



Programação de Banco de Dados com SQL

14-3

Gerenciando Constraints



Objetivos

Esta lição abrange os seguintes objetivos:

- Listar as quatro funções diferentes que a instrução ALTER pode executar em constraints
- Gravar instruções ALTER TABLE para adicionar, eliminar, desativar e ativar constraints
- Dar o exemplo de uma função de negócios que exigiria que o DBA eliminasse, ativasse e/ou desativasse uma constraint ou usasse a sintaxe CASCADE
- Consultar USER_CONSTRAINTS no dicionário de dados e interpretar as informações retornadas

Finalidade

- Faria diferença se o ID de um novo aluno fosse incluído no banco de dados da escola se nenhum aluno tivesse se matriculado?
- Existe alguma possibilidade de que uma operadora de cartão de crédito emitisse o mesmo número de cartão de crédito para mais de uma pessoa ou que uma empresa contratasse um funcionário para um departamento que não existe?
- O que você acredita que aconteceria se uma empresa não pudesse confiar nas informações em seu banco de dados?

Finalidade

- Um sistema de banco de dados deve ser capaz de impor as regras de negócios e, ao mesmo tempo, impedir o acréscimo, a modificação ou a exclusão de dados que possam violar sua integridade referencial.
- Nesta seção, você aprenderá a fazer mudanças nas constraints da tabela de um modo que a integridade referencial e a confiabilidade do banco de dados sejam mantidas quando os dados precisarem ser modificados.

Gerenciando Constraints

- A instrução ALTER TABLE é usada para fazer mudanças nas constraints em tabelas existentes.
- Essas mudanças podem ser o acréscimo ou a eliminação de constraints, a ativação ou a desativação de constraints e a inclusão de uma constraint NOT NULL a uma coluna.



Gerenciando Constraints

- As diretrizes para fazer mudanças nas constraints são:
 - Você pode adicionar, eliminar, ativar ou desativar uma constraint, mas não modificar sua estrutura.
 - Você pode adicionar uma constraint NOT NULL a uma coluna existente usando a cláusula MODIFY da instrução ALTER TABLE.
 - MODIFY é usado porque NOT NULL é uma alteração nível da coluna.
 - Você poderá definir uma constraint NOT NULL somente se a tabela estiver vazia ou se a coluna contiver um valor para cada linha.

A instrução ALTER

- A instrução ALTER requer:
 - nome da tabela
 - nome da constraint
 - tipo da constraint
 - nome da coluna afetada pela constraint
- No exemplo de código mostrado abaixo, que usa a tabela de funcionários, a constraint de chave primária poderia ter sido adicionada após a criação original da tabela.

```
ALTER TABLE employees  
ADD CONSTRAINT emp_id_pk PRIMARY KEY (employee_id);
```


Adicionando Constraints

- Para adicionar uma constraint a uma tabela existente, use a seguinte sintaxe SQL:

```
ALTER TABLE nome_tabela  
ADD [CONSTRAINT nome_constraint] tipo da constraint (nome_coluna);
```

- Se a constraint for FOREIGN KEY, a palavra-chave REFERENCES deve ser incluída na instrução.
- Sintaxe:

```
ALTER TABLE nomedatabela  
ADD CONSTRAINT nome_constraint FOREIGN KEY(nome_coluna) REFERENCES  
nomedatabela(nome_coluna);
```

Exemplo de Acréscimo de Constraints

- Considere o banco de dados de funcionários.
- A chave primária da tabela DEPARTMENTS é incluída na tabela EMPLOYEES como chave estrangeira.

DEPARTMENTS - Pai

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Filho

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Exemplo de Acréscimo de Constraints

- O exemplo a seguir demonstra a sintaxe para adicionar a chave estrangeira à tabela EMPLOYEES:

```
ALTER TABLE employees
ADD CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
REFERENCES departments (department_id)
ON DELETE CASCADE;
```

DEPARTMENTS - Pai

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Filho

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Condições de Acréscimo de Constraints

- Se a constraint for NOT NULL, a instrução ALTER TABLE usa MODIFY, em vez de ADD.
- Constraints NOT NULL podem ser adicionadas somente se a tabela estiver vazia ou se a coluna contiver um valor para cada linha:

```
ALTER TABLE nome_tabela  
MODIFY (nome_coluna CONSTRAINT nome_constraint NOT NULL);
```

```
ALTER TABLE employees  
MODIFY (email CONSTRAINT emp_email_nn NOT NULL);
```



Porque Ativar e Desativar Constraints?

- Para impor as regras definidas pelas constraints de integridade, é preciso sempre ativá-las.
- No entanto, em certas situações, é preferível desativar temporariamente as constraints de integridade de uma tabela por questões de desempenho, como:
 - Quando se está carregando grandes quantidades de dados em uma tabela
 - Quando se está realizando operações em lote que façam muitas alterações em uma tabela (como mudar o número de todos os funcionários com o acréscimo do valor 1.000)

Eliminando Constraints

- Para eliminar uma constraint, você precisa saber o nome dela.
- Se não souber, poderá encontrá-lo em `USER_CONSTRAINTS` e `USER_CONS_COLUMNS`, no dicionário de dados.
- A opção `CASCADE` da cláusula `DROP` faz quaisquer constraints dependentes também serem eliminadas.
- Observe que, quando você elimina uma constraint de integridade, ela não é mais imposta pelo Servidor Oracle e não fica mais disponível no dicionário de dados.

Eliminando Constraints

- Quando você elimina uma constraint, nenhuma linha ou dado é excluído das tabelas afetadas.

```
ALTER TABLE nome_tabela  
DROP CONSTRAINT nome [CASCADE]
```

```
ALTER TABLE copy_departments  
DROP CONSTRAINT c_dept_dept_id_pk CASCADE;
```

Desativando Constraints

- Por padrão, sempre que uma constraint de integridade for definida em uma instrução CREATE ou ALTER TABLE, ela será ativada (imposta) automaticamente pelo Oracle, a não ser que seja criada especificamente em um estado desativado usando a cláusula DISABLE.



Desativando Constraints

- Você pode usar a opção DISABLE de ALTER TABLE para desativar uma constraint sem eliminá-la ou recriá-la.
- DISABLE permite a entrada de dados, estejam eles em conformidade com a constraint ou não.
- Essa função permite que os dados sejam adicionados a uma tabela filho sem que haja valores correspondentes na tabela pai.
- DISABLE simplesmente desabilita a constraint.

Usando a Cláusula DISABLE

- Você pode usar a cláusula DISABLE nas instruções ALTER TABLE e CREATE TABLE.

```
CREATE TABLE copy_employees
( employee_id NUMBER(6,0) PRIMARY KEY DISABLE,
  ...
  ...);

ALTER TABLE copy_employees
DISABLE CONSTRAINT c_emp_dept_id_fk;
```

- A desativação de uma constraint exclusiva ou de chave primária remove o índice exclusivo.

Usando a Cláusula CASCADE

- A cláusula CASCADE desativa constraints de integridade dependentes. Se, posteriormente, a constraint for ativada, as constraints dependentes não serão habilitadas automaticamente.
- Sintaxe e exemplo:

```
ALTER TABLE nome_tabela  
DISABLE CONSTRAINT nome_constraint [CASCADE];
```

```
ALTER TABLE copy_departments  
DISABLE CONSTRAINT c_dept_dept_id_pk CASCADE;
```

Ativando Constraints

- Para reativar uma constraint de integridade, use a cláusula ENABLE na instrução ALTER TABLE.
- ENABLE garante que todos os dados de entrada estejam em conformidade com a constraint.
- Sintaxe e exemplo:

```
ALTER TABLE nome_tabela  
ENABLE CONSTRAINT nome_constraint;
```

```
ALTER TABLE copy_departments  
ENABLE CONSTRAINT c_dept_dept_id_pk;
```

- Você pode usar a cláusula ENABLE nas instruções CREATE TABLE e ALTER TABLE.

Considerações sobre a Ativação de Constraints

- Se você ativa uma constraint, ela se aplica a todos os dados na tabela.
- Todos os dados na tabela devem estar de acordo com a constraint.
- Se você ativa uma constraint UNIQUE KEY ou PRIMARY KEY, um índice UNIQUE ou PRIMARY KEY é criado automaticamente.
- A ativação de uma constraint PRIMARY KEY que foi desativada com a opção CASCADE não habilita chaves estrangeiras que sejam dependentes da chave primária.
- ENABLE reativa a constraint após uma desativação.

Colocando Constraints em Cascata

- A colocação em cascata das constraints de integridade referencial permite que você defina as ações executadas pelo servidor do banco de dados quando o usuário tentar excluir ou atualizar uma chave para a qual apontam as chaves estrangeiras existentes.
- A cláusula CASCADE CONSTRAINTS é usada junto com a cláusula DROP COLUMN.
- Ela elimina todas as constraints de integridade referencial que se refiram às chaves primárias e exclusivas definidas nas colunas eliminadas.
- Ela também elimina todas as constraints multicoluna definidas nas colunas eliminadas.

Colocando Constraints em Cascata

- Se uma instrução ALTER TABLE não incluir a opção CASCADE CONSTRAINTS, qualquer tentativa de eliminar uma chave primária ou constraint multicoluna fracassará.
- Lembre-se: você não pode excluir um valor pai se valores filhos existirem em outras tabelas.

```
ALTER TABLE nome_tabela  
DROP(nome da coluna(s)) CASCADE CONSTRAINTS;
```

Quando CASCADE Não É Necessário

- Se todas as colunas referenciadas pelas constraints definidas nas colunas eliminadas também forem eliminadas, CASCADE CONSTRAINTS não será necessário.
- Por exemplo, supondo que nenhuma constraint referencial das outras tabelas se refira à coluna PK, é válido enviar a seguinte instrução sem a cláusula CASCADE CONSTRAINTS:

```
ALTER TABLE nomedatabela DROP  
(pk_nome_coluna(s));
```

- No entanto, se alguma constraint for referenciada pelas colunas de outras tabelas ou pelas colunas restantes na tabela de destino, você deverá especificar CASCADE CONSTRAINTS para evitar um erro.

Exibindo Constraints

- Após criar uma tabela, você pode confirmar a existência dela emitindo um comando DESCRIBE.
- NOT NULL é a única constraint que você pode verificar usando DESCRIBE.
- A constraint NOT NULL também aparecerá no dicionário de dados como uma constraint CHECK.

Exibindo Constraints

- Para exibir todas as constraints na sua tabela, consulte a tabela USER_CONSTRAINTS.

```
SELECT constraint_name, table_name, constraint_type, status
FROM USER_CONSTRAINTS
WHERE table_name = 'COPY_EMPLOYEES';
```

CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE	STATUS
COPY_EMP_PK	COPY_EMPLOYEES	P	ENABLED
CDEPT_DEPT_ID_FK	COPY_EMPLOYEES	R	ENABLED

Consulte USER_CONSTRAINTS

- Os tipos de constraint listados no Dicionário de Dados são:
 - P – PRIMARY KEY; R – REFERENCES (chave estrangeira);
 - C – constraint CHECK (incluindo NOT NULL);
 - U – UNIQUE.

CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE	STATUS
COPY_EMP_PK	COPY_EMPLOYEES	P	ENABLED
CDEPT_DEPT_ID_FK	COPY_EMPLOYEES	R	ENABLED

Terminologia

Estes são os principais termos usados nesta lição:

- ALTER TABLE
- Cláusula CASCADE
- Cláusula CASCADE CONSTRAINT
- DISABLE CONSTRAINT
- DROP COLUMN
- DROP CONSTRAINT
- ENABLE CONSTRAINT

Resumo

Nesta lição, você deverá ter aprendido a:

- Listar as quatro funções diferentes que a instrução ALTER pode executar em constraints
- Gravar instruções ALTER TABLE para adicionar, eliminar, desativar e ativar constraints
- Dar o exemplo de uma função de negócios que exigiria que o DBA eliminasse, ativasse e/ou desativasse uma constraint ou usasse a sintaxe CASCADE
- Consultar USER_CONSTRAINTS no dicionário de dados e interpretar as informações retornadas

