



# Programación de bases de datos con SQL

14-2

Restricciones PRIMARY KEY, FOREIGN KEY y CHECK



# Objetivos

En esta lección se abordan los siguientes objetivos:

- Definir y dar un ejemplo de una restricción PRIMARY KEY, FOREIGN KEY y CHECK
- Explicar el objetivo de definir las restricciones PRIMARY KEY, FOREIGN KEY y CHECK
- Demostrar la creación de las restricciones a nivel de columna y a nivel de tabla en una sentencia CREATE TABLE
- Evaluar un problema de negocio que necesita la adición de una restricción PRIMARY KEY y FOREIGN KEY y escribir el código para ejecutar el cambio



# Objetivo

- Como se ha explicado en la última sección, las restricciones se utilizan para evitar la entrada de datos no válidos en las tablas de la base de datos.
- ¿Qué sucedería si, a escondidas o solo por un error negligente, su identificación única personal se proporcionara a otra persona?
- ¿Qué sucedería si mañana en la escuela se atribuyeran a otra persona sus clases de graduación o esta pudiera almorzar con su número de tarjeta para el almuerzo?
- Garantizar la integridad de datos es de lo que tratan las. Después de todo, cada uno es único.



# Restricciones PRIMARY KEY

- Una restricción PRIMARY KEY es una regla de que los valores de una columna o una combinación de columnas deben identificar de forma única cada fila de una tabla.
- No puede aparecer ningún valor de clave primaria en más de una fila de la tabla.
- Para cumplir una restricción PRIMARY KEY, las dos condiciones siguientes deben ser verdaderas:
  - Ninguna columna que forme parte de la clave primaria puede contener un valor nulo.
  - Una tabla solo puede tener una clave primaria.

# Restricciones PRIMARY KEY

- Las restricciones PRIMARY KEY se pueden definir en nivel de columna o de tabla.
- Sin embargo, si se crea una CLAVE PRIMARIA compuesta, se debe definir en el nivel de tabla.
- Al definir columnas PRIMARY KEY, es una buena práctica utilizar el sufijo `_pk` en el nombre de restricción.
- Por ejemplo, el nombre de restricción para la columna PRIMARY KEY denominada `client_number` de la tabla llamada `CLIENTS` podría ser `clients_client_num_pk`.

# Restricciones PRIMARY KEY

- En una sentencia CREATE TABLE, la sintaxis de restricción PRIMARY KEY en el nivel de columna se indica de la siguiente forma:

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

- Tenga en cuenta que el nivel de columna simplemente se refiere al área de la sentencia CREATE TABLE en el que las columnas están definidas.
- El nivel de tabla hace referencia a la última línea de la sentencia debajo de la lista de nombres de columna individuales.

# Restricciones PRIMARY KEY

- Para crear la restricción PRIMARY KEY en el nivel de tabla, la sintaxis es:

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

- Tenga en cuenta que el nombre de la columna PRIMARY KEY sigue el tipo de restricción y está entre paréntesis.



# Restricciones PRIMARY KEY

- Para definir una CLAVE PRIMARIA compuesta, debe definir la restricción en el nivel de tabla en lugar del nivel de columna.
- A continuación se muestra un ejemplo de restricción de clave primaria compuesta:

```
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));
```

# FOREIGN KEY (INTEGRIDAD REFERENCIAL)

## Restricciones

- Las restricciones FOREIGN KEY también se denominan restricciones de "integridad referencial".
- Las restricciones Foreign Key designan una columna o una combinación de columnas como una clave ajena.
- Un clave ajena enlaza a la clave primaria (o una clave única) en otra tabla y este enlace es la base de la relación entre las tablas.

# Visualización de una Clave Ajena

- La tabla que contiene la clave ajena se denomina tabla "secundaria" y la tabla que contiene la clave de referencia se denomina tabla "principal".

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

# Visualización de una Clave Ajena

- En las tablas mostradas, la clave primaria de la tabla DEPARTMENTS, department\_id, también aparece en la tabla EMPLOYEES como una columna de 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

# Restricción de Integridad Referencial

- Para cumplir una restricción de integridad referencial, el valor de clave ajena debe coincidir con un valor existente de la tabla principal o ser un valor NULL.

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

# Restricción de Integridad Referencial

- Puede existir un valor de clave primaria sin un valor de clave ajena correspondiente; sin embargo, una clave ajena debe tener una clave primaria correspondiente.

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

# Regla de Restricción de Integridad Referencial

- La regla es: antes de definir una restricción de integridad referencial en la tabla secundaria, ya debe estar definida la restricción UNIQUE o PRIMARY a la que se hace referencia.

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

# Regla de Restricción de Integridad Referencial

- Es decir, primero debe tener una clave primaria principal definida para que pueda crear una clave ajena en una tabla secundaria.

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



# Restricción FOREIGN KEY

- Para definir una restricción FOREIGN KEY, es una buena práctica utilizar el sufijo `_fk` en el nombre de restricción.
- Por ejemplo, el nombre de restricción para la columna de CLAVE AJENA `department_id` de la tabla de empleados podría ser `emps_dept_id_fk`.

# Sintaxis de Restricción FOREIGN KEY

- La sintaxis para definir una restricción FOREIGN KEY necesita una referencia a la tabla y columna de la tabla principal.
- Una restricción FOREIGN KEY en una sentencia CREATE TABLE se puede definir de la siguiente forma.
- Ejemplo de sintaxis de nivel de columna:

```
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));
```

# Sintaxis de Restricción FOREIGN KEY

- La sintaxis para definir una restricción FOREIGN KEY necesita una referencia a la tabla y columna de la tabla principal.
- Una restricción FOREIGN KEY en una sentencia CREATE TABLE se puede definir de la siguiente forma.
- Ejemplo de sintaxis de nivel de tabla:

```
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: Mantenimiento de Integridad Referencial

- El uso de la opción ON DELETE CASCADE al definir una clave ajena permite la supresión de las filas dependientes en la tabla secundaria cuando se suprime una fila de la tabla principal.
- Si la clave ajena no tiene una opción ON DELETE CASCADE, las filas a las que se hace referencia de la tabla principal no se pueden suprimir.
- Es decir, la restricción FOREIGN KEY de la tabla secundaria incluye el permiso ON DELETE CASCADE que permite a su principal suprimir las filas a las que se hace referencia.

# ON DELETE CASCADE: Mantenimiento de Integridad Referencial

## 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

# ON DELETE CASCADE

- Si la columna `department_id` de empleados se ha creado con la opción `ON DELETE CASCADE` especificada, se ejecutará la sentencia `DELETE` emitida en la tabla de departamentos.
- Si no se ha especificado la opción `ON DELETE CASCADE` al crear `FOREIGN KEY`, fallará el intento de supresión de un departamento de la tabla de departamentos que tiene entradas en la tabla de empleados.

# Sintaxis de ON DELETE CASCADE

- Tabla creada sin 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));
```

- Fallará un intento de supresión de department\_id 110 de la tabla de departamentos ya que hay filas dependientes en la tabla de empleados.

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

# Sintaxis de ON DELETE CASCADE

- Tabla creada con 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);
```

- El intento de supresión de department\_id 110 de la tabla de departamentos se realizará correctamente y las filas dependientes en la tabla de empleados también se suprimirán.
- 1 fila suprimida.



# ON DELETE SET NULL

- En lugar de que se supriman las filas de la tabla secundaria al utilizar una opción ON DELETE CASCADE, las filas secundarias se pueden rellenar con valores nulos mediante la opción 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

- Esto podría resultar útil cuando el valor de la tabla principal se va a cambiar a un nuevo número como, por ejemplo, al convertir números de inventario a números de código de barras.
- No desea suprimir las filas de la tabla secundaria.
- Cuando se introduzcan los nuevos números de código de barras en la tabla principal, se podrán insertar en la tabla secundaria sin tener que volver a crear totalmente cada una de las filas de la tabla secundaria.

# Restricciones CHECK

- La restricción CHECK define explícitamente una condición que se debe cumplir.
- Para cumplir la restricción, cada una de las filas de la tabla debe hacer que condición sea True o desconocida (debido a un valor nulo).
- La condición de una restricción de CHECK puede hacer referencia a cualquier columna de la tabla especificada, pero no a columnas de otras tablas.

# Ejemplo de Restricción CHECK

- Esta restricción CHECK garantiza que un valor introducido para end\_date sea posterior a 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 esta RESTRICCIÓN CHECK hace referencia a dos columnas de la tabla, se DEBE definir en el nivel de tabla.

# Condiciones de la Restricción CHECK

- Una restricción CHECK solo debe estar en la fila en la que se define la restricción.
- Una restricción CHECK no se puede utilizar en consultas que hacen referencia a valores de otras filas.
- La restricción CHECK no puede contener llamadas a las funciones SYSDATE, UID, USER o USERENV.
- La sentencia CHECK(SYSDATE >'05-May-1999') no está permitida.

# Condiciones de la Restricción CHECK

- La restricción USER no puede utilizar las pseudocolumnas CURRVAL, NEXTVAL, LEVEL o ROWNUM.
- La sentencia CHECK(NEXTVAL > 0) no está permitida.
- Una sola columna puede tener varias restricciones CHECK que hagan referencia a la columna en su definición.
- No hay ningún límite en cuanto al número de restricciones CHECK que puede definir en una columna.

# Sintaxis de Restricción CHECK

- Las restricciones CHECK se pueden definir en el nivel de tabla o de columna.
- La sintaxis para definir una restricción CHECK es la siguiente:

– Sintaxis de nivel de columna:

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

– Sintaxis de nivel de tabla:

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

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

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

**ORACLE®**

Academy

DPS14L2  
Restricciones PRIMARY KEY,  
FOREIGN KEY y CHECK

Copyright © 2019, Oracle y/o sus filiales. Todos los derechos reservados.

32



# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Definir y dar un ejemplo de una restricción PRIMARY KEY, FOREIGN KEY y CHECK
- Explicar el objetivo de definir las restricciones PRIMARY KEY, FOREIGN KEY y CHECK
- Demostrar la creación de las restricciones a nivel de columna y a nivel de tabla en una sentencia CREATE TABLE
- Evaluar un problema de negocio que necesita la adición de una restricción PRIMARY KEY y FOREIGN KEY y escribir el código para ejecutar el cambio

