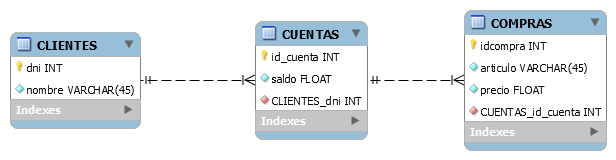
Cuando estamos tratando con [bases de datos relacionales](https://es.wikipedia.org/wiki/Base_de_datos_relacional), estamos implementando una robustez en el tratamiento de los datos que queremos guardar. Pero la gestión de este modelo puede volverse complicado si no conocemos en profundidad este concepto relacional y a la hora de desarrollar podemos perder mucho tiempo creando soluciones muy parciales en lugar de utilizar algunas características que nos ofrece este modelo. En este caso concreto vamos a hablar sobre la opción **ON DELETE CASCADE / ON  UPDATE CASCADE.**

Esta es quizás una de las opciones más utiles y peor explicadas en cualquier tutorial o página oficial de bases de datos. Realmente no es tan complicado pero vamos a realizar varias pruebas para entenderlo y llevarlo a cabo.

Vamos a crear un ejemplo de base de datos y veremos las características de estas opciones. Para empezar se van a crear 3 tablas (**CLIENTES**, **CUENTAS** y **COMPRAS**). Estas 3 tablas irán relacionadas entre sí con una relación 1:n entre **CLIENTES** y **CUENTAS** y una relación 1:n entre **CUENTAS** y **COMPRAS**. Es decir, existe una persona que puede tener abierta una cuenta de banco con unas características concretas y con la que realizará unas compras; pero puede también tener varias  cuentas y cuando no le cojan una tarjeta poder pagar con la otra.

El modelo se visualizaría de la siguiente manera:



Insertamos los valores

-- -----------------------------------------------------

-- Data for table `CLIENTES`

-- -----------------------------------------------------

INSERT INTO `CLIENTES` (`dni`, `nombre`) VALUES (123456789, 'John Carter');

INSERT INTO `CLIENTES` (`dni`, `nombre`) VALUES (987654321, 'JohnQ');

INSERT INTO `CLIENTES` (`dni`, `nombre`) VALUES (193749372, 'John Doe');

-- -----------------------------------------------------

-- Data for table `CUENTAS`

-- -----------------------------------------------------

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (10, 1000, 123456789);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (11, 0, 123456789);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (12, 20, 123456789);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (20, 40.23, 987654321);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (21, 2350.48, 987654321);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (30, 12000000, 193749372);

INSERT INTO `CUENTAS` (`id\_cuenta`, `saldo`, `CLIENTES\_dni`) VALUES (31, 999999999999, 193749372);

-- -----------------------------------------------------

-- Data for table `COMPRAS`

-- -----------------------------------------------------

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (1, 'Barra de pan', 1, 10);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (2, 'Cuota del gimnasio', 30, 10);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (3, 'Caja de cervezas 12 latas', 5, 10);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (4, 'Calzoncillos Boxer', 3, 12);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (5, 'Pack DVD Juego de tronos', 36, 20);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (6, 'Lavadora', 300, 21);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (7, 'Lavavajillas', 460, 21);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (8, 'Jacuzzi de Ikea', 3200, 30);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (9, 'Boeing 747', 17000000, 30);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (10, 'Servicio de compañia de lujo', 1350, 30);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (12, 'Empresa Google', 173652000000, 31);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (11, 'Asesino a sueldo', 3450, 30);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (13, 'Empresa Apple', 246992000000, 31);

INSERT INTO `COMPRAS` (`idcompra`, `articulo`, `precio`, `CUENTAS\_id\_cuenta`) VALUES (14, 'Empresa Microsoft', 115500000000, 31);

Una vez hecho esto, vamos a realizar una cosa tan sencilla, como eliminar un cliente cualquiera y descubriremos el siguiente error.

**Cannot delete or update a parent row: a foreign key contraint fails**

Este error aparece, debido a que se quiere eliminar un registro que tiene asociados un identificador en otras tablas. De manera que si solo eliminamos este registro, en las otras tablas existirían registros de un usuario inexistente y se perdería el modelo relacional.

Aunque pueda parecer un engorro, esto nos ayuda en un futuro a mantener una robustez que nos permite desarrollar aplicaciones más complejas y seguras. El objetivo final es no duplicar información entre tablas y que no haya ninguna fuga de consistencia que en un futuro pueda dar problemas.

Lo más cómodo sería que al borrar un usuario se borrarán también los campos asociados de otras tablas. De esta manera, deberemos cambiar nuestro modelo de base de datos al siguiente, solamente modificando la línea ON DELETE NO ACTION por ON DELETE CASCADE.

Introducimos los mismos registros que habiamos cargado anteriormente y probamos a eliminar un registro de clientes. Observaremos que se eliminarán todos los registros asociados a ese cliente.

Si actualizamos una tabla padre también observaremos que el resto de tablas se actualizan con ese identificador, por ejemplo:

DELETE FROM `cuentas` WHERE `cuentas`.`id\_cuenta` = 10;

DELETE FROM `clientes` WHERE `clientes`.`dni` = 987654321