

¿Qué son?

Recordemos, las restricciones o constraints (como comúnmente se le conocen), se emplean para ayudar a cuidar la integridad de los datos.

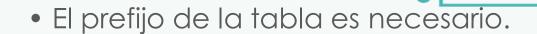
- La sintaxis empleada para crear un constraint es similar entre los distintos manejadores
- Un constraint está formado por un nombre, el tipo de constraint y una expresión que depende del tipo.
- Si no se especifica el nombre, el manejador asigna uno por default, sin embargo, se recomienda especificar siempre un nombre que sea significativo y fácil de identificar a dicho constraint.
- Lo anterior es útil para los casos en los que se comete algún error y se produce un error de violación del constraint. El manejador generará un mensaje de error incluyendo el nombre del constraint.



Nombres

Tipo	Convención de nombrado
Constraint	
not null	Para este tipo de constraint, generalmente se emplea la forma corta y no se especifica un nombre: create table empleado (
	num_empleado number not null);
	Si se desea emplear la sintaxis antes mencionada, la convención es: <pre><pre><pre><pre><pre><pre><pre>o tabla> <nombre columna=""> nn</nombre></pre></pre></pre></pre></pre></pre></pre>
unique	<pre><prefijo_tabla>_<nombre_columna>_uk</nombre_columna></prefijo_tabla></pre>
primary key	<nombre_tabla>_pk</nombre_tabla>
references, foreign key	<pre><prefijo_tabla_hija>_<nombre_columna>_fk</nombre_columna></prefijo_tabla_hija></pre>
check	<pre><prefijo_tabla>_<nombre_columna>_chk</nombre_columna></prefijo_tabla></pre>





- Un constraint es tratado como un objeto más y por lo tanto se requiere que cuente con un nombre único dentro de un mismo esquema.
- En SQL los constraints pueden aparecer en 2 lugares dentro de la definición de una tabla:
 - o A nivel campo (aparecen como parte de la definición del atributo): Column constraints
 - o A nivel tabla (aparecen como parte de la definición de una tabla posterior a la definición de sus atributos): Table constraints.



Column constraints.

EJEMPLO:

Crear la tabla CONCEPTO_PAGO. La clave de cada concepto debe ser única, el tipo de concepto

solo puede tener los valores A, B y C, y el importe de cada concepto no debe pasar de \$100000.

Observar que la PK de la tabla corresponde al campo CONCEPTO_ID





Llaves foraneas

QUINCENA

- NUMERO_QUINCENA NUMERIC(2,0) NOT NULL

DATE NOT NULL

◆ FECHA_FIN

DATE

NOT NULL

\<u>__</u>1

NOMINA



FECHA_CREACION DATE

NOT NULL

- ♠ QUINCENA_ID (FK) NUMERIC(10,0) NOT NULL.
- ♦ IMPORTE_TOTAL NUMERIC(10,2) NOT NULL



Limitantes

- La palabra constraint no puede aparecer más de una vez en la definición del campo.
- Restricciones como son llaves primarias o restricciones Unique que estén formadas por la definición de más de un atributo no pueden ser definidas a través de esta técnica.
- La definición del atributo se vuelve más compleja y en algunos casos, complica la lectura del código SQL ya que se está definiendo un campo y un constraint a la vez.



Table constraints

Son similares a los constraints definidos a nivel columna, con la diferencia de que estas se definen a nivel tabla, posterior a la definición de todos los campos de la tabla. Los tipos de constraints que pueden definirse a nivel de tabla son:

- unique
- primary key
- foreign key
- check

Como se puede observar en el código SQL, la definición del constraint ya no se asocia al campo si no a la tabla separándolos por comas.



(ejemplo3: Crear la tabla venta, agregar su PK. Considerar que los valores para el campo tipo_venta únicamente tendrán los valores MA o EL.)

Referencial Integrity constraints

- Aplican para los constraints references Y foreign key
- Las acciones que se pueden aplicar a los registros referenciados son: eliminarlos, o modificar el valor de la FK.

La opción update no está soportada, solo delete, con las siguientes variantes:

[on delete {no action | cascade | set null}]

set default, restrict no son soportadas en Oracle.

Los constraints de integridad referencial se especifican posterior a la definición del constraint references o foreign key



