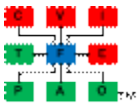

Logical Data Modeling Process

1. INTRODUCTION

Before we begin the process of data modeling, we need to have clear in our minds what is the end product of the data modeling process i.e., What is a data model?
Businesses create their data by measuring business processes. This measured data is placed in databases which are designed to efficiently store the data at each step of the business process being measured.

The semantics of the business processing (business requirements) are very complex – far too complex to explain clearly and reliably in a natural language like English.

By comparing the database development lifecycle to the software development lifecycle, we see that a logical data model is a set of documents for the database specification that is equivalent to the application functional specifications for the business processes.

Both the logical data model and the application functional specification are semantic models intended to be exactly that: a complete, consistent, clear, concise, precise, unambiguous, and validated specification of the business requirements. The logical data model describes the data (logical tables, logical table attributes and logical data types) and relationship semantics terms that the business understands. The application functional specification describes the processing semantics. (Steve Tockey)

The logical data model is a technology and implementation independent specification of the required data structures in the database. A database is a storage place for data records that does not have any duplicate attributes (Codd's first normal form requirement) and does not allow duplicate records (Codd's set requirement). The logical data model is the foundation of the databases that are implemented to create an enterprise data architecture.

Data models not only serve as a semantic data specification for designing a physical database, you can use data models to communicate with other stakeholders and verify that a data model also:

- represents the paths in the database of each of the business processes.
- documents important business rules
- defines concepts and helps to standardize business terminology.
- provides clarity and helps to uncover blurred thinking and ambiguities about business processes.

2. DATA MODELING LIMITATIONS

There are three intrinsic limitations to the semantic/logical data model

- **Arbitrariness** – Jevons, William, Classification
  The partitioning of the data requirements into a logical data model of non-overlapping tables by normalization and the rules for creating the non-overlapping subsets on each level is a classification process (the creation of a taxonomy). The process of creating a taxonomy is an arbitrary process. Jevons has demonstrated that this process depends on the choices of the person creating the taxonomy (data modeler). Kent has shown the difficulty that resides in deciding which labels in the metadata relational data model are the logical master tables.
- **Incompleteness** – Godel, Kurt, Incompleteness Theorem
  The relational data model and normalization are based on set theory and form an axiomatic system. Godel has shown that an axiomatic system cannot show whether it is consistent or not. This makes the acceptance of the logical data model a decision of the stakeholders of the data model.
- **Uncertainty** – Heisenberg, Werner, Uncertainty Principle
  The process of understanding and converting the business data requirements into a logical data model will involve asking the business stakeholders questions about the data requirements. The answers to the questions can tell us what the current data requirement is, but by asking questions about the current data requirement, we change the future state of the data requirement.

3. DATA MODELING TOOLS

To complete the ELDM specification, you should be using a data modeling tool. There are many tools to choose from, ER/Studio, Erwin, Oracle SQL Developer Data Modeler, Visible Analyst, etc. The major difficulty is that the tools focus on the physical database description and not on the logical/semantic specification of the data. For example, Erwin will not let you create a table without specifying the contents of the table and it assumes that the contents are columns and each column has a physical data type. This is the physical database design process not the semantic specification.

A modeling tool that lets you specify the logical attributes using logical data types is Visible Analyst.

4. DATA MODELING PROCESS

There is a five-step process for documenting the Semantic/Logical Data Model.

4.1 Document the Master Logical Tables

For each Account and Journal entity in Conceptual Data Model diagram, create a master Logical Table entry giving:

- the logical table name,
- the logical table type,
- the name of the conceptual entity it is related to
- a description of the contents of the master logical table.

4.2 Fill out the master logical table questionnaire

For each logical master table, fill out the table attributes questionnaire. The attributes are the business names for data not the fields or columns displayed by an application. The attributes questionnaire template is shown in Figure 1.

			Master Table Attribute Pattern			
Table Attribute Name	Logical Data Type	Attribute Description	Question 1: Purpose	Question 2: cardinality	Question 3: occurrence	Entity Pattern Case
			descriptive			
			relationship			
			organizational			

Figure 1: Attributes Questionnaire Template

The logical data types used to describe the table attribute are given in Logical Data Type Definitions. The initial list of attributes can be found using the Conceptual Data Model document delivered in the previous phase. The Master Table Attribute pattern questions can be found in Master Data Attribute Patterns.

An example of the attribute questionnaire for Sales Event/Measurement is shown below:

			Master Table Attribute Pattern			
Table Attribute Name	Logical Data Type	Attribute Description	Question 1: Purpose	Question 2: cardinality	Question 3: occurrence	Entity Pattern Case
Event Date	calendar date	the date on which the event completed	descriptive	constant in time	single valued	11 Description (Type 1)
Start Time	clock time	the start time of the task	descriptive	constant in time	single valued	11 Description (Type 1)
End Time	clock time	the end time of the task	descriptive	constant in time	single valued	11 Description (Type 1)
Elapsed Time	measured quantity	the actual amount of time involved	descriptive	constant in time	single valued	11 Description (Type 1)
Quantity	measured quantity	the quantity of the product that was ordered	descriptive	constant in time	single valued	11 Description (Type 1)
Amount	monetary amount	the price that was charged	descriptive	constant in time	single valued	11 Description (Type 1)
Customer	table key	who ordered the product	relationship	other logical table	single valued	7 Business Process Measurement
Employee	table key	which employee did the work	relationship	other logical table	single valued	7 Business Process Measurement
Inventory Item	table key	what inventory item was delivered to the customer	relationship	other logical table	single valued	7 Business Process Measurement
Supplier	table key	who provided the inventory item	relationship	other logical table	single valued	7 Business Process Measurement
Business Unit	table key	which department made the sale	relationship	other logical table	single valued	7 Business Process Measurement
Product	table key	what product was ordered by the customer	relationship	other logical table	single valued	7 Business Process Measurement
Task	table key	what task was completed for the event	relationship	other logical table	single valued	7 Business Process Measurement
Financial Calendar	table key	Event Date	relationship	other logical table	single valued	7 Business Process Measurement
Source Document	document	the business's accounting document (invoice)	organizational	profile	single valued	3 Description (Type 1)
Reference Document	document	the customer's accounting document	organizational	profile	single valued	3 Description (Type 1)

Figure 2: Sales Event Attributes Questionnaire

4.3 Create the Attributive and Associative Logical Tables

From the last column in the Attribute Questionnaire and Figure 3 in the Master Data Attribute Patterns, create the attributive logical tables and associative logical tables. The naming of the tables is important to avoid confusion. The attributive and associative table names should be of the form <master table name>_<attributive table name>, for example, CUSTOMER_CONTACT, SUPPLIER_ADDRESS, EMPLOYEE_PHONE, etc.

For each Attributive and Associative Logical table, create a Logical Table entry giving:

- the logical table name,
- the logical table type – attributive or associative
- the name of the conceptual entity it is related to
- a description of the contents of the logical table.

4.4 Document the attributes in each Logical Table

For each entry in Logical Table, create Logical Table Attribute entries giving:

- the logical table name,
- logical attribute name
- sequence number – of the attribute within the logical table
- the logical data type name,
- the default value of the attribute.
- a description of the meaning of the logical attribute.

4.5 Document the relationships between the Logical Tables

For each master logical table, document the relationships to the attributive and associative logical tables by entering:

- Master logical table name
- Dependent logical table name
- Relationship sequence number – usually one, unless it is the implementation of a many to many relationship.
- Relationship type – one to many or one to one
- Description

Into the Logical Table Relationship in the enterprise data dictionary.

5. VALIDATE THE LOGICAL DATA MODEL

Once the logical data model is complete, in the CASE tool, ensure that the logical data model contains no redundancies or errors. Using the CASE tool, create the logical data model documentation for each stakeholder consisting of:

- A data model diagram containing the view of the stakeholder
- The data dictionary with the model view components highlighted
- The LDM DDL script with the model view components highlighted.

Validate that the logical data model meets the data requirements of each stakeholder and get signoff.

6. CAUTION

Suggestions to denormalize the database occur during the application implementation. The objective, from the application point of view, is to speed up some aspects of the data access processing. There are many factors that influence the cost of database processing.

Denormalization implies that the answers to the master logical table questionnaire that were used in the logical table pattern analysis are wrong and therefore the data model needs to be modified.

Generally, the suggestion for denormalization is based on using a single database for data capture, data processing and data reporting. Denormalization may improve access performance during queries and reporting, but it requires extra processing during update to access multiple records to remove any data value discrepancies from the database.

The intended use of the database implies the nature of the data processing and the type of database structure required. Given the intended use (data capture or data reporting), the structure and data model simplifications can be implemented correctly and denormalization is not required.

7. DELIVERABLES

At this point, the logical data model documentation process is complete.

The deliverables required to close out the logical data modeling process are:

1. Data model documentation deliverables:
    a. Logical Data Model Document – the list of all the logical tables of the model, their definitions, and their attributes.
    b. Logical Data Model Diagram – the diagram that shows the relationships between the logical tables in the data model. There can be a number of diagrams at various levels of detail.
    c. Logical Data Model Dictionary – this is the alphabetical listing and definitions of all of the items used in the logical data model.
2. Database Implementation deliverables:
    a. Logical Data Model DDL Script – this is the DDL script that will create the logical tables that will be used in the business data layer of the physical database.
    b. Reference Table List – This is the list of reference or lookup tables required to implement the enumerated logical data types used in the data model.
    c. Reference Table DDL Script – this is the DDL script that will create the reference tables that will be used in the physical database.

The implementation deliverables can be used as input to the database engineering phase to algorithmically create a database schema for the associated application development project.

As you would not build a house or a bridge without a blueprint, so why would you build an OLTP, data warehouse, or OLAP application without a semantic/logical data model?

8. REFERENCES

Codd, Edgar, “A relational model for large shared databanks”, Communications of the ACM, Vol. 13, No. 6, Jun 1970

Date, C.J., *The Relational Data Dictionary*, O’Reilly Media, Inc., Sebastopol, CA, 2006

Jevons, William Stanley, The Principles of Science, New York, Dover Publications, 1874

Kent, William, Data and Reality, North Holland Publishing, New York, 1978

Steve Tockey, *a personal email*, Construx Software Builders, Inc, Bellevue. WA, 2018

Logical Data Model Document – Logical Data Model Document

Logical Data Model Diagram – Logical Data Model Diagram

Logical Data Model Dictionary – Logical Data Model Dictionary

Logical Data Model DDL Script – Logical Data Model DDL Script

Reference Table List – Reference Table List

Reference Table DDL Script – Reference Table DDL Script

