The Data Administration Newsletter (TDAN.com)
Robert S. Seiner - Publisher



# SPONSOR EDUCATION SERIES

## ENTERPRISE DATABASE PRINCIPLES

**Michael M. Gorman - Whitemarsh Information Systems**

[Return to Sponsor Education Series Menu]

### Introduction

An enterprise database must embrace an environment that:

- Is database object centered as opposed to data only or process only centered
- Has a metadata repository as the locus of all specification development, implementation, and ongoing maintenance
- Allows for the implementation of continuous-flow, evolutionary systems development as well as the traditional discrete cradle-to-grave systems development

Critical to successful enterprise database is:

- The metadata repository that contains the metadata representing the enterprise business model upon which all the database applications are founded
- The discovery of the metadata for the enterprise's mission and the database objects that reflect the enterprise's past, present and future existence

This paper addresses the set of principles which, if followed, lead to quality enterprise database. Enterprise database clearly resides within the database object column of the Knowledge Worker Framework. The Knowledge Worker Framework book is free and can be downloaded from the Whitemarsh website, www.wiscorp.com.

### Enterprise Database Principles

The fundamental set of activities that form the basic operations of an enterprise are timeless. While these activities are subject to changes around the edges such as presentation layer technologies, deployment strategies, and access and storage strategies, these activities need to be specified only once, and if practical, be implemented only once. And, when maintenance is required, the single instance of the specification and the attendant implementation must be modified and re-implemented in unison. The first principle then is that:

*1. Enterprise database fundamentals must be specified only once, and where practical are implemented only once. Database objects result when implemented through ANSI SQL/3 DBMS.*

The second principle is that:

*2. Enterprise database specifications and, in turn, its implementation are modified together as a single unit according to well defined rules that govern the metadata based information systems (repositories) that accomplish its unambiguous transformations.*

A corollary of the second principle is that:

All maintenance activities begin with the modification of the enterprise database specification, not with the specification's implemented programs or data.

From the early 1980s when first applying these principles, the following has been realized by its clients:

- Increased consistency within and between specifications and implementations across the enterprise,
- Lowered risk because of known processes, deliverables, accurate estimates, and higher quality results,
- Efficiently expended resources for both computing and human resources,
- Enhanced timeliness in project plans and deliveries.

## Comprehensive Model

For enterprise database to be successful, it must be based on a comprehensive model that consists of a completely specified, implemented, and maintained set of database objects. The term comprehensive implies that all database objects required for a complete set of business missions are addressed.

A model of a database object environment consists of four interconnected database object subcomponents: data structure, data structure processes, information systems, and states.

Database objects are independent of any particular vendor's implementation of ANSI/SQL3 DBMS, operating system, and presentation layer.

In brief, the data structure component of database objects contains the defined set of data to be controlled by the database object. It is not uncommon for database objects to contain from one to many tables of simple to complex data.

The database object data structure processes are those that maintain the integrity of individual data structure components of database objects. Within DBMSs, these processes are commonly known as column integrity rules, table constraints, assertions, triggers, and stored procedures.

The database object information systems are those systems contained completely within SQL/3 that transform database objects from one valid state to another.

The database object states are well defined, policy based value representations of the database objects related to well know and policy based business activity results.

Essential to the correct specification of database objects is a single high-level statement that demarches the overall scope and purpose of the enterprise. The third principle then is that:

*3. The enterprise must be represented through a mission statement that defines the functionality and boundaries of the included database objects*

This statement is commonly referred to as the corporation's mission description. The fourth flows from the third, and is:

*4. All database objects must be based on the enterprise's mission description.*

## Database Object Data Structures

The database object structures represent the data structure classes that are both valid and normalized.

The database object data structure is valid if it contains all the data implied by the enterprise database system's mission description.

The database object model is normalized if all the data structures that comprise the database objects are organized according to the rules of third normal form.

The reason for having a database object data structures normalized is that they only have to be defined once; the DBMS only has to store the resultant data once; the programs that add, delete, and modify the data only have to be designed, written, and debugged once; user training only has to be developed once; and finally, the data only has to be documented once. And, with respect to maintenance, the specification is contained in only one place, requiring maintenance In only one place.

Once the specification is modified, the attendant software and data needs to be modified only once to carry out the specification change.

## Valid Representation

The data represented by database objects must embody all the data implied by enterprise missions. To achieve this, the enterprise’s mission description undergoes further analysis to determine the database’s domain, which is then further divided into its major subdomains. Each subdomain is then defined and graphically represented through an entity-relationship diagram. All the subdomain entity-relationship diagrams are coalesced into a single diagram, which is the graphical analog to the data required for the entire mission description. Each entity within the diagram is then employed in the derivation of the enterprise’s database objects. The fifth principle then is that:

*5. The database objects of the enterprise—in combination—represent all the data implied by the database domain.*

The next step in the representation of a valid specification is the identification of data elements. These are found through analysis of policy manuals, forms, reports, user interviews, screen layouts, and the like. These data elements are examined with respect to enterprise database and, if relevant , they become part of the enterprise’s database specification. The sixth and seventh principles then are that:

*6. Each data element represents a standalone, context and implementation independent business*

*fact that is founded on policy analysis.*

*7. All data elements represented through whichever columns are appropriate in databases—in combination—represent all the data implied by the mission description.*

## Normalized Database Object Data Structures

To achieve single-instance data specifications, database data structure segment candidates and data elements are joined and then represented as simple normalized tables and/or complex tables wherein each segment is normalized into at least third normal form. A database object thus typically contains from one to many third normal form tables as its data structure.

Data elements, as abstract representations of context independent business data one level higher than the columns of database object data structures, become the metadata templates for columns. The eighth principle that addresses database object data structure quality is:

To represent data only once, all data must be defined in third normal form and be represented in only one data structure segment of only one database object .

## Database Object Data Structure Types

The four types of database object data structure tables are: root, wholly contained, shared, and referenced. A root database object table is one in which there is a single primary key column that acts as the database object's identifier.

A wholly contained database object data structure table is one in which the high-order primary key column is the primary key from the database object's root data structure table and the other primary key columns are not the part of the primary key of any other database object. A wholly contained database object table cannot therefore be a member of any other database object class.

Further, instances of this database object table cannot belong to any other database object other than the one explicitly valued through the high-order database object primary key's value.

A shared database object data structure table is one in which the primary key columns are from at least two different database objects. Only one database object is defined as the primary owner to enable semantically correct cascade deletes.

The referenced database object data structure table is one that is accessed through a traditional relational foreign key value. The purpose of the referenced database object data structure table to represent factored data from tables in other database objects.

The ninth through eleventh principles, those that specifically address database object data structures are:

*9. All data represented by a database object is defined within wholly contained database object data structure tables.*

*10. Database objects can reference data within other referenced database object data structure through traditional foreign key references.*

*11. Data that is commonly owned by multiple database objects must however be primarily owned by one database object and secondarily owned by other database objects.*

## Database Object Data Structure Processes

The fundamental integrity of database object data structure instances are controlled through rule-based processes that either succeed or fail and that specify the interaction between:

- Columns within one normalized structure,
- Columns in different database object tables,
- Rows of the same table, or
- A column in a collection of rows in one table and a single column of a row in another table.

Database object processes typically consists of two types: acceptable data value specification and acceptable result enforcement. Simply put, database object processes are the codification of business policies that enforce business policy. Acceptable data values relate to valid and invalid values, ranges of values, and the like. Acceptable process results deal with calculations, selection logic, and so forth. In database, all such business policy rules must be "attached" to tables within the database objects in such a way that these rules are automatically enforced regardless of the implementing DBMS environment. If these business rules are encoded in programs that are not integral to DBMS encased database objects,and if there are ten programs evoking these rules, then the rules must be specified identically ten times for the results to be the same.

In addition to having single specification database object processes, the enforcement of these rules by the DBMS allows for the use of SQL/3 DBMS based languages for the assertions and triggers.

If the DBMS does not support these languages all database object processes must be included in middle-layer or presentation layer programs.

The twelfth through fourteen principles, that is, those related to database object processes are that:

*12. Database object processes act solely on the database object within which they are defined.*

*13. Database object processes do not cause database objects to change state. Rather they only test that a changed state is valid.*

*14. Database object processes must be specified such that only validated database objects are allowed within the database.*

## Database Object Information Systems

A database object information system is one that operates completely within the database object domain such that the targeted database objects are transformed from one valid state to another. The environment in which the database objects operate must be independent of any enveloping DBMS, operating system, and presentation layer.

Database objects are changed from one state to another according to the life cycle of the database object.

The database object information systems are valid if they contain all the processes necessary to change the database objects from one state to the next throughout the database object's life cycle. The database object information systems are normalized if the processes within the database object information systems are organized according to process rules analogous to the rules for third normal form data.

When the database object information systems are valid and normalized they only have to be defined once, and the programmers only once have to design, code, and debug the programs that add, delete, and modify the data.

Furthermore, the user training for these database object information systems only has to be developed once, and finally, the database object information systems only have to be documented once. And with respect to maintenance, the specification is contained in only one place, requiring maintenance in only one place. Once the specification is modified, the attendant software and data needs to be modified once to carry out the specification change.

The fifteenth principle then is that:

*15. The database object information systems, individually or within strictly defined collections are the sole agents for transforming database objects from one valid state to another.*

A process within a database object information system is in "third normal form" if it has:

- Only one purpose
- No nested subprograms
- No logic branching based on data values

If the database object information systems contain only third normal form processes, there is no process specification redundancy. These processes must also be implemented as nonredundantly as possible.

The benefit of process normalization (nonredundancy) is that since enterprise policy (translated into computer programs) is specified and implemented in only one place, understanding, documentation, design, programming, debugging, and maintenance all become much simpler.

The sixteenth principle is that:

*16. To represent processes only once, all database object information system processes must be defined in third normal form.*

## Database Object States

As stated above, database object states are the well defined, business policy based value representations of database object structures that map onto well known and policy based business activity results.

Database object states reflect the life cycle of the database object. The states begin with the null state, and then proceed through a set of valued states and then return to the null state. These states correspond to the DBMS verbs, add, modify, and delete. The add process transforms from null to valued. Modify transforms from one valued state to another. The final process deletes the valued state altogether.

The database object states correspond to its life cycle which is determined through an examination of business artifacts such as legal documents, data entry and update forms, and court filings. These materials are examined to determine the exact sequence of transformations of the data structures within the database object. The transformations are laid out in a non-overlapping sequence.

The database object states are examined to determine both the pre-states and the post-states.

The pre-states must be specified so they can be used as the basis for selecting database objects that exist in that pre-state. If the database object is not in a specific pre-state then it cannot be transformed to the valid next state. The post-state must be fully analyzed so that when the transformation is presumed to be complete, its values can be tested to ascertain that the post-state has been achieved. If achieved, then the database object retains the achieved state, otherwise, it is rolled back to its pre-state.

The seventeenth through nineteenth principles related to database object states are:

*17. Database objects are allowed to be transformed from only one valid state to a pre-specified next state.*

*18. The set of all database object states represent the entire life cycle for a database object*

*19. The collection of all states for all database objects represent the complete set of all business events in the enterprise that result in persistent data.*

## Metadata Repository

A metadata repository is essentially a database application in which the stored values represent enterprise database metadata rather than business fact data about personnel, loans, securities, and the like. Ideally, the data about data, or metadata, is able to be accessed by the DBMS so that it can more effectively manage the enterprise database environment. The metadata repository becomes then the live specification of enterprise database objects and other information systems that are in development, in production, and evolution and maintenance.

The twentieth principle is that:

*20. Enterprise database cannot be defined, implemented, and maintained successfully without the aid of a metadata repository.*

By successfully, it is meant that enterprise database must not only be specified according to the principles specified above, it must also be implemented and maintained according to these principles. If enterprise database is only specified according to these rules, but not implemented and maintained according to these rules, disintegration results.

---

*Michael M. Gorman, President of Whitemarsh Information Systems Corporation, has been involved in database and DBMS for almost 30 years. Mr. Gorman has been the Secretary of the ANSI Database Languages Committee, X3H2 for 19 years. X3H2 standardizes SQL. A full list of Whitemarsh's clients and products can be found on the web site, www.wiscorp.com. The goal of the web site, WisWeb, is to make data management books, courses, methodologies, software, and metrics available to the database community through electronic publishing and downloading. WisWeb memberships are very reasonable and are designed for the individual, the ISD organization, universities/colleges, and professional training organizations.*

*Whitemarsh Information Systems Corporation*
*2008 Althea Lane*
*Bowie, Maryland 20716-1518 USA*
*Phone: +1.301.249.1142*
*FAX: +1.301.249.8955*

*Email: mmgorman@wiscorp.com*
*WWWeb: http://www.wiscorp.com*

**[Return to Sponsor Education Series Menu]**

**The Data Administration Newsletter (TDAN.com)**
**Robert S. Seiner - Publisher and Editor - rseiner@tdan.com**