

ORACLE

Academy

Database Programming with SQL

14-1

Introdução a Constraints; Constraints NOT NULL e UNIQUE

ORACLE
Academy



Objetivos

- Esta lição abrange os seguintes objetivos:
 - Definir o termo "constraint" e sua relação com a integridade dos dados
 - Determinar quando é possível definir uma constraint no nível da coluna e no nível da tabela
 - Determinar por que é importante dar nomes significativos a constraints
 - Determinar quais regras de integridade de dados são impostas pelas constraints NOT NULL e UNIQUE

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Gravar uma instrução CREATE TABLE que inclua constraints NOT NULL e UNIQUE nos níveis da tabela e da coluna
 - Explicar como as constraints são criadas junto com a tabela

Finalidade

- Como a sociedade funcionaria sem regras?
- É uma regra parar no sinal vermelho?
- Seria seguro dirigir sem essa regra?
- Para os bancos de dados, é uma regra um valor de chave estrangeira não poder ser incluído sem que antes haja um valor de chave primária
- O que você acha que aconteceria no banco de dados se essa regra não fosse imposta?
- A confiabilidade de um banco de dados depende dos dados existentes nele

Finalidade

- Faria sentido haver valores salariais negativos ou seis alunos com o mesmo ID do aluno ou duas tabelas que não se referenciam mais?
- Sem regras, como você pode confiar na integridade do banco de dados?
- Nas próximas três lições, você estudará como criar as constraints que impõem as "regras"
- Você também aprenderá a gerenciá-las e verá definições de constraint no dicionário de dados
- As constraints são usadas para impedir a entrada de dados inválidos em tabelas

Visão Geral de Constraints

- Então, o que é uma constraint exatamente?
- Pense nelas como as regras do banco de dados
- Todas as definições de constraint estão armazenadas no dicionário de dados
- As constraints impedem a exclusão de uma tabela se houver dependências de outras tabelas
- As constraints impõem regras para os dados sempre que uma linha é inserida, atualizada ou excluída de uma tabela

Visão Geral de Constraints

- As constraints são importantes, e nomeá-las também
- Embora possa nomear uma constraint como "constraint_1" ou "constraint_2", você logo acharia difícil distingui-las e acabaria refazendo boa parte do trabalho

Criando Constraints

- Relembre a sintaxe SQL para criar uma tabela
- Na instrução CREATE TABLE mostrada no próximo slide, cada coluna e tipo de dados são definidos
- Você usa a instrução CREATE TABLE para estabelecer constraints para cada coluna na tabela
- Você pode especificar os detalhes de constraints em dois lugares diferentes da instrução CREATE TABLE:
 - No nível da coluna, próximo ao nome e tipo de dados
 - No nível da tabela, depois que todos os nomes de colunas forem listados

Criando Constraints

```
CREATE TABLE clients
(client_number NUMBER(4),
 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 no Nível da Coluna

- Uma constraint no nível da coluna referencia uma única coluna
- Para estabelecer uma constraint no nível da coluna, a constraint deve ser definida na instrução CREATE TABLE como parte da definição de coluna
- Examine a instrução SQL abaixo que estabelece uma constraint 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));
```

Constraints no Nível da Coluna

- CREATE TABLE clients

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

- O nome da constraint é clients_client_num_pk
- Ela impõe a regra de negócios de que client_number é a chave primária da tabela de clientes

Nomeando Constraints

- Toda constraint no banco de dados tem um nome. Quando uma constraint é criada, pode-se dar um nome a ela, como clients_client_num_pk, ou não. No segundo caso, o sistema é que dá o nome, como SYS_C00585417
- A convenção de nomenclatura pode ser a combinação do nome da tabela abreviado e um nome de coluna abreviado, seguido pela abreviação da constraint:
nome-tabela_nome-coluna_tipo-constrição
- Se a palavra reservada CONSTRAINT for usada na definição CREATE TABLE, você precisará dar um nome à constraint. Os nomes de constraints devem ter, no máximo, 30 caracteres

Nomeando Constraints no Nível da Coluna

- É mais indicado que você mesmo nomeie as constraints porque os nomes gerados pelo sistema não são fáceis de reconhecer
- Observe esta definição de tabela:

```
CREATE TABLE clients
(client_number      NUMBER(4) ,
 last_name         VARCHAR2(13) ,
 email            VARCHAR2(80)) ;
```

Nomeando Constraints no Nível da Coluna

- De acordo com a nossa convenção de nomenclatura:
 - Uma constraint de chave primária em client_number seria nomeada clients_client_number_pk
 - Uma constraint não nula em last_name seria nomeada clients_last_name_nn
 - Uma constraint exclusiva em email seria nomeada clients_email_uk

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
 last_name      VARCHAR2(13) CONSTRAINT clients_last_name_nn NOT NULL,
 email         VARCHAR2(80)  CONSTRAINT clients_email_uk UNIQUE);
```

Exemplo de Nomeação de Constraint

- Este exemplo mostra uma constraint nomeada pelo usuário e uma constraint nomeada pelo sistema:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
 last_name      VARCHAR2(13) NOT NULL,
 email          VARCHAR2(80));
```

Exemplo de Nomeação de Constraint

- Constraint nomeada pelo sistema:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
 last_name      VARCHAR2(13) NOT NULL,
 email          VARCHAR2(80));
```

- Duas constraints foram criadas:
 - Uma constraint nomeada pelo usuário, clients_client_num_pk, para impor a regra de que client_number é a chave primária
 - Uma constraint nomeada pelo sistema, SYS_Cn (onde n é um número inteiro exclusivo), para impor a regra de que last_names não podem ser nulos

Constraints no Nível da Tabela

- As constraints no nível da tabela são listadas separadamente das definições de coluna na instrução CREATE TABLE
- As definições de constraint no nível da tabela são listadas após a definição de todas as colunas da tabela
- No exemplo mostrado, a constraint exclusiva é listada por último na instrução CREATE TABLE

```
CREATE TABLE clients (
    client_number NUMBER(6) NOT NULL,
    first_name      VARCHAR2(20),
    last_name       VARCHAR2(20),
    phone           VARCHAR2(20),
    email           VARCHAR2(10) NOT NULL,
    CONSTRAINT clients_phone_email_uk UNIQUE (email,phone));
```

Regras Básicas para Constraints

- As constraints que se referem a mais de uma coluna (uma chave composta) devem ser definidas no nível da tabela
- A constraint NOT NULL pode ser especificada apenas no nível da coluna, e não no nível da tabela
- As constraints UNIQUE, PRIMARY KEY, FOREIGN KEY e CHECK podem ser definidas no nível da coluna ou da tabela
- Se a palavra CONSTRAINT for usada em uma instrução CREATE TABLE, você precisará dar um nome à constraint

Examine as Violações

```
CREATE TABLE clients(
    client_number      NUMBER(6),
    first_name         VARCHAR2(20),
    last_name          VARCHAR2(20),
    phone              VARCHAR2(20) CONSTRAINT phone_email_uk
                        UNIQUE(email,phone),
    email              VARCHAR2(10) CONSTRAINT NOT NULL,
CONSTRAINT emailclients_email NOT NULL,
CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

Examine as Violações

VIOLAÇÃO DE CHAVE EXCLUSIVA COMPOSTA
Chaves compostas devem ser definidas no nível da tabela.

```
CREATE TABLE clients(
    client_number      NUMBER(6) ,
    first_name         VARCHAR2(20) ,
    last_name          VARCHAR2(20) ,
    phone              VARCHAR2(20) CONSTRAINT phone_email_uk
                      UNIQUE(email,phone),
    email              VARCHAR2(10) CONSTRAINT NOT NULL,
    CONSTRAINT emailclients_email NOT NULL,
    CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number) );
```

VIOLAÇÃO DE NOT NULL
Constraints NOT NULL podem ser definidas apenas no nível da coluna.

VIOLAÇÃO DE NOME
O termo CONSTRAINT deve ser acompanhado pelo nome da constraint.

Cinco Tipos de Constraints

- Existem cinco tipos de constraints em um banco de dados Oracle
- Cada tipo impõe uma regra diferente
- Os tipos são:
 - Constraints NOT NULL
 - Constraints UNIQUE
 - Constraints PRIMARY KEY
 - Constraints FOREIGN KEY
 - Constraints CHECK

Cinco Tipos de Constraints

- Pelo restante desta lição, você aprenderá sobre constraints NOT NULL e UNIQUE
- A próxima lição ensinará os outros três tipos



Constraint NOT NULL

- Uma coluna definida com uma constraint NOT NULL requer que, para cada linha incluída na tabela, um valor deve existir para essa coluna
- Por exemplo, se a coluna de e-mails em uma tabela de funcionários for definida como NOT NULL, todo funcionário incluído na tabela DEVE ter um valor nessa coluna
- Quando se define colunas NOT NULL, é costume usar o sufixo _nn no nome da constraint
- Por exemplo, o nome da constraint para a coluna de e-mails NOT NULL na tabela de funcionários poderia ser emp_email_nn

Constraint UNIQUE

- Uma constraint UNIQUE requer que todo valor em uma coluna ou conjunto de colunas (uma chave composta) seja exclusivo, ou seja, não pode haver valores duplicados nas linhas da tabela
- Por exemplo, pode ser importante para uma empresa garantir que duas pessoas não tenham o mesmo endereço de e-mail
- A coluna de e-mails poderia ser definida usando a constraint UNIQUE
- A coluna ou o conjunto de colunas definido como UNIQUE é chamado de chave exclusiva

Constraint UNIQUE

- Se a combinação de duas ou mais colunas deve ser exclusiva a cada entrada, a constraint é considerada uma chave exclusiva composta
- A determinação de que todas as combinações de e-mails e sobrenomes deve ser exclusiva é um exemplo de uma chave exclusiva composta
- A palavra "chave" refere-se às colunas, e não aos nomes das constraints

Exemplo de Constraint Exclusiva

- Se a coluna de e-mails na tabela for definida com uma constraint UNIQUE, entradas de clientes diferentes não podem ter um e-mail idêntico
- E se dois clientes morarem na mesma casa e usarem o mesmo endereço de e-mail?

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net

```
INSERT INTO clients (client_number, first_name, last_name, phone,
email)
VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');
```

ORA-00001: unique constraint (constraint exclusiva)
(USWA_SKHS_SQL01_T01.CLIENT_EMAIL_UK) violated
(USWA_SKHS_SQL01_T01.CLIENT_EMAIL_UK) violado)

Definindo Constraints UNIQUE

- Quando se define constraints UNIQUE, é costume usar o sufixo _uk no nome da constraint
- Por exemplo, o nome da constraint para a coluna de e-mails UNIQUE na tabela de funcionários poderia ser emp_email_uk
- Para definir uma chave exclusiva composta, você deve definir a constraint no nível da tabela, e não da coluna
- Um exemplo do nome da constraint de uma chave exclusiva composta é:

```
CONSTRAINT clients_phone_email_uk UNIQUE(email,phone)
```

Chave Exclusiva Composta

- Constraints UNIQUE permitem a entrada de valores nulos, a não ser que a coluna também tenha uma constraint NOT NULL definida
- Um valor nulo em uma coluna (ou em todas as colunas de uma chave exclusiva composta) sempre satisfaz uma constraint UNIQUE porque valores nulos são considerados iguais a nada

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net

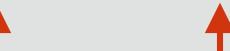


Esta combinação de colunas deve ser **exclusiva**

Chave Exclusiva Composta

- Para satisfazer uma constraint que designe uma chave exclusiva composta, duas linhas da tabela não podem ter a mesma combinação de valores nas colunas de chaves
- Além disso, qualquer linha que contenha valores nulos em todas as colunas de chaves satisfaz automaticamente a constraint

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net



Esta combinação de colunas deve ser **exclusiva**

Constraints Feitas na Criação da Tabela

- Quando você adicionar uma constraint NOT NULL como parte de uma instrução de criação de tabela, o banco de dados Oracle criará uma Constraint de Verificação no banco de dados para impor um valor na coluna NOT NULL
- Essa criação de constraint pode ser quase invisível para você quando cria a sua tabela. O Oracle simplesmente a faz
- Ao final da sua instrução de criação de tabela, a mensagem "Table created" (Tabela criada) é exibida, mas não são fornecidos detalhes sobre o número ou tipos de constraints que também foram feitos

Terminologia

- Estes são os principais termos usados nesta lição:
 - Constraint CHECK
 - Constraint
 - Constraint no nível da coluna
 - FOREIGN KEY
 - Constraints NOT NULL
 - PRIMARY KEY

Terminologia

- Estes são os principais termos usados nesta lição:
 - Constraint no nível da tabela
 - Constraints UNIQUE
 - CHAVE EXCLUSIVA

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Definir o termo "constraint" e sua relação com a integridade dos dados
 - Determinar quando é possível definir uma constraint no nível da coluna e no nível da tabela
 - Determinar por que é importante dar nomes significativos a constraints
 - Determinar quais regras de integridade de dados são impostas pelas constraints NOT NULL e UNIQUE

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Gravar uma instrução CREATE TABLE que inclua constraints NOT NULL e UNIQUE nos níveis da tabela e da coluna
 - Explicar como as constraints são criadas junto com a tabela

ORACLE

Academy