**<Residential Life>**

**Padrão para Nomenclatura de Banco de Dados**

Versão <1.0>

**Histórico**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 22/06/2013 | 1.0 | Versão Inicial | Tauany Santana |
|  |  |  |  |
|  |  |  |  |

[1 Introdução 4](#_Toc203191522)

[2 BANCO DE DADOS 4](#_Toc203191523)

[2.1 Regra Geral 4](#_Toc203191524)

[2.2 Sintaxe 4](#_Toc203191525)

[2.3 Exemplos 4](#_Toc203191526)

[3 TABELAS 5](#_Toc203191527)

[3.1 Regra Geral 5](#_Toc203191528)

[3.2 Sintaxe 5](#_Toc203191529)

[3.3 Exemplos 5](#_Toc203191530)

[4 ATRIBUTOS 5](#_Toc203191531)

[4.1 Regra Geral 5](#_Toc203191532)

[4.2 Sintaxe 5](#_Toc203191533)

[4.3 Exemplos 6](#_Toc203191534)

[5 VIEWS 6](#_Toc203191535)

[5.1 Regra Geral 6](#_Toc203191536)

[5.2 Sintaxe 6](#_Toc203191537)

[5.3 Exemplos 6](#_Toc203191538)

[6 ÍNDICES 7](#_Toc203191539)

[6.1 Regra Geral 7](#_Toc203191540)

[6.2 Sintaxe 7](#_Toc203191541)

[6.3 Exemplos 7](#_Toc203191542)

[7 STORED PROCEDURES 7](#_Toc203191543)

[7.1 Regra Geral 7](#_Toc203191544)

[7.2 Sintaxe 7](#_Toc203191545)

[7.3 Exemplos 8](#_Toc203191546)

[8 TRIGGER 8](#_Toc203191547)

[8.1 Regra Geral 8](#_Toc203191548)

[8.2 Sintaxe 8](#_Toc203191549)

[8.3 Exemplos 9](#_Toc203191550)

[9 CHECK 9](#_Toc203191551)

[9.1 Regra Geral 9](#_Toc203191552)

[9.2 Sintaxe 9](#_Toc203191553)

[9.3 Exemplos 9](#_Toc203191554)

[10 SEQUENCE 9](#_Toc203191555)

[10.1 Regra Geral 9](#_Toc203191556)

[10.2 Sintaxe 9](#_Toc203191557)

[10.3 Exemplos 10](#_Toc203191558)

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 do sistema Residential Life 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.

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

Condominio

1. TABELAS
   1. Regra Geral

O nome das tabelas deve ser sugestivo e refletir os dados armazenados nesta. O nome deve ser pré-fixado pela palavra TB.

* 1. Sintaxe

TB\_{[A..Z][a..z]} → TB\_Xxxxx, onde:

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

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
  1. Exemplos

TB\_Pessoa – Tabela de Pessoas

TB\_Condominio – Tabela de Condomínios

TB\_Movimentacao\_Financeira – Tabela de Movimentação Financeira

1. ATRIBUTOS
   1. Regra Geral

O nome dos atributos devem identificar a coluna de maneira clara e descritiva.

* 1. Sintaxe

{[A..Z][a..z]} → Xxxxx, onde:

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

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas. Abreviações limitadas a 4 caracteres.
  1. Exemplos

*Sem Abreviações*

Nome

Email

Login

*Com Abreviações*

Pes\_Nome

Pes\_Email

Pes\_Login

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.

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
  1. Exemplos

VW\_Pessoa – Visão da Tabela de Pessoas

VW\_Condominio – 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.

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
  1. Exemplos

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

IDX\_CondominioCodigo – Índice da Tabela Condomínio pelo atributo Código.

IDX\_BlocoNome – Índice da Tabela Bloco pelo atributo Nome.

1. STORED PROCEDURES
   1. Regra Geral

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

* 1. Sintaxe

SP\_{INS|DEL|UPD|SLC}\_{[A..Z][a..z]} → SP\_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.

* 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 acentuações 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 endentação 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

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

SP\_UPD\_Bloco – Stored Procedure para atualização do Bloco.

SP\_SLC\_Espaco – Stored Procedure para consulta dos Espaços cadastrados.

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.

* 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 acentuações 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 endentação 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\_Bloco – Trigger para atualização dos dados do Bloco.

TRG\_SLC\_Espaco – Trigger para consultar os Espaços.

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.

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
  1. Exemplos

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

CHK\_Pessoa\_Sexo – Check no campo sexo da tabela Pessoa.

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 sequenciais 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.

* 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 acentuações ou caracteres especiais.
* Evite usar abreviações, se necessário use as conhecidas.
  1. Exemplos

SQC\_Reserva – Sequence para a reserva.

SQC\_Ocorrencia – Sequence para uma ocorrência.