

Combinación de varias tablas

Contenido

Introducción	1
Uso de alias en los nombres de tablas	2
Combinación de datos de varias tablas	4
Combinación de varios conjuntos de resultados	20

Notas para el instructor

Este módulo proporciona a los alumnos información general acerca de cómo consultar varias tablas mediante la utilización de distintos tipos de combinaciones, combinar los conjuntos de resultados con el operador UNION y crear tablas mediante la instrucción SELECT INTO.

Al final de este módulo, los alumnos serán capaces de:

Utilizar alias en los nombres de tablas.

Combinar datos de dos o más tablas mediante combinaciones.

Combinar varios conjuntos de resultados en un único conjunto de resultados mediante el operador UNION.

Introducción

Objetivo del tema

Presentar los temas que abarca este módulo.

Explicación previa

En este módulo aprenderá a combinar varias tablas.

- Uso de alias en los nombres de tablas
- Combinación de datos de varias tablas
- Combinación de varios conjuntos de resultados

Este módulo proporciona a los alumnos un resumen de cómo pueden consultar varias tablas mediante la utilización de distintos tipos de combinaciones, combinar los conjuntos de resultados con el operador UNION y crear tablas mediante la instrucción SELECT INTO.

Después de finalizar este módulo, el alumno será capaz de:

Uso de alias en los nombres de tablas.

Combinación de datos de varias tablas mediante combinaciones.

Combinación de varios conjuntos de resultados en un único conjunto de resultados mediante el operador UNION.

Uso de alias en los nombres de tablas

Objetivo del tema

Describir cómo usar alias en los nombres de tablas.

Explicación previa

El uso de alias en los nombres de tablas mejora la legibilidad de las secuencias de comandos, facilita la escritura de combinaciones complejas y simplifica el mantenimiento de Transact-SQL.

■ Ejemplo 1 (sin un nombre de alias)

```
USE joindb
SELECT buyer_name, sales.buyer_id, qty
FROM buyers INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id
GO
```

■ Ejemplo 2 (con un nombre de alias)

```
USE joindb
SELECT buyer_name, s.buyer_id, qty
FROM buyers AS b INNER JOIN sales AS s
ON b.buyer_id = s.buyer_id
GO
```

El uso de alias en los nombres de tablas mejora la legibilidad de las secuencias de comandos, facilita la escritura de combinaciones complejas y simplifica el mantenimiento de Transact-SQL.

Al escribir secuencias de comandos, puede sustituir un nombre de tabla descriptivo largo y complejo por un alias sencillo y abreviado. El alias se utiliza en lugar del nombre completo de la tabla.

Sintaxis parcial

```
SELECT * FROM servidor.baseDeDatos.esquema.tabla AS aliasTabla
```

Ejemplo 1

En este ejemplo se muestran los nombres de los clientes, el identificador del cliente y la cantidad vendida de las tablas **buyers** y **sales**. Esta consulta no utiliza alias en las tablas de la sintaxis de JOIN.

```
USE joindb
SELECT buyer_name, sales.buyer_id, qty
FROM buyers
INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id
GO
```

Ejemplo 2

En este ejemplo se muestran los nombres de los clientes, el identificador del cliente y la cantidad vendida de las tablas **buyers** y **sales**. Esta consulta utiliza alias en las tablas de la sintaxis de JOIN.

```
USE joindb
SELECT buyer_name, s.buyer_id, qty
  FROM buyers AS b
  INNER JOIN sales AS s
    ON b.buyer_id = s.buyer_id
GO
```

Nota Algunas veces, la compleja sintaxis de JOIN y las subconsultas deben usar alias en los nombres de tablas. Por ejemplo, al combinar una tabla consigo misma deben utilizarse alias.

◆ Combinación de datos de varias tablas

Objetivo del tema

Explicar las diferentes maneras de combinar datos de dos o más tablas o conjuntos de resultados.

Explicación previa

Se pueden combinar datos de dos o más tablas, incluso si las tablas residen en bases de datos diferentes.

- Introducción a las combinaciones
- Uso de combinaciones internas
- Uso de las combinaciones externas
- Uso de las combinaciones cruzadas
- Combinación de más de dos tablas
- Combinación de una tabla consigo misma

Una combinación es una operación que permite consultar dos o más tablas para producir un conjunto de resultados que incorpore filas y columnas de cada una de las tablas. Las tablas se combinan en función de las columnas que son comunes a ambas tablas.

Cuando se combinan tablas, Microsoft® SQL Server™ 2000 compara los valores de las columnas especificadas fila por fila y, después, utiliza los resultados de la comparación para combinar los valores que cumplan los criterios especificados en nuevas filas.

Hay tres tipos de combinaciones: combinaciones internas, combinaciones externas y combinaciones cruzadas. Adicionalmente, en una instrucción SELECT se pueden combinar más de dos tablas mediante un conjunto de combinaciones o se puede combinar una tabla consigo misma mediante una autocombinación.

Introducción a las combinaciones

Objetivo del tema

Explicar cómo se implementan las combinaciones.

Explicación previa

Las tablas se combinan para producir un único conjunto de resultados que incorpore elementos de dos o más tablas.

■ Selección de columnas específicas de varias tablas

- La palabra clave JOIN especifica qué tablas se van a combinar y cómo
- La palabra clave ON especifica la condición de combinación

■ Consultas de dos o más tablas para producir un conjunto de resultados

- Use claves principales y externas como condiciones de combinación
- Para combinar tablas, utilice columnas comunes a las tablas especificadas

Las tablas se combinan para producir un único conjunto de resultados que incorpore filas y columnas de dos o más tablas.

Sintaxis parcial

```
SELECT columna [, columna ...]
FROM {<tablaOrigen >} [, ...n]
    <tipoCombinación > ::=
    [ INNER | { { LEFT | RIGHT | FULL } [OUTER] } ]
    [ <sugerenciaCombinación> ]
JOIN
    <tablaCombinada> ::=
    <tablaOrigen > <tipoCombinación > <tablaOrigen > ON
    <condiciónBúsqueda>
    | <tablaOrigen > CROSS JOIN <tablaOrigen >
    | <tablaCombinada>
```

Sugerencia

Haga referencia a los Libros en pantalla de SQL Server para mostrar la instrucción SELECT completa y para hacer hincapié en las combinaciones.

Selección de columnas específicas de varias tablas

Una combinación permite seleccionar columnas de varias tablas al expandir la cláusula FROM de la instrucción SELECT. En la cláusula FROM se incluyen dos palabras clave adicionales: JOIN y ON.

La palabra clave JOIN especifica qué tablas se van a combinar y cómo.

La palabra clave ON especifica las columnas que las tablas tienen en común.

Consultas de dos o más tablas para producir un conjunto de resultados

Una combinación permite consultar dos o más tablas para producir un único conjunto de resultados. Al implementar combinaciones, tenga en cuenta los siguientes hechos e instrucciones:

Especifique la condición de combinación en función de claves principales y externas.

Si una tabla tiene una clave principal compuesta, cuando combine tablas debe hacer referencia a toda la clave en la cláusula ON.

Para combinar tablas, utilice columnas comunes a las tablas especificadas. Dichas columnas deben tener tipos de datos iguales o similares.

Haga referencia al nombre de la tabla si las columnas de las tablas que va a combinar tienen el mismo nombre. Califique los nombres de las columnas con el formato *tabla.columna*.

Limite el número de tablas de las combinaciones porque cuantas más tablas combine, mayor será la duración del proceso de la consulta.

Puede incluir varias combinaciones en una instrucción SELECT.

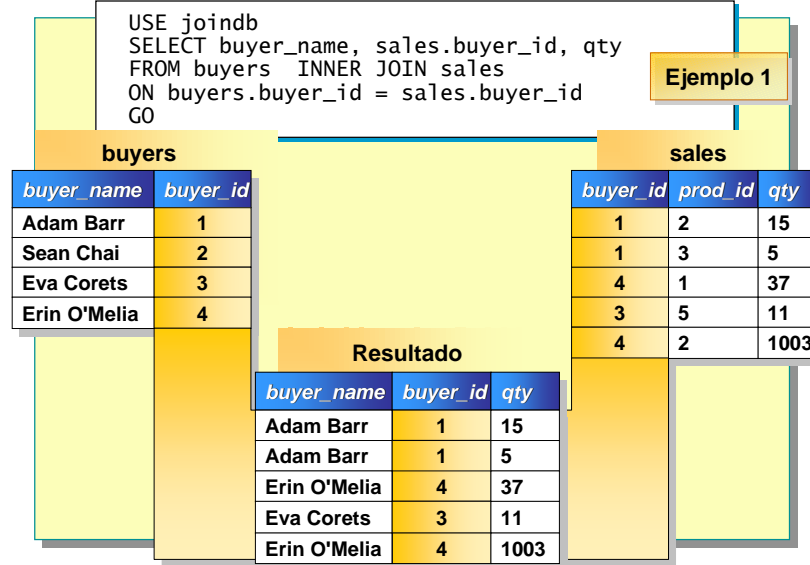
Uso de combinaciones internas

Objetivo del tema

Definir y demostrar las combinaciones internas.

Explicación previa

Utilice las combinaciones internas para combinar tablas en las que los valores de las columnas comparadas sean iguales.



Las combinaciones internas combinan tablas mediante la comparación de los valores de las columnas que son comunes a ambas tablas. SQL Server sólo devuelve las filas que cumplen las condiciones de la combinación.

Nota Los ejemplos de este módulo provienen de la base de datos **joindb** (una base de datos creada específicamente para explicar los diferentes tipos de combinaciones). La base de datos **joindb** está incluida en el disco compacto Material del alumno.

Sugerencia

Los ejemplos de las diapositivas de este módulo provienen de la base de datos **joindb** (una base de datos creada específicamente para explicar los diferentes tipos de combinaciones). La base de datos **joindb** está incluida en el disco compacto Material del alumno.

Explique que SQL Server no garantiza el orden del conjunto de resultados a menos que se especifique con una cláusula **ORDER BY**.

Por qué se utilizan combinaciones internas

Utilice combinaciones internas para obtener información de dos tablas independientes y combinar dicha información en un conjunto de resultados. Al utilizar combinaciones internas, tenga en cuenta los siguientes hechos e instrucciones:

Las combinaciones internas son el tipo predeterminado de SQL Server. Puede abreviar la cláusula **INNER JOIN** como **JOIN**.

Para especificar las columnas que desea presentar en el conjunto de resultados, incluya los nombres calificados de las columnas en la lista de selección.

Incluya una cláusula **WHERE** para restringir las filas que se devuelven en el conjunto de resultados.

No utilice valores **NULL** como condición de combinación, ya que no se evalúan como iguales entre sí.

Nota SQL Server no garantiza el orden del conjunto de resultados a menos que se especifique con una cláusula **ORDER BY**.

Ejemplo 1

Este ejemplo devuelve los valores **buyer_name**, **buyer_id** y **qty** de los clientes que han adquirido algún producto. Los clientes que no hayan adquirido nada no se incluyen en el conjunto de resultados. Los clientes que han adquirido más de un producto aparecen una vez por cada compra realizada.

Sugerencia

Señale que se puede hacer referencia a la columna **buyer_id** de ambas tablas en la lista de selección.

Las columnas **buyer_id** de las dos tablas pueden especificarse en la lista de selección.

```
USE joindb
SELECT buyer_name, sales.buyer_id, qty
FROM buyers
INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id
GO
```

Resultado

buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Erin O'Melia	4	37
Eva Corets	3	11
Erin O'Melia	4	1003
(5 filas afectadas)		

Ejemplo 2

En este ejemplo se devuelven los nombres de los productos y de las compañías que los suministran. Los productos sin suministradores asignados y los suministradores sin productos asignados no se incluyen en el conjunto de resultados.

```
USE northwind
SELECT productname, companyname
FROM products
INNER JOIN suppliers
ON products.supplierid = suppliers.supplierid
GO
```

Resultado

productname	companyname
Chai	Exotic Liquids
Chang	Exotic Liquids
Aniseed Syrup	Exotic Liquids
Chef Anton's Cajun Seasoning	New Orleans Cajun Delights
.	
.	
.	
(77 filas afectadas)	

Ejemplo 3

En este ejemplo se devuelven los nombres de los clientes que han hecho pedidos después del 1/1/98. Observe que se utiliza una cláusula **WHERE** para restringir las filas devueltas en el conjunto de resultados.

```
USE northwind
SELECT DISTINCT companyname, orderdate
FROM orders INNER JOIN customers
ON orders.customerid = customers.customerid
WHERE orderdate > '1/1/98'
GO
```

Resultado

companyname	orderdate
Alfreds Futterkiste	1998-01-15 00:00:00.000
Alfreds Futterkiste	1998-03-16 00:00:00.000
Alfreds Futterkiste	1998-04-09 00:00:00.000
Ana Trujillo Emparedados y helados	1998-03-04 00:00:00.000
.	
.	
.	
(264 filas afectadas)	

Ejemplo 4

En este ejemplo se devuelve el número de título y el número de miembro del lector para todos los libros prestados, desde las tablas **copy** y **loan** de la base de datos **library**. Las dos tablas, **copy** y **loan**, tienen una clave principal compuesta con las columnas **isbn** y **copy_no**. Cuando se combinan las dos tablas, hay que especificar las dos columnas como condiciones de combinación, porque identifican de forma única cada copia concreta de un libro específico.

Sugerencia

En este ejemplo se utiliza la base de datos **library**, ya que la base de datos **northwind** no tiene dos tablas con claves principales compuestas relacionadas entre sí.

```
USE library
SELECT copy.title_no, loan.member_no
FROM copy
INNER JOIN loan
ON copy.isbn = loan.isbn
AND copy.copy_no = loan.copy_no
WHERE copy.on_loan = 'Y'
GO
```

Resultado

title_no	member_no
1	611
1	2145
1	312
1	1846
.	
.	
.	
(2000 filas afectadas)	

Uso de las combinaciones externas

Objetivo del tema

Definir las combinaciones externas y describir los tres tipos.

Explicación previa

Mediante las combinaciones externas izquierda, derecha y completa se pueden incluir en el conjunto de resultados filas que no cumplan la condición de combinación.

```
USE joindb
SELECT buyer_name, sales.buyer_id, qty
FROM buyers LEFT OUTER JOIN sales
ON buyers.buyer_id = sales.buyer_id
GO
```

Ejemplo 1

buyers		sales		
buyer_name	buyer_id	buyer_id	prod_id	qty
Adam Barr	1	1	2	15
Sean Chai	2	1	3	5
Eva Corets	3	4	1	37
Erin O'Melia	4	3	5	11
		4	2	1003

Resultado		
buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Erin O'Melia	4	37
Eva Corets	3	11
Erin O'Melia	4	1003
Sean Chai	NULL	NULL

Sugerencia

Señale los valores NULL de la diapositiva para Sean Chai. Las filas que no cumplen la condición de combinación presentan NULL en el conjunto de resultados.

Las combinaciones externas izquierda y derecha combinan filas de dos tablas que cumplen una condición de combinación, más las filas de la tabla izquierda o derecha que no la cumplen, tal como se especifica en la cláusula JOIN. Las filas que no cumplen la condición de combinación presentan NULL en el conjunto de resultados. También se pueden utilizar combinaciones externas completas para presentar todas las filas de las tablas combinadas, independientemente de si hay valores que coincidan en las tablas.

Por qué se utilizan las combinaciones externas izquierda o derecha

Utilice las combinaciones externas izquierda o derecha cuando necesite una lista completa de los datos de una de las tablas combinadas más la información que cumpla la condición de combinación. Al utilizar las combinaciones externas izquierda o derecha, tenga en cuenta los siguientes hechos e instrucciones:

SQL Server sólo devuelve filas únicas cuando se utilizan las combinaciones externas izquierda o derecha.

Utilice una combinación externa izquierda para presentar todas las filas de la primera tabla especificada (la tabla de la parte izquierda de la expresión). Si invierte el orden en el que aparecen las tablas en la cláusula FROM, la instrucción devuelve el mismo resultado que una combinación externa derecha.

Sugerencia

Pregunta: ¿qué cambiaría en la consulta de ejemplo de la diapositiva para obtener el mismo resultado con una cláusula RIGHT OUTER JOIN?

Respuesta: invierta el orden de las tablas en la cláusula FROM y utilice la cláusula RIGHT OUTER JOIN.

Utilice una combinación externa derecha para presentar todas las filas de la segunda tabla especificada (la tabla de la parte derecha de la expresión). Si invierte el orden en el que aparecen las tablas en la cláusula FROM, la instrucción devuelve el mismo resultado que una combinación externa izquierda.

Sugerencia

Utilice siempre la sintaxis de combinación de ANSI SQL-92 con ANSI_NULLS establecido en ON.

Puede abreviar las cláusulas LEFT OUTER JOIN o RIGHT OUTER JOIN como LEFT JOIN o RIGHT JOIN.

Las combinaciones externas sólo se pueden utilizar entre dos tablas.

Ejemplo 1

En este ejemplo se devuelven los valores **buyer_name**, **buyer_id** y **qty** de todos los clientes y sus compras. Observe que los clientes que no han adquirido ningún producto se enumeran en el conjunto de resultados, pero aparecen valores NULL en las columnas **buyer_id** y **qty**.

```
USE joindb
SELECT buyer_name, sales.buyer_id, qty
FROM buyers
LEFT OUTER JOIN sales
ON buyers.buyer_id = sales.buyer_id
GO
```

Resultado

buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Sean Chai	NULL	NULL
Eva Corets	3	11
Erin O'Melia	4	37
Erin O'Melia	4	1003

(6 filas afectadas)

Nota El orden del conjunto de resultados puede ser diferente porque en el ejemplo no se utiliza la cláusula ORDER BY.

Ejemplo 2

En este ejemplo se presentan todos los clientes y las fechas de sus pedidos. Con una combinación externa izquierda se obtiene una fila por cada cliente y filas adicionales si el cliente ha efectuado varios pedidos. Se devuelve NULL en la columna **orderdate** del conjunto de resultados para los clientes que no han hecho ningún pedido. Observe las entradas NULL para los clientes FISSA y Paris Spécialités.

```
USE northwind
SELECT companyname, customers.customerid, orderdate
FROM customers
LEFT OUTER JOIN orders
ON customers.customerid = orders.customerid
GO
```

Resultado	companyname	customerid	orderdate
	Vins et alcools Chevalier	VINIT	1996-07-04 00:00.0
	Toms Spezialitäten	TOMSP	1996-07-05 00:00.0
	Hanari Carnes	HANAR	1996-07-08 00:00.0
	Victuailles en stock	VICTE	1996-07-08 00:00.0
	.		
	.		
	.		
	Paris specialities	PARIS	NULL
	FISSA Fabrica Inter. Salchichas S.A.	FISSA	NULL
	(832 filas afectadas)		

Uso de las combinaciones cruzadas

Objetivo del tema

Mostrar cómo funcionan las combinaciones cruzadas y describir el conjunto de resultados.

Explicación previa

Utilice combinaciones cruzadas para presentar todas las combinaciones de filas posibles de las columnas seleccionadas en las tablas combinadas.

```
USE joindb
SELECT buyer_name, qty
FROM buyers
CROSS JOIN sales
GO
```

Ejemplo 1

buyers		sales			Resultado	
buyer_id	buyer_name	buyer_id	prod_id	qty	buyer_name	qty
1	Adam Barr	1	2	15	Adam Barr	15
2	Sean Chai	1	3	5	Adam Barr	5
3	Eva Corets	4	1	37	Adam Barr	37
4	Erin O'Melia	3	5	11	Adam Barr	11
		4	2	1003	Adam Barr	1003
					Sean Chai	15
					Sean Chai	5
					Sean Chai	37
					Sean Chai	11
					Sean Chai	1003
					Eva Corets	15
				

Sugerencia

Señale que la palabra clave ON y la lista de columnas asociada no se utilizan en la instrucción SELECT porque las combinaciones cruzadas devuelven todas las combinaciones posibles de las filas de cada tabla especificada.

En las combinaciones cruzadas no se requiere una columna común.

Las *combinaciones cruzadas* presentan todas las combinaciones de todas las filas de las tablas combinadas. En este tipo de combinaciones no se requiere una columna común.

Por qué se utilizan las combinaciones cruzadas

Aunque las combinaciones cruzadas no se suelen utilizar en una base de datos normalizada, se pueden utilizar para generar datos de prueba para una base de datos o listas de todas las combinaciones posibles para elaborar listas de comprobación o patrones comerciales.

Cuando se utilizan combinaciones cruzadas, SQL Server genera un producto cartesiano en el que el número de filas del conjunto de resultados es igual al número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla. Por ejemplo, si hay 8 filas en una tabla y 9 filas en la otra, SQL Server devuelve un total de 72 filas.

Ejemplo 1

En este ejemplo se presentan todas las combinaciones posibles de los valores de las columnas **buyers.buyer_name** y **sales.qty**.

```
USE joindb
SELECT buyer_name, qty
FROM buyers
CROSS JOIN sales
GO
```

Resultado	buyer_name	qty
	Adam Barr	15
	Adam Barr	5
	Adam Barr	37
	Adam Barr	11
	Adam Barr	1003
	Sean Chai	15
	Sean Chai	5
	.	
	.	
	.	
	(20 filas afectadas)	

Ejemplo 2

En este ejemplo se presenta una combinación cruzada entre las tablas **shippers** y **suppliers** que puede ser útil para enumerar todas las formas de envío de los proveedores.

Sugerencia

Ejecute esta consulta y explique que este ejemplo es útil para enumerar todas las formas de envío de los proveedores.

Al utilizar una combinación cruzada se presentan todas las combinaciones posibles de las filas de las dos tablas. La tabla **shippers** tiene 3 filas y la tabla **suppliers** tiene 29 filas. El conjunto de resultados contiene 87 filas.

```
USE northwind
SELECT suppliers.companyname, shippers.companyname
FROM suppliers
CROSS JOIN shippers
GO
```

Resultado	companyname	companyname
	Aux joyeux ecclésiastiques	Speedy Express
	Bigfoot Breweries	Speedy Express
	Cooperativa de Quesos 'Las Cabras'	Speedy Express
	Escargots Nouveaux	Speedy Express
	.	
	.	
	.	
	Aux joyeux ecclésiastiques	United Package
	Bigfoot Breweries	United Package
	Cooperativa de Quesos 'Las Cabras'	United Package
	Escargots Nouveaux	United Package
	.	
	.	
	.	
	Aux joyeux ecclésiastiques	Federal Shipping
	Bigfoot Breweries	Federal Shipping
	Cooperativa de Quesos 'Las Cabras'	Federal Shipping
	Escargots Nouveaux	Federal Shipping
	.	
	.	
	.	
	(87 filas afectadas)	

Combinación de más de dos tablas

Objetivo del tema

Explicar cómo combinar más de dos tablas.

Explicación previa

Hasta ahora, hemos tratado la combinación de sólo dos tablas. Sin embargo, se pueden combinar más de dos tablas.

```
USE joindb
SELECT buyer_name, prod_name, qty
FROM buyers
INNER JOIN sales ON buyers.buyer_id = sales.buyer_id
INNER JOIN produce ON sales.prod_id = produce.prod_id
GO
```

Ejemplo 1

buyers		sales			produce	
buyer_id	buyer_name	buyer_id	prod_id	qty	prod_id	prod_name
1	Adam Barr	1	2	15	1	Apples
2	Sean Chai	1	3	5	2	Pears
3	Eva Corets	3	1	37	3	Oranges
4	Erin O'Melia	4	5	11	4	Bananas
		2	2	1003	5	Peaches

Resultado		
buyer_name	prod_name	qty
Erin O'Melia	Apples	37
Adam Barr	Pears	15
Erin O'Melia	Pears	1003
Adam Barr	Oranges	5
Eva Corets	Peaches	11

Sugerencia

La primera combinación se efectúa entre las tablas **buyers** y **sales**. La segunda se efectúa entre las tablas **sales** y **produce**.

Haga hincapié en que cualquier tabla a la que haga referencia en una operación de combinación se puede combinar con otra mediante una columna común.

Se puede combinar cualquier número de tablas. Cualquier tabla a la que se haga referencia en una operación de combinación se puede combinar con otra tabla mediante una columna común.

Por qué se combinan más de dos tablas

Puede utilizar combinaciones múltiples para obtener información relacionada de varias tablas. Al combinar más de dos tablas, tenga en cuenta los siguientes hechos e instrucciones:

Debe tener una o varias tablas con claves externas relacionadas con las tablas que vaya a combinar.

Debe definir una cláusula JOIN por cada columna que forme parte de una clave compuesta.

Incluya una cláusula WHERE para limitar el número de filas que se devolverá.

Ejemplo 1

En este ejemplo se devuelven las columnas **buyer_name**, **prod_name** y **qty** de las tablas **buyers**, **sales** y **produce**. La columna **buyer_id** es común a las tablas **buyers** y **sales**, y se utiliza para combinar dichas tablas. La columna **prod_id** es común a las tablas **sales** y **produce**, y se utiliza para combinar la tabla **produce** con el resultado de la combinación de las tablas **buyers** y **sales**.

```
USE joindb
SELECT buyer_name, prod_name, qty
FROM buyers
INNER JOIN sales
  ON buyers.buyer_id = sales.buyer_id
INNER JOIN produce
  ON sales.prod_id = produce.prod_id
GO
```

Resultado	buyer_name	prod_name	qty
	Erin O'Melia	Apples	37
	Adam Barr	Pears	15
	Erin O'Melia	Pears	1003
	Adam Barr	Oranges	5
	Eva Corets	Peaches	11

(5 filas afectadas)

Ejemplo 2

En este ejemplo se presenta información de las tablas **orders** y **products** mediante el uso de la tabla **order details** como vínculo. Por ejemplo, si desea obtener la lista de los productos que se piden cada día, necesitaría información de las tablas **orders** y **products**. Un pedido puede contener muchos productos y un producto puede aparecer en muchos pedidos.

Para obtener la información de las tablas **orders** y **products**, puede utilizar una combinación interna a través de la tabla **order details**. Incluso aunque no se obtenga ninguna columna de la tabla **order details**, debe incluir esa tabla como parte de la combinación interna para relacionar la tabla **orders** con la tabla **products**. En este ejemplo, la columna **orderid** es común a las tablas **orders** y **order details**, y la columna **productid** es común a las tablas **order details** y **products**.

```
USE northwind
SELECT orderdate, productname
FROM orders AS O
INNER JOIN [order details] AS OD
ON O.orderid = OD.orderid
INNER JOIN products AS P
ON OD.productid = P.productid
WHERE orderdate = '7/8/96'
```

Resultado	orderdate	productname
	1996-07-08	Jack's New England Clam Chowder
	1996-07-08	Manjimup Dried Apples
	1996-07-08	Louisiana Fiery Hot Pepper Sauce
	1996-07-08	Gustaf's Knakebrod
	1996-07-08	Ravioli Angelo
	1996-07-08	Louisiana Fiery Hot Pepper Sauce

(6 filas afectadas)

Combinación de una tabla consigo misma

Objetivo del tema

Explicar la autocombinación.

Explicación previa

Aunque las combinaciones se suelen utilizar comúnmente para combinar varias tablas, también puede utilizar una autocombinación para combinar una tabla consigo misma.

```
USE joindb
SELECT a.buyer_id AS buyer1, a.prod_id
      ,b.buyer_id AS buyer2
FROM sales AS a
INNER JOIN sales AS b
  ON a.prod_id = b.prod_id
WHERE a.buyer_id > b.buyer_id
GO
```

Ejemplo 3

sales a

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

sales b

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Resultado

buyer1	prod_id	buyer2
4	2	1

Sugerencia

El ejemplo de la diapositiva muestra el resultado deseado cuando una tabla se combina consigo misma. Utilice los ejemplos del cuaderno de trabajo del alumno para explicar cómo se combina una tabla consigo misma.

Si desea buscar las filas que tienen valores en común con otras filas de la misma tabla, puede utilizar una autocombinación para combinar una tabla con otra copia de sí misma.

Por qué se utilizan las autocombinaciones

Aunque las autocombinaciones no se suelen utilizar en una base de datos normalizada, se pueden utilizar para reducir el número de consultas que se ejecutan cuando se comparan valores de diferentes columnas de la misma tabla. Al utilizar las autocombinaciones, tenga en cuenta las siguientes instrucciones:

Debe especificar alias de tabla para hacer referencia a las dos copias de la tabla.

Recuerde que los alias de tabla son diferentes de los alias de columna. Los alias de tabla se designan como el nombre de tabla seguido del alias.

Cuando cree autocombinaciones, cada fila que coincida consigo misma y las parejas que coincidan entre ellas se repiten, por lo que se devuelven filas duplicadas. Utilice una cláusula WHERE para eliminar dichas filas duplicadas.

Ejemplo 1

En este ejemplo se presenta la lista de todos los clientes que han adquirido los mismos productos. Observe que las filas primera y tercera del conjunto de resultados son filas en las que **buyer1** coincide consigo mismo. Las filas cuarta y séptima son filas en las que **buyer4** coincide consigo mismo. Las filas segunda y sexta son filas en las que una es el reflejo de la otra.

Sugerencia

Señale los duplicados en los que las filas coinciden consigo mismas (filas 1, 3, 4 y 7). Utilice una cláusula WHERE con el operador no igual a (<>) para eliminar este tipo de duplicados.

```
USE joindb
SELECT a.buyer_id AS buyer1, a.prod_id, b.buyer_id AS buyer2
FROM sales AS a
INNER JOIN sales AS b
  ON a.prod_id = b.prod_id
WHERE a.buyer_id <> b.buyer_id
GO
```

Resultado	buyer1	prod_id	buyer2
	1	2	1
	4	2	1
	1	3	1
	4	1	4
	3	5	3
	1	2	4
	4	2	4

(7 filas afectadas)

Ejemplo 2

En este ejemplo se presenta una lista de los clientes que han adquirido los mismos productos, pero se eliminan las filas duplicadas, como aquéllas en las que **buyer1** y **buyer4** coincidían consigo mismos.

Sugerencia

Señale que el ejemplo no eliminará las filas duplicadas en las que una es el reflejo de la otra.

Compare los conjuntos de resultados de los ejemplos 1 y 2. Observe que, al utilizar la cláusula WHERE con el operador no igual a (<>), se eliminan las filas duplicadas. Sin embargo, las filas duplicadas que son el reflejo de otra siguen devolviéndose en el conjunto de resultados.

USE joindb

```
SELECT a.buyer_id AS buyer1, a.prod_id, b.buyer_id AS buyer2
FROM sales AS a
INNER JOIN sales AS b
ON a.prod_id = b.prod_id
WHERE a.buyer_id <> b.buyer_id
GO
```

Resultado	buyer1	prod_id	buyer2
	4	2	1
	1	2	4

(2 filas afectadas)

Ejemplo 3

En este ejemplo se presenta una lista de los clientes que han adquirido los mismos productos.

Sugerencia

Señale que el uso de la cláusula WHERE con el operador mayor que (>) o menor que (<) elimina los duplicados del ejemplo 2.

Observe que, cuando la cláusula WHERE incluye el operador mayor que (>), se eliminan todas las filas duplicadas.

USE joindb

```
SELECT a.buyer_id AS buyer1, a.prod_id, b.buyer_id AS buyer2
FROM sales AS a
INNER JOIN sales AS b
ON a.prod_id = b.prod_id
WHERE a.buyer_id > b.buyer_id
GO
```

Resultado	buyer1	prod_id	buyer2
	4	2	1

(1 fila afectada)

Ejemplo 4

En este ejemplo se presentan parejas de empleados que tienen el mismo cargo. Cuando la cláusula WHERE *incluye* el operador menor que (<), se eliminan las filas que coinciden consigo mismas y las filas duplicadas.

```
USE northwind
SELECT a.employeeid, LEFT(a.lastname,10) AS name
      ,LEFT(a.title,10) AS title
      ,b.employeeid, LEFT(b.lastname,10) AS name
      ,LEFT(b.title,10) AS title
FROM employees AS a
INNER JOIN employees AS b
  ON a.title = b.title
WHERE a.employeeid < b.employeeid
GO
```

Resultado

employeeid	Name	title	employeeid	name	title
1	Davolio	Sales Repr	3	Leverling	Sales Repr
1	Davolio	Sales Repr	4	Peacock	Sales Repr
1	Davolio	Sales Repr	6	Suyama	Sales Repr
1	Davolio	Sales Repr	7	King	Sales Repr
1	Davolio	Sales Repr	9	Dodsworth	Sales Repr
3	Leverling	Sales Repr	4	Peacock	Sales Repr
3	Leverling	Sales Repr	6	Suyama	Sales Repr
3	Leverling	Sales Repr	7	King	Sales Repr
3	Leverling	Sales Repr	9	Dodsworth	Sales Repr
4	Peacock	Sales Repr	6	Suyama	Sales Repr
4	Peacock	Sales Repr	7	King	Sales Repr
4	Peacock	Sales Repr	9	Dodsworth	Sales Repr
6	Suyama	Sales Repr	7	King	Sales Repr
6	Suyama	Sales Repr	9	Dodsworth	Sales Repr
7	King	Sales Repr	9	Dodsworth	Sales Repr

(15 filas afectadas)

Combinación de varios conjuntos de resultados

Objetivo del tema

Explicar el propósito y la función del operador UNION.

Explicación previa

Se pueden combinar los resultados de dos o más instrucciones SELECT en un único conjunto de resultados mediante el operador UNION.

- Use el operador UNION para crear un conjunto de resultados único a partir de varias consultas
- Cada consulta debe tener:
 - tipos de datos similares
 - el mismo número de columnas
 - el mismo orden de columnas en la lista de selección

```
USE northwind
SELECT (firstname + ' ' + lastname) AS name
      ,city, postalcode
FROM employees
UNION
SELECT companyname, city, postalcode
FROM customers
GO
```

El operador UNION combina el resultado de dos o más instrucciones SELECT en un único conjunto de resultados.

Utilice el operador UNION cuando los datos que desee obtener residan en ubicaciones diferentes y no se pueda tener acceso a ellas con una única consulta. Al utilizar el operador UNION, tenga en cuenta los siguientes hechos e instrucciones:

SQL Server requiere que las tablas a las que se hace referencia tengan tipos de datos similares, el mismo número de columnas y el mismo orden de columnas en la lista de selección de cada consulta.

SQL Server quita las filas duplicadas del conjunto de resultados. Sin embargo, si utiliza la opción ALL, se incluyen todas las filas en el conjunto de resultados, incluso las duplicadas.

Punto clave

Cuando utilice el operador UNION, las tablas a las que se hace referencia deben tener tipos de datos similares, el mismo número y el mismo orden de columnas en la lista de selección de cada consulta.

Debe especificar los nombres de las columnas en la primera instrucción SELECT. Por lo tanto, si desea definir nuevos encabezados de columnas en el conjunto de resultados, deberá crear los alias de columnas en la primera instrucción SELECT.

Si desea que todo el conjunto de resultados aparezca con un orden específico, debe especificar la ordenación con una cláusula ORDER BY en la instrucción UNION. De lo contrario, puede que el conjunto de resultados no se devuelva en el orden que desea.

Puede lograr un mayor rendimiento si divide una consulta compleja en varias instrucciones SELECT y, después, utiliza el operador UNION para combinarlas.

Sintaxis

instrucciónSelect UNION [ALL] *instrucciónSelect*

Ejemplo

En este ejemplo se combinan dos conjuntos de resultados. El primero devuelve el nombre, la ciudad y el código postal de todos los clientes desde la tabla **customers**. El segundo devuelve el nombre, la ciudad y el código postal de todos los empleados desde la tabla **employees**. Observe que, cuando se utiliza el operador UNION para combinar ambos conjuntos de resultados, se devuelven los alias de las columnas de la primera lista de selección.

Sugerencia

Demuestre este ejemplo con el Analizador de consultas SQL.

```
USE northwind
SELECT (firstname + ' ' + lastname) AS name, city, postalcode
FROM employees
UNION
SELECT companyname, city, postalcode
FROM customers
GO
```

Resultado

name	city	postalcode
Alfreds Futterkiste	Berlin	12209
Ana Trujillo Emparedados y helados	México D.F.	05021
Antonio Moreno Taquería	México D.F.	05023
Around the Horn	Londres	WA1 1DP
B's Beverages	Londres	EC2 5NT
.		
.		
.		
Andrew Fuller	Tacoma	98401
Robert King	Londres	RG 19SP
Janet Leverling	Kirkland	98033
Anne Dodsworth	Londres	WG2 7LT

(100 filas afectadas)

Sugerencia

Pregunta: en el conjunto de resultados, ¿por qué aparecen antes los clientes que los empleados, al contrario de lo que la sintaxis parece indicar?

Respuesta: SQL Server no garantiza un orden concreto a menos que se especifique con una cláusula ORDER BY.