

Mestrado integrado em Eng<sup>a</sup> e Gestão  
de Sistemas de Informação



Bases de Dados

1º Ano / 2º Semestre

2016/17

José Luís Pereira

Departamento de Sistemas de Informação

Universidade do Minho

MIEGSI - 2016/17

1

Modelo Relacional de  
Bases de Dados



• A SQL como linguagem de definição de dados

- Criação de tabelas

```
CREATE TABLE <nome_tabela> (  
  < definição de colunas e  
  restrições de integridade >  
)
```

- Quais as colunas da tabela?
- Quais os seus domínios (INTEGER, DECIMAL, CHAR, ...)?
- Quais as restrições de integridade?
- Alguns exemplos avulso:

custo      Nome da Coluna      Domínio      Restrição de Integridade  
DECIMAL(5, 2)      NOT NULL

MIEGSI - 2016/17

2

Modelo Relacional de  
Bases de Dados



• A SQL como linguagem de definição de dados (cont.)

- Chave Primária (supondo a tabela Linhas\_de\_Fatura)

```
num_fatura      CHAR(5),  
cod_produto    CHAR(6),  
PRIMARY KEY (num_fatura, cod_produto)
```

- Chave Secundária (supondo a tabela Fornecedores)

```
cod_fornecedor      CHAR(4) PRIMARY KEY,  
nome_fornecedor    VARCHAR(100) NOT NULL,  
UNIQUE (nome_fornecedor)
```

MIEGSI - 2016/17

3

Modelo Relacional de  
Bases de Dados



• A SQL como linguagem de definição de dados (cont.)

- Condições genéricas sobre os dados das tabelas

```
...  
peso      INTEGER,  
classificacao    SMALLINT,  
genero      CHAR(1),  
...  
CHECK (peso > 50),  
CHECK (classificacao BETWEEN 0 AND 20),  
CHECK (genero IN ('F', 'M'))
```

→ Ainda não disponível na generalidade dos SGBD...

MIEGSI - 2016/17

4

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Restrições de integridade referencial

Departamentos	<u>cod_dep</u>	departamento	...
	abc	Informática	...
	bcd	Produção	...
	cde	Comercial	...

Funcionários	<u>cod_func</u>	nome	...	<u>cod_dep</u>
	1234	J. Silva	...	abc
	2345	A. Costa	...	cde
	3456	F. Martins	...	abc
	4567	C. Cunha	...	abc
	5678	B. Alves	...	bcd

```
CREATE TABLE Funcionarios (
...
cod_dep CHAR(3),
FOREIGN KEY (cod_dep)
REFERENCES Departamentos (cod_dep)
ON UPDATE CASCADE
ON DELETE SET NULL,
...
)
```

MiEGSI - 2016/17

5

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Exemplo de definição de um Esquema de Bases de Dados

- Considerando o já conhecido modelo relacional:

**Clientes** (cod\_cliente, cliente, profissao, localidade)

**Agencias** (cod\_agencia, agencia, localidade)

**Contas** (num\_conta, tipo\_conta, cod\_cliente, cod\_agencia, saldo)

**Emprestimos** (num\_emprestimo, cod\_cliente, cod\_agencia, valor)

- O seu esquema poderia ser definido em SQL:

MiEGSI - 2016/17

6

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

```
CREATE TABLE Contas (
num_conta CHAR(10),
tipo_conta CHAR(5),
cod_agencia CHAR(3),
cod_cliente CHAR(4) NOT NULL,
saldo DECIMAL(10,2) NOT NULL,
CONSTRAINT tipos_de_contas
CHECK (tipo_conta IN ('ordem','prazo')),
CONSTRAINT valor_saldo CHECK (saldo >= 10000),
CONSTRAINT ch_prim_Contas
PRIMARY KEY (num_conta),
CONSTRAINT ch_estr_Agencias_Contas
FOREIGN KEY (cod_agencia)
REFERENCES Agencias (cod_agencia)
ON UPDATE CASCADE
ON DELETE SET NULL,
CONSTRAINT ch_estr_Clientes_Contas
FOREIGN KEY (cod_cliente)
REFERENCES Clientes (cod_cliente)
ON UPDATE CASCADE
ON DELETE RESTRICT
);
```

MiEGSI - 2016/17

7

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

```
CREATE TABLE Emprestimos (
num_emprestimo CHAR(5),
cod_agencia CHAR(3),
cod_cliente CHAR(4) NOT NULL,
valor INTEGER NOT NULL,
CONSTRAINT valor_emprestimo
CHECK (valor BETWEEN 100000 AND 1000000000),
CONSTRAINT ch_prim_Emprestimos
PRIMARY KEY (num_emprestimo),
CONSTRAINT ch_estr_Agencias_Emprestimos
FOREIGN KEY (cod_agencia)
REFERENCES Agencias (cod_agencia)
ON UPDATE CASCADE
ON DELETE SET NULL,
CONSTRAINT ch_estr_Clientes_Emprestimos
FOREIGN KEY (cod_cliente)
REFERENCES Clientes (cod_cliente)
ON UPDATE CASCADE
ON DELETE CASCADE
);
```

MiEGSI - 2016/17

8

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

```
CREATE TABLE Agencias (
  cod_agencia CHAR(3),
  agencia VARCHAR(50) NOT NULL,
  localidade VARCHAR(50) NOT NULL,
  CONSTRAINT ch_candidata_Agencias
    UNIQUE (agencia),
  CONSTRAINT ch_prim_Agencias
    PRIMARY KEY (cod_agencia)
);

CREATE TABLE Clientes (
  cod_cliente CHAR(4),
  cliente VARCHAR(50) NOT NULL,
  profissao VARCHAR(20),
  localidade VARCHAR(50) NOT NULL,
  CONSTRAINT ch_prim_Clientes
    PRIMARY KEY (cod_cliente)
);
```

MIEGSI - 2016/17

9

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

- Alteração e remoção de tabelas (alguns exemplos)

```
ALTER TABLE Clientes
  ADD COLUMN nacionalidade VARCHAR(15)
  DEFAULT 'portuguesa';
```

```
ALTER TABLE Clientes
  MODIFY COLUMN nacionalidade VARCHAR(25)
  DEFAULT 'portuguesa';
```

```
ALTER TABLE Clientes
  DROP COLUMN nacionalidade;
```

```
DROP TABLE Clientes;
```

MIEGSI - 2016/17

10

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

- Tabelas virtuais (views)

```
CREATE VIEW <view> [(<colunas>)]
  AS <questão>
  [WITH CHECK OPTION];
```

- Exemplo

```
CREATE VIEW Super_Clientes (cod_cliente, cliente)
  AS (SELECT Cl.cod_cliente, Cl.cliente
    FROM Clientes Cl
    WHERE (SELECT AVG(saldo)
      FROM Contas)
    < ALL
      (SELECT Saldo
        FROM Contas Co
        WHERE Co.cod_cliente = Cl.cod_cliente))
```

MIEGSI - 2016/17

11

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)

- As views são muito utilizadas na simplificação de questões

- Exemplo:

"quais os clientes, com empréstimos contraiados neste banco, cujas contas têm saldos superiores à média (ou seja, são super clientes)?"

```
SELECT *
  FROM Super_Clientes
 WHERE cod_cliente IN (SELECT cod_cliente
    FROM Empréstimos)
```

Remoção da view

```
DROP VIEW Super_Clientes;
```

MIEGSI - 2016/17

12

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Atualização de *views*
    - Nem todas as *views* são atualizáveis...
    - As que forem, podem evitar atualizações fora do seu âmbito

```
CREATE VIEW Grandes_Contas
AS (SELECT *
    FROM Contas
    WHERE saldo > 1000000)
WITH CHECK OPTION
```

MIEGSI - 2016/17

13

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Restrições do tipo *assertion*
    - Não ficam “presas” a qualquer tabela
    - Úteis para especificar restrições que envolvam várias tabelas
    - Exemplo: “O saldo total das contas de um cliente (caso possua alguma!) não pode ser superior ao valor de qualquer empréstimo contraído por esse cliente”

```
ALTER TABLE Empréstimos
ADD CONSTRAINT Empréstimos_Contas
CHECK ((SELECT SUM(C.saldo)
    FROM Contas C
    WHERE C.cod_cliente = cod_cliente)
=<
(SELECT MIN(E.valor)
    FROM Empréstimos E
    WHERE E.cod_cliente = cod_cliente));
```

MIEGSI - 2016/17

14

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Esta restrição associada apenas à tabela “Empréstimos” não garante a integridade da base de dados. O mesmo acontece se for associada apenas à tabela “Contas”
  - **Solução:** adicionar a mesma restrição às duas tabelas ou, melhor ainda, criar uma *assertion*:

```
CREATE ASSERTION Condições_de_empréstimo
CHECK (NOT EXISTS (
    SELECT *
    FROM Clientes Cl
    WHERE (SELECT SUM(C.saldo)
        FROM Contas C
        WHERE C.cod_cliente = Cl.cod_cliente)
    >=
    (SELECT MIN(E.valor)
        FROM Empréstimos E
        WHERE E.cod_cliente = Cl.cod_cliente)
));
```

Infelizmente não está ainda disponível na generalidade dos SGBD!

MIEGSI - 2016/17

15

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Código na base de dados
  - Da mesma forma que o SGBD permite o armazenamento e partilha de dados pelo nível aplicacional, também pode permitir o armazenamento e partilha de código – **Funções e Procedimentos**
  - Exemplo: definição de uma função que calcula o saldo total de contas à ordem que um dado cliente possui numa dada agência

```
CREATE FUNCTION Saldo_Total_Cliente(cliente CHAR(4),
    agencia CHAR(3))
RETURNS DECIMAL(10,2)
BEGIN
    DECLARE resultado DECIMAL(10,2);
    SELECT SUM(Saldo)
    INTO resultado
    FROM Contas
    WHERE cod_cliente = cliente AND
    cod_agencia = agencia AND tipo_conta = 'Ordem';
    RETURN resultado;
END;
```

MIEGSI - 2016/17

16

## Modelo Relacional de Bases de Dados



### • A SQL como linguagem de definição de dados (cont.)

- Código na base de dados (cont.)
- Se as funções retornam o resultado de algum tipo de cálculo na Base de Dados, os procedimentos permitem atuar sobre o seu conteúdo
- Exemplo: procedimento para transferir uma conta bancária entre duas agências

```
CREATE PROCEDURE Transfere (IN conta CHAR(4),
                           agencia_destino CHAR(3))
BEGIN
    UPDATE Contas
        SET cod_agencia = agencia_destino
    WHERE num_conta = conta;
END;
```

MiEGSI - 2016/17

17

## Modelo Relacional de Bases de Dados



### • A SQL como linguagem de definição de dados (cont.)

- Código na base de dados (cont.)
- Exercício: desenvolver um procedimento para consolidar, numa nova conta, as contas à ordem que um dado cliente possuir numa agência

```
CREATE PROCEDURE Consolidar (IN cliente CHAR(4),
                             agencia CHAR(3),
                             nova_conta CHAR(4))
BEGIN
    DECLARE total DECIMAL(10,2);
    SET total = Saldo_Total_Cliente(cliente, agencia);
    DELETE FROM Contas
    WHERE cod_cliente = cliente AND
          cod_agencia = agencia AND tipo_conta = 'Ordem';
    INSERT INTO Contas (num_conta, tipo_conta,
                       cod_agencia, cod_cliente, saldo)
        VALUES (nova_conta, 'Ordem', agencia, cliente,
                total);
END;
```

MiEGSI - 2016/17

18

## Modelo Relacional de Bases de Dados



### • A SQL como linguagem de definição de dados (cont.)

- **Triggers**
  - Permitem libertar o nível aplicacional de tarefas rotineiras
  - Manutenção da integridade (desnormalização, dados replicados)
  - Suporte de regras de negócio

Exemplo: Relativamente ao esquema anterior considerar que "quando um cliente contrai um empréstimo, o banco oferece sempre 1% do valor desse empréstimo, a creditar em cada uma das suas contas, mas apenas se esse cliente possuir mais de três contas no banco"

Como implementar esta regra de negócio?

MiEGSI - 2016/17

19

## Modelo Relacional de Bases de Dados



### • A SQL como linguagem de definição de dados (cont.)

- **Triggers** (cont.)
- Por exemplo, num SGBD como o MySQL, o *trigger* anterior poderia ser:

```
CREATE TRIGGER Bonus
AFTER INSERT ON Empréstimos
FOR EACH ROW
BEGIN
    DECLARE num_contas SMALLINT;

    SELECT COUNT(*)
    INTO num_contas
    FROM Contas
    WHERE cod_cliente = NEW.cod_cliente;
    IF (num_contas > 3) THEN
        UPDATE Contas
        SET saldo= saldo + 0.01 * NEW.valor
        WHERE cod_cliente= NEW.cod_cliente;
    END IF;
END;
```

MiEGSI - 2016/17

20