



Programación de bases de datos con SQL

14-3

Gestión de Restricciones



Objetivos

En esta lección se abordan los siguientes objetivos:

- Enumerar cuatro funciones diferentes que puede realizar la sentencia ALTER en las restricciones
- Escribir sentencias ALTER TABLE para agregar, borrar, desactivar y activar restricciones
- Nombrar una función de negocio que necesitaría un DBA para borrar, activar y/o desactivar una restricción o utilizar la sintaxis CASCADE
- Consultar USER_CONSTRAINTS en el diccionario de datos e interpretar la información devuelta

Objetivo

- ¿Supondría alguna diferencia que un nuevo número de identificación de alumno se introdujera en la base de datos de la escuela cuando no hubiera ningún alumno real inscrito?
- ¿Es probable que una compañía de tarjeta de crédito emita el mismo número de tarjeta de crédito a más de una cuenta o que una empresa contrate a un empleado para un departamento que no existe?
- ¿Qué predice que sucedería si una empresa no pudiera confiar en la fiabilidad de la información de su base de datos?

Objetivo

- Un sistema de base tiene que poder aplicar reglas de negocio y, al mismo tiempo, evitar la adición, la modificación o la supresión de datos que pueda dar como resultado una violación de la integridad referencial de la base de datos.
- En esta sección, aprenderá a realizar cambios en las restricciones de tabla para que la integridad referencial y, a su vez, la fiabilidad de la base de datos se mantengan cuando sea necesario cambiar los datos.

Gestión de Restricciones

- La sentencia ALTER TABLE se utiliza para realizar cambios en las restricciones de tablas existentes.
- Estos cambios pueden incluir agregar o borrar restricciones, activar o desactivar restricciones, así como agregar una restricción NOT NULL a una columna.



Gestión de Restricciones

- Las directrices para realizar cambios en las restricciones son:
 - Puede agregar, borrar, activar o desactivar una restricción, pero no puede modificar su estructura.
 - Puede agregar una restricción NOT NULL a una columna existente mediante la cláusula MODIFY de la sentencia ALTER TABLE.
 - MODIFY se utiliza porque NOT NULL es un cambio de nivel de columna.
 - Solo puede definir una restricción NOT NULL si la tabla está vacía o si la columna tiene un valor para cada fila.

Sentencia ALTER

- La sentencia ALTER necesita lo siguiente:
 - nombre de la tabla
 - nombre de la restricción
 - tipo de restricción
 - nombre de la columna a la que afecta la restricción
- En el ejemplo de código que se muestra a continuación, con la tabla de empleados, la restricción de clave principal se puede haber agregado después de que se creara originalmente la tabla.

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


Adición de Restricciones

- Para agregar una restricción a una tabla existente, utilice la siguiente sintaxis SQL:

```
ALTER TABLE table_name  
ADD [CONSTRAINT constraint_name] type of constraint  
(column_name);
```

- Si la restricción es una restricción FOREIGN KEY, la palabra clave REFERENCES se debe incluir en la sentencia.
- Sintaxis:

```
ALTER TABLE tablename  
ADD CONSTRAINT constraint_name FOREIGN KEY(column_name)  
REFERENCES  
tablename(column_name);
```

Ejemplo de Adición de Restricciones

- Considere la base de datos de empleados.
- La clave primaria de la tabla DEPARTMENTS se introduce en la tabla EMPLOYEES como una clave ajena.

DEPARTMENTS: principal

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

EMPLOYEE: secundaria

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

Ejemplo de Adición de Restricciones

- En el siguiente ejemplo se muestra la sintaxis para agregar esta clave ajena a la tabla EMPLOYEES:

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

DEPARTMENTS: principal

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

EMPLOYEE: secundaria

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

Condiciones para la Adición de Restricciones

- Si la restricción es una restricción NOT NULL, la sentencia ALTER TABLE utiliza MODIFY en lugar de ADD.
- Las restricciones NOT NULL solo se pueden agregar si la tabla está vacía o si la columna tiene un valor para cada fila:

```
ALTER TABLE table_name  
MODIFY (column_name CONSTRAINT constraint_name NOT NULL);
```

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

¿Por Qué Activar y Desactivar Restricciones?

- Para aplicar las reglas definidas por restricciones de integridad, las restricciones deben estar siempre activadas.
- En determinadas situaciones, se recomienda desactivar temporalmente las restricciones de integridad de una tabla por motivos de rendimiento, por ejemplo:
 - Al cargar grandes cantidades de datos en una tabla
 - Al ejecutar operaciones por lotes que realizan cambios masivos en una tabla (como puede ser cambiar el número de empleado de todas las personas agregando 1.000 al número existente)

Borrado de Restricciones

- Para borrar una restricción, debe saber el nombre de la restricción.
- Si no lo sabe, puede buscar el nombre de la restricción en `USER_CONSTRAINTS` y `USER_CONS_COLUMNS` en el diccionario de datos.
- La opción `CASCADE` de la cláusula `DROP` provoca que se borren también las restricciones dependientes.
- Tenga en cuenta que al borrar una restricción de integridad, Oracle Server ya no aplica esa restricción y deja de estar disponible en el diccionario de datos.

Borrado de Restricciones

- No se borra ninguna fila ni ningún dato en ninguna de las tablas afectadas al borrar una restricción.

```
ALTER TABLE table_name  
DROP CONSTRAINT name [CASCADE]
```

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

Desactivación de Restricciones

- Por defecto, siempre que una restricción de integridad está definida en una sentencia CREATE o ALTER TABLE, Oracle activa (aplica) automáticamente la restricción, a menos que se cree específicamente con un estado desactivado con la cláusula DISABLE.



Desactivación de Restricciones

- Puede desactivar una restricción sin borrarla o volver a crearla mediante DISABLE de la opción ALTER TABLE.
- DISABLE permite datos entrantes, tanto si se ajustan a la restricción como si no.
- Esta función permite agregar datos una tabla secundaria sin tener los valores correspondientes en la tabla principal.
- DISABLE simplemente desactiva la restricción.

Uso de la Cláusula DISABLE

- Puede utilizar la cláusula DISABLE tanto en la sentencia ALTER TABLE como en la sentencia 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;
```

- La desactivación de una restricción UNIQUE o PRIMARY KEY elimina el índice único.

Uso de la Cláusula CASCADE

- La cláusula CASCADE desactiva las restricciones de integridad dependientes. Si la restricción se activa posteriormente, las restricciones dependientes no se activan automáticamente.
- Sintaxis y ejemplo:

```
ALTER TABLE table_name  
DISABLE CONSTRAINT constraint_name [CASCADE];
```

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

Activación de Restricciones

- Para activar una restricción de integridad actualmente desactivada, utilice la cláusula ENABLE en la sentencia ALTER TABLE.
- ENABLE garantiza que todos los datos entrantes se ajustan a la restricción.
- Sintaxis y ejemplo:

```
ALTER TABLE table_name  
ENABLE CONSTRAINT constraint_name;
```

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

- Puede utilizar la cláusula ENABLE tanto en la sentencia CREATE TABLE como en la sentencia ALTER TABLE.

Consideraciones sobre la Activación de Restricciones

- Si activa una restricción, se aplica a todos los datos de la tabla.
- Todos los datos de la tabla deben cumplir la restricción.
- Si activa una CLAVE UNIQUE o una restricción PRIMARY KEY, se crea un índice UNIQUE o PRIMARY KEY automáticamente.
- La activación de una restricción PRIMARY KEY desactivada con la opción CASCADE no activa ninguna clave ajena dependiente de la clave primaria.
- ENABLE vuelve a activar la restricción después de desactivarla.

Restricciones en Cascada

- Las restricciones de integridad referencial en cascada definen las acciones que lleva a cabo el servidor de base de datos cuando un usuario intenta suprimir o actualizar una clave a la que apuntan las claves ajenas existentes
- La cláusula CASCADE CONSTRAINTS se utiliza junto con la cláusula DROP COLUMN.
- Borra todas las restricciones de integridad referencial que hacen referencia a las claves primarias y únicas definidas en las columnas borradas.
- Borra también todas las restricciones de varias columnas definidas en las columnas borradas.

Restricciones en Cascada

- Si una sentencia ALTER TABLE no incluye la opción CASCADE CONSTRAINTS, cualquier intento de borrar una restricción de clave primaria o varias columnas fallará.
- Recuerde que no puede suprimir un valor principal si existen valores secundarios en otras tablas.

```
ALTER TABLE table_name  
DROP(column name(s)) CASCADE CONSTRAINTS;
```

Cuando No Es Necesario CASCADE

- Si las columnas a las que hacen referencia las restricciones definidas en las columnas borradas también se borran, CASCADE CONSTRAINTS no es necesario.
- Por ejemplo, si ninguna restricción referencial de otras tablas hace referencia a la columna PK, es válido ejecutar la siguiente sentencia sin la cláusula CASCADE CONSTRAINTS:

```
ALTER TABLE tablename DROP  
(pk_column_name(s));
```

- Sin embargo, si las columnas de otras tablas o las columnas que quedan en la tabla de destino hacen referencia a las restricciones, se debe especificar CASCADE CONSTRAINTS.

Visualización de Restricciones

- Después de crear una tabla, puede confirmar su existencia emitiendo un comando DESCRIBE.
- La única restricción que se puede verificar con el comando DESCRIBE es la restricción NOT NULL.
- La restricción NOT NULL también aparece en el diccionario de datos como una restricción CHECK.

Visualización de Restricciones

- Para ver todas las restricciones de la tabla, consulte la tabla 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	D	ENABLED

Consulta USER_CONSTRAINTS

- Los tipos de restricción que aparecen en el diccionario de datos son:
 - P – PRIMARY KEY; R – REFERENCES (clave ajena);
 - C – restricción CHECK (incluida NOT NULL);
 - U – UNIQUE.

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

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

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

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Enumerar cuatro funciones diferentes que puede realizar la sentencia ALTER en las restricciones
- Escribir sentencias ALTER TABLE para agregar, borrar, desactivar y activar restricciones
- Nombrar una función de negocio que necesitaría un DBA para borrar, activar y/o desactivar una restricción o utilizar la sintaxis CASCADE
- Consultar USER_CONSTRAINTS en el diccionario de datos e interpretar la información devuelta

