**Guia de Projeto de Banco de Dados**

SUMÁRIO

[1.Introdução 2](#__RefHeading__690_1126619083)

[2.Guia de Projeto de Banco de Dados 2](#__RefHeading__692_1126619083)

[2.1.Padrões de Nomenclatura 2](#__RefHeading__694_1126619083)

[2.1.1.Identificadores de objeto 2](#__RefHeading__696_1126619083)

[2.1.2.Identificadores de coluna 2](#__RefHeading__698_1126619083)

[2.1.3.Nomeação de Arquivos 3](#__RefHeading__700_1126619083)

[2.2.Definindo a solução 4](#__RefHeading__702_1126619083)

[2.2.1.Evitar duplicidade de tabelas e ou campos 4](#__RefHeading__704_1126619083)

[2.2.2.Definir chave primária para tabela 4](#__RefHeading__706_1126619083)

[2.2.3.Manter consistência dos objetos 4](#__RefHeading__708_1126619083)

[2.2.4.Elaborar e manter a documentação 4](#__RefHeading__710_1126619083)

[2.3.Implementação da Solução 4](#__RefHeading__712_1126619083)

[2.3.1.Criando Diagrama EER no MySQL Workbench 4](#__RefHeading__714_1126619083)

[2.3.2.Gerando script SQL no MySQL Workbench 6](#__RefHeading__716_1126619083)

# Introdução

Esse guia tem como objetivo direcionar as atividades realizadas pela equipe de Projeto de Banco de Dados.

# Guia de Projeto de Banco de Dados

## Padrões de Nomenclatura

### Identificadores de objeto

Os identificadores de objeto (tabela/visão) devem obedecer às seguintes regras:

* Devem ser compostos de uma primeira parte, que identifica o tipo do objeto (tb para tabela ou vw para visão), e de uma segunda parte que deve representar o tipo de informação dos registros do objeto.
  + Ex.: tb\_cliente
  + Ex.: vw\_vendas\_por\_departamento
  + Ex.: ~~cliente~~
* Devem fazer uso de underscore (“\_”) para separar nomes compostos.
  + Ex.: tb\_cliente\_cnpj
  + Ex.: ~~tb\_clientecnpj~~
* Devem estar no singular, e não no plural.
  + Ex.: tb\_departamento
  + Ex.: ~~tb\_departamentos~~
* Devem ter limite máximo de 40 caracteres.
  + Ex.: vw\_vendas\_por\_departamento
  + Ex.: ~~vw\_vendas\_totais\_por\_departamento~~
* Devem estar em letras minúsculas.
  + Ex.: tb\_consultas
  + Ex.: ~~TB\_CONSULTAS~~
* Não devem possuir acentuação gráfica.
  + Ex.: tb\_solicitacao
  + Ex.: ~~tb\_solicitação~~
* Podem possuir um nome composto, no caso de tabelas de relacionamento.
  + Ex.: tb\_funcionario\_departamento ou tb\_lotacao, para lotação de funcionário em departamento.
* Devem possuir uma terminação “log”, no caso de tabelas de histórico.
  + Ex.: tb\_cliente\_alteracoes\_log, para tabela de registro de alterações no cadastro do cliente.

### Identificadores de coluna

Os identificadores de coluna devem obedecer às seguintes regras:

* Devem ser compostos de uma primeira parte, que identifica o tipo do dado da coluna (conforme abreviação contida na Tabela 1), e de uma segunda parte que deve representar o tipo de informação da coluna.
  + Ex.: int\_codigo
  + Ex.: ~~codigo~~
* Devem fazer uso de underscore (“\_”) para separar nomes compostos.
  + Ex.: vchr\_nome\_completo
  + Ex.: ~~vchr\_nomecompleto~~
* Devem estar no singular, e não no plural.
  + Ex.: vchr\_endereco
  + Ex.: ~~vchr\_enderecos~~
* Devem ter limite máximo de 30 caracteres, com exceção das chaves estrangeiras.
  + Ex.: vchr\_endereco\_completo
  + Ex.: ~~vchr\_endereco\_completo\_com\_estado\_e\_pais~~
* Devem estar em letras minúsculas.
  + Ex.: dt\_nascimento
  + Ex.: ~~DT\_NASCIMENTO~~
* Não devem possuir acentuação gráfica.
  + Ex.: ts\_emprestimo
  + Ex.: ~~ts\_empréstimo~~
* No caso de a coluna ser chave primária (PRIMARY KEY), o nome deve ter o prefixo “pk”.
  + Ex.: pk\_int\_codigo
  + Ex.: ~~int\_codigo~~
* No caso de a coluna ser chave estrangeira (FOREIGN KEY), o nome deve ser composto pelo prefixo “fk”, pelo tipo de dado da coluna (conforme Tabela 1), pelo nome da tabela referenciada e pelo nome da coluna na tabela referenciada sem o seu tipo de dado.
  + Ex.: fk\_int\_tb\_departamento\_codigo
  + Ex.: ~~fk\_int\_tb\_departamento\_int\_codigo~~
  + Ex.: ~~fk\_tb\_departamento\_int\_codigo~~
  + Ex.: ~~fk\_tb\_departamento\_codigo~~
* Opcionalmente, pode haver um outro nome para diferenciar os nomes de duas chaves estrangeiras para a mesma tabela.
  + Ex.: fk\_int\_id\_mandante\_tb\_time e fk\_int\_id\_visitante\_tb\_time, para uma chave estrangeira para o time mandante de um jogo e uma chave estrangeira para o time visitante do mesmo jogo.

**Tabela 1 – Abreviações de tipos de dado de coluna.**

|  |  |  |
| --- | --- | --- |
| **Abreviação** | **Tipo do dado da coluna** | **Exemplo** |
| int | INTEGER | int\_codigo |
| sint | SMALLINT | sint\_idade |
| bint | BIGINT | bint\_calculo |
| srl | SERIAL | srl\_id |
| bsrl | BIGSERIAL | bsrl\_id |
| real | REAL | real\_fator |
| dbl | DOUBLE PRECISION | dbl\_indice |
| num | DECIMAL/NUMERIC | num\_salario |
| vchr | VARCHAR | vchr\_nome\_completo |
| chr | CHAR | chr\_uf |
| txt | TEXT | txt\_obs |
| dt | DATE | dt\_nascimento |
| hr | TIME | hr\_inicio\_consulta |
| ts | TIMESTAMP | ts\_cadastro |
| itv | INTERVAL | itv\_intervalo |
| bool | BOOLEAN | bool\_ativo |

### Nomeação de Arquivos

Os nomes dos arquivos devem obedecer às seguintes regras:

* Os scripts SQL devem ter extensão .sql.
* Devem ser compostos por “db”, seguido do nome do projeto e da data (ano, mês e dia) e hora (hora, minuto e segundo) de geração do script.
  + Ex.: db\_loja\_2017\_05\_10\_08\_50\_32.sql

## Definindo a solução

### Evitar duplicidade de tabelas e ou campos

* Antes de criar uma tabela, procurar por tabela já existente.
* Antes de criar uma coluna, verificar se uma coluna com a informação já existe e utilizá-la (se for o caso).

### Definir chave primária para tabela

* Cada tabela deve ter uma chave primária.
* A chave primária de uma tabela pode ser composta por mais de uma coluna.

### Manter consistência dos objetos

* Avaliar os impactos da criação de objetos e/ou colunas no sistema como um todo.
* Não excluir objetos nem colunas sem prévia autorização do responsável pelo projeto ou por pessoa designada para tal.
* Seguir o padrão de nomenclatura para tabelas e colunas, contido neste documento.

### Elaborar e manter a documentação

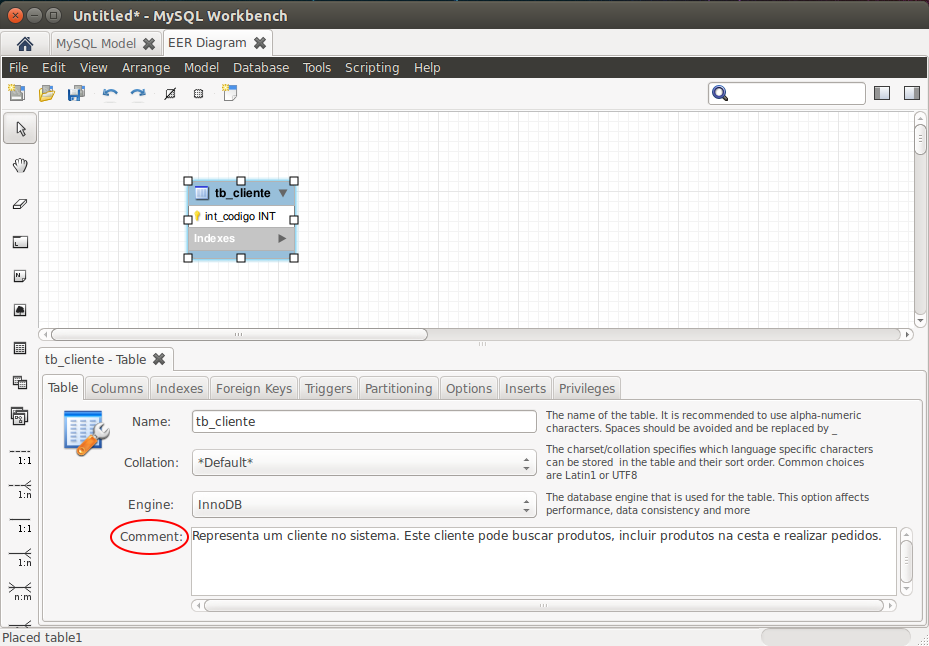
* Ao criar um objeto (tabela/visão), documentar com pelo menos as seguintes informações: breve descrição do objeto.
* Ao criar uma coluna em um objeto, documentar com pelo menos as seguintes informações: breve descrição da coluna, domínio dos dados da coluna (por exemplo, intervalo de valores possíveis) e exemplos de valores válidos para a coluna (pelo menos 3).
  + Ex.: Coluna int\_idade na tabela tb\_cliente
    - Breve descrição: representa a idade do cliente.
    - Domínio: inteiros não negativos.
    - Exemplos: 0 (idade indeterminada), 1, 2, 90.
* Ao fazer qualquer alteração no projeto do banco de dados, verificar e manter a documentação do objeto e/ou coluna afetada.

## Implementação da Solução

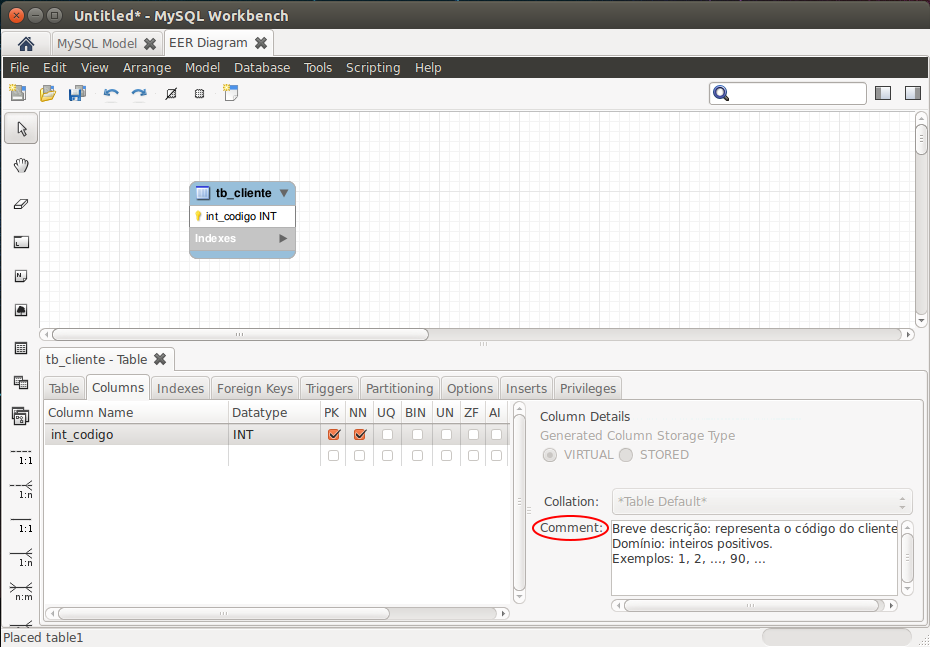
Uma vez definida a solução, seguem os passos para implementá-la.

### Criando Diagrama EER no MySQL Workbench

* Executar o MySQL Workbench (versão 6.3).
* Criar um modelo, selecionando o menu File > New Model.
* Criar um diagrama, selecionando o menu File > Model > Add Diagram.
* As tabelas devem ser documentadas no campo Comment, conforme figura a seguir.



* As colunas devem ser documentadas no campo Comment da coluna, conforme figura a seguir.



### Gerando script SQL no MySQL Workbench

* Exportar o modelo, selecionando o menu File > Export > Forward Engineer SQL CREATE Script...
* Em Filter Objects, selecionar as opções Export MySQL Table Objects e Export MySQL View Objects.