

Oracle básico (I): Creación y manejo de tablas

Curso de Oracle

Por Carlos Galarza

www.softdownload.com.ar

En el artículo anterior (ver **Algoritmo** nº 7, págs. 5-10) dimos un vistazo general a *Oracle* como lenguaje de programación. Con esta entrega iniciamos una serie de artículos sobre aquellos aspectos que consideramos básicos para iniciarse en la programación en *Oracle*, entre los cuales destacamos los siguientes:

- Creación y manejo de tablas
- Consultas con SQLPlus
- Pantallas de ingreso con SQLForms
- Programación en PL/SQL
- Informes con SQLReport
- Diseño de menú con SQLMenu

Ajustándonos a esta lógica, el tema central de este artículo y punto de partida en cualquier curso de *Oracle* será el diseño y creación de tablas.

Creación de tablas

Como expusimos en nuestro artículo anterior, en *Oracle* cada estructura de información se denomina TABLA las cuales, junto a los índices y al diccionario de datos del sistema, componen la base de datos. Por lo tanto, la creación de las tablas en el proceso de programación en *Oracle* juegan un papel muy importante. En el momento de crear las tablas se definen características a dos niveles: Tabla y Columna, como se muestra a continuación:

A nivel de Tabla

- Nombre:** Nombre de la tabla puede ser de 1 a 30 caracteres.
- Propietario:** La tabla tiene como propietario al usuario que las crea. En nuestro caso somos el usuario *EIDOS*. Otro usuario que desee usar nuestras tablas debe tener autorización para ello y hacer referencia a la tabla como *eidos.clientes (propietario.tabla)*
- Cantidad de Columnas:** Una tabla puede tener un máximo de 254 columnas.

A nivel de Columna

- Nombre:** Puede tener de 1 a 30 caracteres.
- Tipo de dato y su ancho**
- | | |
|---------|--|
| CHAR | Máximo de 255. Por defecto 1. |
| NUMBER | Máximo de 105 dígitos. Por defecto 44. |
| INTEGER | Numérico sin decimal. Por defecto 38. |
| DATE | Hasta el 31 de diciembre de 4712. |
| LONG | Tipo caracter con tamaño variable hasta 65535 bytes. Permite una sola columna LONG por tabla. No se puede usar en subconsultas, funciones o índices. |
| RAW | Dato en binario puro (imágenes y sonido) con un ancho máximo de 255. |

LONGRAW Igual que *LONG*, pero para almacenar datos en binario puro.

Restricciones: Su función es definir reglas de validación de la columna.

Para facilitar la continuidad del análisis, usaremos como ejemplo las tablas definidas en el artículo anterior: *Clientes* y *VENTAS*.

La definición de *restricciones* al crear las tablas permite establecer reglas de validación de datos, así como los controles necesarios para mantener la integridad referencial entre tablas a través de las columnas claves. Las restricciones que se pueden definir son:

Valor obligatorio: En Oracle existe el concepto de valor nulo (*NULL*), como un valor indefinido o ausencia de valor y que es diferente al número 0 o al carácter espacio. Por lo tanto, para que una columna siempre tenga valor (sea obligatoria) se define como *NOT NULL*.

Rango de valores: Sirven para chequear que el valor sea mayor a un valor determinado o para que se encuentre entre dos valores.

Clave Primaria: Columnas que identifican de forma única al registro, es un valor único y no nulo (*NOT NULL*). Por ejemplo: el código del cliente es una *clave primaria* que identifica de forma única e irrepetible a cada cliente.

Clave Externa: Columna de la tabla que hace referencia a un valor que tiene que estar registrado en otra tabla. Por ejemplo: la columna código de la tabla *VENTAS* es una clave externa que hace referencia a un valor de la columna código (clave primaria) de la tabla *Clientes*.

En la versión 6 de Oracle (que dado lo reciente de la versión 7 aún se usa ampliamente) la única restricción que estaba activa era la de valor obligatorio (*NOT NULL*), siendo las otras restricciones sólo declarativas, o sea, que quedaban registradas en la definición de la tabla, pero no se podían activar. En la versión 6, para garantizar la unicidad de la clave primaria, era necesario crear índices con claves únicas, aspecto éste que retomaremos más adelante. En la versión 7 de Oracle estas restricciones están implementadas, garantizándose la verificación y corrección de datos en cualquier momento sin tener que programar estos controles.

Destacadas estas cuestiones veamos, entonces cómo se procede para crear las tablas *Clientes* y *Ventas*.

Tabla *Clientes*

Objetivo: Ficha con datos para identificar al cliente. Consta del código del cliente (número secuencial), fecha de alta al sistema, nombre, teléfono, dirección y alguna anotación.

Requisitos: Se debe identificar a cada cliente con un código único (clave primaria), registrando su nombre, teléfono y fecha de registro (estos datos son obligatorios). La dirección y anotaciones son campos opcionales.

Creación de la tabla: Ver el fuente 1

```
Fuente 1
CREATE TABLE clientes
(
    codigo      integer NOT NULL
                PRIMARY KEY,
    fecha       date     NOT NULL,
    nombre      char(30)  NOT NULL,
```

```
telefono char(20) NOT NULL,
direccion char(100),
anotacion LONG
);
```

Tabla Ventas

Objetivo: Registrar las ventas con la siguiente información: Código del cliente, fecha de la venta, artículo y valor de la venta.

Requisitos: El número del cliente es una clave externa que hace referencia a la columna código en la tabla Clientes. En este caso, todos los datos son obligatorios. Se controla que la columna valor sea mayor a cero.

Creación de la tabla: ver fuente 2

```
Fuente 2
CREATE TABLE ventas
(
    codigo    INTEGER      NOT NULL
    REFERENCES clientes(codigo),
    fecha     DATE         NOT NULL,
    articulo  CHAR(20)     NOT NULL,
    valor     NUMBER(10,2) NOT NULL
    CHECK (valor>0)
);
```

Las restricciones de Claves Primaria y Clave Externa se definieron a nivel de columna, pero se pueden definir a nivel de tabla, al final de la misma, como se muestra en el fuente 3:

```
Fuente 3
CREATE TABLE clientes
(codigo    INTEGER      NOT NULL,
nombre     CHAR(30)     NOT NULL,
direccion  CHAR(100),
anotacion  LONG,
PRIMARY KEY (codigo));

CREATE TABLE ventas
(codigo    INTEGER      NOT NULL,
fecha     DATE         NOT NULL,
articulo  CHAR(10),
valor     NUMBER(6,2)  NOT NULL
    CHECK (valor>0),
FOREIGN KEY (codigo) REFERENCES clientes(codigo));
```

La definición de la clave a nivel de tabla es necesaria cuando la misma está formada por más de una columna.

Unicidad de la clave con índices

Para garantizar la unicidad de los valores de la *clave primaria* de la tabla *Clientes* (en la versión 6 donde esta restricción sólo es declarativa y no está activa), se debe crear un índice que garantice

la unicidad de la clave principal. Un requisito importante para la unicidad de la clave principal es que las columnas de la clave se definen como *NOT NULL*.

A continuación mostraremos cómo crear el índice *cliente_codigo* para garantizar la unicidad de la clave primaria:

```
CREATE UNIQUE INDEX cliente_codigo  
ON clientes(código);
```

Secuencias: codificación numérica

La codificación numérica del cliente se puede realizar con una secuencia que automáticamente genera los números enteros en orden ascendente, no siendo necesario recordar cuál fue el último número asignado ; esto evita la duplicidad de códigos.

La secuencia es un objeto que genera valores enteros únicos y se emplean para crear claves primarias numéricas, con el uso del siguiente mandato:

```
CREATE SEQUENCE codigo_cliente  
INCREMENT BY 1  
START WITH 1;
```

Para registrar un nuevo código con la secuencia definida anteriormente se usa la pseudo-columna *codigo_cliente.NEXTVAL*, la cual nos dará el siguiente valor que le corresponde a la secuencia, la forma en que esto se realiza se explicará más adelante, cuando analicemos el ingreso de datos.

Para conocer el valor actual de la secuencia, o sea, el último código asignado, se usa la pseudo-columna *codigo_cliente.CURRVAL*, desde la tabla *DUAL* del sistema, cuyo fin es poder consultar pseudo_columnas (como se muestra a continuación):

```
SELECT user,sysdate,codigo_cliente.currval  
FROM DUAL;
```

donde:

- 1.- *user* es el nombre del usuario
- 2.- *sysdate* es la fecha del sistema
- 3.- *codigo_cliente.currval* es el último valor asignado a la secuencia.

Ingreso de datos

Una vez creadas las tablas, índices y secuencias, estamos en condiciones de ingresar datos en la tabla.

El ingreso, modificación y eliminación de registros se realiza fundamentalmente con el diseño de pantallas (*formularios*) desde el módulo *SQLFORMS* (que será tema de análisis específico en otro artículo). No obstante, en este artículo veremos el uso de los mandatos *INSERT UPDATE* y

DELETE.

Para ingresar un nuevo registro debemos ensayar lo que se muestra en la tabla 1:

Nombre de la Tabla (acciones)	Columnas
<pre>INSERT INTO CLIENTES VALUES (codigo_cliente.NEXTVAL, 'PINTURERIAS PROPIOS', '45 67 89' TO_DATE('10/04/95', 'DD/MM/YY'), 'Uruguay 1234', 'LIBRERIA');</pre>	<pre>codigo= secuencia nombre teléfono fecha dirección anotación</pre>

Tabla1: Inserción de nuevos registros

Como se podrá observar, en este ejemplo no se especificó la lista de columnas a insertar, lo que indica que se van a ingresar datos para todas las columnas. Por lo tanto, los valores para cada columna se tienen que ingresar en el orden en que están definidos en la tabla. Además, es de destacar que la palabra reservada *VALUES* indica la lista de valores a ingresar; que los datos tipo carácter van entre comillas; que la fecha se registra como una cadena de caracteres usando la función *TO_DATE* (encargada de transformar la cadena de caracteres '10/04/95' en fecha, a partir de un formato de fecha especificado '-DD/MM/YY'-).

También observamos que a la columna *codigo* se le asignó el siguiente valor de la secuencia *codigo_cliente* (*codigo_cliente.NEXTVAL*).

En caso de que sólo se asignaran valores a algunas columnas se debe dar la lista de columnas como se muestra en el fuente 4 correspondiente a la lista de columnas.

```
Fuente 4
INSERT INTO CLIENTES (codigo,nombre,teléfono,fecha)
VALUES
(codigo_cliente.NEXTVAL, 'CASA AUGÉ DEPORTES',
'598768', TO_DATE('15/04/95', 'DD/MM/YY'));
```

Las columnas a las que se les ingresa información se listan después del nombre de la tabla, en el orden deseado. Las columnas no listadas tendrán valor *NULL*, por ello todas las columnas definidas como obligatorias (*NOT NULL*) deben estar en la lista.

Los siguientes ejemplos muestran posibles errores y sus correspondientes mensajes en el registro de datos:

1.- Falta dato del teléfono que es obligatorio:

```
INSERT INTO CLIENTES(codigo,nombre)
VALUES (codigo_cliente.NEXTVAL,
'EMPRESA D'
);
```

Mensaje de error:

ORA-01400: mandatory (NOT NULL) column is missing or NULL during insert.

2.- Intento de registrar cliente con código ya existente:

```
INSERT INTO CLIENTES(codigo,nombre,teléfono)
VALUES (1,'EMPRESA TTT','341234');
```

Mensaje de error:

ORA-00001: duplicate key in index

Obsérvese en el caso 2 que la secuencia *codigo_cliente* **no** fue usada al ingresar el valor del código, y sí en el caso 1, provocando error de duplicidad de código. Esto ocurre porque la creación de la secuencia no garantiza la unicidad del código, ya que podemos registrar un código de cliente sin su uso. Sin embargo, la *unicidad* esta garantizada por la definición del índice único visto anteriormente. Si siempre se usa la secuencia la unicidad por supuesto que está garantizada, pero la simple definición de la secuencia no es garantía de su uso.

Listados de registros

A continuación veremos cómo obtener listados para revisar la información registrada, para lo cual seleccionaremos (*select*) registros desde (*from*) una tabla. En realidad el mandato *SELECT* será tema de análisis más detallado en la próxima entrega, por lo que ahora sólo lo trataremos con el objetivo de visualizar los datos ingresados.

Para obtener un listado de todas las columnas y todos los registros de la tabla *Clientes* debemos seguir este procedimiento:

Nombre de la Tabla

```
SELECT * FROM clientes;
```

Donde * = Todas las columnas

El resultado es el que se muestra en la tabla 2:

Código	Fecha	Nombre	Teléfono	Dirección	Anotación
1	10-Apr-95	Pinturerías propios	45 67 89	Uruguay 1234	Ferretería
2	15-Apr-95	Casa Auge deportes	598768		
3	20-Apr-95	Feria del libro	(0567)845677		
4	30-Apr-95	Club de tenis	905877		

Tabla 2: Datos de la tabla *Clientes*

Para listar sólo algunas columnas de la tabla *clientes* el procedimiento a seguir es:

```
SELECT codigo,nombre FROM clientes;
```

El resultado es el que se muestra en la tabla 3:

Código	Nombre
1	Pinturerías propios
2	Casa Auge deportes
3	Feria del libro
4	Club de tenis

Tabla 3: Datos de la tabla *Clientes* para *Codigo* y *Nombre*

Modificación de registros.

Para modificar valores de la tabla usaremos el mandato *UPDATE*, con el objetivo de modificar el teléfono y la dirección del cliente *Feria del libro*. Para ello, basta con definir:

```
UPDATE clientes
  SET telefono='234567',
      direccion='Andes 945'
WHERE nombre='Feria del libro';
```

donde:

Clientes es el nombre de la tabla

SET es para indicar el inicio de la lista de columnas y sus nuevos valores.

WHERE garantiza la selección de la fila del cliente.

Es importante destacar que si no se usa la cláusula *WHERE*, se modificará el valor de la columna en todas las filas de la tabla.

Eliminación de registros

La eliminación de registros se realiza con el mandato *DELETE*. El siguiente ejemplo eliminará los clientes con el código cero:

```
DELETE FROM clientes
WHERE codigo=0;
```

En este caso, si se omite la cláusula *WHERE* serán eliminados todos los registros de la tabla.

SQLPlus

Todas las tareas anteriormente estudiadas se realizan con el módulo *SQLPlus* de Oracle, que trabaja en forma interactiva. A continuación enunciaremos los pasos necesarios para usar *SQLPLUS* y poder crear tablas, índices o secuencias, así como insertar datos y obtener listados:

- 1.- Llamar al programa

SQLPLUS

- 2.- Identificación del usuario

Enter user-name: EIDOS

Enter password:

Si la identificación es correcta se obtiene mensaje de:
Connected to: ORACLE

Si, por el contrario, la identificación es incorrecta se recibe el siguiente mensaje:
ERROR: ORA-01017:
invalid username/password;
logon denied

- 3.- Indicador en pantalla de que *SQLPlus* está a la espera de la orden:

SQL>_

- 4.- Escribir los mandatos de creación de tablas, índices y secuencia en un archivo (*TABLAS.SQL*) con el uso del editor.

SQL>edit tablas

- 5.- Ejecutar los mandatos escritos en el archivo *TABLAS.SQL*

SQL>@tablas

- 6.- Para salir de *SQLPlus*

SQL>exit

Otras tareas

A continuación examinaremos una serie de mandatos, a nivel de definición de las tablas, gracias a los cuales se puede:

- 1.- Listar estructuras de las tablas (*DESCRIBE*)
- 2.- Modificar la estructura de las tablas (*ALTER TABLE*)
- 3.- Renombrar las tablas (*RENAME*)
- 4.- Eliminar una tabla (*DROP TABLE*)
- 5.- Eliminar un índice (*DROP INDEX*)
- 6.- Consultar las tablas del diccionario
- 7.- Listado de tablas, Índices y secuencias propiedad del usuario.

Detengámonos en los detalles más significativos de cada una de dichas tareas:

- 1.- Listar estructura de las tablas (*DESCRIBE*)

Para obtener la estructura (descripción de una tabla) el mandato que se debe emplear es:

```
SQL>DESCRIBE clientes;
```

con lo que el resultado será el que se ofrece en la tabla 4.

Name	Null?	Type
NUMERO	NOT NULL	NUMBER(38)
FECHA	NOT NULL	DATE
NOMBRE	NOT NULL	CHAR(30)
TELEFONO	NOT NULL	CHAR(20)
DIRECCION		CHAR(100)

ANOTACION	LONG
-----------	------

Tabla 4: Resultado del uso del mandato *DESCRIBE*

2.- Modificar la estructura de las tablas (*ALTER TABLE*)

La modificación de la estructura de las tablas con el uso de *ALTER* permite:

- Añadir nuevas columnas.
- Añadir restricciones a una columna, en este caso la tabla no debe contener datos.
- Modificar el ancho de la columna.
- Modificar el tipo de datos de la columna sólo si la columna no contiene datos o está vacía.
- Modificar al tipo LONG sólo una columna sin restricciones.

El siguiente ejemplo muestra cómo añadir, en la tabla *Ventas*, las columnas *Factura* (para registrar el número de factura) y *Cobro* (tipo carácter con 2 posibles valores, N=NO cobrada, NULL=cobrada) y modificar la columna valor para ampliar su ancho.

```
ALTER TABLE ventas
ADD (
    factura    integer,
    cobro      char
)
MODIFY (
    valor number(10,2)
);
```

3.- Renombrar las tablas (*RENAME*)

Para cambiar el nombre de la tabla *Clientes* a *EMPRESAS* se usa el siguiente mandato:

```
SQL>RENAME clientes TO empresas;
```

4.- Eliminar una tabla (*DROP TABLE*)

Le eliminación de la tabla es como sigue:

```
SQL>DROP TABLE clientes;
```

En este caso se eliminan, también, todos los índices de la tabla.

5.- Eliminar un índice (*DROP INDEX*)

```
SQL>DROP INDEX cliente_codigo;
```

6.- Consultar las tablas del diccionario.

Toda la información de las tablas está registrada en el diccionario del sistema (*Data Dictionary*), que son tablas especiales que se crean en la instalación de ORACLE (que son administradas por el sistema).

Para consultar la lista de tablas que componen el diccionario se escribe:

```
SQL>HELP DATA DICT
```

Gracias a lo cual se muestra una lista con la información de la tabla 5:

Nombre de la tabla	Descripción
ACCESSIBLE_COLUMNS	columns of all tables, views, and clusters
ACCESSIBLE_TABLES	tables and views accessible to the user
AUDIT_ACTIONS	maps action type numbers to action type names
ALL_INDEXES	descriptions of indexes on accessible
ALL_SEQUENCES	descriptions of the user's own sequences
ALL_TABLES	description of tables accessible to the user
.....	
USER_TABLES	descriptions of the user's own tables
USER_TAB_COLUMNS	columns of the user's tables, views, and clusters
USER_TAB_GRANTS	grants on objects where the user is the owner, grantor, or grantee

Tabla 5: Consulta de las tablas que componen el diccionario

También podemos ver la estructura de una tabla del diccionario como se muestra a continuación:

```
SQL>DESCRIBE ALL_TABLES;
SQL>DESCRIBE all_indexes;
SQL>DESCRIBE all_sequences;
```

7.- Listar las tablas, índices y secuencias definidas por un usuario

Para las tablas:

```
SQL>SELECT TABLE_NAME "TABLA"
FROM ALL_TABLES
WHERE OWNER='EIDOS';
```

Resultado: *Clientes y Ventas*

Para los índices:

```
SQL>SELECT table_name,index_name
FROM all_indexes
WHERE owner='EIDOS';
```

Resultado: *Clientes* (con índice *Cliente_Nombre* y *Cliente_Numero*) y *Ventas* (con índice *Venta_Numero*)

Para las secuencias:

```
SQL>SELECT sequence_name
FROM all_sequences
WHERE sequence_owner='EIDOS';
```

Resultado: el nombre de la secuencia usada (*SEQUENCE_NAME*) *Codigo_Cliente*

Como hemos visto, la creación de las tablas constituye el fundamento del diseño de cualquier sistema a desarrollar en Oracle. Una vez definida las tablas el paso lógico siguiente es conocer las técnicas para realizar un adecuado uso de la información contenida en el sistema. Por ello, el próximo artículo lo dedicaremos al lenguaje de Consulta SQL.