-SENTENCIAS SQL

. Introducción

El lenguaje SQL (Structured Query Language) fue desarrollado a principios de los años 70 por técnicos de IBM. Su objetivo consistía en proporcionar a los usuarios sin conocimientos de programación un método para extraer y mostrar la información contenida en una base de datos. Y hacerlo utilizando un lenguaje lo más parecido posible al lenguaje natural (en este caso, el inglés). Así, el lenguaje SQL nos facilita una forma de representar instrucciones como, por ejemplo: "Quiero los nombres y teléfonos de la tabla 'Clientes' ordenados alfabéticamente", o "Borra todos los registros de la tabla Clientes cuyo teléfono esté en blanco".

Con el paso del tiempo, SQL se ha convertido en un estándar de la industria (aunque también es cierto que existen considerables diferencias entre las versiones que interpretan unas bases de datos y otras). Las órdenes, o sentencias SQL, se dividen en varios grupos. En esta guía nos vamos a centrar en dos de ellos:

El lenguaje de consulta de datos (Data Query Language). Obtiene los datos de las tablas y determina el modo en que se presentan los datos. La orden SELECT es la instrucción principal de esta categoría.

El lenguaje de manipulación de datos (Data Manipulation Language). Proporcionan las órdenes INSERT y DELETE para añadir y borrar registros, y la orden UPDATE, que modifica el valor de los datos.

Es posible que se pregunte de qué le puede servir a usted el conocimiento del lenguaje SQL (de hecho, ya domina los listados programables, los filtros, los borrados múltiples etc., herramientas suficientes en principio para efectuar consultas a la Base de Datos).
Conociendo el lenguaje SQL puede llegar incluso más lejos. SQL le puede servir para realizar:

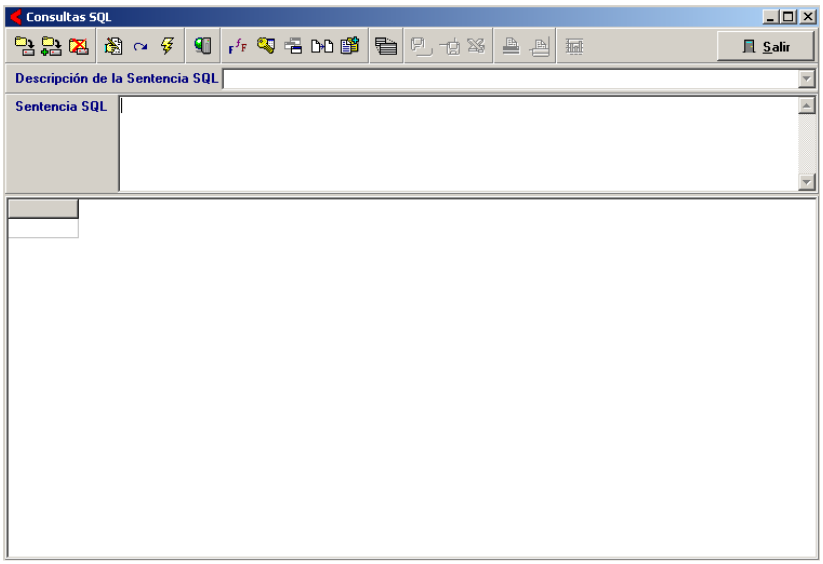
- Consultas sofisticadas de datos.
- Informes especiales no contemplados en ninguna de las opciones estándares de la aplicación.
- Listados que por su complejidad escapan del alcance de los listados programables.
- Correcciones de datos en casos excepcionales.
- Actualizaciones masivas de datos
- etc.

Está claro que muchas de estas situaciones no se plantean habitualmente al usuario medio de Vector. Sin embargo, estamos seguros de que a medida que vaya conociendo la herramienta, encontrará interesantes aplicaciones para su negocio.

.El interfaz SQL de Vector

Empezaremos por estudiar el modo de ejecutar sentencias SQL desde Vector. Para ello, entre en cualquier módulo, elija la opción Base de Datos, y seleccione Consultas SQL.

Verá la siguiente pantalla:

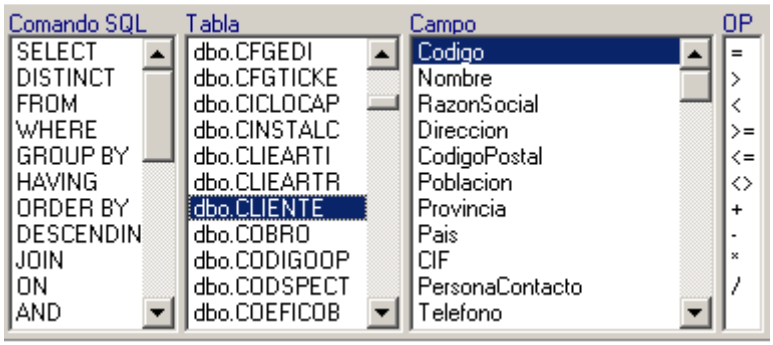


Interfaz SQL

Observe el recuadro situado a la derecha de "Sentencia SQL". Es el espacio reservado para editar la sentencia SQL. En el panel inferior podrá ver el resultado de la sentencia (en el caso de que se trate de una consulta), y en la casilla asociada a "Descripción de la Sentencia SQL" podrá visualizar la lista de las sentencias previamente generadas.

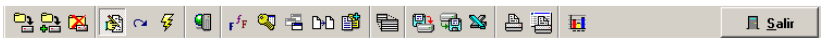
Evidentemente, la primera vez que entre en este punto verá esta lista en blanco. Irá creciendo a medida que vaya elaborando y guardando sus propias sentencias.

Como ayuda para conocer los comandos SQL y las tablas que componen la Base de Datos, dispone de una utilidad que puede activar pulsando el botón derecho en el panel de "Sentencia SQL". Verá que se despliega un pequeño menú. Seleccione cualquiera de las tres opciones (Insertar Comando SQL, Tabla, Campo) y podrá ver la siguiente ventana:



En la columna titulada Comando SQL podrá consultar una lista de palabras reservadas del lenguaje SQL. En la columna Tabla verá los nombres de todas las tablas utilizadas en Vector. Normalmente son bastante descriptivos. Si selecciona cualquier tabla, en la columna Campo obtendrá la lista de todos sus campos. Por último, en la columna de la derecha tiene los operadores que puede utilizar para construir expresiones. Observe que haciendo doble clic sobre cualquiera de los elementos anteriores, éste pasa al panel de edición. De esta forma puede ir componiendo la sentencia SQL deseada. Evidentemente, también puede teclear directamente los comandos sin pasar por esta pantalla.

Preste ahora atención a la barra de herramientas en la parte superior de la ventana:



En la siguiente tabla describimos brevemente los iconos de la barra de herramientas. Veremos algunos ejemplos de su utilización a medida que avancemos en la exposición del lenguaje SQL.

Icono	Texto asociado	Función
	Guardar la Sentencia SQL	Guardar la Sentencia SQL activa, es decir, la que puede visualizar en el panel Sentencia SQL.
	Guardar la Sentencia SQL como	Guarda la sentencia SQL solicitando una descripción o nombre para la misma.
	Eliminar la Sentencia SQL	Elimina la Sentencia SQL activa, es decir, la mostrada en el panel Sentencia SQL.
	Activar/Desactivar Modificación de Datos	Activando este modo de trabajo es posible editar (modificar, eliminar o insertar) los datos mostrados en la rejilla. Se debe activar este modo para ejecutar sentencias SQL de borrado (DELETE) o actualización (UPDATE)
	Refrescar el contenido de la rejilla	Refresca (permite ver las modificaciones realizadas en otras sesiones o por otros usuarios) los datos mostrados en la rejilla
	Ejecutar la sentencia SQL(F9)	Ejecuta la sentencia activa. Como alternativa, puede utilizar la tecla de función F9
	Cambio de Base de Datos	Cambia el origen de los datos (básicamente, el directorio donde están ubicadas las tablas) sobre los que se aplican las sentencias SQL
	Cambio de tipo de letra(pantalla e impresora)	Accede a una pantalla en la que podrá seleccionar el tipo de letra, tamaño, efectos etc..
	Restringir el Acceso a la Consulta a determinados Usuarios	Sirve para permitir la ejecución de la sentencia SQL seleccionada sólo a un conjunto de usuarios en concreto. Se

		mostrará una lista de usuarios del sistema, de la cual se deben seleccionar aquellos a los que se desea permitir la sentencia SQL seleccionada. Aquellos usuarios no administradores que no estén en la relación de usuarios permitidos, simplemente no tendrán seleccionable la sentencia SQL. En cambio aquellos usuarios administradores que intenten ejecutar esta sentencia SQL, y no estén en la relación de usuarios permitidos, les mostrará la advertencia de: "¡Vd. no tiene acceso a la consulta!".
	Personalizar la Consulta en un Menú de Aplicación	Sirve para permitir el uso de la sentencia SQL seleccionada como si formara parte del menú "***Adicional***" del módulo de aplicación seleccionado. Se mostrará una relación de módulos de aplicación de VECTOR-ERP, donde se deberá seleccionar bajo cuál se desea mostrar como disponible entre los informes el diseño de sentencia SQL seleccionado. Hay que tener en cuenta que sólo será visible para aquellos usuarios a los que se les ha dado acceso a la sentencia SQL mediante el botón anterior.
	Copiar la configuración de esta consulta a otras consultas	Sirve para extender la configuración de la sentencia SQL seleccionada a otras sentencias SQL ya almacenadas en el campo lista ***Descripción de la Sentencia SQL***. Se mostrará una lista de las sentencias SQL disponibles, y se deberá seleccionar aquellas a las que se quiere extender la configuración de accesibilidad de usuario y de menú establecida en la sentencia actual.
	Copiar todos los Informes a otras Empresas	Sirve para copiar a otras empresas las sentencias SQL disponibles en la empresa actual.
	Activar/Desactivar el zoom de rejilla	Oculta el panel Sentencia SQL, dando más tamaño a la rejilla de datos.
	Importar datos desde un fichero de texto(ASCII)	Incorpora datos a una tabla de Vector desde un fichero externo en formato ASCII.
	Exportar los datos a un fichero de texto(ASCII)	Exporta los datos mostrados en la rejilla a un fichero de texto. Puede utilizar posteriormente este fichero para importarlo desde Access, Excel etc..
	Exportar los datos a una Hoja Excel	Exporta los datos mostrados en la rejilla a una hoja Excel.
	Imprimir los datos en Impresora	Imprime los datos de la rejilla en su impresora. Puede realizar previamente una visualización en pantalla.
	Imprimir los datos (en impresora en formato apaisado)	Imprime los datos de la rejilla en formato apaisado. Le será útil cuando el número de campos es elevado.
	Ver los datos en representación gráfica	Genera un gráfico de los datos mostrados en la rejilla. Tenga en cuenta que no todas las consultas son susceptibles de visualizarse adecuadamente mediante un gráfico.

.El lenguaje SQL

No pretendemos describir en su totalidad el lenguaje SQL. Concretamente, vamos a estudiar la forma de realizar consultas, modificaciones y borrados mediante una serie de ejemplos sencillos. Puede ampliar y completar los conceptos aquí expuestos recurriendo a la numerosa bibliografía existente sobre el tema.

Consultas.

La sentencia SELECT se utiliza para obtener un conjunto de registros que puede proceder de una o más tablas. A grandes rasgos, la estructura de la instrucción Select es la siguiente (las cláusulas encerradas entre corchetes son opcionales):

SELECT [DISTINCT] *lista-de-expresiones*
FROM *lista-de-tablas*
[WHERE *condición-de-selección*]
[GROUP BY *lista-de-columnas*]
[HAVING *condición-de-agrupación*]
[ORDER BY *lista-de-columnas*]

Veamos un caso elemental. Sitúese en el recuadro situado a la derecha de "Sentencia SQL" y teclee (puede escribir indistintamente en mayúsculas o minúsculas):

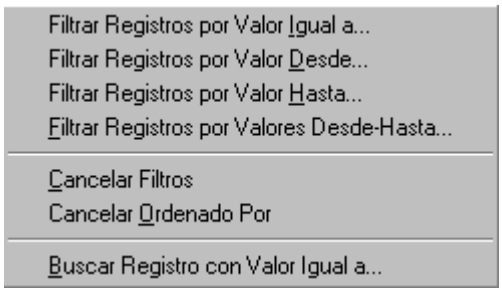
SELECT * FROM CLIENTE

A continuación pulse el botón ***Ejecutar la sentencia SQL***. Si ha cometido un error, el sistema le devolverá un mensaje de advertencia. Si no es así, verá que en la parte inferior de la ventana aparece una rejilla con los datos de sus clientes.

Analicemos brevemente la sentencia SQL. En primer lugar aparece la palabra SELECT seguida de un asterisco. Este asterisco indica que queremos visualizar todos los campos de la tabla. La palabra FROM se utiliza para indicar la tabla o tablas de donde queremos extraer la información, en este caso CLIENTE.

Detengámonos un momento para examinar algunas particularidades del interfaz SQL de Vector.

Pulse con el botón derecho en cualquier punto de la rejilla de datos. Verá un menú emergente:



Puede utilizar las opciones de este menú para seleccionar de una forma sencilla los registros a mostrar. Por ejemplo, si sólo desea visualizar los datos del cliente 1, pulse con el botón derecho en la columna Código, seleccione Filtrar Registros por Valor Igual a…. e introduzca el valor **1**. (Como veremos más adelante, el propio lenguaje SQL también nos proporciona un método alternativo para realizar este tipo de selección).

Observe que puede moverse por la rejilla de datos con las teclas <RePag>, <AvPag>, las flechas, o las barras de desplazamiento horizontal y vertical. Verá que los campos no son editables, es decir, no los puede modificar. Sin embargo, existe la posibilidad de hacerlo.

Para ello, pulse el botón ***Activar/Desactivar Modificación de Datos***. Observe que este es un botón que actúa como un conmutador. Una vez pulsado (activado), queda sombreado en un tono gris claro. Si lo vuelve a pulsar, recupera su aspecto original y queda desactivado. De esta forma podrá modificar o suprimir cualquier dato de la rejilla. Lógicamente, esta es una operación a realizar con precaución extrema, ya que Vector no valida los cambios realizados desde el interfaz SQL, es decir, no se comprueba la coherencia de la información introducida.

También cuenta con la posibilidad de imprimir el resultado de la consulta. Esto convierte al interfaz SQL en un potente generador de listados, que puede utilizar como complemento a los listados programables vistos anteriormente. Para imprimir los datos de la rejilla, pulse el icono ***Imprimir los datos en Impresora***, o el icono situado a su derecha si quiere obtener el listado en formato apaisado.

Veamos cómo guardar la sentencia. A continuación pulse el icono titulado ***Guardar la Sentencia SQL como***. El sistema le solicitará una descripción para identificar la sentencia generada. Puede escribir algo parecido a "Consulta de Clientes". Esta sentencia quedará almacenada para su posterior utilización.

Supongamos ahora que queremos obtener sólo determinados campos de la tabla, por ejemplo el nombre, la población, la provincia y el teléfono de sus clientes. Ejecute la siguiente sentencia:

```
SELECT NOMBRE, POBLACION, PROVINCIA, TELEFONO FROM CLIENTE
```

Indicaremos en este caso los nombres de los campos deseados separados por comas. Estos nombres deben ser exactamente iguales a los definidos en la Base de Datos (recuerde que puede consultarlos desplegando la ventana de ayuda con el botón derecho).

La cláusula WHERE. Sirve para restringir el número de registros a mostrar de acuerdo con alguna condición. Esta condición admite los seis operadores de comparación: =, <>, <, >, <=, >=
Por otra parte, cuando existan varias condiciones, estas se pueden agrupar utilizando los operadores AND/OR.
Por ejemplo, para obtener los datos de sus clientes de MADRID y BARCELONA:

```
SELECT NOMBRE, POBLACION, PROVINCIA, TELEFONO FROM CLIENTE
 WHERE PROVINCIA = 'MADRID' OR PROVINCIA = 'BARCELONA'
```

Si presta atención a la rejilla de datos, verá que los títulos de las columnas coinciden con los nombres de los campos de la tabla. Puede personalizar estos títulos utilizando el operador AS. Pruebe la anterior sentencia de esta forma:

```
SELECT NOMBRE, POBLACION, PROVINCIA, TELEFONO AS Tfno FROM CLIENTE
WHERE POBLACION = 'MADRID' OR POBLACION = 'BARCELONA'
```

Observará que la columna correspondiente al Teléfono se titula ahora Tfno, en lugar de TELEFONO. En realidad, puede incluso obviar el operador AS y colocar directamente el título después del campo: …TELEFONO Tfno FROM….

Fíjese que cada vez que quiera obtener los clientes de una provincia distinta, tendrá que modificar el texto entrecomillado directamente en la sentencia SQL. Para casos similares, le puede ser de utilidad el uso de parámetros en lugar de valores constantes. Para ello, utilice un nombre de parámetro precedido de los símbolos **:$**. Pruebe a escribir la siguiente sentencia:

```
SELECT NOMBRE, POBLACION, PROVINCIA, TELEFONO FROM CLIENTE
WHERE PROVINCIA = :$PROVINCIA
```

Al ejecutar la sentencia, Vector le solicitará un valor para la variable PROVINCIA. De esta forma puede utilizar la misma sentencia para consultar los clientes de cualquier provincia.

El uso de parámetros es muy sensible al tipo del dato que queremos comparar (en el caso anterior, Provincia es un campo de tipo alfanumérico, es decir, puede contener tanto caracteres como números). Para no extendernos demasiado sobre los tipos de datos y su problemática, tenga en cuenta las siguientes reglas:

Para campos alfanuméricos utilice :$VARIABLE
Para campos de tipo numérico utilice :#VARIABLE
Para campos de tipo fecha utilice :&VARIABLE

Si obtiene un mensaje de error del estilo "Type mismatch in expression" es probable que no haya seguido correctamente estas normas.

Por ejemplo, si quisiera obtener la relación de albaranes entre dos fechas determinadas, podría ejecutar una sentencia similar a esta:

```
SELECT NUMEROALBARAN, FECHA, CODIGOCLIENTE, NOMBRE
FROM ALBARAN
WHERE FECHA BETWEEN :&DESDEFECHA AND :&HASTAFECHA
```

Vector le solicitará en este caso dos valores: uno para la fecha inicial y otro para la fecha final.
Observe la utilización del operador BETWEEN para indicar un rango desde-hasta.
Evidentemente, también sería válida la cláusula:

```
WHERE FECHA >= :&DESDEFEHA AND FECHA <= :&HASTAFECHA
```

La cláusula ORDER BY nos permite ordenar los resultados. Por ejemplo, podríamos ordenar la anterior consulta por población y nombre:

```
SELECT NOMBRE, POBLACION, TELEFONO FROM CLIENTE
ORDER BY POBLACION, NOMBRE
```

Si quiere invertir el orden, utilice la palabra reservada DESCENDING, o su abreviatura DESC. Así por ejemplo, si tratamos con valores numéricos el orden por defecto que se aplica es de menor a mayor. Si utilizamos DESC sería de mayor a menor.

La cláusula GROUP BY nos permite agrupar los elementos devueltos utilizando como criterio el valor de alguno de los campos. Por ejemplo, podríamos obtener el total de unidades servidas por artículo:

```
SELECT CODIGOARTICULO, SUM(CANTIDAD)
FROM LALBARAN
GROUP BY CODIGOARTICULO
```

La palabra reservada SUM se utiliza como función agregada, que nos devuelve el sumatorio del campo CANTIDAD. También disponemos de las funciones MIN (devuelve el Mínimo de un conjunto de valores) MAX (Máximo) , AVG(Media) y COUNT (el número total de valores).

En combinación con esta cláusula puede utilizar la palabra reservada HAVING, que le servirá para restringir los datos devueltos según el valor calculado por la función agregada. Suponga que desea obtener la consulta anterior, pero sólo de aquellos artículos que hayan superado las 10 unidades vendidas.

```
SELECT CODIGOARTICULO, SUM(CANTIDAD)
FROM LALBARAN
GROUP BY CODIGOARTICULO
HAVING SUM(CANTIDAD) > 10
```

En muchas ocasiones, es necesario obtener la información desde varias tablas. Por ejemplo, podría interesarnos una relación de clientes incluyendo el nombre del agente asignado a cada uno de ellos. Observe que el nombre del agente es un dato de la tabla AGENTE, mientras que en la tabla CLIENTE se guarda el código del agente asignado. Pruebe con la siguiente sentencia:

```
SELECT NOMBRE, POBLACION, TELEFONO, AGENTE.NOMBRE FROM CLIENTE, AGENTE
WHERE CLIENTE.CODIGOAGENTE = AGENTE.CODIGO
```

Verá que a continuación de la cláusula SELECT aparecen campos procedentes de dos tablas, CLIENTE y AGENTE, detalladas en la cláusula FROM y separadas por comas. El campo que guardan en común, en este caso el campo CodigoAgente (este campo se llama Codigo en la tabla AGENTE, y CodigoAgente en la tabla CLIENTE), es el nexo que permite relacionar la información de ambas tablas. Este nexo de unión aparece detallado en la cláusula WHERE. Fíjese que para indicar el nombre del agente en la cláusula SELECT, hemos añadido el nombre de la tabla seguido de un punto (AGENTE.NOMBRE).

Las consultas de varias tablas pueden complicarse tanto como se desee. Por ejemplo, podría obtener una relación de agentes a los que no se les ha asignado ningún cliente:

```
SELECT NOMBRE FROM AGENTE
WHERE CODIGO NOT IN (SELECT CODIGOAGENTE FROM CLIENTE)
```

En este caso hemos empleado una subconsulta, es decir, una sentencia SQL dentro de otra sentencia SQL (concretamente en la cláusula WHERE). Hemos utilizado también el operador NOT IN , que localiza los registros que no aparecen en el conjunto de valores obtenidos por la subconsulta.

De forma similar, podríamos conocer los clientes a los que no hemos servido nada, los artículos que no han tenido ventas, los más vendidos, etc. Y utilizando adecuadamente las cláusulas ORDER BY y WHERE, podríamos aplicar diferentes criterios de ordenación combinados con diferentes rangos desde-hasta. (Tenga en cuenta, sin embargo, que no se puede conseguir "cualquier cosa" con una sentencia SQL. También tienen sus limitaciones, y en ocasiones, es inevitable el empleo de lenguajes de programación clásicos).

Modificaciones y Borrados

Hay que recalcar que las modificaciones y/o borrados deben realizarse por usuarios expertos o bajo las instrucciones de técnicos de IDS.

Para ejecutar una sentencia de este tipo, debe estar activado el modo Edición. Recuerde que para ello debe pulsar el icono ***Activar/Desactivar Modificación de Datos***.

Veamos algún ejemplo sencillo de modificación de datos:

```
UPDATE CLIENTE SET PROVINCIA = 'ALAVA'
```

Esta sentencia se aplicaría a todos los clientes de las base de datos.
Observe el uso de la palabra reservada SET delante del campo que queremos modificar.
Lógicamente, se puede restringir el alcance de la actualización mediante la cláusula WHERE. Por ejemplo:

```
UPDATE CLIENTE SET PROVINCIA = 'ALAVA'
WHERE POBLACION = 'VITORIA'
```

Veamos a continuación un ejemplo de borrado:

```
DELETE FROM CLIENTE
```

Esta sentencia borraría todos los registros de la tabla de clientes. También cuenta con la posibilidad de utilizar la cláusula WHERE, que puede llegar a ser tan sofisticada como sea necesario. Siguiendo con el ejemplo:

```
DELETE FROM CLIENTE
WHERE POBLACION = 'VITORIA'
```

Para terminar, tenga en cuenta que no hemos hecho más que arañar la superficie de una herramienta compleja y poderosa.

Le animamos a seguir profundizando en el tema. Para ello puede consultar la numerosa bibliografía existente en la web o bien informarse sobre los cursos que se imparte en IDS en esta materia.