

TEMA 4: El lenguaje SQL II: Introducción, modificación y borrado de datos.

4.1 Actualización de datos en SQL.

SQL es un lenguaje completo de manipulación de datos que no solo se utiliza para consultas, sino también para modificar y actualizar los datos de la base de datos. Comparadas con la complejidad de la sentencia SELECT, las sentencias de SQL que modifican los contenidos de la base de datos son sencillas. En este tema se describen las tres sentencias SQL que se emplean para modificar los datos de una base de datos:

- INSERT, que añade nuevas filas de datos a una tabla.
- DELETE, que elimina filas de datos de una tabla.
- UPDATE, que modifica datos existentes en la base de datos.

4.2 Adición de datos a la base de datos.

Una nueva fila de datos se añade a una base de datos relacional cuando una nueva entidad representada por la fila “aparece en el mundo exterior”. Por ejemplo, se contrata un nuevo vendedor, se firma con un nuevo cliente, se realiza un nuevo pedido, etc. En cada caso, la nueva fila se añade para mantener la base de datos como un modelo preciso del mundo real. En general, un DBMS basado en SQL proporciona tres maneras de añadir nuevas filas de datos a una base de datos:

- Una sentencia INSERT de una fila que añade una única nueva fila de datos a una tabla. Su uso es habitual en aplicaciones diarias, por ejemplo, en programas de entrada de datos.
- Una sentencia INSERT multifila, que extrae filas de datos de otra parte de la base y las añade a una tabla.
- Una utilidad de carga masiva, que añade datos a una tabla desde un fichero externo a la base de datos. Se utiliza habitualmente para cargar inicialmente la base de datos o para incorporar datos transferidos desde otro sistema informático o recolectados desde muchas localizaciones.

4.2.1 La sentencia INSERT de una fila.

La sentencia INSERT de una fila añade una nueva fila a una tabla. Su diagrama sintáctico puede verse en la figura 4.2.1.1. La cláusula INTO especifica la tabla que recibe la nueva fila (la tabla destino) y la cláusula VALUE especifica los valores de datos que la nueva fila contendrá. La lista de columnas indica qué valor va a qué columna de la nueva fila.

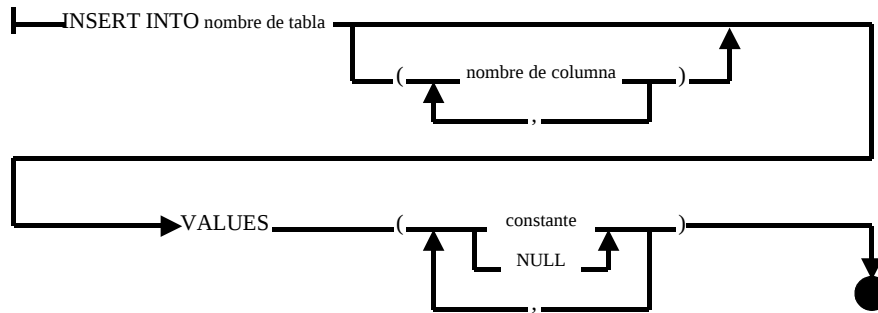


Figura 4.2.1.1: Diagrama sintáctico de la sentencia INSERT de una fila.

Supongamos que en nuestra base de datos ejemplo, se acaba de contratar a un nuevo vendedor con los siguientes datos:

Nombre: Enrique Jordan
 Edad: 36
 Número de empleado: 111
 Título: Director de ventas
 Oficina: Alicante (número de oficina 13)
 Fecha de contrato: 25 de Julio de 2000.
 Cuota: Aún no asignada.
 Ventas anuales hasta la fecha: 0

La sentencia INSERT que añade el nuevo vendedor a la base de datos es:

```
INSERT INTO repventas(nombre, edad, num_empl, ventas, titulo, contrato,
oficina_rep) VALUES('Enrique Jordan',36,111,0,'Dir. Ventas','25/07/2000',13)
```

Quedando la tabla REPVENTAS de la siguiente forma:

NUM_EMPL	NOMBRE	EDAD	OFICINA_REP	TITULO	CONTRATO
106	Jose Maldonado	52	11	VP Ventas	14/06/1998
104	Carlos Martinez	33	12	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999
108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	12	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	12	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
111	Enrique Jordan	36	13	Dir. Ventas	25/07/2000
DIRECTOR	CUOTA	VENTAS			
NULL	25.000	32.958			
106	17.500	0			
104	30.000	39.327			
106	27.500	7.105			
106	30.000	58.533			
108	30.000	22.776			
104	27.500	26.628			
101	NULL	23.123			
104	25.000	2.700			
108	27.500	34.432			
NULL	NULL	0			

Tabla 4.2.1.1: La Tabla REPVENTAS al incluir el nuevo empleado.

Supongamos ahora que el nuevo representante recibe ahora su primer pedido, de Corporacion Oeste S.A., un nuevo cliente que tiene asignado el número de cliente 2126. El pedido es de 20 "Artículo Tipo 4" ACI-41004 por un precio total de 2.340 y le ha sido asignado el número de pedido 113069. Las sentencias INSERT que añaden el nuevo cliente y pedido a la base de datos son:

```
INSERT INTO clientes(empresa, num_clie, limite_credito, rep_clie) VALUES
('Corporacion Oeste S.A.', 2126, 15000, 111)
```

```
INSERT INTO pedidos(importe, fab, producto, cant, fecha_pedido, num_pedido, clie,
rep) VALUES (2340, 'ACI', '41004', 20, CURRENT DATE, 113070, 2126, 111)
```

Quedando las tablas de la siguiente forma:

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	105	50.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	106	35.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	25.000
2105	MALB S.A.	101	45.000
2126	Corporacion Oeste S.A.	111	15.000

Tabla 4.2.1.2: La tabla CLIENTES al incluir el nuevo cliente.

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
112961	17/12/1999	2117	106	REI	2A44L	7	31.500
113012	11/01/2000	2111	105	ACI	41003	35	3.745
112989	03/01/2000	2101	106	FEA	114	6	1.458
113051	10/02/2000	2118	108	QSA	XK47	4	1.420
112968	12/10/1999	2102	101	ACI	41004	34	3.978
110036	30/01/2000	2107	110	ACI	4100Z	9	22.500
113045	02/02/2000	2112	108	REI	2A44R	10	45.000
112963	17/12/1999	2103	105	ACI	41004	28	3.276
113013	14/01/2000	2118	108	BIC	41003	1	652
113058	23/02/2000	2108	109	FEA	112	10	1.480
112997	08/01/2000	2124	107	BIC	41003	1	652
112983	27/12/1999	2103	105	ACI	41004	6	702
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
113062	24/02/2000	2124	107	FEA	114	10	2.430

112979	12/10/1999	2114	102	ACI	4100Z	6	15.000
113027	22/02/2000	2103	105	ACI	41002	54	4.104
113007	08/01/2000	2112	108	IMM	773C	3	2.825
113069	02/03/2000	2109	107	IMM	775C	22	31.350
113034	29/01/2000	2107	110	REI	2A45C	8	632
112992	04/11/1999	2118	108	ACI	41002	10	760
112975	12/10/1999	2111	103	REI	2A44G	6	2.100
113055	15/02/2000	2108	101	ACI	4100X	6	150
113048	10/02/2000	2120	102	IMM	779C	2	3.750
112993	04/01/2000	2106	102	REI	2A45C	24	1.896
113065	27/02/2000	2106	102	QSA	XK47	6	2.130
113003	25/01/2000	2108	109	IMM	779C	3	5.625
113049	10/02/2000	2118	108	QSA	XK47	2	776
112987	31/12/1999	2103	105	ACI	4100Y	11	27.500
113057	18/02/2000	2111	103	ACI	4100X	24	600
113042	02/02/2000	2113	101	REI	2A44R	5	22.500
113070	22/10/2000	2126	111	ACI	41004	20	2.340

Tabla 4.2.1.3: La tabla PEDIDOS al incluir el nuevo pedido.

Conviene resaltar el hecho de que existe una sentencia `CURRENT DATE` que permite introducir el día actual sin necesidad de introducirlo como fecha. Además, SQL automáticamente asigna un valor `NULL` a cualquier columna cuyo nombre falte de la lista, aunque es posible especificar de forma explícita que una columna tiene valor `NULL`. Así, cuando hemos introducido el nuevo vendedor podríamos haberlo realizado como:

```
INSERT INTO repventas(nombre, edad, num_empl, ventas, titulo, director, cuota,
contrato, oficina_rep) VALUES('Enrique Jordan',36,111,0,'Dir. Ventas', NULL, NULL,
'25/07/2000',13)
```

Además, SQL permite omitir la lista de columnas de la sentencia `INSERT`. Cuando se omite la lista de columnas, SQL genera automáticamente una lista formada por todas las columnas de la tabla, en secuencia de izquierda a derecha. Esta es la misma secuencia de columnas generadas por SQL cuando se utiliza una consulta `SELECT *`. Utilizando esta forma, la sentencia `INSERT` anterior podría escribirse como:

```
INSERT INTO repventas VALUES(111, 'Enrique Jordan', 36, 13, 'Dir. Ventas',
'25/07/2000', NULL, NULL, 0)
```

4.2.2 La sentencia `INSERT` multifila.

La segunda forma de la sentencia `INSERT` añade múltiples filas de datos a su tabla destino. Su diagrama sintáctico puede verse en la figura 4.2.2.1. En esta forma de la sentencia `INSERT`, los valores de datos para las nuevas filas no son especificados explícitamente dentro del texto de la sentencia. En su lugar, la fuente de las nuevas filas es una consulta de base de datos, especificada en la sentencia.

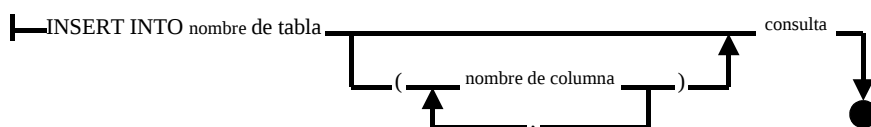


Figura 4.2.2.1: Diagrama sintáctico de la sentencia `INSERT` multifila.

La adición de filas cuyos valores provienen de la propia base de datos puede parecer un poco extraña, pero es muy útil en ocasiones especiales. Por ejemplo, supongamos que se desea copiar el número de pedido, la fecha y el importe de todos los pedidos remitidos con anterioridad al 1 de Enero de 2000, a partir de la tabla PEDIDOS en otra tabla, llamada ANTPEDIDOS. La sentencia INSERT multifila proporciona un modo de copiar los datos.

```
INSERT INTO antpedidos(num_pedido, fecha_pedido, importe) SELECT num_pedido,
    fecha_pedido, importe FROM pedidos WHERE fecha_pedido<'01/01/2000'
```

NUM_PEDIDO	FECHA_PEDIDO	IMPORTE
112961	17/12/1999	31.500
112968	12/10/1999	3.978
112963	17/12/1999	3.276
112983	27/12/1999	702
112979	12/10/1999	15.000
112992	04/11/1999	760
112975	12/10/1999	2.100
112987	31/12/1999	27.500

Tabla 4.2.2.1: La tabla ANTPEDIDOS después de la inserción múltiple.

El estándar SQL especifica varias restricciones sobre la consulta que aparece dentro de la sentencia INSERT multifila:

- La consulta no puede contener una cláusula ORDER BY.
- La consulta no puede ser la UNION de varias sentencias SELECT diferentes.
- La tabla destino de la sentencia INSERT no puede aparecer en la cláusula FROM de la consulta o de ninguna subconsulta que ésta contenga.

Además, los resultados de la consulta deben contener el mismo número de columnas que la lista de columnas de la sentencia INSERT (o de la tabla destino entera, si se ha omitido la lista de columnas), y los tipos de datos deben ser compatibles, columna a columna.

4.2.3 Utilidades de carga masiva.

Los datos a insertar en una base de datos son con frecuencia extraídos de otro sistema informático o recolectados de otros lugares y almacenados en un archivo secuencial. Para cargar los datos en una tabla, se podría escribir un programa con un bucle que leyera cada registro del archivo y utilizara la sentencia INSERT de una fila para añadir la fila a la tabla. Sin embargo, el ejecutar repetidamente la sentencia INSERT de una fila produce un costo de computación, y por tanto un tiempo de ejecución, bastante elevado.

Por esta razón, los productos DBMS comerciales incluyen una capacidad de carga masiva que carga los datos desde un archivo a una tabla a alta velocidad. El estándar SQL no considera esta función, y suele ser suministrada como un programa de utilidad autónomo en lugar de forma parte del lenguaje SQL. La utilidad que cada

vendedor suministra dispone de un conjunto diferente de características, funciones y órdenes.

4.3 Supresión de datos de la base de datos.

Típicamente, una fila de datos se suprime de una base de datos cuando la entidad representada por la fila “desaparece del mundo exterior”. Por ejemplo, un cliente cancela un pedido, un vendedor abandona la empresa, etc. En cada caso, la fila se suprime para mantener la base de datos como un modelo preciso del mundo real. La unidad más pequeña de datos que puede ser suprimida de una base de datos relacional es una única fila.

4.3.1 La sentencia DELETE.

La sentencia DELETE, cuyo diagrama sintáctico se muestra en la figura 4.3.1.1, elimina filas seleccionadas de datos de una única tabla.

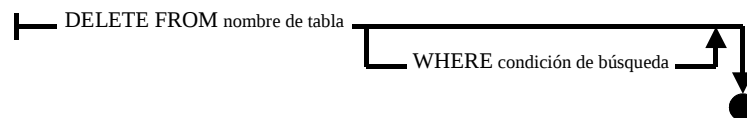


Figura 4.3.1.1: Diagrama sintáctico de la sentencia DELETE.

La cláusula FROM especifica la tabla destino que contiene las filas. La cláusula WHERE especifica qué filas de la tabla van a ser suprimidas.

Supongamos que Corporacion Oeste S.A. (número de cliente 2126) decide cancelar todos los pedidos. Podemos eliminar sus pedidos de la tabla PEDIDOS mediante la orden:

DELETE FROM pedidos WHERE clie=2126

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
112961	17/12/1999	2117	106	REI	2A44L	7	31.500
113012	11/01/2000	2111	105	ACI	41003	35	3.745
112989	03/01/2000	2101	106	FEA	114	6	1.458
113051	10/02/2000	2118	108	QSA	XK47	4	1.420
112968	12/10/1999	2102	101	ACI	41004	34	3.978
110036	30/01/2000	2107	110	ACI	4100Z	9	22.500
113045	02/02/2000	2112	108	REI	2A44R	10	45.000
112963	17/12/1999	2103	105	ACI	41004	28	3.276
113013	14/01/2000	2118	108	BIC	41003	1	652
113058	23/02/2000	2108	109	FEA	112	10	1.480
112997	08/01/2000	2124	107	BIC	41003	1	652
112983	27/12/1999	2103	105	ACI	41004	6	702
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
113062	24/02/2000	2124	107	FEA	114	10	2.430
112979	12/10/1999	2114	102	ACI	4100Z	6	15.000
113027	22/02/2000	2103	105	ACI	41002	54	4.104
113007	08/01/2000	2112	108	IMM	773C	3	2.825
113069	02/03/2000	2109	107	IMM	775C	22	31.350
113034	29/01/2000	2107	110	REI	2A45C	8	632
112992	04/11/1999	2118	108	ACI	41002	10	760
112975	12/10/1999	2111	103	REI	2A44G	6	2.100
113055	15/02/2000	2108	101	ACI	4100X	6	150

113048	10/02/2000	2120	102	IMM	779C	2	3.750
112993	04/01/2000	2106	102	REI	2A45C	24	1.896
113065	27/02/2000	2106	102	QSA	XK47	6	2.130
113003	25/01/2000	2108	109	IMM	779C	3	5.625
113049	10/02/2000	2118	108	QSA	XK47	2	776
112987	31/12/1999	2103	105	ACI	4100Y	11	27.500
113057	18/02/2000	2111	103	ACI	4100X	24	600
113042	02/02/2000	2113	101	REI	2A44R	5	22.500

Tabla 4.3.1.1: La tabla PEDIDOS después de eliminar los pedidos del cliente 2126.

Si ahora 'Enrique Jordan', el nuevo vendedor, decide abandonar la empresa con los clientes que posee, las sentencias DELETE que eliminan sus clientes y su fila de la tabla REPVENTAS son:

DELETE FROM clientes WHERE rep_clie=111

DELETE FROM repventas WHERE nombre='Enrique Jordan'

Quedando las tablas como:

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	105	50.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	106	35.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	25.000
2105	MALB S.A.	101	45.000

Tabla 4.3.1.2: La tabla CLIENTES después de eliminar los clientes del representante número 111.

NUM_EMPL	NOMBRE	EDAD	OFICINA_REP	TITULO	CONTRATO
106	Jose Maldonado	52	11	VP Ventas	14/06/1998
104	Carlos Martinez	33	12	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999
108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	12	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	12	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
DIRECTOR	CUOTA	VENTAS			

NULL	25.000	32.958
106	17.500	0
104	30.000	39.327
106	27.500	7.105
106	30.000	58.533
108	30.000	22.776
104	27.500	26.628
101	NULL	23.123
104	25.000	2.700
108	27.500	34.432

Tabla 4.3.1.3: La Tabla REPVENTAS después de eliminar el vendedor 'Enrique Jordan'.

La cláusula WHERE puede seleccionar varias filas de una tabla SQL y eliminar todas las filas seleccionadas. Esto puede verse en el siguiente ejemplo:

DELETE FROM pedidos WHERE fecha_pedido < '15/11/1999'

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
112961	17/12/1999	2117	106	REI	2A44L	7	31.500
113012	11/01/2000	2111	105	ACI	41003	35	3.745
112989	03/01/2000	2101	106	FEA	114	6	1.458
113051	10/02/2000	2118	108	QSA	XK47	4	1.420
110036	30/01/2000	2107	110	ACI	4100Z	9	22.500
113045	02/02/2000	2112	108	REI	2A44R	10	45.000
112963	17/12/1999	2103	105	ACI	41004	28	3.276
113013	14/01/2000	2118	108	BIC	41003	1	652
113058	23/02/2000	2108	109	FEA	112	10	1.480
112997	08/01/2000	2124	107	BIC	41003	1	652
112983	27/12/1999	2103	105	ACI	41004	6	702
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
113062	24/02/2000	2124	107	FEA	114	10	2.430
113027	22/02/2000	2103	105	ACI	41002	54	4.104
113007	08/01/2000	2112	108	IMM	773C	3	2.825
113069	02/03/2000	2109	107	IMM	775C	22	31.350
113034	29/01/2000	2107	110	REI	2A45C	8	632
113055	15/02/2000	2108	101	ACI	4100X	6	150
113048	10/02/2000	2120	102	IMM	779C	2	3.750
112993	04/01/2000	2106	102	REI	2A45C	24	1.896
113065	27/02/2000	2106	102	QSA	XK47	6	2.130
113003	25/01/2000	2108	109	IMM	779C	3	5.625
113049	10/02/2000	2118	108	QSA	XK47	2	776
112987	31/12/1999	2103	105	ACI	4100Y	11	27.500
113057	18/02/2000	2111	103	ACI	4100X	24	600
113042	02/02/2000	2113	101	REI	2A44R	5	22.500

Tabla 4.3.1.4: La tabla PEDIDOS después de eliminar los pedidos remitidos antes del 15 de Noviembre de 1999.

4.3.2 Supresión de todas las filas.

La cláusula WHERE de la sentencia DELETE es opcional, pero casi siempre está presente. Si se omite la cláusula WHERE de una sentencia DELETE, se suprimen todas las filas de la tabla destino, como en el ejemplo siguiente:

DELETE FROM pedidos

Aunque esta sentencia DELETE produce una tabla vacía, no borra la tabla PEDIDOS de la base de datos. La definición de la tabla PEDIDOS y sus columnas siguen estando almacenadas en la base de datos. La tabla aún existe, y nuevas filas pueden ser insertadas en la tabla PEDIDOS con la sentencia INSERT.

4.3.3 DELETE con subconsulta.

Las sentencias DELETE con condiciones de búsqueda simples, como las de los ejemplos anteriores, seleccionan las filas a suprimir basándose únicamente en los propios contenidos de las filas. A veces, la selección de las filas debe efectuarse en base a datos contenidos en otras tablas. Por ejemplo, supongamos que se desea suprimir todos los pedidos aceptados por 'Soledad Martinez'. Sin saber su número de empleado no se pueden determinar los pedidos a eliminar consultando únicamente la tabla PEDIDOS. Podría pensar en hallar todos los pedidos utilizando una consulta de dos tablas como la siguiente:

```
SELECT num_pedido, importe FROM pedidos, repventas WHERE rep=num_empl AND
nombre='Soledad Martinez'
```

NUM_PEDIDO	IMPORTE
112979	15.000
113048	3.750
112993	1.896
113065	2.130

Tabla 4.3.3.1: Pedidos aceptados por Soledad Martinez.

Pero no es posible utilizar una composición en una sentencia DELETE. El modo de manejar la petición es con una de las condiciones de búsqueda subconsulta. Por ejemplo:

```
DELETE FROM pedidos WHERE rep=(SELECT num_empl FROM repventas WHERE
nombre='Soledad Martinez')
```

NUM_PEDIDO	FECHA_PEDIDO	CLIE	REP	FAB	PRODUCTO	CANT	IMPORTE
112961	17/12/1999	2117	106	REI	2A44L	7	31.500
113012	11/01/2000	2111	105	ACI	41003	35	3.745
112989	03/01/2000	2101	106	FEA	114	6	1.458
113051	10/02/2000	2118	108	QSA	XK47	4	1.420
112968	12/10/1999	2102	101	ACI	41004	34	3.978
110036	30/01/2000	2107	110	ACI	4100Z	9	22.500
113045	02/02/2000	2112	108	REI	2A44R	10	45.000
112963	17/12/1999	2103	105	ACI	41004	28	3.276
113013	14/01/2000	2118	108	BIC	41003	1	652
113058	23/02/2000	2108	109	FEA	112	10	1.480
112997	08/01/2000	2124	107	BIC	41003	1	652
112983	27/12/1999	2103	105	ACI	41004	6	702
113024	20/01/2000	2114	108	QSA	XK47	20	7.100
113062	24/02/2000	2124	107	FEA	114	10	2.430
113027	22/02/2000	2103	105	ACI	41002	54	4.104
113007	08/01/2000	2112	108	IMM	773C	3	2.825
113069	02/03/2000	2109	107	IMM	775C	22	31.350
113034	29/01/2000	2107	110	REI	2A45C	8	632
112992	04/11/1999	2118	108	ACI	41002	10	760
112975	12/10/1999	2111	103	REI	2A44G	6	2.100

113055	15/02/2000	2108	101	ACI	4100X	6	150
113003	25/01/2000	2108	109	IMM	779C	3	5.625
113049	10/02/2000	2118	108	QSA	XK47	2	776
112987	31/12/1999	2103	105	ACI	4100Y	11	27.500
113057	18/02/2000	2111	103	ACI	4100X	24	600
113042	02/02/2000	2113	101	REI	2A44R	5	22.500
113070	22/10/1999	2126	111	ACI	41004	20	2.340

Tabla 4.3.3.2: La tabla PEDIDOS después de eliminar los pedidos aceptados por 'Soledad Martinez'.

Permitiría borrar las cuatro filas de pedidos aceptados por 'Soledad Martinez'. Otros ejemplos de sentencias DELETE que utilizan condiciones de búsqueda subconsulta son:

```
DELETE FROM clientes WHERE rep_clie IN (SELECT num_empl FROM repventas
WHERE ventas<(0.8*cuota))
```

La cual suprime los clientes atendidos por vendedores cuyas ventas son inferiores al 80% de su cuota.

```
DELETE FROM repventas WHERE (0.02*cuota) >(SELECT SUM(importe) FROM
pedidos WHERE rep=num_empl)
```

Que suprime los vendedores cuyo total de pedidos actual es menor que el 2% de sus cuotas.

```
DELETE FROM clientes WHERE NOT EXISTS (SELECT * FROM pedidos WHERE
clie=num_clie AND fecha_pedido>='10/11/1999')
```

Suprime los clientes que no han realizado pedidos desde el 10 de Noviembre de 1999.

4.4 Modificación de datos en la base de datos.

Generalmente, los valores de los datos almacenados en una base de datos se modifican cuando se producen cambios correspondientes en el mundo exterior. Por ejemplo, cuando un cliente modifica un pedido, un director se traslada de una oficina a otra, etc. En cada caso, los valores de los datos en la base de datos se actualizan para mantener la base de datos como un modelo preciso del mundo real. La unidad más pequeña de datos que puede modificarse en una base de datos es una única columna de una única fila.

4.4.1 La sentencia UPDATE.

La sentencia UPDATE, cuyo diagrama sintáctico se muestra en la figura 4.4.1.1, modifica los valores de una o más columnas en las filas seleccionadas de una única tabla. La tabla destino a actualizar se indica en la sentencia, y es necesario disponer de permiso para actualizar la tabla además de cada una de las columnas individuales que serán modificadas. La cláusula WHERE selecciona las filas de la tabla a modificar. La cláusula SET especifica que columnas se van a actualizar y calcula los nuevos valores.

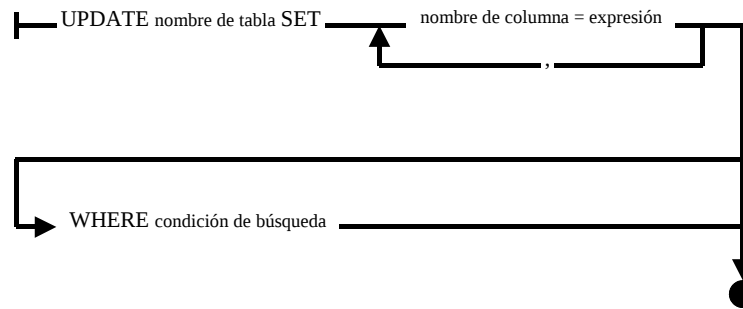


Figura 4.4.1.1: Diagrama sintáctico de la sentencia UPDATE.

Por ejemplo, la siguiente sentencia UPDATE modifica el límite de crédito y el vendedor asignado a un cliente:

```
UPDATE clientes SET limite_credito=60000,rep_clie=109 WHERE empresa='Pino S.L.'
```

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	109	60.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	106	35.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	25.000
2105	MALB S.A.	101	45.000

Tabla 4.4.1.1: La tabla CLIENTES después de modificar los datos de la empresa 'Pino S.L.'.

En el ejemplo anterior, la cláusula WHERE identifica una sola fila de la tabla CLIENTES y la cláusula SET asigna nuevos valores a dos de las columnas de esta fila. Sin embargo, al igual que la sentencia DELETE, la sentencia UPDATE puede actualizar varias filas de una vez, tal y como sucede en el ejemplo siguiente:

```
UPDATE repventas SET oficina_rep=11,cuota=0.9*cuota WHERE oficina_rep=12
```

NUM_EMPL	NOMBRE	EDAD	OFICINA_REP	TITULO	CONTRATO
106	Jose Maldonado	52	11	VP Ventas	14/06/1998
104	Carlos Martinez	33	11	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999

108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	11	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	11	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
DIRECTOR	CUOTA	VENTAS			
NULL	25.000	32.958			
106	15.750	0			
104	30.000	39.327			
106	27.500	7.105			
106	30.000	58.533			
108	30.000	22.776			
104	24.750	26.628			
101	NULL	23.123			
104	22.500	2.700			
108	27.500	34.432			

Tabla 4.4.1.2: La tabla REPVENTAS después de la actualización.

En este caso, la cláusula WHERE selecciona varias filas de la tabla REPVENTAS, y el valor de las columnas OFICINA_REP y CUOTA se modifica en todas ellas. Otros ejemplos de sentencias UPDATE son:

UPDATE clientes SET rep_clie=102 WHERE rep_clie IN (105, 106, 107)

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	102	50.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	102	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	102	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	102	35.000
2122	JPF S.L.	102	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	25.000
2105	MALB S.A.	101	45.000

Tabla 4.4.1.3: La tabla CLIENTES después de asignar los clientes atendidos por los empleados 105, 106 ó 107 al empleado número 102.

UPDATE repventas SET cuota=30000 WHERE cuota IS NULL

NUM_EMPL	NOMBRE	EDAD	OFICINA_REP	TITULO	CONTRATO
106	Jose Maldonado	52	11	VP Ventas	14/06/1998
104	Carlos Martinez	33	12	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999

108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	12	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	12	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
DIRECTOR	CUOTA	VENTAS			
NULL	25.000	32.958			
106	17.500	0			
104	30.000	39.327			
106	27.500	7.105			
106	30.000	58.533			
108	30.000	22.776			
104	27.500	26.628			
101	30.000	23.123			
104	25.000	2.700			
108	27.500	34.432			

Tabla 4.4.1.4: La tabla REPVENTAS después de asignar una cuota de 30.000 a los vendedores que no tenían asignada una cuota.

La expresión en cada asignación puede ser cualquier expresión SQL que genere un valor de tipo de dato apropiado para la columna destino. La expresión debe ser calculable basada en los valores de la fila actualmente en actualización en la tabla destino. No pueden incluirse funciones de columna ni subconsultas.

Si una expresión en la lista de asignación referencia a una de las consultas de la tabla destino, el valor utilizado para calcular la expresión es el valor de esa columna en la fila actual antes de que se aplique ninguna actualización. Lo mismo es aplicable a las referencias de columna que se producen en la cláusula WHERE. Por ejemplo:

UPDATE oficinas SET objetivo=40000,ventas=objetivo WHERE objetivo<40000

OFICINA	CIUDAD	REGION	DIR	OBJETIVO	VENTAS
22	Toledo	Centro	108	40.000	27.500
11	Valencia	Este	106	52.500	40.063
12	Barcelona	Este	104	70.000	29.328
13	Alicante	Este	105	40.000	30.000
21	Madrid	Centro	108	60.000	81.309

Tabla 4.4.1.5: La tabla OFICINAS después de la actualización.

Antes de la actualización, las oficinas de Toledo y Alicante tenían unos objetivos de 27.500 y 30.000 respectivamente, después de la actualización sus ventas son 27.500 y 30.000 respectivamente y no 40.000 como se esperaría en caso de suponer que la actualización se realiza con el valor del nuevo objetivo.

4.4.2 Actualización de todas las filas.

La cláusula WHERE en la sentencia UPDATE es opcional. Si se omite la cláusula WHERE entonces se actualizan todas las filas de la tabla destino:

UPDATE repventas SET cuota=1.05*cuota

NUM_EMPL	NOMBRE	EDAD	OFICINA_REP	TITULO	CONTRATO
106	Jose Maldonado	52	11	VP Ventas	14/06/1998

104	Carlos Martinez	33	12	Dir. Ventas	19/05/1997
105	Belen Aguirre	37	13	Dir. Ventas	12/02/1998
109	Maria Garcia	31	11	Rep. Ventas	12/10/1999
108	Lorenzo Fernandez	62	21	Dir. Ventas	12/10/1999
102	Soledad Martinez	48	21	Rep. Ventas	10/12/1996
101	Daniel Gutierrez	45	12	Rep. Ventas	20/10/1996
110	Antonio Valle	41	NULL	Rep. Ventas	13/01/2000
103	Pedro Cruz	29	12	Rep. Ventas	01/03/1997
107	Natalia Martin	49	22	Rep. Ventas	14/11/1998
DIRECTOR	CUOTA	VENTAS			
NULL	26.250	32.958			
106	18.375	0			
104	31.500	39.327			
106	28.875	7.105			
106	31.500	58.533			
108	31.500	22.776			
104	28.875	26.628			
101	NULL	23.123			
104	26.250	2.700			
108	28.875	34.432			

Tabla 4.4.2.1: La tabla REPVENTAS después de actualizar la cuota.

A diferencia de la sentencia DELETE, en donde la cláusula WHERE casi nunca se omite, la sentencia UPDATE sin una cláusula WHERE realiza una función útil, actualizar de forma masiva toda una tabla, como demuestra el ejemplo anterior.

4.4.3 UPDATE con subconsultas.

Al igual que la sentencia DELETE, las subconsultas pueden jugar un papel importante en la sentencia UPDATE ya que permiten seleccionar filas a actualizar en base a información contenida en otras tablas. Por ejemplo:

```
UPDATE clientes SET limite_credito=limite_credito+5000 WHERE num_clie IN
(SELECT DISTINCT clie FROM pedidos WHERE importe>25000)
```

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	103	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	105	55.000
2123	Hnos. Martinez S.A.	102	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	55.000
2121	Hernandez & hijos S.L.	103	45.000
2114	Componentes Fernandez S.A.	102	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	109	55.000
2117	Hnos. Ramon S.L.	106	40.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	102	50.000
2106	Construcciones Leon S.A.	102	65.000
2119	Martinez & Garcia S.L.	109	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	104	20.000
2109	Roda & Castedo S.L.	103	30.000
2105	MALB S.A.	101	45.000

Tabla 4.4.3.1: La tabla CLIENTES después de elevar en 5000 el limite de crédito de cualquier cliente que haya remitido un pedido de más de 25000.

```
UPDATE clientes SET rep_clie=105 WHERE rep_clie IN (SELECT num_empl FROM repventas WHERE ventas<(0.8*cuota))
```

NUM_CLIE	EMPRESA	REP_CLIE	LIMITE_CREDITO
2111	EVBE S.A.	105	50.000
2102	Exclusivas del Este S.L.	101	65.000
2103	Pino S.L.	105	50.000
2123	Hnos. Martinez S.A.	105	40.000
2107	Distribuciones Sur S.A.	110	35.000
2115	AFS S.A.	101	20.000
2101	Exclusivas Soriano S.A.	106	65.000
2112	Lopez Asociados S.L.	108	50.000
2121	Hernandez & hijos S.L.	105	45.000
2114	Componentes Fernandez S.A.	105	20.000
2124	Domingo S.L.	107	40.000
2108	Zapater Importaciones S.A.	105	55.000
2117	Hnos. Ramon S.L.	106	35.000
2122	JPF S.L.	105	30.000
2120	Distribuciones Montiel S.L.	105	50.000
2106	Construcciones Leon S.A.	105	65.000
2119	Martinez & Garcia S.L.	105	25.000
2118	Exclusivas Norte S.A.	108	60.000
2113	Importaciones Martin S.L.	105	20.000
2109	Roda & Castedo S.L.	105	25.000
2105	MALB S.A.	101	45.000

Tabla 4.4.3.2: La tabla CLIENTES después de asignar todos los clientes atendidos por vendedores cuyas ventas son menores al 80% de sus cuotas al vendedor número 105.

4.5 Control de las transacciones.

En SQL se entiende como transacción a una secuencia de una o más sentencias SQL que juntas forman una unidad de trabajo. Las sentencias SQL que forman la transacción suelen estar estrechamente relacionadas y efectuar acciones interdependientes. Cada sentencia de una transacción efectúa una parte de una tarea, pero todas ellas son necesarias para completar la tarea. La agrupación de las sentencias en una sola transacción indica al DBMS que la secuencia de sentencias entera debe ser ejecutada atómicamente; todas las sentencias deben completarse para que la base de datos esté en un estado consistente. Algunos ejemplos de transacciones son:

- Añadir un pedido. Para aceptar el pedido de un cliente, el programa de entrada de pedido debería:
 1. Consultar la tabla PRODUCTOS para asegurar que el producto está en existencias.
 2. Insertar el pedido en la tabla PEDIDOS.
 3. Actualizar la tabla PRODUCTOS, restando la cantidad solicitada de las existencias del producto.

4. Actualizar la tabla REPVENTAS sumando el importe del pedido a las ventas totales del vendedor que lo aceptó.
 5. Actualizar la tabla OFICINAS, añadiendo el importe del pedido a las ventas totales de la oficina en donde el vendedor trabaja.
- Cancelar un pedido. Para cancelar el pedido de un cliente, el programa debería:
 1. Suprimir el pedido de la tabla PEDIDOS.
 2. Actualizar la tabla PRODUCTOS, ajustando el total de existencias del producto.
 3. Actualizar la tabla REPVENTAS, restando el importe del pedido del total de ventas del vendedor.
 4. Actualizar la tabla OFICINAS, restando el importe del pedido de las ventas totales de la oficina.

En cada uno de estos casos se requiere una sentencia de cuatro o cinco acciones, cada una formada por una sola sentencia SQL, para manejar la única transacción “lógica”. En un DBMS basado en SQL las sentencias de una transacción se ejecutan como una unidad atómica de trabajo en la base de datos. O todas las sentencias son ejecutadas con éxito, o ninguna de las sentencias es ejecutada. El DBMS es el responsable de mantener este compromiso incluso si el programa de aplicación aborta o se produce un fallo a mitad de la transacción.

SQL soporta las transacciones de base de datos mediante dos sentencias de procesamiento de transacciones SQL, cuyo diagrama sintáctico se muestra en la figura 4.5.1.

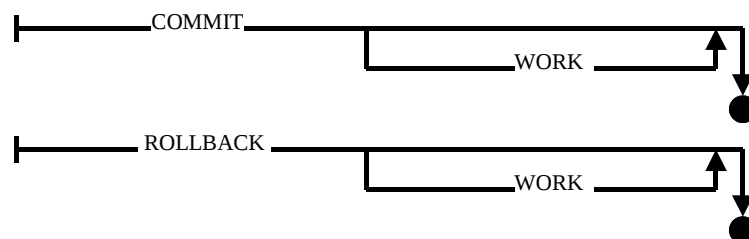


Figura 4.5.1: Diagramas sintácticos de las sentencias COMMIT y ROLLBACK.

La sentencia COMMIT señala el final correcto de una transacción. Informa al DBMS que la transacción está ahora completa; todas las sentencias que forman la transacción han sido ejecutadas, y la base de datos se encuentra en un estado consistente.

La sentencia ROLLBACK señala el final sin éxito de una transacción. Informa al DBMS que el usuario no desea completar la transacción; en vez de ello, el DBMS debe deshacer los cambios efectuados a la base de datos durante la transacción.