



# Programação de Banco de Dados com SQL

12-2

Atualizando Valores de Coluna e Excluindo Linhas



# Objetivos

Nesta lição, você aprenderá a:

- Construir e executar uma instrução UPDATE
- Construir e executar uma instrução DELETE
- Construir e executar uma consulta que utiliza uma subconsulta para atualizar e excluir dados de uma tabela
- Construir e executar uma consulta que utiliza uma subconsulta correlacionada para atualizar e excluir dados de uma tabela

# Objetivos

Nesta lição, você aprenderá a:

- Explicar como as constraints de integridade de chave estrangeira e primária afetam as instruções UPDATE e DELETE
- Explicar o propósito da Cláusula FOR UPDATE em uma instrução SELECT

# Finalidade

- Não seria maravilhoso se, depois que algo ficasse pronto, nunca parasse de se modificar e refazer?
- A sua cama se faria sozinha, suas roupas permaneceriam limpas e você sempre tiraria notas boas.
- Infelizmente, nos bancos de dados, assim como na vida, "Nada é permanente, exceto a mudança".
- A atualização, a inserção, a exclusão e o gerenciamento de dados são tarefas do Administrador do Banco de Dados (DBA).
- Nesta lição, você se tornará o DBA do seu próprio esquema e aprenderá a gerenciar seu banco de dados.

**ORACLE®**

Academy

DPS12L2  
Atualizando Valores de Coluna e Excluindo Linhas

Copyright © 2019, Oracle e/ou suas empresas afiliadas. Todos os direitos reservados.

5



# UPDATE

- A instrução UPDATE é usada para modificar as linhas existentes em uma tabela.
- UPDATE requer quatro valores:
  - o nome da tabela
  - o nome da(s) coluna(s) cujos valores serão modificados
  - um novo valor para cada coluna que será modificada
  - uma condição que identifique as linhas da tabela que serão modificadas.
- O novo valor de uma coluna pode ser o resultado de uma subconsulta de linha única.

# UPDATE

- O exemplo mostrado usa uma instrução UPDATE para mudar o número de telefone de um funcionário na tabela de funcionários.
- Observe que a tabela copy\_employees é usada nesta transação.

```
UPDATE copy_employees  
SET phone_number = '123456'  
WHERE employee_id = 303;
```

É recomendável deixar a instrução UPDATE em uma linha própria.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Wright	123456

# UPDATE

- Podemos mudar várias colunas e/ou linhas em uma instrução UPDATE.
- Este exemplo muda o número de telefone e o sobrenome de dois funcionários.

```
UPDATE copy_employees  
SET phone_number = '654321', last_name = 'Jones'  
WHERE employee_id >= 303;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
303	Angelina	Jones	654321
304	Test	Jones	654321



# UPDATE

- Tome cuidado ao atualizar valores de coluna.
- Se a cláusula WHERE for omitida, todas as linhas da tabela serão atualizadas.
- Essa pode não ser a intenção.

```
UPDATE copy_employees  
SET phone_number = '654321', last_name = 'Jones';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	PHONE_NUMBER
100	Steven	Jones	654321
101	Neena	Jones	654321
102	Lex	Jones	654321
200	Jennifer	Jones	654321
205	Shelley	Jones	654321
206	William	Jones	654321
149	Eleni	Jones	654321
174	Ellen	Jones	654321
...	...	...	...

# Atualizando uma Coluna com um Valor de uma Subconsulta

- Podemos usar o resultado de uma subconsulta de linha única para fornecer o novo valor para uma coluna atualizada.

```
UPDATE copy_employees
SET salary = (SELECT salary
              FROM copy_employees
              WHERE employee_id = 100)
WHERE employee_id = 101;
```

- Este exemplo altera o salário de um funcionário (id = 101) para que fique igual ao salário de outro funcionário (id = 100).

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	24000

# Atualizando uma Coluna com um Valor de uma Subconsulta

```
UPDATE copy_employees
SET salary = (SELECT salary
              FROM copy_employees
              WHERE employee_id = 100)
WHERE employee_id = 101;
```

- Como é de costume, a subconsulta é executada primeiro e recupera o salário para o funcionário com id = 100.
- Em seguida, esse valor é usado para atualizar o salário do funcionário com id = 101.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
100	Steven	King	24000
101	Neena	Kochhar	24000

# Atualizando Duas Colunas com Duas Instruções de Subconsulta

- Para atualizar várias colunas em uma instrução UPDATE, é possível gravar várias subconsultas de linha única, uma para cada coluna.
- No exemplo a seguir, a instrução UPDATE altera o salário e id do cargo de um funcionário (id = 206) para que fiquem iguais aos mesmos valores de outro funcionário (id = 205).

# Atualizando Duas Colunas com Duas Instruções de Subconsulta

```
UPDATE copy_employees
SET salary = (SELECT salary
              FROM copy_employees
              WHERE employee_id = 205),
    job_id = (SELECT job_id
              FROM copy_employees
              WHERE employee_id = 205)
WHERE employee_id = 206;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	JOB_ID
205	Shelley	Higgins	12000	AC_MGR
206	William	Gietz	12000	AC_MGR

# Atualizando Linhas com Base em Outra Tabela

- Como você deve estar esperando, a subconsulta pode recuperar informações de uma tabela que, em seguida, serão usadas para atualizar outra tabela.
- Neste exemplo, foi criada uma cópia da tabela de funcionários.
- Depois, os dados da tabela de funcionários original foram recuperados, copiados e usados para preencher a tabela copy\_employees.

```
UPDATE copy_employees
SET salary = (SELECT salary
              FROM employees
              WHERE employee_id = 205)
WHERE employee_id = 202;
```

# Atualizando Linhas com a Subconsulta Correlacionada

- Como você já sabe, as subconsultas são individuais ou correlacionadas.
- Em uma subconsulta correlacionada, você atualiza uma linha em uma tabela com base em um SELECT dessa mesma tabela.



# Atualizando Linhas com a Subconsulta Correlacionada

- No exemplo abaixo, uma cópia da coluna com os nomes dos departamentos foi criada na tabela de funcionários.
- Depois, os dados da tabela de departamentos original foram recuperados, copiados e usados para preencher a cópia da coluna na tabela de funcionários.

```
ALTER TABLE  copy_employees  
ADD (department_name varchar2(30) NOT NULL);
```

```
UPDATE  copy_employees e  
SET e.department_name = (SELECT d.department_name  
                        FROM departments d  
                        WHERE e.department_id = d.department_id);
```



# DELETE

- A instrução DELETE é usada para remover as linhas existentes em uma tabela.
- Essa instrução requer dois valores:
  - o nome da tabela
  - a condição que identifica as linhas a serem excluídas



# DELETE

- O exemplo mostrado usa a cópia da tabela de funcionários para excluir uma linha: o funcionário com o ID 303.

```
DELETE from copy_employees  
WHERE employee_id = 303;
```

- O que você acha que será excluído se a cláusula WHERE for eliminada de uma instrução DELETE?
- Todas as linhas na tabela serão excluídas se você omitir a cláusula WHERE.

# DELETE de subconsulta

- As subconsultas também podem ser usadas nas instruções DELETE.
- O exemplo mostrado exclui linhas da tabela de funcionários para todos os funcionários que trabalham no departamento Shipping.
- Talvez esse departamento tenha sido renomeado ou fechado.

```
DELETE FROM copy_employees
WHERE department_id =
  (SELECT department_id
   FROM departments
   WHERE department_name = 'Shipping');
```

5 linhas excluídas

# DELETE de subconsulta

- O exemplo abaixo exclui as linhas de todos os funcionários que trabalham para um gerente que administra menos de 2 funcionários.

```
DELETE FROM copy_employees e
WHERE  e.manager_id IN
      (SELECT d.manager_id
       FROM employees d
       HAVING count (d.department_id) < 2
       GROUP BY d.manager_id);
```

# Erros de Constraint de Integridade

- As constraints de integridade garantem que os dados sigam um conjunto de regras necessárias.
- As constraints são verificadas automaticamente sempre que é executada uma instrução DML que poderia violar essas regras.
- Se qualquer regra for violada, a tabela não será atualizada e um erro será retornado.

# Erros de Constraint de Integridade

- Este exemplo viola uma constraint NOT NULL porque last\_name tem uma constraint de valor não nulo e o id = 999 não existe. Portanto, a subconsulta retorna um resultado nulo.

```
UPDATE copy_employees
SET last_name = (SELECT last_name
                  FROM copy_employees
                  WHERE employee_id = 999)
WHERE employee_id = 101;
```

```
ORA-01407: cannot update
("US_A009EMEA815_PLSQL_T01"."COPY_EMPLOYEES"."LAST_NAME") to NULL
```

# Erros de Constraint de Integridade

- Quando serão verificadas as constraints de chave primária e estrangeira?
- A tabela EMPLOYEES tem uma constraint de chave estrangeira em department\_id que referencia o department\_id da tabela DEPARTMENTS.
- Isso garante que todos os funcionários pertençam a um departamento válido.
- Na tabela DEPARTMENTS, os department\_id 10 e 20 existem, mas o 15 não.

# Erros de Constraint de Integridade

- Qual das seguintes instruções retornará um erro?

```
1.      UPDATE employees
        SET department_id = 15
        WHERE employee_id = 100;

2.      DELETE FROM departments
        WHERE department_id = 10;

3.      UPDATE employees
        SET department_id = 10
        WHERE    department_id = 20;
```



# Erros de Constraint de Integridade

- Quando estiver modificado suas cópias de tabelas (por exemplo, `copy_employees`), é possível que você veja erros de constraint de valor não nulo, mas não erros de constraint de chave primária e estrangeira.
- O motivo disso é que a instrução `CREATE TABLE .... AS (SELECT ...)` usada para criar a cópia da tabela copia as linhas e as constraints de valor não nulo, mas não as constraints de chave primária e estrangeira.
- Portanto, não existem constraints de chave primária e estrangeira nas tabelas copiadas.
- O acréscimo de constraints é abordado em outra lição.

# Cláusula FOR UPDATE em uma Instrução SELECT

- Quando uma instrução SELECT for executada para uma tabela de banco de dados, nenhum bloqueio será emitido, no banco de dados, nas linhas retornadas pela consulta que você está executando.
- Na maioria das vezes, você vai querer que o número de bloqueios executados no banco de dados seja mínimo.
- No entanto, às vezes, você quer garantir que ninguém atualize ou exclua os registros que a sua consulta está retornando enquanto estiver trabalhando com eles.

# Cláusula FOR UPDATE em uma Instrução SELECT

- É aí que a cláusula FOR UPDATE é usada.
- Assim que a sua consulta for executada, o banco de dados emitirá automaticamente bloqueios exclusivos em todas as linhas retornadas pela sua instrução SELECT, que será retida até que você execute um comando COMMIT ou ROLLBACK.
- Lembrete: a instância on-line/hospedada do APEX fará um commit automático e o bloqueio no nível da linha não ocorrerá.

# Cláusula FOR UPDATE em uma Instrução SELECT

- Se você usar uma cláusula FOR UPDATE em uma instrução SELECT com várias tabelas, todas as linhas de todas as tabelas serão bloqueadas.

```
SELECT e.employee_id, e.salary, d.department_name
FROM employees e JOIN departments d USING (department_id)
WHERE job_id = 'ST_CLERK' AND location_id = 1500
FOR UPDATE
ORDER BY e.employee_id;
```

# Cláusula FOR UPDATE em uma Instrução SELECT

- Estas quatro linhas foram bloqueadas pelo usuário que executou a instrução SELECT e ficarão assim até que ele execute um COMMIT ou ROLLBACK.

EMPLOYEE_ID	SALARY	DEPARTMENT_NAME
141	3500	Shipping
142	3100	Shipping
143	2600	Shipping
144	2500	Shipping

# Terminologia

Estes são os principais termos usados nesta lição:

- DELETE
- Constraint de integridade
- UPDATE
- UPDATE de subconsulta correlacionada
- DELETE de subconsulta correlacionada
- FOR UPDATE de SELECT

# Resumo

Nesta lição, você deverá ter aprendido a:

- Construir e executar uma instrução UPDATE
- Construir e executar uma instrução DELETE
- Construir e executar uma consulta que utiliza uma subconsulta para atualizar e excluir dados de uma tabela
- Construir e executar uma consulta que utiliza uma subconsulta correlacionada para atualizar e excluir dados de uma tabela

# Resumo

Nesta lição, você deverá ter aprendido a:

- Explicar como as constraints de integridade de chave estrangeira e primária afetam as instruções UPDATE e DELETE
- Explicar o propósito da Cláusula FOR UPDATE em uma instrução SELECT



