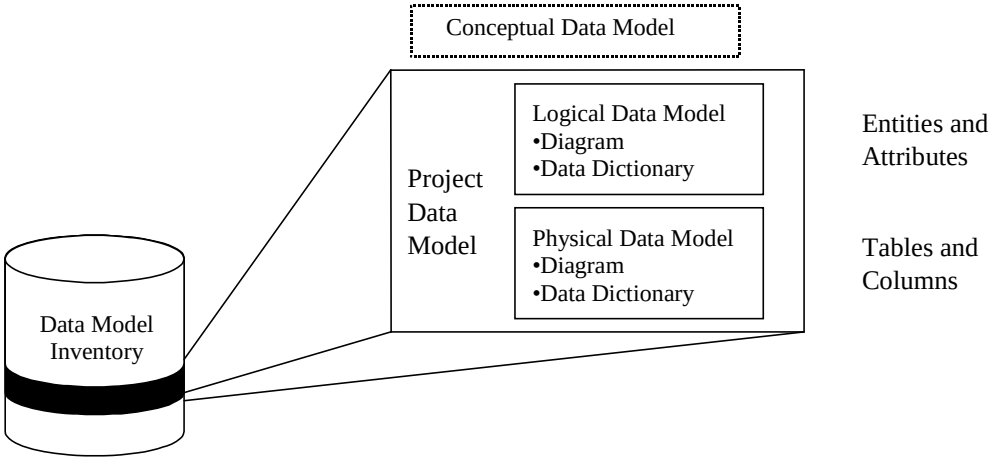
The purpose of data modeling is for all project participants to come to a complete understanding of the project data requirements. The end result of the data modeling process is a data model that documents the data requirements. See Appendix C "Data Model Interpretation" for additional information.

Data models are often compared because:

- A project may use data from other projects or applications
- Other projects or applications may use data being created.

When one data model is compared to other data models, variations or similarities may be discovered. These variations or similarities across projects (or applications) could lead to data and application conflicts or redundancies.

Data Model Integration



At TxDOT, a data model consists of a diagram and a data dictionary. TxDOT uses three types of data models:

- Conceptual Data Model (Recommended)

  A conceptual data model identifies data from the business viewpoint. It helps identify the scope and impact of the project. It defines the entities (persons, places, things, concepts and events) about which the business must keep data, and identifies and clearly defines the important associations between those entities.

- Logical Data Model (Mandatory)

  A logical data model formalizes the data requirements analysis process and identifies all business data and relationships within the scope of the project.

  A logical data model is:

  - Independent of the hardware or software requirements
  - Compliant with the TxDOT Data Standards. (See Chapter 2 of this document.)

(continued...)

Data Models *(continued)*

- Physical Data Model (Mandatory)

  A physical data model transforms a logical data model to meet the requirements of a specific Database Management System (DBMS).

  It represents how data is actually structured and stored.

  A physical data model is:

  - The physical implementation of the logical data model
  - Compliant with the TxDOT Data Standards. See Chapter 2 of this document.

TxDOT System Interface Diagram (TSID)

TxDOT System Interface Diagrams (TSIDs) are a series of graphical diagrams that document the relationships between computer applications and data. TSIDs are created and maintained by the Data and Quality Management Services Branch (DAQ) of the Technology Services Division (TSD).

In cases where there is a complex web of interconnected data and applications, a diagram can help to understand and visualize the dependencies. Understanding these dependencies can be very helpful in performing the analysis necessary to revise applications or to add functionality.

If the application will interface with other applications or data structures, DAQ will work with the project team during the requirements analysis process to begin creation of the TSID.

The TSID should identify the databases created for and used by the project computer application, as well as other databases used by the project computer application. It should also identify other computer applications that will use the databases created by the project. At implementation time, all interfaces will be documented in the finalized TSID. Updates are required when enhancements are made to the application.

For more information, see section on "TSID Interpretation" in Appendix D of this document.

Current TSIDs are available on the TxDOT Online Data and Application Inventory (DAIS) at: http://crossroads/apps/dais/.

Standards Documents

The process of integrating data is enhanced if the data design objects, created during the process, are themselves integrated and documented in the same way. The following data standards documents support the creation of integrated data design objects and integrated data.

- TxDOT Data Standards include standards for diagramming, naming, and defining data. For more information on the standards, see Chapter 2 of this document.
- TxDOT Data Naming Class Words are standard words that are used within an attribute name. For more information, see Chapter 3 of this document.
- TxDOT Data Naming Abbreviations is a document that defines the use of standard abbreviations. See Chapter 4 of this document for more information. Also, the standard abbreviations database, used for creating physical data names, is available on the DAIS webpage at: http://crossroads/apps/dais/.

Specialized Standards

The TxDOT Geographic Information Systems (GIS) Data Standards are included in Chapter 5 of this document.

Understanding Data Design Objects

Successfully integrating data is also dependent on understanding data design objects. The following guides provide such information:

- Data Normalization is a guide to understanding data design and normalization, and can be found in Appendix B of this document.
- Data Model Interpretation is a guide to understanding how to read data models. It is included in Appendix C of this document.
- TSID Interpretation is a guide to understanding how to read TxDOT System Interface Diagrams (TSIDs). It is included in Appendix D of this document.

Software

Currently, TxDOT employs several tools in performing data modeling, analysis, and management. The following table illustrates these tools:

Software Supporting the TxDOT Data Architecture

Software Name	Description
ER/Studio	ER/Studio from Embarcadero Technologies, Inc. is the data modeling tool used to create TxDOT data models. The version used must be compatible with the version of ER/Studio Repository installed at TSD. TSD can supply copies of ER/Studio for use in projects that are developing databases.
Embarcadero Schema Examiner	Embarcadero Schema Examiner from Embarcadero Technologies, Inc. is a tool used by TSD to analyze the structure of databases and data models. It provides detailed reports of possible design errors and suggested performance improvements.
ER/Studio Repository	ER/Studio Repository from Embarcadero Technologies, Inc. is the repository used by TSD to store data models.
Data and Application Inventory System	The Data and Application Inventory System (DAIS), developed by DAQ, is the repository used by TSD to store and manage information about TxDOT test and production data. It defines the current data infrastructure and is available to all of TxDOT on the Intranet at: http://crossroads/apps/dais/
PREDICT Data Dictionary (Legacy)	PREDICT, from Software AG, is an online interactive data dictionary facility that provides the capability of creating, maintaining and retrieving current dictionary information about ADABAS files and their use. PREDICT is also used, to a limited extent, to document some VSAM, Sequential, and other files.
Visio	Visio is used by DAQ to develop and maintain TxDOT System Interface Diagrams (TSIDs).
DBMS Software	See "Core Technology Architecture" for information.

Emerging Technologies and Future Considerations

As new data management tools are developed, they will be reviewed for potential use within the TxDOT Data Architecture.

New Versions of ER/Studio
ER/Studio will be maintained at the latest upgrade or version to be compatible with the latest version of ER/Studio Repository installed at TSD.

New Versions of ER/Studio Repository
As new versions of ER/Studio Repository become available, they will be tested and the existing data models will be converted.

ER/Studio Portal

The ER/Studio Portal will allow TxDOT to view application data models for development (dv), unit test (ut), system test (st), and production (pd) environments.

Data Warehousing
A data warehouse is a subject-oriented, integrated, time-variant, non-volatile collection of data in support of the decision-making process. As appropriate projects arise, data warehousing will be considered.

This page intentionally left blank.

Chapter 2
Data Standards

Contents:

This page intentionally left blank

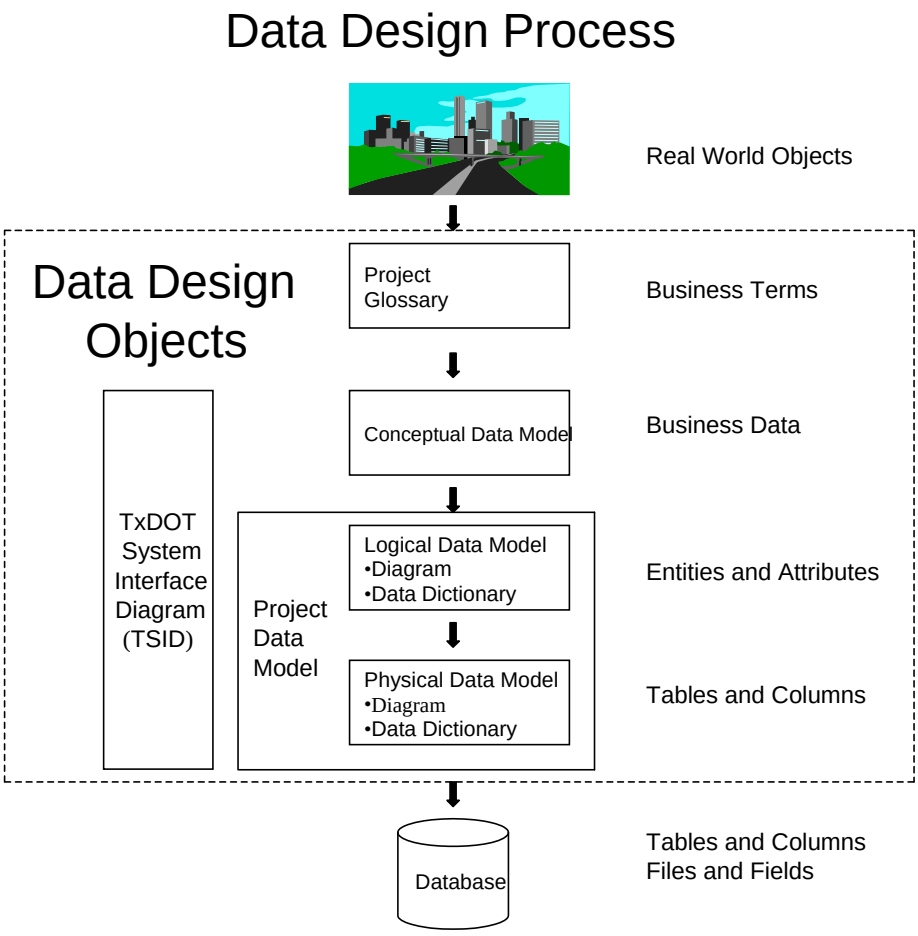
Section 1

Overview

Introduction

This chapter explains the standards and guidelines to be used for the creation and documentation of data at the Texas Department of Transportation (TxDOT). Adherence to these standards will promote the development of well-organized data that can be more easily maintained and integrated with other department data.

These data standards apply to data design objects and should be used by anyone creating or documenting TxDOT data.



(continued...)

Introduction *(continued)*

The following table gives definitions and requirements for data design objects.

Definitions and Requirements

Document	Definition	Requirements
Project Glossary	A glossary is a collection of specialized terms with their meanings. The glossary definitions provide an understanding of the context within which data is used.	Recommended
Conceptual Data Model	A conceptual data model depicts data at a high-level in business terms.	Recommended
TxDOT System Interface Diagram (TSID)	TxDOT System Interface Diagrams (TSID) are a series of graphical diagrams that document the relationships between an application, data created and maintained by the application, and data created and maintained by other applications.	Recommended
Logical Data Model	A logical data model depicts data in detail for the purpose of business understanding. It formalizes the data requirements analysis process and identifies all business data and relationships within the scope of the project. The required components of a logical data model are the Entity-Relationship Diagram and the Data Dictionary.	Mandatory
Physical Data Model	A physical data model depicts data as it will be stored in a database. It is a design for a specific target Database Management System (DBMS) where the data will be physically stored, accessed, and maintained. The physical data model specifies implementation details that may be features of a particular product or version, as well as configuration choices for that database instance. The required components of a physical data model are the physical data schema and the data dictionary.	Mandatory

The following sections contain the guidelines and standards to be used for data design.

Section 2

Glossary

Overview

A glossary is a collection of specialized terms with their meanings, used to identify any process, event, place or thing of importance to TxDOT. A glossary is recommended for each project. Completed project glossaries are incorporated into the Project Glossaries Database by the Data and Quality Management Services Branch and maintained for use in future projects.

Glossary Guidelines

A glossary entry contains:

- The business term
- The short name, if one exists
- The definition stating the meaning of the word as used in the business world
- The source for the definition.

Definition Format

For consistency, a basic definition format was chosen. The definition format is as follows:

- Begin the sentence(s) with the term
- Follow the term with a short name, if one exists
- Follow the term or short name, whichever is last, with the word "is"
- Follow the word "is" with the definition of the term
- Add sentences as needed to provide additional clarification.

NOTE: The term or words comprising the entry should not be repeated in the definition unless they are defined elsewhere in the glossary or they are commonly understood.

Following the format outlined above will provide a definition that may be copied directly into documentation or other documents. Modifications will not be needed.

Example: Glossary Entry

Business Term	Definition	Source
concrete	Concrete is a composite material consisting of a binding medium within which are embedded particles or fragments of aggregate; in hydraulic cement concrete, the binder is formed from a mixture of hydraulic cement and water. NOTE: This definition has been approved by the Specification Committee.	Manual of Testing Procedures

Glossary Special Cases

- When more than one meaning for a business term exists, the separate definitions must be documented.

Example: Documenting Separate Definitions

Business Term	Definition	Source
subgrade	Subgrade (Subgr) is the upper portion of material that acts as a foundation for base course or pavement.	TxDOT Glossary
subgrade	Subgrade (Subgr) is the line that establishes the final elevation for cut and fill operations.	TxDOT Glossary
subgrade	Subgrade (Subgr) is the portion of roadway between crown lines.	TxDOT Glossary

- A business term is the user's term for the thing being identified. When more than one term exists for the same thing:
  - **One** must be selected as the official term
  - The other term must be identified as an alias with a reference to the official term
  - A short name may be entered in the glossary as an alias for the official terms.

Example: Identifying an Alias

Business Term	Definition	Source
Texas Transportation Commission	The Texas Transportation Commission (TTC) is the five-member board that governs the Texas Department of Transportation, which is headed by an executive director selected by the commission. The governor, with the advice and consent of the Texas Senate, appoints commission members, who serve overlapping six-year terms.	TxDOT Glossary
commission	Alias for: Texas Transportation Commission.	Contract Management Manual
TTC	Alias for: Texas Transportation Commission.	TxDOT Glossary

Short Names

A short name is displayed after the term in the definition and enclosed in parentheses. A short name is usually in capital letters.

A short name exists when it is in the form of *one* of the following:

- Terms created by shortening larger terms. Abbreviations are usually pronounced as the original word.

(continued...)

Short Names *(continued)*

- **Acronym**
  Terms created by using the initial letters of the several component words composing the name of any organization, program, or concept to create a new **word** that substitutes for the combination of the component words. Acronyms are pronounced as words.

- **Initialism**
  Terms created by using the initial letters of several component words composing the name of an organization, program, or concept, to create a new term those substitutes for the combination of the component words. Initialisms are pronounced by spelling out the letters.

Examples: Business Terms with Short Names

Term	Abbreviation	Acronym	Initialism
Statewide Traffic Analysis and Reporting System		STARS	
Traffic Operations Division	TRF		
Transportation Planning and Programming Division			TPP

This page intentionally left blank

Section 3

Data Modeling

Overview

The purpose of data modeling is for all project participants to come to a complete understanding of the project data requirements. Those requirements are documented in a data model.

There are three types of data models:

Conceptual
A conceptual data model shows data from a business viewpoint, and identifies persons, places, things, concepts and events.

The components for a conceptual data model are:

- A listing or display of high-level business data sufficient to define the scope of the project
- A glossary to serve as the data dictionary for the model.

  Glossary Guidelines

  The glossary should include all major data entities, as well as other important business terms for the project, and their definitions. The definitions will form the basis of the final data dictionary, and should be clearly understood by all participants.

Conceptual models are very business-oriented and at a very high level.

Logical
A logical data model formalizes the data requirements analysis process and identifies all business data and relationships within the scope of the project. This model will be discussed in detail in Section 4 of this chapter.

Physical
After the logical model has been developed, physical constraints are applied. The physical data model is used to show the specific requirements for the selected DBMS. This model is explained in Section 5 of this chapter.

This page intentionally left blank

Section 4

Logical Data Model

Overview

A logical data model formalizes the data requirements analysis process and identifies all business data and relationships within the scope of the project.

The following section explains the standards and guidelines for naming and describing the data objects represented in the logical data model. These include entities, attributes, and entity relationships. Standards to be followed in the creation of a data dictionary are presented. Normalization is discussed.

Standard Modeling Tool

- ER/Studio is TxDOT's standard data modeling tool for logical data models.

Components of a Logical Data Model

The required components of a Logical Data Model are:

- Entity-Relationship Diagram (ERD)
- Data dictionary.

Diagram

Diagram Standards

A diagram must:

- Use Information Engineering (IE) notation
- Include:
  - Model name
  - Last update date
  - Name of the person who made the last update

(continued...)

Diagram *(continued)*

- Be designed to third normal form
- Show entities
- Show attributes
- Identify primary, foreign, and alternate keys
- Identify entity relationships
- Show at least one direction (parent to child) of the entity relationship verb phrase.

Diagram Guidelines

Data modeling is an engineering discipline, not an artistic one. However, the way in which entities are placed on a page is critical to the viewer's ability to understand a diagram. For this reason, modelers should pay attention to aesthetics.

A diagram should:

- Have related entities grouped together
- Be printed so that the text is large enough to read and reproduce
- Not have relationship lines crossing each other, if at all possible.

NOTE: When a model includes 10 or more entities, subject areas should be created for the major entities. Subject areas are groupings of 10-15 entities that a modeler would typically display together. Subject areas should include only the closely related entities that are generally no more than two relationships away from the major entity. The use of subject areas makes the model easier to read.

Components of a Diagram

The components of a diagram are:

- Entity
- Entity relationship.

Entity

An entity is a person, place, thing, concept or event that the business must manage and for which data is stored.

Entity Naming Standards

Entity names must be:

- Alphanumeric with no special characters
- Separated with a space between each word
- In upper case letters.

Entity Naming Guidelines

Entity names should:

- Use fewer than six words
- Use singular nouns ("*NOTES*" would become "*NOTE*")
- Use verbs in present tense ("*COMPLETED*" would become "*COMPLETE*")
- Use the root form of words ("*CANCELLATION*" would become "*CANCEL*")
- Be derived from the data description
- Be as meaningful as possible
- Be easily distinguishable.

Special Cases for Entities

- Abbreviations

  Abbreviations generally are not used in logical names, but there are exceptions. For example, when an acronym, abbreviation or initialism is normally used by the business user in discussion or memos, then that acronym, abbreviation or initialism may be used (e.g., *FTE*). See Section 2 (Glossary) of this chapter for more information on these terms.

  When the logical name would be excessively long if not abbreviated, the words that most commonly appear in an abbreviated form are acceptable. For example, "*YTD*" would be an acceptable abbreviation for "*YEAR-TO-DATE*" in a long logical name.

  If an abbreviation is used, then the full name must be included in the data dictionary. For more information on abbreviations, see Chapter 4 of this document.

- Hyphens

  Hyphens ("-") must only be used for words that are hyphenated in normal use.

(continued...)

***Entity** (continued)*

Attributes

An attribute is a characteristic that describes or identifies an entity.

Attribute Standards

An attribute must:

- Describe one characteristic of an entity
- Have only one description
- Be in its smallest meaningful unit (see example below).

Example: Attribute Standard

Entity	Attribute(s)
EMPLOYEE	EMPLOYEE ID EMPLOYEE LAST NAME EMPLOYEE FIRST NAME EMPLOYEE MIDDLE NAME EMPLOYEE BIRTH DATE

Attribute Guideline

If they exist:

- Indicate natural primary keys.

Attribute Naming Standards

Attributes names must:

- Be in upper case letters
- Be alphanumeric with no special characters
- Be separated with a space between each word
- Reflect the data that is actually stored
- Be composed of:
  - One or more prime words
  - Zero or more qualifier words
  - One class word.

(continued...)

***Entity** (continued)*

- End with a class word selected from the list of standard class words.

Example: Prime Words, Qualifier Words, Class Words

Logical Name	Prime Word	Qualifier Word(s)	Class Word
EMPLOYEE ID	EMPLOYEE		ID
EMPLOYEE LAST NAME	EMPLOYEE	LAST	NAME

Attribute Naming Guidelines

Attribute names should:

- Use fewer than six words
- Use singular nouns ("*NOTES*" would become "*NOTE*")
- Use verbs in present tense ("*COMPLETED*" would become "*COMPLETE*")
- Use the root form of words ("*CANCELLATION*" would become "*CANCEL*")
- Be derived from the data description
- Be as meaningful as possible
- Be easily distinguishable.

Special Cases for Attributes

- Abbreviations

  Abbreviations generally are not used in logical names, but there are exceptions. For example, when an acronym, abbreviation or initialism is normally used by the business user in discussion or memos, then that acronym, abbreviation or initialism may be used (e.g., *FTE*).

  When the logical name would be excessively long if not abbreviated, the words that most commonly appear in an abbreviated form are acceptable. For example, *"YTD"* would be an acceptable abbreviation for "*YEAR-TO-DATE"* in a long logical name.

  The class word *"IDENTIFIER"* should always be used in its abbreviated form *"ID"*.

  For more on abbreviations, see Chapter 4 of this document.

- Hyphens

  Hyphens ("-") must only be used for words that are hyphenated in normal use.

- Foreign Keys

  Foreign Key attributes should inherit names from the Primary Key to which they are constrained unless there is a need to rename them.

(continued...)

***Entity** (continued)*

Foreign Key attributes should be renamed only when there is:

- More than one relationship from the same entity
- A recursive relationship
- Business user convention.

Entity Relationship

Most data oriented business rules are modeled as relationships between entities. An entity relationship reflects the fact that two data entities interact in some way that we want to record. The interaction is described with a verb phrase.

Cardinality expresses the minimum and maximum expected number of related occurrences between the two entities involved in a given relationship. Cardinality is defined by the symbols used at the start and the end of the relationship line.

For example: Each *CUSTOMER* places zero, one, or more *ORDER*.

In practice, entity relationships have two names: one for each direction of the relationship.

For example: If "Each *CUSTOMER* places zero, one, or more *ORDER*", then "Each *ORDER* is placed by one *CUSTOMER*" may be the second name of the relationship.

Entity Relationship Standards

The entity relationship must have:

- Verb phrase in lower case
- Foreign keys renamed to identify the relationships they support when multiple relationships exist between two entities.

Entity Relationship Guidelines

Entity relationships should:

- Use an active verb phrase, when possible.

Data Dictionary

Data Dictionary Standards

The data dictionary must:

- Include business descriptions of entities and attributes
- Be included with the diagram in the standard modeling tool.

Entity Description Standards

An entity description must contain:

- "Definition:" followed by a definition.

Entity Definition Standards

An entity definition must:

1. Begin with the word "Definition" followed by a colon
2. *Next*, show the logical entity name
3. Follow the entity name with the word "is"
4. Follow the word "is" with the definition.

NOTE: An entity definition should not contain a repeat of the entity name within the definition, if at all possible.

Example: Entity Definition

Entity	Description
DISTRICT	**Definition**: A **DISTRICT** is one of 25 geographical areas within the state of Texas where the Texas Department of Transportation conducts its primary work activities.

Special Case

Entity Subtype: If the entity is an entity subtype, the definition must identify the supertype.

Attribute Description Standards

Each attribute description must contain:

- "Definition:" followed by a definition
- "Purpose:" followed by a purpose
  The purpose describes why something exists in the database but not how it is processed.

(continued...)

Data Dictionary *(continued)*

In addition, an attribute description must contain either:

- "Examples:" followed by examples and an explanation of each example if the values are coded.
  Examples are items or incidents that are representative of a group or a type. Examples should be used when the list of valid values is extensive or can change.

or

- "Valid Values:" followed by valid values and an explanation of each value if the values are coded.
  Valid values include all of the permissible values. When there are many different values or the set of permissible values changes frequently, use Examples instead of Valid Values. However, the dictionary must indicate where a list of these can be found if they are not included in a table in the database.

And must contain:

- "Format:" followed by a display format (String, Number, Datetime, Blob).
  If appropriate, include the display format of the data. For example, "123-12-1234".

Use "N/A" after Examples, Valid Values, or Format headings when not applicable.

Example: Attribute Description

Attribute	Description
DISTRICT NUMBER	**Definition:** A **DISTRICT NUMBER** is a distinguishing number that provides a designation for a DISTRICT. **Purpose: A** DISTRICT NUMBER provides a recognizable identifier that uniquely identifies a DISTRICT. **Examples**: N/A **Valid Values:** 01 THROUGH 25 **Format:** Number
DISTRICT NAME	**Definition:** A **DISTRICT NAME** is a word or phrase that provides a designation for a DISTRICT. **Purpose:** The **DISTRICT NAME** helps identify a DISTRICT. **Example: PARIS** **HOUSTON** **Valid Values:** N/A **Format:** String

DISTRICT ABBREVIATION NAME	**Definition:** A **DISTRICT ABBREVIATION NAME** is a shortened word or phrase that provides a designation for a DISTRICT. **Purpose:** The DISTRICT ABBREVIATED NAME helps identify a DISTRICT more clearly than a DISTRICT NUMBER. **Example: PAR** = PARIS **HOU** = HOUSTON **Valid Values:** N/A **Format:** String

Attribute Definition Standards

The attribute definition must:

1. Begin with the word "Definition" followed by a colon
2. *Next*, show the attribute name
3. Follow the attribute name with the word "is"
4. Follow the word "is" with the definition.

Special Cases for Attributes

1. Foreign Keys

   Foreign Key attributes should inherit definitions from the Primary Key to which they are constrained unless there is a need to redefine them (e.g., the attribute name was changed).

NOTE: For assistance in defining an attribute, refer to the class word definition table in Chapter 3 of this document.

Normalization

Normalization is the decomposition of complex data structures into simpler, more stable structures using a rigorous set of analytical steps. Normalized entities contain non-repeating, non-redundant attributes.

The objective of normalization is to isolate data so that additions, deletions, and modifications of an attribute (field or column) can be made in just one entity (file or table) and then propagated through the rest of the database via the defined relationships.

Normalization Standard

- The standard for logical data models at TxDOT is third normal form (3NF).

For a detailed discussion of normalization, see Appendix B of this document.

This page intentionally left blank.

Section 5

Physical Data Model

Overview

A physical data model represents an implementation of the logical data model in a particular Relational Database Management System (RDBMS).

The following section explains the standards and guidelines for naming and describing the database objects represented in the physical data model. These include tables, columns, and foreign key constraints.

Standards to be followed in the creation of data dictionary entries for database objects are presented. NOTE: The data dictionary describes entities and attributes in the logical data model, and the corresponding database objects in the physical data model. Not all objects exist in both the physical and logical data models. When they do, however, they must match the corresponding objects in the logical data model.

Standard Data Modeling Tool

ER/Studio is TxDOT's standard data modeling tool for physical data models.

Components of a Physical Data Model

The required components of the physical data model are:

- Diagram (physical data schema)
- Data dictionary.

  NOTE: For reverse engineered data models, a data dictionary must be created in ER/Studio

Components of a Physical Data Model

Diagram Standards

A diagram must:

- Use Information Engineering (IE) notation
- Include:
  - Model name
  - Last update date
  - Name of the person who made the last update.
- Show tables
- Show columns
- Identify primary, foreign, and alternate keys.

Diagram Guidelines

A diagram should:

- Not have relationship lines crossing
- Have related entities grouped together
- Be printed so that the text is large enough to read and reproduce.

Components of a Diagram

The components of a diagram are:

- Tables (files)
- Columns (fields) comprising the tables (files)
- Table relationships.

Tables (Files)

An *entity* in the logical data model corresponds to a *table (file)* in the physical data model. The physical data model may contain some tables for which there are no corresponding entities in the logical data model.

Physical data model table (file) names are created from logical data model entity names by replacing each word in the logical name with a short name, and replacing spaces in the logical name with the standard word delimiter special character.

(continued...)

Components of a Diagram *(continued)*

Table Naming Standards

A table name must:

- Be in upper case
- Not exceed 28 characters
- Use the standard word delimiter special character.
  The word delimiter (the space character) in logical names is replaced by the word delimiter special character in physical names. The word delimiter special character is the underscore character ("_").

  NOTE: Legacy data will be handled on a case-by-case basis.

Example: Table (File) Name

Logical Name	Physical Name
COMPUTER APPLICATION	COMP_APPLN

Table Naming Guidelines

- When the same table exists on different platforms, different physical names for the same table may be required. The Data and Quality Management Services Branch (DAQ) will work with the developer to establish a standard alias for that platform.

Columns (Fields)

An *attribute* in the logical data model corresponds to a *column (field)* in the physical data model. The physical data model may contain some columns for which there are no corresponding attributes in the logical data model, and vice versa (attribute with no column), due to the physical constraints/requirements of the platform or database.

Physical data model column (field) names are created from logical data model attribute names by replacing each word in the logical name with a short name, and replacing spaces in the logical name with the standard word delimiter special character.

(continued...)

Components of a Diagram *(continued)*

Column Naming Standards

A column name must:

- Be in upper case
- Not exceed 28 characters
- Use the standard word delimiter special character.

  The word delimiter (the space character) in logical names is replaced by the word delimiter special character in physical names. The word delimiter special character is the underscore character ("_").

  NOTE: Legacy data will be handled on a case-by-case basis.

Example: Column (Field) Names

Logical Name	Physical Name
EMPLOYEE LAST NAME	EMP_LAST_NM

Column Naming Guidelines

- Different platforms may require different physical names for the same column/field due to specific platform constraints.
- In cases where further shortening of the physical name is required or desirable, it is possible, with discretion, to substitute a short name for a group of words.
- Foreign Key columns should inherit names from the Primary Key to which they are constrained unless there is a need to rename them (e.g., more than one relationship from the same table, a recursive relationship, etc.)

The Data and Quality Management Services Branch will provide additional guidance in these cases.

Example: Column (Field) Name

Logical Name	Physical Name
COMPUTER APPLICATION ID	*CA_ID*

Table Relationships

The implementation of relationships in a physical database ensures that referential integrity is maintained. Referential integrity is a feature provided by relational database management systems that prevents users or applications from entering inconsistent data.

Foreign Key Constraint Guidelines

- Table relationships should be implemented using the built-in foreign key constraint facilities of the relational database management system.
- Multiple column foreign keys should be NOT NULL in order to maintain database referential integrity. If the data can be optional, the Primary Key of the parent table should be changed to a surrogate key. If the foreign key consists of multiple columns, and *any* column is NULL, the whole key is considered NULL. The insert is permitted no matter what is on the non-null columns.

Data Dictionary

A Data Dictionary includes a business description of entities and attributes. The descriptions apply for the corresponding objects in the physical data model.

Data Dictionary Standard

- The data dictionary must be included with the diagram in the standard modeling tool.

Table Description Standards

Table descriptions in the physical data model are the same as the entity descriptions in the logical data model. For tables that appear only in the physical model, a table description must be written.

A table description must contain:

- "Definition:" followed by a definition.
- Documentation of any denormalization.

Table Definition Standards

A table definition must:

1. Begin with the word "Definition" followed by a colon
2. *Next*, show the **logical entity name**
3. Follow the entity name with the word "is"
4. Follow the word "is" with the definition.

(continued...)

Data Dictionary *(continued)*

Example: Table Description

Table	Description
DIST	**Definition**: A **DISTRICT** is one of 25 geographical areas within the state of Texas where the Texas Department of Transportation conducts its primary work activities.

Column Description Standards

A data column (field) description is the same as the corresponding attribute description in the logical data model. For columns that appear only in the physical model, a column description must be written.

A column description must contain:

- "Definition:" followed by a definition
  The definition must contain the **logical attribute name** as it appears (or would appear) in the logical data model diagram.
- "Purpose:" followed by a purpose
  The purpose describes why something exists in the database, but not how it is processed.

In addition, a column description must contain either:

- "Examples:" followed by examples
  Examples are items or incidents that are representative of a group or a type

  or

- "Valid Values:" followed by valid values
  Valid values include all permissible values and an explanation of each value. When there are many different values or the set of permissible values changes frequently, the data dictionary should not be used to document all possible values. However, the description must indicate where a list of these can be found.

And must contain:

- "Format:" followed by a display format.
  If appropriate, include the display format of the data. For example, "123-12-1234."

NOTE: Use N/A after Examples, Valid Values, or Format headings when not applicable.

(continued...)

Data Dictionary *(continued)*

Example: Column Description

Column	Description
DIST_NBR	**Definition:** A **DISTRICT NUMBER** is a unique 2 digit identifier for a DISTRICT. **Purpose:** The DISTRICT NUMBER is used to identify transactions related to a DISTRICT. **Examples**: N/A **Valid Values:** 01 THROUGH 25 **Format:** Number
DIST_NM	**Definition:** A **DISTRICT NAME** is a word or phrase that provides a distinctive designation for a DISTRICT. **Purpose:** The **DISTRICT NAME** is used as a common identifier to denote transactions related to a DISTRICT. **Example: PARIS** **HOUSTON** **Valid Values:** N/A **Format:** String
DIST_ABRVN_NM	**Definition:** A **DISTRICT ABBREVIATION NAME** is a shortened word or phrase that provides a distinctive designation for a DISTRICT. **Purpose:** The DISTRICT ABBREVIATION NAME is an identifier used on reports, in lieu of the DISTRICT NUMBER. **Example: PAR** = PARIS **HOU** = HOUSTON **Valid Values:** N/A **Format:** String

Column Definition Standards

The column definition must:

1. Begin with the word “Definition” followed by a colon
2. *Next*, show the **logical attribute name**
3. Follow the attribute name with the word “is”
4. Follow the word “is” with the definition.

Special Cases for Columns

1. Foreign Keys

   Foreign Key columns should inherit definitions from the Primary Key to which they are constrained unless there is a need to redefine them (e.g., the column name was changed).

This page intentionally left blank

Chapter 3
Data Names

Contents:

This page intentionally left blank.

Section 1

Data Name Structure

Overview

Each name in a logical data model is referred to as a logical (business) name and follows a set formula for composition.

Logical names are composed of:

- One or more prime words
- Zero, one, or more qualifier words
- One class word.

Each type of word, comprising the logical name, is discussed in the following section. Section 3 provides a listing of current TxDOT class words. Section 4 provides information on selecting the appropriate class word. Section 5 provides a listing of logical naming standards for common attributes.

This page intentionally left blank.

Section 2

Prime, Qualifier, and Class Words

Prime Word

The **prime word(s)** represents the subject or entity name. In the example below, EMPLOYEE is the prime word in the logical name EMPLOYEE HIRE DATE.

Qualifier Word

A **qualifier word(s)** is a descriptive word that further defines and/or distinguishes those things labeled with **class** and **prime words**. In the example below, HIRE is the qualifier word in the logical name EMPLOYEE HIRE DATE. It is not always necessary to have qualifier words. For examples of frequently used qualifier words, see Section 4.

Class Word

A **class word** is the highest level of qualification and the most important *word* in an attribute name.

The class word:

- Is always the last word of the attribute name
- Indicates the type of attribute being named
- Must be selected from the list of class words in Section 3.

If a new class word is needed, contact the Data and Quality Management Services Branch. Proposed class words should include a definition, purpose, examples, and format.

Examples

The following provides examples of two logical names.

Business (Logical) Name	Prime Word	Qualifier Word(s)	Class Word
EMPLOYEE HIRE DATE	EMPLOYEE	HIRE	DATE
DISTRICT ABBREVIATION NAME	DISTRICT	ABBREVIATION	NAME

This page intentionally left blank.

Section 3
TxDOT-Specific Class Words

TxDOT Class Words

The following Class Words are to be used in naming TxDOT data. Please note that examples may not represent actual TxDOT data.

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
ADDRESS	ADDR	**Definition**: An *ADDRESS* is a specific location at which to find or communicate with a person or an organization. **Purpose:** ADDRESS is generally used for postal mailing or physical addresses, but may be used for things such as email addresses or URL addresses. **Examples**: *RESIDENCY ADDRESS, APPLICANT STREET ADDRESS, URL ADDRESS, EMAIL ADDRESS* **Valid Values**: N/A **Format**: String
AMOUNT	AMT	**Definition**: An *AMOUNT* is any quantity of money. **Purpose:** AMOUNT is always used to identify money. **Examples**: *UNIT COST AMOUNT, ORIGINAL CONTRACT AMOUNT* **Format**: Number
ANGLE	ANGL	**Definition**: An ANGLE is the rotational measurement between two lines and/or planes diverging from a common point and/or line. **Purpose:** ANGLE is used for measuring rate of separation between lines or planes. **Examples**: *AZIMUTH ANGLE, INCLINATION ANGLE, SKEW ANGLE* **Valid Values**: N/A **Format**: Numeric degrees
BLOB	BLOB	**Definition**: A Binary Large Object (BLOB) is a collection of binary data stored as a single entity in a database management system (DBMS). **Purpose:** BLOBs are used primarily to hold multimedia objects such as images, videos, and sound. **Examples:** PDF files, WAV files, TIFF files **Valid Values**: N/A **Format**: Blob

(continued...)

TxDOT Class Words *(continued)*

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
CODE	CD	**Definition**: A *CODE* is a brief indicator that identifies an entity, usually a domain entity. It can have a range of more than two values. Possible values are sometimes contained in a table. **Purpose:** CODE is used for meaningful short identifiers known to the business, in place of NAME or DESCRIPTION. The values usually provide a clue to their meaning, but not always. *CODE* is used only if data values in NAME or DESCRIPTION are coded. **Examples**: *ESTIMATE STATUS CODE, CONTRACTOR PAY CODE* **Valid Values**: N/A **Format**: String
COMMENT	CMNT	**Definition**: A *COMMENT* is a descriptive note or remark about an entity. **Purpose:** COMMENT is generally used to provide additional information or remarks. **Examples**: *ENGINEER COMMENT, STUDENT COMMENT* **Valid Values**: N/A **Format**: String
DATE	DT	**Definition**: A *DATE* is an actual calendar point in time. It **must** be qualified by the following if not the standard format of *YYYYMMDD:* ♦ YEAR(YR) format YYYY ♦ FISCAL YEAR (FY) format YYYY ♦ QUARTER(QTR) format N ♦ MONTH(MTH) format MM ♦ WEEK(WK) format NN ♦ DAY(DAY) format DD **Purpose: DATE is generally used to identify a specific calendar day, but may be qualified to identify a different specific calendar unit.** **Examples**: *CLEAR DATE, BEGIN DATE, PURCHASE DATE, FISCAL YEAR DATE, EFFECTIVE YEAR MONTH DATE* **Valid Values**: N/A **Format**: Datetime
DESCRIPTION	DSCR	**Definition**: A *DESCRIPTION* is explanatory information about the entity. **Purpose:** DESCRIPTION provides clarification and meaning for the specific entity. **Examples**: *COURSE DESCRIPTION, MATERIAL DESCRIPTION* **Valid Values**: N/A **Format**: String

(continued...)

TxDOT Class Words *(continued)*

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
DURATION	DUR	**Definition**: *DURATION* is a time interval or duration representing an elapsed time or the length of time an event lasted. It **must** be qualified by: *YEARS(YR), QUARTERS(QTR), MONTHS(MTH), WEEKS(WK), DAYS(DAY), HOURS(HR), MINUTES(MNTE), SECONDS(SCND)* **Purpose:** DURATION is used to identify elapsed calendar or clock time. **Examples**: *PROJECT DAYS DURATION, ESTIMATE YEARS DURATION* **Valid Values**: N/A **Format**: Number
ELEVATION	EL	**Definition**: An *ELEVATION* is the vertical distance of a point or object above or below a reference surface such as a vertical datum (generally mean sea level). **Purpose**: Elevation, latitude, and longitude can be used in combination to locate any point on the earth's surface in three dimensions. **Example**: +10297.556 or –45.265 **Valid Values**: N/A **Format**: Number Elevation is stored as numeric with 5 places to the left and 3 places to the right of the decimal point and as a positive or negative value.
FLAG	FLAG	**Definition**: A *FLAG* is an indicator having a range of only two values that indicates that some event has happened or should happen. **Purpose: FLAG indicates** yes/no, true/false, off/on, etc. **Examples**: *ESTIMATE FLAG, DELETE FLAG* **Valid Values**: N/A **Format**: String
IDENTIFIER	ID	**Definition**: An *IDENTIFIER* is a distinguishing word or number that uniquely identifies an entity. It is a unique representation or identification, assigned to anything in order to fix its position in a series. **Exception to naming standard**: The abbreviated form ID must always be used. **Purpose:** IDENTIFIER is generally used for machine generated surrogate primary keys, in which case they would have no business meaning. However, business users use IDENTIFIER for coded labels or names to represent and track individual business entities. **Examples**: *OPERATOR ID, TERMINAL ID, SECURITY BADGE ID* **Valid Values**: N/A **Format**: Number (surrogate key) or String (user identified)

(continued...)

TxDOT Class Words *(continued)*

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
LATITUDE	LAT	**Definition**: LATITUDE is the angular distance along a meridian north or south of the equator, usually measured in degrees, at which the entity is located. Lines of latitude are parallel to the equator. Lines of latitude are also called parallels. **Purpose**: The intersection of latitude and longitude can be used in combination to locate any point on the earth's surface. **Example**: +30.31166667 **Valid Values**: N/A **Format**: Number Latitude is stored in Decimal Degrees (DD) with 3 places to the left and 8 places to the right of the decimal point and as a positive or negative value. (In Texas this value is always positive.)
LONGITUDE	LON	**Definition**: A *LONGITUDE* is the angular distance, expressed in degrees, of a point on the earth's surface east or west of a prime meridian (usually the Greenwich meridian). All lines of longitude are great circles that intersect the equator and pass through the north and south poles. **Purpose**: The intersection of latitude and longitude can be used in combination to locate any point on the earth's surface. **Example**: -97.75611111 **Valid Values**: N/A **Format**: Number Longitude is stored in Decimal Degrees (DD) with 3 places to the left and 8 places to the right of the decimal point and as a positive or negative value. (In Texas this value is always negative.)

(continued...)

TxDOT Class Words *(continued)*

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
MEASUREMENT	MS	**Definition**: A *MEASUREMENT* is the number of units of measure that is a dimension of an entity, and the precision with which it is stored, determined by comparison to some standard unit of measure. **Purpose**: Usually, when a measurement is recorded in a database, it is recorded in the same unit of measure and/or on the same scale. For instance, in any given file, distance may always be recorded in miles (unit of measure). Temperature is virtually always recorded in degrees (unit of measure) and in any given file, it will usually be in either Fahrenheit or Celsius (scale). In such cases, the description of the measurement in the data dictionary **must** state which unit of measure is in use and, if applicable, which scale. Occasionally, a measurement is recorded in a database in such a way that each instance may not have the same unit of measure or scale. In those cases it is necessary to record which unit of measure and/or scale is in use for each instance. For example, if distance can be recorded in either miles or kilometers, each record should also contain an element to tell which unit of measure was used in that record. So you would have an element named *DISTANCE MEASUREMENT* and another named *DISTANCE UNIT NAME*. In the first one you would have the number that represented the number of units of distance, and in the second, you would have the word "mile" or "kilometer" to indicate what unit of measure was used in this record, for that specific measurement. Similarly, if temperature was to be recorded in more than one scale, you would need an element called *TEMPERATURE MEASUREMENT* and one called *TEMPERATURE SCALE NAME*. The first one would contain a numerical value that represented the number of degrees and the second would have the word "Fahrenheit" or "Celsius". (There would be no need for *TEMPERATURE UNIT NAME* because both versions would use "degree," and it would be specified in the description of the measurement.) The Class Word should be used for, and should be qualified by: *ALTITUDE, AREA, CAPACITY, DEPTH, DISTANCE, HEIGHT, LENGTH, RATIO, SPEED, TEMPERATURE, VOLUME, WIDTH,* or some other dimension name. **Examples**: *RIGHT OF WAY WIDTH MEASUREMENT, PAVEMENT TEMPERATURE MEASUREMENT* **Valid Values**: N/A **Format**: Number

(continued...)

TxDOT Class Words *(continued)*

TxDOT-Specific Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
NAME	NM	**Definition**: A *NAME* is a word or phrase that provides a designation for an entity. **Purpose:** NAME is generally used for a non-coded identifier for an entity. **Examples**: *AREA ENGINEER NAME, EMPLOYEE LAST NAME, COLOR NAME, STATUS NAME* **Format**: String
NUMBER	NBR	**Definition**: A *NUMBER* is a distinguishing word or number that uniquely identifies an entity. **Purpose:** NUMBER is generally used by the business user as a recognizable identifier for a business entity. **Examples**: *PURCHASE ORDER NUMBER, BID ITEM NUMBER* **Valid Values**: N/A **Format**: String
PERCENT	PCT	**Definition**: A *PERCENT* is a ratio expressed as a part of a whole, generally expressed in hundredths as percentages. **Purpose:** PERCENT is used to identify part of a whole. **Examples**: *TIME PERCENT, RETAINAGE PERCENT* **Valid Values**: N/A **Format**: Number If the data is used for calculations, 50 ½ percent should be stored as ".505". If the data is used primarily for reporting, 50 ½ percent should be stored as "50.5". Either way, the description MUST identify how it is stored by using the statement: "50 ½ percent will be stored as 'xxx'".
QUANTITY	QTY	**Definition**: A *QUANTITY* is the count or number of occurrences. **Purpose**: QUANTITY is used for the count of anything except money. **Examples**: *RECEIPT QUANTITY, INVENTORY PLACED QUANTITY* **Valid Values**: N/A **Format**: Number
TEXT	TXT	**Definition**: *TEXT* is textual information having no defined format, but specific content as a string of characters. **Purpose**: *TEXT* is used to identify a string of characters. **Examples**: *SQL TEXT, PASSWORD TEXT* **Valid Values**: N/A **Format**: String

TIME	TM	**Definition**: *TIME* is an identifiable clock period during which an action, process or condition exists; the point or period when something occurs. It **must** be qualified by the following if not the standard format: *HOUR, MINUTE, SECOND* **Purpose:** TIME is used to identify a specific point of clock time. **Examples**: *START TIME, LAST UPDATE TIME* **Valid Values**: N/A **Format**: Datetime

ADABAS/PREDICT Class Words

The following class words are used only for implementation in ADABAS/PREDICT:

ADABAS/PREDICT Data Element Class Words

Class Word	Abbreviation	Definition/Purpose/Example/Format
GROUP	GRP	**Definition:** A **GROUP** is an association of a number of related elementary data field. **Purpose:** GROUP is used to identify a collection of related fields. **Examples:** *SLRY-CLASS-GRP, POST-INFO-GRP* **Valid Values**: N/A **Format:** N/A
KEY	KEY	**Definition:** A **KEY** is a group of fields used as an index. **Purpose:** KEY is used to identify a multi-column index. **Example**: *CCSJ-RCRD-KEY = CONTROL CSJ NUMBER + RECORD TYPE CODE* **Valid Values**: N/A **Format:** N/A

This page intentionally left blank

Section 4

Selecting the Appropriate Class Word

How to Determine the Class Word

For the programmer, one of the basic pieces of information that is needed before programming can begin is the type (or classification) of the attribute. (See Chapter 2, Section 4 for information on attributes.) If the type of attribute is known, the programmer can make assumptions about methods for inputting data or for accessing data. The types or classifications of attributes are designated by the Class Words.

The selection of the appropriate Class Word can sometimes be confusing. This section provides explanation, standards, and guidelines for identifying the appropriate Class Word.

Code

A code is a user-defined, short representation of data. Generally, codes have a range of more than two values. An example of a code would be the Highway System code. For instance, Interstate Highways are referred to by the code IH. State Highways are referred to by the code SH. Farm to Market roadways are referred to by the code FM.

For data in the code category, the programmer can assume a predefined list of values is available when inputting data into a database. As a result, the programmer can set up a "pick list" so that people that are entering data simply choose an item from the predefined list. "Pick lists" are an excellent way to ensure that data is input correctly. If you can only choose from a listing of highway system codes, incorrect codes can not be entered. That is, FN cannot be mistakenly typed in for FM.

Code Standards

Codes must:

- always be modeled as a separate entity in the logical data model
- include examples (for long lists) or valid values (for short or restricted lists) in the dictionary
- include an explanation in the dictionary for each code used in Examples or Valid Values.

(continued...)

Code *(continued)*

Examples: Dictionary Example Entries for Code with Name

Logical Data Name	Description
EMPLOYMENT STATUS CODE	Definition: … Purpose: … Examples: X = Former Employee, 1 = Regular Full-Time, 2 = Regular Part-Time Valid Values: N/A Format: …
EMPLOYMENT STATUS NAME	Definition: … Purpose: … Examples: Former Employee, Regular Full-Time, Regular Part-Time Valid Values: N/A Format: …

NOTE: Some codes may require both a name and a description. They may be defined within the table as data columns, but may also be required in Examples or Valid Values in the dictionary.

Examples: Dictionary Example Entries for Code with Name and Description

Logical Data Name	Description
VEHICLE TYPE CODE	Definition: … Purpose: … Example: LT = Light Truck = A light truck is any truck that weighs less than 10,000 pounds Valid Values: N/A Format: …
VEHICLE TYPE NAME	Definition: … Purpose: … Example: Light Truck Valid Values: N/A Format: …
VEHICLE TYPE DESCRIPTION	Definition: … Purpose: … Example: A light truck is any truck that weighs less than 10,000 pounds Valid Values: N/A Format: …

:

ID

An ID (identifier) is a number or a series of numbers and characters that uniquely distinguishes something. There are two types of ID, a Surrogate Key and a User-defined ID.

Surrogate Key

A surrogate key is a system generated number used as a Primary Key (PK). That is, a surrogate key is a number that uniquely identifies a record in the database, but does not have meaning for the user.

Surrogate Key Standards

- A surrogate key must be used when a natural key does not exist. NOTE: A natural key is a data column or set of data columns that uniquely identifies a record.
- When a surrogate key is used and a natural key exists, the natural key must be identified as a unique index (alternate key).

Surrogate Key Guideline

- A surrogate key may be used when a natural key is difficult to use. The programmer should work with the data modeler to determine if a surrogate key is needed.

CAUTION: Some system-generated numbers end up providing meaning to the user because there are no other unique single identifiers. Surrogate keys are sometimes referred to by the user as a type of number. For example, LICENSE APPLICATION NUMBER may be a system generated number that should properly be denoted as LICENSE APPLICATION ID.

User-defined ID

A user-defined ID is a number or string of numbers and characters that uniquely distinguishes something. Quite often it is a pre-existing identifier that carries meaning.

User-defined ID Standards

User defined IDs must:

- Have examples in the data dictionary
- Have a format description

Many user-defined IDs are commonly referred to by the user as an ID. For example, USER ID.

CAUTION: Sometimes user-defined IDs, like surrogate keys, are referred to by the user as a type of number.

Number

A number is an identifier for a specific occurrence, a position in a sequence, or a numeric or partly numeric designator. It may contain numbers and characters and quite often has a specific meaningful structure. For example; TELEPHONE NUMBER, ORDER NUMBER, SEQUENCE NUMBER, TXDOT ROUTE NUMBER, etc.

Number Standards

Numbers must:

- Have examples in the data dictionary
- Have a format description if it contains structure

Example: Dictionary Example/Format Entry for Number

CONTROL SECTION JOB NUMBER	Definition: … Purpose: … Examples: 123456789 Valid Values: N/A Format: String - CCCCSSJJJ, where CCCC = Control Number, SS = Section Number, JJJ = Job Number

Date

A date is an actual calendar point in time.

Date Standards

Dates must:

- Have examples in the data dictionary
- Have a format description

Example: Dictionary Example/Format Entries for Date

EMPLOYEE HIRE DATE	Definition: … Purpose: … Examples: 20050901 Valid Values: N/A Format: Datetime - YYYYMMDD
PROJECT FISCAL YEAR DATE	Definition: … Purpose: … Examples: 2005 Valid Values: N/A Format: String - YYYY

Flag

A flag is an indicator having a range of only two values that indicates that some event has happened or should happen, yes or no, true or false, off or on, etc. If additional values are required, such as "Unknown" and/or "Null", then a code must be used.

Flag Standards

Flags must:

- Have names that identify the "on" or "yes" position,
- Have Valid Values defined in the data dictionary.

Example: Dictionary Valid Value Entry for Flag

PROJECT APPROVAL FLAG	Definition: … Purpose: … Examples: N/A Valid Values: Y = Project approved, N = Project Not Approved Format: …

Name

A name is a word or phrase that provides a distinctive designation for an object, quality, or concept. Some examples are STATE NAME, COLOR NAME, EMPLOYEE FIRST NAME. Codes are usually described with names. For example, EMPLOYMENT STATUS CODE such as "X", "1", or "2", is spelled out or described with EMPLOYMENT STATUS NAME such as "Former Employee", "Regular Full-Time", or "Regular Part-Time".

Examples: Dictionary Example Entries for Name

STATE NAME	Definition: … Purpose: … Examples: Texas, Oklahoma Valid Values: N/A Format: …
EMPLOYMENT STATUS CODE	Definition: … Purpose: … Examples: X = Former Employee, 1 = Regular Full-Time, 2 = Regular Part-Time Valid Values: N/A Format: …
EMPLOYMENT STATUS NAME	Definition: … Purpose: … Examples: Former Employee, Regular Full-Time, Regular Part-Time Valid Values: N/A Format: …

Description

A description provides additional defining information about an object.

Example: Dictionary Example Entry for Description

VEHICLE TYPE DESCRIPTION	Definition: … Purpose: … Example: A light truck is any truck that weighs less than 10,000 pounds Valid Values: N/A Format: …

Comment

A comment is a note added to provide explanatory information. It is usually provided by the user to make a special note or remarks about the object.

Example: Dictionary Example Entry for Comment

Logical Data Name	Description
CONTRACT COMMENT	Definition: … Purpose: … Example: Contract renewal in process, pending approval. Valid Values: N/A Format: …

Text

Occasionally, there is a need for content chiefly in the form of words or a string of characters. This content does not name, describe, or comment. Typically, this content contains the "words of something," or a string of characters as they must appear. In the example below, SIGN TEXT contains the words as they appear on the sign.

Example: Dictionary Example Entry for Text

Logical Data Name	Description
SIGN TEXT	Definition: … Purpose: … Example: DANGEROUS CURVE AHEAD Valid Values: N/A Format: …

Section 5

Specific Naming Standard

Overview

Some attributes are common to multiple entities and are used more often in the physical implementation of a database.

The following table displays conventional uses of frequently occurring column names or extensions.

Class Word Qualifiers

Class Word Qualifier(s)	Class Word	Physical Name/Extension
CREATE	DATE	CREATE_DT
UPDATE	DATE	UPDT_DT
CREATE USER	IDENTIFIER	CREATE_USER_ID
UPDATE USER	IDENTIFIER	UPDT_USER_ID

This page intentionally left blank

Chapter 4
Abbreviations

Contents:

This page intentionally left blank.

Section 1

Use of Abbreviations

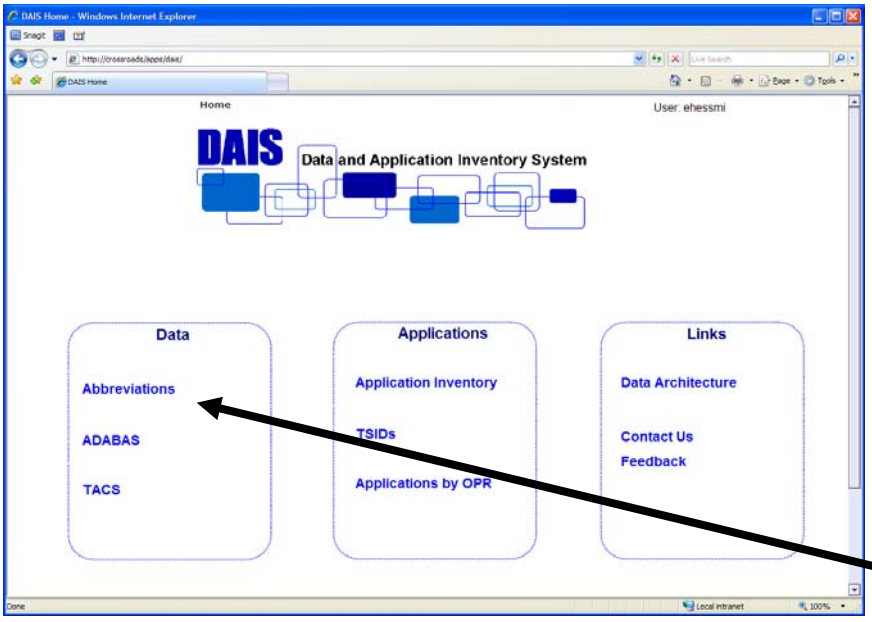
Overview

The standard abbreviations are to be used for physical data names or for reports where abbreviations are needed. Abbreviations generally are not used in logical (business) names, but there are exceptions:

- When an acronym, abbreviation or initialism is normally used by the business user in discussion or memos, then that acronym, abbreviation or initialism may be used (e.g., *FTE*).
- When the logical name would be excessively long if not abbreviated, the words that most commonly appear in an abbreviated form are acceptable. For example, "*YTD*" would be an acceptable abbreviation for "*YEAR-TO-DATE*" in a long logical name.
- When an abbreviation is created by and normally used by an organization outside of TxDOT.

Standard Abbreviations

A list of standard abbreviations can be accessed using the "Abbreviations" button on the home page of DAIS at http://crossroads/apps/dais/



Abbreviations Button

If additional abbreviations are needed, contact Data and Quality Management Services.

This page intentionally left blank.

Chapter 5
GIS Data Standards

Contents:

This page intentionally left blank.

Section 1

Introduction to GIS

Geographic Information Systems

A Geographic Information System (GIS) is a grouping of software and data, designed to manipulate, analyze, and present information that is tied to a spatial (i.e., geographic) location.

TxDOT has many physical assets distributed throughout the state, making GIS a useful tool. Integrating information about TxDOT assets into a GIS adds a new dimension to inventorying, analyzing and presenting information about those assets. TxDOT shares a large amount of information internally and externally. TxDOT receives a significant amount of information from external sources. Combining shared and external data with spatial or geographic information makes new types of complex analysis possible.

A GIS also allows for the display of physical assets information on a map. Currently, TxDOT takes data from internal and external sources and combines the data, using spatial information, to produce the Official Travel Map for the State of Texas.

As existing data structures are modified, or when new data is gathered, if there is a geographic aspect to the data, then the GIS data standards should be followed.

Scope of the GIS Data Architecture

The focus of the GIS Data Architecture is to provide TxDOT with an enterprise-wide blueprint for GIS data design. For software project development teams working with existing databases or developing new databases with GIS components, the GIS Data Architecture is essential.

Topics addressed in this chapter include:

- GIS data standards
- GIS data naming standards
- Exceptions to the GIS data standards.

Benefits of GIS Data Architecture

If the data you are manipulating uses different types of spatial measurement units, coordinate system, or different levels of accuracy, combining the data for analysis or display will be more difficult. Each time the data is updated or refreshed, the conversion process will be required again.

Following the GIS data architecture will simplify:

- Spatially combining and analyzing the data
- Sharing spatially enabled data
- Displaying the information on a map.

Applying the GIS data architecture will produce benefits to TxDOT. Non-compliance, in favor of quick implementation, will almost certainly result in eventual increased costs when modifications to the units of spatial measurement are needed.

Relationship to TxDOT Data Standards

The GIS data standards were written to be a specialized subset of the general TxDOT data standards and guidelines. GIS data must also adhere to the general data standards found in Chapter 2 (Data Standards) of the TxDOT Data Architecture.

Section 2

GIS Data Standards

Overview

As with general data modeling, there are logical and physical components of the GIS data standards. The logical GIS data standards are specific and straight forward. Because of special conditions and/or limitations of specific data storage platforms, the physical GIS data standards are more complex. Physical GIS data standards are outlined for non-relational and relational occurrences.

Logical GIS Data Standards

Locations on the earth's surface are represented by the geographic coordinates; latitude, longitude, and elevation. The linking of geographic coordinates to data makes the data spatial.

Logical data models that contain geographic coordinate data will follow general data standards. In addition, the logical data model must contain the following attributes:

- Latitude (Decimal Degrees)
- Longitude (Decimal Degrees)
- Elevation (U. S. Survey Feet).

Latitude and longitude will be stored in Decimal Degrees. If latitude and longitude are needed in degrees, minutes, and seconds, then data will be programmatically converted.

Elevation will be stored in U. S. Survey Feet. Elevation may not be collected in all cases. However, it should be modeled for potential future use.

GIS Physical Data Standards for Relational Databases

The following are physical relational database implementation standards for GIS coordinate data:

- Latitude and longitude must:
  - Be stored as signed numeric
  - Have 3 places to the left of the decimal
  - Have 8 places to the right of the decimal.

NOTE: If data is not available, value must be null. If a "zero" value is entered the asset will appear off the coast of Africa.

(continued...)

GIS Physical Data Standards for Relational Databases *(continued)*

- Elevation must:
  - Be stored as signed numeric
  - Have 5 places to the left of the decimal
  - Have 3 places to the right of the decimal.

NOTE: Elevation may be null for missing values.

CAUTION*:* Do not use zero to mean that data is missing. A "zero" entered for elevation will mean a value of "zero" for elevation, or "sea level".

GIS Physical Data Standards for Non-Relational Data

The numeric data types available on the mainframe for ADABAS, VSAM, and sequential files do not provide the accuracy or precision needed for spatial data. Special handling must occur for spatial data to meet the accuracy requirement to be merged with other TxDOT spatial data.

For example, ADABAS files store floating point data with eight significant digits to the right of the decimal point. But, the eighth place will be corrupted. Consequently, this misleads GIS data users to assume accurate data is available when it is not. The standard text format solves this problem by making accurate data available in all digits.

The following are physical non-relational database implementation standards for GIS coordinate data:

Latitude and longitude must:

- Be stored as text strings of thirteen characters with a format of ±123.12345678
- Include a sign (+ or -)
- Include the decimal point.

NOTE: The sign (+ or -) and the decimal point cannot be assumed. If data is not available, value must be null. If a "zero" value is entered, the asset will appear off the coast of Africa, where 0 latitude and longitude are located.

(continued...)

GIS Physical Data Standards for Non-Relational Data *(continued)*

Elevation must:

- Be stored as a text string of nine characters with a format of ±12345.123
- Include a sign (+ or -)
- Include the decimal point.

NOTE: The sign (+ or -) and the decimal point cannot be assumed. Elevation may be null for missing values.

CAUTION: Do not use zero to mean that data is missing. A “zero” entered for elevation will mean a value of “zero” for elevation, or “sea level”.

This page intentionally left blank.

Section 3

GIS Data Naming Standards

Overview

General data naming standards must be followed when naming GIS data design objects. In addition to the general standards, there are some data naming standards specific to GIS. This section contains data naming standards specific to:

- GIS entity/table (file) names
- GIS attribute/column (field) names
- GIS class words.

GIS Entity/Table (File) Naming Standards

For data to be displayed on a map, it must be converted from data to geometric elements, such as points or lines. In GIS, these geometric elements are categorized into three general data types; point, line, and polygon.

An example of a point data type would be “City Point”. “City Point” might be used to display cities on a statewide map of Texas. An example of a polygon data type would be “City Polygon”. “City Polygon” might be used to display the area encompassed by a city on a county map. In GIS data naming, some assets may appear as more than one data type. In order for a user of GIS data to differentiate between the GIS data types, a qualifier word must be added to the end of the entity name (logical) and table name (physical).

If the entity, table, or file contains geometric information then the following GIS entity/table (file) data naming standards must be followed:

- Entity name must end with “POINT”, “LINE” or “POLYGON”
- Table (file) name must end with “PNT”, “LN” or “POLY”.

GIS Entity/Table (File) Naming Examples

Examples: Table Data Naming

Logical (Business) Name	Physical Name
REST AREA POINT	REST_AREA_PNT
ROUTE SYSTEM LINE	RTE_SYS_LN
STATE PARK POINT	STATE_PARK_PNT
STATE PARK POLYGON	STATE_PARK_POLY

GIS Attribute/Column (Field) Naming Standards

General TxDOT data naming standards must be followed in naming GIS attributes, columns, or fields. If the attribute, column or field contains geometric information then the following GIS attribute/column (field) data naming standards must be followed:

- Attribute names must end with the appropriate class word, "ELEVATION", "LATITUDE" or "LONGITUDE"
- Column (field) names must end with "EL", "LAT" or "LON".

Attribute and Column Naming Examples

Examples: Data Naming

Logical (Business) Name	Physical Name
REST AREA LATITUDE	REST_AREA_LAT
STATE PARK LONGITUDE	STATE_PARK_LON
RECEIVER BASE ELEVATION	RCVR_BASE_EL

GIS Data Dictionary Standards

GIS data dictionary entries must follow general data dictionary standards. Data dictionary standards can be found in Chapter 2 of the *Data Architecture*.

GIS Class Words

Class words are a portion of the logical name of an attribute. A class word is the highest level of qualification in an attribute name. The class word is always the last word of the attribute name. The class word must indicate the type of attribute being named. The following table lists GIS class words.

NOTE: For a more detailed explanation of class words and how they are used in naming data design objects, see Chapter 3 of the *Data Architecture*.

Class Word List

The following table contains the GIS-specific class words and their standard abbreviations.

GIS Class Word List

Class Word	Abbreviation	Definition/Example/Format
ELEVATION	EL	**Definition**: An *ELEVATION* is the vertical distance of a point or object above or below a reference surface such as a vertical datum (generally mean sea level) measured in U. S. Survey Feet. **Purpose**: Elevation, latitude, and longitude can be used in combination to locate any point on the earth's surface in three dimensions. **Example**: +10297.556 or -45.265 **Valid Values**: N/A **Format**: Number Elevation is stored as numeric with 5 places to the left and 3 places to the right of the decimal point and as a positive or negative value.
LATITUDE	LAT	**Definition**: LATITUDE is the angular distance along a meridian north or south of the equator, usually measured in degrees, at which the entity is located. Lines of latitude are parallel to the equator. Lines of latitude are also called parallels. **Purpose**: The intersection of latitude and longitude can be used in combination to locate any point on the earth's surface. **Example**: +30.31166667 **Valid Values**: N/A **Format**: Number Latitude is stored in Decimal Degrees (DD) with 3 places to the left and 8 places to the right of the decimal point and as a positive or negative value. (In Texas this value is always positive.)
LONGITUDE	LON	**Definition**: A *LONGITUDE* is the angular distance, expressed in degrees, of a point on the earth's surface east or west of a prime meridian (usually the Greenwich meridian). All lines of longitude are great circles that intersect the equator and pass through the north and south poles. **Purpose**: The intersection of latitude and longitude can be used in combination to locate any point on the earth's surface. **Example**: -97.75611111 **Valid Values**: N/A **Format**: Number Longitude is stored in Decimal Degrees (DD) with 3 places to the left and 8 places to the right of the decimal point and as a positive or negative value. (In Texas this value is always negative.)

This page intentionally left blank.

Section 4

Exceptions to the GIS Data Standards

Overview

Exceptions to the GIS Standards occur in two areas:

- Software constraints
- Data from external sources.

Software Constraints

The standard software tools used in GIS at TxDOT are a series of Environmental Systems Research Institute (ESRI) tools. This series of tools include a spatial database engine, ArcSDE. ArcSDE uses certain table and column names to function. Using ArcSDE requires the following exceptions to the TxDOT Data Naming Standards:

Installation

Database tables and columns created by ArcSDE at the time of its installation must not be renamed or deleted.

Registration

Database tables and columns are created by ArcSDE at the time of spatial registration. Database columns OBJECTID and SHAPE must not be renamed or deleted.

Data from External Sources

Frequently, other governmental agencies collect and maintain data about Texas that may be of benefit to TxDOT. For instance, there are many features routinely placed on TxDOT maps that are the result of data collection by other agencies.

Features such as lakes, streams, public lands, and city limits are collected and maintained by Texas Natural Resources Information System, the Texas Parks and Wildlife Department and the United States Geological Survey. To aid in the transfer of information between agencies, an exception to the TxDOT Data Standards may be needed.

Using data from other sources requires the following exception to the TxDOT Data Naming Standards: Data obtained from other governmental agencies should not be modified, restructured, or renamed.

The use of external data should be closely coordinated with the Surveying and GIS Support Branch of the Information Systems Division.

This page intentionally left blank.

Chapter 6
Commercial Off-the-Shelf Software

Contents:

This page intentionally left blank.

Section 1

Integrating Commercial Off-the-Shelf Software with TxDOT Data

Overview

This chapter explains the data needs/requirements to be considered when integrating a Commercial Off-the-Shelf (COTS) package with TxDOT Data.

Note: For assistance in choosing and implementing a COTS package, contact the Infrastructure Services Section of TSD.

What is a COTS Package?

A COTS product is one that is intended to be used "as-is". COTS products are designed to be easily installed and to interoperate with existing system components. Almost all software bought by the average computer user fits into the COTS category: operating systems, office product suites, word processing, and e-mail programs are some examples. In addition to the commonly used COTS products, there are other COTS products developed for specific markets, such as PeopleSoft or Cognos.

COTS packages are an option available to address specific business needs. Many of the COTS packages available provide impressive user interfaces and reporting capabilities. Despite appearances, careful analysis must be completed to ensure that the COTS packages fit TxDOT's data and data needs.

COTS Packages at TxDOT

TxDOT currently uses many COTS packages. Some of the COTS packages are for general use in the workplace, such as GroupWise, Word, Excel, etc. Other of the COTS packages that TxDOT uses are for specific business functions, such as PeopleSoft for human resources, Cognos for budget, and SiteManager for construction. These COTS packages may either access data from TxDOT databases or maintain TxDOT data in their internal databases.

The following sections present guidelines from a data perspective to assist in the selection and implementation process for COTS packages that will use or contain TxDOT data.

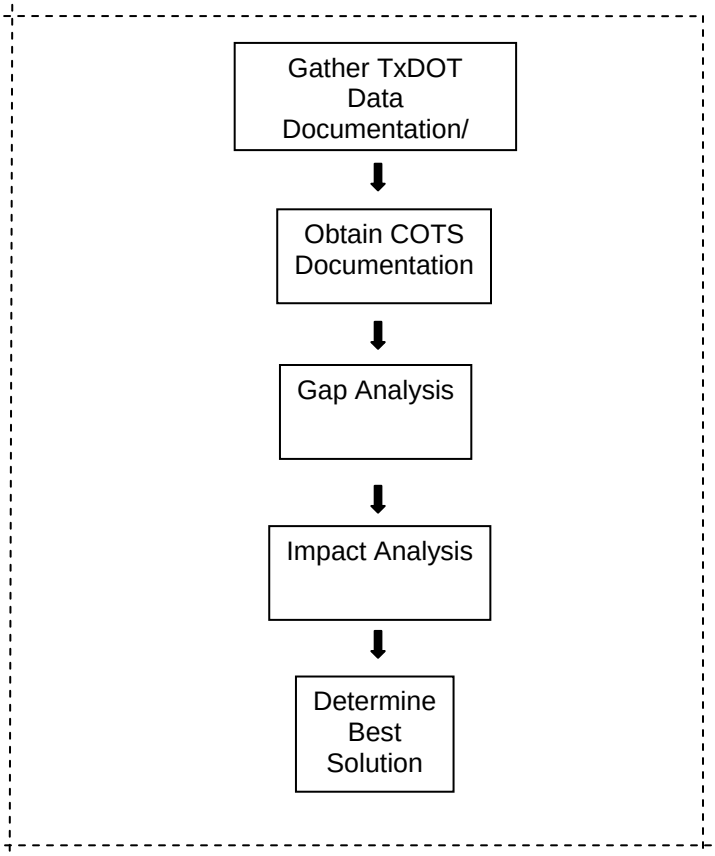
TxDOT Data and the COTS Selection Process

In order to appeal to the most customers, COTS packages must provide common functionality with generic data requirements. When software and databases already exist to support a business area, introducing a COTS package requires integration and additional effort.

COTS Data Integration Analysis Process

When considering COTS packages for selection, following a data integration analysis process is essential to selecting a package appropriate to the data needs.

Data Integration Analysis Process



Gather TxDOT Data Documentation and Requirements
Use existing logical data models and data dictionaries. If a logical data model and data dictionary do not exist contact Data and Quality Management Services branch for assistance in reverse engineering existing databases to obtain information. Use a glossary and conceptual data model as described in Chapter 2 to document new or additional data requirements.

Obtain COTS Documentation
A logical data model and a data dictionary are preferred. If they are not available, determine what data requirement documentation is available.
NOTE: The documentation that is gained at this point is the documentation that will be used for the life of the system.

Gap Analysis
Completing a Gap Analysis (determining the differences between TxDOT data requirements and a COTS package or the differences between existing TxDOT

databases and a COTS package) is also an essential element to choosing and successfully implementing the best COTS solution. COTS packages will rarely, if ever, meet all TxDOT data requirements or match existing TxDOT databases. Therefore, it is very important to analyze the gap. Understanding the gap will help to make decisions on which COTS package to choose and how to address the gap once a package is chosen. The gap may be addressed by changing existing TxDOT data requirements to more closely match the COTS package which would result in reduced development costs. Or, the gap can sometimes be addressed by adding to (customizing) the COTS package to meet additional TxDOT data needs.

Impact Analysis

Obtain a copy of the TxDOT System Interface Diagram (TSID, see Chapter 1) to identify existing relationships and determine the impact of integrating the COTS package. Using the TSID to complete a thorough Impact Analysis is essential to successfully implementing a COTS solution. If other systems are dependent on existing software and/or databases, moving to a COTS package solution could disable or break other systems. To avoid disrupting other systems all dependencies must be identified and mitigated before a COTS package can be implemented.

Determine Best Solution

Use the Impact Analysis and the Gap Analysis as part of the criteria to determine if the COTS package is appropriate for TxDOT

COTS Package Data Customization Standards

When the Gap or Impact Analysis shows a difference between TxDOT data requirements and the COTS package requirements and the solution for addressing the gap or impact is to add to or customize the COTS package, the following standards must be followed:

- Begin any new data components with a prefix of “TX_”.
  This will easily identify customization components when upgrades occur.
- Follow the “TX_” prefix with a name that follows the COTS package naming standard. If the COTS package does not follow a naming standard, use the TxDOT naming standards outlined in Chapter 2.
- Document the new data components in the COTS package data dictionary.
  If a COTS data dictionary does not exist, create a customization component data dictionary using the TxDOT dictionary standards outlined in Chapter 2.

This page intentionally left blank.

Appendix A
Data Architecture Glossary

A

Abbreviation
An abbreviation is a term created by shortening longer terms. An abbreviation is usually pronounced as the original word.

Acronym
An acronym is a term created by using the initial letters of the several component words composing the name of any organization, program, or concept to create a new ***word*** that substitutes for the combination of the component words. Acronyms are pronounced as words.

Alias
An alias is an approved alternate name for something, to be used when conditions make it impracticable to use the official name.

Alternate Key
An alternate key is an attribute (or attributes) of an entity other than the primary key that uniquely identifies an instance of that entity. An alternate key must hold a unique value for each record or row.

Architecture
Architecture is the manner in which the components of a computer or computer system are organized and integrated.

ArcSDE
ArcSDE is a spatial database engine from the Environmental Systems Research Institute (ESRI) geographical information system (GIS) tools.

Attribute
An attribute is a property, characteristic, or relationship that describes an entity. It is a fact that has the same format, interpretation, and domain for all occurrences of an entity.

B

Business Rule
A business rule is a rule, standard, policy, or procedure, formal or informal, within which the business must operate. Business rules may be depicted graphically as relationships on entity relationship diagrams, or written in textual descriptions.

C

Cardinality
Cardinality is the definition of a relationship that determines the number of entities on one side of the relationship that may be joined to a single entity on the other side.

Class Word
A class word is a word within an attribute (element) name that defines the category of data used at TxDOT.

Column
A column is the physical implementation of an attribute in a table in a relational database.

Conceptual Data Model
A conceptual data model is a representation of data as seen from a business viewpoint. It identifies persons, things, concepts and events.

D

Data
Data is the representation of facts. It is the raw material of information and is used as a basis for reasoning, decision-making, discussion, or calculation.

Data Architecture
Data Architecture is the manner in which data components are organized and integrated.

Data Dictionary
A data dictionary is a directory of the definitions, purpose, valid values, examples, and formats about data. It describes every data item in a database in enough detail for users and application developers to know what the data is and how to make use of it.

Data Model
A data model is a graphic representation of entities, with descriptions, and the relationships among them, often accompanied by a description of the attributes of the entities.

(continued...)

D *(continued)*

Database
A database is a physical collection of data organized for flexibility of access, usually under the control of a Database Management System (DBMS). Also, a database may be the entire collection of data and/or databases for an enterprise or part or an enterprise.

Database Management System (DBMS)
A Database Management System (DBMS) is a set of software that controls the organization, storage, retrieval, security and integrity of data in a database. It accepts requests from the application and instructs the operating system to transfer the appropriate data.

Data Integrity
Data integrity is a measure of the quality of information. The integrity of data is measured in terms of the following characteristics: accuracy, reliability, timeliness, conciseness, non-redundancy, and consistency.

Decimal Degrees
Decimal Degrees (DD) is a unit of measure for latitude and longitude expressed in degrees instead of degrees, minutes, and seconds. Decimal Degrees are computed with the formula: DD = degrees + minutes/60 + seconds/3,600.

Denormalization
Denormalization is the process of modifying a normalized database design for performance reasons. Denormalization makes the system less flexible, and should be done only when needed.

E

Elevation
Elevation is the vertical distance of a point or object above or below a reference surface such as a vertical datum (generally mean sea level).

Entity
An entity is a person, place, thing, event, concept or other object of interest that the business must manage and for which data is stored.

Entity Relationship Diagram (ERD)
An Entity Relationship Diagram (ERD) is a graphic representation of the entities and the relationships among them that are relevant to a specific business function.

Entity Subtype
An entity subtype is a subdivision of an entity to which unique rules apply.

F

Field
A field is a physical unit of data in a non-relational database. A collection of fields make up a record.

File
A file is the mainframe term for a collection of data records.

First Normal Form
First normal form is a relation in which the intersection of each row and column contains one and only one value. In first normal form (1nf), all entities have a key composed of one or more attributes that uniquely identify one occurrence of the entity; and for any single occurrence of an entity, each attribute must have only one value - no repeating groups. That is, each field in a file or each column in a table contains different information. See “Normalization.”

Foreign Key
A Foreign Key (FK) is an attribute, or a group of attributes, that is the Primary Key (PK) of some entity other than the entity within which it resides. A Foreign Key is a key attribute (field or column) that identifies instances (records or rows) in a different entity (file or table).

G

Geographic Information System (GIS)
A Geographic Information System (GIS) is a collection of computer hardware, software, and geographic data for capturing, storing, updating, manipulating, analyzing, and displaying all forms of geographically referenced information.

Glossary
A glossary is a collection of terms with their meanings, used to identify any process, event, place, or thing of importance to TxDOT.

I

Initialism
An initialism is a term created by using the initial letters of several component words composing the name of an organization, program, or concept to create a new ***term*** that substitutes for the combination of the component words. Initialisms are pronounced by spelling out the letters.

K

Key
A key is an attribute, or a group of attributes, that uniquely identifies an occurrence (a record) of an entity. A key is an attribute (field or column) that is used to sort data.

L

Logical Data Model
A logical data model is a data dictionary and an Entity Relationship Diagram (ERD) that formalize the data requirements analysis process and identify all business data and relationships within the scope of the project.

Latitude
Latitude is the angular distance along a meridian north or south of the equator, usually measured in degrees. Lines of latitude are also called parallels.

Longitude
Longitude is the angular distance, measured in degrees, of a point on the earth's surface east or west of a prime meridian (usually the Greenwich meridian). All lines of longitude are great circles that intersect the equator and pass through the north and south poles.

N

Natural Key
A natural key is a data column or set of data columns that uniquely identifies a record.

Normalization
Normalization is a process in relational data theory used to reduce redundancy and prevent invalid combinations of data.

P

Physical Data Model
A physical data model is a data dictionary and a diagram (physical data schema) that represent how data is actually structured and stored in the specific Database Management System (DBMS).

(continued…)

P (*continued*)

Physical Data Schema
A physical data schema is a graphical representation of how data is actually structured and stored in a specific Database Management System (DBMS).

Primary Key
A Primary Key (PK) is an attribute (or attributes) of an entity that uniquely identifies an instance of that entity. A Primary Key must hold a unique value for each record or row.

Prime Word
A prime word is a component of an attribute name that identifies the entity that the attribute describes.

Q

Qualifier Word
A qualifier word is a descriptive word that further defines and/or distinguishes those things labeled with class and prime words. Qualifier words are also known as modifier words.

R

Referential Integrity
Referential Integrity (RI) is a feature provided by Relational Database Management Systems (RDBMS) that prevents users or applications from entering inconsistent data.

Relational Database Management System
A Relational Database Management System (RDBMS) is a Database Management System (DBMS) designed to support a relational database. A relational database is a collection of data that is represented as tables that contain only one type of data and are independent of each other.

Relationship
A relationship is a named connection or association between entities. It describes the business rules between the two entities.

S

Second Normal Form
Second normal form is a relation that is in first normal form and every non-primary-key attribute is fully functionally dependent on the primary key. In second normal form (2nf), each non-key attribute within an entity must depend on the key to the entity and all parts of the key. That is, no field (column) values can be derived from another field (column). See “Normalization.”

Spatial Data
Spatial data is the information about the locations and shapes of geographic features, and the relationships between them, usually stored as coordinates and topology.

Standard
A standard is an established definite rule, principle, or measure.

Subject Area
A subject area is a data category that is used to group the data.

Surrogate Key
A surrogate key is a system generated number used as a Primary Key (PK). That is, a surrogate key is a number that uniquely identifies a record in the database, but does not have meaning for the user.

T

Table
A table in a relational database is a data structure that contains data and is represented as rows and columns of data.

Third Normal Form (3nf)
Third normal form is a relation that is in first and second normal form in which no non-primary-key attribute is transitively dependent on the primary key. In third normal form (3nf), each non-key attribute within an entity must not depend on any other non-key attribute within the same entity. Data in 3nf must also be in first normal form (1nf) and second normal form (2nf). That is, no duplicate information is permitted. See
“Normalization.”

This page intentionally left blank.

Appendix B
Data Normalization

Introduction

By the time construction begins on a new section of roadway, a lot of effort has gone into planning and design. Construction could not begin without knowing where the segment is supposed to begin and end, having proper clearances, and making sure that the design meets appropriate standards. Because of the rigor of the planning and design process the roadway will stand up to many years of use and provide a safe travel way for everyone.

Many of these same principles apply when planning a database. A lot of time and money is spent in the creation of a database. Effort must be made to make the database both flexible and robust enough to be useful for as long as practical. One way to ensure flexibility and longevity in a database is to create a relational database using a *normalized* design.

Relational data theory was developed in the 1960s. Two objectives of relational design are to reduce redundancy and prevent invalid combinations of data. The process for accomplishing these objectives is called normalization.

Normalized relational designs tend to have more tables than denormalized designs. This gives the appearance of more complexity. Denormalized designs give the illusion of simplicity but mask any complex data relationships and require maximum design and testing rigor in the programs that maintain the quality of our data. The combination of hidden complexities and skimpy documentation invites data quality problems.

There are several levels of normalization and these are called *normal forms*. Examples of first normal form, second normal form, and third normal form are included in this document.

TxDOT Normalization Standard

The TxDOT data modeling standard is to create logical models in third normal form.

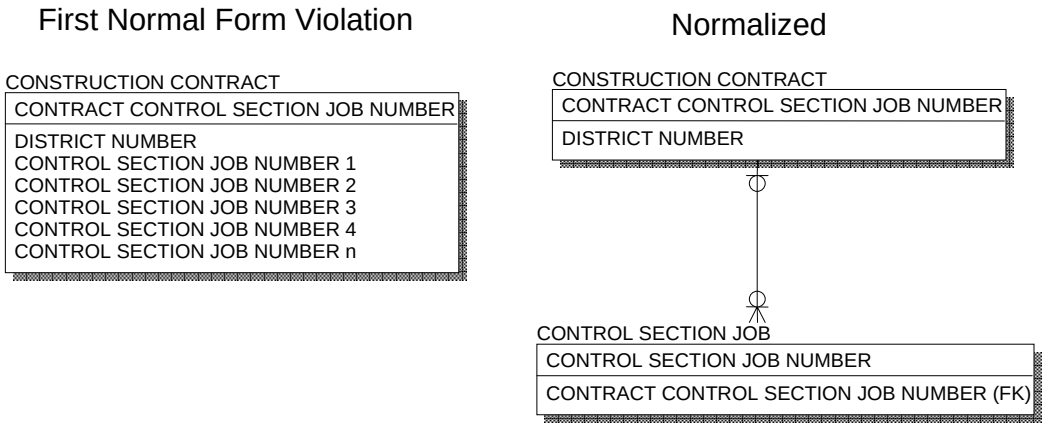
First Normal Form

An *entity* is in first normal form if it contains no repeating groups. A repeating group is one or more non-key *attributes* that can have different values for the same occurrence of the entity.

(continued...)

First Normal Form *(continued)*

In the example below, a CONSTRUCTION CONTRACT can have more than one CONTROL SECTION JOB NUMBER. Therefore, CONTROL SECTION JOB NUMBER is a repeating group.



Creating First Normal Form

The model can be placed in first normal form by decomposing (splitting) the entity into two or more entities. A new entity called CONTROL SECTION JOB will accomplish this. The attribute CONTROL SECTION JOB NUMBER makes up the key to the new entity. There are no longer repeating groups, so the model is in first normal form.

Benefits

There are two benefits to the normalized structure.

First, there is no longer a physical database restriction on the number of CONTROL SECTION JOB NUMBERs per CONTRACT CONTROL SECTION JOB NUMBER. If the business rules change from a maximum of five CONTROL SECTION JOB NUMBERs to a maximum of ten the database would not need any changes.

Second, the new table can be efficiently queried by CONTROL SECTION JOB NUMBER. In the original, denormalized structure, each of the CONTROL SECTION JOB NUMBER columns would have required a separate index and one query per CONTROL SECTION JOB NUMBER column, or a slower physical read with a complex WHERE clause. For tables containing a large number of rows this could be prohibitively expensive.

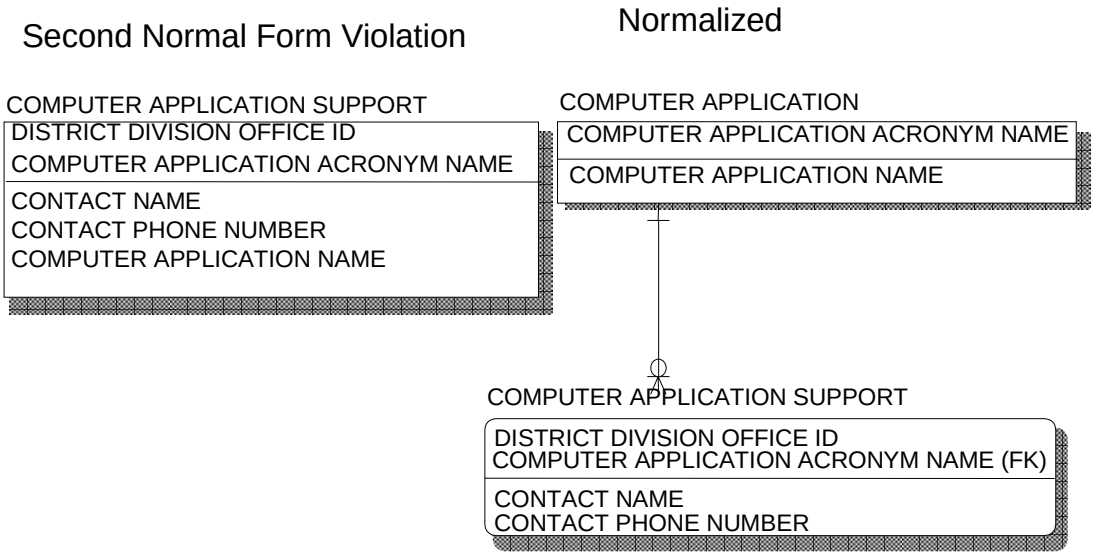
Second Normal Form

An entity is in second normal form, if it is in first normal form and every non-key attribute has a relationship with every attribute in the key. In the example below, COMPUTER

(continued...)

Second Normal Form *(continued)*

APPLICATION NAME has no relationship with DISTRICT DIVISION OFFICE ID. The attribute, COMPUTER APPLICATION NAME only has a relationship with COMPUTER APPLICATION ACRONYM. Therefore, the entity COMPUTER APPLICATION SUPPORT is not in second normal form.



Creating Second Normal Form

To place the model in second normal form, we need to decompose the entity COMPUTER APPLICATION SUPPORT by creating a new entity called COMPUTER APPLICATION. The new entity contains the attributes COMPUTER APPLICATION ACRONYM NAME and COMPUTER APPLICATION NAME.

The attribute COMPUTER APPLICATION ACRONYM NAME appears as a foreign key in COMPUTER APPLICATION SUPPORT. Note the label (FK) in the diagram above.

Since COMPUTER APPLICATION NAME has a relationship with every attribute in the key (COMPUTER APPLICATION ACRONYM NAME), the model is now in second normal form.

Benefits

There are two benefits to second normal form designs.

First, maintenance to COMPUTER APPLICATION NAME is performed in exactly one place, one time. This not only saves effort, it avoids potentially costly mistakes. NOTE: With real data there are often many more attributes than just NAME.

(continued...)

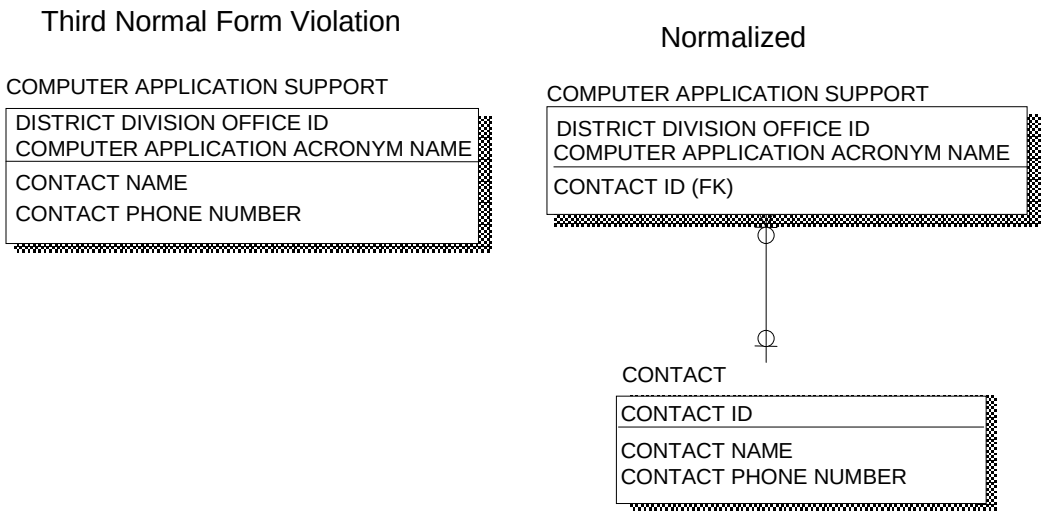
Second Normal Form *(continued)*

Second, a Relational Database Management System can implement this design to provide a pull down list of COMPUTER APPLICATION ACRONYM NAME and disallow any entries that do not exist in COMPUTER APPLICATION. This design prevents invalid data entries when creating a record and during data maintenance.

Third Normal Form

An entity is in third normal form if it is in second normal form and every non-key attribute depends only on the entire key. The definition of third normal form is that each non-key attribute depends on, 'the key, the whole key and nothing but the key.'

If a support responsibility is transferred to someone else, the value of the CONTACT NAME and CONTACT PHONE NUMBER will change, but the DISTRICT DIVISION OFFICE ID and COMPUTER APPLICATION ACRONYM NAME will not. Therefore, the entity, COMPUTER APPLICATION SUPPORT, is not in third normal form.



Creating Third Normal Form

To place the model in third normal form, one must decompose the entity COMPUTER APPLICATION SUPPORT by creating a new entity called CONTACT. The new entity will contain the attributes CONTACT NAME and CONTACT PHONE NUMBER. CONTACT ID is added as a unique identifier of CONTACT. This would allow for two John Smiths to be in the database.

Benefits

The benefits to third normal form are similar to second normal form designs.

Denormalization

Some fully normalized data designs can prove to be slow when implemented as a physical database. Databases with huge quantities of data often require some physical design changes. This may occur when the table size is large or when the record size is large. The physical model may be denormalized and tuned for speed.

Risks

Intentionally altering the data design comes with some risk. Consider the following:

- Denormalized physical models no longer reflect the same business rules as the normalized logical model. Programming logic would have to be added to compensate. This added program logic that preserves data integrity is often omitted or poorly tested. Therefore it is best to denormalize with a careful plan of programming and testing.
- The altered database would "hide" the data integrity rules. Maintaining both logical and physical data models help future designers and programmers remain mindful of data quality issues.

In the past, many programmers and database administrators have assumed that a normalized database cannot be implemented. However, TxDOT has successfully implemented numerous normalized data designs.

As hardware platforms become increasingly powerful and relational database management systems more sophisticated, the case for denormalization is weakening. It is best to demonstrate performance problems, and only then alter the data structure.

Assistance and Consultation Services

The Data and Quality Management Services Branch can assist with consultation on design, design changes, and with the physical implementation.

This page intentionally left blank.

Appendix C
Data Model Interpretation

Introduction

The purpose of a data model is to illustrate the project data requirements, which can be viewed and understood by all project participants (business users, analysts and developers). Data model creation is the first step in setting the scope for the size of a project and the associated database.

A data model consists of a data diagram and a data dictionary. The data diagram is a graphic representation of the data structure of a database. Data models are analogous to plans or blueprints in that they contain structure specifications for a database.

This document will assist the inexperienced data model user by discussing how to interpret:

- Logical data models
- Physical data models

Examples shown follow TxDOT data model design standards found in "Data Standards," Chapter 2 of this document.

Logical Data Model

A logical data model formalizes the data requirements analysis process and identifies all business data and relationships within the scope of the project. A logical data model is independent of the hardware or software requirements. The completed logical data model consists of the following:

- Entity Relationship Diagram (ERD)
- Data dictionary

Entity Relationship Diagram (ERD)

Logical Entity Relationship Diagrams contain entities, attributes, and relationships.

- ***Entities***

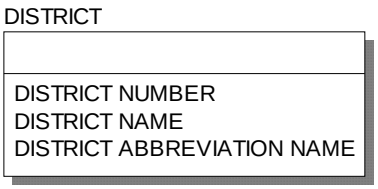
  An entity is represented by a box on a logical data model diagram. It is a person, place, thing, concept or event that the business must manage and for which data is stored. The names of entities are always singular nouns.

(continued...)

Logical Data Model *(continued)*

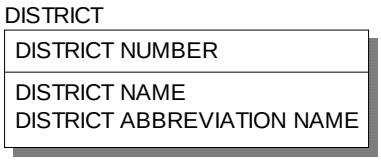
- ***Attributes***

An attribute is a characteristic that describes or identifies an entity. Attributes are placed within the entity box. The last word or class word describes the type or class of data used.



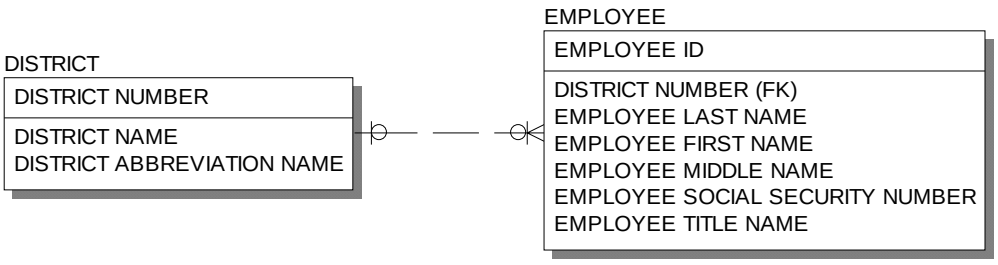
- Primary Keys*:* A Primary Key (PK) is an attribute (or attributes) that uniquely identifies an occurrence of the entity. This unique identifier appears above the line that separates the Primary Key from the other attributes.

  The numbers 1 – 25 are valid values for the attribute DISTRICT NUMBER and uniquely identify occurrences of the entity DISTRICT. The key attribute DISTRICT NUMBER appears above the line inside the entity box for DISTRICT.



- Foreign Keys*:* A Foreign Key (FK) is an attribute of an entity that matches the Primary Key of another entity to which it is related.

  In the following example DISTRICT NUMBER is a Foreign Key in the EMPLOYEE entity. Foreign Keys are identified in the model by placing an "FK" after the attribute name.



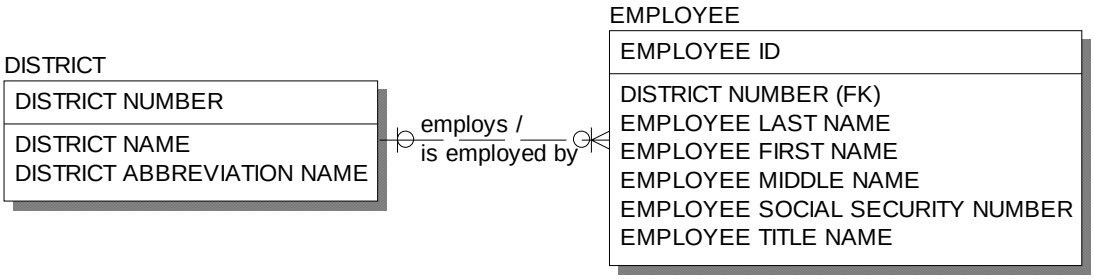
(continued...)

Logical Data Model *(continued)*

- ***Relationships***

Lines that connect two entity boxes together are known as relationships. A relationship is a data-oriented business rule that exists between two entities. These relationships reflect that two entities interact in some way that we want to record. Relationships consist of verb phrases and cardinality.

- *Verb Phrases:* A verb phrase provides meaningful context to a relationship. It reflects in business terms the nature of the association between entities. Each relationship contains two verb phrases, a primary and an inverse.



Primary:
DISTRICT *employs* EMPLOYEE

Inverse:
EMPLOYEE *is employed by* DISTRICT

- Cardinality*:* Cardinality is defined as the minimum and maximum number of occurrences between two related entities. The symbols used at the ends of the relationship line represent cardinality. The following table illustrates and describes what each of the symbols represents.

Symbol	Entity	Description
○	ZERO	This symbol represents that zero is a possible quantity for a particular entity.
|	ONE	This symbol represents that a quantity of one for a particular entity may exist.
⋲	MANY	This symbol represents that a quantity of more than one of a particular entity may exist.

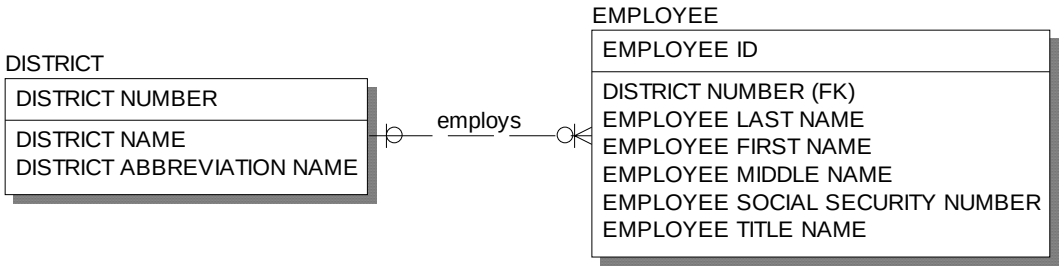
(continued...)

Logical Data Model *(continued)*

The following relationships are read by combining the verb phrases and their cardinality.

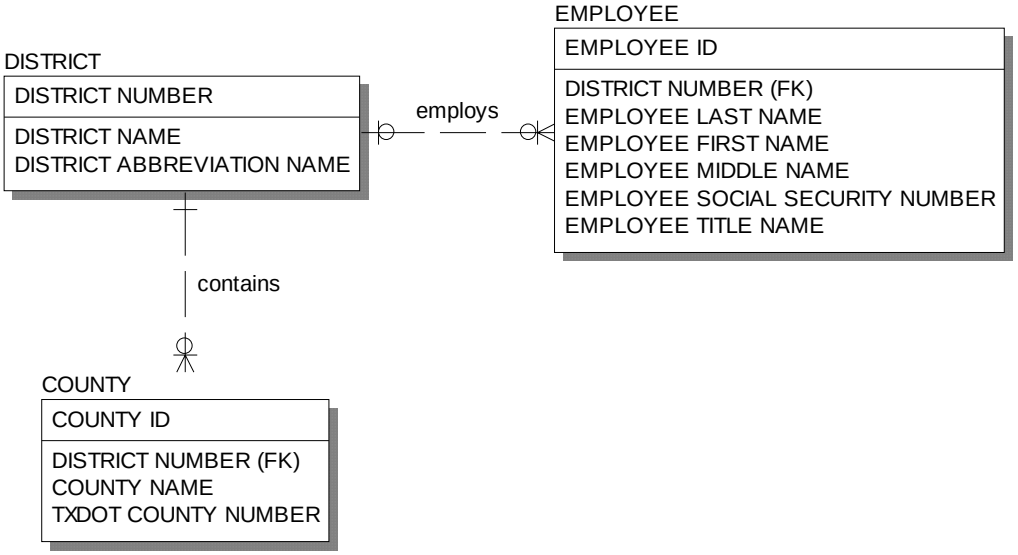
Example: A DISTRICT employs **zero, one, or many** EMPLOYEE(s).
Example: An EMPLOYEE is employed by **zero or one** DISTRICT.

Although an inverse relationship is not typically displayed it verifies cardinality.



- ***Example: Entity Relationship Diagram***

This is an example of a complete Entity Relationship Diagram. This ERD illustrates each of the components discussed above and how they fit together.

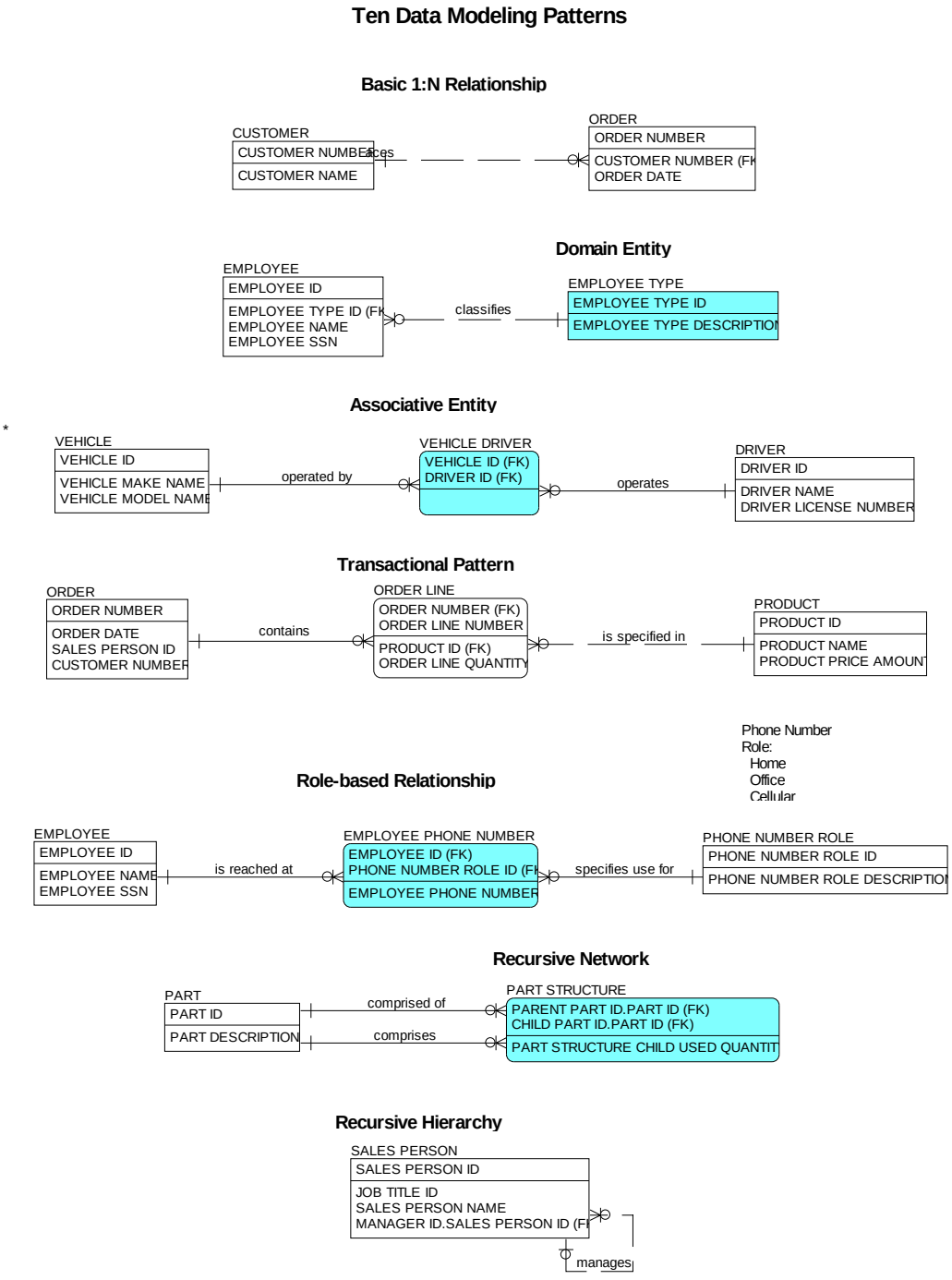


(continued...)

Logical Data Model *(continued)*

- ***Ten Data Modeling Patterns***

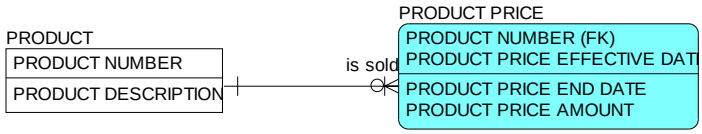
The following are ten common patterns that may appear in data models. (Based on "Ten Data Modeling Patterns", The Inteq Group, Inc., 1995-1997)



(continued...)

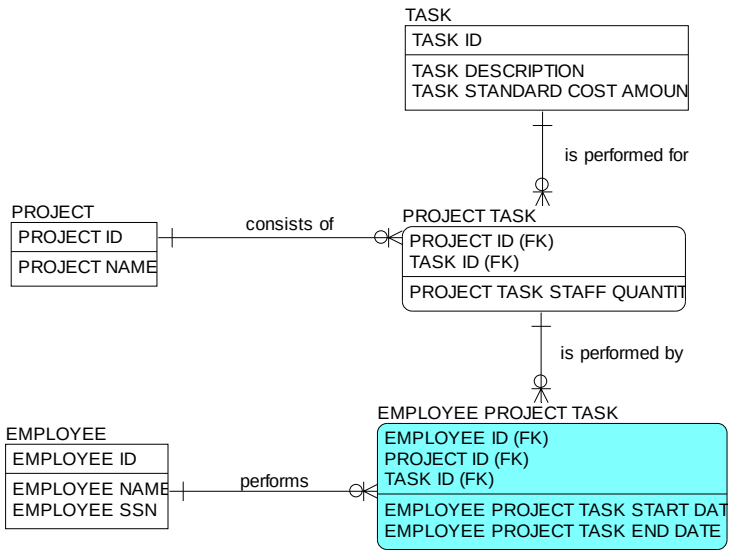
Logical Data Model *(continued)*

Time Dependent Data

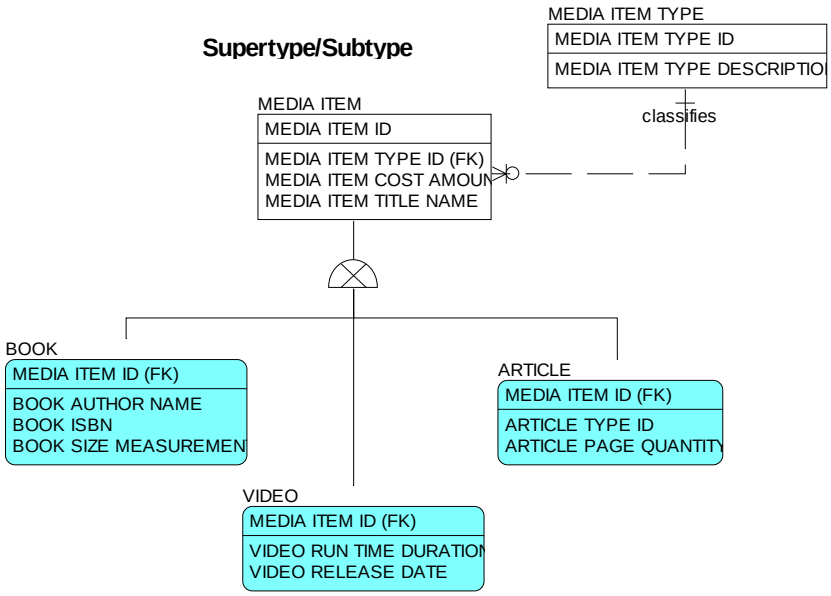


*

Multidimensional Relationship



Supertype/Subtype



(continued...)

Logical Data Model *(continued)*

Data Dictionary

A data dictionary includes a business description of each data object in the data model. A data dictionary is a directory of the definitions, purpose, valid values, examples, and structure about data. It is a compilation of information about the data owned by the enterprise. The data dictionary includes a business description of each data object in the data model.

- ***Entity Description***

  An entity description contains the name of the entity and definition of that entity.

Example: Entity Description

Entity	Description
DISTRICT	**Definition**: A DISTRICT is one of 25 geographical areas within the state of Texas where the department conducts its primary work activities.

- *Attribute Description*

  Each attribute description contains a definition and a purpose. It also contains either an example or valid values. When appropriate, formatting information is shown.

Example: Attribute Description

Business Term	Attribute Description
DISTRICT NUMBER	**Definition**: a **DISTRICT NUMBER** is a distinguishing number that provides a designation for a DISTRICT. **Purpose**: The DISTRICT NUMBER provides a recognizable identifier that uniquely identifies a DISTRICT. **Example**: N/A **Valid Values**: 01 THROUGH 25 Format: Number

Physical Data Model

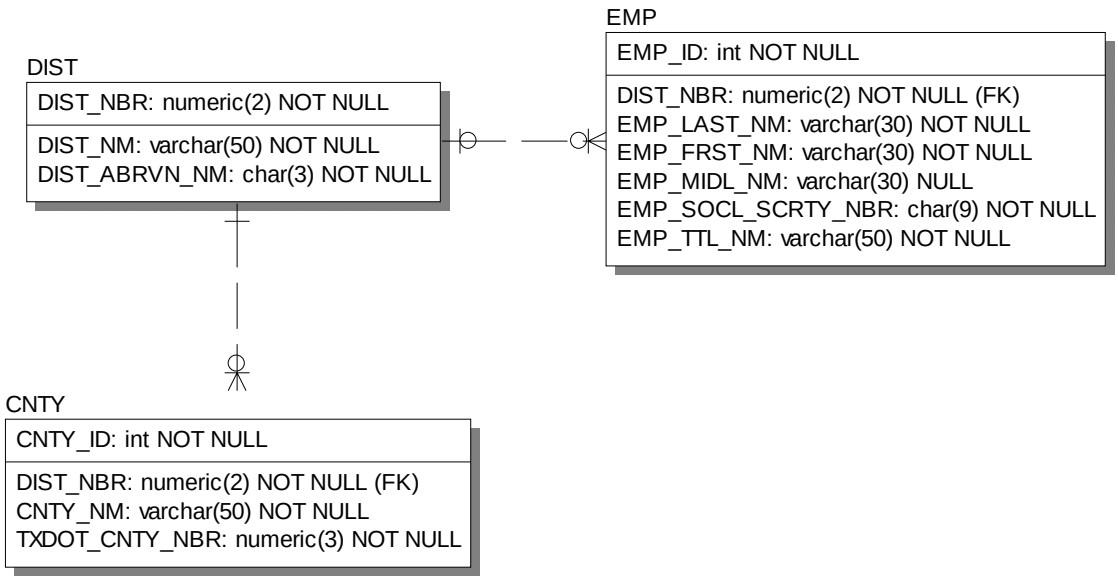
A physical data model is similar to a logical data model with a few noticeable differences. The physical data model contains the following:

- Abbreviated data name(s), physical tables/files, and columns/fields (logical entities and attributes)
- Data format(s) (char, varchar, int, numeric, etc.)

(continued...)

Physical Data Model *(continued)*

- Valid data lengths(s)
- Null/not null option.



Data Dictionary

The data dictionary for the physical data model should match the data dictionary for the logical data model and contain descriptions. Any additional table descriptions will follow the format for the entity descriptions. Any additional column descriptions will follow the format for the attribute descriptions.

Data Model Interpretation Glossary

Attribute
An Attribute is a property, characteristic, or relationship that describes an entity. It is a fact that has the same format, interpretation, and domain for all occurrences of an entity.

Cardinality
Cardinality is the definition of a relationship that determines the number of entities on one side of the relationship that may be joined to a single entity on the other side.

Data Dictionary
A data dictionary is a directory of the definitions, purpose, valid values, examples, and formats about data. It describes every data item in a database in enough detail for users and application developers to know what the data is and how to make use of it.

Data Model
A data model is a graphic representation of entities, with descriptions, and the relationships among them, often accompanied by a description of the attributes of the entities.

Entity
An entity is a person, place, thing, event, concept or other object of interest that the business must manage and for which data is stored.

Foreign Key
A Foreign Key (FK) is an attribute, or a group of attributes, that is the Primary Key of some entity other than the entity within which it resides as a Foreign Key. A Foreign Key is a key attribute (field or column) that identifies instances (records or rows) in a different entity (file or table).

Key
A key is an attribute, or a group of attributes, that uniquely identifies an occurrence (a record) of an entity. A key is an attribute (field or column) that is used to sort data.

Logical Data Model
A logical data model is a data dictionary and an Entity Relationship Diagram (ERD) that formalize the data requirements analysis process and identify all business data and relationships within the scope of the project.

(continued...)

Data Model Interpretation Glossary *(continued)*

Physical Data Model
A physical data model is a data dictionary and a diagram (physical data schema) that represent how data are actually structured and stored in the specific Database Management System (DBMS).

Primary Key
A Primary Key (PK) is an attribute (or attributes) of an entity that uniquely identifies an instance of that entity. A Primary Key must hold a unique value for each record or row.

Relationship
A relationship is the state of being connected or interconnected that describes the business rules between two entities.

Appendix D
TSID Interpretation

Introduction

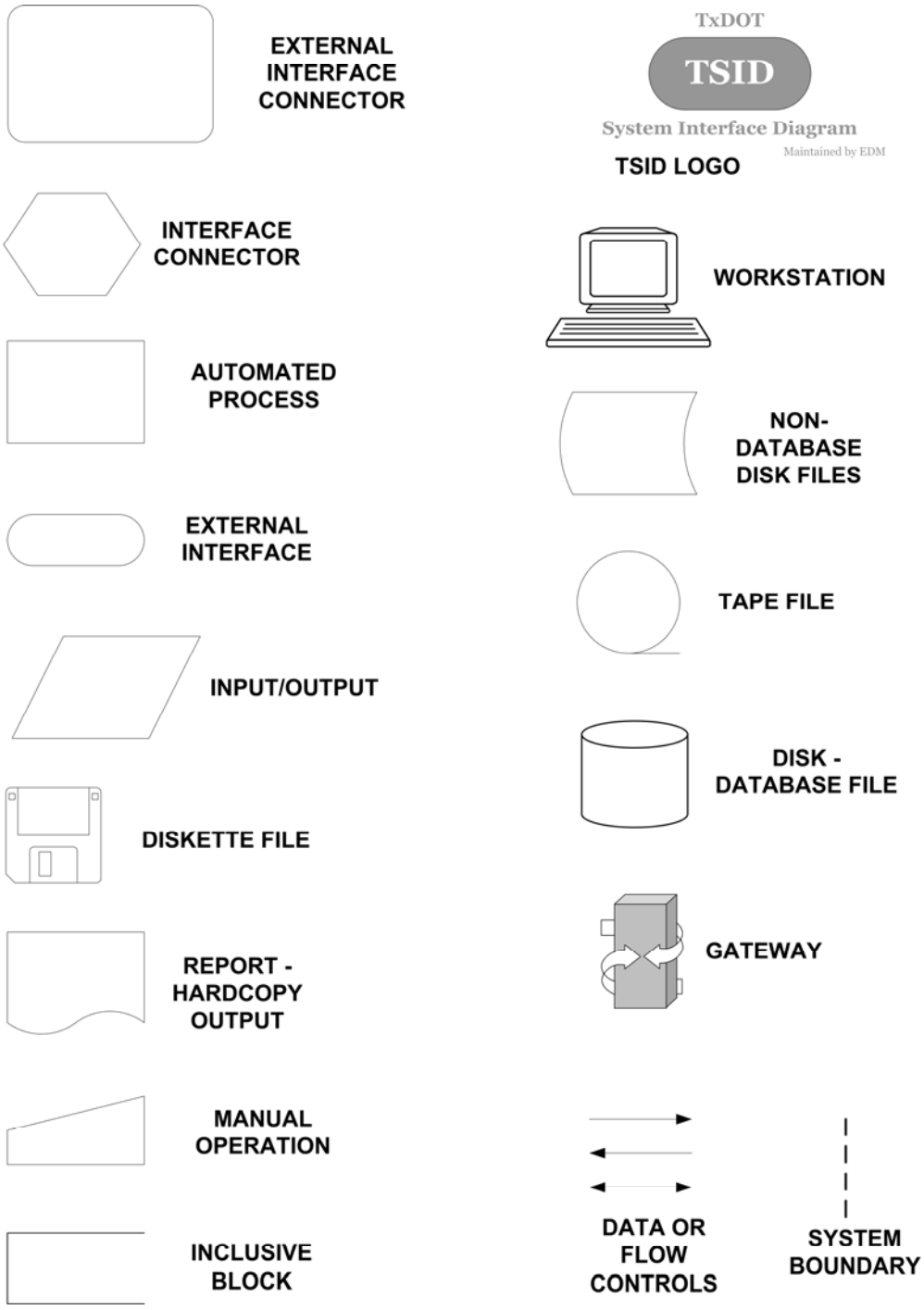
TxDOT System Interface Diagrams (TSID) are a series of graphical diagrams that document the relationships between an application, data created and maintained by the application, and data created and maintained by other interfacing applications. TSID diagrams focus on documenting the major data stores of an application.

The Data and Quality Management Services Branch (DAQ) of TSD maintains TSID diagrams. TSIDs are available through the Data and Application Inventory System (DAIS) within TSD, on the Intranet through the DAIS application, and available on request from the DAQ Branch in hardcopy form. They are designed to be printed on “11x17” paper.

The primary use of a TSID diagram is to aid in the understanding and visualization of dependencies between data and applications. Understanding these dependencies can be very helpful in performing the analysis necessary to revise applications or to add functionality. Large, complex applications may require more than one diagram. For example, the Financial Information Management System (FIMS) has five TSID diagrams. It may be necessary to examine more than one diagram to get a clear understanding of the dependencies of multiple applications.

TSID Legend

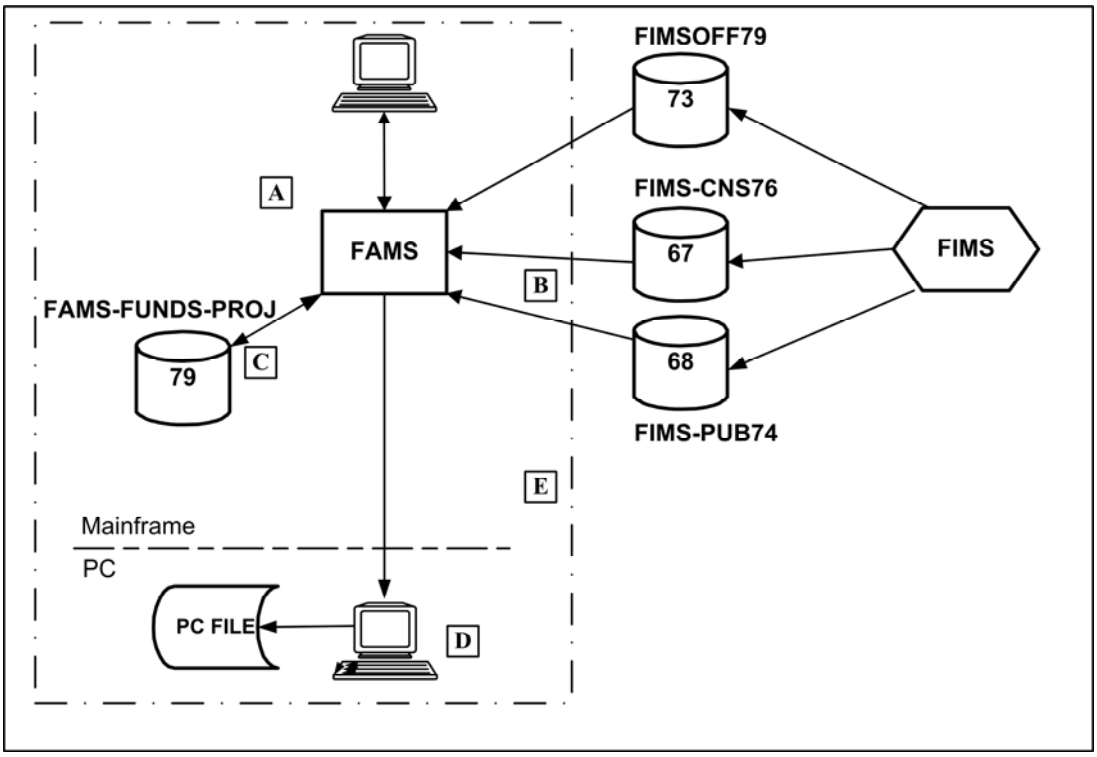
The icons below are used in TSID diagrams. Icons can have labels or names to identify them. The labels or names can appear inside or adjacent to the icons.



How to Interpret Data Flows on TSIDs

The figures below are examples of how to interpret the flow of data on a TSID.

The figure below is a sample portion of a TSID diagram. (The boxed letters are used to flag portions of the diagram that are explained in the paragraph following.)



A. FAMS takes user input via a terminal session.

B. FAMS reads data from three FIMS files but does not update FIMS data.

C. FAMS reads and updates the FAMS-FUNDS-PROJ file.

D. FAMS exports data to a microcomputer file at the Public Transportation Division.

E. Objects inside the box are in the scope of FAMS. Objects outside the application boundary are not part of FAMS. The right side represents only a portion of the FIMS application.