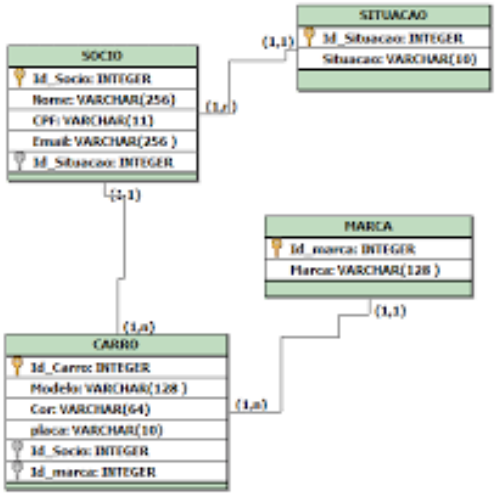
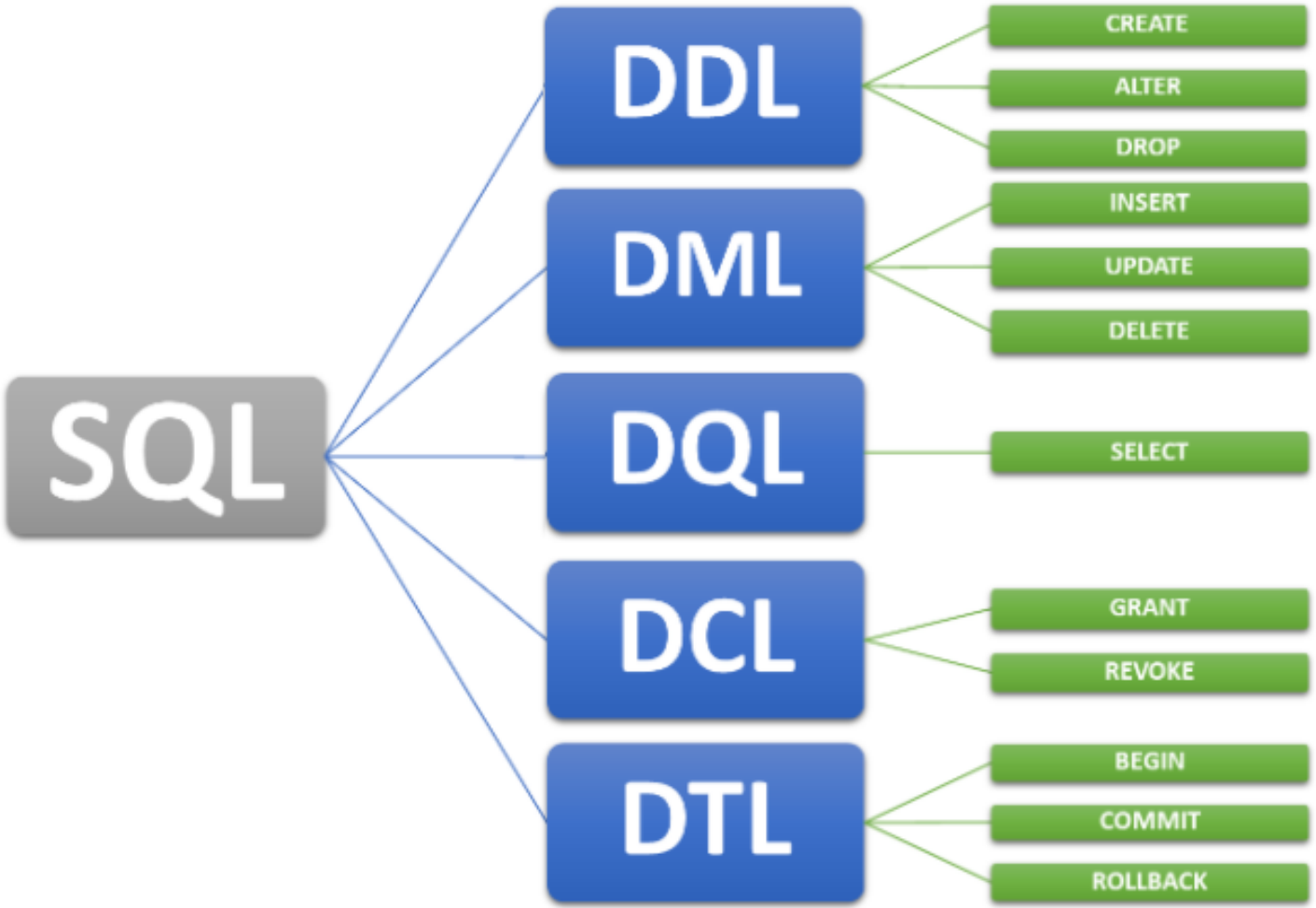
Bancos de Dados

Linguagem SQL – DDL – Data Definition Language



Clóvis José Ramos Ferraro
cferraro@unicid.edu.br

SQL - Structured Query Language


SQL
DDL
CREATE
ALTER
DROP
DML
INSERT
UPDATE
DELETE
DQL
SELECT
DCL
GRANT
REVOKE
DTL
BEGIN
COMMIT
ROLLBACK

SQL - Structured Query Language

- ✔ Linguagem Estruturada de Consulta - Padrão para os SGBDs.
- ✔ Foi desenvolvida nos anos 70 nos laboratórios da IBM, em San Jose.
- ✔ Divide-se em:
  - ❖ DDL – Data Definition Language – Linguagem de Definição de Dados
  - ❖ DML – Data Manipulation Language – Linguagem de Manipulação de Dados
  - ❖ DQL – Data Query Language – Linguagem de Consulta de Dados
  - ❖ DCL – Data Control Language – Linguagem de Controle de Dados
  - ❖ DTL – Data Transaction Language – Linguagem de Transação de Dados

DDL – Data Definition Language – Linguagem de Definição de Dados

- CREATE
  - DATABASE
  - TABLE
  - VIEW
  - USER
- DROP
- ALTER
- TRUNCATE
- COMMENT
- RENAME

DML – Data Manipulation Language – Linguagem de Manipulação de Dados

- INSERT
- DELETE
- UPDATE
- CALL
- EXPLAINPLAN
- LOCK TABLE

DQL – Data Query Language – Linguagem de Seleção de Dados

- SELECT

DCL – Data Control Language – Linguagem de Controle de Dados

- GRANT
- REVOQUE
- DENY

DTL – Data Transaction Language – Linguagem de Transação de Dados

- COMMIT
- SAVEPOINT
- ROLLBACK

Tipos de Dados

Tipos de Dados

- ✔ Bancos de dados associam tipos de dados a colunas, expressões, variáveis e parâmetros.
- ✔ Tipos de dados determinam quais os tipos de valores serão permitidos no armazenamento.
- ✔ Todos os dados são armazenados nos banco de dados em formato de bytes. Essa é a forma como os computadores trabalham, ou seja, quando irão representar a letra A, na realidade, armazenam o código binário 01000001 que a representa.
- ✔ Quando esse dados precisa ser mostrado, o BD traduz o formato binário gravado, na letra A.

Tipos de Dados

Tabela ASCII

ASCII TABLE

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

Fonte: https://upload.wikimedia.org/wikipedia/commons/thumb/d/dd/ASCII-Table.svg/1000px-ASCII-Table.svg.png

UNICODE

O consórcio Unicode de empresas de informática atribuiu *nomes numéricos* (conhecidos como *code points*) a mais de 1 milhão de caracteres. Segue uma minúscula amostra da lista de caracteres e seus números:

número Unicode	caractere
33	!
34	"
45	-
57	9
65	A
66	B
97	a
98	b
126	~
192	À
227	ã

Tipos de Dados

- ✔ Tipos de dados determinam quais os tipos de valores serão permitidos no armazenamento.
- ✔ Principais tipos são agrupados em categorias:

Categoria
Numéricos exatos
Numéricos aproximados
Data e hora
Cadeias de caracteres (Strings)
Caractere Unicode
Binários
Outros tipos

Caracter

TIPO	DESCRIÇÃO
char	String de caracteres de tamanho fixo, máximo de 8000 caracteres.
varchar	String de caracteres de tamanho variável, máximo de 8000 caracteres.
nchar	Dados Unicode de tamanho fixo, máximo de 4000 caracteres
nvarchar	Dados Unicode de tamanho variável, máximo de 4000 caracteres
text	Cadeia de caracteres de tamanho variável. Até 2 GB de dados

Caracter

Tipo de texto	Máximo de bytes
Tinytext	255
Text	65.535
MediumText	16.777.215
Longtext	4.294.967.295

Char	255
VarChar	65.535

Números exatos

TIPO	DESCRIÇÃO	ARMAZENAMENTO
bit	valor booleano = 0, 1 ou nulo	-
decimal ou numeric	Valores exatos, porém permite ter parte fracionária. De -10 x 10^38 +1 a 10 x 10^38 -1	5 - 17 bytes (depende da precisão desejada)

Numerais

TIPO	DESCRIÇÃO	ARMAZENAMENTO
tinyint	Números inteiros de 0 a 255	1 byte
smallint	Números inteiros de -32.768 a 32.767	2 bytes
int	Números inteiros entre -2.147.483.648 e 2.147.483.647	4 bytes
bigint	Números entre -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	8 bytes
real	Números de ponto flutuante entre -3,4 x 10^38 a 3,4x10^38 (precisão de até 15 dígitos)	4 bytes
float	Número de ponto flutuante entre -1,79 x 10^308 a 1,79 x 10^308 (precisão de até 34 dígitos)	8 bytes
money	Dados monetários de -922.337.203.685.477,5808 até 922.337.203.685.477,5807	8 bytes

Dados Numéricos

Tipo	Com sinal	Sem sinal
Tinyint	-128 a 127	0 a 255
Smallint	-32.768 a 32.767	0 a 65.535
MediumInt	-8.388.608 a 8.388.607	0 a 16.777.215
Int / Integer	-9.223.372.036.854.775. 808 a 9.223.372.036.854.775. 807	0 a 4.294.967.295

Float(p,e)	tem uma precisão de 6 casas decimais.
Double(p,e)	tem uma precisão de 10 casas decimais
Decimal(p,e)	pode armazenar no máximo 65 dígitos, sendo esses até 30 dígitos após a casa decimal.

Data e Hora

TIPO	FORMATO	INTERVALO	PRECISÃO	ARMAZENAMENTO
date	AAAA-MM-DD	0001/01/01 a 9999/12/31	1 dia	3 bytes
datetime	AAAA-MM-DD hh:mm:ss[.nnn]	1753/01/01 a 9999/12/31	0,00333 segundos	8 bytes
smalldatetime	AAAA-MM-DD hh:mm:ss	1900/01/01 a 2079/06/06	1 minuto	4 bytes
time	hh:mm:ss[.nnnnnnn]	00:00:00.0000000 a 23:59:59:9999999	100 nanossegundos	3-5 bytes

Tipos de Dados de Alguns outros SGBDs

Tipos de Dados - **Oracle**

Principais Tipos de Dados do Oracle	
Tipo	**Descrição**
CHAR	Conjunto de caracteres de tamanho fixo
VARCHAR2	Conjunto de caracteres de tamanho variável
NUMBER	Números inteiros ou de ponto flutuante
DATE	Datas e horas
CLOB	Campos texto de tamanho indefinido
BLOB	Arquivos multimídia (Imagem, som, etc)

Tipos de Dados - MySQL/MariaDB

Principais Tipos de Dados do MySQL	
Tipo	**Descrição**
CHAR	Conjunto de caracteres de tamanho fixo
VARCHAR	Conjunto de caracteres de tamanho variável
INT	Números inteiros
DOUBLE	Número de ponto flutuante
DATE	Data. Exibido como YYYY-MM-DD
TIME	Hora. Exibido como HH:MM:SS
DATETIME	Data e hora. YYYYMMDDHHMMSS
TIMESTAMP	Data e hora. Exibido como YYYY-MM-DD HH:MM:SS
LongText	Campos texto de tamanho indefinido
BLOB	Arquivos multimídia (Imagem, som, etc)

Tipos de Dados - **Firebird**

Principais Tipos de Dados do Firebird	
Tipo	**Descrição**
CHAR	Conjunto de caracteres de tamanho fixo
VARCHAR	Conjunto de caracteres de tamanho variável
INTEGER	Números inteiros
DOUBLE	Número de ponto flutuante
DATE	Data. Exibido como YYYY-MM-DD
TIME	Hora. Exibido como HH:MM:SS
TIMESTAMP	Data e hora. Exibido como YYYY-MM-DD HH:MM:SS
BLOB	Arquivos multimídia (Imagem, som, etc)

Tipos de Dados - **PostgreSQL**

Principais Tipos de Dados do PostgreeSQL	
Tipo	**Descrição**
tinyint	0 até 256
smallint	-32768 to +32767
integer	-2147483648 to +2147483647
decimal	no limit
numeric	no limit
real	6 decimal digits precision
double	15 decimal digits precision
varchar(n)	comprimento variável com limite
character(n)	comprimento fixo, completado com bracos
char(n)	comprimento fixo, completado com bracos
text	comprimento variável não limitado
Timestamp	tanto data quanto hora
Interval	intervalos de tempo
Date	somente datas
Time	somente a hora do dia

MySql

- ✔ MySQL Community Server
  - ❖ XAMPP https://www.apachefriends.org/pt_br/index.html
- ✔ MySQL Workbench
  - ❖ https://dev.mysql.com/downloads/workbench/
- ✔ Db4free.net

  https://www.db4free.net/index.php?language=pt

Linhas de Comentário

```
-- Comentario
#Comentario
/*
Múltiplas linhas de Comentario
*/
```

Create Database

DDL – Create Database

Criando um Schema/Banco no banco de dados

```
CREATE DATABASE [IF NOT EXISTS]<NOME_BANCO>;

USE <NOME_BANCO>;
```

Create Table

Sintaxe Simplificada

DDL – Create Table

CREATE TABLE [IF NOT EXISTS]<NOME_TABELA>
("coluna 1" "tipo_dados_para_coluna_1",
"coluna 2" "tipo_dados_para_coluna_2",
...);

(MICROSOFT, 2019)

DDL – Create Table

CREATE TABLE IF NOT EXISTS PESSOA
(Primeiro_nome char(50),
Sobrenome char(50),
Endereco char(50),
Cidade char(50),
Estado char(25),
DataNascimento datetime);

(MICROSOFT, 2019)

DDL – Create Table

```
CREATE TABLE clientes (
        cli_codigo INT,
        cli_nome VARCHAR(30),
        cli_cpf CHAR(12),
        cli_data_nasc DATE,
        cli_sexo CHAR(1),
        cli_email VARCHAR(50)
);
```

DDL – Create Table

```
CREATE TABLE filmes (
        fil_codigo INT,
        fil_titulo VARCHAR(40),
        fil_genero VARCHAR(15),
        fil_duracao TIME,
        fil_situacao VARCHAR(12),
        fil_preco NUMERIC(3,2)
);
```

Show

Show Databases

Show Tables

Describe

Desc nome_tabela

Describe nome_tabela

DDL – Manipulação de tabelas

Podemos colocar restrições para limitar o tipo de dados a introduzir numa tabela.

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

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

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

(https://www.1keydata.com, 2023)

DDL – Manipulação de tabelas

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

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

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

(https://www.1keydata.com/, 2023)

DDL – Create Table

```
CREATE TABLE [IF NOT EXISTS]<NOME_TABELA> (
        CAMPO1      TIPODADO [NOT NULL],
        CAMPO2      TIPODADO [DEFAULT <VALOR>],
        CAMPOn      TIPODADO [NOT NULL]
                                [CHECK (CONDICAO)],
        [PRIMARY KEY (CAMPO1[, CAMPOn])],
        [FOREIGN KEY (CPO1[, CAPOn]) REFERENCES
                <TABELA_ORIGEM> (CPO1[, CPOn])]
)
```

(MICROSOFT, 2019)

DDL – Create Table

CREATE TABLE IF NOT EXISTS PESSOA
(Primeiro_nome char(50),
Sobrenome char(50) NOT NULL ,
Endereco char(50),
Cidade char(50),
Idade integer CHECK (idade > 18),
Estado char(25) DEFAULT ‘São Paulo’,
DataNascimento date);

(MICROSOFT, 2019)

DDL – Create Table

```
CREATE TABLE IF NOT EXISTS EXEMPLO (
id int UNSIGNED NOT NULL AUTO_INCREMENT,
numero double NOT NULL,
PRIMARY KEY (id) )
ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8
```

- Estado (SiglaEst, NomeEst)

```
CREATE TABLE ESTADO (
   SIGLA_EST VARCHAR(02) NOT NULL,
   NOME_EST  VARCHAR(100) NOT NULL,
   PRIMARY KEY (SIGLA_EST)
  );
```

- Municipio (CodMun, NomeMun, SiglaEst-CE)

```
CREATE TABLE MUNICIPIO (
        COD_MUN    INT           NOT NULL,
        NOME_MUN   VARCHAR(100) NOT NULL,
        SIGLA_EST  VARCHAR(02),
        PRIMARY KEY (COD_MUN),
        CONSTRAINT FK_EST FOREIGN KEY (SIGLA_EST) REFERENCES
ESTADO (SIGLA_EST)
```

Exemplos em MySQL

DDL – Create Table (MySQL)

```
CREATE TABLE IF NOT EXISTS TIPO_PESSOA (
    COD_TIPO_PESSOAINT,
    DSC_TIPO_PESSOA VARCHAR(250)
);
```

DDL – DROP Table (MySQL)

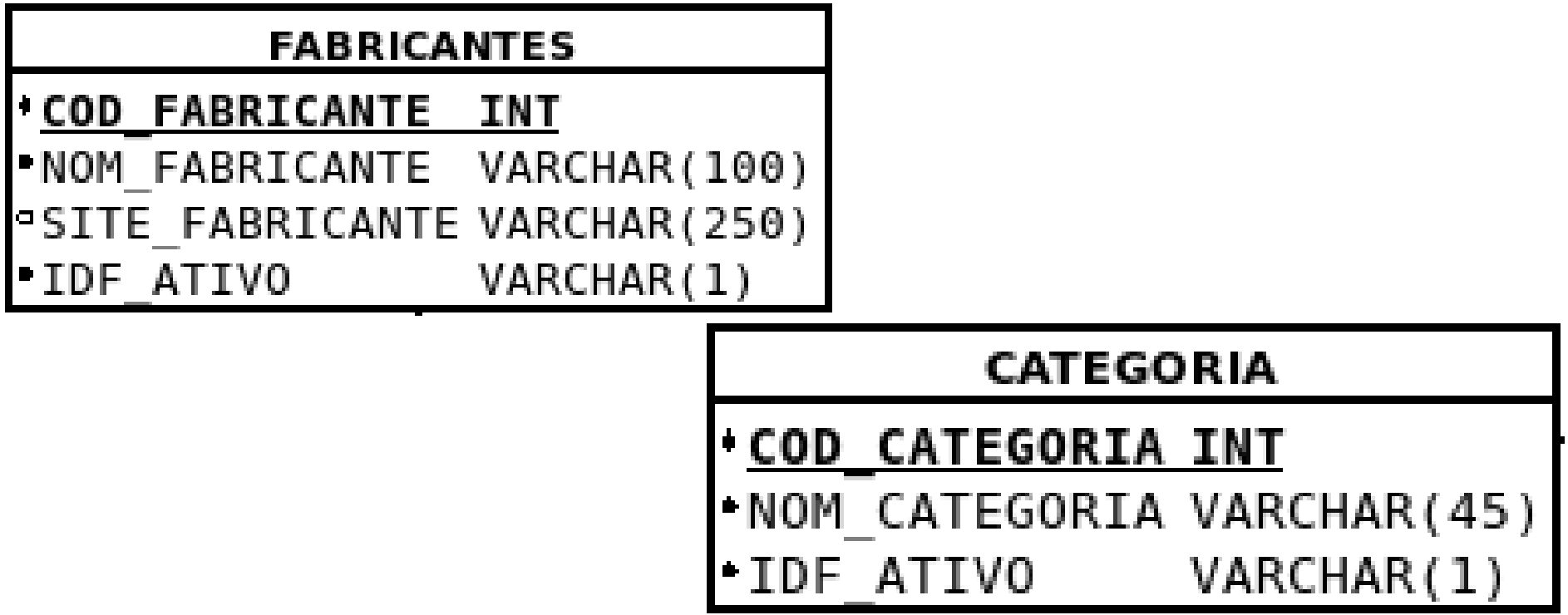
```
CREATE TABLE IF NOT EXISTS PESSOA (
	COD_PESSOA 			INT	,
	NOM_PESSOA 			VARCHAR(250),
	CPF_PESSOA 			VARCHAR(16),
	RG_PESSOA 			VARCHAR(12),
	EMAIL_PESSOA 			VARCHAR(100),
	DTA_NASC_PESSOA 		DATE,
	IDF_SEXO 			VARCHAR(1),
	COD_TIPO_PESSOAINT
);
```

DDL – Drop Table (MySQL)

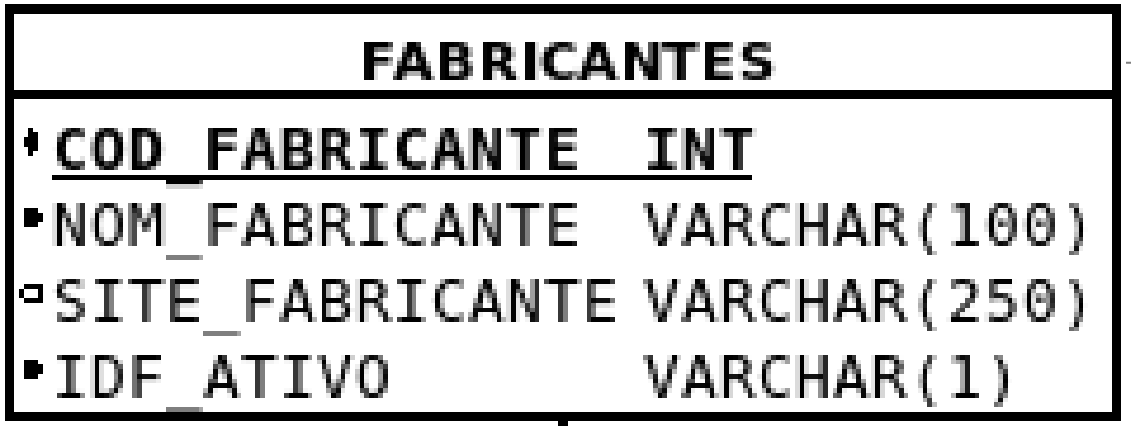
```
DROP TABLE IF EXISTS TIPO_PESSOA;
```

Exercícios

Exercícios

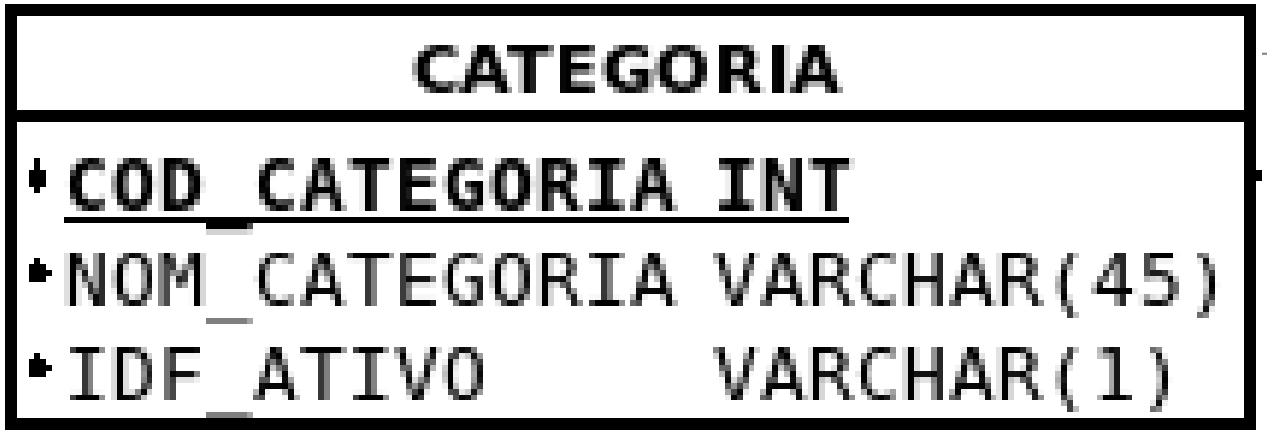

FABRICANTES
COD_FABRICANTE INT
NOM_FABRICANTE VARCHAR(100)
SITE_FABRICANTE VARCHAR(250)
IDF_ATIVO VARCHAR(1)
CATEGORIA
COD_CATEGORIA INT
NOM_CATEGORIA VARCHAR(45)
IDF_ATIVO VARCHAR(1)

Exercícios




```
1  CREATE TABLE FABRICANTES (
2      COD_FABRICANTE  INT,
3      NOM_FABRICANTE  VARCHAR(100),
4      SITE_FABRICANTE VARCHAR(250),
5      IDF_ATIVO       VARCHAR(1)
6  );
```

Exercícios




```
 8  CREATE TABLE CATEGORIA (
 9      COD_CATEGORIA   INT,
10      NOM_CATEGORIA   VARCHAR(45),
11      IDF_ATIVO       VARCHAR(1)
12  );
```

Mapeamento Completo do Ônibus

- Linha (Numero, Nome)
- Funcionario (Matricula, Nome, Cargo)
- Onibus (Prefixo, Placa, Chassis, Carroçaria, EmPe, Sentados)
- Opera (Prefixo-CE, NumeroLinha-CE, Data)
- Trabalha (Prefixo-CE, Matricula-CE, Data, Funcao)

```
CONSTRAINT FK_chave FOREIGN KEY (campo)
           REFERENCES tabela (campo)
```

Referências

W3School - https://www.w3schools.com/sql/

https://www.1keydata.com/pt/sql/sql-chave-externa.php

Obrigado!

Prof. Clóvis José Ramos Ferraro

(Adaptado de Alexandre Rangel)