

Banco de Dados

Prof. Fernando Rodrigues de Almeida Júnior

UFC – Universidade Federal do Ceará

Curso: Eng. da Computação

e-mail: fernandorodrigues@sobral.ufc.br

Sumário

- ❑ Introdução aos Sistemas de Banco de Dados
- ❑ Projeto de Bancos de Dados Relacionais
- ❑ Bancos de Dados Relacionais
- ❑ Projeto Lógico de Bancos de Dados

➡ Linguagem de Manipulação e Consulta



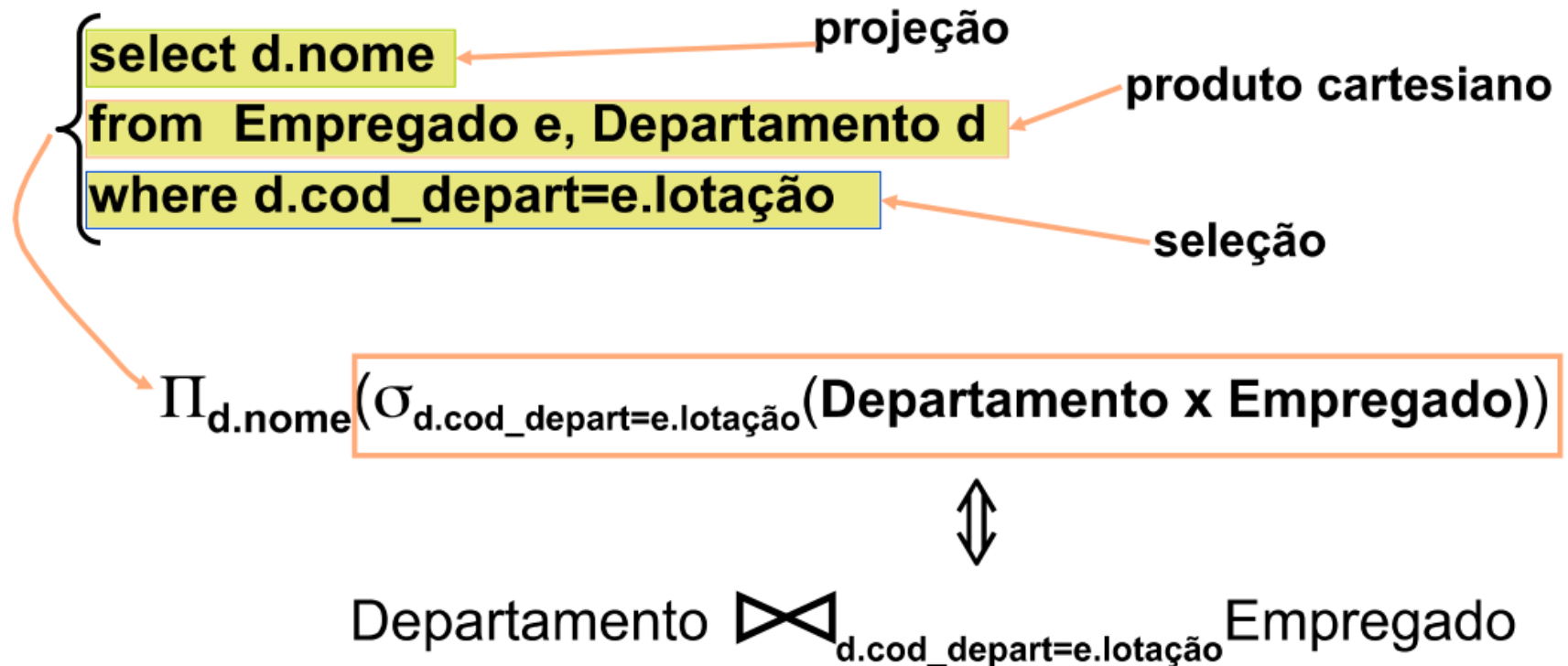
➡ Junções em SQL

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL

→ Listar nome dos departamentos dos empregados

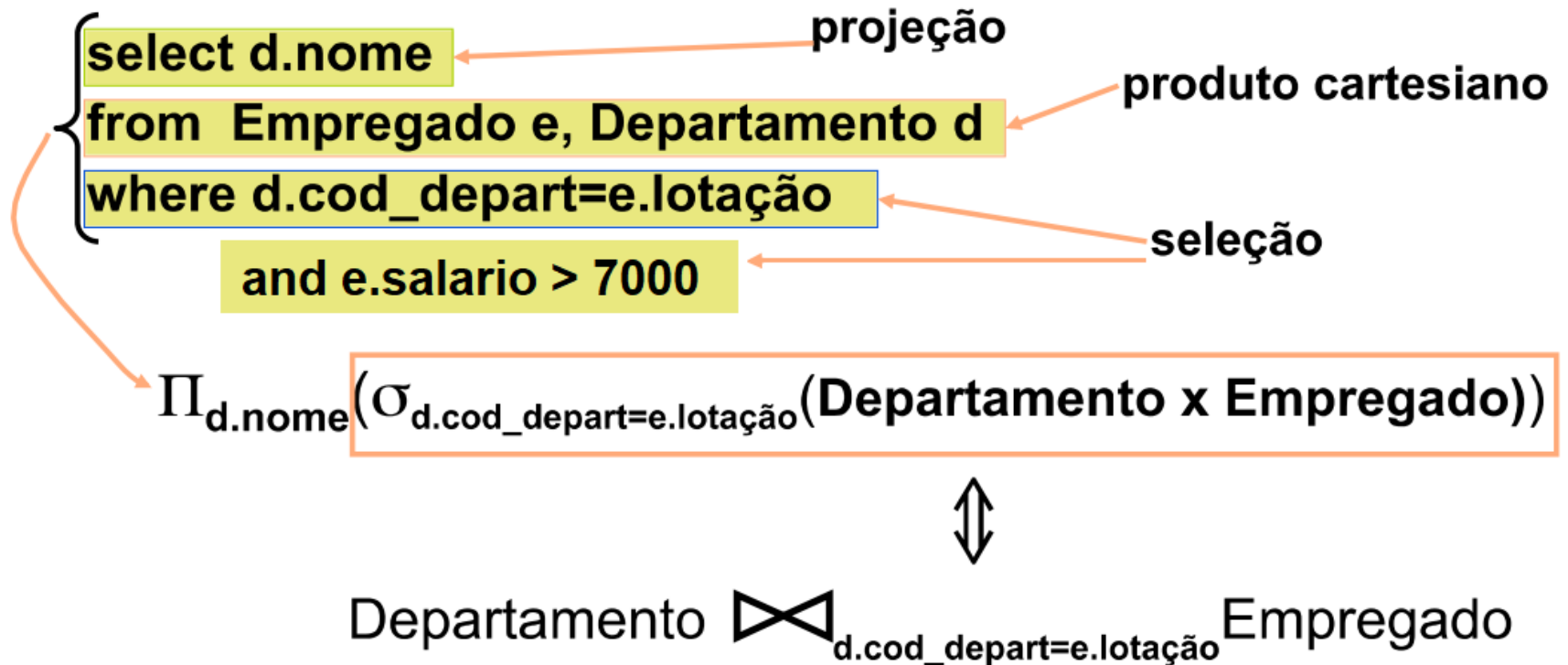


5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL

- Listar nome dos departamentos dos empregados que têm salário maior que 7000



5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

↳ Sintaxe da cláusula FROM

[FROM {<tabela_fonte>} [,...n]]

<table_fonte> ::=

nome_tabela [[AS] *qualificador*] [WITH (<table_hint> [,...n])]

| *nome_tabela* [[AS] *qualificador*] [(*column_alias* [,...n])]

| (*subquery*) [AS] *qualificador* [(*column_alias* [,...n])]

| *nome_visão* [AS] *qualificador* [(*column_alias* [,...n])]

| <tabela_fonte> <tipo_junção> <tabela_fonte> ON

<condição_junção>

<tipo_junção> ::=

[INNER | NATURAL | { { LEFT | RIGHT | FULL } [OUTER] }] JOIN

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❏ Formas de Junção em SQL (cont.)

↳ Sintaxe da cláusula FROM (cont.)

⇒ WITH (<table_hint> [...n])

⇒ Especifica estratégias (dicas) para o otimizador de consultas

➤ Índices

➤ tipo e granularidade de bloqueio (lock)

⇒ (column_alias [...n])

⇒ Especifica *alias* para colunas retornadas de

➤ uma tabela (não suportado pelo SQL Server 7.0) ou

➤ subconsulta

⇒ Um alias para cada coluna especificada na lista do select da subconsulta

⇒ join_hint

⇒ Indica para o otimizador de consulta qual o algoritmo de junção deve ser executado

➤ Nested loop join, merge join ou hash join (SQL Server 2005)

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❏ Formas de Junção em SQL (cont.)

↳ Sintaxe da cláusula FROM (cont.)

⇒ Tipos de junção

⇒ Junção theta

‣ INNER JOIN

⇒ Junção natural

‣ NATURAL JOIN

⇒ Junção externa à esquerda

‣ LEFT OUTER JOIN

⇒ Junção externa à direita

‣ RIGHT OUTER JOIN

⇒ Junção externa completa

‣ FULL OUTER JOIN

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

↳ Sintaxe da cláusula FROM (cont.)

⇒ Tipos de junção

⇒ Junção theta

‣ INNER JOIN

⇒ Junção externa à esquerda

‣ LEFT OUTER JOIN

⇒ Junção externa à direita

‣ RIGHT OUTER JOIN

⇒ Junção externa completa

‣ FULL OUTER JOIN

↳ Junção theta

⇒ Exemplo

⇒ Listar nome dos empregados com o nome do respectivo departamento

```
select e.nome, d.nome  
from Empregado e inner join Departamento d on e.lotação=d.cod_depart
```


5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

↳ Junção theta (INNER JOIN)

⇒ Exemplo

⇒ Listar nome dos empregados com o nome do respectivo departamento

```
select e.nome, d.nome  
from Empregado e inner join Departamento d on e.lotação=d.cod_depart
```

↳ Junção theta (INNER JOIN)

⇒ Exemplo

⇒ Para os empregados que têm salário maior que 700, listar nome com o nome do respectivo departamento,

```
select nome_empregado, d.nome from Departamento d inner join  
(select nome, lotação from Empregado where salário>700) e  
(nome_empregado, departamento_lotação)  
on departamento_lotação=d.cod_depart
```

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

→ Junção natural (NATURAL JOIN)

⇒ Exemplo

⇒ Para os empregados que têm salário maior que 700, listar nome com o nome do respectivo departamento

```
select nome_empregado, nome from Departamento natural join  
(select nome, lotação from Empregado where salário>700) e  
(nome_empregado,cod_depart)
```

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

→ LEFT OUTER JOIN

⇒ Calcula o resultado da junção

⇒ Adiciona ao resultado da junção

- ▶ Tuplas da relação à esquerda que não satisfazem a condição de junção

- ▶ Atribui valores nulos aos atributos não definidos para estas tuplas

⇒ Exemplo

⇒ Listar o histórico de vendas de cada vendedor, considerando o seguinte esquema

- ▶ Vendedor(matr, nome, salário)

- ▶ Vendas(matr, cod_item, qtde, pr-venda, dt-hora-venda)

- ▶ Estoque(cod_item, referência, pr_compra, pr_venda, qtde)

```
select v.nome, e.referência, d.qtde, d.qtde*d.pr_venda  
from Vendedor v left outer join (Vendas d inner join Estoque e  
on d.cod_item=e.cod_item) ve on v.matr=ve.matr
```

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

↳ RIGHT OUTER JOIN

⇒ Calcula o resultado da junção

⇒ Adiciona ao resultado da junção

‣ Tuplas da relação à direita que não satisfazem a condição de junção

‣ Atribui valores nulos aos atributos não definidos para estas tuplas

⇒ Exemplo

⇒ Para cada empregado, listar nome do empregado, nome do departamento e nome dos dependentes

```
select e.nome, d.nome, p.nome  
from Departamento d inner join (Dependente p right outer join  
Empregado e on p.matr_resp=e.matr)pe on d.cod_dep=pe.lotação
```

5. SQL

- Conjunto de Operações para Manipulação de Dados -

❑ Formas de Junção em SQL (cont.)

↳ FULL OUTER JOIN

⇒ Calcula o resultado da junção

⇒ Adiciona ao resultado da junção

‣ Tuplas das relações envolvidas na junção que não satisfazem a condição de junção

‣ Atribui valores nulos aos atributos não definidos para estas tuplas

❑ Unindo tuplas em uma tabela (Outras Junções)

↳ Junção Externa Natural (NATURAL . . . OUTER JOIN)

‣ **Adicionar tuplas extras ao resultado de uma junção natural**

↳ Produto Cartesiano (CROSS JOIN)

‣ **Fazer um produto cartesiano entre duas relações quaisquer**

Referências



- ❑ Notas de Aula – Prof. Angelo Brayner
- ❑ Manual MySQL 5.7



FIM