**ESTANDAR DE NOMENCLATURA DE BASE DE DATOS**

**COPYME**

**[GC]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revisión** | **Fecha** | **Modificación** | **Autor** |
| 00 | 30/08/2014 | Versión inicial | Blanca Coca  Iver Salazar  Diego Antequera  Miguel Hurtado  Javier León |

|  |  |  |
| --- | --- | --- |
| **Elaborado** | **Revisado** | **Aprobado** |
| Firma:  Nombre:  Cargo: | Firma:  Nombre:  Cargo: | Firma:  Nombre:  Cargo: |

**CONTENIDO**

[INTRODUCCION 3](#_Toc397182601)

[Objetivo 3](#_Toc397182602)

[Alcance 3](#_Toc397182603)

[Audiencia 3](#_Toc397182604)

[Fuentes utilizadas 3](#_Toc397182605)

[Fuentes utilizadas Condiciones de uso de este documento 3](#_Toc397182606)

[Convenciones utilizadas en este documento 4](#_Toc397182607)

[Terminología y definiciones 4](#_Toc397182608)

[GUIA RAPIDA 5](#_Toc397182609)

[Convenciones de nomenclatura 5](#_Toc397182610)

[CONVENCIONES DE NOMENCLATURA 6](#_Toc397182611)

[Guías genéricas y buenas prácticas 6](#_Toc397182612)

[Nomenclatura para los elementos de una base de datos 7](#_Toc397182613)

[**La base de datos o schema** 7](#_Toc397182614)

[**Tabla** 7](#_Toc397182615)

[**Vistas** 7](#_Toc397182616)

[**Campos/Columnas** 7](#_Toc397182617)

[**Stored Procedures** 8](#_Toc397182618)

[**Funciones definidas por el usuario** 8](#_Toc397182619)

[**Triggers** 8](#_Toc397182620)

[**Tipos de datos definidos por el usuario** 9](#_Toc397182621)

[**Primary keys** 9](#_Toc397182622)

[**Foreign keys** 9](#_Toc397182623)

[**Indexes** 9](#_Toc397182624)

[**Variables** 10](#_Toc397182625)

# INTRODUCCION

El presente documento describe los estándares para la asignación de nombres a los diversos elementos existentes en una base de datos relacional, independientemente del motor.

## Objetivo

El objetivo de este documento es institucionalizar buenas prácticas y estandarizar la nomenclatura de nombres utilizada en el diseño y mantenimiento de bases de datos en el departamento de desarrollo **COPYME**.

## Alcance

Este documento aplica al diseño y mantenimiento de base de datos en el departamento de desarrollo **COPYME** haciendo foco en el motor de bases de datos: MS SQL Server.

En caso de querer aplicar la nomenclatura para otro manejador de base de datos, distinto de MS SQL Server, se debe decidir si alinearse a la nomenclatura MS SQL Server definidas en este documento en base a factores como:

* Tipo de soporte case sensitive que tenga el manejador y el cliente utilizado.
* La existencia o no de un estándar para dicho manejador

## Audiencia

Este documento se encuentra dirigido a programadores, analistas, jefes de proyecto y especialistas técnicos del departamento de desarrollo de **COPYME**, que tengan entre sus tareas realizar el diseño o mantenimiento de una base de datos.

## Fuentes utilizadas

Entre las fuentes utilizadas para la creación de este documento se encuentran diferentes publicaciones sobre nomenclatura de base de datos, las cuales son referenciadas en la sección de bibliografía, así como también se ha intentado seguir las prácticas utilizadas por Microsoft en el diseño de la base de datos Northwind.

## Fuentes utilizadas Condiciones de uso de este documento

Una regla puede romperse sólo ante razones justificadas, discutidas, con previa autorización del responsable del producto, y en caso que no pueda aplicarse ninguna alternativa razonable. El autor de la excepción, obligatoriamente debe documentar el código explicando la causa de la violación de la regla.

Las preferencias personales no se consideran una razón justificada.

## Convenciones utilizadas en este documento

|  |  |
| --- | --- |
| Abreviaciones | Descripción |
| OBL | Obligatorio |
| REC | Recomendado |
| **Negrita** | Texto con énfasis adicional que debe ser considerado importante. |
| Siempre | Indica que esta regla DEBE ser respetada, en los términos de este manual. |
| *Nunca* | Indica que esta acción NO DEBE ser realizada, en los términos de este manual. |
| *No* *hacer* | Indica que esta acción NO DEBE ser realizada, en los términos de este manual. |
| *Evitar* | Indica que esta práctica debe ser evitada siempre que sea posible, pero pueden existir excepciones AUTORIZADAS para su utilización. |
| *Intentar* | Indica que esta práctica debe aplicarse siempre que sea posible y apropiado. |
| Razón | Explica el propósito y las causas que motivan la regla o recomendación. |

## Terminología y definiciones

|  |  |
| --- | --- |
| Término | Descripción |
| Camel Case | Una palabra con la primera letra en minúsculas, y la primera letra de cada una de las palabras subsecuentes en mayúsculas.  Ejemplo: customerName |
| Magic Number | Cualquier literal numérico utilizado dentro de una expresión (o inicialización de variable) que no posea un significado claro. Usualmente este término no aplica a los valores 0 y 1 y cualquier otra expresión numérica equivalente que su evaluación resulte 0. |
| Pascal Case | Una palabra con la primera letra en mayúsculas, y la primera letra de cada palabra subsecuente también en mayúsculas.  Ejemplo: CustomerName |
| Hungarian Notation | Comienzan con una o mas letras en minúscula que denotan el tipo de la variable  Ejemplo: string sVariable |
| Underscore Separated | Indica palabras separadas con infraguión. Ejemplo: CUSTOMER\_DETAIL |

# GUIA RAPIDA

En esta sección se incluye un breve resumen de los principales estándares descriptos a los largo de este documento. Estas tablas no son detalladas en sus descripciones, pero brindan una rápida referencia a los elementos.

## Convenciones de nomenclatura

|  |  |
| --- | --- |
| c | Camel case |
| P | Pascal case |
| \_ | Prefijo con infraguión (underscore) |
| X | No aplica |
| [] | Lo se encuentre contenido entre paréntesis rectos significa que es opcional. |
| <VAR> | Indica que esa posición debe sustituirse por el valor del campo VAR. En el caso de la variable TABLE se hace la siguiente distinción: TABLE\_S representa el nombre de una tabla en singular (ej.: Customer), mientras que TABLE\_P indica el nombre de una tabla en plural (ej.: Customers). |
| USU | Underscore Separated Upper Case |

|  |  |  |
| --- | --- | --- |
| **Elemento** | **MS SQL Server** | **Observaciones** |
| **Base de datos** | <COUNTRY>\_<CUSTOMER>\_<SOLUTION>[-AUX] | ***Ejemplo: UY\_COPYME\_Northwind*** |
| **Schema** | <COUNTRY>\_<CUSTOMER>\_<SOLUTION>[-AUX] | ***Ejemplo: UY\_COPYME\_Northwind*** |
| **Tablas** | P & plural | Evitar espacios en blanco ***Ejemplo: Customers*** |
| **Vistas** | VW\_<VIEW\_P> | Evitar espacios en blanco ***Ejemplo SQL Server: vw\_SalesByCountry*** |
| **Procedimientos Almacenados** | P | Evitar el uso de prefijos, tipo sp\_ y espacios en blanco ***Ejemplos:  InsertCustomer, GetOrdersByDate*** |
| **Funciones definidas por el ususario** | P |  |
| **Triggers** | <TABLE\_S>\_<OPERATION>[\_<AUX>] | Un trigger está siempre asociado con una tabla y una operación y no tiene sentido fuera de ellos. ***Ejemplos: Orders\_Insert\_ValidateData, Customer\_Insert\_ReplicateEmail*** |
| **Columns** | P  Para las claves <TABLE\_S>Id | No nombrar de forma distinta campos que representen lo mismo. ***Ejemplos:  OrderId, FullName, Address, OrderDate*** |
| **Tipos de Datos definidos por el usuario**  **(data types)** | C | ***Ejemplo: customerId*** |
| **Primary keys** | PK\_<TABLE\_P> | ***Ejemplo: PK\_Customers*** |
| **Foreign keys** | FK\_<TABLE\_P><FIELD>\_<REF\_TABLE\_P><REF\_FIELD> | ***Ejemplo: FK\_OrdersCustomerId\_CustomersCustomerId*** |
| **Indexes** | [IDX\_]<TABLE\_P>\_<FIELD>[\_AUX] | ***Ejemplo: OrderDetails\_OrderID\_U\_NC*** En el ejemplo presentado \_U correspondería a Unique y \_NC correspondería a NonClustered. |
| **Variables** | c |  |

# CONVENCIONES DE NOMENCLATURA

A continuación se presentan un conjunto de guías y buenas prácticas, así como la nomenclatura para utilizar en el diseño de bases de datos.

## Guías genéricas y buenas prácticas

1. **OBL** – **Utilizar nombres en inglés para todos los elementos de la base de datos**, tablas, vistas, campos, etc.
2. **REC** – **Utilizar nombres descriptivos para los campos**. Utilizar nombres que resulten intuitivos y permitan entender el significado de los campos (mnemotécnicos). Evitar las abreviaciones, y si esto no es posible documentarlas bien.
3. **REC** – **No nombrar campos que representan lo mismo de forma distinta**. La forma en que se nombran iguales propiedades debe ser consistente en todo un esquema. Ejemplo: Nombrar al campo clave de la tabla Customers como Id, y después referenciarlo en otras tablas como CustomerId es INCORRECTO. El campo debe ser nombrado CustomerId en todos los casos que se quiera almacenar una clave de Customers.
4. **REC** – **Evitar tener demasiadas columnas NULLABLES en una tabla**. Esto es indicio de un esquema poco o nada normalizado. Falta de normalización puede conllevar problemas de consistencia en los datos en la medida que un mismo campo se puede terminar almacenando en varias tablas. Excesiva normalización puede tener asociada una pérdida de performance en ciertas operaciones sobre la base de datos. Es necesario encontrar el equilibrio correspondiente a los requerimientos de cada proyecto en este punto. Como regla general la tercera forma normal es un buen punto intermedio.
5. **REC** – **Evitar tener tablas sin definición de primary keys**.
6. **REC** – **Evitar tener tablas innecesarias en el sistema**. Un buen diseño es uno simple (keep it simple ;)
7. **REC** – **Intentar evitar el uso de código propietario** en la definición de expresiones SQL.. Intentar utilizar código Standard SQL-92.

## Nomenclatura para los elementos de una base de datos

En esta sección se presenta la nomenclatura definida para los distintos elementos de una base de datos.

### **La base de datos o schema**

La base de datos SQL Server o los schemas deberán nombrarse usando la siguiente nomenclatura:

<PAIS>\_<CLIENTE>\_<SOLUCION>[-AUX]

Donde se reserva AUX para diferenciar dos bases de datos o schemas correspondientes a una misma solución.

Ejemplo:

UY\_COPYME\_Northwind

UY\_COPYME\_ICWorkflow-ICCM

### **Tabla**

Las tablas deben nombrarse:

* en plural,
* en inglés
* sin utilizar espacios en blanco
* Si el nombre es compuesto solo la última palabra debe ir en plural. Por ejemplo: ProductSales es correcto mientras que Product**s**Sales **NO** es correcto.

**MS SQL Server**

Deben nombrarse usando notación pascal.

Ejemplo: **Customers**, **Orders**

### **Vistas**

Las vistas deben nombrarse con la misma notación definida para nombrar tablas, pero prefijadas usando **VW\_**.

Ejemplo:

MS SQL Server: vw\_SalesByCountry

### **Campos/Columnas**

Los campos de una tabla corresponden a los atributos de una entidad, describen propiedades de la misma.

Las columnas deben ser nombradas según los lineamientos a continuación:

1. Los nombres deben ser simples, representativos e intuitivos.
2. Los nombres de las columnas de una tabla deben estar expresados **en singular**.
3. El campo **clave** de una tabla de nombrarse como el nombre de la tabla más el sufijo **Id**. Ejemplo: Para una tabla de clientes Customers, se definirían las claves:
   * MS SQL Server: CustomerId,
4. Campos que representen la misma entidad del mundo real, deben estar nombrados de la misma manera en todas las tablas de un esquema. Por ejemplo nombrar la clave de la tabla *Sales* en una tabla como *SalesId* y en otra *SalesKey* es incorrecto.
5. Se desaconseja prefijar sistemáticamente TODOS los campos de una tabla con el nombre de la tabla o una abreviación del mismo. Entendemos que esto agrega un nivel de redundancia y complejidad al sistema que no es necesario en manejadores modernos.

**MS SQL SERVER**

Usar notación Pascal

### **Stored Procedures**

Los stored procedures son un espacio estándar para incluir lógica en la base de datos, expresada en un lenguaje de scripting que extiende SQL. Los SP pueden ser invocados utilizando SQL estándar desde una aplicación, mediante la instrucción EXEC.

Los stored procedures deben ser nombrados según la siguiente nomenclatura:

**MS SQL SERVER**

Usar notación Pascal

Ejemplo: InsertCustomer

El código SQL extendido de un stored procedure debe ir en Mayúsculas. Ver apéndice C como ejemplo.

### **Funciones definidas por el usuario**

Las funciones definidas por el usuario son un mecanismo no totalmente estándar para incluir lógica en la base de datos, expresada en un lenguaje de scripting que extiende SQL.

La nomenclatura definida es la misma que para los stored procedures.

### **Triggers**

Un trigger es lógica alojada en la base de datos asociada a una determinada acción sobre una tabla. La lógica es disparada cuando ocurre la acción correspondiente.

Un trigger no tiene sentido fuera de una tabla y un trigger tiene asociada siempre una operación, por lo que dicha información debe estar asociada al nombre del trigger.

<TABLA>\_<OPERACION>[\_AUX]

Ejemplo: Customer\_Insert\_InsertInUsers

### **Tipos de datos definidos por el usuario**

Los tipos de datos definidos por el usuario son un mecanismo para mantener la consistencia de tipos en la base de datos. Cuando un mismo tipo de datos es utilizado en varias tablas, en vez de definirlo cada vez por separado, se define un “user defined data type” para luego referenciarlo desde todas ellas y mantener así centralizada su definición.

**MS SQL SERVER**

Usar notación Pascal

Ejemplo: CustomerId

### **Primary keys**

La clave primaria es un conjunto de campos que identifica de forma única un registro en una tabla. Son un caso particular de un índice. La nomenclatura es la siguiente:

PK\_<TABLA>

Ejemplo: PK\_Customers

### **Foreign keys**

Las foreign keys son usadas para definir vínculos entre tablas relacionadas. Una foreign key establece una relación entre una o más columnas de una tabla y la clave primaria de la tabla referenciada. Como patrón para la nomenclatura de la foreign key elegimos el siguiente.

FK\_<TABLA\_QUE\_REFERENCIA>+<CAMPO\_QUE\_REFERENCIA>\_<TABLA\_REFERENCIADA>+<CAMPO\_REFERENCIADO>

Basado en el patrón dado, voy a nombrar la foreign key que referencia el campo CUSTOMER\_ID de la tabla CUSTOMERS desde la tabla ORDERS y el campo CUSTOMER\_ID como:

FK\_ORDERSCUSTOMERID\_CUSTOMERSCUSTOMERID

### **Indexes**

Los índices son un mecanismo para aumentar la eficiencia de localización y acceso de un registro en una tabla en la base de datos, opcionalmente asegurando unicidad de los valores del índice. La definición de índices tiene un impacto positivo en los tiempos de consulta de registro y uno negativo en los de inserción y actualización de los campos del índice.

Los índices están asociados a una tabla y a un conjunto de campos de la tabla, a su vez pueden ser únicos o no y pueden estar definidos en cluster o no. La nomenclatura elegida para nombrarlos es la siguiente:

[IDX\_]<TABLA>\_<CAMPO>[\_AUX]

Prefijar el índice es opcional, pero de hacerlo se debe usar el prefijo especificado.

Ejemplo: OrderDetails\_OrderId\_U\_NC

El ejemplo corresponde a un índice definido sobre la tabla OrderDetails, sobre el campo OrderId, unico y nonclustered.

### **Variables**

Cuando las variables corresponden columnas de una tabla, deben ser nombrados de la misma manera que la columna. La notación elegida para definir las variables es camel. Ver apéndice C como ejemplo.