



Bases de Dados

1ºAno – 2ºSemestre

2008/09

José Luís Pereira
Carlos Sousa Pinto

Departamento de Sistemas de Informação
Universidade do Minho

Modelo Relacional de
Bases de Dados



- Linguagens Relacionais
 - Baseadas na Álgebra Relacional e, sobretudo, no Cálculo Relacional
 - Linguagem mais representativa → SQL
 - No entanto, existem outras linguagens relacionais...

Modelo Relacional de
Bases de Dados



- **QBE** – inspirada no Cálculo Relacional de Domínios
 - Utilização interactiva
 - Para utilizadores menos sofisticados
 - Exemplo, dada a relação:

Alunos (número, nome, curso, média, ano_finalização), a questão:
“quais os alunos (número e nome) com média final superior a 14?”

Alunos	número	nome	curso	média	ano_finalização
	P.	P.		>14	

- A semelhança com a versão correspondente ao Cálculo Relacional de domínios é evidente:
$$\{ab \mid (\exists d) (Alunos(abcde) \wedge d > 14)\}$$

Modelo Relacional de
Bases de Dados



- **QUEL** – inspirada no Cálculo Relacional de Tuplos
 - Utilizada exclusivamente no SGBD Ingres
 - Utilização interactiva ou programada
 - Exemplo, dada a relação:
Alunos (número, nome, curso, média, ano_finalização), a questão:
“quais os alunos (número e nome) com média final superior a 14?”

```
RANGE OF a IS alunos
RETRIEVE (a.numero, a.nome)
WHERE a.media > 14
```
 - A semelhança com a versão correspondente ao Cálculo Relacional de tuplos é evidente:
$$\{a.numero, a.nome \mid Alunos(a) \wedge a.média > 14\}$$

Modelo Relacional de Bases de Dados



- Xbase

- Muito divulgadas na área das *desktop databases*
- Utilização interactiva ou programada
- Exemplo, dada a relação:

Alunos (número, nome, curso, média, ano_finalização), a questão:

“quais os alunos (número e nome) com média final superior a 14?”

```
USE alunos
DISPLAY ALL FIELDS numero, nome
FOR media > 14
```

- Outras linguagens relacionais...

LTSI - 2008/09

5

Modelo Relacional de Bases de Dados



- Linguagem SQL

- Considerada o standard em termos de linguagens relacionais
- Sujeita a normalização (SQL-86/89, SQL2, SQL3, ...)

- A SQL como:

- Linguagem de manipulação de dados (LMD)
 - Interrogação de bases de dados
 - Actualização de bases de dados
- Linguagem de definição de dados (LDD)

LTSI - 2008/09

6

Modelo Relacional de Bases de Dados



- A SQL como linguagem de manipulação de dados

- SQL - Interrogação de bases de dados

- Formato base de uma questão

```
SELECT <colunas>
FROM <tabelas>
[WHERE <condição>]
```

- Considerar o seguinte exemplo:

Cientes (cod_cliente, cliente, profissão, localidade)
Agências (cod_agência, agência, localidade)
Contas (num_conta, tipo_conta, cod_agência, cod_cliente, saldo)
Empréstimos (num_empréstimo, cod_agência, cod_cliente, valor)

LTSI - 2008/09

7

Modelo Relacional de Bases de Dados



- SQL - Interrogação de bases de dados

- *Quais os clientes (cod_cliente e cliente) deste banco?*

```
SELECT cod_cliente, cliente
FROM Clientes
```

- *Quais os clientes que residem em Braga?*

```
SELECT *
FROM Clientes
WHERE localidade = 'Braga'
```

- *Quais os clientes (cod_cliente) com contas na agência cod_agencia='123'?*

```
SELECT DISTINCT cod_cliente
FROM Contas
WHERE cod_agencia = '123'
```

LTSI - 2008/09

8

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes que residem em localidades onde existem agências?

```
SELECT Clientes.*
FROM Clientes, Agencias
WHERE Clientes.localidade = Agencias.localidade
```

- Quais os clientes com empréstimos de valor superior a 500.000?

```
SELECT Clientes.*
FROM Clientes, Empréstimos
WHERE Clientes.cod_cliente = Empréstimos.cod_cliente
AND Empréstimos.valor > 500000
```

Utilizando sinónimos (aliases):

```
SELECT C.*
FROM Clientes C, Empréstimos E
WHERE C.cod_cliente = E.cod_cliente
AND E.valor > 500000
```

LTSI - 2008/09

9

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os nomes dos clientes com a mesma profissão que o cliente com cod_cliente = '1234'?

```
SELECT C1.cliente
FROM Clientes C1, Clientes C2
WHERE C1.profissao = C2.profissao
AND C2.cod_cliente = '1234'
```

- Listar as contas (num_conta, saldo) da agência cujo cod_agencia = '123', por ordem decrescente do seu valor de saldo.

```
SELECT num_conta, saldo
FROM Contas
WHERE cod_agencia = '123'
ORDER BY saldo DESC
```

LTSI - 2008/09

10

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quantas contas existem em todas as agências do banco?

```
SELECT COUNT(*)
FROM Contas
```

Existem outras funções de agregação para o cálculo do máximo, do mínimo, da média e do somatório (respectivamente, MAX, MIN, AVG e SUM).

- Quantos clientes possuem contas na agência cujo cod_agencia = '123'?

```
SELECT COUNT (DISTINCT cod_cliente)
FROM Contas
WHERE cod_agencia = '123'
```

LTSI - 2008/09

11

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Listar o número de contas existentes em cada agência.

```
SELECT cod_agencia, COUNT(*)
FROM Contas
GROUP BY cod_agencia
```

- Para cada agência (cod_agencia) com menos de 1000 contas, listar os valores máximo e mínimo dos saldos dessas contas, assim como o saldo médio.

```
SELECT cod_agencia, MAX(saldo), MIN(saldo), AVG(saldo)
FROM Contas
GROUP BY cod_agencia
HAVING COUNT(*) < 1000
```

LTSI - 2008/09

12

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes cuja profissão é desconhecida?

```
SELECT *
FROM CLIENTES
WHERE profissao IS NULL
```

- Quais os clientes (cod_cliente e cliente) da agência cod_agencia='123'?

```
SELECT Cl.cod_cliente, Cl.cliente
FROM Contas Co, Clientes Cl
WHERE Co.cod_agencia = '123' AND
      Co.cod_cliente = Cl.cod_cliente

UNION

SELECT Cl.cod_cliente, Cl.cliente
FROM Emprestimos E, Clientes Cl
WHERE E.cod_agencia = '123' AND
      E.cod_cliente = Cl.cod_cliente
```

LTSI - 2008/09

13

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes (cod_cliente e cliente) que são, simultaneamente, depositantes e devedores na agência cujo cod_agencia = '123'?

```
SELECT Cl.cod_cliente, Cl.cliente
FROM Contas Co, Clientes Cl
WHERE Co.cod_agencia = '123' AND
      Co.cod_cliente = Cl.cod_cliente

INTERSECT

SELECT Cl.cod_cliente, Cl.cliente
FROM Emprestimos E, Clientes Cl
WHERE E.cod_agencia = '123' AND
      E.cod_cliente = Cl.cod_cliente
```

LTSI - 2008/09

14

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes (cod_cliente e cliente) da agência com cod_agencia = '123' que apenas são depositantes?

```
SELECT Cl.cod_cliente, Cl.cliente
FROM Contas Co, Clientes Cl
WHERE Co.cod_agencia = '123' AND
      Co.cod_cliente = Cl.cod_cliente

EXCEPT

SELECT Cl.cod_cliente, Cl.cliente
FROM Emprestimos E, Clientes Cl
WHERE E.cod_agencia = '123' AND
      E.cod_cliente = Cl.cod_cliente
```

LTSI - 2008/09

15

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes (cod_cliente e cliente) com, pelo menos, um empréstimo no banco?

```
SELECT C.cod_cliente, C.cliente
FROM Clientes C
WHERE EXISTS (SELECT *
              FROM Emprestimo E
              WHERE C.cod_cliente = E.cod_cliente)
```

LTSI - 2008/09

16

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais as agências (*cod_agencia*, *agencia*) com depositantes residentes em Lisboa?

```
SELECT A.cod_agencia, A.agencia
FROM Agencias A, Contas C
WHERE C.cod_cliente IN
      (SELECT cod_cliente
       FROM Clientes
        WHERE localidade = 'Lisboa') AND
      C.cod_agencia = A.cod_agencia
```

LTSI - 2008/09

17

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes cujo saldo total das suas contas é superior ao valor de qualquer empréstimo contraído neste banco?

```
SELECT Cl.*
FROM Clientes Cl
WHERE (SELECT SUM(Co.saldo)
       FROM Contas Co
        WHERE Co.cod_cliente = Cl.cod_cliente)
      > ALL
      (SELECT valor
       FROM Empréstimos)
```

LTSI - 2008/09

18

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes cujo saldo total das suas contas é superior ao valor de qualquer empréstimo contraído neste banco? (*outra solução*)

```
SELECT Cl.*
FROM Clientes Cl
WHERE (SELECT SUM(Co.saldo)
       FROM Contas Co
        WHERE Co.cod_cliente = Cl.cod_cliente)
      >
      (SELECT MAX (valor)
       FROM Empréstimos)
```

LTSI - 2008/09

19

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Quais os clientes que possuem contas em todas as agências do Porto?
- Uma versão equivalente à questão anterior é a seguinte: "quais os clientes para os quais não existe nenhuma agência do Porto em que estes não possuam alguma conta?"

```
SELECT Cl.*
FROM Clientes Cl
WHERE NOT EXISTS (
  SELECT *
  FROM Agencias A
   WHERE A.localidade = 'Porto' AND
         NOT EXISTS (
           SELECT *
           FROM Contas Co
            WHERE Co.cod_cliente = Cl.cod_cliente
              AND Co.cod_agencia = A.cod_agencia))
```

LTSI - 2008/09

20

Modelo Relacional de Bases de Dados



• SQL - Interrogação de bases de dados (cont.)

- Para cada cliente (cod_cliente) apresentar o seu saldo total.

```
SELECT cod_cliente, SUM (saldo)
FROM Contas
GROUP BY cod_cliente
```

Com SQL-92, outra forma de resolver a mesma questão é a seguinte:

```
SELECT CL.cod_cliente, (SELECT SUM(saldo)
                        FROM Contas CO
                        WHERE CO.cod_cliente = CL.cod_cliente)
FROM Clientes CL
```

LTSI - 2008/09

21

Modelo Relacional de Bases de Dados



• SQL – Actualização de bases de dados

- Inserção de dados

```
INSERT INTO <tabela> [( <colunas> )]
VALUES ( <valores> )
```

- Alguns exemplos

```
INSERT INTO Clientes
VALUES ( '1234', 'J.Silva', 'Estudante', 'Braga' )
```

```
INSERT INTO Clientes
(cod_cliente, cliente, localidade)
VALUES ( '1235', 'A.Costa', 'Guimarães' )
```

```
INSERT INTO Contas_Prazo (num_conta, saldo)
SELECT (num_conta, saldo)
FROM Contas
WHERE saldo < 15000
```

LTSI - 2008/09

22

Modelo Relacional de Bases de Dados



• SQL – Actualização de bases de dados (cont.)

- Alteração de dados

```
UPDATE <tabela>
SET <coluna> = <expressão>,
    ... = ...
[WHERE <condição>]
```

- Alguns exemplos

```
UPDATE Contas
SET saldo = saldo + 1000
WHERE num_conta = '1234567890'
```

```
UPDATE Contas
SET saldo = (SELECT MAX (saldo)
             FROM Contas
             WHERE cod_cliente = '1234')
WHERE cod_cliente = '1234'
```

LTSI - 2008/09

23

Modelo Relacional de Bases de Dados



• SQL – Actualização de bases de dados (cont.)

- Remoção de dados

```
DELETE FROM <tabela>
[WHERE <condição>]
```

- Alguns exemplos

```
DELETE FROM Contas
WHERE num_conta = '1234567890'
```

```
DELETE FROM Clientes
WHERE cod_cliente IN (SELECT cod_cliente
                     FROM Contas
                     WHERE cod_agencia = '123')
```

LTSI - 2008/09

24

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 os atributos/campos da tabela?
- Quais os seus domínios (INTEGER, DECIMAL, CHAR, ...)?
- Quais as restrições de integridade ?
- Exemplos:

```
  • NOT NULL  
  
  custo          DECIMAL(5,2) NOT NULL
```

Modelo Relacional de Bases de Dados



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

- PRIMARY KEY

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

- UNIQUE

```
cod_fornecedor   CHAR(4) PRIMARY KEY,  
nome_fornecedor  CHAR(20) NOT NULL,  
UNIQUE (nome_fornecedor)
```

Modelo Relacional de Bases de Dados



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

- CHECK

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

Modelo Relacional de Bases de Dados



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

- Restrições de integridade referencial

Departamentos	cod_dep	departamento	...	Funcionários	cod_func	nome	...	cod_dep
	abc	Informática	...		1234	J. Silva	...	abc
	bcd	Produção	...		2345	A. Costa	...	cde
	cde	Comercial	...		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 ),  
  ...
```

Modelo Relacional de Bases de Dados



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

- Considerando o caso:

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:

LTSI - 2008/09

29

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 CASCADE  
)
```

LTSI - 2008/09

30

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  
)
```

LTSI - 2008/09

31

Modelo Relacional de Bases de Dados



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

```
CREATE TABLE Agencias (  
  cod_agencia    CHAR(3),  
  agencia        VARCHAR(20) NOT NULL,  
  localidade     VARCHAR(10) 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(30) NOT NULL,  
  profissao      VARCHAR(10),  
  localidade     VARCHAR(10) NOT NULL,  
  CONSTRAINT ch_prim_Clientes  
    PRIMARY KEY (cod_cliente)  
)
```

LTSI - 2008/09

32

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
```