Banco de Dados - Ciclo 3 - Webaula 21

WEBAULA 21

- Seleção de Dados com Várias Tabelas
- Referências

QUESTION BANKS

Seleção de Dados com Várias Tabelas

Agora você terá a oportunidade de conhecer os recursos para realizar consultas de dados em duas ou mais tabelas através do comando SELECT. O conceito que envolve o recurso de busca em vários objetos de banco de dados relacionais (tabela) está atrelado ao termo junção. Uma junção (*join*) em banco de dados relacionais é uma condição pela qual os dados são submetidos a outro conjunto de dados (uma tabela, por exemplo). Existem quatro tipos de junções, as quais serão apresentadas a partir de agora. Antes, porém, vamos conhecer algumas premissas para o uso das junções e a sintaxe utilizada.

> **Dica**
>
> Assim como nos conteúdos anteriores, recomendamos que você vá executando os comandos de exemplo para colocar em prática os conceitos apresentados. E lembre-se: qualquer dúvida, procure interagir com o professor e seus colegas.

Para utilizar uma junção, você deve relacionar colunas de diferentes tabelas no banco de dados. A forma mais comum de fazer isso é através da sintaxe apresentada no Quadro 59.

Quadro 59 – Sintaxe da utilização de junção

```
SELECT    tabela1.coluna, tabela2.coluna
FROM      tabela1, tabela2
```

```
WHERE     tabela1.coluna = tabela2.coluna;
```

Não é uma condição elementar comum a todos os SGBDs relacionais, mas uma boa prática na escrita de sentenças, utilizar como prefixo de cada coluna o nome da tabela (ou o apelido), conforme os exemplos acima e a seguir (Quadro 60). Isso vai ajudar na legibilidade da sentença, assim como prevenir possíveis problemas que o SGBD relacional que você estiver utilizando venha apresentar na submissão do comando.

Quadro 60 – Exemplo da utilização de junção com apelido

```
SELECT    t1.coluna, t2.coluna
FROM      tabela1 t1, tabela2 t2
WHERE     t1.coluna = t2.coluna;
```

Para um melhor entendimento do uso da junção, vamos a um exemplo. Considere o cenário da Figura 52, que envolve estruturas já conhecidas por você. Vamos criar comandos para exemplificar os conceitos utilizando os dados destas tabelas.

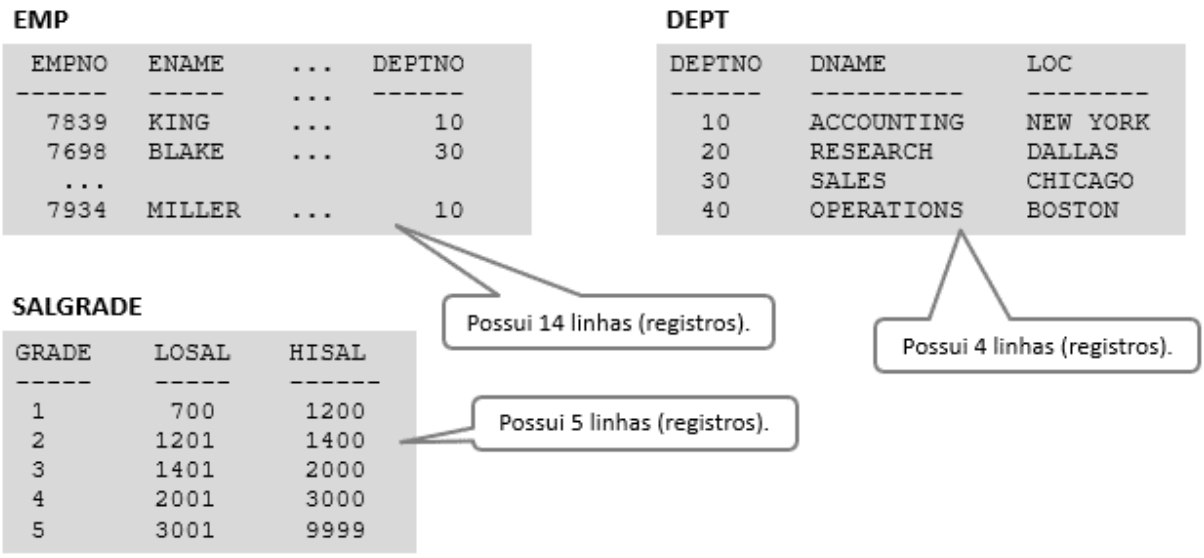


Figura 52

O objetivo de trabalhar com um cenário em que você já está familiarizado é para facilitar o entendimento sobre o conceito de junção, assim como exemplificar os tipos existentes. Reforço a afirmação de que ser trata de um cenário adaptado de ORACLE (2017).

Observe que temos as tabelas EMP, DEPT e SALGRADE. Na tabela EMP a chave primária é a coluna EMPNO. Esta tabela EMP apresenta duas chaves estrangeiras, sendo que uma para tabela DEPT, através da coluna DEPTNO, e outra para a própria tabela através da coluna MGR, caracterizando assim um autorrelacionamento. A tabela DEPT apresenta uma coluna DEPTNO como chave primária e não apresenta chave estrangeira. Por último, temos a tabela SALGRADE, que tem como chave primária a coluna GRADE e também não possui chave estrangeira. Passemos a seguir para a exploração dos 4 (quatro) tipos de junções encontradas nos SGBDs relacionais.

A mais comum e utilizada é a **junção idêntica**. A junção idêntica ocorre quando fazemos uma condição onde uma chave estrangeira (FK) de uma tabela é associada a uma chave primária (PK) de outra tabela. No exemplo a seguir (Figura 53), temos a coluna "DEPTNO", que é uma FK da tabela "EMP", sendo comparada com a coluna "DEPTNO", que é a PK da tabela "DEPT".

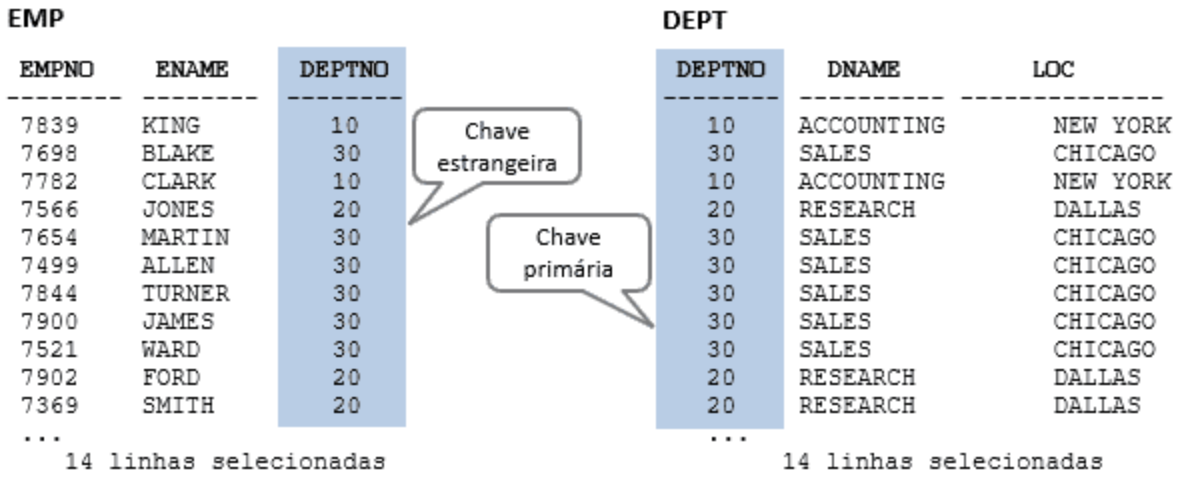


Figura 53

Agora, imagine que você queira produzir uma consulta cujo resultado apresente as colunas "EMPNO", "DEPTNO" e "LOC", conforme visualizado na Figura 54.

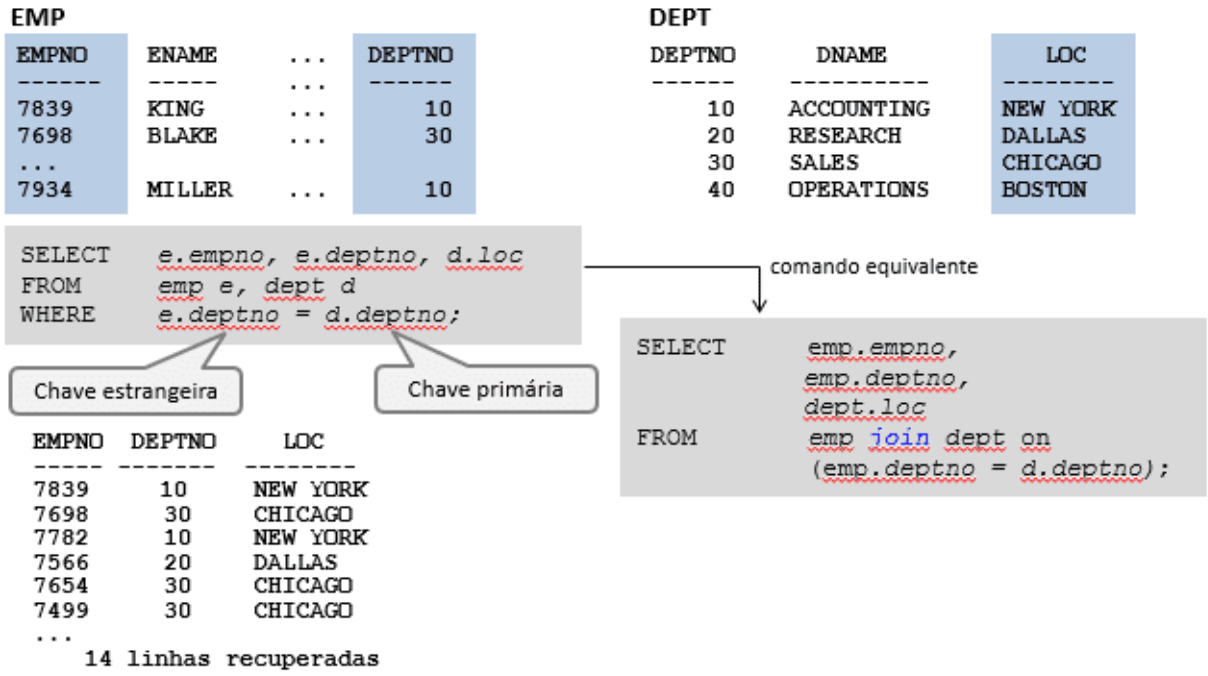


Figura 54

Como dito, na ilustração acima temos um exemplo da utilização da junção idêntica. Conforme já apresentado, a junção idêntica envolve sempre uma chave primária e uma chave estrangeira. Observe que no exemplo da sentença SQL estamos relacionando a chave estrangeira da tabela EMP com a chave primária da tabela DEPT. O primeiro exemplo (esquerda) é mais praticado e recomendado pela literatura, porém o segundo (direita), que utiliza a palavra reservada JOIN, é equivalente suportado em todos os SGBDs relacionais.

Um aspecto que exige especial atenção no momento da utilização da junção entre tabelas está relacionado a ligações entre as tabelas. Isso porque é comum a ocorrência do problema do produto cartesiano. O produto cartesiano é caracterizado quando todas as linhas da primeira tabela estiverem

unidas a todas as linhas da segunda tabela, formando todas as combinações possíveis de resultado. Um produto cartesiano é formado quando:

- uma condição de junção estiver omitida;
- uma condição de junção estiver inválida;

Acompanhe a ilustração da Figura 55, onde temos a representação de um produto cartesiano.

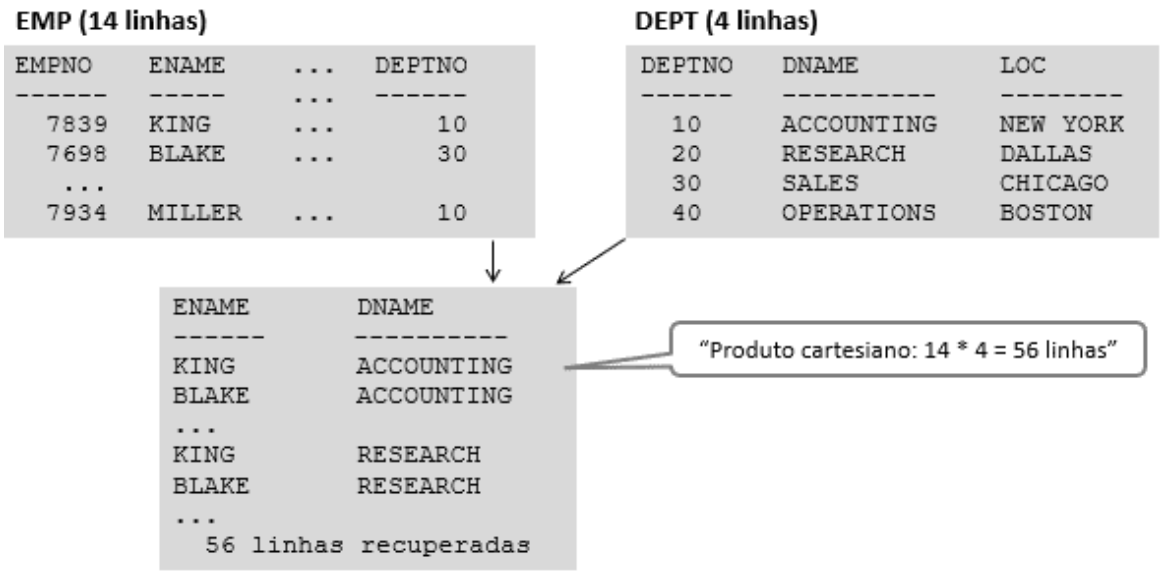


Figura 55

Dica

Atenção! Fique atento sempre que criar sentenças do comando SELECT que envolver mais de uma tabela na cláusula FROM. A omissão de condição de junção é muito comum.

Quando declaramos mais de uma tabela na cláusula FROM e omitimos uma condição de junção, ou então indicamos uma condição de junção inválida, o produto cartesiano entre as tabelas envolvidas irá ocorrer. Veja que no cenário (Figura 55) temos 14 linhas na tabela EMP e quatro linhas na tabela DEPT, logo o produto cartesiano vai gerar 56 linhas que nada mais são do que a multiplicação da quantidade de linhas de cada uma das tabelas envolvidas. Portanto, veja que o empregado KING está associado a todos os departamentos, assim como os demais também estarão. A exceção para que o resultado não seja confundido com o produto cartesiano ocorre quando todas as linhas de uma mesma tabela estão relacionadas a todas as linhas da outra tabela, mas isto é algo muito raro de ocorrer.

Para evitar um produto cartesiano, sempre inclua uma condição de junção válida em uma cláusula WHERE. Outra dica importante é sempre observar a quantidade de tabelas adicionadas na cláusula FROM. Há uma regra simples que vale a pena seguir: o número de tabelas na cláusula FROM menos um (n.º tabelas – 1) de junções válidas na cláusula WHERE. Vamos a um exemplo. Abaixo temos no Quadro 61 nosso cenário com a sentença SQL gerando um produto cartesiano.

Quadro 61 – Exemplo de sentença com produto cartesiano

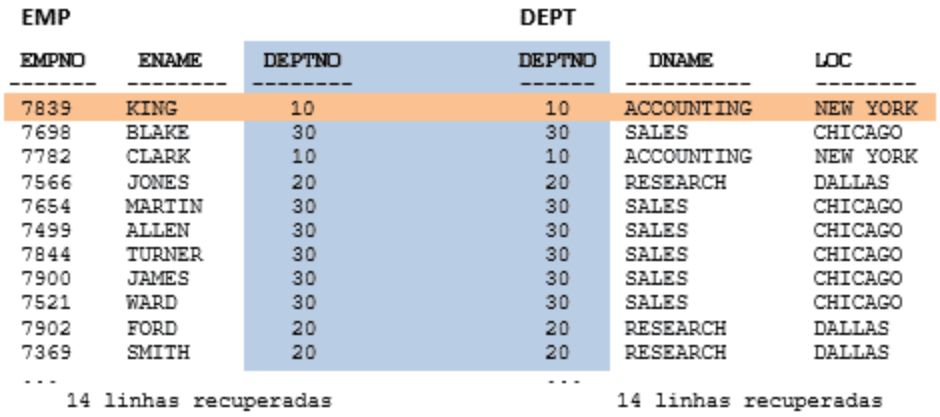
```
SELECT    e.empno, e.deptno, d.loc
FROM        emp e, dept d;
```

No exemplo acima, você deve ter percebido que não há cláusula WHERE. Consequentemente, nenhuma condição de junção foi criada. No entanto, observe que há duas tabelas declaradas da cláusula FROM. Assim, ao aplicarmos nossa regra, no mínimo uma junção válida deveria existir.

Dica

Atenção! Quando a chave estrangeira de uma tabela é composta por duas ou mais colunas, cada coluna deverá apresentar uma junção para que o produto cartesiano seja evitado.

Você pode acrescentar, aliada(s) à junção, uma ou mais condições adicionais para filtrar o resultado. Considere o cenário exemplo da Figura 56, onde se espera recuperar as informações do empregado "KING".



EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20

14 linhas recuperadas

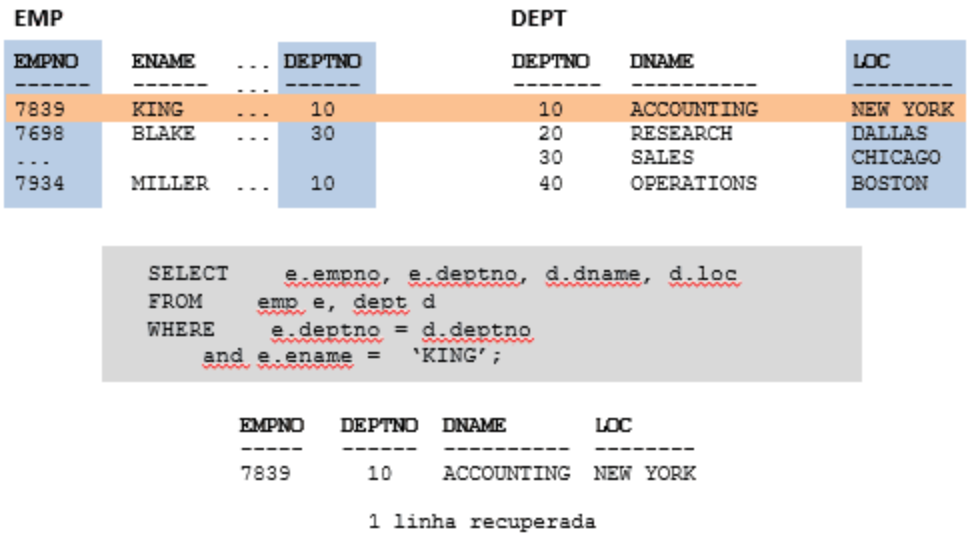
DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS

14 linhas recuperadas

Figura 56

Cenários como o apresentado podem ser denominados junção idêntica condicionada. Para construirmos uma sentença SQL de exemplo, imagine que você queira produzir uma consulta cujo resultado apresente as colunas "EMPNO", "DEPTNO" e "LOC", conforme visualizado na Figura 57.



EMP

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SELECT    e.empno, e.deptno, d.dname, d.loc
FROM      emp e, dept d
WHERE     e.deptno = d.deptno
      and e.ename =  'KING';
```

EMPNO	DEPTNO	DNAME	LOC
7839	10	ACCOUNTING	NEW YORK

1 linha recuperada

Figura 57

Na ilustração temos o cenário, seguido da sentença e resultado produzido. Vemos claramente a cláusula WHERE com a junção idêntica (relação entre a FK e a PK) e em seguida o filtro, ou seja, nome (ENAME) do empregado sendo comparado a “KING”.

> **Dica**
>
> A junção idêntica é muito comum e frequente nas sentenças SQL. Quando desejamos adicionar uma ou mais condições para filtrar o resultado, recomendamos que seja feito após a declaração das junções, pois isso minimiza a possibilidade de ocorrência de produto cartesiano por um eventual esquecimento, ou seja, prioritariamente definem-se as ligações, depois os filtros.

Passemos agora para o segundo tipo de junção: a externa. Utilizamos uma **junção externa** para consultar também todas as linhas que em geral não atendem à condição de junção idêntica. No exemplo ilustrado na Figura 58, podemos visualizar um cenário onde o departamento 40

(OPERATIONS) não está vinculado a nenhum empregado, isto é, há um registro na tabela DEPT com este valor, porém ele não está associado a nenhuma linha da tabela EMP.

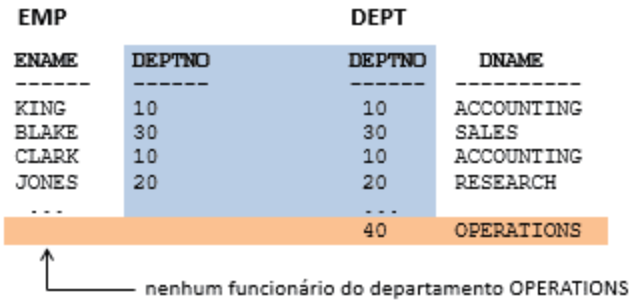


Figura 58

Se você construir um comando SELECT utilizando uma junção idêntica, a linha (registro) que não estiver relacionada estará ausente do resultado. E é exatamente esse o papel da junção externa, ou seja, permitir que todas as linhas sejam apresentadas, independentemente da condição de relação.

A sintaxe de utilização da junção externa conforme o padrão ANSII apresenta duas formas de utilização, conforme o Quadro 62.

Quadro 62 – Sintaxe de escrita da junção externa

```
SELECT    tabela1.coluna1, tabela2.coluna1
FROM      tabela1 LEFT JOIN tabela2 ON
          (tabela1.coluna1 = tabela2.coluna1);


OU


SELECT    tabela1.coluna1, tabela2.coluna1
FROM      tabela1, RIGHT JOIN tabela2 ON
```

```
(tabela1.coluna1 = tabela2.coluna1);
```

Especificamente no caso do SGBD Oracle, como alternativa, é possível realizar uma junção externa através da simbologia "(+)" ao lado da coluna. O Quadro 63 apresenta a sintaxe da junção externa como alternativa, exclusivamente, do SGBD Oracle.

Quadro 63 – Sintaxe de escrita da junção externa do SGBD Oracle

```
SELECT      tabela1.coluna1, tabela2.coluna1
FROM        tabela1, tabela2
WHERE       tabela1.coluna1 (+) = tabela2.coluna1;
```

Dica

No caso do SGBD Oracle, que utiliza como alternativa o sinal de mais entre parênteses (+) ao lado da coluna, é importante destacar que não é possível utilizá-lo simultaneamente ao lado de ambas as colunas da condição.

As formas de utilização da junção externa, segundo o padrão ANSII, possibilitam a realização da junção considerando o lado esquerdo da condição imposta, conforme ilustrado no primeiro exemplo (Quadro 64). A segunda forma permite que sejam considerados os valores existentes na tabela do lado direito da condição de junção, conforme você pode observar no segundo exemplo (Quadro 64).

Quadro 64 – Exemplo de sentença utilizando junção externa padrão ANSII

```
SELECT    e.empno, e.deptno, d.name, d.loc
FROM      emp e LEFT JOIN dept d
ON        (e.deptno = d.deptno);



SELECT    e.empno, e.deptno, d.dname, d.loc
FROM      emp e RIGHT JOIN dept d
           ON (e.deptno = d.deptno);
```

O exemplo apresentado acima considera o mesmo cenário e o conjunto de dados já visto anteriormente. No primeiro exemplo a sentença exibe todos os empregados e seus respectivos departamentos. No caso de o conjunto de dados recuperado apresentar um ou mais empregados sem a indicação de um departamento de vinculação, este também será apresentado sem a indicação do departamento, obviamente. Já no segundo exemplo temos o comando com a utilização da junção externa considerando o lado direito da condição. Neste caso, havendo um ou mais departamentos que não apresentem empregado vinculado, este por sua vez também será apresentado.

As duas primeiras formas de junção entre tabelas indicam, necessariamente, o envolvimento de chave primária e chave estrangeira. Porém, é possível realizar uma junção entre duas ou mais tabelas sem que exista uma chave estrangeira estabelecendo a relação/ligação com outra tabela (chave primária). Essa junção é conhecida como **junção não idêntica**. Nesse caso, o relacionamento é obtido por meio de um operador que não o igual "=". Isso porque o operador de igualdade (=) é peculiar à junção idêntica e à junção externa.

No exemplo ilustrado na Figura 59, podemos visualizar um cenário onde a tabela EMP apresenta uma coluna de salário (SAL). Há também uma tabela SALGRADE, que contempla faixas salarias indicando um número correspondente (coluna GRADE).

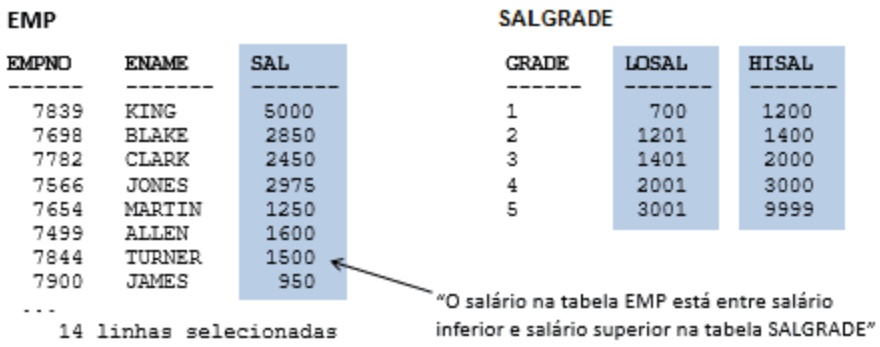


Figura 59

Considerando o contexto apresentado acima, observe um exemplo de utilização da junção não idêntica no Quadro 65.

Quadro 65 – Exemplo de sentença utilizando junção não idêntica

```
SELECT      e.empno, e.ename, s.grade
FROM        emp e, salgrade s
WHERE       e.sal BETWEEN s.losal AND s.hisal


-- comando equivalente


SELECT      e.empno, e.ename, s.grade
FROM        emp e, salgrade s
WHERE       e.sal >= s.losal
          and e.sal <= s.hisal;
```

Em ambos os exemplos, que correspondem à mesma sentença, podemos observar que não há indicação de relação de junção entre chave estrangeira e chave primária entre as tabelas. Como

resultado serão apresentados os dados do empregado, juntamente com o número correspondente ao nível salarial em que ele está enquadrado, segundo a tabela SALGRADE.

> **Dica**
>
> A junção não idêntica é pouco utilizada, pois se aplica em cenários bem específicos, geralmente no tratamento de dados com o viés de classificação a partir de parâmetros preestabelecidos em estruturas de tabelas.

Chegamos à quarta e última forma de junção. Antes de falarmos sobre ela, reflita. Você já parou para analisar como produzir resultados utilizando estruturas que apresentam autorrelacionamento ou relacionamento recursivo binário? A solução é simples: utilizar a **autojunção**. Assim como no relacionamento, onde uma tabela exerce o papel de duas (distintas), o mesmo ocorre na autojunção. Considere o cenário apresentado na Figura 60.

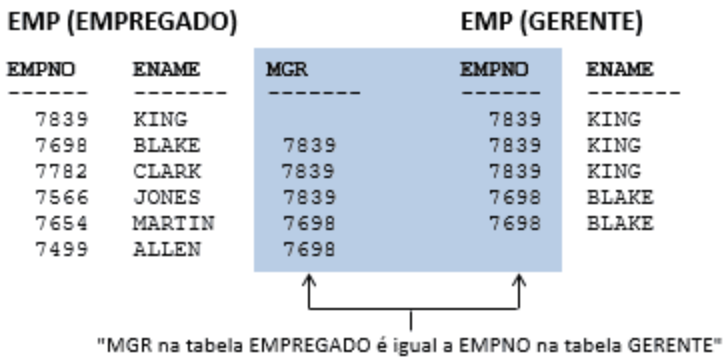


Figura 60

No exemplo acima, observe que a coluna “MGR” apresenta valores que correspondem à coluna “EMPNO” da própria tabela. É importante ressaltar ainda que a coluna “MGR” é uma chave estrangeira para a própria tabela – característica esta peculiar do autorrelacionamento.

Vamos construir uma sentença SQL para recuperar o nome do empregado e o nome do gerente (que também é um empregado). Para isso, vamos utilizar a tabela “EMP” exercendo o papel do empregado e, ao mesmo tempo, exercendo o papel do gerente. Veja novamente o cenário e a sentença SQL construída na Figura 61.

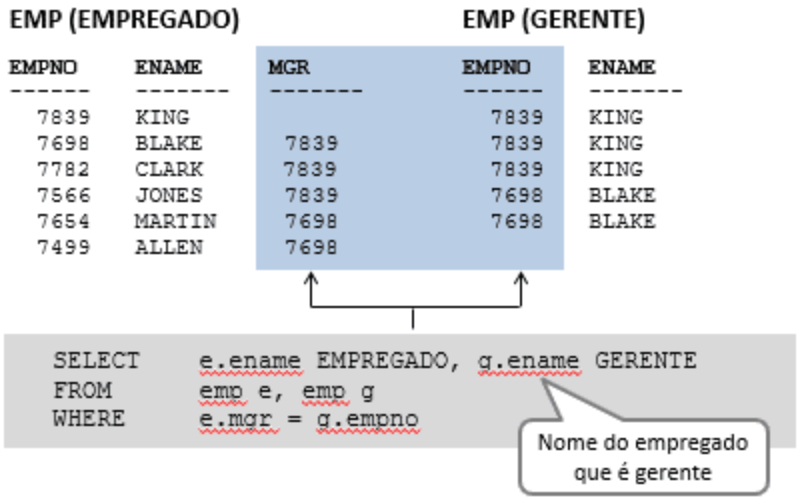


Figura 61

No cenário exemplo acima, podemos observar que a autojunção contempla uma condição de igualdade entre uma chave estrangeira e uma chave primária. A diferença para a junção idêntica é que nesta as colunas PK e FK pertencem à mesma tabela. Daí o nome “autojunção”. Como resultado da sentença do exemplo (Figura 61), teremos o nome do empregado com o respectivo nome do gerente ao lado.

Encerramento

Chegamos ao final de mais uma seção, onde conhecemos as características, aplicações e exemplos de utilização dos quatro tipos de junções: idêntica, externa, não idêntica e autojunção. A próxima seção irá tratar das funções de agrupamento de dados.

CONTINUE

Referências

Referências

ORACLE. Documentação de utilização do sistema (Versão 11g). Califórnia: Oracle Corp., 2017.

CARDOSO, Vírginia M. **Linguagem SQL**: fundamentos e práticas. São Paulo: Saraiva, 2009.

Referências de Imagens

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019.