

SQL

STRUCTURED QUERY LANGUAGE
LINGUAGEM DE CONSULTA ESTRUTURA

AGENDA

- Introdução
- SQL introdução
- SCHEMA
- CREATE
- TIPOS DE DADOS
- CHECK



INTRODUÇÃO

- Uma das grandes responsáveis pelo sucesso dos SGBD
- Por ter um padrão, facilita a migração entre diferentes sistemas
- Cada SGBD pode ter especificações, mas sempre há o padrão SQL
- Foi criada pela IBM Research
- A criação da SQL foi esforço do American National Standards Institute (ANSI) e da International Standard Organization (ISO)
- Houveram vários padrões ao longo do tempo

DEFINIÇÕES E TIPOS DE DADOS EM SQL



| SQL | MODELO RELACIONAL |
|--------|-------------------|
| TABELA | RELAÇÃO |
| LINHA | TUPLA |
| COLUNA | ATRIBUTO |

SCHEMA

- Agrupamento de tabela e outras construções que pertencem a mesma aplicação de BD
- Geralmente os SGBD possuem um SCHEMA padrão.
- Podem ser criados outros geralmente por qualquer usuário
- Padrão:

```
CREATE SCHEMA EMPRESA AUTHORIZATION 'jsilva'
```

CREATE TABLE

- Especificar uma nova relação dando-lhe um nome e especificando seus atributos e restrições iniciais.
- Após a criação, usar o comando ALTER TABLE para modificações.

CREATE TABLE FUNCIONARIO

| | | |
|---------------------|----------------|-----------|
| (Primeiro_nome | VARCHAR(15) | NOT NULL, |
| Nome_meio | CHAR, | |
| Ultimo_nome | VARCHAR(15) | NOT NULL, |
| Cpf | CHAR(11), | NOT NULL, |
| Data_nascimento | DATE, | |
| Endereco | VARCHAR(30), | |
| Sexo | CHAR, | |
| Salario | DECIMAL(10,2), | |
| Cpf_supervisor | CHAR(11), | |
| Numero_departamento | INT | NOT NULL, |

PRIMARY KEY (Cpf));

CREATE TABLE DEPARTAMENTO

| | | |
|---------------------|-------------|-----------|
| (Nome_departamento | VARCHAR(15) | NOT NULL, |
| Numero_departamento | INT | NOT NULL, |
| Cpf_gerente | CHAR(11), | NOT NULL, |
| Data_inicio_gerente | DATE, | |

PRIMARY KEY (Numero_departamento),

UNIQUE (Nome_departamento),

CREATE TABLE

Exemplo criação
modelo empresa de
antes

```
FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf) );  
  
CREATE TABLE LOCALIZACOES_DEPARTAMENTO  
  
    (Numero_departamento    INT                        NOT NULL,  
     Local                    VARCHAR(15)                NOT NULL,  
  
     PRIMARY KEY (Numero_departamento, Local),  
  
     FOREIGN KEY (Numero_departamento) REFERENCES  
     DEPARTAMENTO(Numero_departamento) );  
  
CREATE TABLE PROJETO  
  
    (Nome_projeto            VARCHAR(15)                NOT NULL,  
     Numero_projeto          INT                        NOT NULL,  
     Local_projeto           VARCHAR(15),  
     Numero_departamento    INT                        NOT NULL,  
  
     PRIMARY KEY (Numero_projeto),  
     UNIQUE (Nome_projeto),  
  
     FOREIGN KEY (Numero_departamento) REFERENCES DEPARTAMENTO(Numero_  
     departamento) );
```

CREATE TABLE

Exemplo criação
modelo empresa de
antes

```
CREATE TABLE TRABALHA_EM
```

| | | |
|------------------|--------------|------------------|
| (Cpf_funcionario | CHAR(11) | NOT NULL, |
| Numero_projeto | INT | NOT NULL, |
| Horas | DECIMAL(3,1) | NOT NULL, |

```
PRIMARY KEY (Cpf_funcionario, Numero_projeto),
```

```
FOREIGN KEY (Cpf_funcionario) REFERENCES FUNCIONARIO(Cpf),
```

```
FOREIGN KEY (Numero_projeto) REFERENCES PROJETO(Numero_projeto) );
```

```
CREATE TABLE DEPENDENTE
```

| | | |
|------------------|-------------|------------------|
| (Cpf_funcionario | CHAR(11), | NOT NULL, |
| Nome_dependente | VARCHAR(15) | NOT NULL, |
| Sexo | CHAR, | |
| Data_nascimento | DATE, | |
| Parentesco | | VARCHAR(8), |

```
PRIMARY KEY (Cpf_funcionario, Nome_dependente),
```

```
FOREIGN KEY (Cpf_funcionario) REFERENCES FUNCIONARIO(Cpf) );
```


TIPOS DE DADOS - NUMÉRICOS

INTEIROS:

INTEGER ou INT, SMALLINT

PONTO FLUTUANTE:

FLOAT ou REAL, DOUBLE PRECISION

NÚMERO FORMATADOS:

DECIMAL(i,j), DEC(i,j), NUMERIC(i,j) -> i = número de dígitos
/ j = número de dígitos após o ponto decimal

TIPOS DE DADOS – CADEIAS DE TEXTO

TAMANHO FIXO:

CHAR(n) ou CHARACTER(n) -> n = número de caracteres

TAMANHO VARIÁVEL:

VARCHAR(n) ou CHAR VARYING(n) ou CHARACTER VARYING(n) -> n = número máximo de caracteres

Os valores devem estar sempre entre ' '. Ex: 'FERNANDO' ou 'F'.

Para concatenar valor, usa-se o ||.

Ex: 'FER' || 'NANDO' = 'FERNANDO'

TIPOS DE DADOS – CADEIAS DE TEXTO

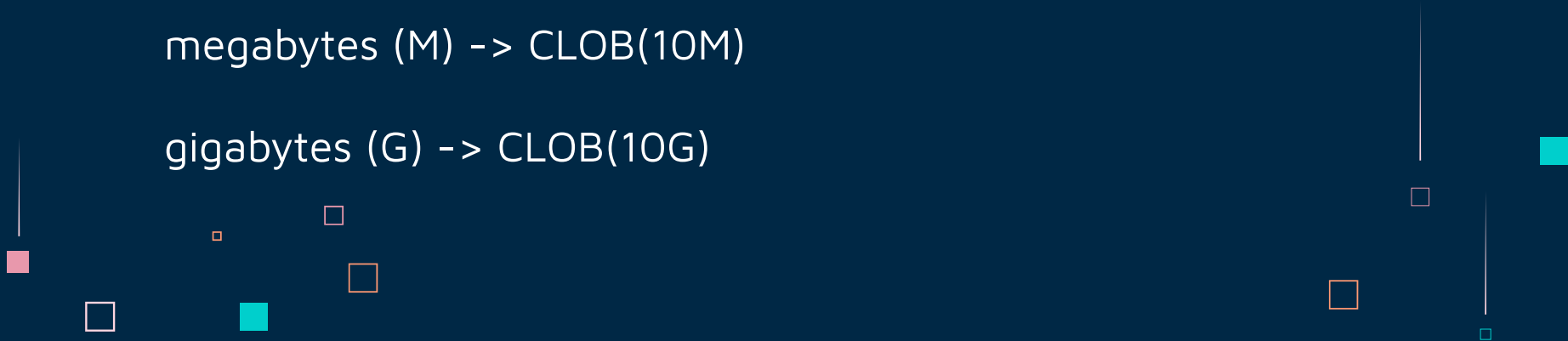
TAMANHO VARIÁVEL PARA GRANDES VOLUMES DE TEXTO:

CHARACTER LARGE OBJECT ou CLOB -> o tamanho máximo pode em:

killobytes (K) -> CLOB(10K)

megabytes (M) -> CLOB(10M)

gigabytes (G) -> CLOB(10G)



TIPOS DE DADOS - SEQUÊNCIA DE BITS

TAMANHO FIXO:

BIT(n) -> n = número de bits

TAMANHO VARIÁVEL:

BIT VARYING(n) -> n = número de bits

BINARY LARGE OBJECT ou BLOB -> para grandes valores de binários, como imagens. Segue o mesmo padrão do CLOB

Os valores devem estar sempre entre ' ' precedidos de um B.
Ex: B'101010001'.

TIPOS DE DADOS - BOOLEANO

REPRESENTAM VALORES TRUE ou FALSE.

PODE-SE TER UM VALOR UNKNOWN

assim tem-se uma lógica com três valores.



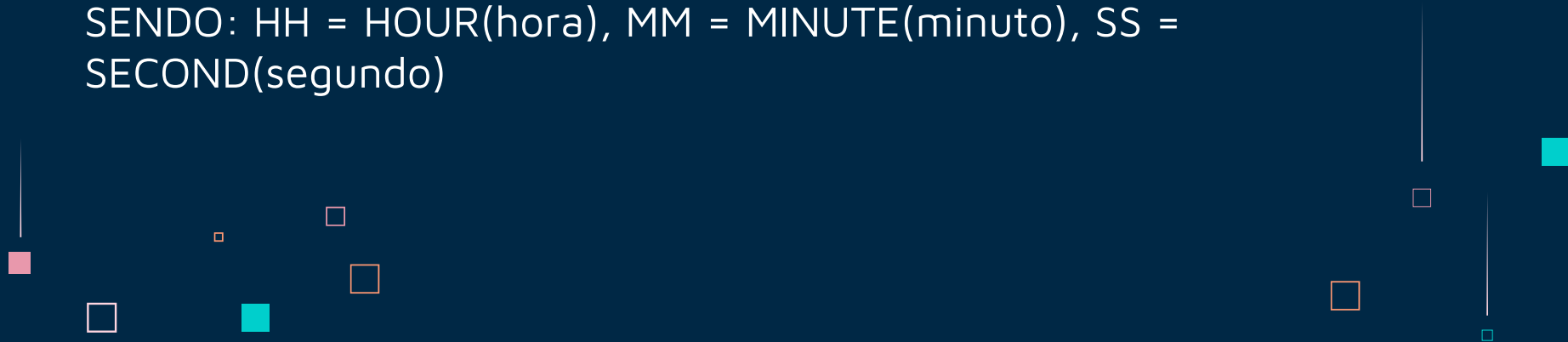
TIPOS DE DADOS - DATE

DATE = POSSUI 10 POSIÇÕES: DD-MM-YYYY

SEND0: DD = DAY(dia), MM = MONTH(mês), YYYY = YEAR(ano)

TIME = POSSUI OITO POSIÇÕES: HH:MM:SS

SEND0: HH = HOUR(hora), MM = MINUTE(minuto), SS = SECOND(segundo)



TIPOS DE DADOS – OUTROS

TIMESTAMP = Junção de DATE e TIME, mais um mínimo de seis posições para frações decimais de segundos e opcionalmente um qualificador de WITH TIME ZONE.



CONSTRAINTS - RESTRIÇÕES

CHECK= Limitar valores do atributo do domínio, após sua declaração.

```
CREATE TABLE DEPARTAMENTO (  
    numero_departamento INT NOT NULL CHECK  
        (numero_departamento > 0 AND numero_departamento < 21)  
)
```

```
CREATE DOMAIN D_NUM AS INTEGER  
CHECK (D_NUM > 0 AND D_NUM < 21)
```