****

**Padrão de Nomenclatura de Objetos**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 22/05/2014 | 1.0 | Criação | Igor Godoi |
| 25/02/2015 | 1.1 | Complementação e melhorias | Cleonaldo Freire |
|  |  |  |  |

Sumário

[1. Introdução 3](#_Toc413323660)

[2. Nomenclatura de Objetos de Banco de Dados 3](#_Toc413323661)

[2.1 Regras Gerais, válidas para qualquer tipo de objeto. 3](#_Toc413323662)

[2.2 Tabelas 3](#_Toc413323663)

[2.3 Colunas 4](#_Toc413323664)

[2.4 Sequences 6](#_Toc413323665)

[2.5 Views 6](#_Toc413323666)

[2.6 Views Materializadas 6](#_Toc413323667)

[2.7 Índices Não Únicos 7](#_Toc413323668)

[2.8 Índices de Contexto 7](#_Toc413323669)

[2.9 Índices Únicos (Unique Keys) 7](#_Toc413323670)

[2.10 Constraints: Primary Keys 7](#_Toc413323671)

[2.11 Constraints: Foreign Keys 8](#_Toc413323672)

[2.12 Constraints: Check 8](#_Toc413323673)

[2.13 Sinônimos 9](#_Toc413323674)

[2.14 Funções 9](#_Toc413323675)

[2.15 Procedures 9](#_Toc413323676)

[2.16 Triggers 10](#_Toc413323677)

[2.17 Package 11](#_Toc413323678)

[2.18 Schema 11](#_Toc413323679)

# ****Introdução****

O objetivo deste documento é definir um padrão de nomenclatura de objetos (*tables, indexes, sinonyms, columns, etc..)* para utilização nos projetos que envolvam o uso do banco de dados Oracle.

# Nomenclatura de Objetos de Banco de Dados

Apresentaremos neste documento as regras gerais de nomenclatura dos objetos de banco de dados (tabelas, colunas, índices, chaves, etc.), bem como as regras específicas para cada tipo de objeto. O Administrador de Dados (AD) tem por responsabilidade homologar cada alteração antes de publicá-las no ambiente de produção.

## Regras Gerais, válidas para qualquer tipo de objeto.

1. Todo objeto deve ter um comentário significativo cadastrado.
2. Todo nome de objeto deve ter no máximo 30 caracteres, no singular, sem acentuação, sem cedilhas, sem espaços e sem outros caracteres especiais. Se o nome exceder os 30 caracteres, será permitido abreviá-lo.
3. Todo nome de objeto é formado através da junção de qualificadores separados por “\_” (*underline*), conforme regras apresentadas ao longo deste documento. Entende-se por qualificador como uma palavra única, abreviada ou não, que representa um termo único do glossário do projeto ou de um glossário corporativo. Na falta de um glossário corporativo, deve-se reutilizar o nome, comentários e tipagem de dados de outro objeto que tenha o mesmo significado semântico.
4. Como o Oracle Server, por default, não diferencia minúsculas de maiúsculas em seu dicionário de dados (*case insensitive*), por isso, não há necessidade em distinguir caracteres maiúsculos de minúsculos.
5. Todo nome de objeto deve iniciar com caracter alfabético.

## Tabelas

|  |  |
| --- | --- |
| Formato | TB\_Qualificador1[\_Qualiligador2[\_Qualificadorn]] |
| Exemplos | *TB\_MACROAREA,TB\_USUARIO\_ADMINISTRADOR.* |
| Comentários | Até 30 caracteres alfanuméricos, iniciando sempre com um caracter alfabético, nomes na íntegra, no singular, sem acentuação e sem cedilha. Não deve possuir nenhum tipo de prefixo ou sufixo. Formado através da junção de qualificadores separados por “\_” (*underline*).  Obs: A tabela deve ter um comentário que explique sua finalidade. |

## Colunas

|  |  |
| --- | --- |
| Formato | <Tipagem>\_ Qualificadr1[\_Qualiligador2\_Qualificadorn] |
| Exemplo | *CD\_MACROAREA, NM\_LOGIN* |
| Comentários | Iniciar o nome com prefixo que identifique a tipo dos dados com 2 caracteres, acrescido de “\_” (*underline*) e junção de qualificadores separados por “\_” (*underline*). Os prefixos a serem utilizados são:  ID – Identificador único da tabela  IN – Campos booleanos (‘S’ ou ‘N’)  DN – Campo Domínio (Ex: DN\_SEXO pode ser [‘M’, ‘F’])  TP – Tipo  CD – Código  NM – Nome  VL – Valor  DS – Descrição  DT – Data  SG – Sigla  NR – Número  HS – Hash  DC – Documento, normalmente tipo de dados BLOB, CLOB ou BYTEA  IM – Imagem (campos BLOB e BYTEA) (Ex: IM\_FOTO)  Caso necessite acrescentar mais prefixos, favor enviar sugestões.  Obs: todas as colunas devem ter um comentário que explique a sua finalidade, se possível com exemplo. |

## Sequences

|  |  |
| --- | --- |
| Formato | SQ\_NomeTabela |
| Exemplo | *SQ\_MACROAREA* |
| Comentários | Até 30 caracteres alfanuméricos, iniciando sempre com os caracteres “SQ\_”. *NomeTabela* é o nome da tabela para a qual a *sequence* criada está relacionada. No caso da sequence estar relacionada a mais de uma tabela utilizar o nome da tabela principal (ou a mais importante). |

## Views

|  |  |
| --- | --- |
| Formato | VW\_Nometabela\_Comentario |
| Exemplo | VW\_NotaFiscal\_Cancelada |
| Comentários | *NomeTabela* será o nome da tabela principal do comando select utilizado para criar a view. *Comentario* serve para diferenciar a view da tabela que a originou, além de qualificá-la. Quando a view faz referência a uma tabela em um esquema diferente da tabela original, e o comando select que a originou conter apenas aquela tabela na cláusula from, este *Comentario* torna-se opcional. |

## Views Materializadas

|  |  |
| --- | --- |
| Formato | VM\_Nometabela\_Comentario |
| Exemplo | VM\_NotaFiscal\_Cancelada |
| Comentários | *NomeTabela* será o nome da tabela principal do comando select utilizado para criar a view materializada. *Comentario* serve para diferenciar a view materializada da tabela que a originou, além de qualificá-la. Quando a view materializada faz referência a uma tabela em um esquema diferente da tabela original, e o comando select que a originou conter apenas aquela tabela na cláusula from, este *Comentario* torna-se opcional. |

## Índices Não Únicos

|  |  |
| --- | --- |
| Formato | IX\_Nometabela\_Nomecoluna[\_NN] |
| Exemplo | IX\_COMPETENCIA\_NM\_COMPETENCIA, IX\_COMPETENCIA\_NM\_COMPETENCIA\_01 |
| Comentários | *Nometabela* é sobre a qual se está criando o índice. Nomecoluna é o da coluna que compõe o índice. Quando for o caso de múltiplas colunas comporem o índice, utiliza-se a principal. Quando há mais de um índice para a mesma coluna principal, acrescenta-se “\_” e um sequencia numérica de 2 números. |

## Índices de Contexto

|  |  |
| --- | --- |
| Formato | CX\_Nometabela\_Nomecoluna |
| Exemplo | CX\_COMPETENCIA\_NM\_COMPETENCIA |
| Comentários | *Nometabela* é sobre a qual se está criando o índice. Nomecoluna é o da coluna que compõe o índice. Quando for o caso de múltiplas colunas comporem o índice, utiliza-se a principal. |

## Índices Únicos (Unique Keys)

|  |  |
| --- | --- |
| Formato | UK\_Nometabela\_Nomecoluna[\_NN] |
| Exemplo | UK\_COMPETENCIA\_NM\_COMPETENCIA, UK\_COMPETENCIA\_NM\_COMPETENCIA\_01 |
| Comentários | *Nometabela* é sobre a qual se está criando o índice. Nomecoluna é o da coluna que compõe o índice. Quando for o caso de múltiplas colunas comporem o índice, utiliza-se a principal. Quando há mais de um índice para a mesma coluna principal, acrescenta-se “\_” e uma sequencia numérica de 2 números. |

## Constraints: Primary Keys

|  |  |
| --- | --- |
| Formato | PK\_NomeTabela |
| Exemplo | PK\_PROCESSO, PK\_USUARIO\_ADMINISTRADOR |
| Comentários | Iniciar com “PK\_” antes do nome da tabela. |

## Constraints: Foreign Keys

|  |  |
| --- | --- |
| Formato | FK\_NomeTabelaFilha\_NomeTabelaPai |
| Exemplo | FK\_SUBAREA\_MACROAREA |
| Comentários | Iniciar com “FK\_”. *NomeTabelaFilha* é o nome abreviado da tabela sobre a qual se está criando a foreign key. *NomeTabelaPai* é o nome abreviado da tabela à qual a foreign key sendo criada faz referência (*references*). |

## Constraints: Check

|  |  |
| --- | --- |
| Formato | CK\_NomeTabela\_NomeColuna |
| Exemplo | CK\_COMPETENCIA\_TP\_GRUPO |
| Comentários | Iniciar com “CK\_”. *NomeTabela* é o nome da tabela sobre a qual se está criando a constraint. *NomeColuna* é o nome da coluna sobre a qual se está criando a constraint. |

## Sinônimos

|  |  |
| --- | --- |
| Formato | SN\_NomeTabela sem esquema |
| Exemplo | SN\_COMPETENCIA |
| Comentários | *NomeTabela* é o nome da tabela a qual o sinônimo se refere. Normalmente o sinônimo serve para encapsular o esquema da tabela, facilitando a referência à tabela. |

## Funções

|  |  |
| --- | --- |
| Formato | fn\_nomefuncao([p\_parametro1 TIPO,  p\_ parametron TIPO]) RETURN TIPO |
| Exemplo | fn\_obter\_total(p\_id\_pessoa NUMBER,  p\_ano NUMBER) RETURN NUMBER |
| Comentários | O nome da função deve estar relacionado ao seu propósito. Deve conter no máximo 30 caracteres e começar obrigatoriamente pelo prefixo fn\_. Caso seja composta por mais de uma palavra, elas devem ser separadas por underline (\_). |

## Procedures

|  |  |
| --- | --- |
| Formato | pr\_nomeprocedimento([p\_parametro1 TIPO,  p\_ parametron [OUT] TIPO]) |
| Exemplo | pr\_processar\_tabela(p\_id\_pessoa NUMBER,  p\_ano OUT NUMBER) |
| Comentários | O nome do procedimento deve estar relacionado ao seu propósito. Deve conter no máximo 30 caracteres e começar obrigatoriamente pelo prefixo pr\_. Caso seja composto por mais de uma palavra, elas devem ser separadas por underline (\_).  Obs: No PostgreSQL, existe apenas o objeto function. Caso queira criar uma “procedure”, deve-se criar uma função retornando “void”. Nesse caso, utilizar a nomenclatura de procedure, ou seja, o prefixo “pr”. |

## Triggers

|  |  |
| --- | --- |
| Formato | TG\_NomeTabela\_{B | A} {I | U | D} {S} |
| Exemplo | TG\_COMPETENCIA\_AI |
| Comentários | O nome inicia com “TG\_”. Depois acrescentar o *NomeTabela* é a tabela sobre a qual se está criando a trigger. O sufixo (B | A) indica o momento que é disparada a trigger (B – before e A - after), o sufixo (I/U/D) indica o tipo de trigger que se está criando, para respectivamente insert, update ou delete, podendo ter um ou vários sufixos. O último sufixo (S) significa o nível da trigger (trigger de linha (row) ou de sentença (statement)). O nível default é row, nesse caso não acrescentar nada, no caso do nível de statement, acrescentar o S. |

## Package

|  |  |
| --- | --- |
| Formato | PC\_NomeDaPackage |
| Exemplo | PC\_Precatorio |
| Comentários | A package agrupará procedimento e funções afins. Deve ter no máximo 30 caracteres e iniciar obrigatoriamente com o prefixo PC\_. Dentro da package, os procedimentos devem começar pelo prefixo p\_ e as funções pelo prefixo f\_ para diferenciar dos procedimentos e funções externos a package. |

## Schema

|  |  |
| --- | --- |
| Formato | SC\_NomeDoEsquema |
| Exemplo | SC\_IMPORTACAO |
| Comentários | O esquema agrupará os objetos afins. Deve ter no máximo 30 caracteres e começar obrigatoriamente com o prefixo SC\_.  Obs: essa opção se aplica ao PostgreSQL, pois nesse existe o objeto schema. No caso do Oracle, só existe o objeto user, nesse caso, pode-se convencionar que o nome do user que será utilizado para criar os objetos e funcionar como um esquema começar com o prefixo SC\_. |