



BOAS PRÁTICAS PARA O BANCO DE DADOS

Julho de 2021

HISTÓRICO DE ALTERAÇÕES

ALTERAÇÃO	RESPONSÁVEL	DATA
Criação do documento.	Elizabeth Cristina	08/04/2021
Atualização dos tipos.	Elizabeth Cristina	09/04/2021
Atualização de regras de escrita.	Elizabeth Cristina	14/04/2021
Inserção de informações sobre dados.	Elizabeth Cristina	15/04/2021
Regras para Foreign Key.	Elizabeth Cristina	19/04/2021
Inserção de regras para Index.	Elizabeth Cristina	28/04/2021
Atualização de regras de relacionamento entre tabelas.	Daniel Alves, Elizabeth Cristina	17/06/2021
Atualização da regra de relacionamento entre tabelas.	Daniel Alves, Elizabeth Cristina	24/06/2021
Regra para campos de tamanho fixo e flexível	Daniel Alves, Elizabeth Cristina	01/07/2021
Index de mais de um atributo e alteração no sessão 2 para uso e desuso	Daniel Alves, Elizabeth Cristina	09/07/2021

1. APRESENTAÇÃO

Este documento apresenta os padrões que devem ser utilizados no banco de dados.

2. PORQUE USAR (OU NÃO) USAR O INDEX

2.1. PORQUE USAR

O index é utilizado para fins de otimização, permitindo uma localização mais rápida de um registro quando uma consulta é efetuada. É uma estrutura que possibilita acesso a um item indexado, desde que a busca tenha complexidade inferior à complexidade linear, podendo ser logarítmica ou constante.

2.1. PORQUE NÃO USAR

O index traz bastantes benefícios, mas seu uso consome recursos internos do SGBD para mantê-los atualizados e associados. Sua manutenção requer tempo e recursos, portanto, não deve ser criado se seu uso não for, efetivamente, ser feito. Quando há grande quantidade de dados duplicados, os índices deixam de ser um benefício e passam a ser um custo. O mesmo ocorre se o índice for usado para atributos com pouca variação, como, por exemplo, “sexo”, ou campos flag, como status, que geralmente contém dados do tipo ‘ativo’, ‘inativo’.

3. POR QUE USAR FK

FK (Foreign Key) é a chave que permite a referência a registros oriundos de outras tabelas, ou seja, é o campo que compõe a chave primária de uma outra tabela. A sua utilização possibilita a implementação da integridade de dados diretamente no banco de dados, conhecida como integridade referencial. Uma chave estrangeira é a representação de um relacionamento entre tabelas.

4. TIPOS E TAMANHOS SUPORTADOS

Ao fazer e/ou alterar, levar em consideração os pontos abaixo:

- Fazer uso do datatype inadequado pode consumir mais espaço que o necessário.
- A escolha adequada do tipo e sua limitação evita alterações futuras.

TIPO	DESCRIÇÃO
Tinyint	Aceita valores entre 0 a 255 e ocupa 1 byte.
Smallint	Aceita valores entre -32768 até 32767 e ocupa 2 bytes.

Int	Os valores aceitos aqui variam entre -2^{31} a 2^{31-1} e ocupa 4 bytes.
Bigint	Aceita valores entre -2^{63} e 2^{63-1} e ocupa 8 bytes.
Enum ¹	Uma coluna deste tipo inclui uma lista de valores permitidos, cada um dos quais sendo chamado de "membro" da lista. São dados na forma de uma lista de strings, entre aspas e separadas por vírgulas. o MySQL armazena as strings como números inteiros, usando os valores 1 a n para uma coluna com n membros na enumeração.
Float	Aceita até 7 casas decimais (em precisão) e ocupa 4 bytes.
Double	Aceita até 14 casas decimais (em precisão) e ocupa 8 bytes.
Char ²	Aceita como valor qualquer dígito, sendo que o espaço ocupado no disco é de um dígito por caractere. É possível utilizar até 8 mil dígitos.
Varchar	Aceita como valor qualquer dígito e o espaço ocupado em disco é de um dígito por caractere. Permite usar também no máximo 8 mil dígitos.
Blob	O blob (Binary Large Object - grande objeto binário) é um campo criado para o armazenamento de qualquer tipo de informação em formato binário, armazenando até 16535 caracteres, equivalente a 16 bits.
MEDIUMBLOB	O mediumblob (Binary Large Object - grande objeto binário) é um campo criado para o armazenamento de qualquer tipo de informações em formato binário, armazenando até 24 bits.
Data	Permite o uso de valores no formato aaaa-mm-dd, no intervalo de 0001-01-01 a 9999-12-31. Sua precisão é de 1 dia e ocupa 3 bytes.
Datetime	Permite o uso de valores entre 01/01/1753 e 31/12/9999. Este datatype ocupa 8 bytes e sua precisão atinge 3.33 milissegundos.
Time	Permite o uso de valores no formato hh:mm:ss[.nnnnnnn], no intervalo de 00:00:00.0000000 a 23:59:59.9999999. Sua precisão é de 100 nanossegundos e ocupa de 3 a 5 bytes.
Timestamp	Possui comportamento e propriedade variados, dependendo da versão do MySQL e do modo SQL que o servidor está executando.

Tabela 1 – Tabela com os tipos de dados para o BD e seus tamanhos limites.

Enum¹: Usar quando for necessário fazer uma pré-listagem. Exemplo: Status → enum = 'falha_no_envio','enviada','respondida','impossivel_enviar'.

Char²: Usar em casos do tipo: UF que recebe apenas 2 caracteres.

5. COMANDOS PARA ATUALIZAR, EDITAR E EXCLUIR

Essas alterações só devem ser feitas se houver necessidade.

COMANDO	AÇÃO
ALTER TABLE nome_tabela ADD nome_coluna tipo_dados;	Adiciona uma coluna a tabela.
ALTER TABLE nome_tabela DROP COLUMN nome_coluna;	Exclui uma coluna da tabela.
ALTER TABLE nome_tabela MODIFY COLUMN nome_coluna tipo_dados;	Modifica o tipo do dado a coluna da tabela.
ALTER TABLE nome_tabela MODIFY nome_coluna tipo_dados NOT NULL;	Coloca constraint NOT NULL a coluna da tabela
ALTER TABLE nome_da_tabela CHANGE nome_atual novo_nome [Tipo de Dados];	Renomeia o nome de uma coluna da tabela.
ALTER TABLE nome_tabela DROP CONSTRAINT nome_constraint;	Exclui um constraint de uma tabela.

Tabela 2 – Comandos para usar quando houver necessidade.

6. REGRAS PARA PADRONIZAÇÃO DA ESCRITA DO DDL E REGRAS DE CONSULTAS SQL

Ao fazer consultas ou fazer e/ou alterar DDL, levar em consideração os pontos abaixo.

Obs.: Mais regras vão ser acrescentadas ao longo do tempo.

6.1. REGRA PARA TABELAS E COLUNAS

- Fazer uso de nome em Português, exceto quando estritamente necessário; com exceção da coluna **id**, que por padrão é PK da tabela;
- Evitar uso de siglas. Com excesso: CPF, CNPJ, RG e casos do mesmo estilo;
- Usar o (underline) para separar os nomes, exemplos: data_cadastro, porta_dianteira_direita;
- Se o nome ultrapassar 60 caracteres, deve ser simplificado de forma que sua leitura fique mais intuitiva.

6.2. REGRA PARA COLUNAS

- Usar no identificador PK da tabela o nome “id” sem acompanhamento;
- Limitar ao, máximo possível, as colunas pelo tamanho do dado (quando se sabe). ex.: uf → char(2), cpf → char(11);

6.3. PADRÃO DE REFERÊNCIA FK

- A coluna FK deve ser renomeada como “id_nome_da_tabela_referenciada”;
- Tabela intermediária: tabela com as FKs seguem a regra acima;
- Quando a FK for id então seguir: fk__tabela_origem__tabela_referenciada;
- As demais colunas devem seguir:
fk__tabela_origem__nome_coluna__tabela_referenciada;
- No caso de duas ou mais colunas da mesma tabela referenciarem uma coluna de outra tabela, utilizar: fk__tabela_origem__coluna__tabela_referenciada__coluna_tabela_atual, ex.: fk__telefone__tabela_referenciada_telefone_comercial e
fk__telefone__tabela_referenciada_telefone_residencial
- Se devido à regra acima o nome ficar muito extenso, repetir o que o exemplo abaixo mostra:
 - **Colunas:** id_cliente_antigo e id_cliente_atual.
 - **FKs:** fk__cad_clientes__auditoria_iscas__cliente_antigo e
fk__cad_clientes__auditoria_iscas__cliente_atual

6.4. PADRÃO DO INDEX

- Ao indexar as colunas utilizar: idx_nome_coluna ou idx_coluna1_coluna2_colunaN;
- Se a coluna é FK utilizar:
idx_fk__nome_tabela_origem__nome_coluna__nome_tabela_referenciada, ex.:
“idx_fk__contato_usuario__cpf_usuario_gestor”;
- Index de mais de um atributo:
idx_“primeiro_atributo”__“segundo_atributo”__“n_atributo”.

6.5. NÃO UTILIZAÇÃO E SUBSTITUIÇÃO (QUANDO NECESSÁRIO)

- Não utilizar numeração para status do tipo Ativo/Inativo, Ligado/Delisdado e afins. Fazer uso de enum e listar, já nele, as opções. Ex.: `status` enum('inativo','ativo');
- Não fazer uso do _ (underline) no início ou no fim da palavra. Ex.: cnpj_.

6.6. O QUE NÃO FAZER

- Evitar, sempre que possível, o uso de SELECT *, pois o SGBD precisará processar todas as colunas das tabelas especificadas no FROM;
- Evitar, sempre que possível, o uso de negativas no WHERE, pois quebram o index (quando tem), ignorando sua função própria.

7. REGRAS PARA COLUNAS DAS TABELAS

COLUNAS	TIPO (TAMANHO)
cpf	char(11)
cpf_cnpj	varchar(14)

Tabela 3 – Tabela com os tipos de dados para o BD e seus tamanhos limites.