Junção de tabelas

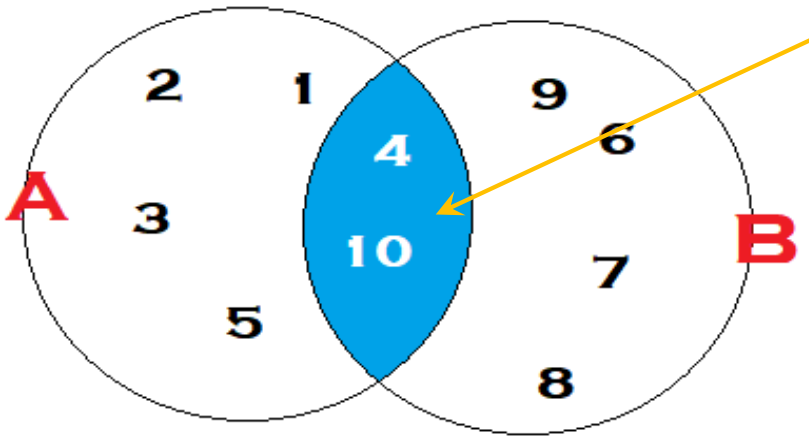
Banco de Dados II

Profa. Simone Carboni Garcia

tsi tecnologia em sistemas para internet

Junção de tabelas

- Para obter informações de um banco de dados, muitas vezes é necessário acessar simultaneamente várias tabelas. Esse processo leva à realização de **junções** entre as tabelas, para extração das informações necessárias à consulta formulada.



Junção
(intersecção)

A={1,2,3,4,5,10}

B={4,6,7,8,10}

A ∩ B = {4,10}

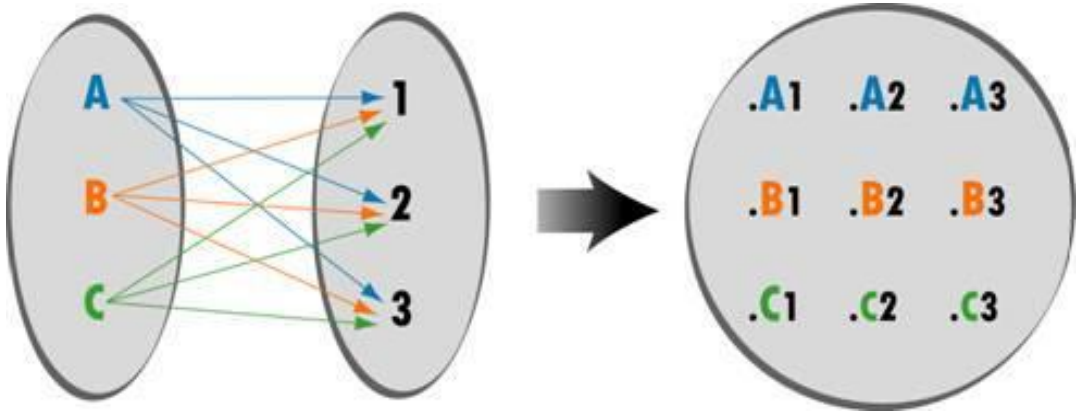
Junção de tabela

- Assim, **junção** é uma condição necessária para se obter dados provenientes de mais de uma tabela.
- Os registros de uma tabela podem ser ligados a outros registros de outra tabela por meio de **valores em comum** que possam existir nos atributos correspondentes, normalmente atributos que servem de chave primária e chave estrangeira.

Produto Cartesiano

- Se a **condição de junção** é inválida ou omitida, o resultado é o **produto cartesiano** em que a combinação de todos os registros das tabelas especificadas no SELECT é exibida.
- O produto cartesiano tende a gerar um grande número de registros e o seu resultado raramente é utilizado.
- Utiliza-se **produto cartesiano** quando é necessário unir todos os registros das tabelas.

Produto Cartesiano



- Exemplo

emp (130 registros)

Kcd_emp	nm_emp	. . .	cd_dept
1073	Jacson C.	. . .	10
1118	Pedro F.	. . .	30
1231	Ramiro A.	. . .	100
⋮	⋮	⋮	⋮
1781	Paula J.	. . .	30

dept (10 registros)

Kcd_dept	nm_dept	nm_loc
10	Marketing	Pelotas
20	GRH	POA
30	Contabilidade	POA
⋮	⋮	⋮
100	Almoxarifado	POA

```
SELECT nm_emp, nm_dept, nm_loc
FROM emp, dept;
```

resultado (1300 registros)

nm_emp	nm_dept	nm_loc
Jacson C.	Marketing	Pelotas
Pedro F.	Marketing	Pelotas
Ramiro A.	Marketing	Pelotas
⋮	⋮	⋮
Jacson C.	GRH	POA
Pedro F.	GRH	POA
⋮	⋮	⋮

Produto Cartesiano

- No exemplo do *slide* e anterior, foi executado o **produto cartesiano** das tabelas EMP e DEPT. Neste caso, poucas informações úteis podem ser extraídas da tabela resultante. Devem ser aplicada **condição de junção** à consulta para um resultado mais preciso.

Junção

- A **junção** ocorre quando há uma condição de igualdade entre os atributos das tabelas (normalmente entre a chave primária de uma tabela e a chave estrangeira de outra tabela).

emp (130 registros)

Kcd_emp	nm_emp	. . .	cd_dept
1073	Jacson C.	. . .	10
1118	Pedro F.	. . .	30
1231	Ramiro A.	. . .	100
⋮	⋮	⋮	⋮
1781	Paula J.	. . .	30

Chave estrangeira

dept (10 registros)

Kcd_dept	nm_dept	nm_loc
10	Marketing	Pelotas
20	GRH	POA
30	Contabilidade	POA
⋮	⋮	⋮
100	Almoxarifado	POA

Chave primária

```
SELECT emp.nm_emp, dept.nm_dept,
nm_loc
  FROM emp, dept
WHERE emp.cd_dept =
dept.Kcd_dept;
```

resultado (130 registros)

nm_emp	nm_dept	nm_loc
Jacson C.	Marketing	Pelotas
Pedro F.	Contabilidade	POA
Ramiro A.	Almoxarifado	POA
⋮	⋮	⋮
Paula J.	Contabilidade	POA

Tipo e condição de junção

- Cada variante das operações de junção consiste em um tipo de junção e em uma condição de junção.
- Tipo de junção:
  - Determina como os registros de uma tabela que não possui nenhuma correspondência com os registros de outra tabela devem ser tratados.
- Condição de junção:
  - Define quais registros das duas tabelas apresentam correspondência e quais atributos são apresentados no resultado de uma junção.

Tipos de junção

- inner join
- non equijoin
- left outer join
- right outer join
- full outer join
- self join

Condições de junção

- natural
- on <predicado>
- using ($atributo_1$, $atributo_2$, ..., $atributo_n$)

INNER JOIN

- Este tipo de junção, também chamada de junção simples ou interna, é uma junção onde há uma condição de igualdade entre os atributos das tabelas (normalmente entre a chave primária de uma tabela e a chave estrangeira de outra tabela).

INNER JOIN

- No exemplo ao lado, para determinar o departamento de um empregado, é necessário comparar o valor do atributo cd_dept na tabela emp com o valor do atributo kcd_dept na tabela dept.
- A relação entre as duas tabelas é uma equi-junção, ou seja, os valores em ambos os atributos devem ser iguais.

Kcd_emp	nm_emp	. . .	cd_dept
1073	Jacson C.	. . .	10
1118	Pedro F.	. . .	30
1231	Ramiro A.	. . .	100
⋮	⋮	⋮	⋮
1781	Paula J.	. . .	30

Chave estrangeira

Kcd_dept	nm_dept	nm_loc
10	Marketing	Pelotas
20	GRH	POA
30	Contabilidade	POA
⋮	⋮	⋮
100	Almoxarifado	POA

Chave primária

```
SELECT emp.nm_emp, dept.nm_dept,
nm_loc
  FROM emp, dept
WHERE emp.cd_dept =
dept.Kcd_dept;
```

resultado (130 registros)

nm_emp	nm_dept	nm_loc
Jacson C.	Marketing	Pelotas
Pedro F.	Contabilidade	POA
Ramiro A.	Almoxarifado	POA
⋮	⋮	⋮
Paula J.	Contabilidade	POA

INNER JOIN – condição de junção NATURAL JOIN

- Condição de junção: é utilizada na cláusula FROM.
  - NATURAL JOIN
    - Utilizada para simplificar o comando quando a chave primária e a chave estrangeira têm o mesmo nome em ambas as tabelas.

      ```
      SELECT emp.nm_emp, dept.nm_dept, nm_loc
      FROM emp NATURAL JOIN dept;
      ```

    - Observação: para este exemplo ser válido, deve-se considerar que os dois atributos nas duas tabelas possuem o nome **cd_dept** ou o nome **Kcd_dept**.

INNER JOIN – condição de junção USING

- USING
  - Utilizada para simplificar o comando quando a chave primária e a chave estrangeira têm o mesmo nome em ambas as tabelas.

    SELECT emp.nm_emp, dept.nm_dept, nm_loc

    FROM emp **JOIN** dept **USING** (cd_depto);
  - Observações:
    - Para este exemplos ser válido, deve-se considerar que os dois atributos nas duas tabelas possuem o nome **cd_depto**.
    - A palavra INNER é opcional.

Inner join – condição de junção ON

- ON
  - Utilizada quando a chave primária e a chave estrangeira não possuem o mesmo nome nas tabelas relacionadas na cláusula FROM.

    SELECT emp.nm_emp, dept.nm_dept, nm_loc

    FROM emp **INNER JOIN** dept **ON** (emp.cd_dept= dept.Kcd_depto);
  - Observação: A palavra INNER é opcional.

Junção de duas ou mais tabelas

- NATURAL JOIN
  - A ordem em que são colocadas as tabelas na cláusula FROM determina quais tabelas serão pesquisadas primeiro.
  - Logo, se colocarmos as tabelas menores em primeiro lugar a busca será mais rápida.

```
SELECT emp.nm_emp, dept.nm_dept, categ_salarial.vl_sal
FROM emp NATURAL JOIN dept NATURAL JOIN categ_salarial;
```

Junção de duas ou mais tabelas

- **JOIN**

  SELECT lista de atributos

  FROM tabela1 [inner] **JOIN** tabela2 ON|USING condição de junção
  [**JOIN** tabela3 ON|USING ...]

Condições suplementares

- Pode-se acrescentar critérios suplementares por meio da cláusula WHERE.

```
SELECT emp.nm_emp, dept.nm_dept, categ_salarial.vl_sal
FROM emp NATURAL JOIN dept NATURAL JOIN categ_salarial
WHERE emp.cd_dept = 30;
```

NON_EQUIJOIN

- Esta junção é utilizada quando não existe uma relação direta entre os atributos de duas tabelas, ou seja, quando a relação não é obtida por um operador relacional de igualdade (=).
- A junção é obtida por um intervalo (between).

NON_EQUIJOIN

emp

Kcd_emp	nm_emp	vl_salario	cd_dept
1073	Jacson C.	1300	10
1118	Pedro F.	1550	30
1231	Ramiro A.	720	100
⋮	⋮	⋮	⋮
1781	Paula J.	1430	30

categ_salarial

Kcd_categ	vl_nim	vl_max
1	700	1200
2	1201	1500
3	1501	1700
4	1701	2300

intervalo

Qual a categoria salarial de cada empregado?

```
SELECT emp.nm_emp, categ_salarial.kcd_categ, emp.vl_sal
  FROM  emp, categ_salarial
WHERE emp.vl_salario
BETWEEN categ_salarial.vl_min AND categ.salarial.vl_max;
```

resposta

nm_emp	Kcd_categ	vl_salario
Jacson	2	1300
Pedro	3	1550
Ramiro	1	720
⋮	⋮	⋮
Paula	2	1430

OUTER JOIN

- A junção OUTER JOIN é também conhecida como junção externa.
- É aplicada quando se deseja mostrar registros de uma tabela que não satisfazem a condição de junção com outra tabela.
- O atributo com valor faltante recebe no resultado da consulta um valor nulo (null).

OUTER JOIN

- Tipos de junções externas:
  - left outer join
  - right outer join
  - full outer join
- Observação: A palavra OUTER é opcional em qualquer um dos tipos.

LEFT OUTER JOIN

- Esta junção de união pela esquerda, incluirá no resultado da seleção todos os registros da primeira tabela listada na cláusula FROM, mesmo que não tenham relação com os registros da segunda tabela.

emp

Kcd_emp	nm_emp	vl_salario	cd_categ
1073	Jacson C.	1300	2
1118	Pedro F.	1550	3
1231	Ramiro A.	720	1
1781	Paula J.	1430	2

categ_salarial

Kcd_categ	vl_nim	vl_max
1	700	1200
2	1201	1500
3	1501	1700
4	1701	2300

```
SELECT categ_salarial.kcd_categ, emp.nm_emp
  FROM  categ_salarial LEFT OUTER JOIN emp
    ON  (categ_salarial.kcd_categ=emp.cd_categ);
```

resposta

Kcd_categ	nm_emp
1	Ramiro A.
2	Jacson C.
2	Paula J.
3	Pedro F.
4	NULL

RIGHT OUTER JOIN

- Na união externa à direita todos os registros da segunda tabela serão incluídos na busca, mesmo sem haver relação com os registros da primeira tabela.

```
SELECT categ_salarial.kcd_categ, emp.nm_emp
 FROM  emp RIGHT OUTER JOIN categ_salarial
  ON (categ_salarial.kcd_categ=empl.cd_categ);
```

resposta

Kcd_categ	nm_emp
1	Ramiro A.
2	Jacson C.
2	Paula J.
3	Pedro F.
4	NULL

FULL OUTER JOIN

- Combina os tipos de junções externa à esquerda e à direita.
- Os registros da tabela do lado esquerdo que não correspondem a nenhum dos registros do lado direito são preenchidos com valores nulos e, depois, adicionados ao resultado da seleção.
- Os registros do lado direito que não coincidem com nenhum dos registros da tabela do lado esquerdo são também preenchidos com nulos e adicionados ao resultado.

```
SELECT categ_salarial.kcd_categ, emp.nm_emp
  FROM  emp FULL OUTER JOIN categ_salarial
    ON (categ_salarial.kcd_categ=empl.cd_categ);
```

SELF JOIN (auto-junção)

- Muitas vezes é necessário fazer uma junção numa só tabela.
- Por exemplo:
  - Para saber o nome do chefe de cada empregado é necessário fazer uma junção da tabela emp com ela mesma, isto porque um chefe é ao mesmo tempo um empregado.

emp

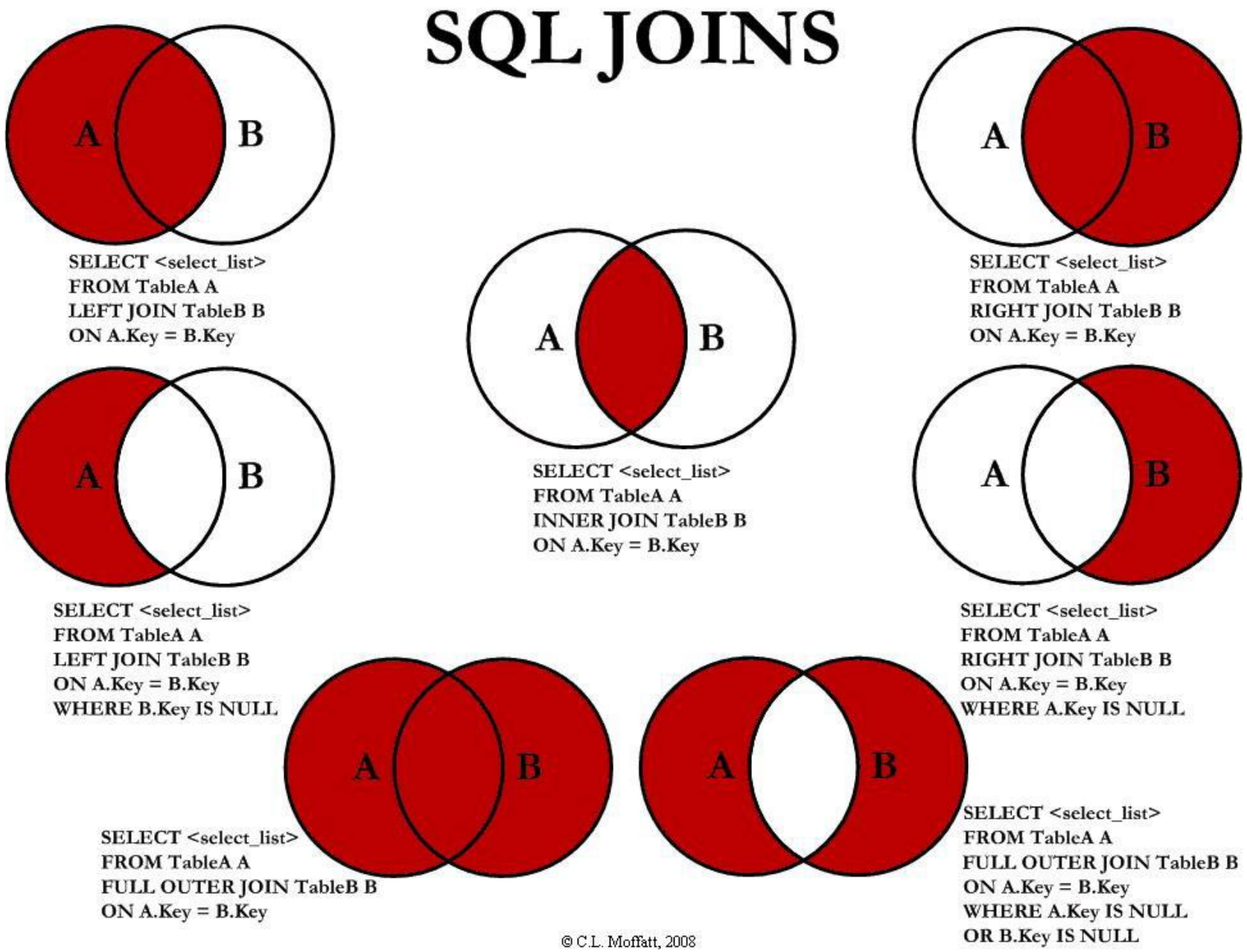
Kcd_emp	nm_emp	cd_chefe	cd_dept
1073	Jacson C.	1032	10
1118	Pedro F.	1032	30
1231	Ramiro A.	1032	100
⋮	⋮	⋮	⋮
1781	Paula J.	1032	30
1032	Carla M.	1032	40

```
SELECT funcion.nm_emp, gerencia.nm_emp as nm_chefe
  FROM  emp gerencia, emp funcion
WHERE gerencia.cd_chefe = funcion.kcd_emp;
```

resposta

nm_emp	nm_chefe
Jacson C.	Carla M.
Pedro F.	Carla M.
Ramiro A.	Carla M.
⋮	⋮
Paula J.	Carla M.
Carla M.	Carla M.

Resumindo...


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

Referências

- Oliveira, Celso H. P. de **SQL: Curso Prático**. São Paulo: Novatec, 2002.
- KORTH, Henry F.; SILBERSCHATZ, Abraham. **Sistemas de Banco de dados**. 3ª ed. São Paulo: Makron Books, 1999.
- **Select–nível 2**. Disponível em: <http://materialdornel.readthedocs.io/pt_BR/latest/linguagem-sql/select_ParteDois.html>.
- E-knowledge. **Produto cartesiano**. Disponível em: <http://e-reality-database.blogspot.com.br/2007/12/produto-cartesiano.html>