

Listado de conjuntos de sentencias LDD del SQL de Oracle, denominados *guiones*, definidas para ser ejecutadas una a una, secuencialmente, e ir ilustrando su sintaxis.

En los guiones aparecen tanto instrucciones LDD como consultas al diccionario de datos, que permiten visualizar los resultados causados por las operaciones llevadas a cabo.

## 00

```
CREATE TABLE Cero (Textode20_fijo char (20), -- Columna de texto de tamaño fijo (Max. 2000 caracteres)
    Textode1 char, -- El tamaño por defecto es 1
    Textode20_variable varchar2(20), -- Columna de texto de tamaño variable (Max. 4000 caracteres)
    NumericoEnterode5 number(5), -- Número entero de tamaño 5 dígitos (Max. 38 dígitos)
    NumericoComaFija number(4,2), -- Número en coma fija de tamaño 4 dígitos y 2 de ellos decimales
    NumericoComaFlotante3 number, -- Número en coma flotante
    FechaHora date); -- Columna de tipo fecha y hora
```

/\*

Ni en Oracle 10g ni en 11g existe un tipo de datos adecuado para los *identificadores artificiales*. En su lugar existe un tipo de objeto llamado *secuencia*. Dicho tipo de objeto proporciona en sucesivas llamadas valores enteros, siguiendo una secuencia previamente definida por un valor de inicio y un incremento. Los valores obtenidos de las llamadas realizadas se pueden insertar en sucesivas filas en una columna numérica que actúe como identificador artificial, entre otras utilidades.

\*/

```
CREATE SEQUENCE id_sec; -- Si no se especifica nada sobre la secuencia, por defecto empieza en 1 y el incremento es 1.
```

/\*

*dual* es una tabla creada automáticamente por Oracle junto con el *diccionario*. Está en el esquema sys, pero es accesible con el nombre *dual* a todos los usuarios. Tiene una columna llamada *dummy*, de tipo *varchar2 (1)*, y contiene una única fila con el valor 'X'. Seleccionar de la tabla *dual* es útil para calcular una expresión de la cláusula SELECT. Dado que la tabla *dual* tiene una única fila, el valor de dicha expresión se devuelve sólo una vez. Alternativamente, se podría seleccionar de cualquier otra tabla, pero el valor de la expresión aparecería tantas veces como filas hubiera en la tabla.

\*/

```
select 'Hola mundo' from dual; -- Escribir una constante, textual en este caso.
```

```
select 5*7 from dual; -- La tabla dual como calculadora.
```

```
select id_sec.nextval from dual; -- Ahora se usa como medio de mostrar el primer valor de la secuencia id_sec recién creada.
```

/\*

La expresión *id\_sec.nextval* proporciona cada vez que se invoca el siguiente valor de la secuencia *id\_sec*. Ejecútense sucesivas veces la última sentencia select para comprobarlo.

\*/

/\*

Antes de empezar con el resto de guiones, será útil saber que una restricción semántica puede declararse en la misma línea en la que se define una columna (declaración *a nivel de columna* o bien declaración *en línea*), o puede declararse, de la misma forma en que se hace en la sintaxis pseudo-SQL, después de declarar todas las columnas de la tabla (declaración *a nivel de tabla* o bien declaración *fuera de línea*); sin que esa variación afecte a su significado.

En cualquiera de los dos formatos de declaración, a una restricción se le puede asignar un nombre explícitamente, haciendo uso de la palabra clave *constraint* seguida del nombre elegido. De cualquier forma, toda restricción semántica tiene nombre. Si no se le asigna nombre explícitamente, tendrá un nombre implícitamente asignado por el sistema.

\*/

## 11

```
CREATE TABLE Uno (Dni varchar2 (10) CONSTRAINT RestriccionSemantica1 PRIMARY KEY,
    Nombre varchar2 (25),
    Apellido varchar2 (25),
    FechaDeNacimiento date);
```

/\*

A continuación se consultan 3 de las tablas del diccionario de datos:  
la primera (y en cierto modo la segunda) contiene una fila por cada tabla del esquema del usuario, la tercera muestra una fila por cada objeto (tablas, índices, vistas, ...) del esquema del usuario.

\*/

```
-- user_tables contiene mucha más información aparte del nombre de cada tabla
```

```
describe user_tables
```

```
-- Sólo se muestra el nombre de las tablas
```

```
select table_name from user_tables;
```

```
-- user_catalog muestra el nombre de las tablas y vistas del esquema del usuario
```

```
-- (y algún que otro tipo de objeto que se estudiará más adelante)
```

```
-- (cat es un sinónimo de user_catalog)
```

```
select * from user_catalog;
```

```

-- user objects también contiene muchos más datos aparte de los mostrados

select object_name, object_type from user_objects;

-- Si presenta problemas por el tamaño del campo object_name se puede solucionar
-- con un formato de columna

column object_name format a25

select object_name, object_type from user_objects;

column object_name clear

/*
  Ahora vamos a consultar otra tabla del diccionario a fin de comprobar la
  existencia de las restricciones previstas.
  La columna C corresponde en realidad a constraint_type, pero la C es todo
  lo que se logra ver de su cabecera dado que su tipo de datos es de 1 carácter.
  El valor P en dicha columna indica que la restricción es del tipo clave primaria.
*/

select constraint_name, constraint_type, table_name
from user_constraints;

/*
  Si se desea consultar los campos de la tabla que están implicados en la restricción:
*/

column column_name format a30
select constraint_name, column_name
from user_cons_columns;
column column_name cle

/*
  Para ver los índices que se han generado (implícitamente) en la creación
  de la tabla uno:
*/

select index_name, index_type, table_name, uniqueness
from user_indexes;

/*
  Para consultar los campos afectados por los índices:
*/

column index_name format a21
column table_name format a10
column column_name format a12
column column_position format 99999
select *
from user_ind_columns;
clear column

```

## 12

CREATE INDEX IndiceApellidoNombre ON Uno (Apellido DESC, Nombre);

```

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

```

```

column index_name format a21
column table_name format a10
column column_name format a12
column column_position format 99999
select *
from user_ind_columns;
clear column

```

```

/*
  El apellido descendente causa que el tipo del índice sea 'function-based normal'
  en vez de 'normal'. Por ello, al consultar los campos afectados por el índice

```

en vez de 'apellido' y 'nombre', aparecen un nombre de sistema (SYS\_NC00005\$) y 'nombre'. La tabla del diccionario user\_ind\_expressions indica que SYS\_NC00005\$ corresponde a una expresión sobre el campo 'apellido'.

\*/

```
column index_name format a21
column table_name format a10
column column_expression format a25
select *
from user_ind_expressions;
cle col
```

```
select constraint_name, constraint_type, table_name
from user_constraints;
```

## 14

```
DROP INDEX indiceapellidoonombre;
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

## 17

```
DROP INDEX restriccionsemantica1;
```

## 19

```
drop table uno;  
select * from cat;
```

```
/*
```

Como el administrador de la base de datos tiene la papelera de reciclaje activada, la tabla borrada aparece en el diccionario. Podemos eliminar a voluntad el contenido de la papelera:

```
*/
```

```
purge recyclebin;  
select * from cat;
```

```
CREATE TABLE Uno (Dni varchar2 (10) CONSTRAINT RestriccionSemantica1 PRIMARY KEY,  
Nombre varchar2 (25),  
Apellido varchar2 (25) DEFAULT 'Pérez',  
FechaDeNacimiento DATE DEFAULT sysdate-1);
```

## 201

```
-- Los nombres de restricciones han de ser únicos
```

```
CREATE TABLE Dos (Dni varchar2(10) CONSTRAINT RestriccionSemantica1 PRIMARY KEY,  
Nombre varchar2(25),  
Apellido varchar2(25),  
FechaDeNacimiento DATE NOT NULL,  
CONSTRAINT RestriccionSemantica2 UNIQUE (Nombre, Apellido) );
```

## 202

```
CREATE TABLE Dos (Dni varchar2(10) CONSTRAINT RestriccionSemantica12 PRIMARY KEY,  
Nombre varchar2(25),  
Apellido varchar2(25),  
FechaDeNacimiento DATE NOT NULL,  
CONSTRAINT RestriccionSemantica2 UNIQUE (Nombre, Apellido) );
```

```
describe dos
```

```
column object_name format a25  
select object_name, object_type from user_objects;  
column object_name clear
```

```
/*
```

El valor U en constraint\_type indica una restricción de clave alternativa  
y el valor C indica una restricción de chequeo (de no nulidad en este caso)

```
*/
```

```
column constraint_name format a25  
column table_name format a10  
column search_condition format a35  
select constraint_name, constraint_type, table_name, search_condition  
from user_constraints;  
cle col
```

```
column column_name format a30  
select constraint_name, column_name, position  
from user_cons_columns;  
column column_name cle
```

```
column index_name format a25  
column table_name format a11  
select index_name, index_type, table_name, uniqueness  
from user_indexes;  
cle col
```

```
column index_name format a22  
column table_name format a10  
column column_name format a12  
column column_position format 99999  
set line 81  
select *
```

```
from user_ind_columns;
set line 80
clear column
```

## 203

```
/*
La opción purge en el borrado evita el paso por la papelera de reciclaje
*/
```

```
drop table dos purge;
select * from cat;
```

```
CREATE TABLE Dos (Dni varchar2(10) PRIMARY KEY,
Nombre varchar2(25),
Apellido varchar2(25),
FechaDeNacimiento DATE NOT NULL,
UNIQUE (Nombre, Apellido) );
```

```
describe dos
```

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

## 204

```
drop table dos purge;
```

```
CREATE TABLE Dos (Dni varchar2(10) CONSTRAINT RestriccionSemantica12 PRIMARY KEY,
Nombre varchar2(25),
Apellido varchar2(25),
FechaDeNacimiento DATE NOT NULL,
CONSTRAINT RestriccionSemantica2 UNIQUE (Nombre, Apellido) );
```

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

## 21

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

```

```

DROP INDEX restriccionsemantica2;

```

## 22

```

/*
  Ahora se intenta simular con un índice una clave alternativa sin declarar la
  restricción correspondiente. El primer intento falla por duplicidad del nombre
  del índice. El segundo tiene éxito, pero esta es una maniobra completamente
  DESACONSEJADA, dado que el sistema mantiene una limitación sin conocimiento de
  la restricción correspondiente.
*/

```

```

CREATE UNIQUE INDEX RestriccionSemantica2 ON dos (nombre, fechadenacimiento);

```

```

CREATE UNIQUE INDEX RestriccionSemantica22 ON dos (nombre, fechadenacimiento);

```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

```

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

```

```

/*
  Eliminamos la limitación no declarada
*/

```

```

drop index restriccionsemantica22;

```

## 30

```

-- Falla el primer intento de creación porque no se puede declarar una restricción NOT NULL
-- a nivel de tabla.

```

```

CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
                    Nombre varchar2(25),
                    Apellido varchar2(25),
                    FechaDeNacimiento DATE,
                    CONSTRAINT RestriccionSemantica3 NOT NULL (FechaDeNacimiento),
                    CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );

```

```

-- A nivel de campo ya no plantea problemas

```

```

CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
                    Nombre varchar2(25),
                    Apellido varchar2(25),
                    FechaDeNacimiento DATE CONSTRAINT RestriccionSemantica3 NOT NULL,
                    CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );

```

```

describe tres

```

```

column object_name format a25
select object_name, object_type from user_objects;
column object_name clear

```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

```

column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column

```

## 332

```
drop table tres purge;
```

```

/*
  En realidad las restricciones NOT NULL constituyen un caso particular de las
  restricciones CHECK.
*/

```

```

CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
Nombre varchar2(25) CONSTRAINT RestriccionSemantica4 CHECK (nombre is not null),
Apellido varchar2(25),
FechaDeNacimiento DATE CONSTRAINT RestriccionSemantica3 NOT NULL,
CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );

```

```
describe tres
```

```

column object_name format a25
select object_name, object_type from user_objects;
column object_name clear

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column

```

## 334

```
drop table tres purge;
```

```
/*
```



Las restricciones CHECK pueden llevar nombre explícito o implícito. Cada columna puede soportar un número arbitrario de restricciones CHECK, especificadas a nivel de campo y/o de tabla. Obsérvese el uso de la última para garantizar un NOT NULL sobre un campo compuesto.

\*/

```
CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
    Nombre varchar2(25) CHECK (nombre is not null)
    CONSTRAINT RestriccionSemantica4 CHECK (nombre in ('Yeray','Aday','Romén')),
    Apellido varchar2(25),
    FechaDeNacimiento DATE,
    CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido),
    CONSTRAINT RestriccionSemantica5 CHECK (Nombre<>Apellido),
    CHECK (length(Nombre)=length(Apellido)),
    CHECK (not(Apellido is null and FechaDeNacimiento is null)) );
```

describe tres

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

```
column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column
```

## 336

```
drop table tres purge;
```

```
-- Restauramos la tabla tres a su estado original
```

```
CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
    Nombre varchar2(25),
    Apellido varchar2(25),
    FechaDeNacimiento DATE CONSTRAINT RestriccionSemantica3 NOT NULL,
    CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );
```

describe tres

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```

column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle

column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col

column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column

```

## 34

```

/*
Existe una sintaxis alternativa:
ALTER TABLE tres DROP (FechaDeNacimiento);
en la que se pueden situar uno o varios campos entre paréntesis. No es necesario que sea el último campo de la tabla, pero no se
puede elegir columnas que estén afectadas por restricciones de unicidad de tipo P o U que estén referenciadas, o que estén afectadas
por restricciones multicampo; en caso contrario se tendría que eliminar primero la restricción y luego eliminar el campo o bien
aplicar la opción:
ALTER TABLE tres DROP (FechaDeNacimiento) cascade constraints;
*/

```

```
ALTER TABLE tres DROP COLUMN FechaDeNacimiento;
```

```
describe tres
```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

## 342

```
drop table tres purge;
```

-- Generamos una tabla Tres sin la columna FechaDeNacimiento, para poder añadirle posteriormente dicho campo

```

CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
Nombre varchar2(25),
Apellido varchar2(25),
CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );

```

```
describe tres
```

```

column object_name format a25
select object_name, object_type from user_objects;
column object_name clear

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

```

column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle

```

```

column index_name format a25
column table_name format a11

```

```
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

```
column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column
```

## 35

```
/*
  Añadimos ahora el campo que le falta con su restricción correspondiente.
  La columna añadida se sitúa siempre al final del registro.
  Si hay registros en la tabla, contendrá el valor null para esos registros.
  Por lo tanto, en principio no se podrá añadir una columna con una restricción
  de no nulidad en caso de existir registros en la tabla.
*/
```

```
ALTER TABLE tres ADD FechaDeNacimiento DATE CONSTRAINT RestriccionSemantica3 NOT NULL;
```

```
describe tres
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

## 36

```
-- Primero quitamos la no nulidad de la fecha en tres.
```

```
alter table tres drop constraint restriccionsemantica3;
```

```
/*
  Ahora eliminamos la clave primaria. Tiene una sintaxis alternativa:
  alter table tres drop primary key;
  que hace referencia de forma no ambigua al tipo de restricción a eliminar
*/
```

```
alter table tres drop constraint restriccionsemantica13;
```

```
-- Ahora quitamos la clave alternativa.
```

```
alter table tres drop unique (nombre,apellido);
```

```
describe tres
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

## 37

```
-- Restauramos la clave primaria de tres:
```

```
alter table tres add constraint RestriccionSemantica13 primary key (dni);
```

```
-- Sintaxis alternativa con nombre implícito: alter table tres add primary key (dni);
```

```
-- Ahora reponemos la clave alternativa:
```

```
alter table tres add constraint RestriccionSemantica23 unique (nombre,apellido);
```

```
-- Sintaxis alternativa: alter table tres add unique (nombre,apellido);
```

```
-- Podemos modificar una columna existente con una restricción de no nulidad:

alter table tres modify (fechadenacimiento constraint RestriccionSemantica3 not null);

-- Sintaxis alternativa: alter table tres modify (fechadenacimiento not null);

-- Las modificaciones pueden alcanzar también al tipo de datos y/o al valor por defecto.
-- En este último caso, el valor por defecto afectará sólo a los futuros registros.
-- Obsérvese que realmente el ejemplo de uso del 'default' no es muy acertado, dado que
-- 'dni' es clave primaria y consecuentemente no admitirá nulos en las inserciones.
```

```
alter table tres modify (dni varchar2(11) default 'hola');
```

```
describe tres
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

```
column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column
```

## 38

```
drop table tres purge;
```

```
-- Restauramos la tabla tres a su estado original
```

```
CREATE TABLE Tres (Dni varchar2(10) CONSTRAINT RestriccionSemantica13 PRIMARY KEY,
Nombre varchar2(25),
Apellido varchar2(25),
FechaDeNacimiento DATE CONSTRAINT RestriccionSemantica3 NOT NULL,
CONSTRAINT RestriccionSemantica23 UNIQUE (Nombre, Apellido) );
```

```
describe tres
```

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

```
column index_name format a25
```

```
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

```
column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column
```

## 41

-- Creamos la tabla cuatro, para su uso posterior

```
drop table cuatro purge;
```

```
CREATE TABLE Cuatro (Dni varchar2(10) CONSTRAINT RestriccionSemantica14 PRIMARY KEY,
Nombre varchar2(25) NOT NULL,
Apellido varchar2(25) NOT NULL,
FechaDeNacimiento DATE NOT NULL,
CONSTRAINT RestriccionSemantica24 UNIQUE (Nombre, Apellido) );
```

```
describe cuatro
```

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

```
column index_name format a22
column table_name format a10
column column_name format a12
column column_position format 99999
set line 81
select *
from user_ind_columns;
set line 80
clear column
```

## 50

-- La tabla cinco referencia a la tabla cuatro

```
drop table cinco purge;
```

```
Create table Cinco (NumSS varchar2(10) CONSTRAINT RestriccionSemantica15 PRIMARY KEY,
Dnrelacionado varchar2(10)
CONSTRAINT RestriccionSemantica25 REFERENCES Cuatro);
```

```
describe cinco
```

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

```

column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle

```

## 502

-- Establecemos la regla de borrado para la clave ajena en propagación. Si no se explicita a regla de borrado -- estará en restricción.

```
drop table cinco purge;
```

```

Create table Cinco (NumSS varchar2(10) CONSTRAINT RestriccionSemantica15 PRIMARY KEY,
                  Dnirelacionado varchar2(10)
                  CONSTRAINT RestriccionSemantica25 REFERENCES Cuatro
                  on delete cascade);

```

-- Alternativamente puede expresarse 'on delete set null'

-- No admite regla de actualización. (Siempre en restricción).

```

column constraint_name format a25
column table_name format a10
select constraint_name, constraint_type, table_name, r_constraint_name, delete_rule
from user_constraints;
cle col

```

-- Restablecemos la regla de borrado para la clave ajena en 'restricted'

```
drop table cinco purge;
```

```

Create table Cinco (NumSS varchar2(10) CONSTRAINT RestriccionSemantica15 PRIMARY KEY,
                  Dnirelacionado varchar2(10)
                  CONSTRAINT RestriccionSemantica25 REFERENCES Cuatro);

```

```

column constraint_name format a25
column table_name format a10
select constraint_name, constraint_type, table_name, r_constraint_name, delete_rule
from user_constraints;
cle col

```

## 55

-- Eliminamos la integridad referencial

```
ALTER TABLE cinco DROP CONSTRAINT RestriccionSemantica25;
```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

## 56

-- Restituimos la integridad referencial

```

ALTER TABLE cinco ADD CONSTRAINT RestriccionSemantica25
FOREIGN KEY (DNIRELACIONADO) REFERENCES Cuatro;

```

```

column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col

```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

## 58

-- No podemos eliminar una clave primaria que soporte alguna integridad referencial

```
alter table cuatro drop constraint restriccionsemantica14;
```

-- Sin embargo, Oracle proporciona la opción de propagación, con lo cual se eliminará la  
 -- clave primaria y la(s) correspondiente(s) clave(s) ajena(s). Hay que ser muy cuidadoso con esta opción,  
 -- porque podríamos eliminar accidentalmente un buen número de restricciones referenciales.

```
alter table cuatro drop constraint restriccionsemantica14 cascade;
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

## 585

-- Devolvemos las tablas cuatro y cinco a su estado original

```
drop table cuatro purge;
```

```
CREATE TABLE Cuatro (Dni varchar2(10) CONSTRAINT RestriccionSemantica14 PRIMARY KEY,
                      Nombre varchar2(25) NOT NULL,
                      Apellido varchar2(25) NOT NULL,
                      FechaDeNacimiento DATE NOT NULL,
                      CONSTRAINT RestriccionSemantica24 UNIQUE (Nombre, Apellido) );
```

```
drop table cinco purge;
```

```
Create table Cinco (NumSS varchar2(10) CONSTRAINT RestriccionSemantica15 PRIMARY KEY,
                   Dnirrelacionado varchar2(10)
                   CONSTRAINT RestriccionSemantica25 REFERENCES Cuatro);
```

## 59

-- Intentamos borrar la tabla cuatro

```
drop table cuatro purge;
```

-- Oracle dispone de la propagación del borrado de tablas a la(s) restriccion(es)  
 -- referencial(es) que le afecte(n).

```
drop table cuatro cascade constraint purge;
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

## 595

-- Devolvemos las tablas cuatro y cinco a su estado original

```
drop table cuatro purge;
```

```
CREATE TABLE Cuatro (Dni varchar2(10) CONSTRAINT RestriccionSemantica14 PRIMARY KEY,
                      Nombre varchar2(25) NOT NULL,
                      Apellido varchar2(25) NOT NULL,
                      FechaDeNacimiento DATE NOT NULL,
                      CONSTRAINT RestriccionSemantica24 UNIQUE (Nombre, Apellido) );
```

```
drop table cinco purge;
```

```
Create table Cinco (NumSS varchar2(10) CONSTRAINT RestriccionSemantica15 PRIMARY KEY,
Dnrelacionado varchar2(10)
CONSTRAINT RestriccionSemantica25 REFERENCES Cuatro);
```

## 64

-- Creamos la tabla Seis con una CLAVE AJENA a Cuatro y una restricción referencial  
-- sobre la clave alternativa de Cuatro.

```
drop table seis purge;
Create table Seis (NumSS varchar2(10) CONSTRAINT RestriccionSemantica16 PRIMARY KEY,
Dnrelacionado varchar2(10)
CONSTRAINT RestriccionSemantica4 REFERENCES Cuatro,
Nombrelacionado varchar2(25),
Apellidorelacionado varchar2(25),
CONSTRAINT RestriccionSemantica36
FOREIGN KEY (Nombrelacionado, Apellidorelacionado)
REFERENCES Cuatro (Nombre, Apellido) );
```

describe seis

```
column object_name format a25
select object_name, object_type from user_objects;
column object_name clear
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
column column_name cle
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

## 65

-- Eliminamos la restricción referencial sobre la clave alternativa de Cuatro

```
ALTER TABLE seis DROP CONSTRAINT RestriccionSemantica36;
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

## 66

-- Restauramos la restricción referencial sobre la clave alternativa de cuatro.

```
ALTER TABLE seis ADD CONSTRAINT RestriccionSemantica36
FOREIGN KEY (Nombrelacionado, Apellidorelacionado)
REFERENCES Cuatro (Nombre, Apellido);
```

```
column constraint_name format a25
column table_name format a10
column search_condition format a35
select constraint_name, constraint_type, table_name, search_condition
from user_constraints;
cle col
```

```
column column_name format a30
select constraint_name, column_name, position
from user_cons_columns;
```



```
column column_name cle
```

```
column index_name format a25
column table_name format a11
select index_name, index_type, table_name, uniqueness
from user_indexes;
cle col
```

## 70

```
-- Intenta crear una restricción referencial sobre un campo de Cuatro que no es clave
-- candidata.
```

```
Create table Siete (NumSS varchar2(10) CONSTRAINT RestriccionSemantica17 PRIMARY KEY,
    Dnirelacionado varchar2(10)
    CONSTRAINT RestriccionSemantica5 REFERENCES Cuatro,
    Fecharelacionada DATE
    CONSTRAINT RestriccionSemantica6
    REFERENCES Cuatro (FechaDeNacimiento) );
```

## 80

```
-- Comprobamos que si no se indica la clave candidata a la que se desea referenciar
-- crea por defecto una clave ajena (a la clave primaria). En este caso falla porque
-- se aplica la opción de defecto, pero se usa como referenciante a un campo que no
-- está en el dominio de la clave primaria de Cuatro.
```

```
Create table Ocho (NumSS varchar2(10) CONSTRAINT RestriccionSemantica18 PRIMARY KEY,
    Dnirelacionado varchar2(10)
    CONSTRAINT RestriccionSemantica6 REFERENCES Cuatro (Dni),
    Nombrelacionado varchar2(25),
    Apellidorelacionado varchar2(25),
    CONSTRAINT RestriccionSemantica7
    FOREIGN KEY (Nombrelacionado, Apellidorelacionado)
    REFERENCES Cuatro);
```

## Restricciones de chequeo en columnas de tipo *date*

/\*

Las restricciones de CHECK tienen un problema especial en las constantes de fecha. Para estas constantes se exige que estén completamente especificadas, es decir, día, mes año completo, horas, minutos y segundos. Esta especificación se ha de ajustar al parámetro de inicio de formato de fecha por defecto (nls\_date\_format) y al del lenguaje de fecha por defecto (nls\_date\_language). Estos parámetros pueden estar o no especificados en el fichero de inicio; y aunque lo estén, las restricciones CHECK que afectan a literales de fecha exigen una especificación completa a fin de no depender de lo que esté establecido por defecto para cada sistema, base de datos o sesión. Por ello, la mejor manera de solventar el uso de estas restricciones es generar los valores de fecha a través de la función to\_date, especificando la fecha y el formato de forma concordante y el lenguaje a utilizar, sin tener que modificar ni depender de los parámetros de inicio o de defecto.

Por lo demás, a efectos de uso en el resto de las sentencias SQL, bastará con especificar el literal entre comillas simples y con los formatos que suele admitirse por defecto: '1-1-1', '01-01-01', '1-1-1901', '1/1/1', '01/01/2001', '1-ene-1', '01-ene-2001', '01/ene/01', ...

\*/

```
create table check_fecha (a date check (a between to_date('1-ene-1970, 00:00:00',
    'dd-mon-yyyy, hh24:mi:ss',
    'NLS_DATE_LANGUAGE=spanish')
and to_date('31-dic-1979, 23:59:59',
    'dd-mon-yyyy, hh24:mi:ss',
    'NLS_DATE_LANGUAGE=spanish')));
```

o bien...

```
create table check_fecha (a date check (a between to_date('1-1-1970, 00:00:00',
    'dd-mm-yyyy, hh24:mi:ss')
and to_date('31-1-1979, 23:59:59',
    'dd-mm-yyyy, hh24:mi:ss')));
```