Tabla de contenido

[Estándar de nomenclatura para Base de Datos 1](#_Toc432590880)

[Objetivo 4](#_Toc432590881)

[Alcance 4](#_Toc432590882)

[Audiencia 5](#_Toc432590883)

[Fuentes utilizadas 5](#_Toc432590884)

[Condiciones de uso de este documento 5](#_Toc432590885)

[Convenciones utilizadas en este documento 6](#_Toc432590886)

[Terminología y definiciones 7](#_Toc432590887)

[Guía rápida 8](#_Toc432590888)

[Convenciones de nomenclatura 8](#_Toc432590889)

[Convenciones de Nomenclatura 10](#_Toc432590890)

[Guías genéricas y buenas prácticas 10](#_Toc432590891)

[Nomenclatura para los elementos de una base de datos 11](#_Toc432590892)

[Schemas de Aplicacion 11](#_Toc432590893)

[Schemas de Usuario 12](#_Toc432590894)

[Tablas 12](#_Toc432590895)

[Nombres Cortos de Tablas 13](#_Toc432590896)

[Vistas 13](#_Toc432590897)

[Vistas Materializadas 13](#_Toc432590898)

[Columns 14](#_Toc432590899)

[Primary keys 15](#_Toc432590900)

[Foreign keys 15](#_Toc432590901)

[Unique keys 15](#_Toc432590902)

[Check keys 16](#_Toc432590903)

[Indexes 16](#_Toc432590904)

[Sequences 17](#_Toc432590905)

[Synonyms 17](#_Toc432590906)

[Stored Procedures 18](#_Toc432590907)

[Functions 18](#_Toc432590908)

[Triggers 19](#_Toc432590909)

[Tipos de datos definidos por el usuario 19](#_Toc432590910)

[Packages 20](#_Toc432590911)

Introducción

El presente documento describe la nomenclatura a utilizar en el diseño de base de datos Oracle en el Centro de Soluciones Comerciales Belcorp.

## 

## 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 Oracle en el Centro de Soluciones Comerciales Belcorp.

## 

## Alcance

Este documento aplica al diseño y mantenimiento de base de datos en el en el Centro de Soluciones Comerciales Belcorp haciendo foco solo en el manejador de bases de datos Oracle.

El documento debe entenderse como una propuesta de nomenclatura estándar que sigue los patrones más comúnmente utilizados, lo que significa que puede estar sujeta a modificaciones derivadas de las necesidades de BELCORP o de peculiaridades propias de cada Sistema. El ejerce influencia en los siguientes procesos:

* La Realización de la Construcción de un Sistema, en cuanto utiliza objetos de la Base
* El Acceso a los distintos objetos de la Base de Datos.
* La Migración de Datos.

## 

## Audiencia

Este documento se encuentra dirigido a programadores, analistas, jefes de proyecto y especialistas técnicos del Centro de Soluciones Comerciales Belcorp, 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.

## 

## 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 |

# Guía rápida

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 siguente 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** | **Nomenclatura** | **Observaciones** | | SCHEMAS DE APLICACION | ESQUEMA COMUN:  <SISTEMA>\_COMUN ESQUEMA PAIS MARCA:  < SISTEMA >\_<PAIS>\_<MARCA> | Ejemplo:  OWN\_COMUN, SSICC\_COMUN  OWN\_PE\_ES, SSICC\_CO\_ES | | SCHEMAS DE USUARIO | USU\_<USUARIO> | Ejemplo:  USU\_MONITOR, USU\_LECT\_PE\_ES | | TABLAS | <MODULO>\_<ABREVIATURA NOMBRE DE TABLA> | Ejemplo:  BAS\_PROCE\_BATCH, MAE\_CLIEN | | NOMBRES CORTOS DE TABLAS | <MODULO>\_<NOMBRE CORTO DE LA TABLA> | Ejemplo:  Tabla: STO\_DOCUM\_DIGIT  Nombre corto: STO\_DODI | | VISTAS | V\_<NOTACION DE TABLAS> | Ejemplo:  V\_MAE\_CLIE\_UNIDA\_ADMIN | | VISTAS MATERIALIZADAS | <MODULO>\_MV\_<NOTACION DE TABLAS> | Ejemplo:  SOA\_VW\_REGIO, SOA\_MV\_CAMPA | | COLUMNS | <PREFIJO USO>\_<NOMBRE DE CAMPO> | Ejemplo:  OID\_CLIE, COD\_CLIE, IND\_FICH\_INSC, FEC\_INGR, IND\_PROL, USU\_MODI, VAL\_RECL\_PEND | | PRIMARY KEYS | <MODULO>\_<NOMBRE CORTO DE LA TABLA> \_PK | Ejemplo:  STO\_DODI\_PK (PK de la table STO\_DOCUM\_DIGIT) | | FOREIGN KEYS | <MODULO>\_<NOMBRE CORTO DE LA TABLA> \_< NOMBRE CORTO DE LA TABLA QUE REFERENCIA>\_PK | Ejemplo:  STO\_DODI\_TIDO\_FK (FK de la tabla STO\_DOCUM\_DIGIT a la tabla STO\_TIPO\_DOCUM\_DIGIT) | | UNIQUE KEYS | <MODULO>\_<NOMBRE CORTO DE LA TABLA> \_< PREFIJO COLUMNAS UNICAS>\_ UK | Ejemplo:  MAE\_CLIE\_UK | | CHECK KEYS | <NOMBRE CORTO DE LA TABLA> \_< PREFIJO COLUMNAS A VALIDAR>\_ CK | Ejemplo:  SOA\_DODI\_FEDI\_CK | | INDEXES | <NOMBRE CORTO DE LA TABLA> \_< PREFIJO CAMPOS INDEXADOS>\_ IDX | Ejemplo: STO\_DODI\_FEDI\_IDX | | SEQUENCES | <MODULO>\_<NOMBRE DEL SEQUENCE> \_SEQ | Ejemplo:  STO\_DODI\_SEQ (Secuencia de la table STO\_DOCUM\_DIGIT) | | SYNONYMS | Son objetos de base de datos utilizados para hacer referencia a objetos que se encuentran en otros esquemas sin utilizar el esquema. | Ejemplo: SSICC\_PE\_ES.MAE\_CLIEN (Sinónimo de la tabla OWN\_PE\_ES.MAE\_CLIEN) | | STORED PROCEDURES | <MODULO>\_PR \_<NOMBRE DEL PROCEDIMIENTO> | Ejemplo: MAE\_PR\_PROCE\_EJEMP | | FUNCTIONS | <MODULO>\_FN \_<NOMBRE DE LA FUNCION> | Ejemplo: MAE\_FN\_FUNCT\_EJEMP | | TRIGGERS | <MODULO>\_TRG \_<OPERACION>\_<NOMBRE CORTO DE LA TABLA> | Ejemplo: BAS\_TRG\_AI\_AUDI (Es un trigger que se ejecuta en las tablas de auditoria After Insert) | | TIPOS DE DATOS DEFINIDOS POR EL USUARIO | Ejemplo: T\_<IDENTIFICADOR DE TIPO> | T\_ESTRU\_ARCHI | | PACKAGES | <MODULO>\_PKG\_<NOMBRE DE PAQUETE> | Ejemplo:  IMP\_PKG\_PROCE\_ELECT (PAQUETE de procesos de impression Electronica) | |

# 

# 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).
3. **OBL**- **utilizar solo mayúsculas para nombrar los elementos de la base de datos, schemas, tablas y campos.**
4. **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.
5. **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 perdida 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.
6. **REC** – **Evitar tener tablas sin definición de primary keys**.
7. **REC** – **Evitar tener tablas innecesarias en el sistema**. Un buen diseño es uno simple (keep it simple ;)
8. **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.

### Schemas de Aplicacion

Son esquemas creados para los sistemas que accederán a la base de datos**,** de acuerdo a su uso se ha definido el siguiente formato:

#### Esquema COMUN

Son esquemas que tienen información corporativa que será replicada a otros países.

Debe mantener la siguiente estructura:

<SISTEMA>\_COMUN

Donde SISTEMA es el identificador del sistema que está accediendo a la base de datos.

Ejemplo:

**OWN\_COMUN, SSICC\_COMUN**

#### Esquema PAIS MARCA

Son esquemas que tienen información propia del país.

Debe mantener la siguiente estructura:

< SISTEMA >\_<PAIS>\_<MARCA>

Donde SISTEMA es el prefijo o identificador del sistema que está accediendo a la base de datos.

Ejemplo:

**OWN\_PE\_ES, SSICC\_CO\_ES**

### Schemas de Usuario

Son esquemas que tienen como fin la consulta/modificación directa a base de datos, sujeta a las políticas de la empresa.

Debe mantener la siguiente estructura:

USU\_<USUARIO>

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

Ejemplo:

**USU\_MONITOR, USU\_LECT\_PE\_ES**

### 

### Tablas

Son las entidades que guardan información, deben de mantener el siguiente estándar.

* Deben nombrarse con notación Underscore Separated, en mayúsculas.
* Sin utilizar espacios en blanco
* Palabras o abreviaturas de a lo sumo cinco caracteres de longitud separados por el carácter de Underscore.
* Deben Mantener la siguiente estructura:

<MODULO>\_<ABREVIATURA NOMBRE DE TABLA>

Ejemplos:

**BAS\_PROCE\_BATCH**, **MAE\_CLIEN**

### Nombres Cortos de Tablas

Son prefijos que identifican una tabla para que puedan ser referenciados y permitan una rápida identificación de la tabla o sus relaciones.

Debe mantener la siguiente estructura:

<MODULO>\_<NOMBRE CORTO DE LA TABLA>

El nombre corto no debe tener más de 4 caracteres,

Ejemplo:

Tabla: **STO\_DOCUM\_DIGIT** Nombre corto: **STO\_DODI**

### Vistas

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

Ejemplo:

**V\_MAE\_CLIE\_UNIDA\_ADMIN**

### Vistas Materializadas

Las vistas materializadas deben nombrarse con la misma notación definida para nombrar tablas, pero debe agregarse el prefijo \_MV después del Módulo de sistema.

Ejemplo:

**SOA\_VW\_REGIO, SOA\_MV\_CAMPA**

### Columns

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. Su longitud total no puede superar 25 caracteres
3. Los nombres de las columnas de una tabla deben estar formados por palabras o abreviaturas de no más de 4 caracteres separadas con el carácter Underscore.
4. Las Columnas que sean claves identificadoras de una tabla generalmente tendrán un prefijo tipo COD o OID.
5. El primer prefijo debe tener 3 caracteres y debe indicar el uso dado, ver tabla adjunta.

|  |  |
| --- | --- |
| **Prefijo** | **Descripcion** |
| COD | Codigo |
| VAL | Valor |
| DES | Descripcion |
| OID | Id o clave subrogada |
| FEC | Fecha |
| USU | Usuario |

1. Debe mantener la siguiente estructura:

<PREFIJO USO>\_<NOMBRE DE CAMPO>

Ejemplos:

**OID\_CLIE, COD\_CLIE, IND\_FICH\_INSC, FEC\_INGR, IND\_PROL, USU\_MODI, VAL\_RECL\_PEND**

### 

### 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:

<MODULO>\_<NOMBRE CORTO DE LA TABLA> \_PK

Ejemplo:

**STO\_DODI\_PK (PK de la table STO\_DOCUM\_DIGIT)**

### 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.

<MODULO>\_<NOMBRE CORTO DE LA TABLA> \_< NOMBRE CORTO DE LA TABLA QUE REFERENCIA>\_PK

Ejemplo:

**STO\_DODI\_TIDO\_FK (FK de la tabla STO\_DOCUM\_DIGIT a la tabla STO\_TIPO\_DOCUM\_DIGIT)**

### Unique keys

Los Unique keys son usadas para definir campos que serán validados por la base de datos como únicos en la tabla. Como patrón para la nomenclatura de la foreign key elegimos el siguiente.

<MODULO>\_<NOMBRE CORTO DE LA TABLA> \_< PREFIJO COLUMNAS UNICAS>\_ UK

Ejemplo:

**MAE\_CLIE\_UK**

***Esta nomenclatura aplica para todos casos a excepción de los Unique Indexes generados automáticamente por la base de datos cuando se crean Primary Keys.***

### Check keys

Los Check keys son usadas para definir campos que serán validados por la base de datos como formatos.

Como patrón para la nomenclatura de la foreign key elegimos el siguiente.

<NOMBRE CORTO DE LA TABLA> \_< PREFIJO COLUMNAS A VALIDAR>\_ CK

Ejemplo:

SOA\_DODI\_FEDI\_CK (Check Constraint que valida que la fecha tenga un formato definido)

### 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.

La nomenclatura elegida para nombrarlos es la siguiente:

<NOMBRE CORTO DE LA TABLA> \_< PREFIJO CAMPOS INDEXADOS>\_ IDX

Ejemplo:

STO\_DODI\_FEDI\_IDX

***Esta nomenclatura aplica para todos casos a excepción de los índices generados automáticamente por la base de datos cuando se crean Primary Keys y Unique Indexes.***

### Sequences

Son objetos de base de datos utilizados para generar números de secuencia y son muy comúnmente usados para la creación de valores únicos de los Primary Keys en modelos con claves subrogadas.

<MODULO>\_<NOMBRE DEL SEQUENCE> \_SEQ

**En el caso en que el sequence sea utilizado para generar secuencias para el primary keys el nombre del sequence debe ser igual al nombre corto de la tabla para identificar su relación y uso.**

Ejemplo:

**STO\_DODI\_SEQ (Secuencia de la table STO\_DOCUM\_DIGIT)**

### Synonyms

Son objetos de base de datos utilizados para hacer referencia a objetos que se encuentran en otros esquemas sin utilizar el esquema.

En el caso del sistema comercial estos objetos No deben ser públicos, es decir que deben tener un propietario y deben de mantener el nombre original salvo exista restricción técnica y sea autorizado por los administradores de la aplicación.

Ejemplo:

**SSICC\_PE\_ES.MAE\_CLIEN (Sinónimo de la tabla OWN\_PE\_ES.MAE\_CLIEN)**

### 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.

Los stored procedures deben ser nombrados según la siguiente nomenclatura:

<MODULO>\_PR \_<NOMBRE DEL PROCEDIMIENTO>

Usar notación Underscore Separated

El nombre del procedimiento debe estar compuesto por abreviaturas de a los 5 caracteres.

Ejemplo:

**MAE\_PR\_PROCE\_EJEMP**

### Functions

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 siguiente:

<MODULO>\_FN \_<NOMBRE DE LA FUNCION>

Usar notación Underscore Separated

El nombre de la función debe estar compuesto por abreviaturas de a los 5 caracteres.

Ejemplo:

**MAE\_FN\_FUNCT\_EJEMP**

### 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.

<MODULO>\_TRG \_<OPERACION>\_<NOMBRE CORTO DE LA TABLA>

Donde Operación: deberá tener las iniciales de la acción donde se ejecuta el trigger

Ejemplo:

**BAS\_TRG\_AI\_AUDI (Es un trigger que se ejecuta en las tablas de auditoria After Insert)**

### 

### 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.

Usar notación Underscore Separated

La nomenclatura definida es la siguiente:

T\_<IDENTIFICADOR DE TIPO>

Ejemplo:

T\_ESTRU\_ARCHI

### Packages

Son objetos de base de datos utilizados para agrupar lógicamente objetos de tipo PL/SQL como tipo, elementos y subprogramas, por lo general tienen dos partes, una especificación y un cuerpo , aunque a veces el cuerpo es innecesario.

En la especificación se declaran los tipos, variables, constantes, excepciones, cursores y subprogramas disponibles para su uso. El cuerpo define plenamente cursores y subprogramas, y así implementa la especificación.

Los **packages** deben ser nombrados según la siguiente nomenclatura:

<MODULO>\_PKG\_<NOMBRE DE PAQUETE>

Donde el nombre del paquete debe ser abreviado con a lo más 5 caracteres separados por el carácter underscore.

Ejemplo:

IMP\_PKG\_PROCE\_ELECT (Paquete de procesos de impression Electronica)

La nomenclatura de los elementos que componen el paquete debe de seguir el estándar definido en los puntos anteriores