[MAC0426] Sistemas de Bancos de Dados
[IBI5013] Bancos de Dados para Bioinformática

Aula 15
Linguagem SQL (Parte 2)

Consultas Básicas

22 de maio de 2017

Profª Kelly Rosa Braghetto

(Adaptação dos slides do prof. Jeffrey Ullman, da *Stanford University*)

Comandos
Select-From-Where

SELECT <lista de atributos>

FROM <lista de tabelas>

WHERE <condição>

Exemplo para a aula

- As consultas SQL de exemplo serão baseadas no seguinte esquema de BD relacional:

  Refrigerante(nome, fabricante)
  Lanchonete(nome, endereco, cnpj)
  Cliente(nome, endereco, telefone)
  Apreciador(nome_cliente, nome_refri)
  Vendedor(nome_lanch, nome_refri, preco)
  Frequentador(nome_cliente, nome_lanch)

Exemplo

- Usando Refrigerante(nome, fabricante), quais "refris" são feitos por *Cola-Coca* ?

```
SELECT nome
FROM Refrigerante
WHERE fabricante = 'Cola-Coca';
```

Resultado da consulta

nome
Fanfa
Kuaif
Sprife
. . .

A resposta é uma relação com um único atributo, **nome**, e tuplas com o nome de cada refrigerante produzido pela *Cola-Coca*.

“Processamento” de uma consulta sobre uma única relação

- Começa com a relação na cláusula FROM.
- Aplica-se a seleção indicada na cláusula WHERE.
- Aplica-se a projeção indicada pela cláusula SELECT.

Semântica operacional - visão geral

- Processamento de uma consulta:
  - Considere que há uma *variável-tupla* percorrendo cada tupla da relação mencionada na cláusula FROM.
  - Verifique se a tupla “atual” satisfaz a cláusula WHERE.
  - Se sim, compute os atributos ou expressões da cláusula SELECT usando os componentes dessa tupla.

Semântica operacional

Verifica se é *Cola-Coca*

Se sim, inclui t.nome no resultado

A variável-tupla *t* percorre todas as tuplas

```
SELECT nome
FROM Refrigerante
WHERE fabricante = 'Cola-Coca';
```

O * em cláusulas SELECT

- Quando há apenas uma relação na cláusula FROM, um * na cláusula SELECT equivale a "todos os atributos dessa relação".
- Exemplo:

  Usando Refrigerante(nome, fabricante)

```
SELECT *
FROM Refrigerante
WHERE fabricante = 'Cola-Coca';
```

Resultado da consulta

nome	fabricante
Fanfa	Cola-Coca
Kuaif	Cola-Coca
Sprife	Cola-Coca
. . .	. . .

Agora, o resultado possui todos os atributos de Refrigerante.

Renomeando atributos

- Para modificar os nomes dos atributos no resultado, use **“AS <novo nome>”** para renomear um atributo.
- Exemplo:

  usando Refrigerante(nome, fabricante)

```
SELECT nome AS refri, fabricante
FROM Refrigerante
WHERE fabricante = 'Cola-Coca';
```

Resultado da consulta

refri	fabricante
Fanfa	Cola-Coca
Kuaif	Cola-Coca
Sprife	Cola-Coca
. . .	. . .

Expressões em cláusulas SELECT

- Atributos, constantes e funções podem aparecer como elementos na cláusula SELECT.
- Exemplo:

  Usando Venda(nome_lanch, nome_refri, preco)

```
SELECT nome_lanch, nome_refri,
  preço*30.904 AS preco_em_yen
  FROM Venda;
```

Resultado da consulta

nome_lanch	nome_refri	preco_em_yen
Sujinhos	Fanfa	155
Bar do Zé	Sprife	142
...	...	...

Exemplo: Constantes como expressões

Usando Apreciador(nome_cliente, nome_refri):

```
SELECT nome_cliente,
       'aprecia Fanfa' AS descricao
FROM Apreciador
WHERE nome_refri = 'Fanfa';
```

Resultado da consulta

cliente	descricao
Sally	aprecia Fanfa
Fred	aprecia Fanfa
...	...

Condições complexas para a cláusula WHERE

- Operadores booleanos AND, OR, NOT.
- Comparações =, <>, <, >, <=, >=.
  - E muitos outros operadores que produzem valores booleanos como resultado.

Exemplo: Condição “complexa”

- Usando Venda(nome_lanch, nome_refri, preco), encontre o preço cobrado no *Sujinhos* pela *Fanfa*:

```
SELECT preco
FROM Venda
WHERE nome_lanch = 'Sujinhos'
    AND nome_refri = 'Fanfa';
```

Padrões

- Uma condição pode comparar uma *string* com um padrão (~ expressão regular) usando:
  - <Atributo> **LIKE** <padrão> ou <Atributo> **NOT LIKE** <padrão>

- *Padrão* é uma *string* contendo caracteres especiais:
  - '**%**' "casa" com qualquer *string*
  - '**_**' "casa" com qualquer (um) caractere

Exemplo: LIKE

- Usando

  Cliente(nome, endereço, telefone), encontre os clientes com DDD de *São Paulo*:

```
SELECT nome
FROM Cliente
WHERE telefone LIKE '(11)%';
```

Exemplo(2): LIKE

- Usando

  Cliente(nome, endereço, telefone), encontre os clientes cujo primeiro nome tem 3 letras:

```
SELECT nome
FROM Cliente
WHERE nome LIKE '___ %';
```

Caracteres especiais em expressões com o LIKE

- Para usar '%' ou o '_' em um padrão sem que eles exerçam a função de caractere especial, é preciso fazer o "*scape*" deles.
- A SQL nos permite usar qualquer caractere como *scape.*
- Exemplo: padrão que "casa" o valor do atributo *s* com uma *string* iniciada e finalizada por '%'.

  Observe que apenas o '%' do meio será considerado um caractere especial

```
s LIKE 'x%%x%' ESCAPE 'x'
```

Comparação de *strings*, datas e horários

- Também podemos usar os operadores

  >, >=, < e <=

  para comparar *strings*, datas e horários
- Quando comparamos *strings* com o <, por exemplo, estamos perguntando se uma *string* precede a outra na ordem lexicográfica
- Exemplos:

  'facada' < 'farpa' e 'bar' < 'barganha'

Valores NULL

- Tuplas em relações SQL podem ter o NULL como valor para um ou mais de seus atributos.
- O significado do NULL depende do contexto. Dois casos comuns:
  - *Valor desconhecido* – ex.: sabemos que o *Sujinhos* tem um endereço, mas não sabemos qual é.
  - *Não aplicável* – ex.: o valor do atributo nome_cônjuge para uma pessoa solteira.

Comparando NULL com outros valores

- A lógica das condições em SQL é uma lógica ternária: **TRUE**, **FALSE**, **UNKNOWN**.
- Comparar qualquer valor (incluindo o próprio NULL) com NULL resulta em **UNKNOWN**.
- Uma tupla é incluída no conjunto resposta de uma consulta se e somente se a cláusula WHERE é **TRUE** .

Lógica ternária (ou *trivalente*)

- Para entender como o **AND**, **OR** e o **NOT** funcionam na lógica ternária, pense que TRUE = 1, FALSE = 0 e UNKNOWN = ½.
- AND = MIN; OR = MAX, NOT(*x*) = 1-*x*.
- Exemplo:

TRUE AND (FALSE OR NOT(UNKNOWN)) =
MIN(1, MAX(0, (1 - ½))) =

MIN(1, MAX(0, ½)) = MIN(1, ½) = ½.

Qual é o resultado da consulta a seguir?

- A partir da relação Venda a seguir:

nome_lanch	nome_refri	preco
Sujinhos	Fanfa	NULL

```
SELECT nome_lanch  FROM Venda
WHERE preco < 2.00 OR preco >= 2.00;
```

Um exemplo surpreendente

- A partir da relação Venda a seguir:

nome_lanch	nome_refri	preco
Sujinhos	Fanfa	NULL

```
SELECT nome_lanch  FROM Venda
WHERE preco < 2.00 OR preco >= 2.00;
```

UNKNOWN UNKNOWN

UNKNOWN

Resultado: nenhuma tupla é selecionada!

Razão: Leis para a lógica binária != Leis para a lógica ternária

- Algumas leis comuns, como a comutatividade do AND, valem na lógica ternária.
- Mas outras **não valem**
  - Exemplo: *lei do meio excluído*

    p OR NOT *p* = TRUE

  - Quando *p* = UNKNOWN, o lado esquerdo é MAX(½, (1 – ½)) = ½ != 1.

Verificando se o valor de um atributo é NULL

- Para comparar o valor de um atributo com NULL em SQL, não deve-se usar os operadores = ou <>, mas sim operadores ***is*** e ***not is***.
- Exemplo: Selecione os nomes das lanchonetes para as quais não se conhece o endereço.

```
SELECT nome FROM Lanchonete
                    WHERE endereço is NULL;
```

Ordenação do resultado de uma consulta

- É possível ordenar as tuplas da relação resultante de uma consulta por meio da cláusula

  ORDER BY <lista de atributos> [ASC | DESC]

- A ordenação ascendente (ASC) é a padrão
- Exemplos:

```
SELECT * FROM Cliente ORDER BY
                                nome,telefone;
```

ou

```
SELECT * FROM Cliente ORDER BY nome DESC;
```

Consultas envolvendo múltiplas relações

- Consultas interessantes frequentemente combinam dados de mais de uma relação.
- Podemos considerar várias relações em uma consulta listando-as na cláusula FROM.
- Para distinguir atributos de relações diferentes que possuem o mesmo nome: “<relação>.<atributo>” .

Exemplo: junção de duas relações

- Usando a relação Apreciador(nome_cliente, nome_refri) e Frequentador(nome_cliente, nome_lanch), encontre os refris apreciados por pelo menos uma pessoa que frequenta a lanchonete Sujinhos.

```
SELECT nome_refri
FROM Apreciador, Frequentador
WHERE nome_lanch = 'Sujinhos' AND
Frequentador.nome_cliente =
Apreciador.nome_cliente;
```

Semântica formal

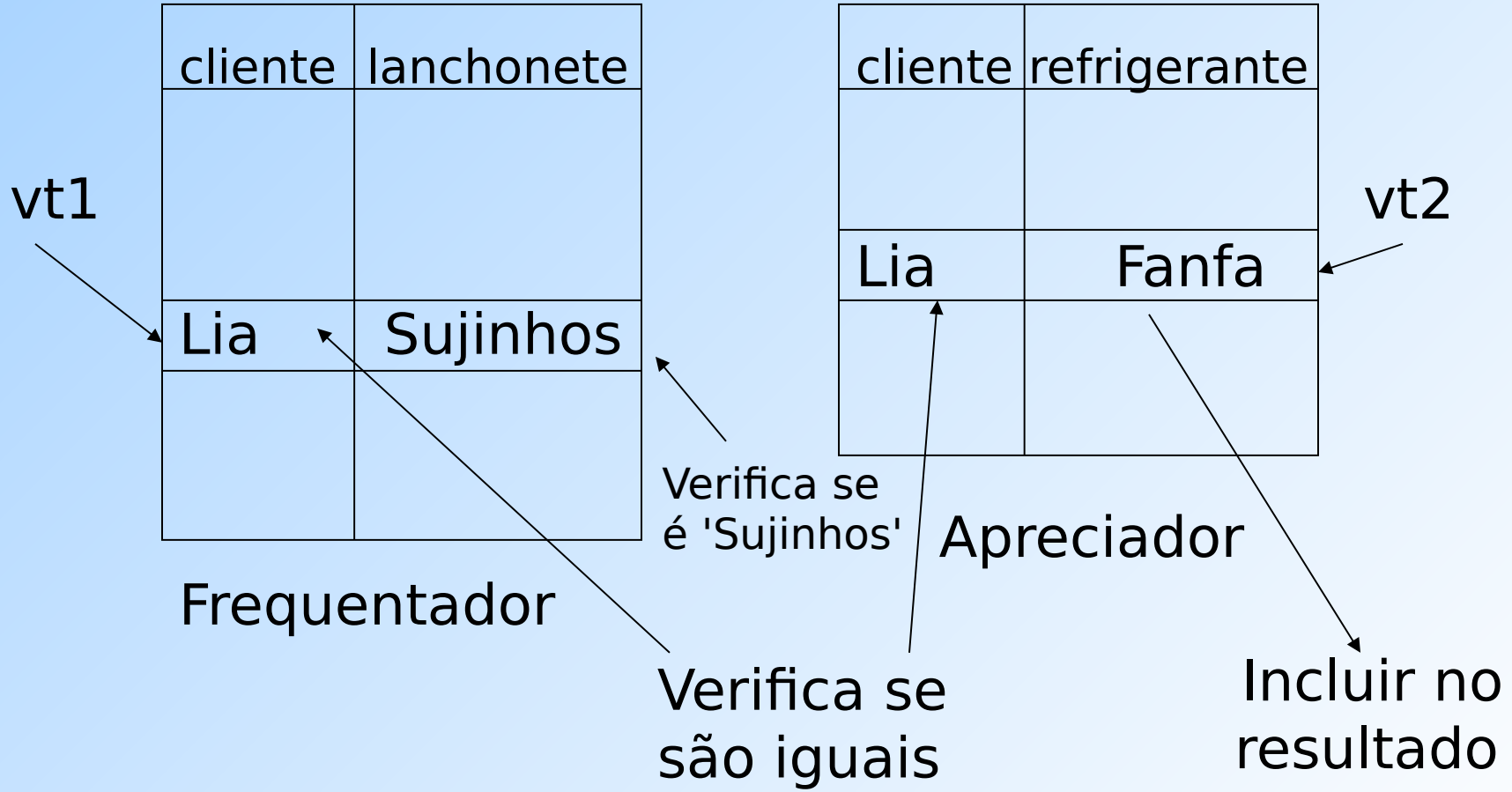
- Quase a mesma que a das consultas sobre uma única relação:
  1. Comece com o produto cartesiano de todas as relações da cláusula FROM.
  2. Aplique a condição de seleção da cláusula WHERE.
  3. Projete sobre a lista de atributos e expressões da cláusula SELECT.

Semântica operacional

- Imagine uma variável-tupla para cada relação na cláusula FROM.
  - Essas variáveis visitam cada combinação possível de tuplas, uma de cada relação.
- Se as variáveis-tuplas apontam para tuplas que satisfazem a cláusula WHERE, envie essas tuplas para a cláusula SELECT.

Exemplo

```
SELECT nome_refri
FROM Apreciador, Frequentador
WHERE nome_lanch = 'Sujinhos' AND
Frequentador.nome_cliente = Apreciador.nome_cliente;
```



Variáveis-tuplas explícitas

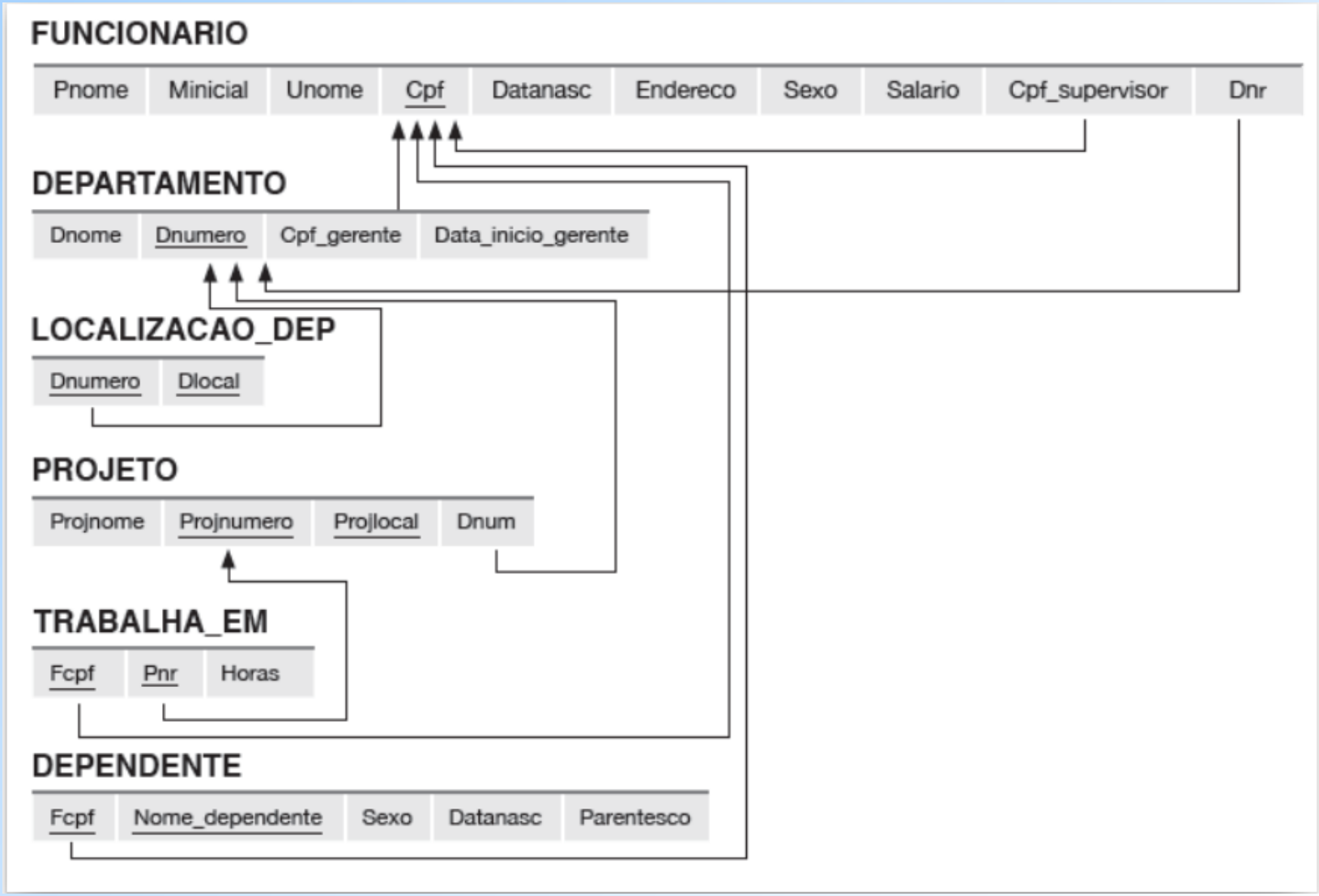
- Às vezes, uma consulta precisa usar duas cópias de uma mesma relação.
- Para diferenciar as cópias, acrescente o nome de uma variável-tupla na frente do nome da relação na cláusula FROM.
- É sempre possível renomear uma relação desta forma (mesmo quando isso não é indispensável para a consulta).

Exemplo: auto-junção

- A partir de Refrigerante(nome_refri, fabricante), encontre todos os pares de refri feitos por um mesmo fabricante. Restrições:
  - Não produza pares como (*Fanfa*, *Fanfa*).
  - Produza pares em ordem alfabética, p.e., (*Fanfa*, *Sprife*), mas não (*Sprife*, *Fanfa*).

```
SELECT r1.nome, r2.nome
  FROM Refrigerante r1, Refrigerante r2
  WHERE r1.fabricante = r2.fabricante
          AND r1.nome < r2.nome;
```

Exercícios


FUNCIONARIO
Pnome
Minicial
Unome
Cpf
Datanasc
Endereco
Sexo
Salario
Cpf_supervisor
Dnr
DEPARTAMENTO
Dnome
Dnumero
Cpf_gerente
Data_inicio_gerente
LOCALIZACAO_DEP
Dnumero
Dlocal
PROJETO
Projnome
Projnumero
Projlocal
Dnum
TRABALHA_EM
Fcpf
Pnr
Horas
DEPENDENTE
Fcpf
Nome_dependente
Sexo
Datanasc
Parentesco

Exercícios (1)

1) Recuperar a data de nascimento e o endereço do(s) funcionário(s) que se chama(m) ‘João B. Silva’.
2) Selecione todos os atributos dos funcionários que trabalham no departamento de número 5.
3) Selecione todos os nomes de departamentos.
4) Recuperar todos os funcionários cujo endereço esteja em São Paulo, SP.

Exercícios (2)

5) Recuperar os nomes de todos os funcionários que não possuem supervisor.
6) Recuperar o nome e o endereço de todos os funcionários que trabalham no departamento ‘Pesquisa’.
7) Selecione todas as combinações de CPF de funcionário e nome de departamento.
8) Para cada projeto localizado em ‘Mauá’, liste o número do projeto, o número do departamento que o controla e o sobrenome, endereço e data de nascimento do gerente do departamento.

Exercícios (3)

9) Para cada funcionário, recupere o primeiro e o último nome do funcionário e o primeiro e último nome do seu supervisor imediato.

10) Mostrar os salários resultantes se cada funcionário que trabalha no departamento ‘Compras’ receber um aumento de 10% .

11) Recuperar uma lista dos funcionários e dos projetos em que estão trabalhando, ordenada por departamento e, dentro de cada departamento, ordenada alfabeticamente pelo sobrenome e depois pelo nome.

Referências bibliográficas

- *A First Course in Database Systems,*
  Ullman e Widom. 1997.
  Capítulo 5
- *Database Systems – The Complete Book*, Garcia-Molina, Ullman e Widom. 2002.
  Capítulo 6
- *Sistemas de Bancos de Dados* (6ª edição), Elmasri e Navathe. 2010.
  Capítulos 4 e 5