

Licenciatura em Tecnologias e
Sistemas de Informação



Bases de Dados

1ºAno / 2ºSemestre

2008/09

José Luís Pereira
Carlos Sousa Pinto


Departamento de Sistemas de Informação

Universidade do Minho

LTSI - 2008/09

1

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))
```

LTSI - 2008/09

2

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 contraídos neste banco, cujas contas têm saldos superiores à media (super clientes) ?”

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


Remoção da view

```
DROP VIEW Super_Clientes
```

LTSI - 2008/09

3

Modelo Relacional de
Bases de Dados



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

Atualização de views

Nem todas as views são actualizáveis...

As que forem, podem evitar actualizações fora do seu âmbito

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

LTSI - 2008/09

4

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 sobre várias tabelas
 - Exemplo: “O saldo total das contas de cada cliente (caso possua alguma!) não pode ser superior ao valor de qualquer empréstimo contraído por esse cliente”

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

Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
 - Esta restrição associada apenas à tabela “Emprestimos” 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 então, criar uma *assertion*:

```
CREATE ASSERTION Condicoes_de_emprestimo
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 Emprestimos E
          WHERE E.cod_cliente = Cl.cod_cliente)
))
```

Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
 - Armazenamento de código na base de dados
 - Da mesma forma que o SGBD permite a partilha de dados e restrições de integridade pelo nível aplicacional, também pode permitir a 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 agência

```
CREATE FUNCTION Saldo_Ordem_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;
```

Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
 - Armazenamento de código na base de dados (cont.)
 - Se as funções retornam resultados, os procedimentos permitem actuar sobre a base de dados
 - 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;
```

Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
 - Armazenamento de código na base de dados (cont.)
 - Exercício: desenvolver um procedimento para consolidar as contas à ordem que um dado cliente possui numa agência, numa nova conta
- ```
CREATE PROCEDURE Consolidar (IN cliente CHAR(4),
 agencia CHAR(3),
 nova_conta CHAR(4))
BEGIN
 DECLARE total DECIMAL (10,2);
 SET total = Saldo_Ordem_Cliente(cliente, agencia);
 DELETE FROM Contas
 WHERE cod_cliente = cliente AND
 cod_agencia = agencia;
 INSERT INTO Contas (num_conta, tipo_conta,
 cod_agencia, cod_cliente, saldo)
 VALUES (nova_conta, 'Ordem', agencia, cliente,
 total);
END;
```

LTSI - 2008/09

9

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
    - Triggers
      - Permite libertar o nível aplicacional de tarefas rotineiras
      - Manutenção da integridade (desnormalização, dados replicados, etc.)
      - Suporte de regras de negócio
- Por 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, desde que esse cliente possua mais de três contas no banco ”

Como implementar esta regra de negócio?

LTSI - 2008/09

10

## Modelo Relacional de Bases de Dados



- A SQL como linguagem de definição de dados (cont.)
  - Triggers (cont.)
    - Utilizando, por exemplo, a linguagem PL/SQL da família de SGBDs Oracle, o trigger anterior poderia ser escrito:

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

Maior desvantagem:  
→ Falta de portabilidade!

LTSI - 2008/09

11