Junção

Junção

- Cross Join (Junção Implícita)

  - O resultado do produto cartesiano raramente é útil, pois mostra combinações de linhas que não têm relação entre si e por isso não são úteis para o usuário.

  - A sua execução causa muito I/O na base de dados

```
SELECT empregado.numeroEmpregado,
  empregado.nomeEmpregado,  empregado.numeroDepartamento,
  departamento.numeroDepartamento, departamento.nomeDepartamento
FROM empregado CROSS JOIN departamento;
```

Junção

- Junção Interna

  - A junção interna é uma operação de junção horizontal entre duas tabelas, que usa uma comparação por igualdade entre a(s) coluna(s) comum(ns).

  - Normalmente a(s) coluna(s) comum(ns) é(são) Foreign Key numa tabela e Primary Key ou Unique Key na outra.

  - A junção interna pode ser vista como um produto cartesiano filtrado, pois exige que as linhas da tabela da esquerda tenham correspondente na tabela da direita, sendo o valor da coluna comum igual.

  - A junção interna é a operação mais importante nas bases de dados relacionais, utilizada várias cláusulas

Junção

- **Cláusula ON**

  - As tabelas EMPREGADO e DEPTARTAMENTO possuem uma relação entre si, implementada através da coluna comum NUMERODEPARTAMENTO.

  - Na tabela EMPREGADOS sabemos qual o número do departamento em que o empregado trabalha.

  - Na tabela DEPARTAMENTO sabemos o número, nome e localização desse departamento.

```
SELECT empregados.numeroEmpregado,  empregado.nome,
  empregado.numeroDepartamento,  departamento.numeroDepartamento,
  departamento.nomeDepartamento,  departamento.localidade
FROM empregados
INNER JOIN departamento ON
(empregados.numeroDepartamento=departamento.numeroDepartamento);
```

Junção

- **Cláusula ON**
  - Com esta operação o usuário consegue visualizar o nome do empregado, o nome do departamento em que trabalha e a sua localização, ou seja, vê os dados de duas tabelas relacionadas como se fossem uma única
  - Esta junção pode ser interpretada como um produto cartesiano ao qual foram eliminadas as linhas que não satisfazem a condição de junção

Junção

- **Cláusula USING**
  - A cláusula USING pode ser usada em vez da cláusula ON sempre que a(s) coluna(s) usada(s) na junção tenha(m) o mesmo nome em ambas as tabelas.
  - Esta cláusula pode ser usada mesmo que existam outras colunas com o mesmo nome em ambas as tabelas.
  - No caso das tabelas EMPREGADOS e DEPARTAMENTO a junção pode ser feita com USING
  - Esta cláusula facilita a escrita do query, mas requer a validação prévia de que a(s) coluna(s) usada(s) na junção tem(êm) o mesmo nome em ambas as tabelas;
  - A cláusula USING obriga a que a(s) coluna(s) usada(s) na junção seja(m) referenciada(s) sem o nome da tabela a que pertence(m)

Junção

- **Cláusula USING**

```
SELECT empregado.numeroEmpregado,
  empregado.nome,
  numeroDepartamento,
  departamento.nomeDepartamento,
  departamento.localidade
FROM empregados
INNER JOIN departamento USING (numeroDepartamento);
```

- **Cláusula NATURAL JOIN**

  - A cláusula NATURAL JOIN pode ser usada em vez da cláusula ON ou em vez da cláusula USING sempre que:

    - A(s) coluna(s) usada(s) na junção tenha(m) o mesmo nome em ambas as tabelas;

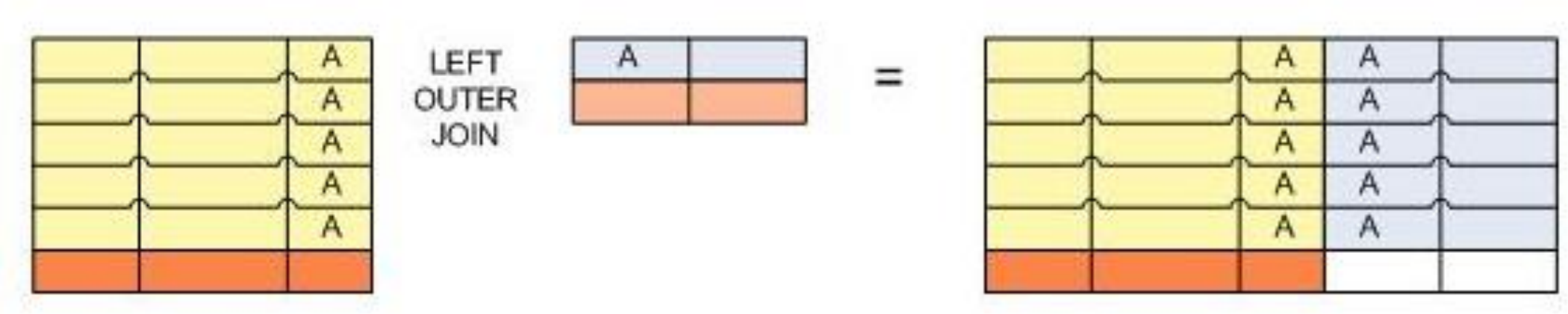
    - A(s) coluna(s) usada(s) na junção é(são) a(s) única(s) com o mesmo nome em ambas as tabelas;

```
SELECT empregado.numeroEmpregado, empregado.nome,  numeroDepartamento,
  departamento.nomeDepartamento, departamento.localidade
FROM empregados
NATURAL JOIN departamento;
```

- **Junção Externa à esquerda**

  - A junção externa à esquerda estende o resultado da junção interna, pois mostra todas as linhas que são devolvidas pela junção interna e acrescenta as linhas da tabela da esquerda que não têm correspondência na tabela da direita.

  - A junção envolve sempre duas tabelas, sendo a da esquerda a que primeiro aparece no comando.



- **Junção Externa à esquerda**

  - Os comandos apresentados a seguir mostram como fazer a junção externa à esquerda entre as tabelas EMPREGADOS e DEPARTMENTO.

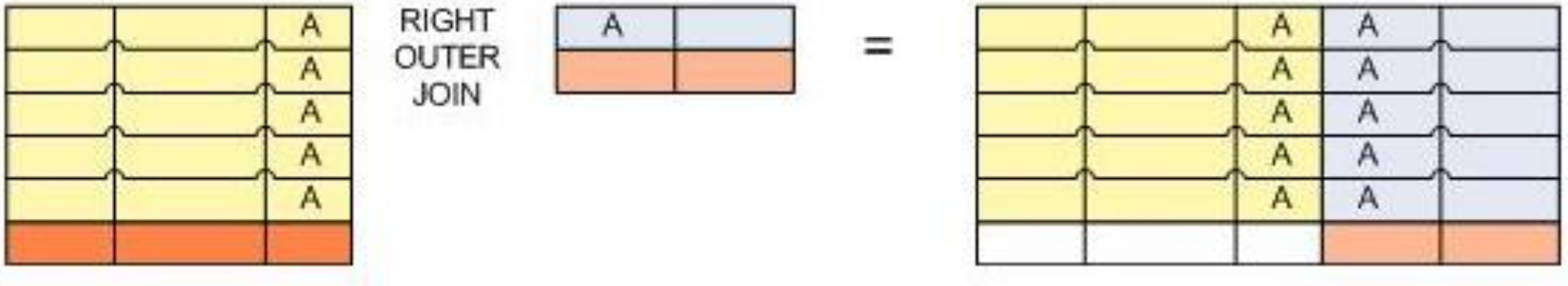
  - Esta operação vai mostrar as linhas obtidas pela junção interna, acrescida dos empregados que não têm um departamento atribuído, o que é possível porque a coluna numeroDepartamento em EMPREGADOS suporta valores nulos

```
SELECT
   e.nome,
   e.sobrenome,
   d.numeroDepartamento,
   d.nomeDepartamento
FROM empregados e
LEFT OUTER JOIN departamento d ON (e.numeroDepartamento=d.numeroDepartamento);
```

- **Junção Externa à direita**

  - A junção externa à direita funciona da mesma forma que a junção externa à esquerda, mas mostra as linhas da tabela da direita que não têm correspondência com linhas da tabela da esquerda.

  - Este resultado pode ser obtido por uma junção à esquerda trocando a ordem das colunas




```
SELECT
   e.nome, e.sobrenome, d.numeroDepartamento, d.nomeDepartamento
FROM empregados e
LEFT OUTER JOIN departamento d ON (e.numeroDepartamento=d.numeroDepartamento);
```

Junção

- Junção de uma tabela com ela própria
  - É possível relacionar uma tabela com ela própria
  - Esta operação junta linhas de uma tabela com linhas da própria tabela
  - Para isso é necessário utilizar sinônimos para os nomes das tabelas
  - Estes criam uma cópia virtual da tabela, originando assim uma relação entre duas tabelas

Junção

- Junção de uma tabela com ela própria
  - No exemplo abaixo, o campo numeroEmpregado contém o número de empregado.
  - O campo gerente contém o número de empregado que é “gerente" deste empregado. É uma relação para a própria tabela.
  - O comando abaixo permite listar todos os empregados que ganham menos que os seus “gerentes":

```
SELECT
  e.nome "Nome empregado" ,  e.salario "Salário empregado",
  g.nome "Nome do Gerente",   g.salario "Salário do Gerente"
FROM empregados e
INNER JOIN  empregados g ON (e.gerente=g.numeroEmpregado)
WHERE e.salario < g.salario;
```

Junção

- Não equi-junção entre tabelas

  - Sempre que o relacionamento entre duas tabelas é obtido por um operador que não seja = estamos numa NÃO-EQUI-JUNÇÃO.

  - Para determinar o nível de salarial de um empregado (CARGO) temos que encontrar o intervalo de valores em que o salário do empregado se enquadra, estando as categorias armazenadas na tabela SALARIOCARGO.

  - O relacionamento entre as tabelas EMPREGADOS e SALARIOCARGO é uma não-equi-junção, pois nenhuma coluna de EMPREGADOS corresponde diretamente a uma coluna de SALARIOCARGO sendo a relação obtida utilizando o operador BETWEEN.

```
select e.nome, e.funcao, s.cargo
from empregado e
inner join salarioCargo s on (e.salario between s.minSalario and s.maxSalario);
```

Exercícios

Junção

- Liste todos os empregados e respectivos nomes de departamento, por ordem do nome de departamento

```
select e.nome, e.numeroDepartamento, d.nomeDepartamento
from empregados e
inner join departamento d on
(e.numeroDepartamento=d.numeroDepartamento)
order by d.nomeDepartamento;
```

Junção

- Liste o nome, número e nome de departamento de todos os empregados

```
select e.nome, d.numeroDepartamento, d.nomeDepartamento
from empregados e
inner join  departamento d on
(e.numeroDepartamento=d.numeroDepartamento);
```

Junção

- Liste o nome, departamento e localidade do departamento dos empregados cujo salário mensal seja superior a 1500

```
select e.nome,d.nomeDepartamento,d.localidade
from empregados e
inner join  departamento d on (e.numeroDepartamento =
d.numeroDepartamento)
where salario > 1500;
```

Junção

- Produza uma lista que mostre o Cargo de cada empregado

```
select nome, funcao, salario, Cargo
from empregado
inner join salarioCargo on (salario between salarioCargo.minSalario
and salarioCargo.maxSalario);
```

Junção

- Produza uma lista dos empregados com cargo 3

```
select nome, funcao, salario, Cargo
from empregado
inner join salarioCargo on (salario between salarioCargo.minSalario
and salarioCargo.maxSalario)
where Cargo = 3;
```

Junção

- Mostre todos os empregados de Betim (Departamento.Localidade)

```
select empregados.nome, empregados.salario, departamento.localidade
from empregados
inner join departamento on
(empregados.numeroDepartamento=departamento.numeroDepartamento)
where departamento.localidade = 'BETIM';
```

Junção

- Liste nome, função, salário, cargo e nome de departamento para todos os empregados, exceto empregados de escritório (CONTADOR). Ordene por salário apresentando primeiro o mais elevado

```
select e.nome, e.funcao, e.salario, s.Cargo, d.nomeDepartamento
from empregados e
inner join salarioCargo s on (e.salario between s.minSalario and s.maxSalario)
inner join departamento d on
(e.numeroDepartamento=d.numeroDepartamento)
where e.funcao != 'CONTADOR'
order by e.salario desc;
```

Junção

- Liste o nome, função, salário anual, número de departamento, nome de departamento e cargo para os empregados que ganham mais de 36000 por ano (12 vezes o salário mais a comissão) ou que sejam empregados de escritório (CONTADOR)

```
select e.nome, e.funcao, e.salario*12+nvl(e.comissao,0),
d.numeroDepartamento, d.nomeDepartamento, s.Cargo
from empregados e
inner join salarioCargo s on (e.salario between s.minSalario and s.maxSalario)
inner join departamento d on
(e.numeroDepartamento=d.numeroDepartamento)
and (e.salario*12+nvl(e.comissao,0) > 36000
    or e.funcao = 'CONTADOR');
```

Junção

- Liste todos os departamentos que não têm empregados

```
select d.numeroDepartamento, d.nomeDepartamento
from empregados e
right outer join departamento d on (e.numeroDepartamento =
d.numeroDepartamento)
where e.nome is null;
```

Junção

- Encontre o departamento que contratou funcionários na primeira metade de 2001 e na primeira metade de 2002

```
select e.numeroDepartamento, d.nomeDepartamento
from empregados e
inner join
departamento d on (e.numeroDepartamento=d.numeroDepartamento)
where
      e.dataContratacao between to_date('2001-01-01','YYYY-MM-DD') and to_date('2001-06-30','YYYY-MM-DD')
intersect
select e.numeroDepartamento, d.nomeDepartamento
from empregados e
inner join
       departamento d on (e.numeroDepartamento=d.numeroDepartamento)
             where   dataContratacao
                          between to_date('2002-01-01','YYYY-MM-DD') and to_date('2002-06-30','YYYY-MM-DD');
```

Junção

- Encontre todos os empregados que entraram para a empresa antes do seu gerente

```
select e.nome as empregado_nome, e.dataContratacao as empregado_data,
      m.nome as gerente_nome, m.dataContratacao as gerente_data
from empregados e
inner join empregados m on (e.gerente=m.numeroEmpregado)
and e.dataContratacao > m.dataContratacao;
```