
ORACLE
Academy

Programação de Banco de Dados com SQL

14-2
Constraints PRIMARY KEY, FOREIGN KEY e CHECK

ORACLE Academy

Objetivos

Esta lição abrange os seguintes objetivos:

- Definir e dar um exemplo de uma constraint PRIMARY KEY, FOREIGN KEY e CHECK
- Explicar o propósito da definição de constraints PRIMARY KEY, FOREIGN KEY e CHECK
- Demonstrar a criação de constraints no nível da coluna e da tabela em uma instrução CREATE TABLE
- Avaliar um problema comercial que requeira o acréscimo de uma constraint PRIMARY KEY e FOREIGN KEY e gravar o código para executar a mudança

Finalidade

- Como discutido na última seção, as constraints são usadas para impedir a entrada de dados inválidos nas tabelas do banco de dados.
- O que aconteceria se, ilicitamente ou apenas por causa de um erro, a sua identificação pessoal exclusiva fosse dada para outra pessoa?
- E se amanhã, na escola, alguém recebesse as suas notas na prova ou conseguisse pegar um ônibus com o seu passe escolar?
- Garantir a integridade dos dados é a função das constraints. Afinal, você é único!

Constraints PRIMARY KEY

- Uma constraint PRIMARY KEY é uma regra que determina que os valores em uma coluna ou combinação de colunas devem identificar exclusivamente cada linha de uma tabela.
- Um valor de chave primária não pode aparecer em mais de uma linha da tabela.
- Para satisfazer uma constraint PRIMARY KEY, ambas as condições abaixo devem ser verdadeiras:
  - Uma coluna que faça parte da chave primária não pode conter um valor nulo.
  - Uma tabela pode ter apenas uma chave primária.

Constraints PRIMARY KEY

- As constraints PRIMARY KEY podem ser definidas no nível da coluna ou da tabela.
- No entanto, se for criada uma PRIMARY KEY composta, ela deverá ser definida no nível da tabela.
- Quando se define colunas PRIMARY KEY, recomenda-se usar o sufixo _pk no nome da constraint.
- Por exemplo, o nome da constraint para a coluna PRIMARY KEY chamada client_number na tabela chamada CLIENTS poderia ser clients_client_num_pk.

Constraints PRIMARY KEY

- Em uma instrução CREATE TABLE, é determinada a sintaxe da constraint PRIMARY KEY no nível da coluna:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
first_name VARCHAR2(14),
last_name VARCHAR2(13));
```

- Observe que o nível da coluna simplesmente se refere à área da instrução CREATE TABLE onde as colunas são definidas.
- O nível da tabela refere-se à última linha da instrução, abaixo da lista de nomes de colunas individuais.

Constraints PRIMARY KEY

- A sintaxe para criar a constraint PRIMARY KEY no nível da tabela é:

```
CREATE TABLE clients
(client_number NUMBER(4),
 first_name VARCHAR2(14),
 last_name VARCHAR2(13),
CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

- Observe que o nome da coluna PRIMARY KEY vem depois do tipo da constraint e está entre parênteses.

Constraints PRIMARY KEY

- Para definir uma PRIMARY KEY composta, você deve definir a constraint no nível da tabela, e não da coluna.
- Um exemplo da constraint de uma chave primária composta é mostrado abaixo:

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
CONSTRAINT copy_jhist_id_st_date_pk PRIMARY KEY(employee_id, start_date));
```

Constraints FOREIGN KEY (INTEGRIDADE REFERENCIAL)

- As constraints FOREIGN KEY também são chamadas de constraints de "integridade referencial".
- Esse tipo de constraint designa uma coluna ou combinação de colunas como uma chave estrangeira.
- Uma chave estrangeira está vinculada à chave primária (ou a uma chave exclusiva) de outra tabela, e esse elo é a base do relacionamento entre tabelas.

Exibindo uma Chave Estrangeira

- A tabela que contém a chave estrangeira é chamada de "filho" e a que contém a chave referenciada é chamada de "pai".

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

Exibindo uma Chave Estrangeira

- Nas tabelas mostradas, a chave primária da tabela DEPARTMENTS, department_id, também aparece na tabela EMPLOYEES como uma coluna da 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

Constraint de integridade referencial

- Para satisfazer uma constraint de integridade referencial, um valor de chave estrangeira deve ser equivalente a um valor existente na tabela pai ou ser nulo.

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

Constraint de integridade referencial

- Um valor de chave primária pode existir sem um valor de chave estrangeira correspondente. No entanto, uma chave estrangeira precisa ter uma chave primária correspondente.

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

Regra da Constraint de Integridade Referencial

- A regra é: antes de definir uma constraint de integridade referencial na tabela filho, a constraint UNIQUE ou PRIMARY KEY referenciada na tabela pai já deve estar definida.

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

Regra da Constraint de Integridade Referencial

- Em outras palavras, é preciso definir uma chave primária pai antes de criar uma chave estrangeira em uma tabela filho.

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

Constraint FOREIGN KEY

- Para definir uma constraint FOREIGN KEY, recomenda-se usar o sufixo _fk no nome da constraint.
- Por exemplo, o nome da constraint para a coluna FOREIGN KEY department_id na tabela de funcionários poderia ser nomeada emps_dept_id_fk.

Sintaxe da Constraint FOREIGN KEY

- A sintaxe para definir uma constraint FOREIGN KEY requer uma referência à tabela e à coluna na tabela pai.
- Uma constraint FOREIGN KEY em uma instrução CREATE TABLE pode se definida como aparece abaixo.
- Exemplo de sintaxe no nível da coluna:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0) CONSTRAINT c_emps_dept_id_fk
                           REFERENCES departments(department_id),
 email VARCHAR2(25));
```

Sintaxe da Constraint FOREIGN KEY

- A sintaxe para definir uma constraint FOREIGN KEY requer uma referência à tabela e à coluna na tabela pai.
- Uma constraint FOREIGN KEY em uma instrução CREATE TABLE pode se definida como aparece abaixo.
- Exemplo de sintaxe no nível da tabela:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT c_emps_dept_id_fk FOREIGN KEY (department_id)
                              REFERENCES departments(department_id));
```

ON DELETE CASCADE - Mantendo a Integridade Referencial

- O uso da opção ON DELETE CASCADE quando se define uma chave estrangeira permite às linhas dependentes na tabela filho serem excluídas quando uma linha na tabela pai é excluída.
- Se a chave estrangeira não tiver uma opção ON DELETE CASCADE, as linhas referenciadas na tabela pai não poderão ser excluídas.
- Em outras palavras, a constraint FOREIGN KEY da tabela filho inclui a permissão ON DELETE CASCADE, possibilitando que sua tabela pai exclua as linhas às quais se refere.

ON DELETE CASCADE - Mantendo a Integridade Referencial

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

ON DELETE CASCADE

- Se a coluna department_id em employees foi criada com a opção ON DELETE CASCADE especificada, a instrução DELETE emitida na tabela de departamentos será executada.
- Se a opção ON DELETE CASCADE não tiver sido especificada quando FOREIGN KEY foi criado, a tentativa de excluir um departamento da tabela de departamentos que possui entradas na tabela de funcionários fracassará.

Sintaxe de ON DELETE CASCADE

- Tabela criada sem ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
        REFERENCES copy_departments(department_id));
```

- A tentativa de excluir department_id 110 da tabela de departamentos fracassa, pois existem linhas dependentes na tabela de funcionários.

```
ORA-02292: integrity constraint (US_A009EMEA815_PLSQL_T01.CDEPT_DEPT_ID_FK)
violated - child record found
```

Sintaxe de ON DELETE CASCADE

- Tabela criada com ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
  REFERENCES copy_departments(department_id) ON DELETE CASCADE);
```

- A tentativa de excluir department_id 110 da tabela de departamentos é bem-sucedida, e as linhas dependentes na tabela de funcionários também são excluídas.
- 1 linha excluída.

ON DELETE SET NULL

- Em vez de excluir as linhas na tabela filho usando a opção ON DELETE CASCADE, as linhas filhos podem ser preenchidas com valores nulos usando a opção ON DELETE SET NULL.

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
  REFERENCES copy_departments(department_id) ON DELETE SET NULL);
```

ON DELETE SET NULL

- Isso pode ser útil quando o valor da tabela pai está sendo alterado para um novo número, como converter números de inventário para códigos de barra.
- Você não vai querer excluir as linhas na tabela filho.
- Quando os novos códigos de barra forem incluídos na tabela pai, passará a ser possível inseri-los na tabela filho sem precisar recriar totalmente cada linha da tabela filho.

Constraints CHECK

- A constraint CHECK define explicitamente uma condição que deve ser atendida.
- Para satisfazer a constraint, cada linha na tabela deve tornar a condição Verdadeira ou desconhecida (devido a um valor nulo).
- A condição de uma constraint CHECK pode se referir a qualquer coluna na tabela especificada, mas não a colunas de outras tabelas.

Exemplo da Constraint CHECK

- Esta constraint CHECK garante que o valor incluído para end_date seja posterior ao de start_date.

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 end_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
 CONSTRAINT cjhist_emp_id_st_date_pk
     PRIMARY KEY(employee_id, start_date),
 CONSTRAINT cjhist_end_ck CHECK (end_date > start_date));
```

- Como está referenciando duas colunas na tabela, essa constraint CHECK DEVE ser definida no nível da tabela.

Condições da Constraint CHECK

- CHECK deve estar apenas na linha em que a constraint é definida.
- Uma constraint CHECK não pode ser usada em consultas que referenciam valores em outras linhas.
- A constraint CHECK não pode conter chamadas às funções SYSDATE, UID, USER ou USERENV.
- A instrução CHECK(SYSDATE >'05-May-1999') não é permitida.

Condições da Constraint CHECK

- A constraint CHECK não pode usar as pseudocolunas CURRVAL, NEXTVAL, LEVEL ou ROWNUM.
- A instrução CHECK(NEXTVAL > 0) não é permitida.
- Uma coluna pode ter várias constraints CHECK que referenciem a coluna em sua definição.
- Não há limite para o número de constraints CHECK que você pode definir em uma coluna.

Sintaxe da Constraint CHECK

- As constraints CHECK podem ser definidas no nível da coluna ou da tabela.
- A sintaxe para definir uma constraint CHECK é:
  - Sintaxe no nível da coluna:

```
salary NUMBER(8,2) CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

  - Sintaxe no nível da tabela:

```
CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

Terminologia

Estes são os principais termos usados nesta lição:

- Constraint CHECK
- Constraint FOREIGN KEY
- REFERENCES
- NOT NULL
- ON DELETE CASCADE
- ON DELETE SET NULL
- Constraint PRIMARY KEY

Resumo

Nesta lição, você deverá ter aprendido a:

- Definir e dar um exemplo de uma constraint PRIMARY KEY, FOREIGN KEY e CHECK
- Explicar o propósito da definição de constraints PRIMARY KEY, FOREIGN KEY e CHECK
- Demonstrar a criação de constraints no nível da coluna e da tabela em uma instrução CREATE TABLE
- Avaliar um problema comercial que requeira o acréscimo de uma constraint PRIMARY KEY e FOREIGN KEY e gravar o código para executar a mudança


ORACLE
Academy