

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

LABORATORIO DE COMPUTACION II

Unidad Temática 1: Sentencias de Manipulación de Datos

Guía de estudio

Curso: 1 er Año. 2 ndo Cuatrimestre

Índice

Problema N° 1:	2
Problema N° 2:	6
Problema N° 3	9

Problema N° 1:

Una librería mayorista cuenta con una base de datos llamada LIBRERIA que tiene la siguiente estructura:

BARRIOS	CLIENTES	ARTICULOS	VENDEDORES
(PK)cod_barrio	(PK)cod_cliente	(PK)cod_articulo	(PK)cod_vendedor
barrio	nom_cliente	descripcion	nom_vendedor
	ape_cliente	stock_minimo	ape_vendedor
	calle	pre_unitario	calle
	altura	observaciones	altura
	cod_barrio(FK)		cod_barrio(FK)
	nro_tel		nro_tel
	[e-mail]		fec_nac
			[e-mail]

FACTURAS	DETALLE_FACTURAS
(PK)nro_factura	(PK) { nro_factura(FK)
fecha	cod_articulo(FK)
cod_cliente(FK)	pre_unitario
cod_vendedor(FK)	cantidad

Para tener esta base de datos y poder ir probando y ejecutando las sentencias en su PC, baje del aula virtual el archivo llamado Script de la base de datos Librería, abra el archivo, copie todo el texto presente en el mismo, péguelo en una nueva consulta del Management Studio del SQL Server y ejecútelo completo.

Problema N° 1.1:

El encargado de la librería en cuestión necesita realizar una serie de actualizaciones, en primer lugar ingresar los artículos que este negocio tiene a la venta, ¿cómo lo haremos? Para ello vamos a hacer un repaso de las sentencias del transact-SQL para actualización de datos

Y dentro de este grupo de sentencias las que nos sirvan para insertar registros nuevos a una tabla es decir la:

Sentencia Insert

Cuya sintaxis es:

INSERT INTO Nombre_Tabla (lista_columnas) VALUES (lista_valores)

En donde:

Lista_columnas: son columnas de la tabla, en la que se quiere insertar datos.

lista_valores: son los valores que se van a insertar en las columnas antes especificadas. Los valores deben coincidir con el tipo de dato de la columna correspondiente, y debe existir tantos valores como columnas se especifiquen en la lista. Si una columna no tiene valor puede introducir un valor Null (si la definición de la columna lo permite) Los datos de tipo fecha y texto (varchar por ejemplo) van entre comillas simples y el separador de decimales es el punto. La coma se utiliza como separador de listas.

Ahora sí, ingresaremos a la base de datos los artículos que están en esta lista:

Descripción del artículo	Stock Mínimo	Precio unitario	Observaciones
Lápices Color largos * 12 u. Bic		101,50	
Conjunto Geométrico Maped		20,50	Regla, escuadra y transportador
Repuesto Gloria rallado	120	326,30	200 hojas
Repuesto Rivadavia		465,90	260 hojas, margen reforzado

Por ejemplo, para agregar a la tabla de artículos el primero de la lista anterior, deberemos escribir esta sentencia:

```

Insert into articulos (descripcion, pre_unitario)
values ('Lápices Color largos x 12u. Bic' 101.5)

--Una vez ejecutada ésta, podremos consultar todos los datos de la tabla artículos de esta manera
select * from articulos

--o bien, podremos consultar todos los datos de la tabla artículos cuya descripción comience con "Lápices"
select * from articulos
where descripción like 'Lápices%'

```

de esta forma se puede ir verificando lo realizado. De la misma manera continúe con el resto de los artículos, y tenga en cuenta lo siguiente:

- El campo cod_articulo no se incluye en la lista de campos porque es Identity, lo mismo que cod_vendedor y cod_cliente.
- Si un campo clave principal no es identity se debe incluir en la lista de campos porque de lo contrario da error.
- Cuando ingrese campos de tipo char, varchar y datetime se ingresan entre comillas simples.
- Todos los campos incluidos en la lista de campos deben tener un dato en la lista values o bien 0 (cero) para numéricos, "" (cadena de longitud cero) para char o varchar o bien NULL
- Si el número de teléfono es int y si se incluye código de área, la cantidad de dígitos no alcanza.
- si el valor que intenta agregar a una de las columnas no cumple con alguno de los constraints establecidos la operación abortará inmediatamente.

También es posible agregar múltiples filas a través del siguiente formato:

INSERT INTO Nombre_Tabla1

SELECT lista_campos FROM Tabla2

De esta forma se pueden insertar registros en una tabla, obteniéndolos de otra.

La librería además necesita que se agreguen los siguientes vendedores:

Ahora sí, ingresaremos a la base de datos los artículos que están en esta lista:

Apellido	Nombre	Calle	Altura	Tel.	Mail	Fecha Nac.	Barrio
Monti	Juan	Altoaguirre	1245	4522122			5
Sena	Rosa	Av. Velez Sarsfield	25		rsena@hotmail.com	15/5/1968	1

Agregue usted tres clientes.

Problema N° 1.2:

Días después se encontraron algunos errores en la carga de datos y será necesario modificarlos.

Para esto existe una sentencia de actualización que permite actualizar los datos ya cargados en los campos de la base de datos:

Sentencia Update

La sintaxis es:

```
UPDATE      Nombre_Tabla
SET         columna = Nuevo Valor,
           columna2 = Nuevo_Valor2
[WHERE      condición]
```

En la cláusula SET se especifican todas las columnas a las que se les quiere cambiar su valor, separadas por comas.

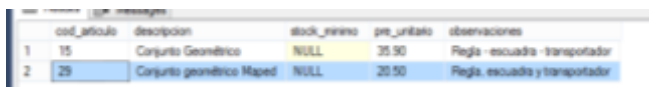
En la cláusula WHERE se puede especificar un filtro para modificar solo aquellos registros que cumplan con la condición establecida.

La modificación pedida fue que para el artículo cuya descripción es “Conjunto Geométrico” hay que cambiarla por “Conjunto Geométrico de Plástico”. Primeramente vamos a verificar el cod_articulo que el mismo posee:

Select * from articulos

where descripcion like 'Conjunto%'

y nos devuelve:



	cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	15	Conjunto Geométrico	NULL	35.50	Regla - escuadra - transportador
2	29	Conjunto geométrico Maped	NULL	20.50	Regla, escuadra y transportador

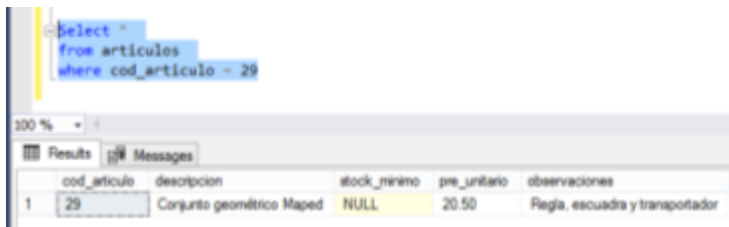
El código de artículo a modificar es el 29 entonces modificamos con la sentencia update. Verifique en su base de datos cuál es el código que corresponda, recuerde que el mismo es identity y es la base de datos la que le otorga el número y puede variar de una a otra.

update articulos

Set descripcion = 'Conjunto geométrico de plástico'

Where cod_arti = 29

Y verificamos si se logró realizar la actualización correctamente:



	cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	29	Conjunto geométrico Maped	NULL	20.50	Regla, escuadra y transportador

Luego se nos pidieron las siguientes modificaciones:

1. Para el artículo cuya descripción es “Lápices Color largos * 12 u” Bic; cambie el stock mínimo por 100, las observaciones por “Caja con motivos de Disney” y al precio por 17.20.
2. Para el artículo cuya descripción es “Repuesto Rivadavia”, cambie la descripción por “Repuesto Rivadavia cuadriculado” y las observaciones por “48 hojas”
3. Al vendedor Monti Juan cambiar la fecha por 10/10/1970 y agregarle una dirección de e-mail.
4. Al vendedor Sena Rosa, cambiar el teléfono por 4522221 y la dirección por Av. Vélez Sarsfield 125 - Centro.

5. Aumentar el precio unitario en un 15% pero de aquellos artículos cuyo precio unitario sea inferior a 20 (tener en cuenta además que la descripción comience con su legajo).
6. Aumentar el precio unitario en un 10% pero de aquellos artículos cuyo precio unitario esté entre a 20 y 30. (tener en cuenta además que la descripción comience con su legajo)

Tener en cuenta que si la actualización de una fila no cumple con una restricción o regla, infringe la configuración de valores NULL o si el nuevo valor es de un tipo de datos incompatible, se cancela la instrucción, se devuelve un error y no se actualiza ningún registro.

Problema N° 1.3:

Por errores en la carga de datos de algunos registros habrá que eliminarlos y para ello utilizaremos la:

Sentencia Delete

La sintaxis de DELETE es:

DELETE Nombre_tabla

[WHERE Condición]

Tener en cuenta que si no se especifica la cláusula WHERE, se borran todas las filas de una tabla.

Los registros a borrar son los siguientes:

1. Al vendedor cuyo apellido y nombre es "Sena Rosa".
2. Al artículo cuya descripción es "Repuesto Gloria rallado".
3. Elimine dos registros de los que ud. haya ingresado en los puntos anteriores.

Vamos a eliminar el vendedor "Sena Rosa" pero primero vamos a verificar su código de vendedor (la clave principal que es única) para asegurarnos de que sea el correcto, y para ello vamos a consultar los vendedores:

Select * from vendedores

Este es listado obtenido donde podemos ver que código de Sena Rosa es el 9

cod_vendedor	nom_vendedor	ape_vendedor	calle	altura	cod_sento	no_tel	e-mail	fec_nac
1	Martin	Carlos	San Lorenzo	363	2	NULL	mcampo@latemail.com	NULL
2	Maria	Ledemira	Chacabuco	1560	5	4526060	NULL	1979-02-22 00:00:00
3	Alfredo	Lopez	Alena	12	3	4812525	NULL	1975-03-08 00:00:00
4	Marcos	Ricardo	Rivers Indata	320	1	NULL	NULL	1986-10-07 00:00:00
5	Gabriel	Maria	Altoque	1205	6	4522122	NULL	1962-07-30 00:00:00
6	Subira	Juan	Arriaga	1795	9	458978	NULL	1962-07-30 00:00:00
7	Juan	Maria	Altoque	2145	5	NULL	NULL	NULL
8	Rosa	Sena	Av. Velaz Sanfeld	25	1	NULL	NULL	NULL

Ahora utilizamos la sentencia delete para eliminar el registro

delete vendedores

where cod_vendedor=9

Volvemos a consultar los vendedores:

Select *

from vendedores

Este es listado obtenido donde podemos ver que código de Sena Rosa es el 9

post_vendedor	nom_vendedor	ape_vendedor	calle	altura	cod_barrio	nro_tel	e-mail	fec_nac
1	Marín	Centzo	San Lorenzo	369	2	NULL	mcantzo@latinmail.com	NULL
2	Manela	Ledesma	Chachapoyas	1560	5	4526060	NULL	1979-02-22 01
3	Alejandro	Lopez	Alania	12	3	4612525	NULL	1975-03-06 01
4	Marcelo	Miranda	Rivers Indarte	320	1	NULL	NULL	1986-10-07 01
5	Gabriel	Monti	Altoaguire	1245	4	4522122	NULL	1982-07-30 01
6	Susana	Juarez	Antofagasta	1785	9	458976	NULL	1982-07-30 01
7	Juan	Monti	altoaguire	2145	5	NULL	NULL	NULL

Proceda del mismo modo para los otros dos registros.

Problema N° 2:

Los usuarios finales del sistema de la base de datos Librería necesitan obtener información para el funcionamiento del negocio y la toma de decisiones. Utilizaremos en este caso la sentencia select

La Sentencia SELECT

Esta sentencia, que se utiliza para expresar consultas, es la más potente y compleja de las sentencias de SQL. La sentencia SELECT recupera datos de una base de datos y los devuelve en forma de tablas que no quedan guardadas en la base. La sintaxis completa es la siguiente:.

Por ejemplo si lo que se necesita es obtener todos los datos de todos los clientes

Select *

from clientes

cod_cliente	nom_cliente	ape_cliente	calle	altura	cod_barrio	nro_tel	e-mail
1	Roberto	Perez	San Martín	120	1	NULL	NULL
2	Marta Analia	Castillo	Pedro Lucero	1450	7	NULL	castillo_marta@yahoo.com
3	Hector	Alonso	Los Góngora	160	12	4701314	hbalonso@hotmail.com
4	Santiago	Monde	Ladrón y Pizarro	55	2	155471516	NULL
5	Carlos Antonio	Perez	A. García	455	2	4554466	NULL
6	Pilar	Monde	Ladrón y Pizarro	55	2	155471516	NULL
7	Roque	Perez	Humberto Pérez	79	1	4262630	NULL
8	Elvira Josefa	Lopez	Manuelito Llanos	360	3	4502828	NULL
9	Marcos	Rios	Rivers Indarte	700	1	NULL	rios_marcos@hotmail.com
10	Roberto	Monti	Santa Rosa	71	1	NULL	NULL
11	Ana Maria	Perez	Av. Colón	1880	2	NULL	NULL
12	Adriana	Gonzalez	San Jerónimo	763	1	NULL	NULL

Otro ejemplo, si se desea obtener nombre, dirección y teléfonos de todos los clientes:

Select nom_cliente, ape_cliente, calle, altura, nro_tel

from clientes

Observe respetar el nombre exacto que cada campo tiene en la tabla de la base de datos al incluirla en la lista de selección

nom_cliente	ape_cliente	calle	altura	nro_tel
1	Roberto	Perez	San Martín	120
2	Marta Analia	Castillo	Pedro Lucero	1450
3	Hector	Alonso	Los Góngora	160
4	Santiago	Monde	Ladrón y Pizarro	55
5	Carlos Antonio	Perez	A. García	455
6	Pilar	Monde	Ladrón y Pizarro	55
7	Roque	Perez	Humberto Pérez	79
8	Elvira Josefa	Lopez	Manuelito Llanos	360
9	Marcos	Rios	Rivers Indarte	700
10	Roberto	Monti	Santa Rosa	71
11	Ana Maria	Perez	Av. Colón	1880
12	Adriana	Gonzalez	San Jerónimo	763

Si prestamos atención al encabezado de cada columna de los ejemplos anteriores vemos que aparece en nombre del campo origen del dato o bien columna sin nombre.

Podemos agregarle un alias a los campos que aparecerán como encabezados de columnas:

select cod_cliente Código, ape_cliente 'Apellido del cliente', nro_tel Teléfono

from clientes

Código	Apellido del cliente	Teléfono
1	Perez	NULL
2	Castillo	NULL
3	Alonso	4701314
4	Monde	155471516
5	Perez	4554466
6	Monde	155471516
7	Perez	4262630
8	Lopez	4502828
9	Rios	NULL
10	Monti	NULL

El alias va entre comillas salvo que no contenga caracteres especiales (como espacios, -, *, /, +, ?, =, etc.)

Además de las columnas cuyos valores provienen directamente de los campos de las tablas de la base de datos, se pueden incluir columnas calculadas cuyos valores surgen a partir de resolver expresiones.

La siguiente consulta calcula el subtotal de las ventas:

select nro_factura, cod_articulo, pre_unitario, cantidad, pre_unitario*cantidad Subtotal

from detalle_facturas

nro_factura	cod_articulo	pre_unitario	cantidad	Subtotal
555	458	15.50	25	337.50
1017	450	0	27.00	0.00
1051	450	0	12.00	0.00
555	451	16.00	28	336.00
100	401	0	12.50	0.00
101	402	0	25.00	0.00

cod_cliente	Clientes
1	Perez Roberto
2	Castillo Marta Analia
3	Alonso Hector
4	Monde Santiago
5	Perez Carlos Antonio
6	Monde Pilar
7	Perez Roque
8	Lopez Elvira Josefa

Si se quiere mostrar los datos de los clientes donde aparezca el apellido y nombre en la misma columna:

```
select cod_cliente,ape_cliente+' '+nom_cliente Cliente
from clientes
```

En algunos casos las consultas pueden dar como resultados filas duplicadas en el resultado de la consulta es decir dos o más filas exactamente iguales, como el caso de la siguiente consulta que muestra los artículos vendidos y el precio al cual se vendió:

```
select cod_articulo,pre_unitario
from detalle facturas
```

id	cod_cliente	preço
10	1	15.50
11	1	15.50
20	1	15.50
21	1	15.50
22	1	15.50
23	1	15.50
24	1	15.50
25	1	15.50
26	2	2.00
27	2	2.00
28	2	2.00
29	2	2.00
30	2	2.00
31	2	2.00
32	2	2.00
33	2	2.00

	cod_categoria	pre_unitario
5	1	14.50
6	1	15.00
7	1	15.50
8	1	15.90
9	2	22.00
10	2	3.00
11	2	4.00
12	2	12.00
13	2	22.00
14	2	22.30
15	2	22.50
16	2	45.00
17	3	4.00
18	3	4.00
19	3	3.00

Para eliminar las filas duplicadas utilizamos el predicado DISTINCT con lo que cada código de artículo con su precio de venta aparecería una sola vez, entonces la consulta quedaría:

```
select distinct cod_articulo,pre_unitario
from detalle facturas
```

Se quiere listar las facturas emitidas antes del 10/7/2008; vamos a utilizar en la cláusula where un test de comparación de la siguiente forma:

```
Select *
from facturas
Where fecha < '10/07/2008'
```

row_id		col_datetime	col_double	col_smallint
35	35	2005-02-03 00:00:00	5	1
36	36	2005-02-11 00:00:00	7	4
37	37	2005-02-24 00:00:00	1	3
38	38	2005-02-02 00:00:00	6	2
39	39	2005-02-13 00:00:00	8	4
40	40	2005-02-19 00:00:00	6	1
41	41	2005-02-12 00:00:00	4	1
42	42	2005-02-19 00:00:00	7	2
43	43	2005-02-03 00:00:00	4	1

Para encontrar por ejemplo los artículos cuyo precio esté entre 50 y 100 pesos podemos utilizar dos test de comparación o bien un test de rango de esta forma:

Select *
from articulos
Where pre_un

	cod_producto	descripcion	stock_real	pre_unitario	observaciones
1	16	Catuchera Lata	60	65.50	2 pzas
2	26	Lápices Coloridos * 12 FABER	NUL	70.99	NUL

El test de pertenencia al conjunto examina si un valor de dato coincide con uno de una lista de valores objetivos. Por ejemplo listar los clientes cuyo código sean los siguientes: 1,3, 7, 8 y 12

Select *

from clientes

	cod_cliente	nom_cliente	ape_cliente	calle	altura	cod_bairro	ren_tel	e-mail
1	5	Rodolfo	Pereira	San Martin	120	1	N.I.L.L.	N.I.L.L.
2	3	Hector	Alonso	Los Gorgones	150	12	4701314	halsonso@hotmail.com
3	7	Floque	Perez	Humberto Pires	70	1	4262630	N.I.L.L.
4	8	Brito Josefa	Lopez	Manuelo Guadalupe	360	3	4503229	N.I.L.L.
5	12	Adriana	Gonzalez	San Andres	763	1	N.I.L.L.	N.I.L.L.

Where cod cliente in (1,3,7,8,12)

Si se quiere listar aquellos registros donde el valor de dato de una columna se ajusta a un patrón especificado utilizando uno o más caracteres comodines, por ejemplo, listar todos los artículos que comiencen con “L” se puede utilizar un test de correspondencia con patrón como se muestra en el siguiente ejemplo:

	cod_articulo	descripcion	stock_promio	pre_promio	observaciones
1	5	Lámp. Evolution H22 * 4 u	NULX	15.50	NULX
2	4	Lámpicas Color corbe * 12 u	54	20.50	NULX
3	6	Lámpicas Color lagos * 12 u	NULX	27.90	NULX
4	22	Lámpicas con goma * 2 u	NULX	15.50	NULX
5	25	Lámpicas Color lagos * 12 FABER	NULX	79.90	NULX
6	26	Lámpicas Bio Azul Inca for	NULX	4.90	NULX
7	28	Lámpicas Color lagos x 12u.	NULX	101.50	NULX

Select *

from articulos

Where descripcion like 'L%'

Con el test de valor nulo podemos comprobar si un campo contiene o no valores Null en una condición de búsqueda. Por ejemplo listar los artículos para los que no existan observaciones

Select *

from articulos

Where observaciones is null

cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	Lápis Evolutor HB2 "4 u	NULL	15.50	NULL
2	Papel p-forma Fantás "10 u	130	22.30	NULL
3	Lápices Color contos "12 u	54	25.50	NULL
4	Fibras contos "6	40	25.30	NULL
5	Lápices Color largos "12 u	NULL	27.30	NULL
6	Adhesivo sintético 35 gr	NULL	6.00	NULL
7	Cuaderno tamaño invitado rayado	30	12.70	NULL
8	Cuaderno tamaño invitado cuadrado	35	12.70	NULL
9	Catuchera de Tela	55	32.60	NULL

En todas las condiciones de búsqueda se puede utilizar un NOT para seleccionar filas en donde la condición de búsqueda es falsa. Por ejemplo listar los artículos cuyo precio no esté entre 10 y 100

Select *

from articulos

Where pre_unitario not between 10 and 100

cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	Adhesivo sintético 35 gr	NULL	6.00	NULL
2	Repuesto Glora rayado	120	141.00	400 hojas
3	Repuesto Glora cuadrado	50	141.00	400 hojas
4	Resma hoja A4	40	185.00	NULL
5	Lápices Bic Azul trazo fino	NULL	4.99	NULL
6	Lápices Color largos x 12u	NULL	101.50	NULL

Utilizando las reglas de la lógica, se pueden combinar estas condiciones de búsqueda simples para formar otras más complejas utilizando los operadores lógicos OR y AND

OR: Se utiliza para combinar dos condiciones de búsqueda cuando una o la otra o ambas deban ser ciertas:

Select *

from articulos

Where stock_minimo > 100 or pre_unitario > 80

cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	Papel p-forma Fantás "10 u	130	22.30	NULL
2	Papel p-forma Ana "10 u	150	25.50	Color rojo - azul - verde
3	Repuesto Glora rayado	120	141.00	400 hojas
4	Repuesto Glora cuadrado	50	141.00	400 hojas
5	Resma hoja A4	40	185.00	NULL
6	Lápices Color largos x 12u	NULL	101.50	NULL

En el ejemplo se listarán los registros cuyo stock minimo sea mayor a 100 o cuyo precio mayor a 80. Preste atención al 1er. registro, cumple con la condición del stock mínimo mayor a 100 pero no cumple con la del precio unitario mayor a 80; el registro 3 cumple con ambas condiciones y el registro cuatro solo cumple la condición del precio unitario.

AND Se utiliza para combinar dos condiciones de búsqueda que deban ser ciertas simultáneamente, como es el caso del listado de vendedores cuyo nombre comience con A y nacidos antes de 1980:

Select *

from vendedores

Where nom_vendedor like 'A%' and fec_nac < '1/1/1980'

	cod_vendedor	nom_vendedor	ape_vendedor	calle	altura	cod_barrio	nro_tel	e-mail	fec_nac
1	3	Alejandro	Lopez	Alina	12	3	4612525	NULL	1975-03-06 00:00:00

En este caso solo se muestra el registro que cumple con ambas condiciones

Las filas de los resultados de una consulta no están dispuestas en ningún orden particular. Se puede pedir a SQL que ordene los resultados de una consulta incluyendo la cláusula Order By en la sentencia Select. Por ejemplo si se quiere listar los artículos ordenados por precio y descripción:

Select *

from articulos

order by pre_unitario, descripcion

cod_articulo	descripcion	stock_minimo	pre_unitario	observaciones
1	Resma hoja A4	40	185.00	NULL
2	Repuesto Glora rayado	120	141.00	400 hojas
3	Repuesto Glora cuadrado	50	141.00	400 hojas
4	Lápices Color largos x 12u	NULL	101.50	NULL
5	Lápices Color largos "12 FABER	NULL	70.99	NULL
6	Catuchera Lata	60	65.50	2 piezas
7	Potomero	NULL	48.90	NULL
8	Conjunto Geométrico	NULL	35.50	Regla - escuadra - transportador
9	Cuaderno tapa dura rayado	20	32.59	NULL
10	Catuchera de Tela	55	32.60	NULL
11	Correctores Bic Lápis "1 u	NULL	31.40	NULL
12	Goma para lapiceros "10 u	NULL	25.50	NULL
13	Lápices Color largos "12 u	NULL	27.30	NULL

Por omisión SQL ordena los datos en secuencia ascendente. Para ordenar en secuencia descendente se incluye la palabra DESC en la especificación de la ordenación a la derecha de cada columna que se quiera ordenar en forma descendente como ordenar los artículos por precio en forma descendente

Select *

from articulos

order by pre_unitario desc

La cláusula Order by admite que en lugar de usar el nombre de columna se pueda utilizar también el número de columna o el alias de la misma.

Microsoft SQL Server tiene algunas funciones incorporadas para trabajar con cadenas de caracteres, fechas y funciones matemáticas cuya explicación y ejemplo se pueden encontrar en el material teórico de la materia y en el material complementario.

Ahora estamos en condiciones de resolver el Problema N° 2: Los usuarios finales del sistema de la base de datos Librería necesitan obtener información para el funcionamiento del negocio y la toma de decisiones. Lo que se necesita es lo siguiente:

4. Se quiere listar todos los datos de los clientes con nro. de teléfono conocidos. Muestre el nombre y apellido en una misma columna. Ordene por nombre completo. Rotule en forma conveniente.
5. Se quiere saber el subtotal de todos los artículos vendidos, para ello liste el código y multiplique la cantidad por el precio unitario (de la tabla: detalle_facturas). Ordene por código en forma ascendente y subtotal en forma descendente. No muestre datos duplicados.
6. Muestre el código, nombre, apellido (todo el apellido en mayúsculas) y dirección (calle y altura en una sola columna; para la altura utilice una función de conversión) de todos los clientes cuyo nombre comience con "C" y cuyo apellido termine con "Z". Rotule como CÓDIGO DE CLIENTE, NOMBRE, DIRECCIÓN.
7. Ídem al anterior pero el apellido comience con letras que van de la "D" a la "L" y cuyo nombre no comience con letras que van de la "A" a la "G".
8. Liste los artículos cuyo precio sea menor a 40 y sin observaciones. Ordene por descripción y precio ambos descendente.
9. Muestre los datos de los vendedores cuyo nombre no contenga "Z" y cuya fecha de nacimiento sea posterior a 1/1/1970.
10. Mostrar las facturas realizadas entre el 1/1/2007 y el 1/5/2009 y cuyos códigos de vendedor sean 1, 3 y 4 o bien entre el 1/1/2010 y el 1/5/2011 y cuyos códigos de vendedor sean 2 y 4.
11. Muestre las ventas (tabla detalle_facturas) de los artículos cuyo precio unitario sea mayor o igual a 10 o cuyos códigos de artículos no sea uno de los siguientes: 2, 5, 6, 8, 10. En ambos casos que los números de facturas oscilen entre 50 y 100.
12. Listar todos los datos de los artículos cuyo stock mínimo sea superior a 10 o cuyo precio sea inferior a 20. En ambos casos su descripción no debe comenzar con las letras "p" ni la letra "r".
13. Listar los datos de los vendedores nacidos en febrero, abril, mayo o septiembre.

Problema N° 3

Los usuarios finales del sistema necesitan obtener la siguiente información para el funcionamiento del negocio y la toma de decisiones:

1. Listado de vendedores y el barrio en el que viven

Para ello es necesario realizar una consulta que involucre dos tablas la de vendedores y la de barrios. Aquí es donde aplicamos composiciones (o combinación de tablas)

Para realizar en SQL composiciones multitabla se puede utilizar la sentencia SELECT con una condición de búsqueda que especifique la comparación de columnas de tablas diferentes; previamente la cláusula FROM lista las dos tablas intervinientes.

Entonces la sentencia SELECT para la solución del Problema 3.1, sería la siguiente:

```
select ape_vendedor+' '+nom_vendedor Vendedor, barrio
from vendedores, barrios
where vendedores.cod_barrio=barrios.cod_barrio
```

Genera resultados solo para los pares de filas en los que el número de barrio (cod_barrio) de la tabla barrios coincide con el número de barrio de la tabla vendedores.

Hay que tener en cuenta que si un campo tiene el mismo nombre en dos tablas diferentes se debe anteponer al mismo el nombre de la tabla a la cual pertenece separado por un punto.

Podemos darle alias a las tablas en el FROM para evitar escribir tantas veces el nombre de la misma teniendo en cuenta que cada alias en cada consulta debe ser único. En el ejemplo anterior le vamos a dar un alias a vendedores "v" y un alias a barrios "b"

Results		Messages
Vendedor	barrio	
1 Camizo Martín	ALTO ALBERDI	
2 Ledesma Mariela	GENERAL PAZ	
3 Lopez Alejandro	OBSERVATORIO	
4 Miranda Marcelo	CENTRO	
5 Monti Gabriel	JARDIN	
6 Juarez Susana	SAN VICENTE	
7 Ortega Ana	ALTO ALBERDI	
8 Monti Juan	GENERAL PAZ	

```
select ape_vendedor+' '+nom_vendedor Vendedor, barrio
from vendedores v, barrios b
where v.cod_barrio=b.cod_barrio
```

SQL puede combinar tres o más tablas utilizando las mismas técnicas básicas utilizadas para las consultas de dos tablas. La siguiente consulta lista los artículos facturados con sus respectivas facturas

```
select f.nro_factura, fecha, descripcion, cantidad, d.pre_unitario,
cantidad*d.pre_unitario Subtotal
from facturas f, detalle_facturas d, articulos a
where f.nro_factura = d.nro_factura
and a.cod_articulo = d.cod_articulo
```

Results		Messages				
	nro_factura	fecha	descripcion	cantidad	pre_unitario	Subtotal
1	1	2002-02-02 00:00:00	Cartuchera Lata	30	2.00	60.00
2	1	2002-02-02 00:00:00	Goma para lápiz * 10 u	20	1.00	20.00
3	2	2002-03-13 00:00:00	Lápices Color largos * 12 u	15	6.00	90.00
4	2	2002-03-13 00:00:00	Cartuchera Lata	10	1.00	10.00
5	3	2002-04-30 00:00:00	Sepadores tamaño invadavia * 6 u	20	1.00	20.00
6	3	2002-04-30 00:00:00	Carteta fibra negra - Tamaño invadavia	30	1.00	30.00
7	4	2002-05-03 00:00:00	Cartuchera Lata	30	1.00	30.00
8	4	2002-05-03 00:00:00	Correctores Bic Lápiz * 1 u	24	5.00	120.00
9	5	2002-06-06 00:00:00	Cuaderno tamaño invadavia rayado	20	1.00	20.00
10	5	2002-06-06 00:00:00	Correctores Bic Lápiz * 1 u	10	4.00	40.00
11	6	2002-09-03 00:00:00	Lápices Color cortos * 12 u	33	2.00	66.00

La condición de búsqueda que especifica las columnas de emparejamiento en una consulta multitabla puede combinarse con otras condiciones de búsqueda para restringir aún más los contenidos de los resultados. Por ejemplo si se quiere listar las facturas del mes de mayo de 2015 con sus clientes:

```
select f.nro_factura, fecha, ape_cliente+' '+nom_cliente Cliente
from facturas f, clientes c
where f.cod_cliente=c.cod_cliente
and month(fecha)=5 and year(fecha)=2015
```

Results		Messages
nro_factura	fecha	Cliente
1 149	2015-05-12 00:00:00	Abarca Héctor
2 150	2015-05-15 00:00:00	Abarca Héctor
3 151	2015-05-26 00:00:00	Paez Roque
4 181	2015-05-10 00:00:00	Morales Santiago

Continuando con el problema 3 y habiendo visto como crear consultas multitablas, resolver todos requerimientos de los usuarios del sistema

- Liste número de factura, fecha de venta y vendedor (apellido y nombre), para los casos en que el código del cliente van del 2 al 6. Ordene por vendedor y fecha, ambas en forma descendente.
- Emitir un reporte con los datos de la factura del cliente y del vendedor de aquellas facturas confeccionadas entre el primero de febrero del 2008 y el primero de marzo del 2010 y que el apellido del cliente no contenga "C".
- Listar los datos de la factura, los del artículo y el importe (precio por cantidad); para las facturas emitidas en el 2009, 2010 y 2012 y la descripción no comience con "R". Ordene por número de factura e importe, este en forma descendente. Rotule.
- Se quiere saber qué artículos se vendieron, siempre que el precio al que fue vendido no esté entre \$10 y \$50. Rotule como: Código de Artículo, Descripción, Cantidad e Importe (El importe es el precio por la cantidad).

6. Liste todos los datos de la factura (vendedor, cliente, artículo, incluidos los datos de la venta); emitidas a clientes con teléfonos o direcciones de e-mail conocidas de aquellas facturas cuyo importe haya sido superior a \$250. Agregue rótulos presentación y ordene el listado para darle mejor presentación.
7. Se quiere saber a qué cliente, de qué barrio, vendedor y en qué fecha se les vendió con los siguientes nros. de factura: 12, 18, 1, 3, 35, 26 y 29.
8. Emitir un reporte para informar qué artículos se vendieron, en las facturas cuyos números no esté entre 17 y 136. Liste la descripción, cantidad e importe. Ordene por descripción y cantidad. No muestre las filas con valores duplicados
9. Listar los datos de las facturas (cliente, artículo, incluidos los datos de la venta incluido el importe) emitidas a los clientes cuyos apellidos comiencen con letras que van de la "l" a "s" o los artículos vendidos que tengan descripciones que comiencen con las mismas letras. Ordenar el listado.
10. Realizar un reporte de los artículos que se vendieron en lo que va del año. (Muestre los datos que sean significativos para el usuario del sistema usando rótulos para que sea más legible y que los artículos no se muestren repetidos).
11. Se quiere saber a qué clientes se les vendió el año pasado, qué vendedor le realizó la venta, y qué artículos compró, siempre que el vendedor que les vendió sea menor de 35 años.
12. El usuario de este sistema necesita ver el listado de facturas, de aquellos artículos cuyos precios unitarios a los que fueron vendidos estén entre 50 y 100 y de aquellos vendedores cuyo apellido no comience con letras que van de la "l" a la "m". Ordenado por vendedor, fecha e importe.
13. Se desea emitir un listado de clientes que compraron en enero, además saber qué compraron cuánto gastaron (mostrar los datos en forma conveniente)
14. Emitir un reporte de artículos vendidos en el 2010 a qué precios se vendieron y qué precio tienen hoy.
15. Listar los vendedores que hace 10 años le vendieron a clientes cuyos nombres o apellidos comienzan con "C"
16. El encargado de la librería necesita tener información sobre los artículos que se vendían a menos de \$ 10 antes del 2015. Mostrar los datos que se consideren relevantes para el encargado, rotular y ordenar.