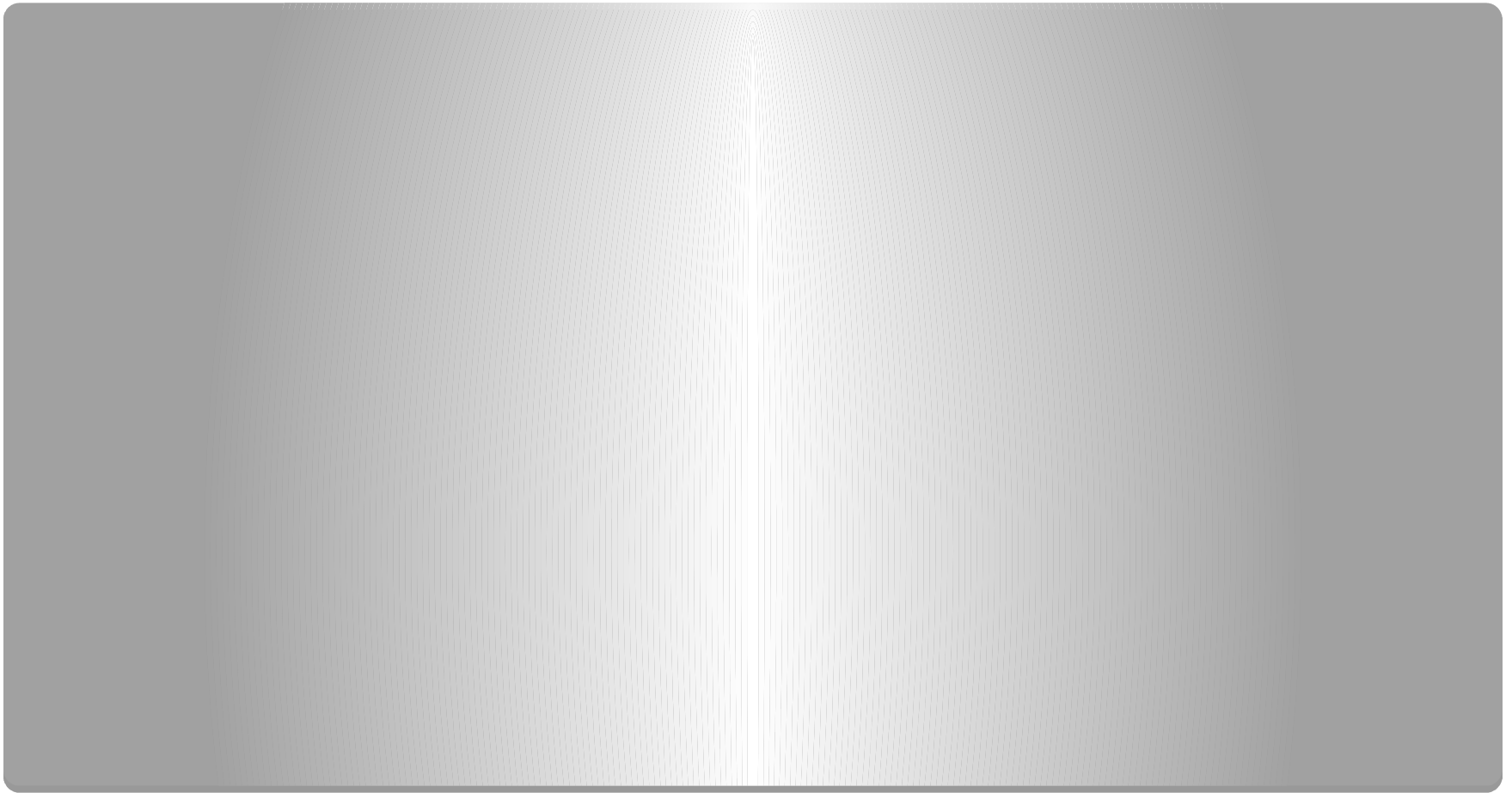


# Consultas de múltiples tablas y uniones

Segunda parte



foreign key

## Clave ajena o Foreign key

Campo que sirve para enlazar con otra tabla, en la cual es clave primaria.

Es el campo que sirve de enlace a los **join** y que expresamos en el **on**.

## Restricción foreign key

Método para asegurar la integridad de los datos en SQL. Definirá la clave ajena de una tabla, no permitiendo que mediante insert se pueda entrar valores diferentes de los establecidos en la otra tabla.

Con la restricción **FOREIGN KEY** se define el campo o conjunto de campos que deberán coincidir con la **primary key** o la **unique** de otra tabla.

Generará una **integridad referencial**, que asegurará que se mantenga el enlace entre las tablas en cualquier insert o update de la tabla donde se define y en cualquier update o delete de la tabla referenciada.

Formato de la instrucción que modifica la estructura de una tabla añadiéndole esa restricción Foreign key:

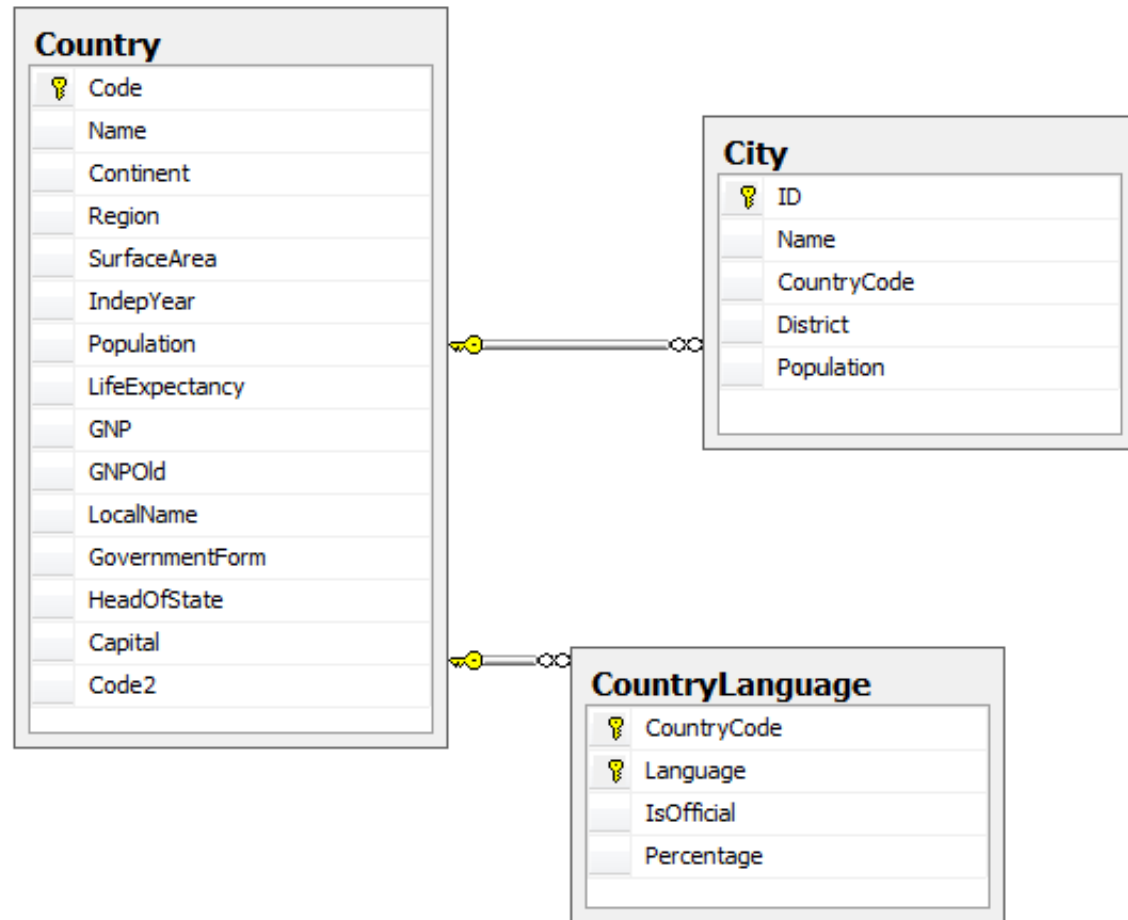
```
alter table TABLA1  
  add constraint NOMBRERESTRICCION  
  foreign key (CAMPOCLAVEFORANEA) references TABLA2 (CAMPOCLAVEPRIMARIA);
```



En Tabla2

**No es foreing**

# Mundo



Para esta Base de datos los enlaces serían...

```
alter table City  
add constraint FK_CityCountry  
foreign key (CountryCode)  
references Country (Code);  
go
```

```
alter table CountryLanguage  
add constraint FK_CountryLanguage  
foreign key (CountryCode)  
references Country (Code);  
go
```

Estos enlaces impedirían dar de alta una Ciudad de un país no existente.

Ni un idioma de un país inexistente.

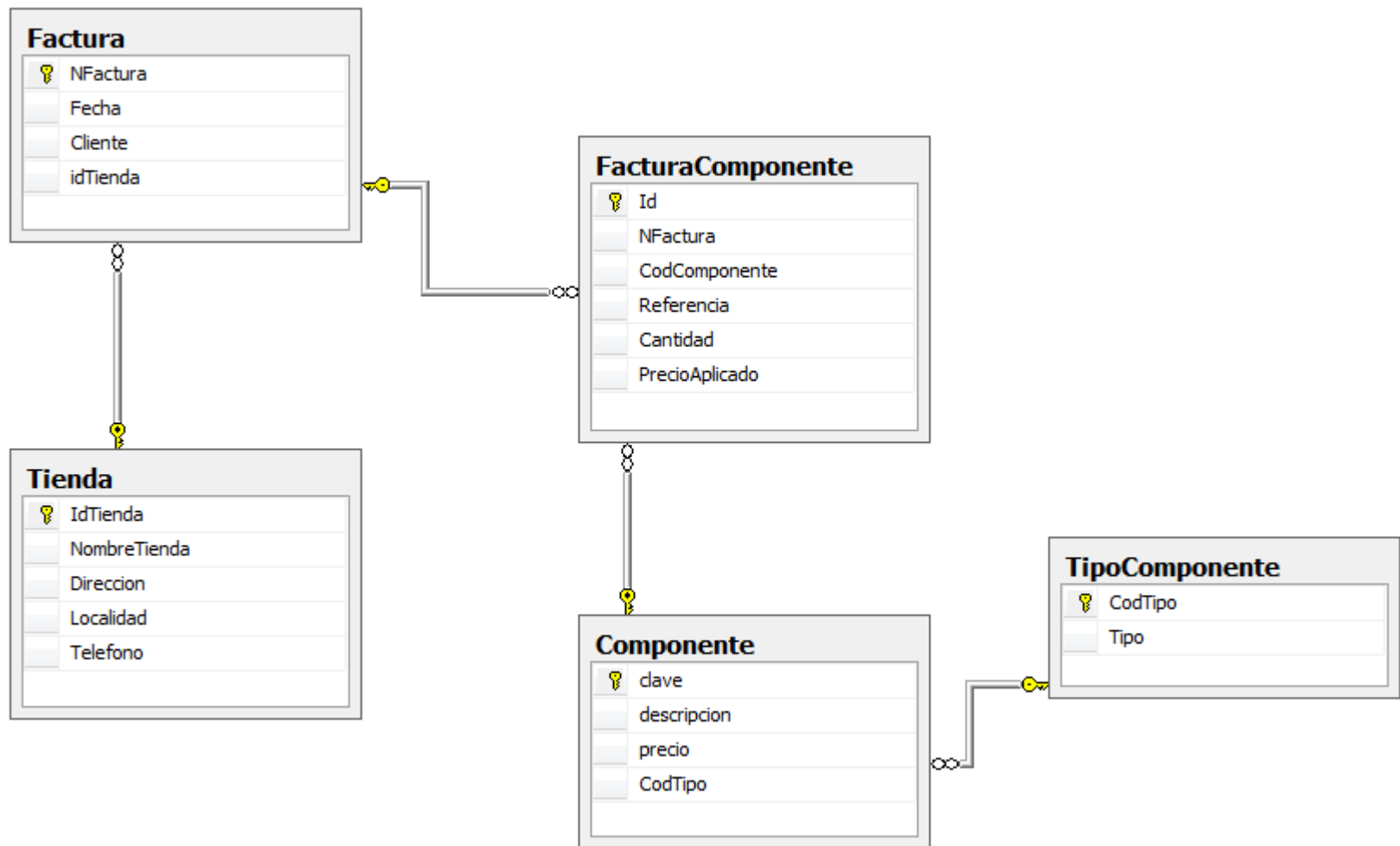
Funcionará cuando demos de alta Ciudades o Idiomas, no permitiéndonos hacerlo de países que no existan.

Tampoco permitirá eliminar países que tengan idiomas o ciudades asignadas.

Tampoco permitirá cambiar el Code del país si ese código tiene países asignados.



# Empresas informáticas



## En la Base de Datos EmpresasInformaticas...

```
alter table factura
add constraint FK_facturatienda
foreign key (idTienda)
references Tienda (idTienda);
go
```

```
alter table facturacomponente
add constraint FK_facturacomponente1
foreign key (Nfactura)
references Factura (NFactura);
go
```

```
alter table facturacomponente
add constraint FK_facturacomponente2
foreign key (CodComponente)
references Componente (clave);
go
```

```
alter table componente
add constraint FK_componentetipo
foreign key (CodTipo)
references TipoComponente (CodTipo);
go|
```

Si aplicamos la instrucción y algunos datos ya grabados no la cumplen, dará error y no se aplicará.

Lo que sí **permitirá**, si no indicamos que el campo no pueda ser nulo en la creación de la tabla, es que insertemos valores **NULL** en el campo o campos que definan la FOREIGN KEY.

Una tabla puede tener varias Foreign key, tantas como campos de referencia necesite.

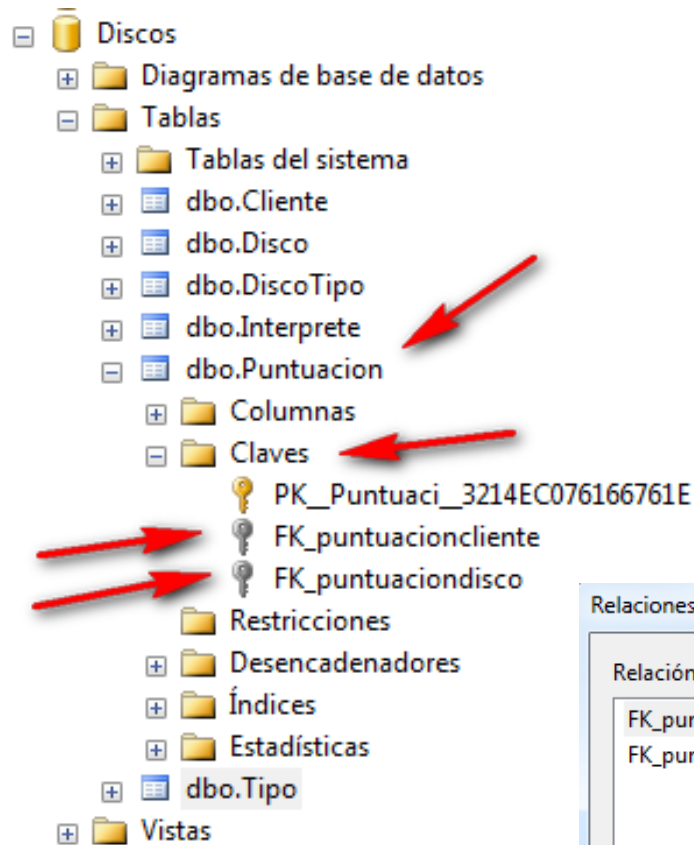
Las Foreign key no pueden modificarse. Se eliminan y se crean de nuevo, para que validen si los datos están en condiciones para aplicarse.

Las modificaciones en las características de los campos que las componen pueden afectarlas. Es mejor borrarlas y crearlas de nuevo.

Para ver información de la restricción podemos hacerlo mediante **sp\_helpconstraint** o mirando en el árbol de la Base de datos mediante el Management Studio.

```
use discos
go
sp_helpconstraint puntuacion
go
```

Resultados		Mensajes					
Object Name							
1	puntuacion						
	constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1	FOREIGN KEY	FK_puntuacioncliente	No Action	No Action	Enabled	Is_For_Replication	Idcliente
2							REFERENCES Discos.dbo.Cliente (Id)
3	FOREIGN KEY	FK_puntuaciondisco	No Action	No Action	Enabled	Is_For_Replication	iddisco
4							REFERENCES Discos.dbo.Disco (IdDisco)
5	PRIMARY KEY (clustered)	PK_Puntuaci__3214EC076166761E	(n/a)	(n/a)	(n/a)	(n/a)	Id



#### Relaciones de clave externa

Relación seleccionada:

FK\_puntuacioncliente  
FK\_puntuaciondisco

Editar propiedades para el tipo de relación existente.

Comprobar datos existentes al ☒ Sí

☒ Especificación de tablas y columnas

Columnas de clave externa Idcliente

Columnas de claves Primary id

Tabla base de clave externa Puntuacion

Tabla base de claves Primary Cliente

☒ Diseñador de tablas

☒ Especificación de INSERT y UPDATE

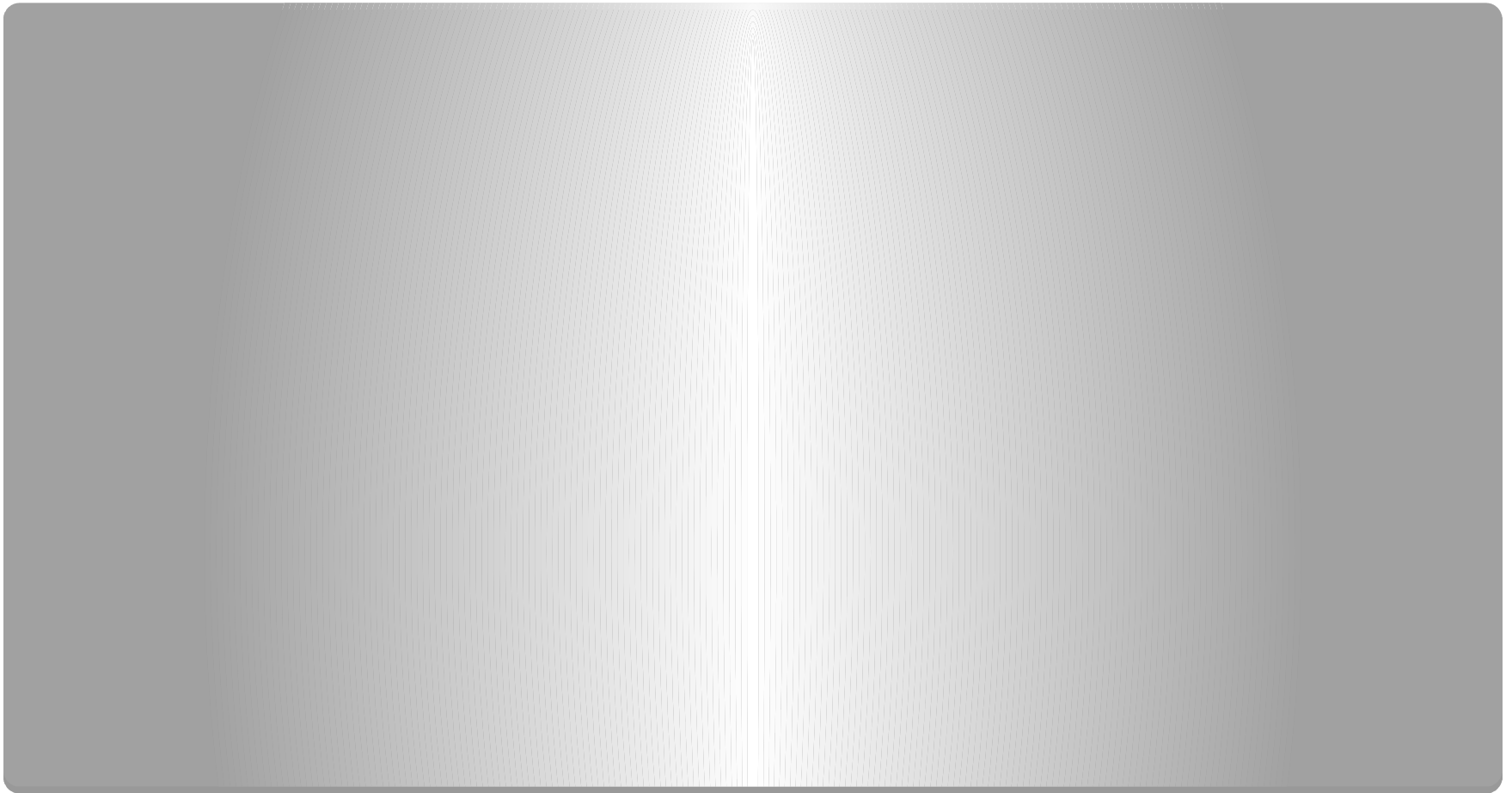
Exigir para replicación ☒ Sí

Exigir restricción de clave externa ☒ Sí

☒ Identidad

Agregar

Eliminar



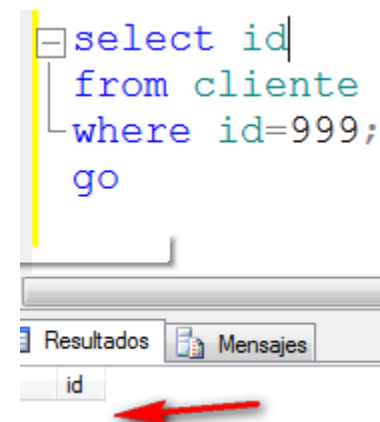
# Efecto del foreign key en operaciones

insert y update de la tabla con el foreign key  
delete y update de la tabla referenciada

Si intentamos insertar valor con idcliente o iddisco que no están en las respectivas tablas, nos dará error.

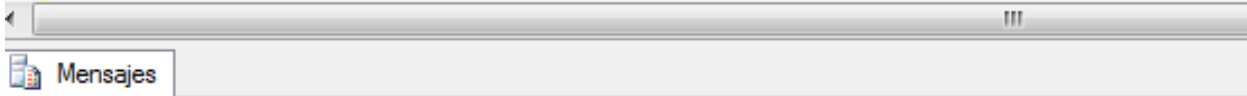
```
--insert tabla con foreign key
set dateformat dmy
go
insert into puntuacion
(id,idcliente,iddisco,puntuacion,fecha)
values
(9997,999,999,8,'12/01/2014')
GO
```

Mens. 547, Nivel 16, Estado 0, Línea 1  
Instrucción INSERT en conflicto con la  
restricción FOREIGN KEY  
"FK\_puntuacioncliente". El conflicto ha  
aparecido en la base de datos "Discos",  
tabla "dbo.Cliente", column 'id'.  
Se terminó la instrucción.



Sin embargo sí permitirá añadir valores null si la tabla no lo controla.

```
--insert tabla con foreign key valores null
set dateformat dmy
go
insert into puntuacion
(id,idcliente,iddisco,puntuacion,fecha)
values
(9997,null,null,8,'12/01/2014');|
GO
```

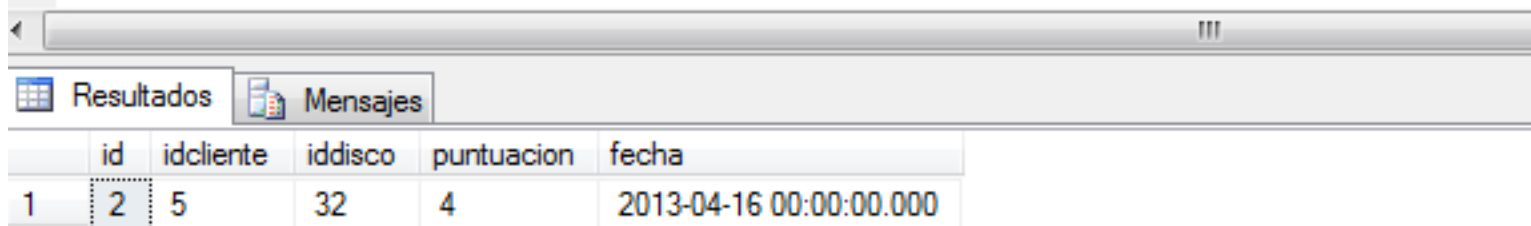


(1 filas afectadas)



Igualmente si intentamos modificarlo

```
select id,idcliente,iddisco,puntuacion,fecha
from puntuacion
where id=2;
go
```



The screenshot shows the SQL Server Enterprise Manager interface. At the top, there are tabs for 'Resultados' (Results) and 'Mensajes' (Messages). Below the tabs, a table displays the results of the previous query. The table has six columns: 'id', 'idcliente', 'iddisco', 'puntuacion', and 'fecha'. The first row contains the values 1, 2, 5, 32, 4, and 2013-04-16 00:00:00.000. The cell containing the value '2' in the 'id' column is highlighted with a dashed border.

	id	idcliente	iddisco	puntuacion	fecha
1	2	5	32	4	2013-04-16 00:00:00.000

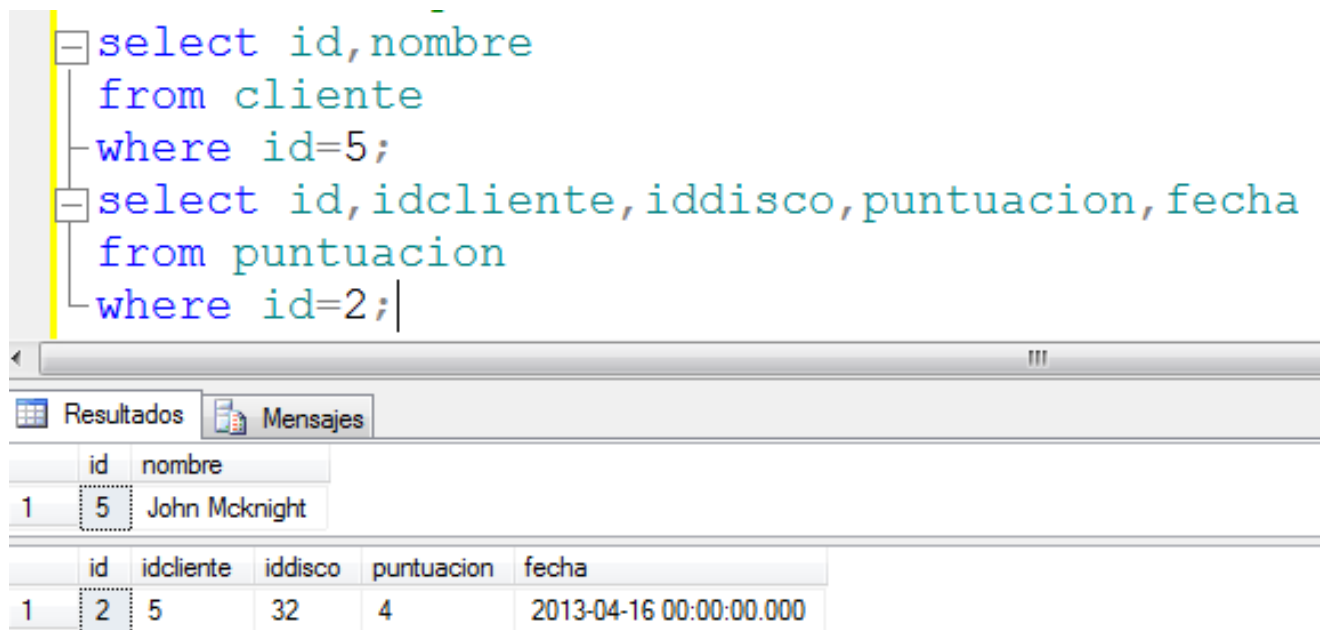
```
update puntuacion
set idcliente=999
where id=2;
go
```

Mens. 547, Nivel 16, Estado 0, Línea 1  
Instrucción UPDATE en conflicto con la  
restricción FOREIGN KEY  
"FK\_puntuacioncliente". El conflicto ha  
aparecido en la base de datos "Discos",  
tabla "dbo.Cliente", column 'id'.

Los controles sobre la tabla referenciada también existen.

No se permitirá borrar un elemento que tenga ya alguna referencia.

```
select id,nombre
from cliente
where id=5;
select id,idcliente,iddisco,puntuacion,fecha
from puntuacion
where id=2;
```



The screenshot shows a database query interface with two tabs: 'Resultados' (Results) and 'Mensajes' (Messages). The 'Resultados' tab is active, displaying two query results. The first query, 'select id,nombre from cliente where id=5;', returns a single row with id 5 and name John McKnight. The second query, 'select id,idcliente,iddisco,puntuacion,fecha from puntuacion where id=2;', returns a single row with id 2, idcliente 5, iddisco 32, puntuacion 4, and fecha 2013-04-16 00:00:00.000.

	id	nombre
1	5	John McKnight

	id	idcliente	iddisco	puntuacion	fecha
1	2	5	32	4	2013-04-16 00:00:00.000

Identificamos un cliente que tiene puntuacion grabada

Intentamos borrarlo...

```
delete cliente  
where id=5;
```

Mens. 547, Nivel 16, Estado 0, Línea 1  
Instrucción DELETE en conflicto con la  
restricción REFERENCE "FK\_puntuacioncliente".  
El conflicto ha aparecido en la base de datos  
"Discos", tabla "dbo.Puntuacion", column  
'Idcliente'.

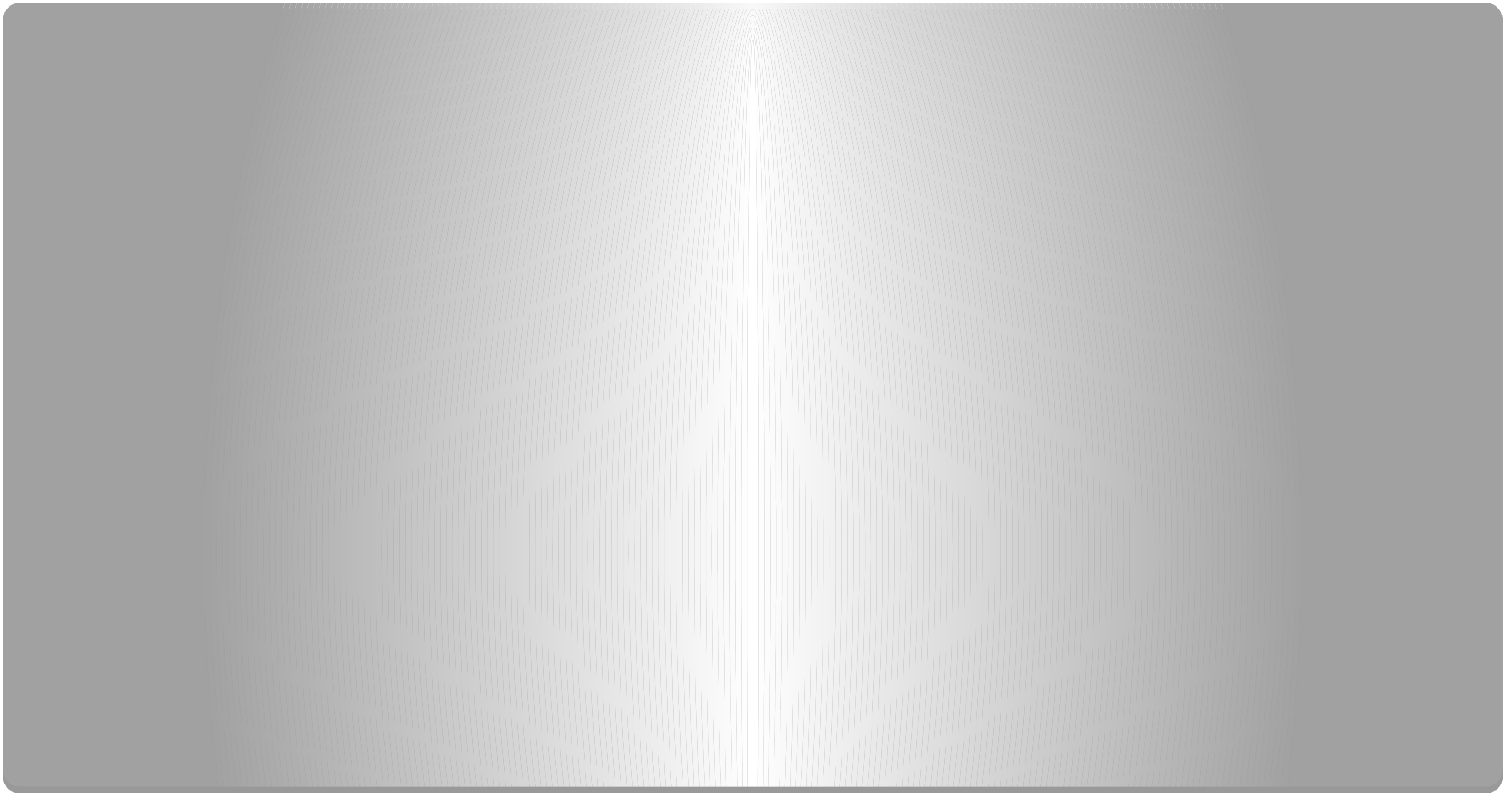
Tampoco se permitirá actualizar la clave principal de la tabla referenciada si algún registro hace referencia a ese valor.

Intentamos modificar el id de la tabla cliente...

```
update cliente  
    set id=999  
where id=5;
```

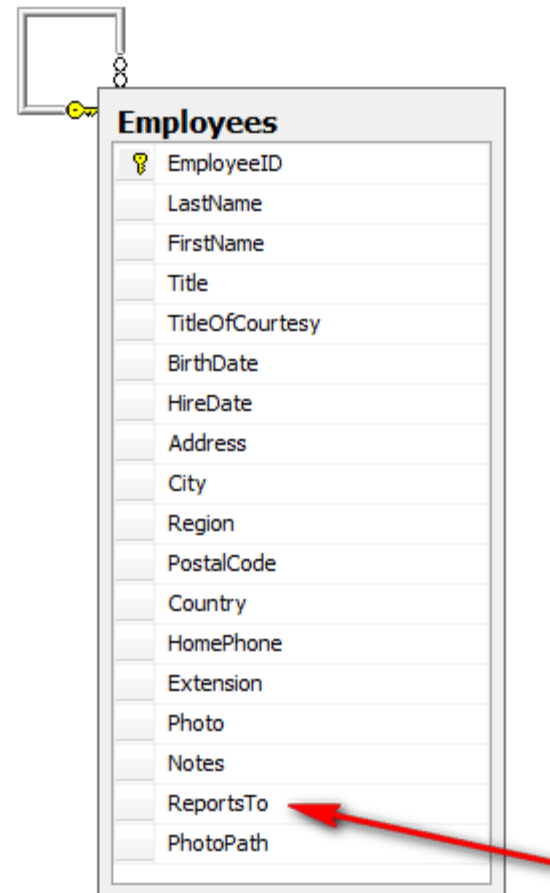
Mens. 547, Nivel 16, Estado 0, Línea 1  
Instrucción UPDATE en conflicto con la  
restricción REFERENCE  
"FK\_puntuacioncliente". El conflicto ha  
aparecido en la base de datos "Discos",  
tabla "dbo.Puntuacion", column 'Idcliente'.

No lo permite porque hay puntuacion con que hace referencia a ese id



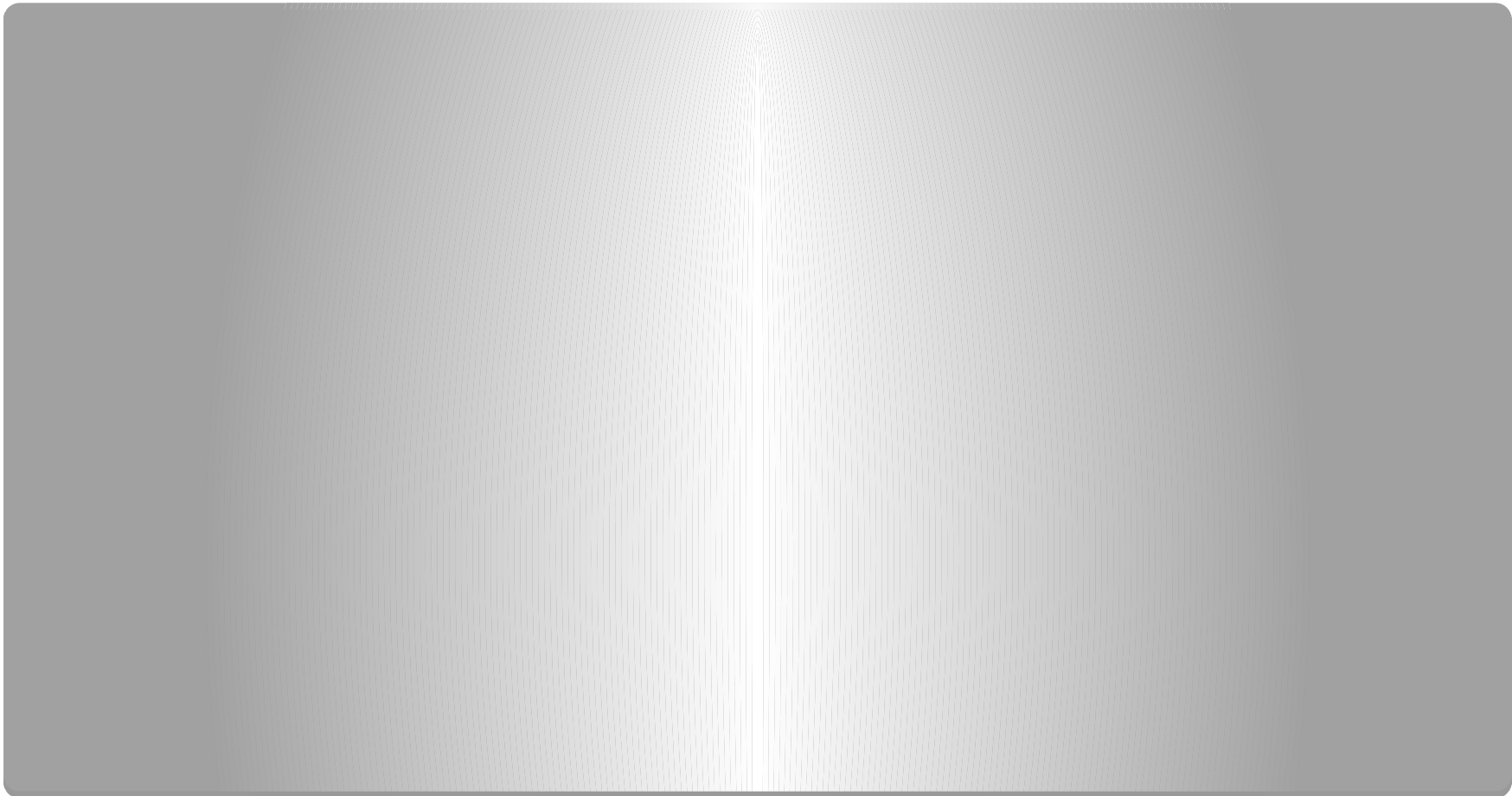
foreign key en la misma tabla

Se puede hacer referencia a la misma tabla.



```
- ALTER TABLE [dbo].[Employees]
  ADD CONSTRAINT [FK_Employees_Employees]
  FOREIGN KEY([ReportsTo])
  REFERENCES [dbo].[Employees] ([EmployeeID])
  GO
```

Obligaré a que el campo ReportsTo contenga la clave de un empleado o NULL



deshabilitar y eliminar restricciones



Si deshabilitamos una foreign key dejará de restringir la actualización de datos hasta que la activemos de nuevo.

Para desactivarla tendremos que detectar el nombre primero y después aplicar...

```
alter table NOMBRETABLA  
nocheck constraint NOMBRERESTRICCION;
```

Para activarla de nuevo...

```
alter table NOMBRETABLA  
check constraint NOMBRERESTRICCION;
```

Por ejemplo, vamos a desactivar la restricción que no nos permite modificar la clave de un cliente que tiene puntuaciones asignadas.

Al activar la restricción no validará los datos ya introducidos.

```
--buscar nombre de la restricción  
sp_helpconstraint puntuacion  
go
```

	constraint_type	constraint_name
1	FOREIGN KEY	FK_puntuacioncliente
-		

```
-- desactivar restricción  
alter table puntuacion  
    nocheck constraint FK_puntuacioncliente;  
go
```

```
--actualizar cliente
update cliente
    set id=999|
where id=5;
go
select id,nombre
from cliente
where id=999;
go
```

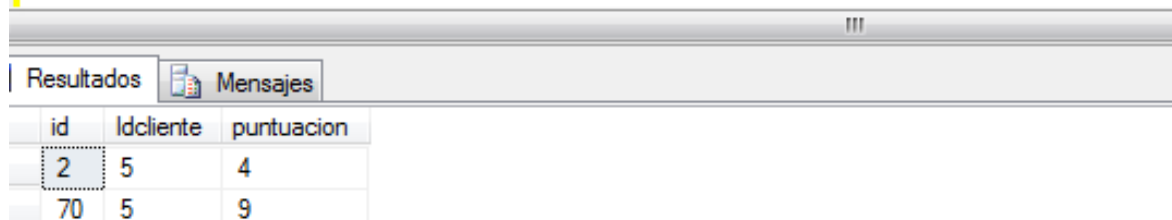
Resultados			Mensajes		
	id	nombre			
1	999	John Mcknight			

Como tenemos desactivada la restricción ahora permite ejecutar la modificación.

No dará problemas al activar la restricción, ya que no valida los datos, aunque hay puntuaciones del cliente modificado, ya que modificamos el id de la tabla clientes pero no el de la tabla puntuacion.

```
--activar restricción
alter table puntuacion
    check constraint FK_puntuacioncliente;
go

select id, Idcliente, puntuacion
from puntuacion
where idcliente=5;
go
```



Resultados		Mensajes	
id	Idcliente	puntuacion	
2	5	4	
70	5	9	

Si queremos cambiar otra cosa en la restricción debemos borrarla y crearla de nuevo.

Para borrar una restricción es con:

```
alter table TABLA  
  drop constraint NOMBRERESTRICCION;
```

```
--borrar restricción  
⌋ alter table puntuacion  
  ⌋ drop constraint FK_puntuacioncliente;  
go  
,
```

En nuestro ejemplo si la creamos de nuevo daría error, ya que hay datos que la incumplen.

```
--crearla de nuevo  
alter table puntuacion  
add constraint FK_puntuacioncliente  
foreign key (idcliente)  
references cliente (id);  
go
```

Mens. 547, Nivel 16, Estado 0, Línea 1  
Instrucción ALTER TABLE en conflicto con  
la restricción FOREIGN KEY  
"FK\_puntuacioncliente". El conflicto ha  
aparecido en la base de datos "Discos",  
tabla "dbo.Cliente", column 'id'.

Tenemos dos opciones:

**1.-** Crear la restricción con la opción nocheck y después activarla, con lo que no afecta a los registros ya grabados.

```
--crearla con nocheck y activarla
alter table puntuacion
with nocheck
add constraint FK_puntuacioncliente
foreign key (idcliente)
references cliente (id);
go
alter table puntuacion
check constraint FK_puntuacioncliente;
go
```

Mensajes

Comandos completados correctamente.



**2.-** Actualizar los datos que dan problemas y después borrar y crear de nuevo la restricción con el check activado.

```
--actualizar registros y crear la restricción
update puntuacion
    set idcliente=999
where idcliente=5;
go
alter table puntuacion
    drop constraint FK_puntuacioncliente;
go
alter table puntuacion
add constraint FK_puntuacioncliente
foreign key (idcliente)
references cliente (id);
go
select id, idcliente, puntuacion
from puntuacion
where idcliente=999;
go|
```

Resultados		Mensajes	
	id	idcliente	puntuacion
1	2	999	4
2	70	999	9



acciones automáticas del foreign key

Podemos configurar las restricciones Foreign key para que se comporten de otra forma.

### **Borrado en cascada:**

Que cuando eliminemos un elemento de la tabla referenciada elimine todos los elementos de la tabla con el foreign key.

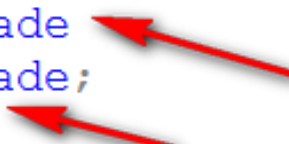
### **Actualización en cascada:**

Que cuando actualicemos un elemento de la tabla referenciada actualice el id de todos los elementos de la tabla con el foreign key.

Son **operaciones automáticas delicadas** que no piden confirmación y sólo deben hacerse en entornos que lo requieran para facilitar el mantenimiento de los datos.

Actualizamos la restricción (detectar el nombre, borrar y crear de nuevo)

```
--permitir borrado y actualización en cascada
[=] alter table puntuacion
    drop constraint FK_puntuacioncliente;
go
[=] alter table puntuacion
    add constraint FK_puntuacioncliente
    foreign key (idcliente)
    references cliente (id)
    on update cascade
    on delete cascade;
go
```



```
--miramos los datos
select id,nombre
from cliente
where id=999;
go
select id, idcliente,puntuacion
from puntuacion
where idcliente=999;
go
```

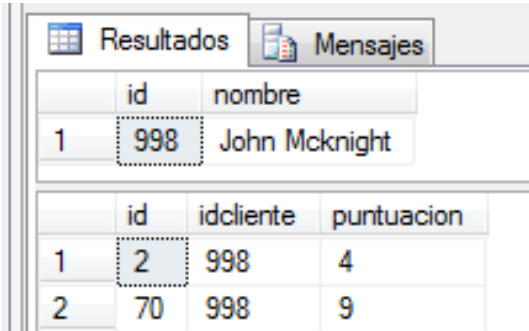
Resultados		Mensajes	
	id	nombre	
1	999	John Mcknight	

	id	idcliente	puntuacion
1	2	999	4
2	70	999	9

```
--modificamos el cliente  
update cliente  
    set id=998  
where id=999;  
go
```

```
--miramos los datos de nuevo  
select id,nombre  
from cliente  
where id=998;  
go  
select id, idcliente,puntuacion  
from puntuacion  
where idcliente=998;  
go
```



The screenshot shows a SQL query results window with two tabs: 'Resultados' (Results) and 'Mensajes' (Messages). The 'Resultados' tab is active, displaying two tables. The first table, 'cliente', has columns 'id' and 'nombre'. The second table, 'puntuacion', has columns 'id', 'idcliente', and 'puntuacion'. In both tables, the first row is highlighted with a dashed border.

	id	nombre
1	998	John Mcknight

	id	idcliente	puntuacion
1	2	998	4
2	70	998	9

Se modificaron los de las dos tablas

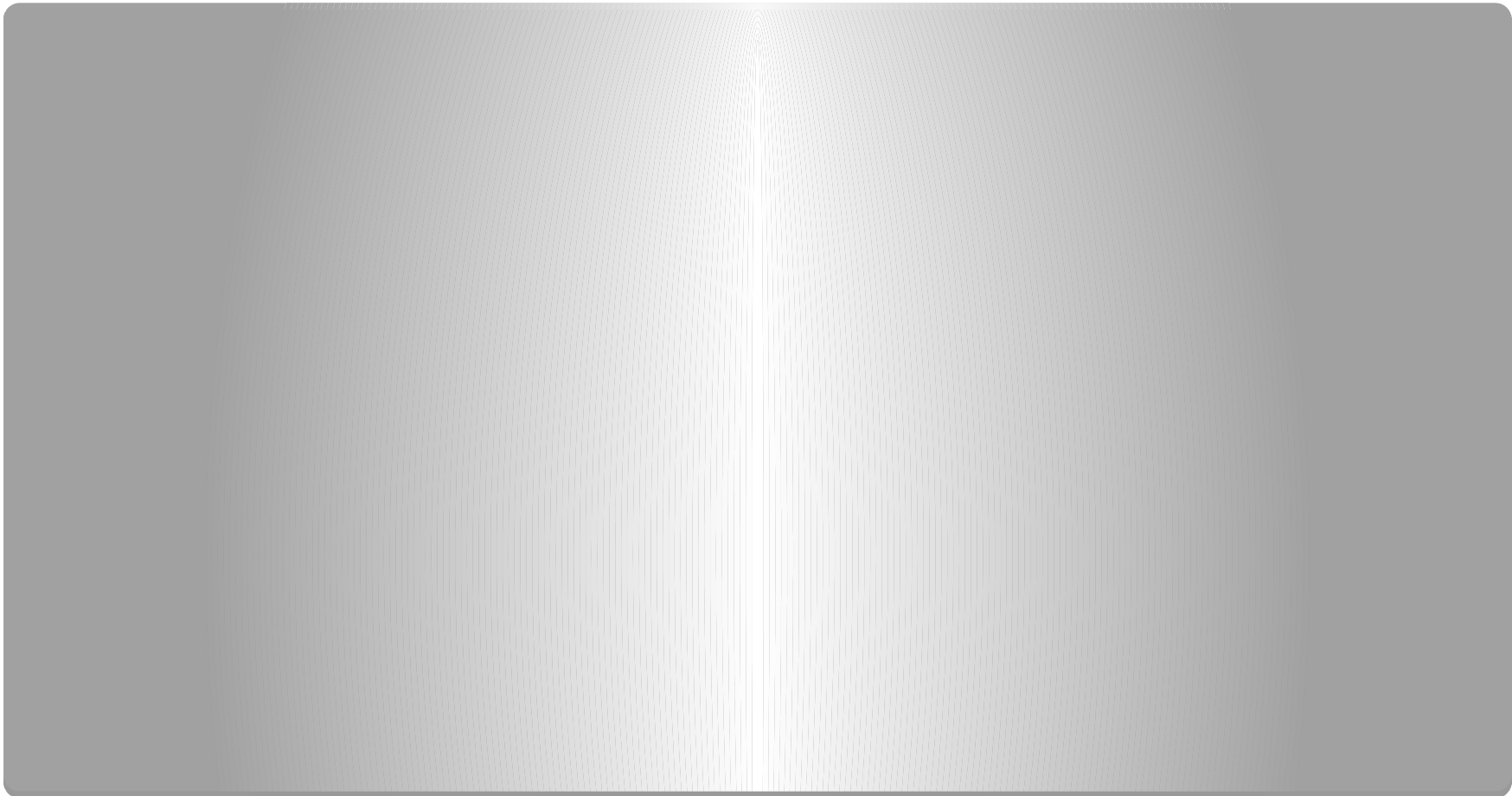
Comprobamos el efecto del on delete cascade.

```
--comprobar borrado en cascada
delete cliente
where id=998;
go

--miramos los datos de nuevo
select id,nombre
from cliente
where id=998;
go
select id, idcliente,puntuacion
from puntuacion
where idcliente=998;
```

Resultados		Mensajes	
id	nombre		
id	idcliente	puntuacion	

Se eliminaron los datos de ambas tablas.



Formato completo del foreign key



**Modificando** la estructura de la tabla

**Para crearla** (decidiendo entre los valores entre llaves, el subrayado es el valor por defecto):

```
alter table TABLA1
  with {check/nocheck}
  add constraint NOMBRERESTRICCION
  foreign key (CAMPOCLAVEFORANEA)
  references TABLA2 (CAMPOCLAVEPRIMARIA)
  on update {cascade/no action}
  on delete {cascade/no action}
;
```

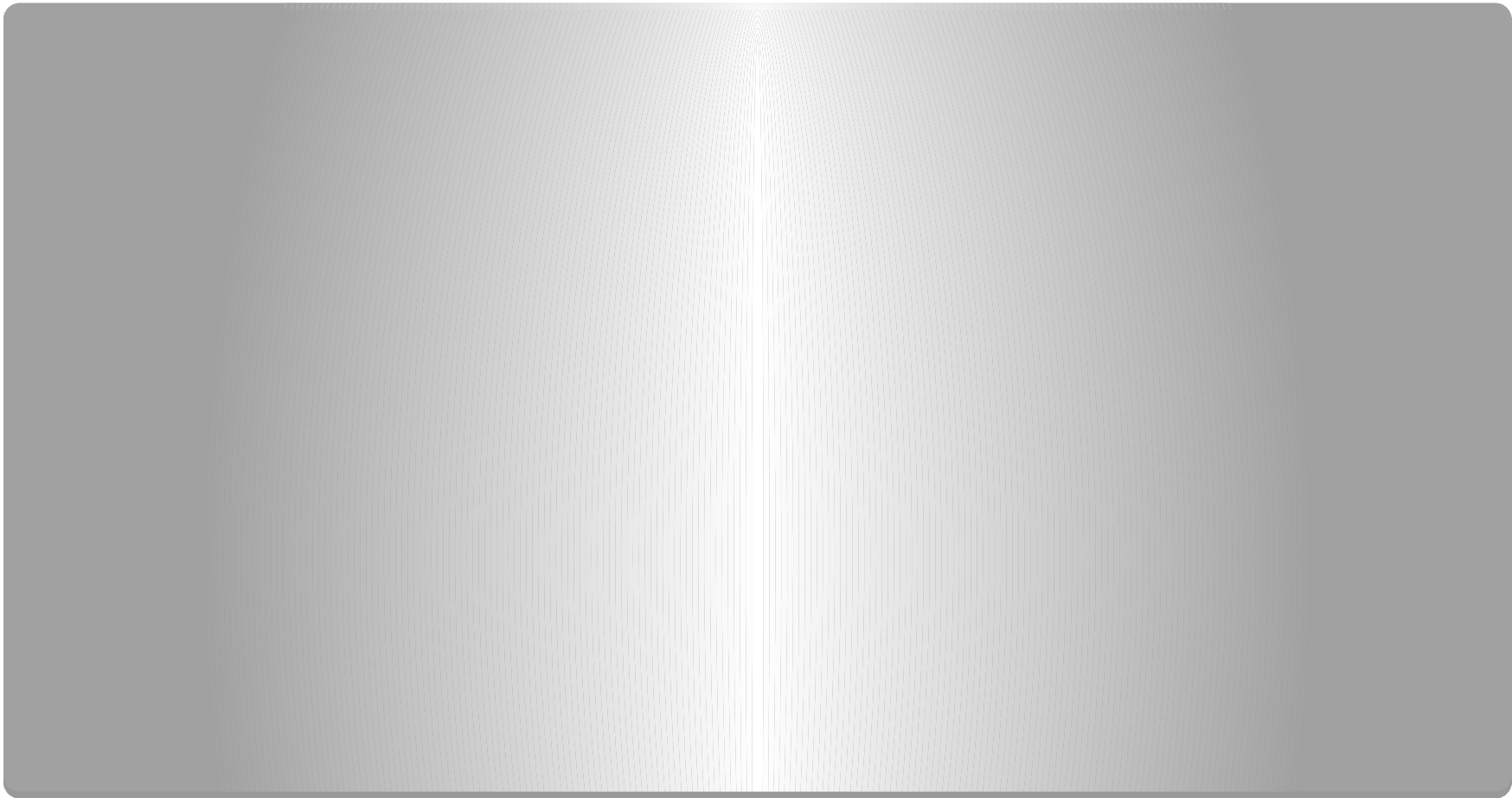
**Para borrarla:**

```
alter table TABLA1
  drop constraint NOMBRERESTRICCION;
```

Añadiéndolo en la **creación** de tablas...

```
create table TABLA1
( campo1 tipo... ,
  campo2 tipo... ,
  ... ,
  {constraint NombreRestricción}
  foreign key (CAMPOCLAVEFORANEA)
  references TABLA2 (CAMPOCLAVEPRIMARIA)
  on update {cascade/no action}
  on delete {cascade/no action},
  ...
);
```

Teniendo en cuenta que debe definirse el campo antes de usarlo en la definición de la Foreign key.  
Si no le damos nombre a la restricción el SGBD inventa uno.



union

El operador "union" **combina** el resultado de dos o más instrucciones "**select**" en un único resultado. Se usa cuando **los datos** que se quieren obtener **pertenecen a distintas tablas** y no se puede acceder a ellos con una sola consulta.

Es necesario que las tablas referenciadas tengan **tipos de datos similares**, la **misma cantidad de campos y el mismo orden** de campos en la lista de selección de cada consulta. No se incluyen las filas duplicadas en el resultado, a menos que coloque la opción "**all**".

Se deben especificar los nombres de los campos en la primera instrucción "select".

Para acceder a datos de distintas bases de datos lo hacemos especificando el nombre de la tabla como:

`basededatos.esquema.tabla`

Normalmente el esquema es dbo.

Tabla clientes de Videojuegos:

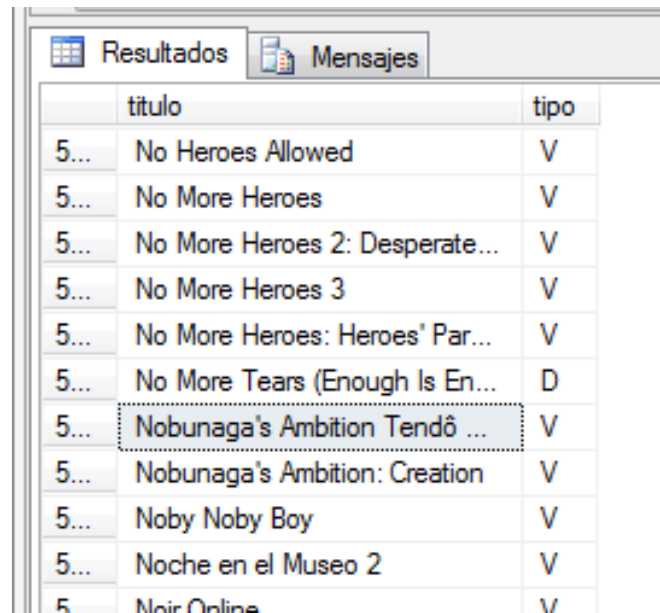
`videojuegos.dbo.cliente`

Tabla componentes de empresasinformaticas:

`empresasinformaticas.dbo.componente`

Vamos a ver los clientes de Videojuegos y discos, primero sin all...

```
select titulo, 'D' as tipo
from discos.dbo.disco
union
select juego , 'V' as tipo
from videojuegos.dbo.juego;
go
```



	titulo	tipo
5...	No Heroes Allowed	V
5...	No More Heroes	V
5...	No More Heroes 2: Desperate...	V
5...	No More Heroes 3	V
5...	No More Heroes: Heroes' Par...	V
5...	No More Tears (Enough Is En...	D
5...	Nobunaga's Ambition Tendô ...	V
5...	Nobunaga's Ambition: Creation	V
5...	Noby Noby Boy	V
5...	Noche en el Museo 2	V
5...	Noir Online	V

Veamos la unión con las tablas de tablaprueba...

## Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

## Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

```
select id,valor1  
from Tabla1  
union  
select id,valor2  
from Tabla2;  
go
```

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto
6	6	Sexto
7	7	Séptimo
8	8	Octavo

No repite registros

Cuando usamos all no unifica registros con valores iguales...

## Tabla1

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto

## Tabla2

	id	valor2
1	1	Primero
2	2	Segundo
3	3	Tercero
4	6	Sexto
5	7	Séptimo
6	8	Octavo

```
select id,valor1
from Tabla1
union all
select id,valor2
from Tabla2;
go
```

	id	valor1
1	1	Primero
2	2	Segundo
3	3	Tercero
4	4	Cuarto
5	5	Quinto
6	1	Primero
7	2	Segundo
8	3	Tercero
9	6	Sexto
10	7	Séptimo
11	8	Octavo

Repite los valores de registros con union all



También podemos hacerlo con más select.  
Sólo tenemos que validar que el nº de campos es igual y  
que los tipos se corresponden.  
La estructura de campos resultante tendrá la misma  
denominación que el primer select.

```
select ...  
  union {all}  
select ...  
  union {all}  
select...  
...
```