Database Design Process

1. INTRODUCTION

At the end of the logical data modeling process, the deliverables passed to the database design process include a DDL script to create the database of the logical data model that has the logical attributes replaced by their logical data type implementations. These replacements can be done automatically in the CASE tool used for the logical data modeling.

This DDL script will create the tables of the logical data model but the script does not implement the reference tables and their relationships to the logical tables nor does it implement the non-business model tables specified in the Database Structure document.

The database design process executes the following tasks:

1. Determining the purpose of the database
2. Validating the inputs from the Logical Data Modeling phase
3. Selecting the tools for the database implementation
4. Creating the database tables

The objective of the database design process is to convert the logical data model into a complete physical database design and to generate the DDL script necessary to create the physical database that relies on declarative referential integrity for validating the contents of the database.

2. DATABASE PURPOSE

The primary purpose or use of the physical database will determine its structure. According to the Database Types document and the Data Life Cycle document the purpose is one of:

- Data sourcing – using the database to gather the data from a primary source of data, e.g., OLTP application, web app, IOT app, etc.
- Data reconciliation – using the database to detect errors in the data, to standardize the data (converting all the source gender codes to the code used in the ELDM) and to validate that the data conforms to the definition in the ELDM.
- Data reporting – using the database to generate reports for the business users

The database stakeholders (project budget controller, project manager, business sponsor, etc.) must all agree on the primary purpose of the database because the performance of the database when it is used for another purpose will be less than optimum. For example, using the data reconciliation database for reporting will cause delays in both reconciling the data and the speed of the reporting.

3. INPUT

The logical data modeling phase will provide at least six deliverables to document the ELDM. Depending on the deliverable, it can have more that one version with each version having a different viewpoint and different levels of detail about the logical data model.

The deliverables that are used to start the database design are the DDL scripts:

- the Logical Data Model DDL script
- the Reference Tables DDL script

The CASE tool used in the Logical Data Modeling should have automatically converted:

- the logical tables into physical tables with keys, alternate keys, indexes and relationships,
- the attributes into columns with physical data types and default values according to the Logical Data Type Implementation document.

The next step in the database design process is to validate that elements of the delivered DDL scripts follow the Database Naming Convention document for tables, columns, keys, alternate keys, indexes and relationships

4. DATABASE DESIGN ENVIRONMENT

Before the database design process can begin, the design working environment must be set up.

The first step is to choose a target DBMS. The DBMS chosen depends on the purpose of the database:

- For data sourcing – the target database may be one of ORACLE, SQL Server, etc.
- For data reconciliation – the target database may be one of ORACLE, SQL Server, Azure, AWS, etc.,
- For data reporting – the target database may be one of HANA, Neteeza, etc.

The next step is to choose physical data modeling (CASE) tool. Since there are no special processing requirement for the physical data model, the CASE tool may be one of ERWIN, ER/Studio, Power Designer, InfoSphere, TOAD, Visible Analyst, etc.

5. DATABASE STRUCTURE

The physical database will have a layered structure according to the Database Structure document. The logical data model only provided an incomplete view of the business layer of the physical database. Below are the sections that describe the work necessary for each of the required layers.

5.1 Business Data Layer

To complete the business layer of the database:

- Reverse engineer the logical DDL script to create the physical data tables
- Reverse engineer the reference table DDL script to create reference tables for each column that implements an enumerated logical data type attribute
- Create the relationships between the reference tables and the columns in the database that implemented the enumerated logical data type attributes.

Business layer structure depends on the database purpose:

- Data reconciliation – the business layer structure developed above is complete.
- Data capture – the scope of the business layer structure is too great for the data capture process. Remove the tables and relationships that are not part of the scope of the data capture application as outlined in the Enterprise Bus Matrix document.
- Data reporting – the structure of the business layer, which is in 5th normal form, must be converted into a 1st normal form or materialized view database for reporting on the particular business view of the data required by the end user.

5.2 Error Logging Layer

To define the error handling layer of the database, create the tables described in the Database Structure document.

5.3 Compliance Auditing Layer

To define the compliance auditing layer of the database, create the tables described in the Database Structure document.

5.4 User Security Layer

To define user security layer of the database, create the tables described in the Database Structure document.

6. DELIVERABLES

At this point, the database design process is complete. In the design tool, remove any unused items from the design project and generate the DDL for the database. This is the first deliverable for the design phase.

The deliverables required to close out the design process are:

1. DDL for database – this is the script used in the DBMS to create the tables.
2. Physical Database Diagram – the diagram that shows the relationships between the tables in the database. There may be more than one diagram and the diagrams may be at various levels of detail.
3. Data Dictionary Report – the list of all the elements of the database, their physical names, logical names and definitions.
4. Physical Table List – the list of all the tables in the database. This list is used to populate the Database Table in the Data Inventory Database.
5. Physical Column List – the list of all the columns in each of the tables in the database. This list is used to populate the Database Table Column in the Data Inventory Database.
6. Database Interface List – the list of all the single table stored procedures required to implement the database interface. This list is used to populate the Database Interface Procedure in the Data Inventory Database.

7. REFERENCES

Database Structure – http://thedataorg.com/Page05/DatabaseStructure.pdf

Database Types – http://thedataorg.com/Page05/DatabaseTypes.pdf

Data Life Cycle – http://thedataorg.com/Page09/DataLifeCycleArchitecture.pdf

Logical Data Type Implementation –
http://thedataorg.com/Page05/ImplementingLogicalDataTypes.pdf

Database Naming Convention – http://thedataorg.com/Page05/DatabaseNamingConvention.pdf

Enterprise Bus Matrix Diagram – http://thedataorg.com/Page09/EnterpriseBusMatrix.pdf