

### **CONVENCIÓN DE NOMENCLATURA PARA OBJETOS DE UNA BASE DATOS**

El objetivo principal de una convención de nomenclatura para objetos de una base datos es poder identificar fácilmente el tipo y la finalidad de todos esos objetos. A continuación se presenta una guía que debe seguir al nombrar a los objetos de base de datos. Tenga presente que la regla más importante es nombrar consistentemente a los objetos. Aún siguiendo esta y otras pautas es posible todavía producir nombres largos y crípticos; en última instancia, la situación particular en que se encuentre dictará la razonabilidad de su convención de nombres. El objetivo de esta convención es producir nombres prácticos, legibles, concisos, inequívocos y coherentes.

Se presentarán convenciones para los siguientes tipos de objetos de una base de datos:

1. Tablas
2. Columnas
3. Índices
4. Restricciones
5. Vistas
6. Procedimientos almacenados
7. Disparadores (triggers)

Consideraciones generales para todos los nombres:

- Limitarlos a 30 caracteres (más corto es mejor)
- Usar letras o caracteres de subrayado (tratar de evitar los dígitos)
- Tratar de utilizar caracteres de subrayado lo menos posible. La notación PascalCase logra la misma separación de palabra sin requerir subrayados.
- Utilice una letra como primer carácter del nombre (no comience los nombres con subrayado).
- Evitar las abreviaturas (puede dar lugar a una interpretación errónea de los nombres) .
- Evitar las siglas (algunos acrónimos tienen más de un significado por ejemplo. "ASP")
- El nombre debe ser legible (no deben sonar gracioso cuando se lee en voz alta) .
- Evitar el uso de espacios en blanco, incluso si el sistema lo permite.

## 1. Tablas

### Plurales

Los nombres de las tablas deben ser plurales: CLIENTES en lugar de CLIENTE. Las tablas son colecciones lógicas de uno o más registros. Si el nombre viene en singular en el modelo conceptual cambiarlo a plural en el modelo relacional. Si el nombre de la tabla incluye varias palabras, en lo posible reordenar el nombre para poner en plural solo la última: UsuarioRoles en lugar de RolesUsuario o RolesUsuarios.

### Prefijos

Si se usan correctamente permiten organizar tablas en grupos relacionados. Si no se usan correctamente obligan a escribir caracteres innecesarios. No usar prefijos como "tbl" o "TBL". No usar subrayado con los prefijos.

### Notación

Usar PascalCase: Usar una mayúscula para separar las partes de un nombre multi-palabra; las demás letras van en minúscula.

### Caracteres especiales

No deben usarse. Aún si el sistema los permite, obliga entonces a agregar caracteres adicionales para delimitar los nombres que los usan. Por ejemplo, [nombre tabla]. Evitar caracteres particulares a otros lenguajes aparte del inglés: tildes, eñes, etc.

### Abreviaturas

No usarlas en lo posible. Usar "Horas" en lugar de "Hrs". Puede que no todos comprendan la abreviatura.

### Tablas para relaciones muchos-a-muchos

Se nombran concatenando los nombres de las tablas que participan en la relación. Por ejemplo, se tienen tablas Doctores y Pacientes; un doctor puede tener varios pacientes y un paciente puede tener varios doctores; se requiere una tabla explícita para llevar dicha relación. Dicha tabla debe llamarse DoctoresPacientes o DoctoresxPacientes.

Como esta convención puede producir nombres largos, se pueden usar abreviaturas a su discreción.

## 2. Columnas

Como las columnas son parte de una tabla, no se requiere mencionar el nombre de la tabla en el nombre de la columna. La llave primaria es típicamente la única excepción a esta regla; se puede incluir el nombre de la tabla con el fin de tener un nombre más descriptivo que simplemente "Id": IdCliente es aceptable pero no requerido.

Como en el caso de las tablas, evitar abreviaturas, siglas y caracteres especiales. Usar PascalCase.

### Campos de llave primaria

Usar simplemente "Id". Solo en casos calificados agregar el nombre de la tabla "IdCliente". Esto evita repetir el nombre de la tabla en operaciones DML:

```
Select Cliente.Nombre, Orden.Id
FROM Cliente, Orden
WHERE Cliente.Id = Orden.IdCliente
```

### Campos de llaves foráneas

Deberán tener el mismo nombre del campo de llave primaria al que apuntan, con una modificación. Se debe incluir el nombre de la tabla. Por ejemplo, en la tabla Cliente su llave primaria es "Id". En la tabla Orden la cual incluye un campo para cliente, ese campo se llamará "IdCliente".

Si hay más de un campo de llave foránea a la misma tabla, se debe agregar un descriptor que permita distinguirlos. Por ejemplo, se tiene una tabla Direcciones con llave primaria "Id"; en la tabla Cliente se tienen varias llaves foráneas a diferentes direcciones del cliente: "IdDireccionCasa", "IdDireccionTrabajo", "IdDireccionPostal", "IdDireccionEntrega".

### Llaves compuestas

Si una tabla tiene llaves compuestas, entonces en lugar de simplemente "Id", se usará un descriptor antes del "Id". Por ejemplo "ModuloId" y "CodigoId" pueden ser los nombres de una llave compuesta (Modulo, Codigo).

### Prefijos

No usar prefijos como "fld" o "col". Incluir un prefijo con dos o tres caracteres para señalar el tipo de datos no es recomendado.

### Nombres para tipos de datos específicos

Los campos booleanos deben tener nombres como "EstaBorrado", "TienePermiso", "EsValido" de modo que el significado del dato en el campo no sea ambiguo. Si el campo almacena fechas u horas debe incluir "Fecha"/"Hora" en alguna parte del nombre. A veces es apropiado agregar la unidad de tiempo al nombre, especialmente si el campo almacena números enteros. Campos así deben nombrarse como "HorasEjecucion", "MinutosComprados".

### 3. Índices

Dado que los índices están siempre relacionados con una tabla o vista, tiene más sentido incluir el nombre de la tabla o la vista en el nombre del índice. También se incluyen las columnas indexadas, junto con algunos caracteres que especifican el tipo del índice. Esta convención de nomenclatura también le permite, si se busca a una lista de índices, para ver los índices ordenados por tabla, columna y tipo de índice.

#### Estructura de los nombres de índices

<Nombre tabla> <Columnas indexadas> (U|N) (C|N)

U|N: único | no único

C|N: clustered | non-clustered

No es necesario agregar caracteres como "idx".

Se concatenan los nombres de las columnas que participan en el índice.

Ejemplos:

ProductosIdUC:

índice único y clustered sobre la columna Id de la tabla Productos.

OrdenDetallesIdOrdenIdClienteNN:

índice no único y no clustered sobre las columnas IdOrden e IdCliente de la tabla OrdenDetalles

Como este último nombre es largo se podría abreviar un poco: OrdenDetallesIdOrdIdCliNN

#### Prefijos

Evitar poner prefijos o sufijos como "idx" o "IDX" antes de los índices. La estructura vista anteriormente identifica claramente a los índices.

### 4. Restricciones

Las restricciones se refieren a las columnas, de modo que el nombre de la columna debe ser incluido en el nombre de la restricción. También se debe incluir el tipo de restricción (Check, Referential Integrity, Primary Key, Unique) y el nombre de la tabla.

#### Estructura de los nombres de restricciones

<tipo restricción> <nombre tabla>\_<nombre columna>

Por ejemplo:

PkProductos\_Id: restricción de llave primaria en el campo Id de la tabla Productos.

FkOrdenes\_IdProducto: restricción de llave foránea en el campo IdProducto de la tabla Ordenes.

CkClientes\_IdRepresentante: restricción check sobre el campo IdRepresentante de la tabla Clientes.

Se usan subrayados para separar el nombre de la tabla del nombre de la columna y evitar posibles confusiones sobre dónde termina uno y empieza el otro.

### Prefijos

Usar los siguientes prefijos en los nombres de restricciones:

Pk: llave primaria

Fk: llave foránea

Ck: check

Un: único

## **5. Vistas**

Seguir las mismas reglas que para los nombres de tablas, con las diferencias que se detallan a continuación

### Prefijos

Es útil usar prefijos con vistas con el fin de aclarar que no se trata de tablas. Usar al menos dos caracteres en el prefijo y no usar simplemente "V". Por ejemplo, "Vw", "View", "Vi".

### Tipos de vistas

El nombre de las vistas debe variar dependiendo el tipo o el propósito de las vistas.

- Para vistas que simplemente combinan una o más tablas sin criterios de selección, combinar el nombre de las tablas: una vista que combina la tabla Provincias con la tabla Clientes por criterio geográfico se puede llamar VwClientesProvincias.
- Para vistas usadas para generar datos para reportes con valores específicos en la cláusula WHERE, reflejar hasta donde se pueda las condiciones. Por ejemplo: VwVentasDel2014.

## **6. Procedimientos almacenados**

Los procedimientos almacenados no están lógicamente atados a una tabla o columna. Típicamente realizan una de las operaciones CRUD (Create, Read, Update, Delete) sobre una tabla, o alguna otra acción. Puesto que los procedimientos almacenados siempre realizan algún tipo de operación el nombre debe describir la operación que realiza.

Usar un verbo para describir el tipo de operación, seguido de las tablas sobre las que se realiza la operación.

### Prefijos y sufijos

La forma en que nombre los procedimientos almacenados ayuda a agruparlos cuando son listados.

- Si se desean agrupar por tipo de operación CRUD, entonces usar prefijos "Create", "Get", "Update", o "Delete" en el nombre. Por ejemplo, CreateProductos o CreateOrdenes.
- Si se desean agrupar por una tabla sobre la que operan, usar las frases anteriores como sufijos en lugar de prefijos. Por ejemplo, ProductosCreate, OrdenesCreate.
- Si el procedimiento retorna un valor escalar o si hace operaciones como validaciones, no usar los prefijos o sufijos anteriores. Usar una combinación verbo-sustantivo. Por ejemplo, ValidarLogin.

#### Prefijos de agrupamiento

Si se tienen muchos procedimientos almacenados, se puede considerar usar un prefijo que los agrupe e identifique como parte de una aplicación. Por ejemplo, usar "Rh" como prefijo para agrupar todos los procedimientos relacionados con recursos humanos. Por ejemplo, RhCreateSolicitudVacacion.

#### Malos prefijos

No usar como prefijos cualquier frase que confunda al sistema y lo considere un procedimiento del sistema. Por ejemplo, en SQL Server no empezar un procedimiento con "sp\_", "xp\_" o "dt\_".

### **7. Disparadores (triggers)**

Aunque los triggers tienen muchas cosas en común con los procedimientos almacenados, tienen dos diferencias fundamentales:

- i. Los triggers no existen independientemente de una tabla. De modo que se debe incluir el nombre de la tabla en el nombre del trigger.
- ii. Los triggers son ejecutados únicamente cuando se lleva a cabo operaciones de Insert, Update o Delete en uno o más tuples de su tabla. Así que también se debe incluir en el nombre el tipo de acción que disparó al trigger.

#### Prefijos y sufijos

Usar "Trg" como prefijo o sufijo. Por ejemplo, Trg\_ProductosIns, ProductosInsTrg, Productos\_InsTrg, InsProductos\_Trge. Escoger una opción y mantener la consistencia.

#### Operaciones múltiples

Si un trigger maneja más de una operación (por ejemplo INSERT y UPDATE), entonces incluir ambas operaciones en el nombre. Por ejemplo, Productos\_InsUpdTrg o TrgProductos\_UpdDel.

### Triggers múltiples

Algunos sistemas permiten que varios triggers se disparen para una misma operación sobre una tabla.

En este caso, los nombres deben permitir poder distinguirlos fácilmente. Por ejemplo,

"Usuarios\_ValidarEmail\_InsTrg", "Usuarios\_CrearEntradas\_InsTrg".