



# MODELO DE DADOS RELACIONAL E SQL

**Dr. Fabiano Cavalcanti Fernandes**  
**fabiano.fernandes@ifb.edu.br**

Brasília - DF



# Esquema e Instância

## Esquema

- A descrição da organização dos dados de um BD.
- Um esquema de BD é especificado durante o projeto do BD e não é freqüentemente modificado

## Instância

- Os dados armazenados em um BD, em um momento específico, são denominados *instâncias* do BD - (fotografia do BD em um instante)



# Modelo de Dados Relacional

O Modelo Relacional representa os dados em um BD, por meio de um conjunto de relações.

Estas relações contém informações sobre entidades ou relacionamentos existentes no domínio da aplicação utilizada como alvo da modelagem.

Informalmente uma relação pode ser considerada como uma tabela de valores, onde cada linha desta tabela representa uma coleção de valores de dados inter-relacionados.



# Modelo de Dados Relacional

Os nomes fornecidos às tabelas e às suas colunas podem auxiliar na compreensão do significado dos valores armazenados em cada uma de suas linhas (registros).

Na terminologia do Modelo Relacional, cada linha da tabela é chamada de tupla, a tabela é denominada relação, o nome da coluna é denominado atributo da relação, e o conjunto de valores que cada atributo pode assumir em uma determinada relação, forma o seu domínio.

# Modelo de Dados Relacional

O domínio consiste de um grupo de valores atômicos a partir dos quais um ou mais atributos retiram seus valores reais.

O esquema de uma relação consiste de um conjunto de atributos que descrevem as características dos elementos a serem modelados. O número (quantidade) de atributos em uma relação consiste no grau da relação. (fornecedor tem 4) .

**Fornecedor**

The diagram illustrates the components of a relational database model using a table example. The table has four columns: Código, Nome, Categoria, and Cidade. The rows represent data instances. Annotations include: a grey arrow pointing to the 'Cidade' column header labeled 'domínio'; a red bracket on the right side of the table labeled 'Esquema'; a blue bracket on the right side of the table labeled 'Relação'; a green bracket on the right side of the table labeled 'Instâncias'; and a pink oval around the last row (F4, Matilde, 20, Brasília) labeled 'tupla'.

Código	Nome	Categoria	Cidade
F1	Paulo	20	Lins
F2	César	10	Palmas
F3	Carlos	30	Lins
F4	Matilde	20	Brasília

# Modelo de Dados Relacional

A instância de uma relação consiste no conjunto de valores que cada atributo, definido no esquema, assume em um determinado instante, formando o conjunto de tuplas. As instâncias das relações formam os dados que são armazenados no BD.

## Exemplo:

O domínio do atributo **Cidade** consiste no conjunto de todos os nomes válidos de cidades (Lins, Palmas e Brasília).

O esquema do Fornecedor (código,nome,categoria,cidade) possui grau 4 (4 atributos).

As instâncias são os dados que são armazenados no BD.



# Modelo de Dados Relacional

## Características das Relações

- Não há tuplas duplicadas em uma relação.
- A ordem das tuplas não é relevante para diferenciar uma relação de outra.
- Existe ordem dos valores nas tuplas, uma vez que o cabeçalho da relação (ou esquema) é definido como um conjunto de atributos.
- Os valores dos atributos devem ser atômicos, não sendo divisíveis em componentes. Atributos multivalorados são representados por meio de uma outra relação e atributos compostos pelos seus componentes.



# Modelo de Dados Relacional

## Chaves e Restrições de Integridade Relacional

Como não pode haver uma tupla repetida (duplicada) em uma instância da relação, isto significa que é possível identificar cada tupla separadamente uma da outra, por meio da escolha de algum atributo (ou conjunto de atributos).

Este atributo (ou atributos) identificam uma única tupla da relação e são conhecidos como chave da relação.

Com a definição de uma chave para identificação na relação, esta chave será conhecida como chave primária. Quando a chave primária for composta por mais que um atributo ela será denominada chave primária composta.





# Modelo de Dados Relacional

## Restrições de Integridade

Restrições de Chaves: cada atributo das chaves candidatas deve possuir valor único em todas as tuplas da relação.

Restrição de Integridade de Entidade: uma chave primária não pode assumir valor nulo em qualquer tupla da relação.

Restrição de Integridade Referencial: uma tupla em uma relação que se refere a outra relação, deve se referenciar a uma tupla existente nesta relação. Com esta definição tem-se um novo tipo de chave denominada *estrangeira*.

Restrições de Integridade Semântica: se referem mais especificamente sobre valores ou características que determinados atributos podem assumir no contexto de uma determinada aplicação (por exemplo sexo).



# Modelo de Dados Relacional

## Álgebra Relacional

Consiste em operadores que realização ações sobre as relações. Os operadores da álgebra relacional podem ser divididos em duas categorias:

- Operadores de Conjuntos
- Operadores Relacionais

# Álgebra Relacional

## Operadores de Conjuntos

Estes operadores se aplicam as duas relações que obedecem à “**compatibilidade de união**” , ou seja, ambas as relações devem apresentar como esquema atributos que pertencam respectivamente aos mesmos domínios.

a) União ( $\cup$ ): o resultado da união de duas relações consiste no conjunto de todas as tuplas que pertençam a ambas as relações.

Exemplo:

Seja  $A$  = conjunto de tuplas dos fornecedores do estado de SP

$B$  = conjunto dos fornecedores da peça P\_1

$A$  união  $B$  ( $A \cup B$ ) = conjunto de tuplas dos fornecedores de SP  
ou que fornecem a peça P\_1 (ou ambos)

## Álgebra Relacional

b) Interseção ( $\cap$ ): o resultado da interseção de duas relações consiste no conjunto de todas as tuplas que aparecem ao mesmo tempo nas duas relações.

Exemplo:

Seja  $A$  = conjunto de tuplas dos fornecedores do estado de SP

$B$  = conjunto dos fornecedores da peça P\_1

$A$  interseção  $B$  ( $A \cap B$ ) = conjunto de tuplas dos fornecedores de SP e que forneçam a peça P\_1 (estão em ambos)

## Álgebra Relacional

c) Diferença (-): a diferença em duas relações (R e S por exemplo) consiste no conjunto de tuplas que aparecem na relação R, mas não aparecem na relação S.

Exemplo:

Seja  $A$  = conjunto de tuplas dos fornecedores do estado de SP

$B$  = conjunto dos fornecedores da peça P\_1

$A \text{ minus } B (A-B)$  = conjunto de tuplas dos fornecedores de SP e não fornecem a peça P\_1 (está em A mas não está em B)

$B \text{ minus } A (B-A)$  = conjunto de tuplas dos fornecedores que fornecer a peça P\_1 e que não são de SP (está em B mas não está em A)

Resultados diferentes

## Álgebra Relacional

d) Produto Cartesiano (X): aplica-se as duas relações que não precisam ser “compatíveis de união”, resultando em uma relação que apresenta tuplas formadas pela combinação dos atributos pertencentes a ambas as relações.

Exemplo:

Seja  $A$  = conjunto de todos os códigos dos fornecedores de SP

$B$  = conjunto de todos os códigos de peças

$A$  cartesiano  $B$  ( $A \times B$ ) = conjunto de todos os possíveis pares de códigos de fornecedores com os códigos de todas as peças

## Exercício de Fixação

- 1) Sejam as seguintes relações existentes em um banco de dados:

**Fornecedor (F)**

	Código	Nome	Id_Peça	Valor
1	23	Altar	10	35,00
2	35	Mecânica Jair	22	50,00
3	44	Eletrons	07	99,00
4	57	Thorque	22	47,00
5	89	Rápido	10	35,00

**Peças (P)**

	Código	Nome	Cor	Peso
1	07	Mola estreita	Prata	10 gr.
2	10	Correia lisa	Preto	0,5 gr.
3	22	Amortecedor	Preto	2000 gr.
4	35	Tambor	Azul	500 gr.

**Novos\_Fornecimentos (G)**

	Código	Nome	Id_Peça	Valor
1	57	Thorque	35	45,00
2	90	Solução Final	10	50,00

Seja uma nova relação representando os novos fornecedores: →

## Exercício de Fixação

Construa as relações resultantes das operações algébricas abaixo:

- a) União (de F com G)
- b) Interseção (de F com G)
- c) Diferença em relação ao fornecedor
- d) Produto cartesiano (de P com G)

Observação: entende-se  $F$  = fornecedor,  $P$  = peças e  $G$  = novos\_fornecimentos



## Exercício de Fixação - solução

a) União de F com G

$(F \cup G) =$

	Código	Nome	Id_Peça	Valor
1	23	Altar	10	35,00
2	35	Mecânica Jair	22	50,00
3	44	Eletrons	07	99,00
4	57	Thorque	22	47,00
5	89	Rápido	10	35,00
6	57	Thorque	35	45,00
7	90	Solução Final	10	50,00

b) Interseção de F com G

(mesmo fornecedor)

$(F \cap G) = \emptyset$

## Exercício de Fixação – continuação da solução

c) Diferença de F com G

(F - G) =

	Código	Nome	Id_Peça	Valor
1	23	Altar	10	35,00
2	35	Mecânica Jair	22	50,00
3	44	Eletrons	07	99,00
4	57	Thorque	22	47,00
5	89	Rápido	10	35,00

Diferença de G com F

(G - F) =

	Código	Nome	Id_Peça	Valor
1	57	Thorque	35	45,00
2	90	Solução Final	10	50,00

## Exercício de Fixação – continuação da solução

d) Produto cartesiano de P com G

$(P \times G) =$

	Código	Nome	Cor	Peso	Código	Nome	Id_Peça	Valor
1	07	Mola estreita	Prata	10 gr.	57	Thorque	35	45,00
2	07	Mola estreita	Prata	10 gr.	90	Solução Final	10	50,00
3	10	Correia lisa	Preto	0,5 gr.	57	Thorque	35	45,00
4	10	Correia lisa	Preto	0,5 gr.	90	Solução Final	10	50,00
5	22	Amortecedor	Preto	2000 gr.	57	Thorque	35	45,00
6	22	Amortecedor	Preto	2000 gr.	90	Solução Final	10	50,00
7	35	Tambor	Azul	500 gr.	57	Thorque	35	45,00
8	35	Tambor	Azul	500 gr.	90	Solução Final	10	50,00

# Álgebra Relacional

## Operadores Relacionais

1) Operação de Seleção (*select*): quando aplicado resulta em uma relação contendo tuplas com os mesmos atributos da relação que satisfazem a uma determinada condição de seleção. É um operador unário, sendo executado sobre apenas uma relação, uma tupla de cada vez.

$\sigma$  [<condição de seleção>] (<nome da relação>)

Em geral, pode-se usar os operadores relacionais ( $\neq$ ,  $=$ ,  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ) na operação de seleção, além da condição ser composta por mais que um predicado condicional, interligados pelos conectivos *E* ( $\wedge$ ) e *OU* ( $\vee$ ) lógicos.

## Álgebra Relacional

Exemplo: para a relação *fornecedor* a seguir tem-se:

### Fornecedor

	Código	Nome	Id_Peça	Valor
1	23	Altar	10	35,00
2	35	Mecânica Jair	22	50,00
3	44	Eletrons	07	99,00
4	57	Thorque	22	47,00
5	44	Eletrons	35	52,00

a)  $\sigma$  [Código = 44] (Fornecedor)

	Código	Nome	Id_Peça	Valor
1	44	Eletrons	07	99,00
2	44	Eletrons	35	52,00

# Álgebra Relacional

b)  $\sigma$  [Código = 35] (Fornecedor)

	Código	Nome	Id_Peça	Valor
1	35	Mecânica Jair	22	50,00

c)  $\sigma$  [Código = 89] (Fornecedor)

	Código	Nome	Id_Peça	Valor
0				

2) Operação de Projeção (*project*): seleciona atributos de uma relação de acordo com uma lista de atributos. Os atributos são exibidos na mesma ordem que aparecem na lista. Como resultado é uma relação que não pode existir repetições nas tuplas produzidas

$\pi\langle\text{lista de atributos}\rangle (\langle\text{nome da relação}\rangle)$

Exemplo: para a mesma relação do exemplo anterior (*Fornecedor*) o resultado da projeção seguinte seria:

$\pi \text{ Nome, Id_Peça } (\text{Fornecedor})$

	Nome	Id_Peça
1	Altar	10
2	Mecânica Jair	22
3	Eletrons	07
4	Thorque	22
5	Eletrons	35

Uma operação relacional sempre resulta em uma outra relação que pode ser usada na elaboração de consultas mais complexas.

### **Fornecedor**

	Código	Nome	Id_Peça	Valor
1	23	Altar	10	35,00
2	57	Thorque	35	45,00
3	44	Eletrons	07	99,00
4	57	Thorque	22	47,00

$\pi$  Código, Valor ( $\sigma$  [Nome = “Thorque”] (Fornecedor) )

	Código	Valor
1	57	45,00
2	57	47,00

→ Observe que ao invés de declarar uma relação como argumento na operação de projeção, inseriu-se uma expressão que evoluirá para uma relação.



## Álgebra Relacional

3) Operação de Junção (*join*): é utilizada para combinar tuplas relacionadas de duas relações (operação binária) em uma tupla simples. Esta combinação é realizada de acordo com uma condição indicada.

$\Theta$  [<condição>] (<nome das relações>)

Exemplo: para as relações à seguir observe a junção efetuada entre **Pecas** e **Novos\_Fornecimentos**

# Álgebra Relacional

## Peças

	Código	Nome	Cor	Peso
1	07	Mola estreita	Prata	10 gr.
2	10	Correia lisa	Preto	0,5 gr.
3	22	Amortecedor	Preto	2000 gr.
4	35	Tambor	Azul	500 gr.

## Novos\_Fornecimentos

	Código	Nome	Id_Peça	Valor
1	57	Thorque	35	45,00
2	90	Solution	10	50,00

$\Theta$  [Id\_Peça = Código] (Novos\_Fornecimentos, Peças)

	Código	Nome	Cor	Peso	Código	Nome	Id_Peça	Valor
1	<b>10</b>	Correia lisa	Preto	0,5 gr.	90	Solução Final	<b>10</b>	50,00
2	<b>35</b>	Tambor	Azul	500 gr.	57	Thorque	<b>35</b>	45,00



# Modelo de Dados Relacional

## Funções Agregadas

a) Funções Agregadas: consistem em funções que podem ser aplicadas a valores numéricos. Elas são: Average (media), Count (contador), Sum (soma), Maximum (maior) e Minimum (menor).

## Exercício de Fixação

01) Usando as relações a seguir, escreva a expressão em álgebra relacional que representa o item da solicitação e elabore as relações resultantes das seguintes operações:

- a) União de B com Y
- b) Interseção de B com Y
- c) Diferença de B com Y e de Y com B
- d) Produto cartesiano de B com Y
- e) Projeção de Id\_agência, Cidade e Estado sobre a agência
- f) Seleção dos clientes de Brasília
- g) Junção da conta com a agência
- h) Projeção da agência, tipo conta e cidade da seleção de contas com saldo não negativo
- i) Projeção do nome, saldo, estado da seleção do estado diferente de “DF” da junção do cliente com a conta

**Banco (Y)**

	Código	Nome
1	001	Brasil
2	350	Real

**Bancos (B)**

	Código	Nome
1	001	Brasil
2	104	C.E.F.
3	341	Itaú

Exercício de Fixação – relações do exercício 01

## Agência (A)

	Id_Agência	Rua	Número	Compl.	Bairro	Cidade	Estado	Banco
1	5101	W 3	505	Cnj. 3	A.Norte	Brasília	DF	001
2	930	L 2	407	Bloco A	A.Sul	Brasília	DF	001
3	4146	Q.S.	07	Lote 1	Águas Claras	Taguatinga	DF	341

## Conta (C)

	Conta	Tipo_Conta	Saldo	Agência
1	59431	Poupança	1000,00	4146
2	47856	Corrente	- 50,00	930
3	30124	Corrente	200,00	4146

## Conta\_Cliente (CC)

	N_Conta	N_CPF
1	59431	100
2	47856	200
3	30124	300
4	47856	400

## Cliente (L)

	CPF	Nome	Fone_resid.	Cidade	Estado
1	100	João Castro	4563760	Brasília	DF
2	200	José Sechi	3576721	Brasília	DF
3	300	Ana Moraes	3787289	Taguatinga	DF
4	400	Maria Alves	4684592	Luziânia	GO

## Exercício de Fixação – exercício 01

a)  $B \cup Y =$

b)  $B \cap Y =$

c)  $B - Y =$

$Y - B =$

d)  $B \times Y =$

e)  $\pi_{\text{Id\_agência,Cidade,Estado}} (A) =$

f)  $\sigma [\text{Cidade} = \text{"Brasília"}] (L) =$

g)  $\Theta [\text{Agência} = \text{Id\_Agência}] (C, A) =$

h)  $\pi_{\text{Agência,Tipo\_Conta,Cidade}} (\sigma_{[\text{Saldo} \geq 0]} (C, A)) =$

i)  $\pi_{\text{Nome,Estado,Saldo}} (\sigma_{[\text{Estado} \neq \text{"DF"}]} (\Theta_{[\text{CPF}=\text{N\_CPF}]} (L, CC) \wedge$

$\Theta_{[\text{N\_Conta}=\text{Conta}]} (CC, C) ) ) =$



# Cálculo Relacional

- Permite a descrição da consulta desejada sem especificar os procedimentos para obtenção das informações
- Na lógica de primeira ordem pode-se pensar como uma linguagem de consulta de duas formas:
  - Cálculo relacional de tuplas
  - Cálculo relacional de domínio
- A diferença está no nível em que são utilizadas as variáveis:
  - Nível de Atributo para os Domínios
  - Nível de Tuplas



# Introdução

- O cálculo é uma linguagem de consulta formal onde escrevemos uma expressão declarativa para especificar uma solicitação de recuperação
- A expressão especifica O QUE será recuperado, em vez de COMO será recuperado
  - Linguagem de consulta **não-procedural**, o que o difere **da álgebra**, onde descrevemos uma **sequência de operações**
  - Qualquer consulta que possa ser expressa na álgebra pode ser expressa no cálculo
  - Ambas tem o mesmo poder de expressão





# Introdução

- Existem consultas em linguagens que não podem ser expressas na álgebra ou no calculo relacional
  - Algumas linguagens de consulta são mais completas que a álgebra e o cálculo, devido as operações de agregação e ordenamento

# Cálculo Relacional de Tuplas

- Expressão geral:

$$\{ t \mid \text{COND}(t) \}$$

$t$  = variável de tupla

$\text{COND}(t)$  = é uma expressão condicional envolvendo  $t$

Resultado da consulta = conjunto de todas as tuplas  $t$  que satisfazem  $\text{COND}(t)$

- Exemplo: encontrar **todas as tuplas** de empregado com salário acima de 1200,00

Notação Silberchatz:  $\{ t \mid t \in \text{empregado} \wedge t[\text{salario}] > 1200 \}$

Notação Navathe:  $\{ t \mid \text{empregado}(t) \text{ AND } t.\text{salario} > 1200 \}$

**Leitura:** o conjunto de todas as tuplas  $t$  tal que exista uma tupla  $t$  na relação empregado para a qual o valor no atributo *salario* seja maior que 1200 reais



# Cálculo Relacional de Tuplas

- Na consulta  $\{ t \mid \text{empregado}(t) \text{ AND } t.\text{salario} > 1200 \}$

Cada tupla de EMPREGADO que satisfaça a condição de  $\text{salario} > 1200$  é recuperada

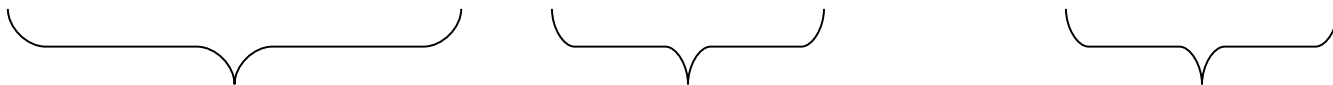
$t.\text{Salario}$  referencia o atributo  $\text{salario}$  da variavel de tupla  $t$  (similar ao SQL)

# Cálculo Relacional de Tuplas

- Na consulta  $\{ t \mid \text{empregado}(t) \text{ AND } t.\text{salario} > 1200 \}$

- Para recuperar apenas alguns atributos ao invés da tupla toda:

$\{ t.\text{nome}, t.\text{sobrenome} \mid \text{empregado}(t) \text{ AND } t.\text{salario} > 1200 \}$



- Equivale a consulta SQL:

```
SELECT t.nome, t.sobrenome
FROM empregado t
WHERE t.salario > 1200
```

- Equivale a consulta da Álgebra:

—  $\pi_{\text{nome, sobrenome}} (\sigma_{\text{salario} > 1.200} (\text{Empregado}))$

# O que é preciso informar numa expressão de cálculo de tuplas?

1. Uma relação  $R(t)$ 
  1. Equivalente a cláusula FROM do SQL
2. Uma condição para selecionar tuplas
  1. Equivalente a cláusula WHERE do SQL e SELECAO da algebra
  2. A condição é após o símbolo |
3. Conjunto de atributos a ser recuperado
  1. Equivalente ao SELECT do SQL

$\{ \underbrace{t.nome, t.sobrenome}_{\text{atributos}} \mid \underbrace{empregado(t)}_{\text{relação}} \underbrace{AND t.salario > 1200}_{\text{condição}} \}$

# Expressões e Fórmulas

- Expressão GERAL:
  - $\{t_1.A_1, t_2.A_2, \dots, t_n.A_n, \mid \text{COND } (t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$

Os  $t$  são variáveis de tuplas  
Os  $A$  são um atributo da relação

A relação faz parte da condição

# Expressões e Fórmulas

- Expressão GERAL:

- $\{t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid \text{COND}(t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$

variáveis livres

predicado aplicado às variáveis  
livres

- Variável livre

- assume valores de tuplas de uma ou mais relações
  - constitui a resposta da consulta
  - Aparece do lado esquerdo da |

- Predicado

- **EXPRESSÃO LÓGICA** que, se verdadeira para determinados valores das variáveis livres retorna os valores destas variáveis na resposta da consulta

# A Expressão Lógica

$\{t_1.A_1, t_2.A_2, \dots, t_n.A_n, \mid \text{COND } (t_1, t_2, \dots, t_n, t_{n+1}, t_{n+2}, \dots, t_{n+m})\}$

- Uma expressão pode ter **várias condições** C (fórmula), ligadas por operadores lógicos AND, OR, e NOT
- Se  $C_1$  e  $C_2$  são uma condição, então  $(C_1 \text{ AND } C_2)$ ,  $(C_1 \text{ OR } C_2)$ , **NOT** ( $C_1$ ) e **NOT** ( $C_2$ )
- Valores verdade para as condições:
  - $(C_1 \text{ AND } C_2)$  é verdadeiro se ambos  $C_1$  e  $C_2$  forem verdadeiros, caso contrário é FALSO
  - $(C_1 \text{ OR } C_2)$  é falso se ambos  $C_1$  e  $C_2$  forem falsos, caso contrário é verdadeiro
  - **NOT** ( $C_1$ ) é verdadeiro se  $C_1$  for FALSO; é FALSO se  $C_1$  for VERDADEIRO
  - **NOT** ( $C_2$ ) é verdadeiro se  $C_2$  for FALSO; é FALSO se  $C_2$  for VERDADEIRO





# Exemplo

- Q1: recupere a data de nascimento e o endereço dos empregados Joao Oliveira

{ t.dataNasc, t.endereco | empregado(t) AND t.nome='Joao' AND  
t.sobrenome='Oliveira' }

# Exercícios – Dado o esquema relacional

- Ambulatório (númeroA, andar, capacidade)
  - Médico (CRM, nome, idade, cidade, especialidade, #númeroA)
  - Paciente (RG, nome, idade, cidade, doença)
  - Consulta (#CRM, #RG, data, hora)
  - Funcionário (RG, nome, idade, cidade, salário)
- 
- 1) buscar os dados dos pacientes que estão com sarampo
  - 2) buscar os dados dos médicos ortopedistas com mais de 40 anos
  - 3) buscar os dados das consultas, exceto aquelas marcadas para os médicos com CRM 46 e 79
  - 4) buscar o número dos ambulatórios do quarto andar que tenham capacidade igual a 50 ou tenham número superior a 10



# Quantificadores Existencial e Universal

- Usados para consultas com mais de uma relação
- Quantificador existencial:  $\exists$ 
  - Chamado EXISTE
- Quantificador universal:  $\forall$ 
  - Chamado PARA TODO
- Em uma expressão uma tupla  $t$  é **livre** quando **não** for quantificada em uma cláusula  $(\exists t)$  ou  $(\forall t)$ , caso contrário é **limite**

# Quantificadores Existencial e Universal

- Valores verdade para os quantificadores
- Sendo C uma condição:
  - $(\exists t)(C)$  é VERDADEIRA se C for verdadeira para ***alguma (pelo menos 1) tupla*** das ocorrências livres de t em C, caso contrário é FALSA
  - $(\forall t)(C)$  é VERDADEIRA se C for VERDADEIRA para ***todas as tuplas (no universo)*** designadas para ocorrências livres de t em C, caso contrário é FALSA
-

# Quantificador Existencial

- Notação **(Silberchatz)**

$$\exists t \in C(t)$$

- Notação **(Navathe)**  
 **$(\exists t)(C)$**

- Define uma variável **não-livre**  $t$  (associada sempre a uma relação  $R$ ) e avalia uma condição  $C(t)$  para ela
  - interpretação: verifica se existe **alguma** tupla  $t$  em  $R$  para o qual  $C(t)$  seja verdadeiro

-

# Quantificador Existencial

- Relações Exemplo
- Projeto (codProj, tipo, descr, **codDep**)
- ProjetoEmpregado (codProj, codEmp, dataIni, tempoAI)
- Empregado (codEmp, nome, salario, **codDep**)
- Departamento (codDep, descricao)

# Quantificador Existencial

- TABELAS: EXEMPLO
- Consulta 1: recupere o **nome** e o **endereço** de todos os **empregados** que trabalham no **departamento de PESQUISA**

{ t.nome, t.endereco | empregado(t) AND  
( $\exists d$ ) (departamento(d) AND d.nome='PESQUISA' AND  
d.codDepto=t.codDepto)) }

Condição de SELEÇÃO da Algebra

Condição de JUNÇÃO

Equivale ao  
C da  
Formula  
( $\exists d$ )(C)

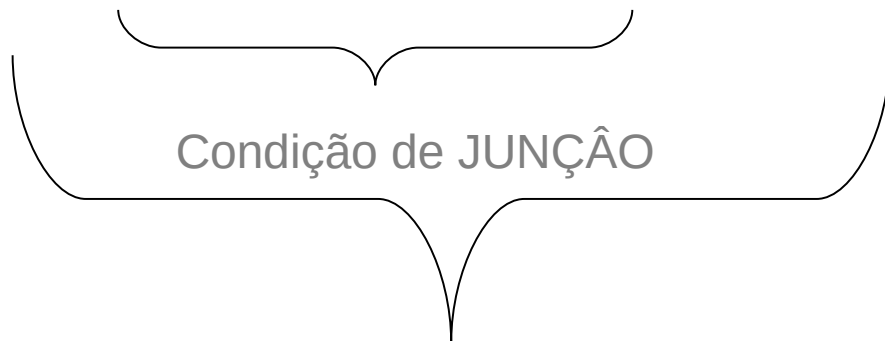
- Neste exemplo, t é uma tupla livre, d não é livre

# Quantificador Existencial

- Consulta 2: Encontre o nome do projeto dos empregados do departamento 5

## MAIS EXEMPLO

- Neste exemplo,  $t$  é uma tupla livre,  $d$  não é livre  
 $\{ t.descr, | projeto(t) \text{ AND } d.codDep=5 \text{ AND } (Empregado(d) \text{ AND } d.codDepo=t.codDepo) \}$



Condição de SELEÇÃO da Algebra



# Quantificador Universal

- Notação

$\forall t \in C(t)$  (Silberchatz)  
 $(\forall t) (C)$  (Navathe)

- Interpretação

- verifica se **toda** tupla  $t$  satisfaz  $C(t)$

- Usado para formular consultas que

- envolvem a associação com tuplas de relações que não vão para a resposta
- similar ao princípio da **divisão** da álgebra relacional

# Exemplos de Consultas c/ o Quantificador Existencial

Projeto (codProj, tipo, descr, **codDep**)

ProjetoEmpregado (codProj, codEmp, dataIni, tempoAI)

Empregado (codEmp, nome, salario, **codDep**)

Departamento (codDep, descricao)

- Consulta: nomes dos empregados que não trabalham em projetos

$\{e.nome \mid empregado(e) \text{ AND } ((\forall x)(\text{NOT} (projetoEmpregado(x)) \text{ OR } \text{NOT} (x.codEmp=e.codEmp))))\}$

\* Para cada tupla: **retorna o nome do empregado se, somente se, na tupla x ( $\forall x$ ) de projetoEmpregado este empregado não exista**

- EQUIVALENTE a

$\{e.nome \mid empregado(e) \text{ AND } (\text{NOT}(\exists x)(projetoEmpregado(x) \text{ AND } x.codEmp=e.codEmp))\}$

\* Para cada tupla, **retorna o nome do empregado somente se não existir  $\text{NOT}(\exists x)$  uma tupla em projetoEmpregado com este empregado**

# Exercícios – Dado o esquema relacional

- Ambulatório (numeroA, andar, capacidade)
  - Médico (CRM, nome, idade, cidade, especialidade, #numeroA)
  - Paciente (RG, nome, idade, cidade, doença)
  - Consulta (#CRM, #RG, data, hora)
  - Funcionário (RG, nome, idade, cidade, salário)
- 
- 5) buscar o nome e a especialidade de todos os médicos
  - 6) buscar o número dos ambulatórios do terceiro andar
  - 7) buscar o CRM dos médicos e as datas das consultas para os pacientes com RG 122 e 725
  - 8) buscar os números dos ambulatórios, exceto aqueles do segundo e quarto andares, que suportam mais de 50 pacientes
  - 9) buscar o nome dos médicos que têm consulta marcada e as datas das suas consultas
  - 10) buscar o número e a capacidade dos ambulatórios do quinto andar e o nome dos médicos que atendem neles
  - 11) buscar os nomes dos médicos ortopedistas com consultas marcadas para o período da manhã (7hs-12hs) do dia 15/04/03
  - 12) buscar os nomes dos pacientes, com consultas marcadas para o médicos João Carlos Santos, que estão com pneumonia

# Exercícios – Dado o esquema relacional

- Ambulatório (numeroA, andar, capacidade)
- Médico (CRM, nome, idade, cidade, especialidade, #numeroA)
- Paciente (RG, nome, idade, cidade, doença)
- Consulta (#CRM, #RG, data, hora)
- Funcionário (RG, nome, idade, cidade, salário)

- 13) buscar os nomes e RGs dos funcionários que recebem salários abaixo de R\$ 300,00 e que não estão internados como pacientes
- 14) buscar os números dos ambulatórios onde nenhum médico dá atendimento
- 15) buscar os nomes e RGs dos funcionários que estão internados como pacientes
- 16) Buscar o nome dos médicos que nunca consultaram