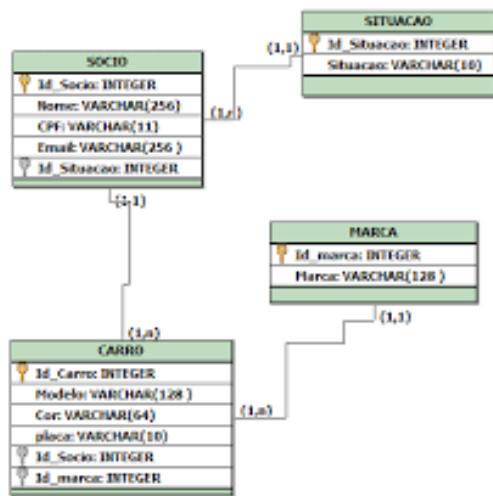


Bancos de Dados

Linguagem SQL – DDL – Data Definition Language

Constraints



Clóvis José Ramos Ferraro
cferraro@unicid.edu.br

Restrições (Constraints)

Restrições (Constraints)

Not Null

Unique

Check

Primary Key

Foreign Key

Usamos o Constrains no processo da criação do banco de dados , conhecido como CRUD.

C – Create (Criar);

R – READ (ler);

U – Update (alteração);

D – Delete (exclusão).

Restrições - Chave primária

- Possui valor único para cada linha da tabela.
- Pode ter no máximo uma chave primária por tabela.
- Pode ser simples ou composta.
- Segue o modelo:

```
CONSTRAINT <nome da primary key> PRIMARY KEY  
(coluna1, coluna2, ...)
```

(W3SCHOOLS, 2019)

Restrições

Alguns tipos comuns de restrições incluem o seguinte:

NOT NULL Constraint: Garante que uma coluna não pode ter o valor NULL.

DEFAULT Constraint: Fornece um valor padrão para uma coluna quando nenhum é especificado.

UNIQUE Constraint: Garante que todos os valores numa coluna são diferentes.

CHECK Constraint: Garante que todos os valores numa coluna satisfazem um determinado critério.

Primary Key Constraint: Utilizado para identificar de forma única uma linha na tabela.

Foreign Key Constraint: Utilizado para garantir a integridade referencial dos dados.

Restrições: Chave Primária

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA                INT                NOT  
    NULL [REDACTED],  
    NOM_PESSOA                VARCHAR(250),  
    CPF_PESSOA                VARCHAR(16),  
    RG_PESSOA                 VARCHAR(12),  
    EMAIL_PESSOA              VARCHAR(100),  
    DTA_NASC_PESSOA           DATE,  
    IDF_SEXO                   VARCHAR(1),  
    [REDACTED]  
    COD_TIPO_PESSOA           INT                ,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA)  
);
```

(W3SCHOOLS, 2019)

Restrições de CHAVE ESTRANGEIRA (MySQL)

Restrições: Chave estrangeira

- Faz o relacionamento entre uma ou mais colunas de uma tabela com a chave primária de outra tabela.
- Uma tabela pode ter várias chaves estrangeiras para outras tabelas, representando o relacionamento que possui com cada uma das outras tabelas.

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

A sintaxe essencial para definir uma restrição de chave estrangeira em uma instrução CREATE TABLE ou ALTER TABLE inclui o seguinte:

```
[CONSTRAINT [apelido]] FOREIGN KEY  
(nome da coluna, ...) REFERENCES nome_da_tabela (nome da coluna,...) [  
ON DELETE referências] [ON UPDATE referências]
```

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT

(W3SCHOOLS, 2019)

Restrições: Chave estrangeira

ON DELETE significa que uma ação referencial será executada quando um registro for **excluído** da tabela pai.

ON UPDATE indica que uma ação referencial será executada quando um registro for **modificado** na tabela pai.

CASCADE: permite excluir ou atualizar os registros relacionados presentes na tabela filha automaticamente, quando um registro da tabela pai for atualizado (ON UPDATE) ou excluído (ON DELETE). É a opção mais comum aplicada.

(W3SCHOOLS, 2019)

Restrições: Chave estrangeira

RESTRICT: Impede que ocorra a exclusão ou a atualização de um registro da tabela pai, caso ainda haja registros na tabela filha. Uma exceção de violação de chave estrangeira é retornada. A verificação de integridade referencial é realizada **antes** de tentar executar a instrução UPDATE ou DELETE

NO ACTION: Essa opção equivale à opção RESTRICT, porém a verificação de integridade referencial é executada **depois** da tentativa de alterar a tabela. É a opção padrão, aplicada caso nenhuma das opções seja definida na criação da chave estrangeira.

SET NULL: Esta opção é usada para definir com o valor NULL o campo na tabela filha quando um registro da tabela pai for atualizado ou excluído.

SET DEFAULT: “Configura Padrão” – Define um valor padrão na coluna na tabela filha, aplicado quando um registro da tabela pai for atualizado ou excluído.

(W3SCHOOLS, 2019)

Restrições: Chave estrangeira

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA          INT                                NOT  
    NULL AUTO_INCREMENT,  
    NOM_PESSOA          VARCHAR(250) ,  
    CPF_PESSOA          VARCHAR(16) ,  
    RG_PESSOA           VARCHAR(12) ,  
    EMAIL_PESSOA        VARCHAR(100) ,  
    DTA_NASC_PESSOA     DATE ,  
    IDF_SEXO            VARCHAR(1) ,  
    IDF_ATIVO           VARCHAR(1) ,  
    COD_TIPO_PESSOA     INT ,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA) ,  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
    REFERENCES TIPO_PESSOA  
    (COD_TIPO_PESSOA)
```

(W3SCHOOLS, 2019)

Restrições: Chave estrangeira

```
CREATE TABLE IF NOT EXISTS TIPO_PESSOA (  
    COD_TIPO_PESSOA                INT                NOT NULL  
    AUTO_INCREMENT,  
    DESC_TIPO_PESSOA                VARCHAR(250),  
    CONSTRAINT PK_TIPO_PESSOA PRIMARY KEY (COD_TIPO_PESSOA)  
);
```

(W3SCHOOLS, 2019)

Restrições: Chave estrangeira

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA          INT                                NOT  
    NULL AUTO_INCREMENT,  
    NOM_PESSOA          VARCHAR(250) ,  
    CPF_PESSOA          VARCHAR(16) ,  
    RG_PESSOA           VARCHAR(12) ,  
    EMAIL_PESSOA        VARCHAR(100) ,  
    DTA_NASC_PESSOA     DATE ,  
    IDF_SEXO            VARCHAR(1) ,  
    IDF_ATIVO           VARCHAR(1) ,  
    COD_TIPO_PESSOA     INT ,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA) ,  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
    REFERENCES TIPO_PESSOA  
    (COD_TIPO_PESSOA)
```

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

```
CREATE TABLE produto (  
  categoria INT NOT NULL,  
  id INT NOT NULL,  
  preco DECIMAL,  
  PRIMARY KEY(categoria, id) )  
ENGINE=INNODB;
```

```
CREATE TABLE cliente (  
  id INT NOT NULL,  
  PRIMARY KEY (id) )  
ENGINE=INNODB;
```

(MYSQL, 2024)

DDL – Create Table (MySQL)

```
CREATE TABLE vendas (  
no INT NOT NULL AUTO_INCREMENT,  
Categoria_produto INT NOT NULL,  
produto_id INT NOT NULL,  
cliente_id INT NOT NULL,  
PRIMARY KEY(no),  
  
FOREIGN KEY (categoria_produto, produto_id) REFERENCES produto  
(categoria, id) ON UPDATE CASCADE ON DELETE RESTRICT,  
  
FOREIGN KEY (cliente_id) REFERENCES cliente(id) ) ENGINE=INNODB;;
```

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

```
CREATE TABLE usuariorelsupermercado (  
idUsuario INT(11) NOT NULL,  
idSupermercado INT(11) NOT NULL,
```

```
CONSTRAINT fk_supermercado_rel FOREIGN KEY (idSupermercado)  
REFERENCES supermercado (idSupermercado)  
ON UPDATE NO ACTION ON DELETE NO ACTION,
```

```
CONSTRAINT fk_usuario_rel FOREIGN KEY (idUsuario) REFERENCES usuario  
(idUsuario) ON UPDATE NO ACTION ON DELETE NO ACTION)
```

```
ENGINE=InnoDB;
```

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

Você possui uma tabela de Cidade e um de Cliente, Supondo:

Cidades:

Id	Estado
S P	São paulo

Cliente:

Id_cliente	Nome do cliente	Id_estado
1	Fulano de tal	SP

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

Ao atualizar / excluir o registro da cidade de são paulo:

RESTRICT/NO ACTION: o banco irá rejeitar o comando, retornando uma exceção de violação de chave estrangeira.

CASCADE: Se alterar o valor da coluna id da Cidade, também será alterado o valor da coluna Cidade_id na tabela de Cliente. Se excluir a cidade São Paulo, também será excluído o Cliente 1, Fulano.

SET NULL: O valor da coluna Cidade_Id dos registros que estiverem utilizando o valor 1, São Paulo, serão definidos como null.

(W3SCHOOLS, 2019)

Exemplo

NULL
e
NOT NULL

Restrições: NULL e NOT NULL

- NULL (Padrão) e NOT NULL
 - Para que valor na coluna seja preenchida obrigatoriamente: NOT NULL

Atenção:

- Alguns SGBDs aceitam NULL ou NULLABLE para os atributos que podem ficar vazios.

(W3SCHOOLS, 2019)

Apagando uma tabela

```
DROP TABLE TIPO_PESSOA
```

Apagando uma tabela

`DROP TABLE PESSOA`

Restrições: NULL e NOT NULL

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA                INT                NOT NULL,  
    NULL [REDACTED],  
    NOM_PESSOA                VARCHAR(250)        NOT NULL,  
    CPF_PESSOA                VARCHAR(16)         Campos que podem ficar vazios  
    RG_PESSOA                 VARCHAR(12) ,  
    EMAIL_PESSOA              VARCHAR(100) ,  
    DTA_NASC_PESSOA           DATE                NOT NULL,  
    IDF_SEXO                   VARCHAR(1)          NOT NULL,  
    IDF_ATIVO                  VARCHAR(1)          NOT NULL,  
    COD_TIPO_PESSOA           INT                NOT NULL,  
    NULL,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA),  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
        REFERENCES TIPO_PESSOA (COD_TIPO_PESSOA)
```

) ;

(W3SCHOOLS, 2019)

UNIQUE

Restrições: Unique

- Podemos ter mais de uma restrição UNIQUE por tabela.
- Especifica uma restrição de unicidade sobre valores das colunas de tabelas.

CLIENTE				
<u>Numero</u>	CPF	RG	UF	Nome

```
CONSTRAINT <nome da unique key> UNIQUE  
(coluna1, coluna2, ...)
```

(W3SCHOOLS, 2019)

Restrições: Unique

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA                INT  
        NOT NULL ██████████,  
    NOM_PESSOA                VARCHAR(250)        NOT  
    NULL,  
    CPF_PESSOA                VARCHAR(16) ,  
    RG_PESSOA                 VARCHAR(12) ,  
    EMAIL_PESSOA              VARCHAR(100) ,  
    DTA_NASC_PESSOA           DATE                NOT  
    NULL,  
    IDF_SEXO                  VARCHAR(1)          NOT  
    NULL,  
    IDF_ATIVO                  VARCHAR(1)  
        NOT NULL,  
    COD_TIPO_PESSOA           INT  
        NOT NULL,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA) ,  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
        REFERENCES TIPO_PESSOA (COD TIPO_PESSOA) ,
```

(W3SCHOOLS, 2019)

CHECK

Restrições: Check

- A avaliação do critério de pesquisa deve usar uma expressão Booleana (true/false) como base e não pode fazer referência a outra tabela.
- A restrição CHECK no nível de coluna pode fazer referência somente à coluna restrita.

(W3SCHOOLS, 2019)

Restrições: Check

- Restrições CHECK oferecem a mesma função de validação dos dados durante instruções INSERT e UPDATE.
- Se existirem uma ou mais restrições CHECK para uma coluna, todas as restrições serão avaliadas.
- `CONSTRAINT <nome da regra> CHECK`
`(<coluna com expressão booleana>)`

(W3SCHOOLS, 2019)

Restrições: Check

O operador **LIKE** é usado em uma condição para procurar um padrão especificado.

Exemplo: PAIS **LIKE** 'Espanha'

O operador **IN** testar vários valores de uma lista:

Exemplo: PAIS **IN** ('Alemanhã', 'França', 'Reino Unido');

PAIS **Not**
IN ('Alemanhã', 'França', 'Reino Unido');

(W3SCHOOLS, 2024)

Restrições: Check

```
CONSTRAINT ckIdade CHECK (Idade <= 100)
```

```
CONSTRAINT ckTaxa CHECK (Taxa >= 1 and Taxa <= 5)
```

```
CONSTRAINT CK_carreira_id CHECK (carreira_id LIKE  
'C[0-9][0-9]')
```

C12

B23

```
CONSTRAINT CK_carreira_id CHECK (carreira_id IN  
( 'C01', 'C11', 'C12', 'C15', 'C17') OR carreira_id  
LIKE '9[0-9][0-9]')
```

(W3SCHOOLS, 2019)

Restrições: Check

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA                INT  
        NOT NULL ██████████ ,  
    NOM_PESSOA                VARCHAR(250) NOT  
    NULL ,  
    CPF_PESSOA                VARCHAR(16) ,  
    RG_PESSOA                 VARCHAR(12) ,  
    EMAIL_PESSOA              VARCHAR(100) ,  
    DTA_NASC_PESSOA           DATE NOT  
    NULL ,  
    IDF_SEXO                  VARCHAR(1)  
        NOT NULL ,  
    IDF_ATIVO                  VARCHAR(1)  
        NOT NULL ,  
    COD_TIPO_PESSOA           INT  
        NOT NULL ,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA) ,  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
        REFERENCES TIPO_PESSOA (COD_TIPO_PESSOA) ,  
    CONSTRAINT UQ_PESSOA_CPF UNIQUE (CPF_PESSOA) ,  
    CONSTRAINT UQ_PESSOA_RG  UNIQUE (RG_PESSOA) ,
```

(W3SCHOOLS, 2019)

DEFAULT

Restrições: Default

- A cláusula DEFAULT especifica o valor fornecido para a coluna quando um valor não for fornecido explicitamente durante uma inserção.
- Somente um valor constante, como uma cadeia de caracteres, números, datas, função de escalar ou NULL, pode ser usado como padrão.

(W3SCHOOLS, 2019)

Restrições: Default

- As definições DEFAULT podem ser aplicadas a qualquer coluna, com exceção as colunas com a propriedade IDENTITY / AUTO_INCREMENT.
- Se um valor padrão for especificado para uma coluna de tipo definido pelo usuário, o tipo deverá oferecer suporte a uma conversão implícita.

(W3SCHOOLS, 2019)

DDL – Create Table (MySQL)

```
CREATE TABLE IF NOT EXISTS PESSOA (  
    COD_PESSOA                INT                NOT NULL  
    ██████████,   
    NOM_PESSOA                VARCHAR(250)        NOT NULL,  
    CPF_PESSOA                VARCHAR(16),  
    RG_PESSOA                  VARCHAR(12),  
    EMAIL_PESSOA              VARCHAR(100),  
    DTA_NASC_PESSOA           DATE                NOT NULL,  
    IDF_SEXO                   VARCHAR(1),  
    IDF_ATIVO                  VARCHAR(1)          DEFAULT  
    'S',  
    COD_TIPO_PESSOA           INT                NOT NULL,  
    CONSTRAINT PK_PESSOA PRIMARY KEY (COD_PESSOA),  
    CONSTRAINT FK_TIPO_PESSOA FOREIGN KEY (COD_TIPO_PESSOA)  
        REFERENCES TIPO_PESSOA (COD_TIPO_PESSOA),  
    CONSTRAINT UQ_PESSOA_CPF UNIQUE (CPF_PESSOA),  
    CONSTRAINT UQ_PESSOA_RG  UNIQUE (RG_PESSOA),  
    CONSTRAINT UQ_PESSOA_EMAIL UNIQUE (EMAIL_PESSOA),  
    CONSTRAINT CK_SEXO CHECK (IDF_SEXO IN ('F','M')),  
    CONSTRAINT CK_ATIVO CHECK (IDF_ATIVO IN ('S','N'))  
);
```

(W3SCHOOLS, 2019)

Geração Automática de Números (MySQL)

Geração de números automática

- Podemos precisar que uma determinada coluna GERE números automaticamente, como por exemplo, RGM.
- Os bancos de dados possuem funções específicas para geração de números e podem ser associadas a uma coluna.
- O MySQL permite que qualquer tabela tenha um campo com autoenumeração, mas apenas uma coluna da tabela pode receber essa funcionalidade: **AUTO_INCREMENT**

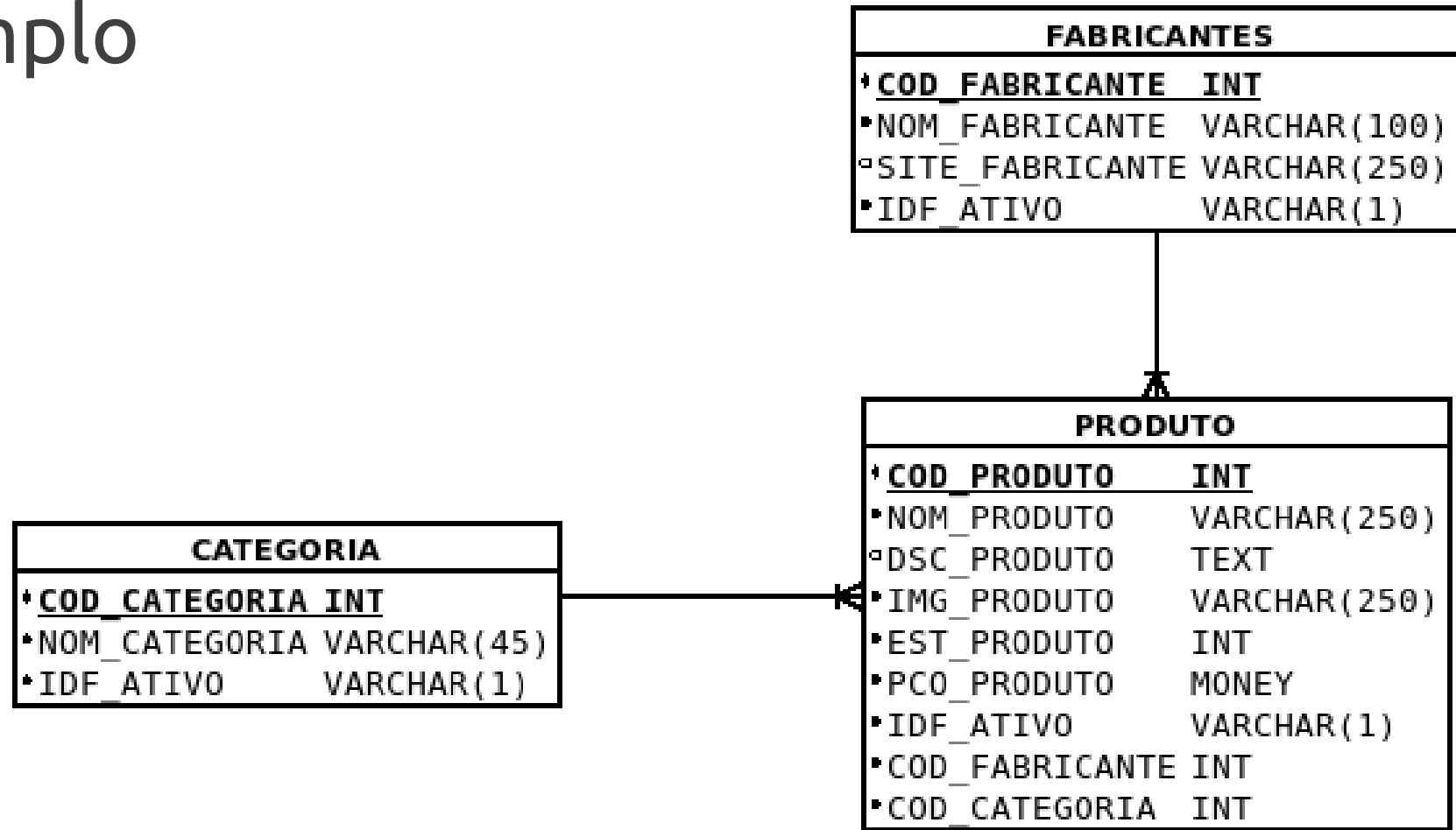
(W3SCHOOLS, 2019)

Identity (SQL SERVER)

```
Create table clientes (  
Codigo int identity(1,1) primary key,  
Nome varchar(200),  
Uf varchar (2));
```

(W3SCHOOLS, 2019)

Exemplo



FABRICANTES	
* <u>COD_FABRICANTE</u>	INT
▪ NOM_FABRICANTE	VARCHAR(100)
▫ SITE_FABRICANTE	VARCHAR(250)
▪ IDF_ATIVO	VARCHAR(1)

Exemplo

```

20 CREATE TABLE FABRICANTES (
21     COD_FABRICANTE      INT                NOT NULL AUTO_INCREMENT,
22     NOM_FABRICANTE      VARCHAR(100)        NOT NULL,
23     SITE_FABRICANTE     VARCHAR(250),
24     IDF_ATIVO           VARCHAR(01)         DEFAULT 'S',
25     CONSTRAINT PK_FABRICANTES PRIMARY KEY (COD_FABRICANTE),
26     CONSTRAINT CK_FABRICANTES_ATIVO CHECK (IDF_ATIVO IN ('S','N'))
27 ) ENGINE = INNODB;

```

CATEGORIA	
* <u>COD_CATEGORIA</u>	INT
* NOM_CATEGORIA	VARCHAR(45)
* IDF_ATIVO	VARCHAR(1)

Exemplo

```

4 CREATE TABLE CATEGORIA (
5     COD_CATEGORIA    INT                NOT NULL AUTO_INCREMENT,
6     NOM_CATEGORIA    VARCHAR(45) NOT NULL,
7     IDF_ATIVO        VARCHAR(01) DEFAULT 'S',
8     CONSTRAINT PK_CATEGORIA PRIMARY KEY (COD_CATEGORIA),
9     CONSTRAINT CK_CATEGORIA_ATIVO CHECK (IDF_ATIVO IN ('S','N'))
10 ) ENGINE = INNODB;
```

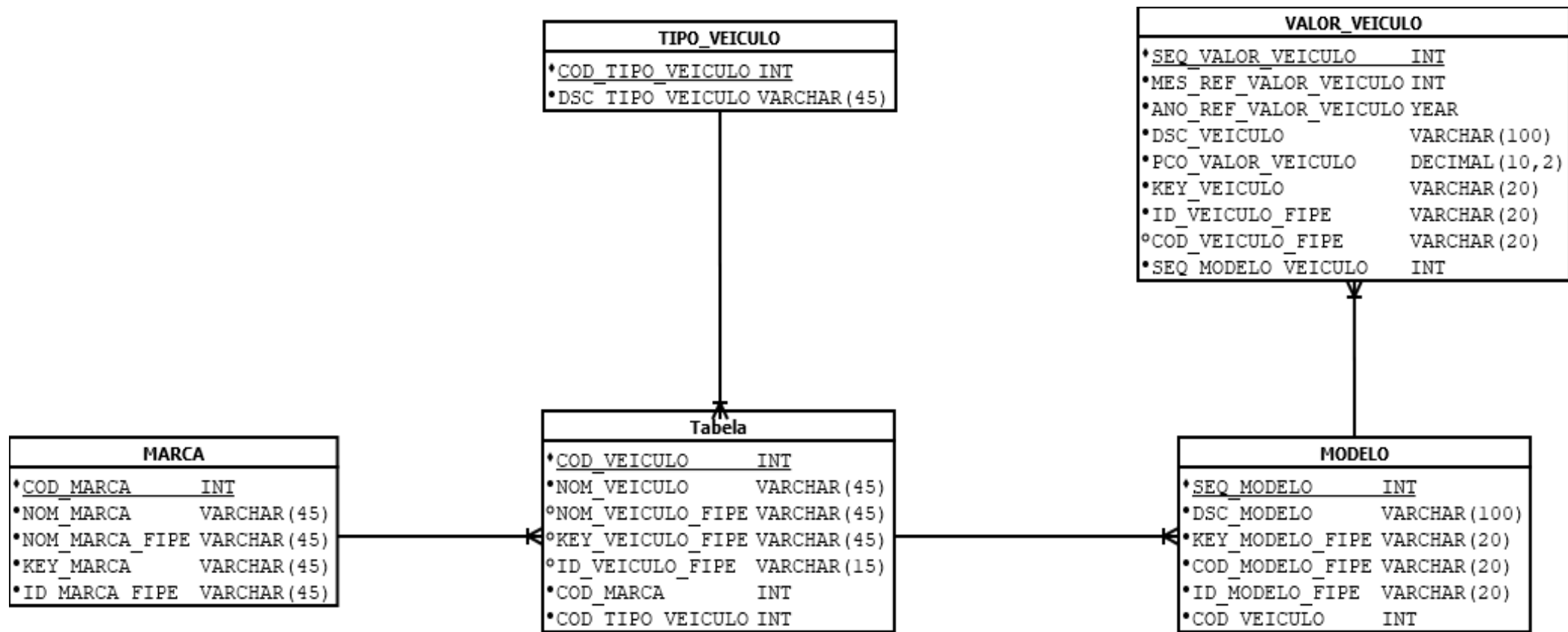
Exemplo

```
51 CREATE TABLE PRODUTO (  
52     COD_PRODUTO          INT          NOT NULL AUTO_INCREMENT,  
53     NOM_PRODUTO          VARCHAR(250)  NOT NULL,  
54     DSC_PRODUTO          LONGTEXT,  
55     IMG_PRODUTO          VARCHAR(250),  
56     EST_PRODUTO          INT           DEFAULT 0,  
57     PCO_PRODUTO          DECIMAL(10,2) DEFAULT 0,  
58     IDF_ATIVO            VARCHAR(01)   DEFAULT 'S',  
59     COD_CATEGORIA        INT          NOT NULL,  
60     COD_FABRICANTE       INT          NOT NULL,  
61     CONSTRAINT PK_PRODUTO PRIMARY KEY (COD_PRODUTO),  
62     CONSTRAINT FK_PRODUTO_2_CATEGORIA FOREIGN KEY (COD_CATEGORIA) REFERENCES CATEGORIA (COD_CATEGORIA),  
63     CONSTRAINT FK_PRODUTO_2_FABRICANTE FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTE (COD_FABRICANTE),  
64     CONSTRAINT CK_PRODUTO_ATIVO CHECK (IDF_ATIVO IN ('S','N'))  
65 ) ENGINE = INNODB;
```

PRODUTO	
▸ <u>COD_PRODUTO</u>	<u>INT</u>
▸ NOM_PRODUTO	VARCHAR(250)
▸ DSC_PRODUTO	TEXT
▸ IMG_PRODUTO	VARCHAR(250)
▸ EST_PRODUTO	INT
▸ PCO_PRODUTO	MONEY
▸ IDF_ATIVO	VARCHAR(1)
▸ COD_FABRICANTE	INT
▸ COD_CATEGORIA	INT

Exercício: Tabela FIPE

Crie as tabelas do diagrama abaixo:



Referências

Referências

- ❖ DEVMEDIA. **Trabalhando com Campos auto incremento (Identity) no SQL Server**. Disponível em: <<https://www.devmedia.com.br/trabalhando-com-campos-auto-incremento-identity-no-sql-server/17974>>. Acesso em: 06 mar. 2024.
- ❖ ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistema de Banco de Dados**. 6. ed. São Paulo: Pearson Education, 2011.
- ❖ MICROSOFT. **CREATE TABLE**: Transact-SQL. 2019. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-table-transact-sql?view=sql-server-2017>>. Acesso em: 18 jun. 2019.
- ❖ MICROSOFT. **Tipos de dados**: Transact-SQL. 2017. Disponível em: <<https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>>. Acesso em: 18 jun. 2019.
- ❖ RANGEL, Alexandre Leite et al (Ed.). **BANCO DE DADOS**. Batatais: Claretiano, 2015. 254 p.
- ❖ RANGEL, Alexandre Leite. **LINGUAGEM SQL**. Batatais: Claretiano, 2016. 97 p.

Referências

- ❖ RANGEL, Alexandre Leite. Construindo um Banco de Dados. In: RANGEL, Alexandre Leite. **MySQL** - Projeto, Modelagem e Desenvolvimento de Bancos de Dados. Rio de Janeiro: Alta Books, 2004. Cap. 5. p. 38-47.
- ❖ W3SCHOOLS. **SQL AUTO INCREMENT Field**. Disponível em: <https://www.w3schools.com/sql/sql_autoincrement.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL CHECK Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_check.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL Constraints**. Disponível em: <https://www.w3schools.com/sql/sql_constraints.asp>. Acesso em: 18 jun. 2019.

Referências

- ❖ W3SCHOOLS. **SQL DEFAULT Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_default.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL FOREIGN KEY Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_foreignkey.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL NOT NULL Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_notnull.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL PRIMARY KEY Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_primarykey.asp>. Acesso em: 18 jun. 2019.

Referências

- ❖ W3SCHOOLS. **SQL UNIQUE Constraint**. Disponível em: <https://www.w3schools.com/sql/sql_unique.asp>. Acesso em: 18 jun. 2019.
- ❖ _____. **SQL IN**. Disponível em: <https://www.w3schools.com/sql/sql_ref_in.asp>. Acesso em: 06 mar. 2023.
- ❖ _____. **SQL LIKE OPERATOR**. Disponível em: <https://www.w3schools.com/sql/sql_like.asp>. Acesso em: 06 mar. 2023.
- ❖ _____. **SQL LIKE OPERATOR**. Disponível em: <https://www.w3schools.com/sql/sql_like.asp>. Acesso em: 06 mar. 2023.

Obrigado!

Prof. Clóvis José Ramos Ferraro

(Adaptado de Alexandre Rangel)

```
CREATE TABLE ANIMAIS (  
id MEDIUMINT NOT NULL AUTO_INCREMENT,  
nome CHAR(30) NOT NULL,  
PRIMARY KEY (id));  
ALTER TABLE ANIMAIS AUTO_INCREMENT =  
3509;  
INSERT INTO ANIMAIS (nome) VALUES  
(‘CACHORRO’), (‘GATO’), (‘PINGUIM’),  
(‘COELHO’), (‘MACACO’), (‘BALEIA’);  
SELECT * FROM ANIMAIS;
```