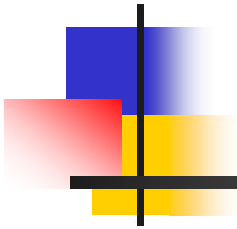


MODELO RELACIONAL



Universidade Federal do Maranhão - UFMA

Departamento de Informática

Projeto de Banco de Dados

Prof^a.MSc Simara Rocha

simararocha@gmail.com/simara@deinf.ufma.br

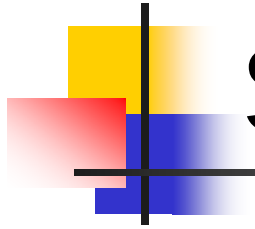
www.deinf.ufma.br/~simara

Referências: Elmasri, R. and Navathe, S.B. Sistemas de Bancos de Dados. Editora Addison-Wesley, 6ª edição, 2001.

Date, C.J. Introdução a Sistemas de Bancos de Dados. Editora Campus, 8ª edição, 2004.

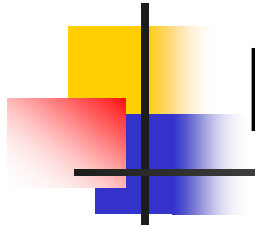
Korth, H.F. e Silberschatz, A. Sistemas de Bancos de Dados. Makron Books, 5ª edição, 2006.

Notas de Aula do Prof. Msc. Tiago Eugenio de Melo



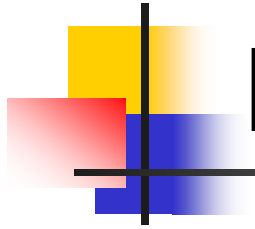
Sumário

- Histórico
- Conceitos
- Características das relações
- Restrições
- Violação das restrições
- Conclusão



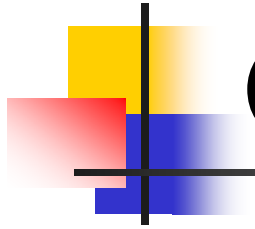
Histórico

- Foi introduzido por Tedd Codd, da IBM Research, em 1970
- Atraiu a atenção em virtude de sua simplicidade e base matemática
- Tornou-se um padrão de fato para aplicações comerciais.



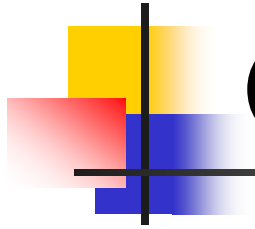
Histórico

- É um modelo formal, baseado na teoria matemática das relações
- As primeiras implementações comerciais tornaram-se disponíveis no início dos anos 80, como o SGBD Oracle e o SQL/DS (IBM).



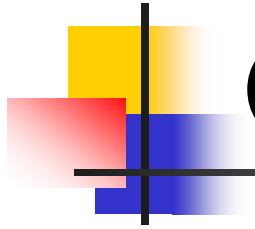
Conceitos

- O Modelo Relacional é simples e sua estrutura uniforme é baseada em conceitos da Teoria dos Conjuntos.
- A simplicidade do modelo relacional faz com que a representação do mundo real, através de seus conceitos, seja, de certa forma, ineficiente
 - o que ocasiona perdas semânticas consideráveis



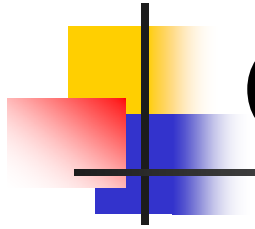
Conceitos

- O MER, ao contrário, utiliza conceitos que permitem a representação mais fiel dos objetos do mundo real e dos relacionamentos entre eles
- O Modelo Relacional tem sido implementado nos vários SGBDs tendo como LDD/LMD a linguagem SQL.



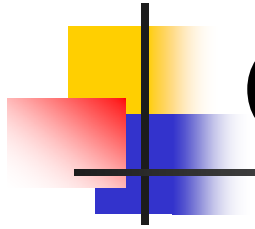
Conceitos

- O MER é hoje a ferramenta mais usada em projetos de banco de dados.
 - dizemos que o MER é um modelo do nível conceitual, pois possui um forte poder semântico, capaz de capturar conceitos do mundo real com um mínimo de perdas semânticas, facilitando o seu entendimento.



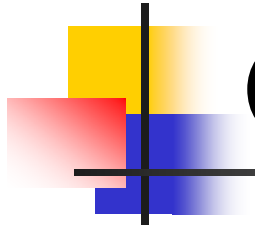
Conceitos

- O modelo relacional é, por outro lado, um modelo do nível lógico
 - porque é utilizado para representação em computador de conceitos do mundo real.
- Representa dados e relacionamentos por um conjunto de tabelas
- O modelo relacional utiliza o conceito de relações ou tabelas no lugar de arquivos.



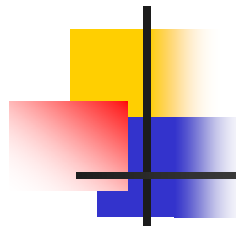
Conceitos

- Objetivos
 - Independência de dados
 - ordem
 - indexação
 - caminhos de acesso
 - reduzir inconsistências
 - regras de projetos (normalização)

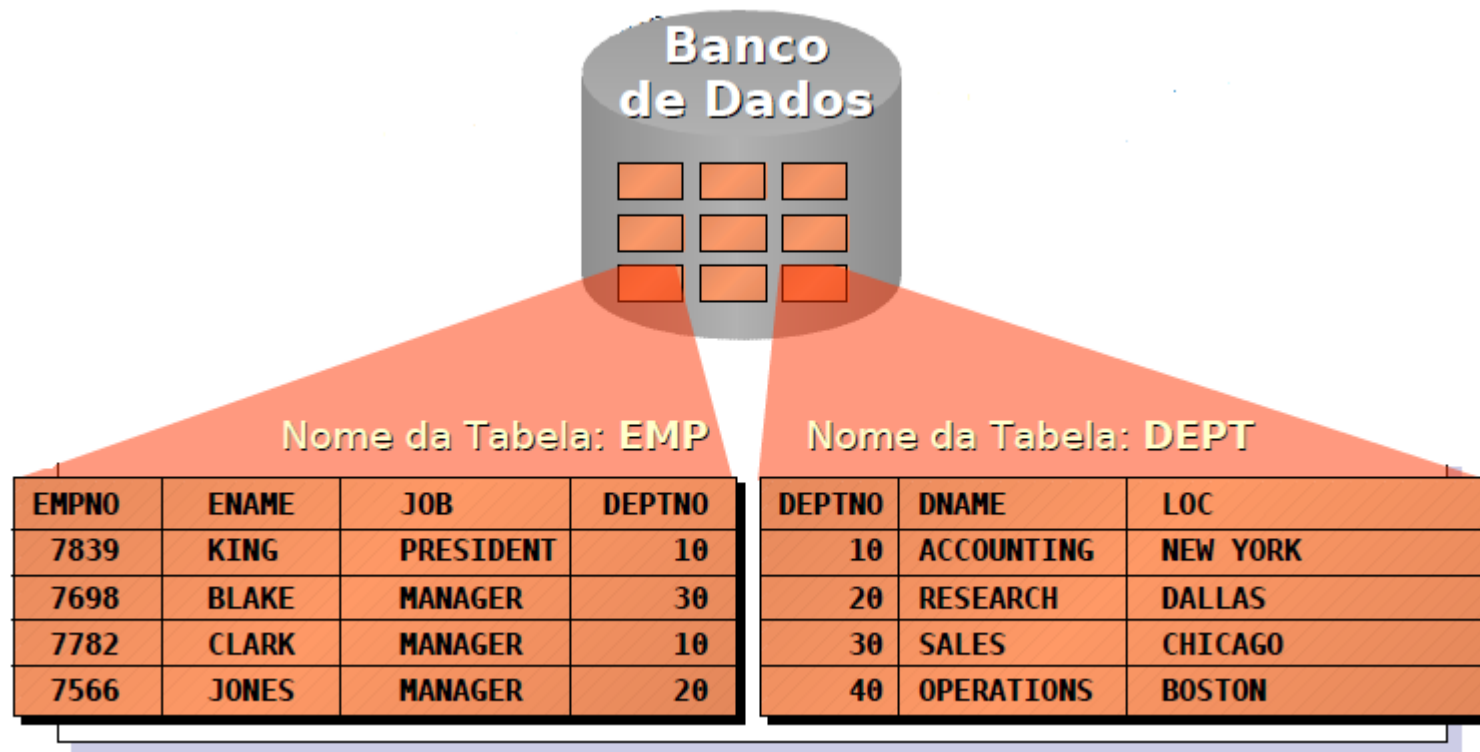


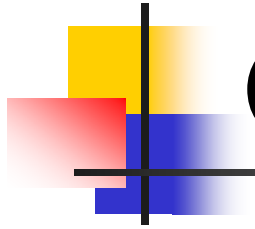
Conceitos

- Um banco de dados relacional é um conjunto de relações ou tabelas bidimensionais, gerenciados por operações relacionais e regidos por restrições de integridade de dados
 - Pode ser acessado e modificado executando instruções SQL (Structured Query Language)
 - Usa um conjunto de operadores



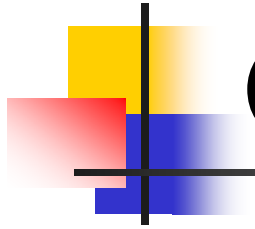
Conceitos





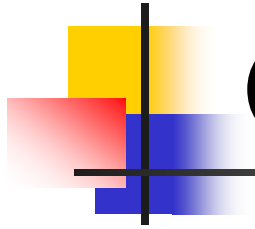
Conceitos

- Cada linha da tabela é representada por uma coleção de dados(valores) relacionados
- Na terminologia do modelo relacional formal, cada linha é uma tupla
 - não existe ordenamento entre as tuplas
 - não existem linhas duplicadas
- O nome de uma coluna é chamado de atributo
- E uma tabela é chamada de relação



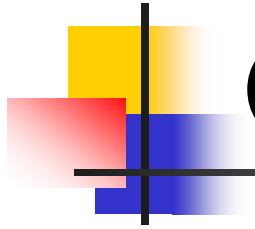
Conceitos

- Cada tabela terá um nome, que será único, e um conjunto de atributos com seus respectivos nomes e domínios.
- o relacionamento das colunas com os domínios estabelece-se a princípio dando os nomes dos domínios às colunas.
- Todos os valores de uma coluna são do mesmo tipo de dados.



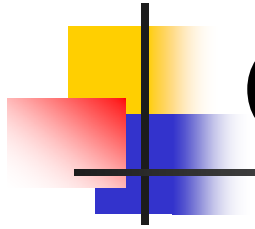
Conceitos

- Os tipos de dados de cada coluna são chamados domínios
- Um domínio é um conjunto de valores atômicos
 - Atômico: cada valor é indivisível
 - Ex.: Fone: conjunto de 10, dígitos. CPF: conjunto de 11 dígitos
- Para um domínio é dado um nome, tipo de dado e um formato



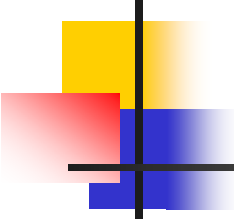
Conceitos

- Um esquema de relação é feito de um nome de relação (R) e uma lista de atributos (A_1, A_2, \dots, A_n).
 - Cada atributo A_i é o nome desempenhado por algum domínio D no esquema R.
- O grau de uma relação é o número de atributos (n) de seu esquema de relação
 - Ex.: Seja o esquema Empregado (matrícula, nome, fone, idade, cpf) \Rightarrow grau = 5

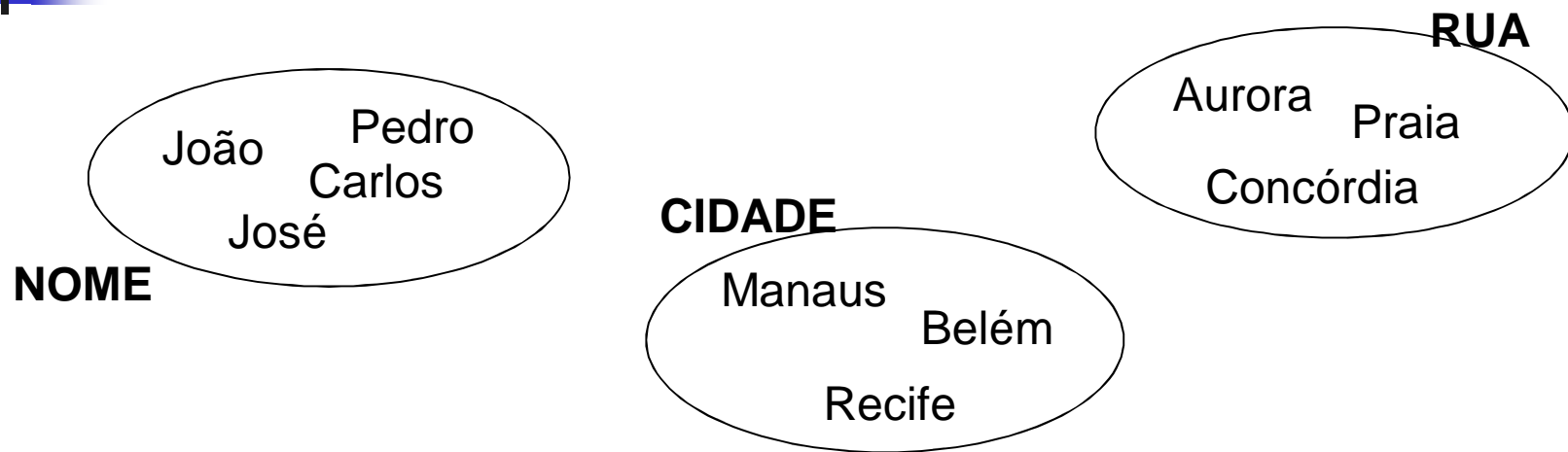


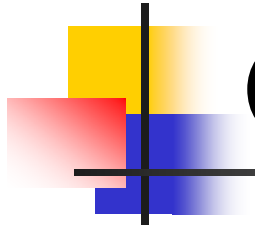
Conceitos

- Uma relação pode ser definida matematicamente como um subconjunto do produto cartesiano dos domínios que definem o esquema R
 - Dados os conjuntos C_1, C_2, \dots, C_n , uma relação r é um subconjunto de $C_1 \times C_2 \times \dots \times C_n$
 - Portanto, uma relação é um conjunto de n-tuplas (c_1, c_2, \dots, c_n) onde $c_i \in C_i$



Conceitos - Exemplo


$$r_1 = \{(\text{João}, \text{Aurora}, \text{Manaus}), (\text{Pedro}, \text{Praia}, \text{Recife}),$$
$$(\text{José}, \text{Concórdia}, \text{Belém}), (\text{Carlos}, \text{Praia}, \text{Manaus})\}$$
$$r_2 = \{(\text{João}, \text{Manaus}, \text{Aurora}), (\text{Pedro}, \text{Recife}, \text{Praia}),$$
$$(\text{José}, \text{Belém}, \text{Concórdia}), (\text{Carlos}, \text{Manaus}, \text{Praia})\}$$



Conceitos - Exemplo

$r1 = \{(Jo\tilde{a}o, Aurora, Manaus), (Pedro, Praia, Recife), (Jos\acute{e}, Conc\acute{o}rdia, Bel\acute{e}m), (Carlos, Praia, Manaus)\}$

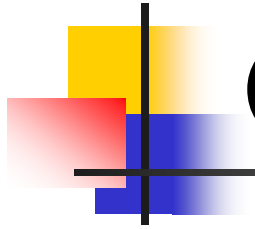
Nome	Rua	Cidade
João	Aurora	Manaus
Pedro	Praia	Recife
José	Concórdia	Belém
Carlos	Praia	Manaus

→ **Nomes dos Domínios**

→ **Dados Relacionados**

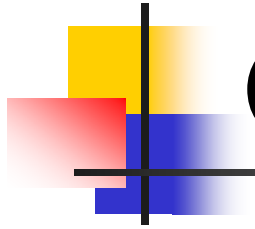
Considerações:

1. Neste caso, os nomes dos atributos são iguais ao nome do conjunto
2. Os tipos de dados de cada coluna são chamados de domínios
 $dom(nome) = dom(rua) = dom(cidade) = string$



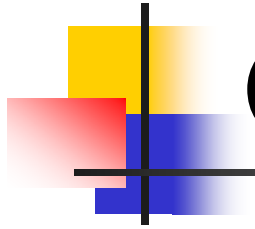
Conceitos

- Para cada relação define-se, usando a DDL (linguagem de definição de dados)
 - nome único
 - atributos
 - restrições de integridade (chave primária, chave estrangeira, valores nulos, etc)



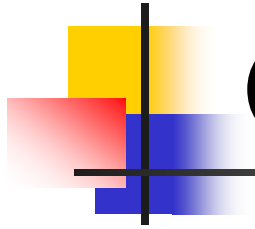
Características das Relações

- Ordenação de tuplas em uma relação
 - Relação definida como um conjunto de tuplas
 - Elementos não possuem ordem entre eles
 - Ou seja, a ordem das tuplas e dos atributos não tem importância.



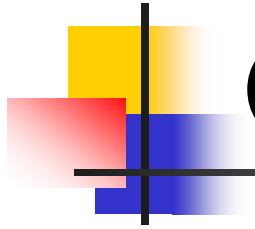
Características das Relações

- Ordem dos valores dentro de uma tupla
 - A ordem dos atributos e seus valores não é tão importante
 - Desde que a correspondência entre atributos e valores seja mantida
- Definição alternativa de uma relação
 - Tupla pode ser considerada um conjunto de pares ($\langle \text{atributo} \rangle, \langle \text{valor} \rangle$)
 - Cada par dá o valor do mapeamento a partir de um atributo A_i para um valor v_i de $\text{dom}(A_i)$



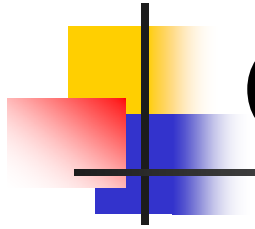
Características das Relações

- Valores nas tuplas
 - Cada valor em uma tupla é um valor atômico
 - Modelo relacional plano
 - Atributos multivalorados e compostos não são permitidos
 - Pressuposto da primeira forma normal



Características das Relações

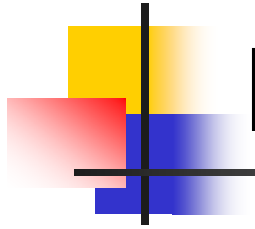
- Valores nas tuplas(cont.)
 - Atributos multivalorados
 - Precisam ser representados por relações separadas
 - Atributos compostos
 - Representados apenas por seus atributos de componentes simples no modelo relacional básico



Características das Relações

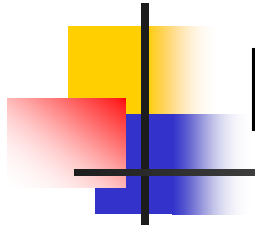
- Valores NULL

- Representam os valores de atributos que podem ser desconhecidos ou não se aplicam a uma tupla
- Significados para valores NULL
 - valor desconhecido
 - valor existe mas não está disponível
 - atributo não se aplica a esta tupla (também conhecido como valor indefinido)



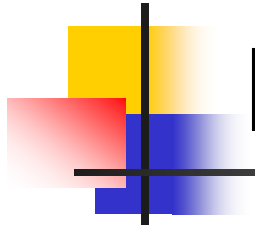
Restrições

- Restrições sobre os valores reais em um estado do banco de dados
- Derivam de regras no minimundo que o banco de dados representa



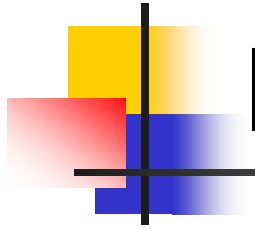
Restrições - Categorias

- Restrições inerentes baseadas no modelo ou restrições implícitas
 - Inerente ao modelo de dados
- Restrições baseadas em esquemas ou restrições explícitas
 - Podem ser expressas diretamente nos esquemas do modelo de dados



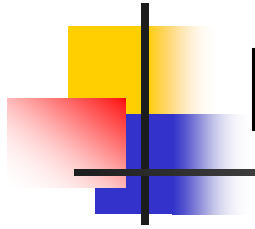
Restrições - Categorias

- Restrições baseadas na aplicação ou restrições semânticas ou regras de negócios
 - Não podem ser diretamente expressas nos esquemas
 - Expressas e impostas pelos programas de aplicação



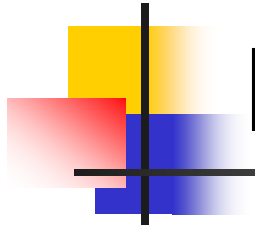
Restrições de Domínio

- Especifica que o valor de cada atributo A deve ser um valor atômico do domínio $\text{dom}(A)$
- Normalmente incluem:
 - Dados numéricos: padrão para inteiros e números reais
 - Caracteres (ou cadeias de tamanhos fixos e variáveis)
 - Booleanos
 - Data, hora: marcador de tempo
 - etc.



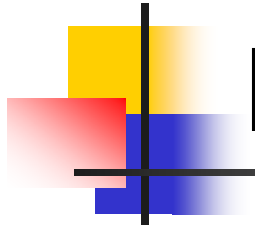
Restrições de Chave

- Duas tuplas não podem ter a mesma combinação de valores para todos os seus atributos.
- Por definição, todos os elementos de um conjunto são distintos.
 - Superchave
 - Chave Candidata
 - Chave Primária



Restrições de Chave

- Uma chave satisfaz duas propriedades:
 - Duas tuplas distintas em qualquer estado da relação não podem ter valores idênticos para (todos) os atributos na chave.
 - Superchave mínima
 - não se pode remover nenhum atributo e ainda manter uma restrição de exclusividade na condição



Restrições de Nulo

- Especifica que o valor **nulo** pode ou não ser permitido
- Valor nulo
 - a inserção de tuplas incompletas pode introduzir valores nulos na base de dados
 - evitar, sempre que possível

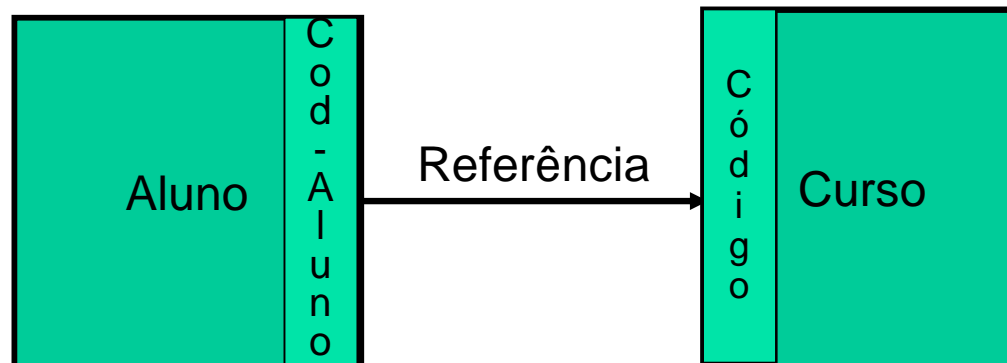


Restrições de Integridade de Entidade

- Nenhum valor de chave primária pode ser nulo
 - isso porque a chave primária é usada para identificar cada tupla da relação
 - é especificada sobre relações individuais

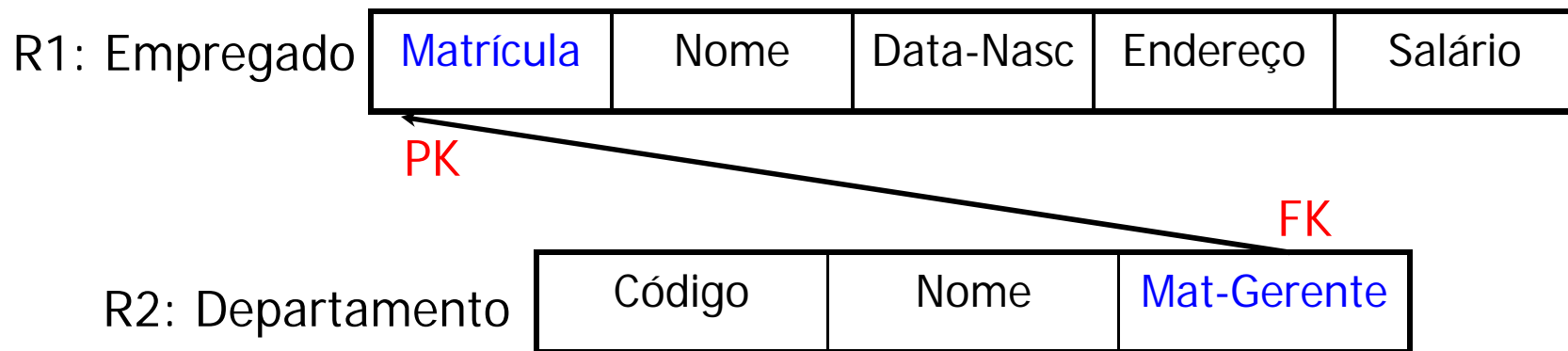
Restrições de Integridade Referencial

- Especificada entre duas relações
- Mantém a consistência entre as tuplas nas duas relações
- Informalmente, a restrição de integridade referencial diz que uma tupla da relação (R_1), que refere-se a outra relação (R_2), deve referir-se a uma tupla existente em R_2



Restrições de Integridade Referencial

- **Chave estrangeira:** duas relações R_1 e R_2 , de tal forma que, se um conjunto de atributos FK na relação R_1 que referencia a relação R_2 se esta satisfaz as duas regras:
 - (i) Os atributos de FK tem o mesmo domínio que os atributos da chave primária PK de R_2 , onde os atributos FK são ditos que se referem a relação R_2
 - (ii) Um valor de FK em uma tupla t_1 de R_1 , ou ocorre em uma tupla t_2 de R_2 ou tem valor nulo





Restrições de Integridade Referencial

- Tipicamente vem dos relacionamentos entre entidades
- É possível mostrar diagramaticamente as restrições de integridade referencial através de uma seta
 - liga cada chave estrangeira a respectiva chave primária da relação referenciada
- Uma chave estrangeira pode referenciar-se a sua própria relação (autorelacionamento).
 - Ex.: `Empregado(matrícula, nome, salário, matr_supervisor)`



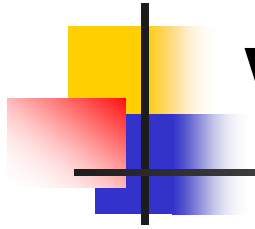
Restrições de Integridade Semântica

- As restrições de integridade referencial não incluem uma grande classe de restrições mais gerais conhecidas como restrições de integridade semântica.
 - Salário de um empregado não pode exceder o salário de seu supervisor
 - Número máximo de horas que um empregado pode trabalhar em todos os projetos por semana não pode ultrapassar 40 horas



Violação de Restrições

- Operação de Inserção
 - Restrição de domínio:
 - valor não correspondente ao domínio especificado
 - Restrição de chave:
 - se um valor de chave já existe
 - Integridade de entidade:
 - se a chave primária é nula
 - Integridade referencial:
 - se o valor da chave estrangeira refere-se a uma tupla que não existe na relação referenciada



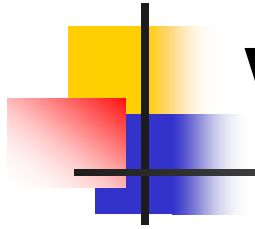
Violação de Restrições

- Operação de Exclusão
 - Pode violar somente integridade referencial
 - Ocorre quando a tupla sendo excluída é referenciada como chave estrangeira de outras tuplas no banco de dados



Violação de Restrições

- Operação de Exclusão (cont.)
 - Três opções estão disponíveis:
 - Rejeitar a exclusão
 - Propagar a exclusão
 - Modificar o valor do atributo sendo referenciado, seja colocando o valor null ou colocar um outro valor válido.
 - Se o atributo referenciado fizer parte da chave primária o atributo não poderá assumir valor nulo
 - Senão será violada a restrição de integridade de entidade



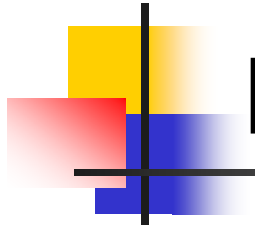
Violação de Restrições

- Operação de Atualização
 - Usada para alterar os valores de um ou mais atributos em uma tupla
 - Atributos que não são chaves primárias ou estrangeiras, não tem problemas, exceto a checagem de tipos e domínios



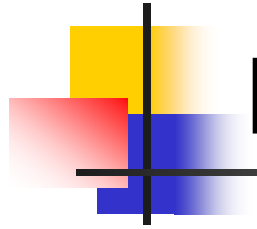
Violação de Restrições

- Operação de Atualização (cont.)
 - Modificar uma chave primária é a mesma coisa que apagar uma tupla e inserir outra em seu lugar.
 - As mesmas regras de inserção e exclusão se aplica aqui
 - Se uma chave estrangeira é modificada, o SGBD tem que garantir que o novo valor refere-se a uma tupla existente na relação referenciada



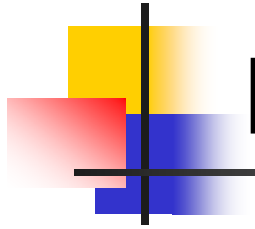
Linguagens de Consulta

- Linguagens onde o usuário pede informação do banco de dados
- Categorias de linguagens:
 - Procedurais: o usuário descreve o algoritmo de acesso aos dados através de uma seqüência de instruções (COMO)
 - Não-procedurais: O usuário descreve a informação que deseja obter sem descrever como obtê-la (O QUÊ)



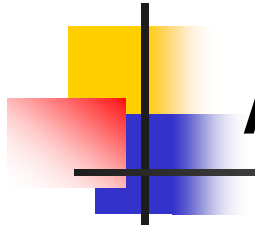
Linguagens de Consulta

- Linguagens "puras"
 - Álgebra Relacional
 - Cálculo Relacional
- Formam a base para as linguagens mais acessíveis



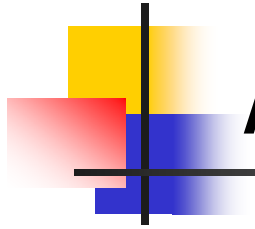
Linguagens de Consulta

- A álgebra relacional apresenta o conjunto mínimo de OPERADORES RELACIONAIS que podem ser combinados para extrair da base de dados, praticamente, todas as informações ali armazenadas (dados e seus relacionamentos)
- O cálculo estende (e completa) a potencialidade da álgebra relacional com a introdução dos quantificadores universal (\forall) e existencial (\exists)



Álgebra Relacional

- Linguagem Procedural
- Os operadores pegam duas ou mais relações como entrada e resulta uma nova relação como saída
- Variantes da Álgebra Relacional são usadas internamente pelos SGBDs relacionais durante a otimização de consultas
- Algumas operações impõem restrições nas relações de entrada



Álgebra Relacional

- Operações
 - Básicas:
 - oriundas da teoria de conjuntos: produto cartesiano, união e diferença
 - específicas para relações: seleção, projeção e renomeção
 - Adicionais
 - oriundas da teoria de conjuntos: interseção
 - específicas para relações: divisão e junção
- Notas:
 - as operações adicionais ajudam a formular certas consultas que seriam muito complexas de exprimir usando apenas as operações básicas



Álgebra Relacional

- Operações oriundas da teoria dos conjuntos

Operação	Notação	Grau do resultado	Cardinalidade do resultado	Restr.
união	$r \cup s$	$gr(r)=gr(s)$	$\leq n(r) + n(s)$	1, 2
diferença	$r - s$	$gr(r)=gr(s)$	$\leq n(r)$	1, 2
prod. cart.	$r \times s$	$gr(r)+gr(s)$	$n(r) \times n(s)$	--
interseção	$r \cap s$	$gr(r)=gr(s)$	$\leq \min(n(r), n(s))$	1, 2

- Restrições
 1. As relações r e s devem ter o mesmo grau (ou aridade)
 2. Os atributos correspondentes devem ser compatíveis



Operação de Seleção

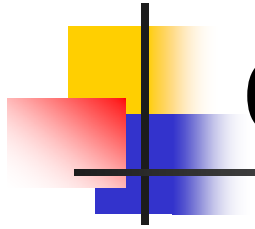
- Seleciona um subconjunto de tuplas de uma relação, de acordo com uma condição
- Notação: $\sigma_p(r)$ - tuplas de r que satisfazem o predicado de seleção P
- Definida como:

$\sigma_p(r) = \{t \mid t \in r \textbf{ and } p(t)\}$, onde P é uma fórmula em cálculo proposicional, com termos na forma:

<atributo> <op> <atributo> ou <constante>

<op> = {=, >, <, ≤, ≥, ≠}

- No predicado podemos ter as cláusulas conectadas por: \wedge (**and**), \vee (**or**), \neg (**not**)



Operação de Seleção - Exemplo

- Considere o esquema esquema relacional de uma empresa:
 - Empregado(matrícula, nomeE, endereço, sexo, salário, supervisor, depto)
 - Departamento(codDepto, nomeD, matrGerente)
 - DepLocalizações(codDepto, Localização)
 - Alocação(matrEmp, codProj, numHoras)
 - Projeto(codProj, nome, localização, deptoControla)
 - Dependentes(matrEmp, nomeDep, sexo, dataNasc, parentesco)



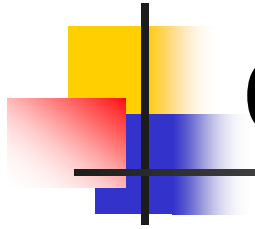
Operação de Seleção - Exemplo

1. Selecionar as tuplas dos empregados que trabalham no departamento de número 5
 - $\sigma_{\text{DEPTO} = 5}(\text{Empregado})$
2. Selecionar as tuplas dos empregados que recebem mais de R\$ 3mil
 - $\sigma_{\text{salário} > 3000}(\text{Empregado})$



Operação de Seleção - Exemplo

1. Selecionar as tuplas dos empregados que trabalham no departamento 4 e recebem acima de R\$ 5mil, ou trabalham no departamento 5 e recebem menos R\$ 500,00
 - $\sigma_{(\text{depto} = 4 \text{ AND } \text{salário} > 5000) \text{ OR } (\text{depto} = 5 \text{ AND } \text{salário} < 500)}(\text{Empregado})$



Operação de Seleção

- Observações
 - O operador de seleção é unário
 - O grau da relação resultante é o mesmo da relação original
 - O número de tuplas da relação resultante é menor ou igual ao número de tuplas da relação original.



Operação de Projeção

- Notação: $\Pi_{A_1, A_2, \dots, A_k}(r)$ - projeção das tuplas de r na lista de atributos A_1, \dots, A_k
- O resultado é definido como a relação de k colunas obtidas por apagar as colunas que não estão listadas
- Se a lista de atributos inclui apenas atributos não-chave, tuplas duplicadas poderão aparecer no resultado, porém, a operação de projeção elimina esta duplicação.



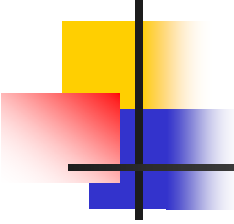
Operação de Projeção

- Exemplo:
 - $\Pi_{\langle \text{sexo}, \text{salário} \rangle}(\text{Empregado})$
- O número de tuplas da relação resultante será menor ou igual ao da relação original



Combinação de operações (seleção e projeção)

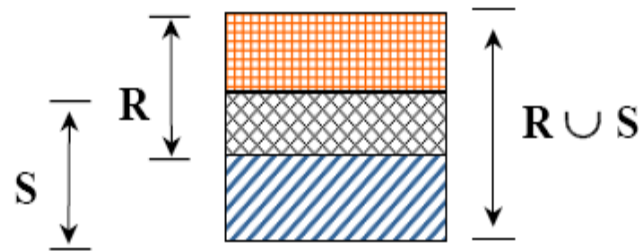
- Encontrar o nome e o salário de todos os empregados que trabalham no Departamento 5
 - $\pi_{\text{Nome, Salário}} (\sigma_{\text{DEPTO} = 5} (\text{Empregado}))$
- Como alternativamente a representação anterior, podemos usar uma notação que usa uma sequência dos resultados dando nome as relações intermediárias:
 - $\text{EmpDepto5} \leftarrow \sigma_{\text{Depto} = 5} (\text{Empregado})$
 $\text{Resultado} \leftarrow \pi_{\text{Nome, Salário}} (\text{EmpDepto5})$

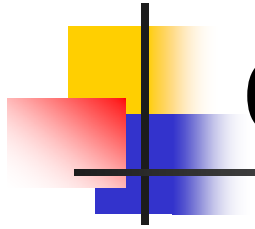


Operação de União

- Notação: $r \cup s$
- Definida como:

$$r \cup s = \{t \mid t \in r \textbf{ or } t \in s\}$$





Operação de União

- Para $r \cup s$ ser válida:
 1. r, s deve ter a mesma aridade (mesmo número de atributos)
 2. Os domínios dos atributos devem ser compatíveis (ex., segunda coluna de r possui o mesmo tipo que a segunda coluna de s)
- As tuplas repetidas são eliminadas



Operação de União - Exemplo

Relações r , s :

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cup s$:

$r \cup s$

A	B
α	1
α	2
β	1
β	3



Operação de União - Exemplo

- Obtenha a matrícula dos empregados que trabalham no departamento 5 ou supervisionam empregados que trabalham no departamento 5.
 - $\text{EmpDepto5} \leftarrow \sigma_{\text{depto} = 5} (\text{Empregados})$
 - $\text{Resultado1} \leftarrow \pi_{\text{matricula}} (\text{EmpDepto5})$
 - $\text{Resultado2} \leftarrow \pi_{\text{supervisor}} (\text{EmpDepto5})$
 - $\text{Resultado} \leftarrow \text{Resultado1} \cup \text{Resultado2}$
 - Ou
 - $\pi_{\text{matricula}}(\sigma_{\text{depto} = 5}(\text{Empregado})) \cup \pi_{\text{supervisor}}(\sigma_{\text{depto} = 5}(\text{Empregado}))$



Operação de Diferença

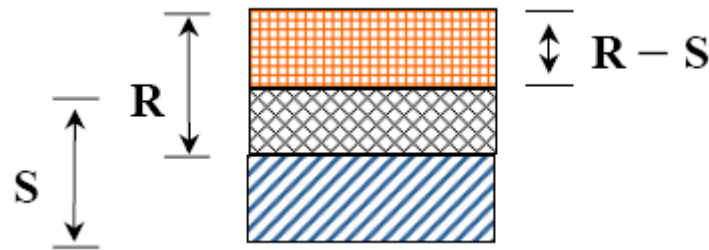
- Notação: $r - s$
- Definida como:

$$r - s = \{t \mid t \in R \textbf{ and } t \notin s\}$$

- A diferença entre duas relações, $r - s$, é o conjunto de tuplas que está em r mas não está em s .

Operação de Diferença

- Diferença deve ser feita entre relações compatíveis
 - r e s devem ter a mesma aridade
 - domínios de atributos de r e s devem ser compatíveis





Operação de Diferença - Exemplo

Relações r , s :

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r - s$:

$r - s$

A	B
α	1
β	1



Operação de Diferença - Exemplo

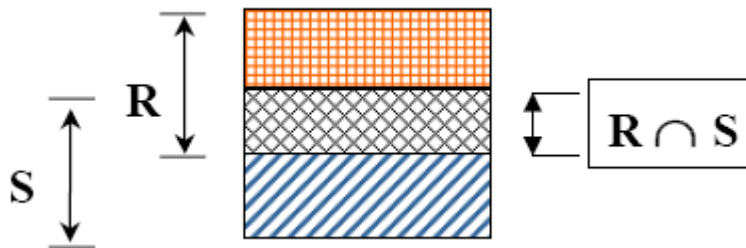
- Obtenha a matrícula dos empregados que trabalham no departamento 5 e não são supervisores
 - $\pi_{matricula}(\sigma_{depto = 5}(\text{Empregado})) - \pi_{supervisor}(\text{Empregado})$

Operação de Interseção

- Notação: $r \cap s$
- Definida como:

$$r \cap s = \{t \mid t \in r \textbf{ and } t \in s\}$$

- A interseção entre duas relações, $r \cap s$, é o conjunto de tuplas que existe tanto em r como em s .
- Para $r \cap s$ ser válida deverão ser observadas as mesmas restrições da união.





Operação de Interseção - Exemplo

Relações r, s :

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cap s$:

$r \cap s$

A	B
α	2



Operação de Interseção - Exemplo

- Obtenha a matrícula dos empregados que trabalham no departamento 5 e são supervisores
 - $\pi_{matricula}(\sigma_{depto = 5}(Empregado)) \cap \pi_{supervisor}(Empregado)$



Produto Cartesiano

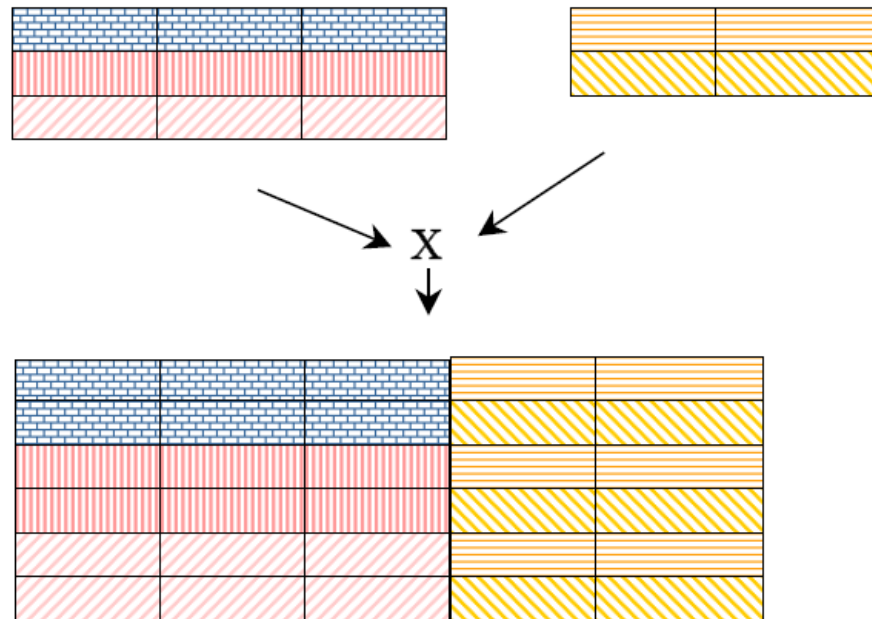
- Notação: $r \times s$
- Definida como:

$$r \times s = \{t \ q \mid t \in r \text{ and } q \in s\}$$

- Assuma que atributos de $r(R)$ e $s(S)$ são disjuntos.
Isto é,
 $R \cap S = \emptyset$.
- Se atributos de $r(R)$ e $s(S)$ não são disjuntos,
renaming deve ser usado.

Produto Cartesiano

- O produto Cartesiano de duas relações $R \times S$ combina cada tupla de R com cada tupla de S





Produto Cartesiano - Exemplo

Relações r e s :

r

A	B
α	1
β	2

s

C	D	E
α	10	a
α	13	a
β	20	b
γ	10	b

$r \times s$:

$r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	α	13	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	α	13	a
β	2	β	20	b
β	2	γ	10	b

Produto Cartesiano – (Composição de Operações)

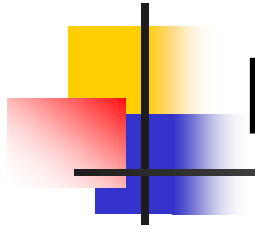
- Pode-se construir expressões combinando várias operações
- Exemplo: $\sigma_{A=C}(r \times s)$

$r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	α	13	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	α	13	a
β	2	β	20	b
β	2	γ	10	b

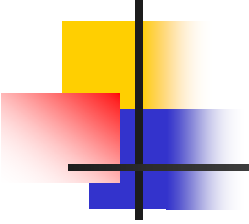
$\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
α	1	α	13	a
β	2	β	20	b



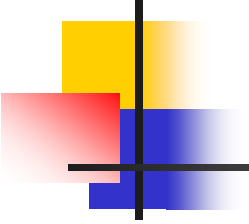
Produto Cartesiano - Exemplo

- Obtenha para cada empregado do sexo feminino, uma lista dos nomes de seus dependentes
 - $\pi_{\text{nomeE}, \text{nomeDep}} (\sigma_{\text{matr} = \text{matrEmp}} (\sigma_{\text{sexo}='F'}(\text{Empregado}) \times \text{Dependentes}))$



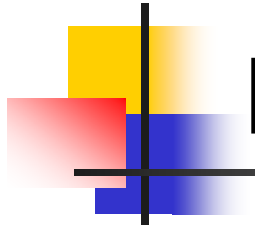
Operação Renomeação (Rename)

- Permite renomear Relações e/ou atributos para que se evite a ambiguidade na hora de compararmos atributos com mesmo nome de diferentes relações.
- Pode renomear uma relação ou os atributos da relação ou ambos. É representada pela letra grega rho(ρ)



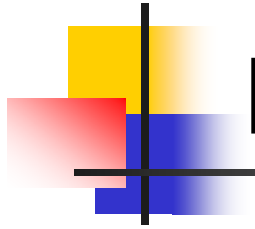
Operação Renomeação (Rename)

- Sintaxe: A operação Rename quando aplicada a uma relação R de grau N é denotada por:
 - $\rho_{S(B_1, B_2, \dots, B_n)}(R)$ ou
 - $\rho_S(R)$ ou
 - $\rho_{(B_1, B_2, \dots, B_n)}(R)$
 - Onde : (ρ denota o operador Rename, S é o nome da nova relação e B_1, \dots, B_n são os nomes dos novos atributos).



Divisão

- Sintaxe: $R \div S$
- A divisão de duas relações $R \div S$, onde $\text{atributos}(S) \subseteq \text{atributos}(R)$, resulta na relação T com $\text{atributos}(T) = \{\text{atributos}(R) - \text{atributos}(S)\}$, onde para cada tupla t que aparece no resultado, os valores de t devem aparecer em R combinado com cada tupla de S
- Quase sempre a divisão é usada quando nas consultas há frases do tipo “para todos”



Divisão - Exemplo

Exemplo:

R

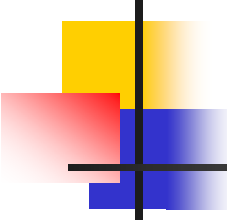
A	a1	a2	a3	a4	a1	a3	a2	a3	a4	a1	a2	a3
B	b1	b1	b1	b1	b2	b2	b3	b3	b3	b4	b4	b4

S

A	a1	a2	a3
---	----	----	----

$R \div S$

B	b1	b4
---	----	----



Divisão - Exemplo

Aluguéis

codFilme	codCli	data
f1	c1	01/01/2000
f2	c3	05/01/2000
f3	c1	01/02/2000
f4	c2	03/01/2000

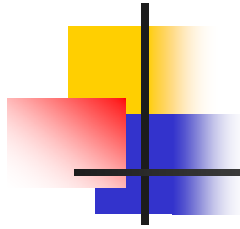
Filmes

codFilme	título	gênero
f1	Coração Valente	Aventura
f2	Se eu Fosse Você	Comédia
f3	E o Vento Levou	Romance
f4	O Máscara	Comédia

Clientes

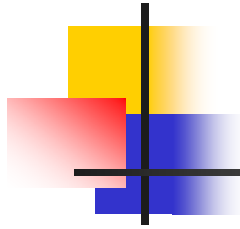
codCli	nome	idade	cidade
c1	João	31	Porto Alegre
c2	José	28	Porto Alegre
c3	Luís	25	Novo Hamburgo

- Encontre os clientes que já alugaram todos os filmes.
 - $(\pi \text{ codFilme, codCli (Aluguéis)}) \div (\pi \text{ codFilme (Filmes)})$



Junção

- A operação de junção (*join*), denotada por \bowtie , é usada para combinar tuplas relacionadas de duas relações em uma única tupla
- Permite combinar **seleções** e um **produto cartesiano** em uma só operação
- Permite recuperar **relacionamentos** entre relações



Junção

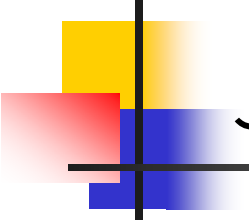
- Sejam: $R \bowtie_{\text{critério}} S$

Onde:

<R e S> são nomes de **tabelas** ou **expressões** de álgebra relacional que resulta em tabelas
<critério> é uma **expressão** booleana envolvendo literais e valores de atributos das duas tabelas

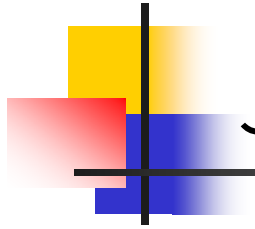
- A Junção tem como operandos duas tabela. O resultado é equivalente a executar:

$$\sigma_{\text{<critério>}} (<tabela>_1 \bowtie <tabela>_2)$$



Junção - Exemplo

- Obtenha o nome do gerente de cada departamento
 - $\pi_{\text{nomeD}, \text{nomeE}} (\text{Departamento} \bowtie_{\text{matrGerente} = \text{Matricula}} \text{Empregado})$
- Na **maior parte dos casos**, o <critério> de junção é uma expressão como mostrada no exemplo
 - Envolve apenas **igualdade** de valores de atributos de diferentes tabelas
 - Esse tipo de junção é chamada de **Equijunção (Equijoin)**

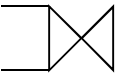
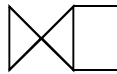
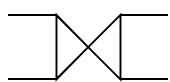


Junção - Exemplo

- Quando as colunas de junção (condição de junção) possuem os mesmos nomes, a lista de nomes das colunas não necessita ser especificadas. Nesse caso, temos uma Junção Natural (Natural join) que é um equijoin onde um dos atributos com valores repetidos é eliminado.



Junção Externa (Outer Join)

- É uma extensão da junção
- Quando a Natural join deixa de fora tuplas das relações envolvidas para as quais não há par (valores nulos)
- Existem três formas:
 - Junção externa à esquerda 
 - Junção externa à direita 
 - Junção externa total 
- Adiciona-se tuplas extras nas colunas com valores nulos



Junção Externa - Exemplo

EMP

Nome	Rua	Cidade
João	Afonso Pena	Rio de Janeiro
Saul	Teresa	Petrópolis
Hiran	Pedro Ernesto	Niterói
Marisa	Lopes Quintas	Rio de Janeiro


EMP_TI

Nome	Salário
João	5300
Saul	1600
Marisa	4000
Josefa	2500


EMP ⋈ **EMP_TI**

Nome	Rua	Cidade	Salário
João	Afonso Pena	Rio de Janeiro	5300
Saul	Teresa	Petrópolis	1600
Marisa	Lopes Quintas	Rio de Janeiro	4000


Junção Externa - Exemplo

EMP

 EMP_TI

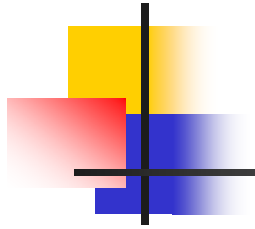
Nome	Rua	Cidade	Salário
João	Afonso Pena	Rio de Janeiro	5300
Saul	Teresa	Petrópolis	1600
Marisa	Lopes Quintas	Rio de Janeiro	4000
Hiran	Pedro Ernesto	Niterói	nulo

EMP

 EMP_TI

Nome	Rua	Cidade	Salário
João	Afonso Pena	Rio de Janeiro	5300
Saul	Teresa	Petrópolis	1600
Marisa	Lopes Quintas	Rio de Janeiro	4000
Josefa	Nulo	Nulo	2500

EMP

 EMP_TI

Nome	Rua	Cidade	Salário
João	Afonso Pena	Rio de Janeiro	5300
Saul	Teresa	Petrópolis	1600
Marisa	Lopes Quintas	Rio de Janeiro	4000
Hiran	Pedro Ernesto	Niterói	nulo
Josefa	Nulo	Nulo	2500



Funções Agregadas

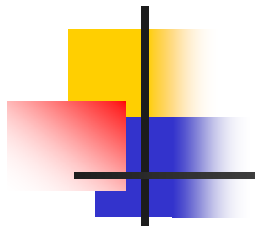
- Sum: descobrir a soma total dos salários de todos os empregados de tempo integral.
 - `sum_salário (EMP_TI)`

- Count: descobrir o nome de cidades existentes na tabela de empregados.
 - `count_Cidade (EMP)`
 - `count-distinct_Cidade (EMP)`



Funções Agregadas

- Avg: descobrir a média dos salários
 - `avg salário (EMP_TI)`
- Min: descobri o menor salário
 - `min salário (EMP_TI)`
- Max: descobrir o maior salário
 - `max salário (EMP_TI)`



Exercícios

Tabela Peças

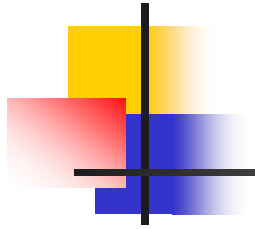
<u>Cod_Peca</u>	Nome	Cor	Peso	Cidade
P1	Eixo	Cinza	10	Recife
P2	Rolamento	Preto	16	Campinas
P3	Mancal	Verde	30	Maceió

Tabela Fornecedor

<u>Cod_Fornec</u>	Nome	Status	Cidade
F1	Silva	5	São Paulo
F2	Souza	10	Rio de Janeiro
F3	Alves	5	São Paulo
F4	Tavares	8	Rio de Janeiro

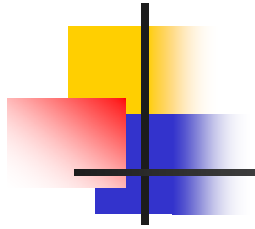
Tabela Embarque

<u>Cod_Peca</u>	<u>Cod_Forne</u>	Qtd_Embarque
P1	F1	300
P1	F2	400
P1	F3	200
P2	F1	300
P2	F4	350



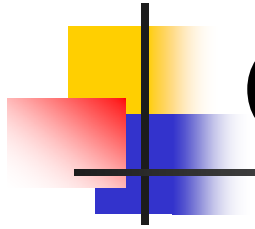
Exercícios

1. Mostre as peças com o código P1
2. Mostre os dados de todos os fornecedores que tenham status maior que 5 e que sejam do Rio de Janeiro
3. Apresente os códigos e os nomes de todas as peças
4. Mostre as cidades dos fornecedores
5. Mostre o código de fornecedor e quantidade embarcada para cada embarque da peça de código P1



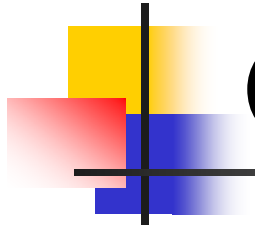
Exercícios

6. Obtenha os códigos de todos os fornecedores que tem embarques e que tem status maior que 5
7. Obtenha os nomes de todas as peças para as quais há embarques



Conclusão

- O Modelo Relacional é simples e sua estrutura uniforme é baseada em conceitos da Teoria dos Conjuntos
- Objetivos
 - Independência de dados
 - reduzir inconsistências



Conclusão

- Restrições derivam de regras do minimundo que o banco de dados representa
- Violação das restrições
 - Inserção
 - Exclusão
 - Atualização
- Álgebra Relacional Operações
 - Básicas
 - Adicionais