DATABASE MANAGEMENT SYSTEMS – Data modelling

Conceptual, Logical and physical data modelling

Basically, data modeling serves as a link between business needs and system requirements.

Three types of data modeling are as follows:

- Conceptual modelling
- Logical modeling
- Physical modeling

If you are going to be working with databases, then it is important to understand the difference between conceptual, logical and physical modeling, and how they relate to one another. Conceptual, Logical and physical modeling are described in more detail in the following subsections.

Conceptual modelling is a high level data modelling process. It captures the overall requirement of the business. It identifies the required entities, their attributes, relationships and constraint. It can include assumptions also. It should include the Primary key, which should be underlined (**but no foreign keys here**). The attributes identified should pertain to one entity only. No calculated/derived attributes here; only base attributes. Relationship can 1:1; 1:m or m:n.

Logical modeling deals with gathering business requirements and converting those requirements into a logical model. Still the logical model revolves around the needs of the business, not the database, although the needs of the business are used to establish the needs of the database. Logical modeling involves gathering information about business processes, business entities (categories of data), and organizational units. **It should include entities, attributes, relationships, primary keys and foreign keys. Any m:n relationships should be broken down into 1: m and n:1 creating an additional associative entities with all of its attributes.** After this information is gathered, diagrams and reports are produced including entity relationship diagrams, business process diagrams, and eventually process flow diagrams. The diagrams produced should show the processes and data that exists, as well as the relationships between business processes and data. Logical modeling should accurately render a visual representation of the activities and data relevant to a particular business.

Logical data modeling is the exercise to document and define the relationships between data elements. Typically, it involves

1. identifying entities (e.g., "customers", "orders") from the business environment
2. identifying how specific instances of each entity are differentiated from other instances, the logical key (e.g., "customer_id", "order_number")
3. grouping together other attributes that describe the entity (e.g., "customer_address", "ship_to_address") **AND** which can also be uniquely determined based on the entity's key

4. finally, documenting the business rules (relationships) between the entities (e.g., "a customer may place one or more orders", "each order must be placed by exactly one customer")
5. Splitting any m:n relationships into 1:m and n:1, resulting in an additional associative entity.

Note that logical data modeling does not consider any physical representation of how the data will be stored and it doesn't attempt to anticipate or correct any performance issues that may arise during implementation.

Tasks such as these occur during the physical data modeling phase. At this point decisions will have to be made about data storage (Oracle relational DB, MS Access, mySQL.). Considerations for how the data needs to be accessed, combined ("joined") and the performance characteristics of the intended deployment environment will be documented.

Taking the purely logical entities, attributes and relationships, the physical modeler makes (and documents the reasons for!!!) altering the logical model. One-to-many relationships may be "denormalized" into the "one side" of the relationship, forming a repeating group (e.g., collapsing "a customer may have multiple phone numbers" into just a "customer" entity with attributes of "home_phone", "work_phone", "mobile_phone", "fax_phone").

Decisions about where to place the data (same database? different databases on different servers?) as well as partitioning, archival, purging plans have to be done within the constraints of the business requirements.

Oddly enough, logical data modeling is more of a science and physical modeling is more of an art in that two business analysts can discuss the logical model and resolve most differences of opinion logically (so to speak) by providing real-world examples that would negate a particular representation. Physical database design is not so precise, however. The modeler must know (or anticipate) a number of things about future uses of the data and about the characteristics of the particular database management system, programming language(s), communication channels, etc. Many assumptions go into the creation of a physical model. How well that model will eventually perform depends, in large part, in the quality of those assumptions.

The diagrams and documentation generated during logical modeling is used to determine whether the requirements of the business have been completely gathered. Management, developers, and end users alike review these diagrams and documentation to determine if more work is required before physical modeling commences.

Typical deliverables of logical modeling include

- Entity relationship diagrams
  An Entity Relationship Diagram is also referred to as an analysis ERD. The point of the

initial ERD is to provide the development team with a picture of the different categories of data for the business, as well as how these categories of data are related to one another.

- Business process diagrams
  The process model illustrates all the parent and child processes that are performed by individuals within a company. The process model gives the development team an idea of how data moves within the organization. Because process models illustrate the activities of individuals in the company, the process model can be used to determine how a database application interface is design.
- User feedback documentation

Physical Modeling

Physical modeling involves the actual design of a database according to the requirements that were established during logical modeling. Logical modeling mainly involves gathering the requirements of the business, with the latter part of logical modeling directed toward the goals and requirements of the database. Physical modeling deals with the conversion of the logical, or business model, into a relational database model. When physical modeling occurs, objects are being defined at the schema level. A schema is a group of related objects in a database. A database design effort is normally associated with one schema.

During physical modeling, objects such as tables and columns are created based on entities and attributes that were defined during logical modeling. Constraints are also defined, including primary keys, foreign keys, other unique keys, and check constraints. Many to many relationships are broken down to 1:many and many :1.Views can be created from database tables to summarize data or to simply provide the user with another perspective of certain data. (This we will do when we do SQL either in Oracle or mySQL) Other objects such as indexes and snapshots can also be defined during physical modeling. Physical modeling is when all the pieces come together to complete the process of defining a database for a business.

Physical modeling is database software specific, meaning that the objects defined during physical modeling can vary depending on the relational database software being used. For example, most relational database systems have variations with the way data types are represented and the way data is stored, although basic data types are conceptually the same among different implementations. Additionally, some database systems have objects that are not available in other database systems.

Typical deliverables of physical modeling include the following:

- Server model diagrams (Physical ERD)
  The server model diagram shows tables, columns, and relationships within a database.
- User feedback documentation

When you do your data dictionary next time it has to cater for the physical data model that you do here.