

# 21053 - Fundamentos de Bases de Dados

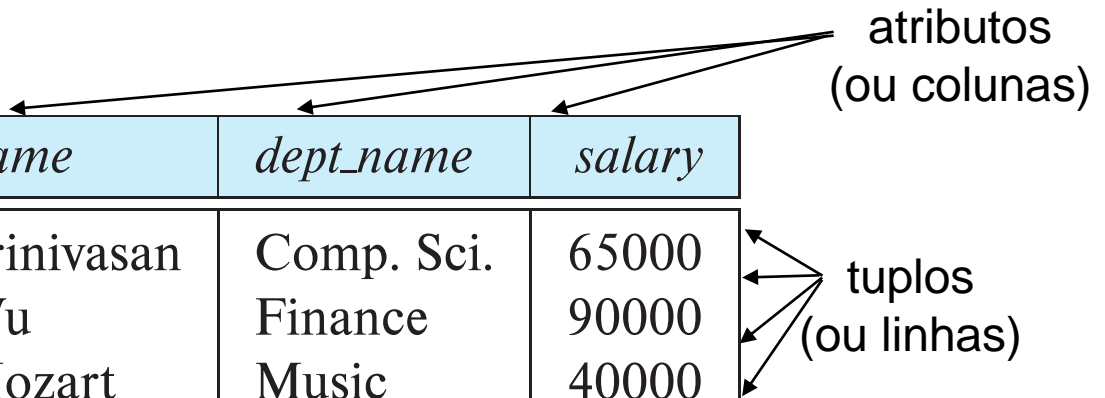
**Professor:**

**Paulo Pombinho**

## **S2 – Introdução ao Modelo Relacional**



# Exemplo de uma tabela “*instructor*”



<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

# Esquema Relacional e Instância

- $A_1, A_2, \dots, A_n$  são atributos
- $R = (A_1, A_2, \dots, A_n)$  é um esquema de relação

Exemplo:

*instructor = (ID, name, dept\_name, salary)*

- Uma relação  $r$  definida sobre o esquema  $R$  é apresentada como  $r(R)$ .
- Os valores atuais de uma relação são especificados por uma tabela
- Um elemento  $t$  da relação  $r$  é chamado de um *tuplo* e é representado por uma *linha* na tabela

# Atributos

- O conjunto de valores permitidos para cada atributo é chamado de **domínio** do atributo
- Os valores de atributos são (normalmente) necessariamente **atômicos**; isto é, indivisíveis
- O valor especial ***null*** é um membro de todos os domínios. Indicado que o valor é "desconhecido"
- O valor nulo causa complicações na definição de muitas operações

# As relações são desordenadas

- A ordem dos tuplos é irrelevante (os tuplos podem ser armazenados por ordem arbitrária)
- Exemplo: relação *instructor* com tuplos não ordenados

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

# Esquema de base de dados

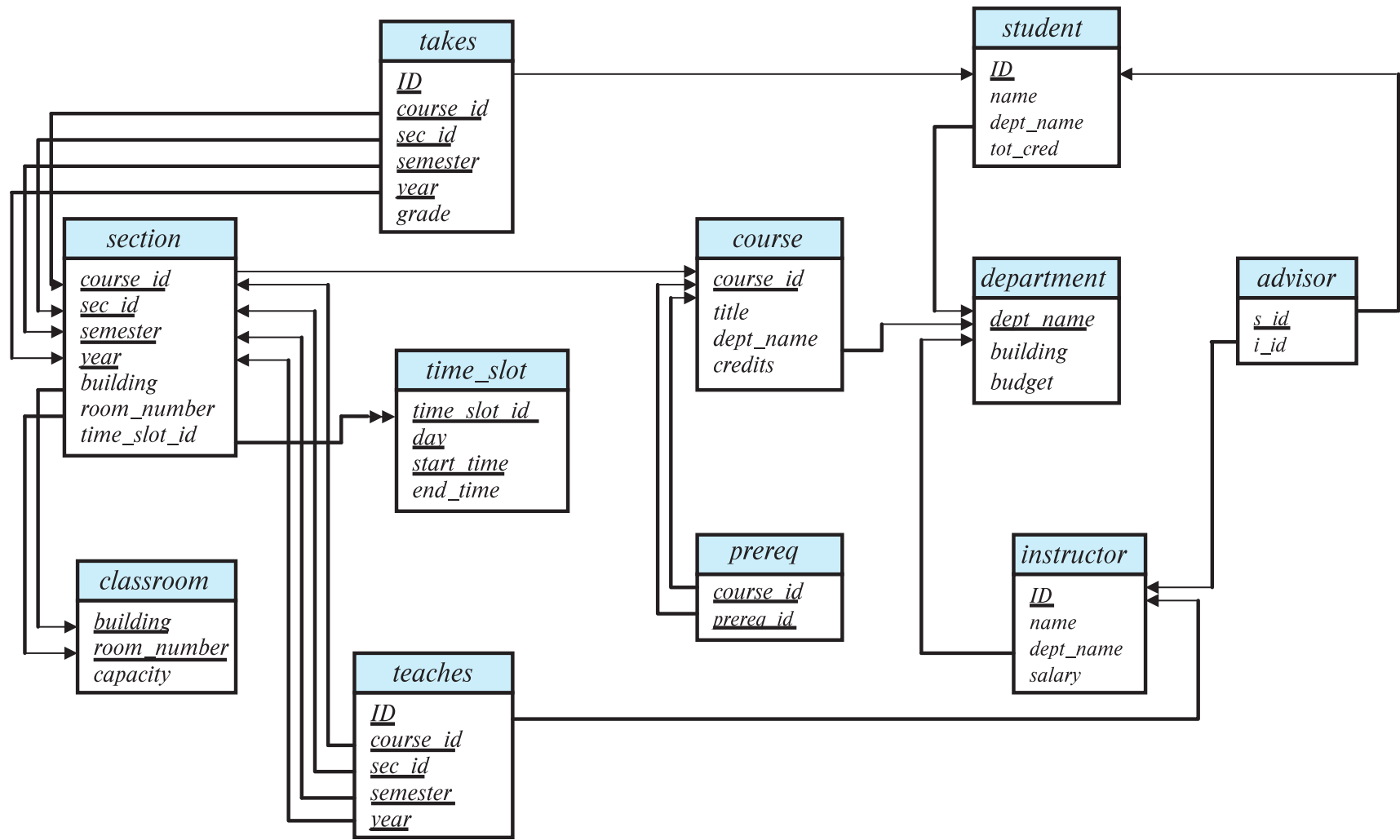
- Esquema de base de dados -- é a estrutura lógica da base de dados.
- Instância da base de dados --. é uma “captura” dos dados na base de dados num dado instante no tempo
- Exemplo:
  - esquema: *instructor (ID, name, dept\_name, salary)*
  - Instância:

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

# Chaves

- Seja  $K \subseteq R$
- $K$  é uma **superchave** de  $R$  se valores para  $K$  são suficiente para identificar um tuplo único em cada relação possível  $r(R)$ 
  - Exemplo:  $\{ID\}$  e  $\{ID, name\}$  são ambos superchaves de *instructor*.
- Superchave  $K$  é uma **chave candidata** se  $K$  é mínimo  
Exemplo:  $\{ID\}$  é uma chave candidate de *Instructor*
- Uma das chaves candidatas é selecionada para ser a **chave primária**.
  - Qual?
- Restrição com **chave estrangeira** : Valor numa relação deve aparecer noutra
  - **Referencia a** relação
  - Exemplo: *dept\_name* em *instructor* é uma chave estrangeira de *instructor* referenciando *department*

# Esquema de Base de Dados “University”





# Linguagens de consulta relacional

- Procedimental versus não procedimental, ou declarativa
- Linguagens "puras":
  - Álgebra relacional
  - Cálculo relacional de tuplo
  - Cálculo relacional de domínio
- As 3 linguagens acima são equivalentes no poder de computação
- Vamos concentrar-nos na álgebra relacional
  - Não Turing-completa
  - Consiste em 6 operações básicas

# Álgebra Relacional

- Uma linguagem procedimental constituída por um conjunto de operações que tomam uma ou duas relações como contributo e produzem uma nova relação como resultado.
- Seis operadores básicos
  - seleção:  $\sigma$
  - projeção:  $\Pi$
  - união:  $\cup$
  - diferença:  $-$
  - Produto cartesiano:  $\times$
  - renomeação:  $\rho$

# Seleção

- O operador de **seleção** seleciona tuplos que satisfazem um dado predicado.
- Notação:  $\sigma_p(r)$
- $p$  é chamado de **predicado de seleção**
- Exemplo: selecionar os tuplos da relação *instructor* onde o professor está no departamento de Física.
  - Consulta

$\sigma_{dept\_name="Physics"}(instructor)$

- Resultado

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

# Seleção

- Permitimos comparações usando

$=, \neq, >, \geq, <, \leq$

no predicado de seleção.

- Podemos combinar vários predicados num predicado maior usando os conectores:

$\wedge$  (**e**),  $\vee$  (**ou**),  $\neg$  (**negação**)

- Exemplo: Encontre os professores de Física com um salário superior a \$90.000, escrevemos:

$\sigma_{dept\_name="Physics" \wedge salary > 90,000} (instructor)$

- O predicado selecionado pode incluir comparações entre dois atributos.
  - Exemplo, encontrar todos os departamentos cujo nome é o mesmo que o nome do seu edifício:
  - $\sigma_{dept\_name=building} (department)$

# Projeção

- Uma operação unária que devolve a sua relação de argumento, com certos atributos deixados de fora.

- Notação:

$$\Pi_{A_1, A_2, A_3 \dots A_k}(r)$$

onde  $A_1, A_2, \dots, A_k$  são nomes de atributos e  $r$  é um nome de relação.

- O resultado é definido como a relação das colunas  $k$  obtidas apagando as colunas que não estão listada
- Linhas duplicadas são removidas do resultado, uma vez que as relações são conjuntos

# Exemplo de Projeção

- Exemplo: eliminar o atributo *dept\_name* da tabela *instructor*
- Consulta:

$\Pi_{ID, name, salary}(instructor)$

- Resultado:

<i>ID</i>	<i>name</i>	<i>salary</i>
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

# Composição das Operações Relacionais

- O resultado de uma operação de álgebra relacional é uma relação e, como tal, as operações podem ser compostas em conjunto numa expressão de álgebra relacional.
- Considere a consulta -- Encontre os nomes de todos os instrutores do departamento de Física.

$$\Pi_{name}(\sigma_{dept\_name = "Physics"}(instructor))$$

- Em vez de dar o nome de uma relação como argumento da operação de projeção, damos uma expressão que se avalia numa relação.

# Produto cartesiano

- O produto cartesiano (denotado por  $\times$ ) permite-nos combinar informações de quaisquer duas relações.
- Exemplo: o produto cartesiano da relação *instructor* e *teaches* é escrito como:

*instructor*  $\times$  *teaches*

- Construimos um tuplo do resultado de cada par de tuplos possíveis: um da relação do *instructor* e outro da relação de *teaches*
- Uma vez que o ID do instrutor aparece em ambas as relações distinguimos entre estes atributos, anexando ao atributo o nome da relação a partir da qual o atributo veio originalmente.
  - *instructor.ID*
  - *teaches.ID*



# A tabela *instructor* X *teaches*

<i>instructor.ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>	<i>teaches.ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

# Natural Join (Junção)

- O Produto Cartesiano

*instructor* X *teaches*

associa cada tuplo de *instructor* com cada tuplo de *teaches*.

- A maioria das linhas resultantes têm informações sobre instrutores que NÃO ensinaram um determinado curso.

- Para obter apenas os tuplos de “*instructor* X *teaches*” que dizem respeito aos instrutores e aos cursos que ensinaram, escrevemos:

$\sigma_{instructor.id = teaches.id} (instructor \times teaches)$

- Obtemos apenas os tuplos de “*instructor* X *teaches*” que dizem respeito aos instrutores e aos cursos que lecionou.

- O resultado desta expressão é mostrado no próximo slide

# Natural Join (Junção)

- A tabela correspondente a:

$$\sigma_{instructor.id = teaches.id} (instructor \times teaches)$$

<i>instructor.ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>	<i>teaches.ID</i>	<i>course_id</i>	<i>sec_id</i>	<i>semester</i>	<i>year</i>
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

# Natural Join (Junção)

- A operação **join** permite-nos combinar uma operação de seleção e um produto cartesiano numa única operação.
- Considere as relações  $r (R)$  e  $s (S)$
- Seja “theta” um predicado sobre atributos no esquema R “união” S. A operação  $\text{join } r \bowtie_{\theta} s$  é definida como se segue:

$$r \bowtie_{\theta} s = \sigma_{\theta} (r \times s)$$

- Assim,

$$\sigma_{\text{instructor.id} = \text{teaches.id}} (\text{instructor} \times \text{teaches})$$

- Pode ser escrito de forma equivalente como

$$\text{instructor} \bowtie_{\text{Instructor.id} = \text{teaches.id}} \text{teaches.}$$

# União

- A União permite-nos combinar duas relações
- Notação:  $r \cup s$
- Para que  $r \cup s$  seja válida.
  1.  $r, s$  devem ter a mesma **aridade** (mesmo número de atributos)
  2. Os domínios do atributo devem ser **compatíveis** (exemplo: 2ª coluna de  $r$  lida com o mesmo tipo de valores como a 2ª coluna de  $s$ )
- Exemplo: para encontrar todos os cursos lecionados no outono de 2017, ou na primavera de 2018, ou em ambos

$$\Pi_{course\_id} (\sigma_{semester="Fall" \wedge year=2017}(section)) \cup \Pi_{course\_id} (\sigma_{semester="Spring" \wedge year=2018}(section))$$

# União

- Resultado de:

$\Pi_{course\_id} (\sigma_{semester="Fall" \wedge year=2017}(section)) \cup$

$\Pi_{course\_id} (\sigma_{semester="Spring" \wedge year=2018}(section))$

<i>course_id</i>
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

# Intersecção

- A operação de intersecção permite-nos encontrar tuplos que estão nas relações de entrada.
- Notação:  $r \cap s$
- Assumir que:
  - $r, s$  têm a mesma aridade
  - atributos de  $r$  e  $s$  são compatíveis
- Exemplo: Encontre o conjunto de todos os cursos lecionados no outono de 2017 e primavera de 2018.

$$\Pi_{course\_id} (\sigma_{semester="Fall" \wedge year=2017}(section)) \cap \Pi_{course\_id} (\sigma_{semester="Spring" \wedge year=2018}(section))$$

- Resultado

<i>course_id</i>
CS-101

# Diferença

- A operação de diferença permite-nos encontrar tuplo que estão numa relação, mas não estão em outra.
- Notação  $r - s$
- As diferenças devem ser feitas entre relações **compatíveis**.
  - $r$  e  $s$  deve ter a mesma aridade
  - Domínio dos atributos de  $r$  e  $s$  têm de ser compatíveis
- Exemplo: para encontrar todos os cursos lecionados no outono de 2017, mas não na primavera de 2018

$$\Pi_{course\_id} (\sigma_{semester="Fall" \wedge year=2017}(section)) - \Pi_{course\_id} (\sigma_{semester="Spring" \wedge year=2018}(section))$$

<i>course_id</i>
CS-347
PHY-101



# Atribuição

- Às vezes é conveniente escrever uma expressão de álgebra relacional atribuindo partes dela a variáveis temporárias.
- A operação de atribuição é denotada por  $\leftarrow$  e funciona como atribuição em uma linguagem de programação.
- Exemplo: Encontre todos os instrutores no departamento de Física e Música.

$Physics \leftarrow \sigma_{dept\_name="Physics"}(instructor)$

$Music \leftarrow \sigma_{dept\_name="Music"}(instructor)$

$Physics \cup Music$

- Com a operação de atribuição, uma consulta pode ser escrita como um programa sequencial composto por uma série de atribuições seguidas por uma expressão cujo valor é apresentado como resultado da consulta.

# Renomeação

- Os resultados das expressões em álgebra relacional não têm um nome que possamos usar para nos referir-mos a elas. O operador renomeado,  $\rho$ , é fornecido para o efeito

- A expressão:

$$\rho_x(E)$$

devolve o resultado da expressão E sob o nome x

- Outra forma da operação de renomeação:

$$\rho_{x(A1,A2, \dots A_n)}(E)$$

# Consultas equivalentes

- Há mais de uma maneira de escrever uma consulta em álgebra relacional.
- Exemplo: Encontre informação sobre cursos lecionados por instrutores no departamento de Física com salário superior a 90.000
- Consulta 1

$\sigma_{dept\_name="Physics" \wedge salary > 90,000}(instructor)$

- Consulta 2

$\sigma_{dept\_name="Physics"}(\sigma_{salary > 90.000}(instructor))$

- As duas consultas não são idênticas; eles são, no entanto, equivalentes - eles dão o mesmo resultado em qualquer base de dados.

# 21053 - Fundamentos de Bases de Dados

**Professor:**

**Paulo Pombinho**

**Bom Estudo!**

