

Implementation Of Quality Standards in Object Nomenclature in Relational Databases

Cesar Yesid Barahona Rodriguez
Universidad de Cundinamarca
Facativá, Cundinamarca, Colombia
cbarahona@ucundinamarca.edu.co

Fredy Fabian Rodriguez Joya,
Efrain Andres Vergara Serrato.
Universidad de Cundinamarca
Facativá, Cundinamarca, Colombia
fredhyjoya@gmail.com
efrainvergara.udec@gmail.com

Abstract— This article presents the considerations and recommendations that must be taken into account to evaluate the nomination criteria in relational database managers (MySQL, PostgreSQL), using the ISO 19139 standard, through its characteristics and metrics, in which Regarding the quality of the databases in their nomenclature, in order to highlight what is their impact on software development and make known in which areas of development this standard has been used and in this way, make known the quality standards necessary when designing and finding the necessary vocabulary to make correct use of the standard in the databases.

Keywords— Database, Database manager, nomenclature, ISO 19139, Standar.

I. INTRODUCTION

En la actualidad cuando se desarrolla software se suelen aplicar estándares de calidad para dar a conocer las necesidades de los usuarios; de tal forma, en algunos casos se incurre en el error de considerar que si cumple con los factores dados por el estándar se asegura calidad, y no se toma en cuenta la revisión de calidad en el desarrollo de software, en este caso, para el correcto nombramiento de los diferentes objetos que llevan las bases de datos relacionales, se debe tener en cuenta un estándares de calidad, no solo a nivel de base de datos sino también a nivel de instancia, ya que los nombres de las propias bases de datos deben seguir normas específicas que sugieran a simple vista su aplicación.

En el desarrollo de software, se habla mucho acerca de buenas prácticas, patrones de diseño y una cantidad de vocabulario como patrones, diseños, etc. Y aunque el desarrollo a nivel de base de datos en muchos casos no exige una arquitectura, es importante tener en cuenta ciertos parámetros a la hora de crear una base de datos y, por consiguiente, la nomenclatura de los objetos que lleva la base de datos (tablas, campos, funciones procedimientos almacenados, vistas, índices, llaves, etc.).

Un sistema de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que permitan a los usuarios acceder y modificar estos archivos. Uno de los

propósitos principales de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.[1]

El presente artículo, tiene como fin, facilitar la tarea de los programadores y desarrolladores, en el diseño de bases de datos relacionales, teniendo en cuenta el uso de gestores de bases de datos como lo es MySQL y PostgreSQL, mediante la implementación de estándares que permitan, tener una correcta nomenclatura en el nombramiento de los diferentes objetos en una base de datos.

II. THEORETICAL FUNDAMENT

En esta sección, se darán a conocer algunas definiciones que serán necesarias, para dar a conocer los correctos lineamientos a la hora de desarrollar y diseñar una base de datos.

A. Base de Datos

Una base de datos es un conjunto estructurado de datos que representa entidades y sus interrelaciones. La representación será única e integrada, a pesar de que debe permitir utilizaciones varias y simultáneas.[2]

De tal forma, una base de datos se diseña para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información.

De igual manera, los datos relacionados se almacenan de forma tal que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc.[3]

B. Aplicaciones de las Bases de Datos

- Bancos: Para información de los clientes, cuentas y

préstamos, y transacciones bancarias.

- **Aerolíneas:** Para reservas e información de planificación. Las líneas aéreas fueron de los primeros en usar las bases de datos de forma distribuida geográficamente (los terminales situados en todo el mundo accedían al sistema de bases de datos centralizado a través de las líneas telefónicas y otras redes de datos).[1]
- **Universidades:** Para información de los estudiantes, matrículas de las asignaturas y cursos.
- **Telecomunicaciones:** Para guardar un registro de las llamadas realizadas, generación mensual de facturas, manteniendo el saldo de las tarjetas telefónicas de prepago y para almacenar información sobre las redes de comunicaciones.[4]
- **Finanzas.** Para almacenar información sobre grandes empresas, ventas y compras de documentos financieros.
- **Ventas.** Para información de clientes, productos y compras.
- **Producción.** Para la gestión de la cadena de producción y para el seguimiento de la producción de elementos en las factorías, inventarios de elementos en almacenes y pedidos de elementos.[2]

C. Sistema Gestor de Bases de Datos (SGBD)

Un sistema gestor de bases de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.[1]

A continuación, se dará a conocer dos de los gestores de bases de datos, que mayor uso tienen en la actualidad, teniendo en cuenta sus características y usos.

• **MySQL**

MySQL es un sistema gestor de bases de datos. Pero la virtud fundamental y la clave de su éxito es que se trata de un sistema de libre distribución y de código abierto. [2]

Lo primero significa que se puede descargar libremente de Internet, lo segundo (código abierto) significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo.

Estas son algunas de las características principales de MySQL.

- Está desarrollado en C/C++.
- Está optimizado para equipos de múltiples procesadores.[5]
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o en aplicaciones.
- Cuenta con diversidad de tipos de datos.
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el sistema gestor de base de datos a cada caso concreto.[2]
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.[1]
- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.[2]
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.

Por consiguiente, este gestor se abarcará más adelante, para dar a conocer la nomenclatura en el nombramiento de sus objetos correspondientes.

• **PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional y libre, que agiliza la interacción de cliente, servidor y base de datos, donde PostgreSQL es el que realiza la mayoría del trabajo referente a bases de datos cuando se le hacen peticiones muy concurrentes.

Estas son algunas de las características principales de PostgreSQL.

- Está desarrollado en C.
- Cuenta con diversidad de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus opciones de conectividad abarcan TCP/IP, sockets Unix y sockets NT, además de soportar completamente ODBC.[2]
- Es altamente confiable en cuanto a estabilidad se refiere.
- Control de concurrencia multi-versión, lo que mejora sensiblemente las operaciones de bloqueo y transacciones en sistemas multi-usuario.[1]
- Tiene soporte para vistas, claves foráneas, integridad referencial, disparadores, funciones, subconsultas y casi todos los tipos y operadores.
- Copias de seguridad en caliente (Online/hot backups).
- Implementación de algunas extensiones de orientación a objetos. En PostgreSQL es posible definir un nuevo tipo de tabla a partir de otra previamente definida.[5]

Por consiguiente, este gestor se abarcará más adelante, para dar a conocer la nomenclatura en el nombramiento de sus objetos correspondientes.

• Objetos de las Bases de datos

En esta sección, se darán a conocer los diferentes objetos que componen una base de datos.

- Campos: Es un espacio de almacenamiento para un dato almacenado en la base de datos, pueden tener diferentes tipos con enteros, texto, valores lógicos etc.[6]

- Tablas: Es el tipo de modelado donde se almacenan los datos en campos que la componen.[6]

- Vistas: Es una consulta de una o varias tablas de una base de datos, se muestra en forma de tabla también.

- Procedimiento Almacenado: Un procedimiento almacenado es un conjunto de comandos SQL que pueden almacenarse en el servidor.

- Funciones: Da la posibilidad de ejecutar las sentencias SQL utilizando lenguajes procedimentales (PL).

- Triggers: Son eventos que se ejecutan cuando se cumplen condiciones determinadas, al hacer un insert, update o delete en una tabla de una base de datos.[2]

- Llaves Primarias: Indicador único en una tabla.

- Llaves Foráneas: Indicador de uno o más campos de una tabla que hacen referencia al campo o campos de una llave primaria de otra tabla.

• Otras Definiciones

Se darán a conocer conceptos que se utilizarán para la nomenclatura de los objetos de la base de datos.

- Camel Case: es una metodología de nombramiento que consiste en dejar la primera palabra en minúsculas y la siguiente solo con la inicial en mayúscula sin espacios Ejemplo: primeraSesion.[6]

- Nomenclatura: Es el conjunto de términos empleados en algún tema en específico.

- Clúster: es una tecnología que permite “unir recursos computacionales” para las bases de datos y permite utilizar hardware que no sea muy costoso.[6]

De acuerdo, a los fundamentos teóricos anteriores, se procederá a detallar el tema central del artículo, el cual es la nomenclatura de las bases de datos.

III. NOMENCLATURE OF OBJECTS IN RELATIONAL DATABASES

En esta sección, se hablará del tema central del artículo, teniendo en cuenta los gestores de bases de datos, MySQL y PostgreSQL y sus diferentes objetos para poder aplicar el correcto estándar en su nomenclatura.

• Esquemas.

- El nombre del esquema debe venir precedido del prefijo SCH (SCHEMA_NAME) y seguido por el carácter (_)

- El nombre debe ser igual al nombre de la base de datos PREFIJO. Ejemplo en la Figura 1.

PREFIJO	EJEMPLO
SCH_	[Servidor].[SCH_Plan_Compras].[Base_datos] * [Servidor].[SCH_Seguridad].[Base_datos] * [Servidor].[SCH_Palacio].[Base_datos] * [Servidor].[SCH_SIG].[Base_datos] *

Fig 1: Esquemas. [6]

• Bases de Datos, Nombres y Rutas de archivo

- El nombre de la base de datos debe venir precedido del prefijo BDS (Base de datos) y seguido por el carácter (_)

- El nombre del archivo debe ser igual al nombre de la base de datos. Ejemplo en la Figura 2.

PREFIJO	EJEMPLO
BDS_	[Servidor].[Schema].[BDS_Plan_Compras]* [Servidor].[Schema].[BDS_SIG]*

Fig 2: Rutas.[6]

• Tablas y Vistas

- El nombre de la tabla debe venir precedido del prefijo TBL (Tabla) y seguido por el carácter (_)

- Los nombres de las vistas siguen las mismas convenciones que los nombres de las tablas. Se recomienda utilizar el prefijo “VWS”. Ejemplo en la Figura 3.

PREFIJO	EJEMPLO
TBL_	[Servidor].[Schema].[Base_datos].[TBL_Cursos]* [Servidor].[Schema].[Base_datos].[TBL_Pagos]*
VWS_	[Servidor].[Schema].[Base_datos].[VWS_Personas]*

Fig 3: Tablas y Vistas.[3]

Campos	PREFIJO	EJEMPLO
Campos Clave o Primary Key	PK_ (Primary Key)	PK_PGS_Curso*
Campos clave foránea	FK_ (Foreign Key)	FK_PSN_Municipio*

Fig 5: Índices.[6]

• Campos

El nombre de los campos debe venir precedido del prefijo PK o FK, seguido por el carácter (_)

- Cada nombre de campo debe ser único dentro de su tabla correspondiente. Ejemplo en la Figura 4.
- Cada nombre de campo debe derivarse del identificador utilizado durante el proceso de análisis de negocio.
- No se deben utilizar palabras reservadas como nombres de campos.

Campos	PREFIJO	EJEMPLO
Campos propios	[servidor].[SCH_Plan_Compras].[bds_Plan_Compras].[TBL_Pagos] [servidor].[SCH_Plan_Compras].[bds_Plan_Compras].[TBL_Contabilidad] [servidor].[SCH_Plan_Compras].[bds_Plan_Compras].[TBL_Personas]	PGS_Descripcion* CTB_ValorMoneda* PSN_Observaciones*

Fig 4: Campos.[6]

• Índices

Los índices se nombran considerando la tabla a la que están relacionados y el propósito del índice.

- Las claves primarias utilizan el texto “PK” como sufijo o prefijo, según se considere conveniente.
- Las claves foráneas utilizan el texto “FK” como sufijo o prefijo, según se considere conveniente.
- Los índices agrupados utilizan el sufijo o prefijo “IDX”, según se considere conveniente. Ejemplo Figura 5

• Procedimientos Almacenados y Funciones

- El nombre de los campos deben venir precedido del prefijo SP o FN, seguido por el carácter (_)
- Cada nombre de campo debe ser único dentro de su tabla correspondiente. Ejemplo Figura 6.
- Cada nombre de campo debe derivarse del identificador utilizado durante el proceso de análisis de negocio.

Campos	PREFIJO	EJEMPLO
Store procedures	SP_	SP_ReporteSalidasFuncionarios*
Funciones	FN_	FN_FormatoFecha*

Fig 6: Funciones y Procedimientos.[3]

IV. QUALITY STANDARDS APPLIED TO THE NOMENCLATURE OF THE DATABASES

- NTC ISO 9000:2005 Sistema de Gestión de la Calidad. Fundamentos y Vocabulario. Uso común en nomenclatura de Campos y Tablas.[3]
- NTC ISO 9001:2008 Sistema de Gestión de la Calidad. Requisitos.
- NTC GP 1000:2009 Norma Técnica de Calidad en la Gestión Pública.
- GSE-13 “Estandarización en el Nombramiento de Tablas y Campos en los Modelos de Datos”. Uso común en nomenclatura de Índices.[3]
- GSE-20 “Implementación de Código PL/SQL en la Base de Datos”.

- ISO 19139:2007 MODELO RELACIONAL DE CAPA, TABLAS Y CAMPOS.

Resultado hablar y poner una tabla o grafico que resuma todo.....y de indicios de la métrica.

V. RESULTS

En la actualidad se está desarrollando un proyecto que permita implementar el estándar de calidad sobre la nomenclatura en bases de datos relacionales en diferentes trabajos de desarrollo, ya sea para hacer la respectiva cuantificación de los diversos componentes que se revisaran en cada archivo SQL, en la cual se encuentren desarrollados bajo gestores de bases de datos como MySql y PostgresSQL ya que se están haciendo las primeras pruebas bajo estos gestores, de esta manera se podrá dar a conocer si las ideas realizadas cumplen con el estándar establecido, otorgando una calificación o prioridad a los diferentes componentes que se evaluarán en los diversos archivos de las bases de datos relacionales, ya sea que se encuentren uno o más componentes varias veces en el archivo a calificar se revisara cuantos cumplen con el estándar establecido y posteriormente se multiplicara con la prioridad o valor que tenga cada componente.

Componente	Estandar	Valor
Base de Datos	BDS_	5
Esquemas	SCH_	4
Tablas	TBL_	4
Vistas	VWS_	4
Llaves Primarias	FK_	5
Llaves Foraneas	FK_	5
Campo Descripcion	PGS_	3
Campo ValorMoneda	CTB_	3
Campo Observaciones	PSN_	3

Fig 7: Componentes, Estándar y valor a calificar.

“De esta manera los estudiantes, desarrolladores, arquitectos de software, directores de proyectos se enfoquen en generar

criterios enfocados a la calidad de software durante todo el ciclo de vida del producto tecnológico.”[7]

VI. CONCLUSIONS

- Es fundamental trabajar con un documento de especificación de estándares de calidad, debido a que ciertas métricas en la nomenclatura de las bases de datos lo necesitan como fuente de medición.
- Es necesario tener en cuenta el gestor de base de datos a utilizar, mediante sus características y diferentes aplicaciones que se le puede dar.
- El uso de estándares de calidad, en la especificación y diseño de la base de datos, aumenta la calidad de manera significativa al diagrama entidad-relación, sino que también al software en general.
- El uso del estándar GSE-13, que se ha implementado en los desarrollos de la Universidad de Pamplona, nos da una manera indispensable de trabajar, no solo con un estándar, sino con varios estándares que estén ligados a la calidad de las bases de datos y en el desarrollo de software en general.

REFERENCES

- [1] A. Silberschatz, H. Korth, S. Sudarshan, and F. Pérez, “Clases de aplicaciones de bases de datos,” *Fundam. bases datos*, p. 523, 2002.
- [2] O. Pérez Mora, C. Martín Escofet, M. Gibert Ginestà, D. Costal Costa, L. A. Casillas Santillán, and R. C. Paré, *Bases de datos*. 2005.
- [3] Universidad de Pamplona, “Estandarización en el Nombramiento de Tablas y Campos en los Modelos de Datos,” 2013.
- [4] Power Data, “La Calidad de Datos,” 2012.
- [5] C. de Sanitat, “Servei d’Epidemiologia del Govern de les Illes Balears,” 2015.
- [6] Presidencia de la Republica de Colombia and T. de I. y Comunicaciones, “LINEAMIENTO PARA NOMBRAMIENTO DE BASES DE DATOS Bogotá D.C., Marzo de 2014.” 2014.
- [7] I. Acosta, E. Nieto, and C. Barahona, “Metodología para la evaluación de calidad de los productos software de la Universidad de Cundinamarca,” vol. 3, no. 2.