



## PRACTICES GUIDE

### PHYSICAL DATA MODEL

Issue Date: <mm/dd/yyyy>  
Revision Date: <mm/dd/yyyy>

## Document Purpose

This Practices Guide is a brief document that provides an overview describing the best practices, activities, attributes, and related templates, tools, information, and key terminology of physical modeling.

## Background

The Department of Health and Human Services (HHS) Enterprise Performance Life Cycle (EPLC) is a framework to enhance Information Technology (IT) governance through rigorous application of sound investment and project management principles, and industry best practices. The EPLC provides the context for the governance process and describes interdependencies between its project management, investment management, and capital planning components. The EPLC framework establishes an environment in which HHS IT investments and projects consistently achieve successful outcomes that align with Department and Operating Division goals and objectives.

A Physical Data Model is created during the Design Phase and is one of the components of the Design. It is a graphical representation of a physical database design. It is designed with the capabilities of a specific database management system in mind and optimizes the implementation of detailed data requirements and business rules in light of technology constraints, application usage and performance requirements.

## Practice Overview

A Physical Data Model is derived from the logical data model and includes all the database entities/tables/views, attributes/columns/fields and the relationship between the entities that were defined in the logical data model. Database performance, indexing strategy, physical storage and denormalization are important considerations of creating the physical data model. How the database is created is dependent on the capabilities and constraints of the database management system. Capacity and security requirements are impacted by the database management system.

## Best Practices

- **Document and Follow Modeling Standards** – Follow established naming conventions, standards for normalization and renormalization, primary key and index design, and referential integrity.
- **Define and Document Performance Thresholds** - Balance query performance and efficiency with overhead for transactions and maintenance of the database.
- **Define and Document Interface Requirements** – Follow guidelines for users and systems that may need access to the database.
- **Define and Document Security and Application Data Access** – Document security and application data access so that users and systems have access to only the information that they need.
- **Design** – Design the new database in accordance with OPDIV physical data modeling standards.

## Practice Activities

The following activities are performed as you create the physical data model from the logical data model:

- **Refer to the Logical Data Model** – Use the logical data model that was developed during the requirements analysis phase, as the basis for the physical data model.
- **Select a Methodology** - Select a methodology to create the database based on the specific project at hand. Several different data modeling methods are available, depending on the nature of the physical database management system selected.
- **Select a Modeling Tool**: Select a modeling tool that supports the creation of the physical

database management system and modeling method that is appropriate.

- **Identify the Entities/Tables** - Every entity/table defined in the logical data model must be mapped to a table depending on the type of database management system or an XML database that you are using. Provide a meaningful each entity/table.
- **Identify the Attributes/Columns/Fields** - Every attribute/column/field defined in the logical data model must be mapped depending on the type of database management system or an XML database that you are using. Provide a meaningful each attribute/column/field.
- **Normalize the entity/table** – Organize the database so that data is available in only one place.
- **Define the Domain Values** - Define domain values, physical data type, length and nullability of each column or field.
- **Define Default Values** - Define default values for columns or fields, especially for NOT NULL constraints.
- **Define Primary Keys and Indexes** - Define primary key and other indexes based on knowledge of the applications that the database must support.
- **Apply Other Techniques**, if needed - Apply several other techniques such as de-normalization, partitioning, etc. De-normalization is used to improve retrieval time potentially at the cost of additional space, additional insert time, and risk to data quality in data-warehousing types of applications.