

# Curso de Oracle

Por Carlos Galarza

www.softdownload.com.ar

*SQLForms* es la herramienta de *Oracle* que permite, de un modo sencillo y eficiente, diseñar pantallas para el ingreso, modificaciones, bajas y consultas de registros. El usuario podrá, una vez definida la forma, trabajar con ella sin necesidad de generar códigos, dado que *Oracle* trae incorporado un conjunto de procedimientos y funciones asociados a las teclas de funciones, como por ejemplo la tecla [F7], que se usa para iniciar una consulta.

El objetivo de este artículo es el estudio de los conceptos básicos de *SQLForms*, a partir de los cuales el lector estará en condiciones de profundizar independientemente con el la documentación existente sobre *Oracle*, que es completa, voluminosa y con ejemplos muy ilustrativos.

## Forma

La forma elegida para el diseño es la de *Cliente-Ventas*, cuyo objetivo, como se muestra en la siguiente figura, es mostrar los datos básicos del cliente y las ventas realizadas:

Las tablas 1 y 2 representan la Forma *Cliente-Ventas*

Código	Fecha	Nombre	Teléfono	Dirección	Anotación
3	27-09-95	Feria del libro	234555	Canelones 1800	

Tabla 1: Bloque Cliente

Fecha	Artículo	Valor
20/09/95	Papel Fanfold	110
11/09/95	Disquete	190
24/08/95	Papel Fotocopia	310

Tabla 2: Bloque Ventas

La forma se organiza en bloques de información, donde cada uno tiene asociado una tabla de datos y las columnas seleccionadas. La forma puede ocupar una o varias pantallas. En el ejemplo, como se puede observar, ocupa una pantalla.

## Bloque

En nuestro ejemplo la forma está compuesta por dos bloques: *Cliente* y *Ventas*. A continuación damos la descripción de cada uno de ellos, con su correspondiente definición en *SQLForms*.

### Cliente

Objetivo: Ficha básica con datos del cliente.

Tabla: CLIENTES.

Registros: Presentación simple, un registro por cliente.  
 Tipo: Bloque Principal (*Master Block*).  
 Orden: Por Nombre del cliente (*ORDER BY NOMBRE*).

Pantalla de definición:

Block: CLIENTE	Records	Array Size:
Table: CLIENTES	Displayed: 1	[ ] Prim Key
Sequence Number: 1	Buffered:	[ ] In Menu
	Lines per:	[ ] Column Sec
Default Where/Order By:	ORDER BY NOMBRE	

## Ventas

Objetivo: Ventas realizadas a un cliente.  
 Tabla: VENTAS.  
 Registros: Presentación Múltiple, varios registros por cliente, donde cada registro ocupa una línea.  
 Tipo: Bloque Detalle, cuya información detalla las ventas del cliente representado en el bloque Principal. La relación entre bloques puede establecerse por uno o más campos. En este caso el campo CODIGO del cliente es el que relaciona ambos bloques. Por eso definimos como condición de relación CLIENTE.CODIGO = VENTAS.CODIGO.  
 Orden: Por fecha de venta en forma descendiente (*ORDER BY FECHA DESC*).

Pantalla de definición:

Block: VENTAS	Records	Array Size: 3
Table: VENTAS	Displayed: 5	[ ] Prim Key
Sequence Number: 2	Buffered: 5	[ ] In Menu
	Lines per: 1	[ ] Column Sec
Default Where/Order By:	ORDER BY FECHA DESC	
Master Block: CLIENTE	[ X ] Delete Details	
Join Condition	CLIENTE.CODIGO = VENTAS.CODIGO	

## Campo

Los datos de la forma se llaman campos, pudiendo los mismos representar columnas de la tabla o variables de memoria. La identificación del campo está compuesta por el nombre del bloque y el nombre del campo, como por ejemplo :*CLIENTE.CODIGO* y :*VENTAS.CODIGO*.

En el momento de crear la forma se determinan para cada campo:

### Definiciones básicas

Nombre.  
 N° de orden.  
 Tipo de dato.  
 Ancho del campo, consulta y visualización.  
 Posición en pantalla.

### Definiciones avanzadas

Formato de presentación. El formato de presentación del campo tipo fecha es *DD-MON-YY*, pudiendo ser cambiado al formato *dd-mm-yy* o *dd/mm/yy*.

Valores por defecto: Si en el momento del alta de clientes se desea generar una codificación numérica secuencial en forma automática, se debe asignar al campo *CODIGO* del bloque *CLIENTE* el siguiente valor de la secuencia *CODIGO\_CLIENTE*, previamente creada en *SQLPLUS*. En este caso se define el valor por defecto como *SEQUENCE.CODIGO\_CLIENTE.NEXTVAL*

A un campo de tipo fecha que se desee iniciar con la fecha del sistema se le asigna un valor por defecto igual a *\$\$date\$\$*, que es la variable del sistema que contiene la fecha.

Rango de valores: Para asignar un rango de valores a un campo se definen sus valores extremos.

Campo de relación: Define si el campo esta relacionado a un campo del bloque principal. En nuestro ejemplo el campo *:VENTAS.CODIGO* se relaciona con el campo *:CLIENTE.CODIGO* del bloque principal.

Lista de valores: Asigna a un campo una lista de valores a consultar. Para consultar la lista de clientes por nombre o por código, se define una lista de valores para el campo *:CLIENTE.CODIGO*, como se muestra a continuación:

```

Titulo:          CLIENTES
Posición:       X: 10 Y: 10
Comando SQL:    SELECT NOMBRE,CODIGO
                INTO :CLIENTE.NOMBRE,:CLIENTE.CODIGO
                FROM CLIENTES ORDER BY NOMBRE

```

En la siguiente figura se muestran las pantallas con las definiciones básicas y avanzadas de los campos: *CLIENTE.CODIGO* y *:VENTAS.CODIGO*

#### **:Cliente.Codigo**

```

Field Name:      CODIGO
Sequence Number: 1
Data Type:       NUMBER ( Select Attributes )
Field Length:    10
Query Length:    10
Display Length:  10
Screen Position: X: 20   Y: 4
Page: 1 ( Editor Attributes )
Format Mask:
Default Value:   SEQUENCE.CODIGO_CLIENTE.NEXTVAL
Hint: Enter value for : CODIGO
Valid Range: Low: High:
Enforce Key:
List of Values: Title: CLIENTES
Pos:             X: 10   Y: 10
List of Values SQL Text:

```

```

SELECT NOMBRE, CODIGO
      INTO :NOMBRE, :CODIGO
      FROM CLIENTES
      ORDER BY NOMBRE

```

#### **:Ventas.codigo**

Field Name: CODIGO  
 Sequence Number: 4  
 Data Type: NUMBER ( Select Attributes )  
 Field Length: 10  
 Query Length: 10  
 Display Length: 10  
 Screen Position: X: Y:  
 Page: ( Editor Attributes )  
 Format Mask:  
 Default Value:  
 Hint:  
 Valid Range: Low: High:  
**Enforce Key:** **CLIENTE.CODIGO**  
 List of Values: Title:  
 Pos: X: Y:

### Atributos

Los atributos definen las siguientes características de un campo:

Tipo de campo:	Columna de tabla o Variable de memoria.
Clave Primaria:	Indica que los registros ingresados tienen un único valor en este campo.
Mostrar:	Muestra el valor del campo.
Obligatorio:	Se requiere ingresar un valor. No puede ser NULL.
Ingresar:	Se puede ingresar información en el campo.
Modificar:	Indica que se puede cambiar el valor del campo después de realizar una consulta.
Modificar si es Null:	Indica que se puede cambiar el valor del campo después de realizar una consulta, solamente en el caso que el valor del campo sea NULL.
Consultar:	Indica que en modo consulta se puede escribir una condición de consulta.
Conversión a Mayúscula:	Conversión automática a mayúscula.
Visualizar:	Visualizar el contenido del campo al momento de ingresar información. En caso contrario se muestra en blanco.
Ancho Fijo:	Indica que el valor a ingresar tiene que ser del ancho del campo.
Salto automático:	Cuando se llena el campo pasa automáticamente al siguiente.
Texto de Ayuda:	Se muestra al momento de ingresar el campo un texto de ayuda.

A continuación se muestra la relación de elementos que perfilan la definición de atributos para el campo *CLIENTE.CODIGO*

☒ Base Table  
☒ Primary Key  
☒ Displayed  
☒ Required  
☒ Input Allowed  
☒ Update Allowed  
☐ Update if Null  
☒ Query Allowed  
☒ Uppercase  
☒ Echo Input  
☐ Fixed Length  
☐ Automatic Skip  
☐ Automatic Hint

## Consultas

Hasta el momento estudiamos la creación de formas. Ahora veremos cómo, sin necesidad de generar ningún código, ya estamos en condiciones de almacenar información y realizar las consultas correspondientes.

Antes de pasar a ver los distintos tipos de consulta, queremos detallar los pasos necesarios para habilitar una consulta:

- Iniciar la forma.
- Ir al bloque a consultar.
- Dar inicio a la consulta (con la tecla [F7]).
- Ubicar el cursor en el campo a consultar.
- Escribir la condición de consulta.
- Realizar la consulta (con la tecla [F8]).
- Ver el resultado de la consulta en pantalla.

A continuación estudiaremos los diferentes tipos de consultas, con un ejemplo para cada caso, estas consultas son:

- Exacta.
- Condicional.
- Aproximada.
- Múltiple.
- Avanzada.

### Exacta

La consulta exacta es la que verifica una condición con el operador de *igual a*. Ilustremos este caso con la consulta de las ventas de un determinado artículo como puede ser **PAPEL FANFOLD**.

Los parámetros de consulta para **CLIENTE** serían:

CODIGO	1
FECHA	10-04-95
NOMBRE	PINTURERIAS PROPIOS
TELEFONO	45 67 89
DIRECCION	Uruguay 1234
ANOTACION	Ferretería y artículos para el Hogar

Y los parámetros correspondientes a **VENTAS**.

FECHA:  
ARTICULO: **PAPEL FANFOLD**  
VALOR:

### Condicional

La consulta condicional es la que incluye algún operador de comparación como <, <=, >, >=, !=.

Para consultar las ventas de cualquier artículo cuyo valor de venta esté por encima de 100, se define la siguiente consulta para *Clientes y Ventas*:

CODIGO 1

FECHA 10-04-95  
NOMBRE PINTURERIAS PROPIOS  
TELEFONO 45 67 89  
DIRECCION Uruguay 1234  
ANOTACION Ferretería y artículos para el Hogar

FECHA:  
ARTICULO:  
VALOR: **>100**

### Aproximada

La consulta aproximada es aquella que tiene un patrón de consulta a partir del cual *SQLForms* construye una condición con el operador *LIKE* (tema ya expuesto en el artículo *Oracle Básico (III)*).

Si necesitáramos consultar las ventas de cualquier tipo de papel para un cliente dado, bastaría con especificar el patrón *PAPEL%*. De esta forma serían consultados todos los artículos cuyo nombre comenzara con *PAPEL*, como se muestra a continuación:

CODIGO 1  
FECHA 10-04-95  
NOMBRE PINTURERIAS PROPIOS  
TELEFONO 45 67 89  
DIRECCION Uruguay 1234  
ANOTACION Ferretería y artículos para el Hogar

FECHA:  
ARTICULO: **PAPEL%**  
VALOR:

### Múltiple

Consulta múltiple es aquella en la que participan varios campos en la condición a verificar. Por ejemplo, si necesitáramos consultar las ventas de cualquier tipo de papel cuyo valor este por encima de 100 pesos, realizaríamos la siguiente consulta:

CODIGO 1  
FECHA 10-04-95  
NOMBRE PINTURERIAS PROPIOS  
TELEFONO 45 67 89  
DIRECCION Uruguay 1234  
ANOTACION Ferretería y artículos para el Hogar

FECHA:  
ARTICULO: **PAPEL%**  
VALOR: **>100**

### Avanzada

Consulta avanzada es aquella que combina diferentes condiciones para un mismo campo, o la que brinda la posibilidad de modificar el orden de presentación de los registros.

En este tipo de consulta se puede construir una condición con todas las posibilidades del mandato *SELECT* (estudiadas en el artículo *Oracle Básico (II)*), excepto el manejo de grupos con *GROUP BY*.

Veamos el siguiente ejemplo donde necesitamos buscar un cliente, cuyo nombre es *LABORATORIO CRUZ DEL SUR*, pero no se recuerda si fue registrado de forma completa o abreviada, como podría ser *L. CRUZ DEL SUR* o *LAB. CRUZ DEL SUR*.

Nuestra condición de consulta sería:

```
WHERE :NOMBRE LIKE 'L%' and :NOMBRE LIKE '%SUR%'
```

donde se buscan los clientes cuyo nombre comiencen con L y contienen la palabra SUR en cualquier lugar.

La pantalla de nuestra consulta sería:

CODIGO:  
FECHA:  
NOMBRE:    **N**

Criteria:        **:N LIKE 'L%' AND :N LIKE '%SUR%'**

En tipo de consulta, una vez que el cursor esta ubicado en el campo a consultar, a diferencia de los otros tipos de consultas debe realizar las siguientes acciones:

- Escribir un nombre de variable, como por ejemplo *:N*, que represente al campo *NOMBRE* en el criterio de consulta. Debe comenzarse con dos puntos para indicar que se está haciendo referencia a una variable y no a un valor de consulta.
- Con la tecla de realizar consulta, [F8] en nuestro caso, se habilita el cuadro donde se escribe el criterio de la consulta.
- Realizar la Consulta con [F10].

A continuación veamos cómo usar la consulta avanzada para cambiar el orden de presentación de los registros. En nuestra forma los registros del bloque *CLIENTE*, por definición, se presentan ordenados por nombre.

En el siguiente ejemplo deseamos consultar aquellos clientes registrado desde el 01/10/95 a la fecha de hoy y cuyo nombre contenga la palabra *LIBRO*, y el resultado debe estar ordenado en forma descendente por la fecha del registro.

CODIGO  
FECHA :**F**  
NOMBRE :**N**

Criteria:

```
:F >= TO_DATE('01/10/95','DD/MM/YY')
AND :N LIKE '%LIBRO%'
ORDER BY FECHA DESC
```

Hasta aquí estudiamos la creación de la forma y sus usos, sin aún pasar a la generación de códigos, tema de nuestro próximo artículo. Para ello, en la próxima entrega nos detendremos en

tareas de programación con el lenguaje *PL/SQL* para crear disparadores (Trigger) y procedimientos que nos permitan automatizar determinadas tareas, como pueden ser:

- La coordinación de consulta entre el bloque principal *CLIENTE* y el bloque de detalle *VENTAS*.
- La validación de las modificaciones (*COMMIT*).

## Bibliografía

### **ORACLE 7 Manual de Referencia**

**Koch, George.**  
Osborne/McGraw-Hill  
1999.

### **ORACLE Manual de Referencia.**

**Koch, George.**  
Osborne/McGraw-Hill.  
1997.

### **Mastering Oracle.**

**Cronin, Daniel.**  
Hayden Books.  
1999.