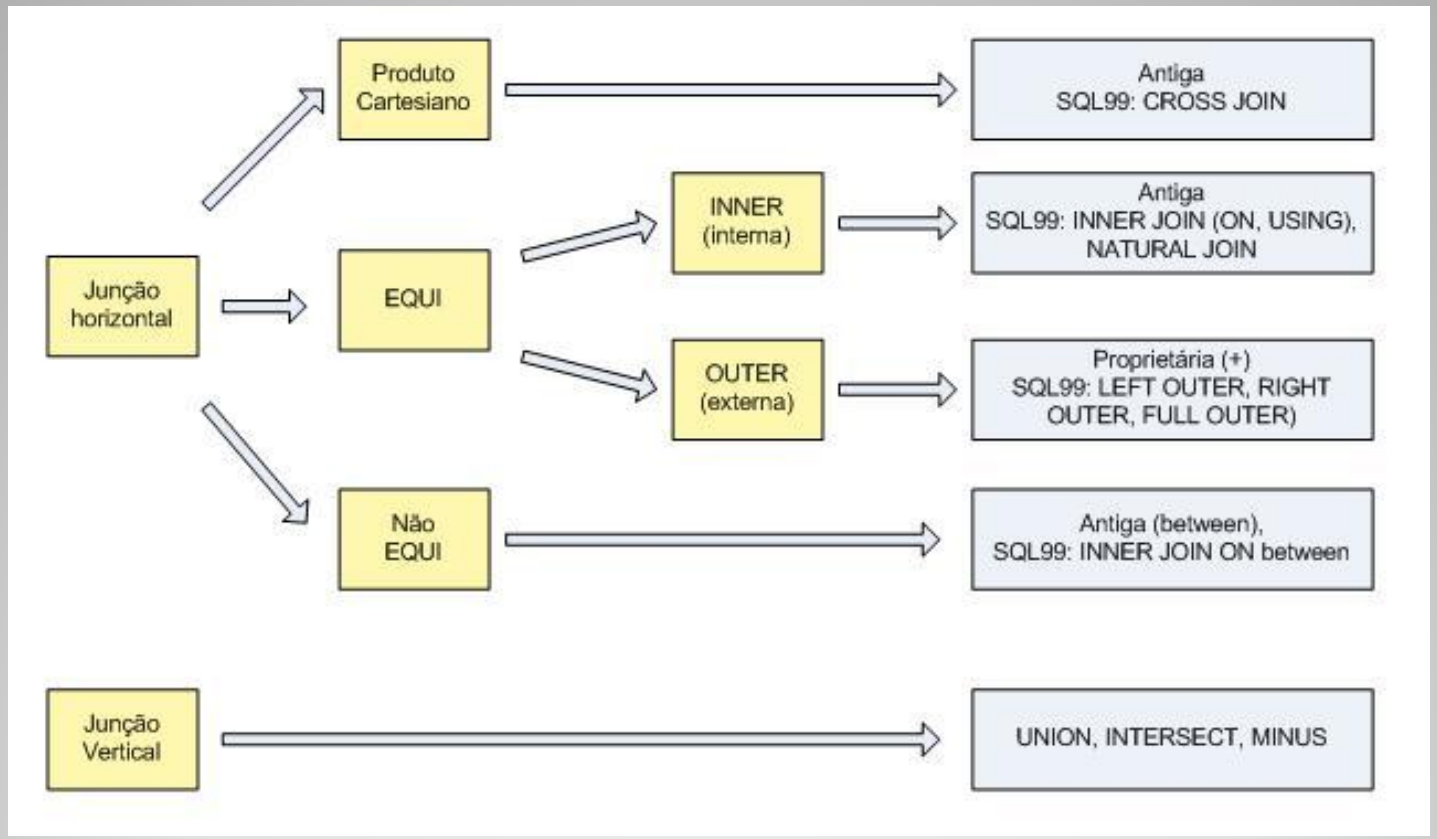
TLBD II

Continuação (04)

A figura abaixo resume as operações de junção:



Junção de Tabelas

Junção horizontal

As junções horizontais atuam sobre **linhas.**

Permite mostrar os dados que estão armazenados em diferentes tabelas como se estivessem armazenados numa única, desde que essas tabelas possuam um relacionamento entre si.

A tabela resultado é construída a partir de uma das tabelas originais, acrescentando colunas da segunda tabela, o que corresponde a um crescimento horizontal.

O **produto cartesiano** entre dois conjuntos é um terceiro conjunto constituído por todos os elementos do primeiro combinados com todos os elementos do segundo.

Os comandos abaixo geram o produto cartesiano entre as tabelas EMP e DEP:

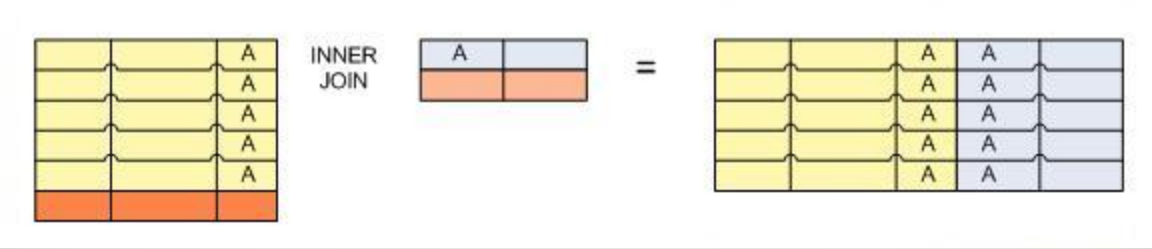
Sintaxe antiga	SQL99
SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome FROM emp, dept;	SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome FROM emp CROSS JOIN dept;

Junção horizontal – Produto cartesiano

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.

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.

O diagrama apresentado a seguir mostra como funciona a junção interna entre duas tabelas:



Junção horizontal - Junção Interna

As tabelas EMP e DEPT possuem uma relação entre si, implementada através da coluna comum DEPTNO. Na tabela EMP sabemos qual o número do departamento em que o empregado trabalha. Na tabela DEPT sabemos o número, nome e localização desse departamento. Para juntar os dois conjuntos efetuamos uma **JUNÇÃO horizontal** das duas tabelas, usando uma igualdade de valores na coluna comum, como ilustrado nos exemplos abaixo:

Sintaxe antiga	SQL99
SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome, dept.loc FROM emp, dept WHERE emp.deptno=dept.deptno;	SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome, dept.loc FROM emp INNER JOIN dept ON (emp.deptno=dept.deptno);

A maioria das operações de junção são internas (INNER) pelo que a palavra reservada INNER é facultativa;

Junção Interna - Cláusula ON

Junção Interna - Cláusula USING

A cláusula **USING** está disponível na sintaxe SQL99 e 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.

Sintaxe antiga	SQL99
SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome, dept.loc FROM emp, dept WHERE emp.deptno=dept.deptno;	SELECT emp.empno, emp.enome, deptno, dept.dnome, dept.loc FROM emp INNER JOIN dept USING (deptno);

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 Interna – NATURAL JOIN

A cláusula **NATURAL JOIN** está disponível na sintaxe SQL99 e 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;

Sintaxe antiga	SQL99
SELECT emp.empno, emp.enome, emp.deptno, dept.deptno, dept.dnome, dept.loc FROM emp, dept WHERE emp.deptno=dept.deptno;	SELECT emp.empno, emp.enome, deptno, dept.dnome, dept.loc FROM emp NATURAL JOIN dept;

A cláusula NATURAL JOIN obriga a que a(s) coluna(s) usada(s) na junção seja(m) referenciada(s) sem o nome da tabela a que pertence(m);

Tabelas:

```
CREATE TABLE locais (
LOCAL_ID NUMERIC(4) PRIMARY KEY,
ENDERECO VARCHAR(40),
CEP VARCHAR(9),
CIDADE VARCHAR(30),
ESTADO VARCHAR(2)) ENGINE=INNODB;

CREATE TABLE departamentos (
DEPT_ID NUMERIC(4) PRIMARY KEY,
DEPT_NOME VARCHAR(30),
GERENTE_ID NUMERIC(6),
LOCAL_ID NUMERIC(4)) ENGINE=INNODB;

CREATE TABLE empregados (
EMP_ID NUMERIC(6) PRIMARY KEY,
PNOME VARCHAR(20),
UNOME VARCHAR(25),
EMAIL VARCHAR(40),
FONE VARCHAR(20),
CONTRATADO DATE,
EMPREGO_ID VARCHAR(10),
SAL NUMERIC(8,2),
COMISSAO_PCT NUMERIC(2,2),
GERENTE_ID NUMERIC(6),
DEPT_ID NUMERIC(4)) ENGINE=INNODB;
```

Relacionamentos:

Trabalha em

```
ALTER TABLE empregados
ADD CONSTRAINT fk1
FOREIGN KEY (DEPT_ID)
REFERENCES departamentos (DEPT_ID);
```

É chefe de

```
ALTER TABLE empregados
ADD CONSTRAINT fk2
FOREIGN KEY (GERENTE_ID)
REFERENCES empregados (EMP_ID);
```

Está localizado em

```
ALTER TABLE departamentos
ADD CONSTRAINT fk1
FOREIGN KEY (LOCAL_ID)
REFERENCES locais (LOCAL_ID);
```

É chefe de departamento

```
ALTER TABLE departamentos
ADD CONSTRAINT fk2
FOREIGN KEY (GERENTE_ID)
REFERENCES empregados (EMP_ID);
```

Crie 3 tabelas e seus relacionamentos.

Tabelas:

empregados:

EMP_ID	PNOME	UNOME	EMAIL	FONE	CONTRATADO	EMPREGO_ID	SAL	COMISSAO_PCT	GERENTE_ID	DEPT_ID
1500	Rafael	Armando	r.armando@email.com	(12)3155-5578	2008-04-02	1	2500.00	0.05	(NULL)	20
1523	Ricardo	Amaral	r.amaral@email.com	(12)3155-7746	2010-02-20	2	900.00	0.15	1560	(NULL)
1534	Luciane	Silva	l.silva@email.com	(12)3155-5577	2008-12-04	2	900.00	0.15	1500	20
1548	Denise	Siqueira	d.siqueira@email.com	(12)3155-7854	2008-08-10	2	850.00	0.15	1500	20
1560	Lucas	Messias	l.messias@email.com	(12)3155-4477	2010-01-15	1	2000.00	0.05	(NULL)	(NULL)
1568	Vinícius	Andrade	v.andrade@email.com	(12)3155-4755	2009-05-03	2	800.00	0.15	1577	10
1574	Leandro	Martínez	l.martinez@email.com	(12)3155-4758	2009-01-28	3	1450.00	0.10	1577	10
1577	Roberto	Borges	r.brges@email.com	(12)3155-4487	2007-09-15	1	2800.00	0.05	(NULL)	10
1587	Daniel	Vasquez	d.vasquez@email.com	(12)3155-5589	2008-05-01	3	1500.00	0.10	1500	30

departamentos:

DEPT_ID	DEPT_NOME	GERENTE_ID	LOCAL_ID
10	DEBIQ	1500	1
20	DEMAR	1577	2
30	DEQUI	(NULL)	1
40	DEBAS	(NULL)	1

locais:

LOCAL_ID	ENDERECO	CEP	CIDADE	ESTADO
1	Estrada do Campinho	12600-000	Lorena	SP
2	Santa Lucrécia	12600-000	Lorena	SP

Insira tais dados nelas

Junção com mais de duas tabelas

Os relacionamentos criados possibilitam a junção entre as 3 tabelas exemplificada pelos comandos abaixo:

Sintaxe antiga	ON
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome, l.cidade, l.estado FROM empregados e, departamentos d, locais l WHERE e.dept_id=d.dept_id AND d.local_id=l.local_id;	SELECT e.pnome, e.unome, d.dept_id, d.dept_nome, l.cidade, l.estado FROM empregados e INNER JOIN departamentos d ON (e.dept_id=d.dept_id) INNER JOIN locais l ON (d.local_id=l.local_id);

Os relacionamentos criados possibilitam a junção entre as 3 tabelas exemplificada pelos comandos abaixo:

USING	NATURAL JOIN
SELECT e.pnome, e.unome, dept_id, d.dept_nome, l.cidade, l.estado FROM empregados e INNER JOIN departamentos d USING (dept_id) INNER JOIN locais l USING (local_id);	Não é possível porque a coluna GERENTE_ID é comum a empregados e a departamentos e não é usada na cláusula de junção deste relacionamento

Junção com mais de duas tabelas

Junção com mais de duas tabelas

Observações:

- Na maior parte das situações é válido o seguinte principio: se há n tabelas temos que usar n-1 condições de junção. Esta regra não é válida quando existe mais que uma relação entre duas tabelas;
- Cada relacionamento entre duas tabelas exige uma condição de junção.
- A condição de junção deve incluir todas as colunas usadas na junção;

A cláusula de junção e a cláusula de filtro

O facto de efetuarmos uma junção entre duas tabelas não impede que se faça um filtro das linhas. O exemplo abaixo mostra como:

Sintaxe antiga	ON
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e, departamentos d WHERE e.dept_id=d.dept_id AND e.pnome LIKE 'R%';	SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e INNER JOIN departamentos d ON (e.dept_id=d.dept_id) WHERE e.pnome LIKE 'R%';

A cláusula de junção e a cláusula de filtro

O facto de efetuarmos uma junção entre duas tabelas não impede que se faça um filtro das linhas. O exemplo abaixo mostra como:

USING	NATURAL JOIN
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e INNER JOIN departamentos d USING (dept_id) WHERE e.pnome LIKE '%r%';	Não é possível porque a coluna GERENTE_ID é comum a Empregados e a Departamentos e não é usada na cláusula de junção deste relacionamento

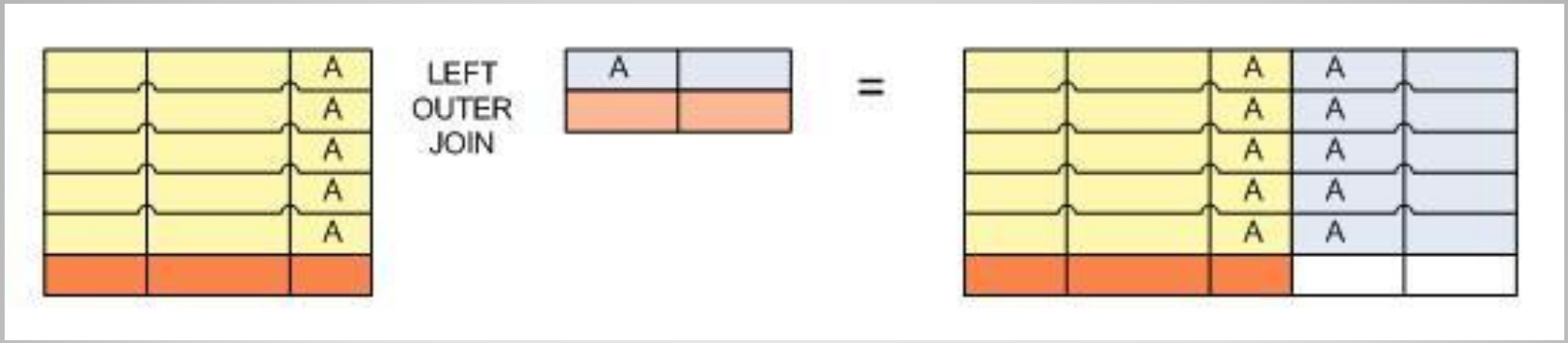
Junção externa

A junção externa é uma extensão da junção interna.

Por exemplo:

Considerando apenas o relacionamento "Trabalha Em" verificamos que um empregado pode não ter um departamento atribuído, assim como um departamento pode não ter empregados atribuídos. Os empregados que não têm departamento e os departamentos que não têm empregados não aparecem no resultado de uma junção interna, mas aparecem na junção externa.

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. O diagrama apresentado a seguir mostra como funciona este tipo de junção:



Junção externa à esquerda

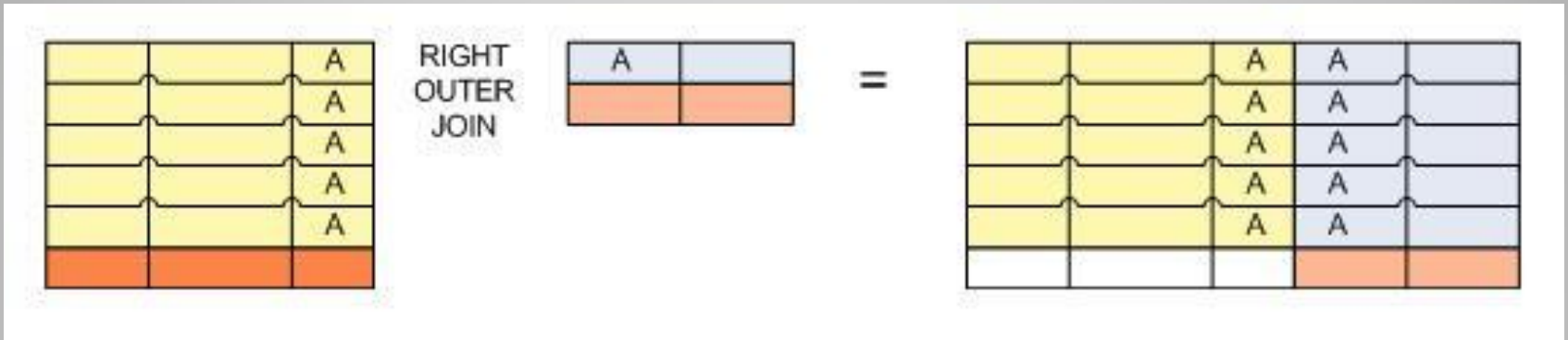
Junção externa à esquerda

Os comandos apresentados a seguir mostram como fazer a junção externa à esquerda entre as tabelas **empregados** e **departamentos**. 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 **DEPT_ID** em **empregados** suporta valores nulos:

ON	USING
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e LEFT JOIN departamentos d ON (e.dept_id=d.dept_id);	SELECT e.pnome, e.unome, dept_id, d.dept_nome FROM empregados e LEFT JOIN departamentos d USING (dept_id);

***Natural Join: Não aplicável**

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. O diagrama apresentado a seguir descreve o funcionamento deste tipo de junção:



Junção externa à direita

Junção externa à direita

Os comandos apresentados a seguir mostram como fazer a junção externa à direita entre as tabelas **empregados** e **departamentos**. Esta operação vai mostrar as linhas obtidas pela junção interna, acrescida dos departamentos que não têm um empregado atribuído:

ON	USING
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e RIGHT JOIN departamentos d ON (e.dept_id=d.dept_id);	SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e RIGHT JOIN departamentos d USING (dept_id);

***Natural Join: Não aplicável**

Junção externa à direita

Os comandos apresentados a seguir mostram como fazer a junção externa FULL entre as tabelas **empregados** e **departamentos**:

ON	USING
SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e RIGHT JOIN departamentos d ON (e.dept_id=d.dept_id);	SELECT e.pnome, e.unome, d.dept_id, d.dept_nome FROM empregados e RIGHT JOIN departamentos d USING (dept_id);

***Natural Join: Não aplicável**

Continua...