

Banco de Dados

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

Sumário

- ❑ Introdução aos Sistemas de Banco de Dados
- ❑ Modelo de Entidades e Relacionamentos
- ❑ Bancos de Dados Relacionais
 - ➔ O Modelo Relacional
 - ➔ Modelagem de Dados
 - ➔ Álgebra Relacional
 - ➔ Linguagem de Manipulação e Consulta
- ❑ Projeto de Bancos de Dados Relacionais
- ❑ Projeto Lógico de Bancos de Dados



5. SQL

- Histórico -

❑ Structured Query Language - SQL

➡ Desenvolvida pela IBM

⇒ Structured English Query Language - Sequel

⇒ Linguagem de consulta para o sistema R

⇒ Primeiro protótipo de SBD relacional

➡ Especificar consultas de forma interativa

⇒ ad hoc queries

❑ Padrão

➡ ANSI

⇒ SQL 86

⇒ SQL 89

⇒ SQL 92 (SQL2)

⇒ SQL 99 (SQL3)

⇒ SQL: 2003

⇒ SQL: 2006

⇒ SQL: 2008

⇒ Mais recursos de BDs de objeto

⇒ Propriedade de SBDs objeto-relacionais

5. SQL

- Definição de Esquemas em SQL -

- ❑ Esquema de um banco de dados relacional
 - Conjunto de esquemas de relação mais um conjunto de restrições de integridade IC
- ❑ Expressões DDL do SQL permitem especificar
 - Esquema de relações(tabelas)
 - Domínio de valores associados a cada atributo
 - Restrições de integridade
 - Conjunto de índices a serem mantidos para cada relação
 - Estrutura de armazenamento físico de cada relação em disco
 - Autorização de acesso para cada relação

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas

➡ Estrutura básica

⇒ CREATE TABLE nome-tabela

(nome-coluna tipo-de-dados [not null],
[nome-coluna tipo-de-dados [not null] ...],
[CONSTRAINT nome-restrição]

UNIQUE nome-coluna

| PRIMARY KEY(nome-coluna {, nome-coluna})

| FOREIGN KEY (nome-coluna {, nome-coluna})

REFERENCES nome-tabela

[ON DELETE CASCADE |

SET NULL | NO ACTION],

[ON UPDATE CASCADE],

| CHECK (predicado)

)

Especifica
chaves
candidatas

Definição
de restrições

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (cont.)

➡ Alguns tipos de dados suportados pelo SQL 92

⇒ char(n)

⇒ string de caracteres de tamanho fixo n

⇒ varchar(n)

⇒ string de caracteres de tamanho variável (máximo n)

⇒ integer (inteiros de 4 bytes)

⇒ smallint (inteiros de 2 bytes)

⇒ decimal(p,d)

⇒ numérico com p dígitos

⇒ Dos p dígitos, d dígitos representam casas decimais após a vírgula

⇒ real

⇒ numérico ponto flutuante

⇒ date: data no formato 'CCAA-MM-DD'

⇒ data de calendário

• Alphabetic date formats ('April 15, 1998')
• Numeric date formats ('4/15/1998', 'April 15, 1998')
• Unseparated string formats ('19981207',
'December 12, 1998')

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (cont.)

➡ Algumas propriedades dos tipos de dados numéricos do MySQL:

⇒ UNSIGNED (UN) / SIGNED

⇒ signed é o default. Unsigned determina valor sem sinal (+ / -), ou seja, só aceita valores positivos, mudando o *range* do tipo utilizado.

⇒ ZEROFILL (ZF)

⇒ preenche com zeros à esquerda do valor dado. O MySQL adicional automaticamente o atributo Unsigned para o campo / coluna em questão.

⇒ AUTO INCREMENT (AI)

⇒ determina que o campo / coluna em questão será automaticamente incrementado, a cada nova inserção na tabela.

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (cont.)

➡ Algumas propriedades dos tipos de dados suportados pelo MySQL

⇒ PRIMARY KEY (PK)

⇒ Chave primária da relação.

⇒ NOT NULL (NN)

⇒ Determina que o campo não pode receber o valor NULL (nulo).

⇒ UNIQUE KEY (UN)

⇒ Chave candidata da relação (Tem valor único / não repetido).

⇒ BINARY (B)

⇒ Campo com valor binário.

⇒ GENERATED (G)

⇒ Campo com valor calculado / derivado.

⇒ DEFAULT / EXPRESSION

⇒ Campo para informar expressão a ser calculada ou valor default do respectivo atributo.

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (cont.)

➡ Alguns tipos de dados suportados pelo MySQL

⇒ ENUM(<lista_de_valores>)

⇒ É um objeto string com um valor escolhido de uma lista de valores permitidos que são enumerados explicitamente na criação do atributo na especificação da tabela.

⇒ Exemplo:

```
CREATE TABLE roupa (  
    nome VARCHAR(40),  
    tamanho ENUM('x-pequeno', 'pequeno', 'médio', 'grande', 'x-grande')  
);
```

```
INSERT INTO roupa (nome, tamanho) VALUES ('vestido','grande'),  
('camisa','médio'), ('camisa polo','pequeno');
```

```
SELECT nome, tamanho FROM roupa WHERE tamanho = 'medio';
```

```
+-----+-----+  
| nome | tamanho|  
+-----+-----+  
| camisa | médio |  
+-----+-----+
```

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (cont.)

➡ Exemplo: Considere o seguinte esquema de banco de dados

Departamento(cod-depart, nome, ender)

Empregado(matr, nome, ender, cpf,salário,lotação)

⇒ Crie as tabelas Departamento e Empregado utilizando a DDL do SQL com as seguintes restrições

⇒ O salário de cada empregado deve ser maior que 200

⇒ Não podem existir valores de cpf repetidos

⇒ Não se pode permitir a remoção de departamentos para os quais ainda existam empregados lotados

⇒ Ao alterar o código de um departamento, alterar o valor do atributo lotação para todos empregados do departamento

5. SQL

- Definição de Esquemas em SQL -

❑ Criando tabelas (Exemplo)

➡ Create Table Departamento
(cod-depart integer not null,
nome varchar(35),
ender varchar(50),
Primary Key(cod-depart));

➡ Create Table Empregado
(matr integer not null,
nome varchar(40) not null,
ender varchar(40),
cpf char(11),
salário decimal(7,5),
lotação integer not null,
Primary Key(matr),
Unique(cpf),
Foreign Key(lotação)
References (Departamento)
On Delete No Action
On Update Cascade
Check (salário > 200));