**<PetLife>**

**Padrão para Nomenclatura de Banco de Dados**

Versão <1.0>

**Histórico**

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 15/07/2023 | 1.0 | Versão inicial. | Valdir Mendonça Santana |
| 17/07/2023 | 1.0 | Revisão ortográfica | Natan Vinícius Santos Pereira |
|  |  |  |  |

[**1**](#_gjdgxs) **INTRODUÇÃO 4**

[**2**](#_30j0zll) **BANCO DE DADOS 4**

[**2.1**](#_1fob9te) **REGRA GERAL** 4

[**2.2**](#_3znysh7) **SINTAXE** 4

[**2.3**](#_2et92p0) **EXEMPLOS** 4

[**3**](#_tyjcwt) **TABELAS 5**

[**3.1**](#_3dy6vkm) **REGRA GERAL** 5

[**3.2**](#_1t3h5sf) **SINTAXE** 5

[**3.3**](#_4d34og8) **EXEMPLOS** 5

[**4**](#_2s8eyo1) **ATRIBUTOS 5**

[**4.1**](#_17dp8vu) **REGRA GERAL** 5

[**4.2**](#_3rdcrjn) **SINTAXE** 5

[**4.3**](#_26in1rg) **EXEMPLOS** 6

[**5**](#_lnxbz9) **VIEWS 6**

[**5.1**](#_35nkun2) **REGRA GERAL** 6

[**5.2**](#_1ksv4uv) **SINTAXE** 6

[**5.3**](#_44sinio) **EXEMPLOS** 6

[**6**](#_2jxsxqh) **ÍNDICES 7**

[**6.1**](#_z337ya) **REGRA GERAL** 7

[**6.2**](#_3j2qqm3) **SINTAXE** 7

[**6.3**](#_1y810tw) **EXEMPLOS** 7

[**7**](#_4i7ojhp) **STORED PROCEDURES 7**

[**7.1**](#_2xcytpi) **REGRA GERAL** 7

[**7.2**](#_1ci93xb) **SINTAXE** 7

[**7.3**](#_3whwml4) **EXEMPLOS** 8

[**8**](#_2bn6wsx) **TRIGGER 8**

[**8.1**](#_qsh70q) **REGRA GERAL** 8

[**8.2**](#_3as4poj) **SINTAXE** 8

[**8.3**](#_1pxezwc) **EXEMPLOS** 9

[**9**](#_49x2ik5) **CHECK 9**

[**9.1**](#_2p2csry) **REGRA GERAL** 9

[**9.2**](#_147n2zr) **SINTAXE** 9

[**9.3**](#_3o7alnk) **EXEMPLOS** 9

[**10**](#_23ckvvd) **SEQUENCE 9**

[**10.1**](#_ihv636) **REGRA GERAL** 9

[**10.2**](#_32hioqz) **SINTAXE** 9

[10.3](#_1hmsyys) Exemplos 10

1. **INTRODUÇÃO**

O maior esforço no desenvolvimento de software é dispensado às atividades de manutenção, desta forma, quanto mais fácil for o entendimento do código do sistema, mais produtiva será a equipe de desenvolvimento. Frequentemente as pessoas que escrevem o código não são as mesmas que o mantém e, quando são, geram uma dependência com o código desenvolvido que dificilmente é dissolvida. Um padrão de codificação/nomenclatura visa minimizar esses problemas, pois estabelece regras, definindo como o código deve ser escrito para favorecer a impessoalidade do artefato.

Este documento tem como objetivo definir um padrão de nomenclatura para definição de bancos de dados que, quando usado, garante um melhor entendimento por qualquer pessoa que conheça e siga o mesmo.

1. **BANCO DE DADOS**

* 1. **Regra Geral**

O nome do banco de dados deverá identificar o negócio que está sendo automatizado ou deverá refletir a sigla da aplicação.

* 1. **Sintaxe**

{[A..Z][a..z]} → Xxxxx, onde:

Xxxxx – indica o nome do banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Usar palavras no singular, sem acentuação ou caracteres especiais.

* 1. **Exemplos**

PetLife

1. **TABELAS**

* 1. **Regra Geral**

O nome das tabelas deve ser sugestivo e refletir os dados armazenados nesta. O nome da tabela não deve usar nenhum prefixo.

* 1. **Sintaxe**

{[A..Z][a..z]} → Xxxxx, onde:

Xxxxx – indica o nome da tabela no banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Limite-se a usar 30 caracteres para definição do nome da tabela – padrão da maioria dos bancos de dados.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.

* 1. **Exemplos**

Usuario – Tabela de Usuário

Administrador – Tabela de Administrador

Venda – Tabela de Venda

Medicamento – Tabela de Medicamento

Pet – Tabela de Pet

PetShop – Tabela de PetShop

Consulta – Tabela de Consulta

Vacina – Tabela de Vacina

Produto – Tabela de Produto

1. **ATRIBUTOS**

* 1. **Regra Geral**

Os nomes dos atributos devem identificar a coluna de maneira clara e descritiva.

* 1. **Sintaxe**

{[A..Z][a..z]} → Xxxxx, onde:

Xxxxx – indica o nome dos atributos de uma determinada tabela do banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas. Abreviação limitada a 4 caracteres.
* Não repetir o nome da tabela para compor o nome do atributo
* Atributo para chaves primárias artificiais devem usar o nome Id

* 1. **Exemplos**

*Sem Abreviações*

*Tabela – Usuario*

Id

Nome

Telefone

Cidade

Rua

Numero

CPF

Email

Senha

CEP

TipoUsuario

DataNascimento

*Tabela – Administrador*

IdAdministrador

*Tabela – Venda*

Id

Valor

Quitar

Numero

*Tabela – Medicamento*

Id

Nome

Dosagem

Medida

Frequencia

Intervalo

Importado

Ativo

*Tabela – Pet*

Id

Nome

Especie

Raca

Sexo

Peso

*Tabela – PetShop*

Id

Nome

CNPJ

*Tabela – Consulta*

Id

Protocolo

Preco

Cliente

*Tabela – Vacina*

Id

Periodo

Tempo

Vacinado

*Tabela – Produto*

Id

Nome

Preco

Status

Quantidade

Descricao

*Com Abreviações*

*Tabela – Usuario*

DataNascimento

TipoUsuario

*Tabela – Administrador*

CodigoADM

*Tabela – Venda*

FormaPagamento

*Tabela – Medicamento*

DataInicio

DataTermino

*Tabela – Pet*

DataNascimento

*Tabela – PetShop*

*Tabela – Consulta*

DataAgendamento

DataConsulta

*Tabela – Vacina*

DataAplicacao

*Tabela – Produto*

1. **VIEWS**

* 1. **Regra Geral**

Deve-se usar a mesma semântica utilizadas para as tabelas. O nome deve ser pré-fixado pela palavra VW.

* 1. **Sintaxe**

VW\_{[A..Z][a..z]} → VW\_Xxxxx, onde:

Xxxxx – indica o nome da view no banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Limite-se a usar 30 caracteres para definição do nome da View.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.

* 1. **Exemplos**

VW\_Pessoa – Visão da Tabela de Pessoas

VW\_Funcionario – Visão da Tabela de Funcionários

1. **ÍNDICES**

* 1. **Regra Geral**

Deve-se usar a mesma semântica utilizadas para as tabelas. O nome deve ser pré-fixado pela palavra IDX.

* 1. **Sintaxe**

IDX\_{[A..Z][a..z]} → IDX\_XxxxxXxxxx, onde:

XxxxxXxxxx – indica o nome do índice no banco de dados. Esse nome deve ser composto pelo nome da tabela mais o nome dos campos chaves.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Limite-se a usar 30 caracteres para definição do nome do Índice.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.

* 1. **Exemplos**

IDX\_PessoaCodigo – Índice da Tabela Pessoa pelo atributo Código.

IDX\_FuncionarioNome – Índice da Tabela Funcionário pelo atributo Nome.

IDX\_CartaoCreditoNumero – Índice da Tabela Cartão de Crédito pelo atributo Número.

1. **STORED PROCEDURES**

* 1. **Regra Geral**

Deve-se usar a mesma semântica utilizadas para as tabelas. O nome deve ser pré-fixado pela palavra STP seguido por uma sigla para indicar a operação principal realizada.

* 1. **Sintaxe**

STP\_{INS|DEL|UPD|SLC}\_{[A..Z][a..z]} → STP\_XXX\_Xxxxx, onde:

XXX – identifica a ação principal da stored procedure: INS – Inserção, DEL – Exclusão, UPD – atualização e SLC – consulta a dados.

Xxxxx – indica o nome da Stored Procedure no banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
* Endentar de modo a tornar a codificação clara e facilitar o trabalho de manutenção. Quando disponível utilizar ferramentas de indentação automática do banco de dados.
* Inserir comentários sempre que necessário. Comentários que descrevem procedimentos óbvios devem ser evitados para evitar poluição do código desenvolvido.
* Evitar aninhamento excessivo de comandos, o que dificulta a manutenção do código.

* 1. **Exemplos**

STP\_INS\_Pessoa – Stored Procedure para atualização de dados de Pessoa.

STP\_UPD\_SaldoCliente – Stored Procedure para atualização de saldo do Cliente.

STP\_SLC\_HistoricoFunc – Stored Procedure para consulta do histórico do Funcionário.

1. **TRIGGER**

* 1. **Regra Geral**

Deve-se usar a mesma semântica utilizadas para as tabelas. O nome deve ser pré-fixado pela palavra TRG seguido por uma sigla para indicar a operação principal realizada.

* 1. **Sintaxe**

TRG\_{INS|DEL|UPD|SLC}\_{[A..Z][a..z]} → TRG\_XXX\_Xxxxx, onde:

XXX – identifica a ação principal da Trigger: INS – Inserção, DEL – Exclusão, UPD – atualização e SLC – consulta a dados.

Xxxxx – indica o nome da Stored Procedure no banco de dados.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
* Endentar de modo a tornar a codificação clara e facilitar o trabalho de manutenção. Quando disponível utilizar ferramentas de indentação automática do banco de dados.
* Inserir comentários sempre que necessário. Comentários que descrevem procedimentos óbvios devem ser evitados para evitar poluição do código desenvolvido.
* Evitar aninhamento excessivo de comandos, o que dificulta a manutenção do código.

* 1. **Exemplos**

TRG\_DEL\_Pessoa – Trigger para excluir dados de Pessoa.

TRG\_UPD\_SaldoCliente – Trigger para atualização de saldo do Cliente.

TRG\_SLC\_HistoricoFunc – Trigger para consultar o histórico do Funcionário.

1. **CHECK**

* 1. **Regra Geral**

Especifica que uma condição deve ser atendida por cada tupla da tabela.

* 1. **Sintaxe**

CHK\_{[A..Z][a..z]}\_{[A..Z][a..z]} → CHK\_Xxxxx\_Xxxxx, onde:

Xxxxx\_Xxxxx – indica o nome da tabela seguido pelo nome da coluna onde haverá o check.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.

* 1. **Exemplos**

CHK\_Pessoa\_Cpf – Check no campo CPF da tabela Pessoa.

CHK\_Cliente\_Sexo – Check no campo sexo da tabela Cliente.

1. **SEQUENCE**

* 1. **Regra Geral**

Uma sequence é um objeto do banco de dados criado pelo usuário, que pode ser compartilhado por vários usuários para gerar números seqüenciais inteiros exclusivos. Normalmente, as sequences são utilizadas para criar um valor de PRIMARY KEY, que deve ser exclusivo para cada linha. Os números da sequence são armazenados e gerados de modo independente das tabelas. Portanto, a mesma sequence pode ser usada para várias tabelas.

* 1. **Sintaxe**

SQC\_{[A..Z][a..z]} → SQC\_Xxxxx, onde:

Xxxxx – indica o nome do campo que utilizará a Sequence.

* A primeira letra deverá ser maiúscula e as demais minúsculas. Para cada palavra interna, primeira letra em maiúsculo.
* Não usar preposições.
* Usar palavras no singular, sem acentuação ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.

* 1. **Exemplos**

SQC\_Inscricao – Sequence para o campo inscrição

SQC\_Solicitacao – Sequence para o campo solicitação.