

Sprint 13. Práctica. 1

Crea una base relacional de ejemplo utilizando un documento de texto o a mano. Diseñala para que contenga las siguientes tablas, y establece las relaciones que consideres necesarias entre ellas:

- Tabla de compras (transacciones)
- Tabla de establecimientos
- Tabla de trabajadores
- Tabla de clientes
- Tabla de productos
- Tabla de tipo de productos

Define las propiedades de las relaciones (1:1, 1:n, n:n), y crea algunas de las variables que podrían contener.

Resolución:

Lo primero que hacemos es crear el modelo entidad relaciona. Partiremos de la definición de atributos. Luego, a través de la relación universal, iremos normalizando la base de datos hasta la cuarta forma normal, lo que nos irá dando las entidades(tablas) a las que luego buscaremos una relación. Por último, crearemos interrelaciones entre tablas, mediante claves foráneas, así como posible atributos nuevos para definir las interrelaciones.

1. Empezamos definiendo los atributos y la relación universal:

En cada atributo pondremos si los valores son únicos(no pueden repetirse) o si admite valores únicos.

NumTienda: es el número de identificación del establecimiento. Los valores son únicos, y no admite nulos

NombreTienda: Nombre del establecimiento. Los valores son únicos, y no admite nulos

DirecciónTienda: dirección del establecimiento. Los valores son únicos, y no admite nulos

IDCliente: Número de identificador del cliente. Los valores son únicos, y no admite nulos. Podría haber un número de identificador para un cliente genérico que no es identificable.

Nombrecliente: nombre del cliente. Admite nulos. Un cliente puede no identificarse, y pueden encontrarse valores repetidos. Ante de falta de información del cliente se pueden crear números de **IDCliente** nuevos con los datos de nombre y teléfono vacíos

Telcliente: teléfono del cliente Admite nulos. Un cliente puede no identificarse, y pueden encontrarse valores repetidos. Ante de falta de información del cliente se pueden crear números de **IDCliente** nuevos con los datos de nombre y teléfono vacíos

Idtrabajador: número de empleado. Los valores son únicos, y no admite nulos

NombreTrabajador: nombre y apellidos del trabajador. Los valores NO son únicos, y no admite nulos,

Tfnotrabajador: teléfono del trabajador. Los valores son únicos, y no admite nulos

Idproducto: identificador de producto. Los valores son únicos, y no admite nulos

Nombreproducto: Nombre del producto. Pueden haber productos duplicados y no admite nulos.

ClaseProducto: tipo de producto. Los valores son únicos, y no admite nulos

IDpedido. Número de identificación del pedido. Los valores son únicos, y no admite nulos

Cantidad: número de cantidad de compras de un producto. Los valores son enteros, y no admite nulos

Precio: Precio de venta de un producto. Valores decimales y no admite nulos. No vinculo el precio al pedido ya que abro la posibilidad de que cada tienda tenga su política de descuentos u ofertas, o no tengan precios unificados

Fechapedido: Fecha en la que se hizo el pedido

Por último definimos la Relación Universal de todos los atributos como :

RELACIÓN UNIVERSAL (NumTienda, NombreTienda, DirecciónTienda, IDCliente, Nombrecliente, Telcliente, Idtrabajador, NombreTrabajador, Tfnotrabajador, Idproducto, Nombreproducto, ClaseProducto, IDpedido, Cantidad, Precio, Fechapedido)

En los siguiente, vamos a ir normalizando la case relacional para obtener las distintas tablas.

1.1 Primera Forma normal: Definimos que todos los valores para cada registro(tupla-fila) de cada atributo son indivisibles. Por lo que cada valor en cada celda de cada tabla(entidad) ha de ser indivisible.

En verdad, esto no aporta nada al tratamiento teórico de la entidad relación que estamos diseñando. Pero al quedar definido así es imposible que una celda sea multivaluada.

1.2 Dependencias Funcionales.

Miramos las dependencias funcionales de los valores:

Podemos ver que NumTienda, NombreTienda y DirecciónTienda podrían ser perfectamente claves candidatas para identificar una tienda, ya que los tres atributos podrían identificar unívocamente la tienda. Pero optaremos por NumTienda, ya que al ser un valor numérico daría pie a menos fallos a la hora de introducir datos, como sería un nombre o la dirección. Así que decimos que NombreTienda y DirecciónTienda tienen dependencia funcional de NumTienda.

IDCliente, Nombrecliente, TelCliente nos informan de los datos del cliente. En este caso la dependencia funcional es de NombreCliente y Teléfono cliente respecto a IdCliente, ya que NombreCliente y Teléfono no son únicos, incluso pueden admitir valores únicos. El IDCliente define perfectamente al cliente. El nombre del cliente no, ya que pueden haber dos cliente con los mismo nombres y apellidos, y el teléfono tampoco ya que un cliente puede no tener teléfono, tenerlo compartido con otros clientes (un teléfono fijo), o tener valor nulo.

El mismo razonamiento que hemos usado para IDCliente es igual de valido para IDtrabajador. Tanto NombreTrabajador como Tfnotrabajador dependen de IDtrabajador..

Respeto al producto tenemos Idproducto, Nombreproducto, ClaseProducto. Podemos ver que el idproducto es determinante de Nombreproducto. A su vez ClasedepProducto tiene dependencia funcional con Nombredeproducto, por lo que a su vez tienen dependencia transitiva con IDProducto.

NumTienda → NombreTienda , DirecciónTienda

IDCliente → Nombrecliente, TelCliente

Idtrabajador → NombreTrabajador, Tfnotrabajador

Idproducto → Nombreproducto→ ClaseProducto

IDpedido→ Fechapedido

Por otro lado, Cantidad y Precio podrían tener dependencia funcional con IdPedido pero a su vez con otras variables foráneas (como más adelante podremos plantear) como idproducto, IDCliente, Idtrabajador incluso número de tienda.

1.3 Transformación en Segunda Forma Normal.

Podríamos definir como clave candidata el subconjunto de atributos **NumTienda**, **IDCliente**, **Idtrabajador**, **Idproducto**, **IDpedido**, puesto que estas cinco claves son candidatas y definen unívocamente el resto del Data Set, pero la dependencia no sería una dependencia total, ya que muchas de variables dependerían sólo de un subconjunto del DataSet. Por ejemplo, la dependencia funcional de NombreTrabajador sería con el subconjunto de la clave candidtata de Idtrabajador.

Así que creamos las 6 relaciones siguientes para que los atributos que no sean clave, dependan totalmente de la clave principa.

ESTABLECIMIENTO (NumTienda, NombreTienda , DirecciónTienda)

TRABAJADOR(Idtrabajador, NombreTrabajador, Tfnotrabajador)

CLIENTE (IDCliente, Nombrecliente, TelCliente)

PRODUCTO(Idproducto , Nombreproducto, ClaseProducto)

PEDIDO(IDpedido, Fechapedido)

COMPRA(IDpedido, Idproducto Idtrabajador , IDCliente , Cantidad, Precio) DONDE {IDpedido} REFERENCIA PEDIDO, DONDE { Idproducto } REFERENCIA PRODUCTO, DONDE { Idtrabajador} REFERENCIA TRABAJADOR , DONDE {IDCliente} REFERENCIA CLIENTE

Nota: en la relación Compra se tiene en cuenta la venta, quien hizo la venta, quien lo compró, que se vendió , cuantas unidades se compraron y el precio.

1.4 transformación en tercera forma normal.

Buscamos eliminar las dependencias transitivas, por lo que transformamos la tabla producto en dos entidades

PRODUCTO(Idproducto , Nombreproducto) DONDE { Nombreproducto} REFERENCIA TIPO

TIPO (Nombreproducto, ClaseProducto)

Por lo que las relaciones quedarían de la siguiente forma.

ESTABLECIMIENTO (NumTienda, NombreTienda , DirecciónTienda)

TRABAJADOR(Idtrabajador, NombreTrabajador, Tfnotrabajador)

CLIENTE (IDCliente, Nombrecliente, TelCliente)

PRODUCTO(Idproducto , Nombreproducto) DONDE { Nombreproducto} REFERENCIA TIPO

TIPO (Nombreproducto, ClaseProducto)

PEDIDO(IDpedido, Fechapedido)

COMPRA(IDpedido, Idproducto Idtrabajador , IDCliente , NumTienda Cantidad, Precio) DONDE {IDpedido}, REFERENCIA PEDIDO, DONDE { Idproducto } REFERENCIA PRODUCTO, DONDE { Idtrabajador} REFERENCIA TRABAJADOR , DONDE {IDCliente} REFERENCIA CLIENTE , DONDE {NumTienda} REFERENCIA ESTABLECIMEINTO

1.5 Transformación en 4FN.

Buscamos eliminar las dependencias multivalente. Podemos ver una dependencia multivalente en COMPRA. Ya que subconjuntos de la clave principal nos generan valores multivaluados.

.

Para resolver el problema deberíamos generar 3 nuevas relaciones que nos preserven la información y las relaciones. Tomaremos como clave principal el IDpedido y el IDproducto ya que ambos son necesarios para identificar las ventas. Aún así no podremos solventar la dependencia multivaluada.

Proponemos las siguientes 3 relaciones:

VENDEDOR(IDpedido, Idproducto Cantidad, Precio) DONDE {IDpedido}, REFERENCIA PEDIDO, DONDE { Idproducto } REFERENCIA PRODUCTO, DONDE { Idtrabajador} REFERENCIA TRABAJADOR
 COMPRADOR (IDpedido, Idproducto , IDcliente , NumTienda Cantidad, Precio) DONDE {IDpedido}, REFERENCIA PEDIDO, DONDE { Idproducto } REFERENCIA PRODUCTO, DONDE {IDcliente} REFERENCIA VENTA_LOCAL(IDpedido, Idproducto , NumTienda Cantidad, Precio) DONDE {IDpedido},REFERENCIA PEDIDO, DONDE { Idproducto } REFERENCIA PRODUCTO, DONDE {NumTienda} REFERENCIA ESTABLECIMEINTO

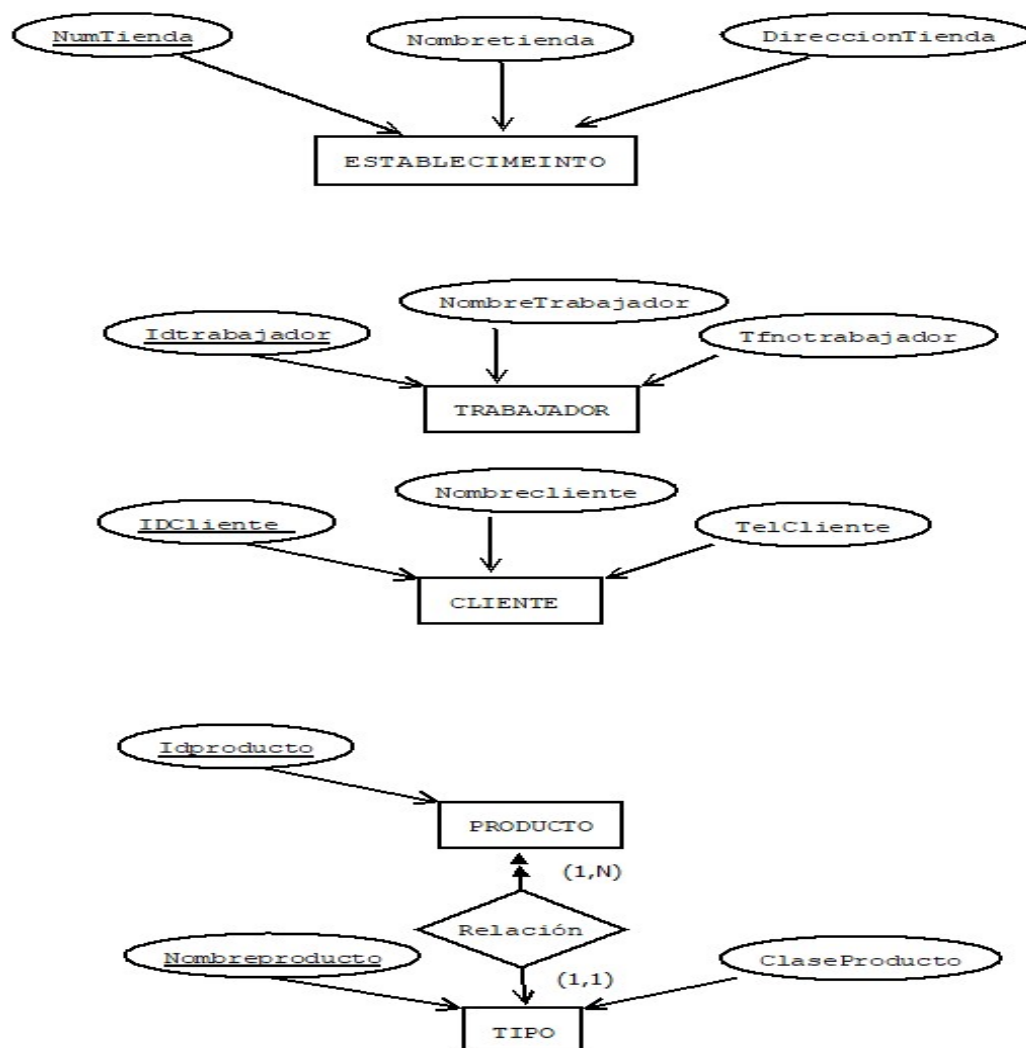
Que las añadimos a las 6 ya exixtentes.

ESTABLECIMIENTO (NumTienda, NombreTienda , DirecciónTienda)
 TRABAJADOR(Idtrabajador, NombreTrabajador, Tfnotrabajador)
 CLIENTE (IDcliente, Nombrecliente, TelCliente)
 PRODUCTO(Idproducto , Nombreproducto) DONDE { Nombreproducto} REFERENCIA TIPO
 TIPO (Nombreproducto, ClaseProducto)
 PEDIDO(IDpedido, Fechapedido)

No aspiramos a la Quinta forma Normal.

2. Definimos el modelo Relacional y usando lo a aplicación de Diagramas Dia.

2.1.Definimos las entidades



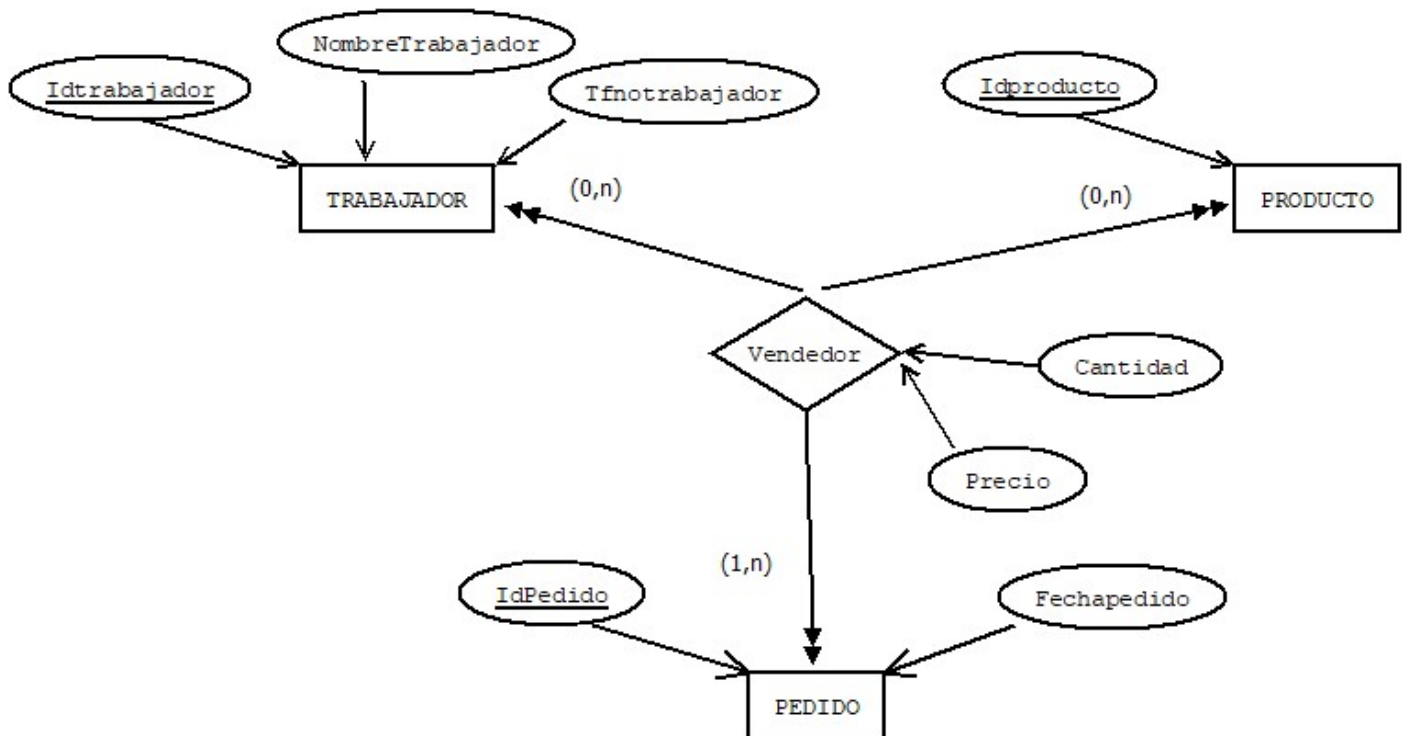
Más abajo miramos la interrelación 1:N

2.2. Interrelaciones.

Vamos a ver unas cuantas interrelaciones de ejemplo entre distintas entidades

a) Interrelación M-N-P

Ahora vamos a realizar las relaciones entre las interrelaciones de VENDEDOR que serían extrapolables a COMPRADOR y VENTALOCAL.



Justificamos esta interrelación.

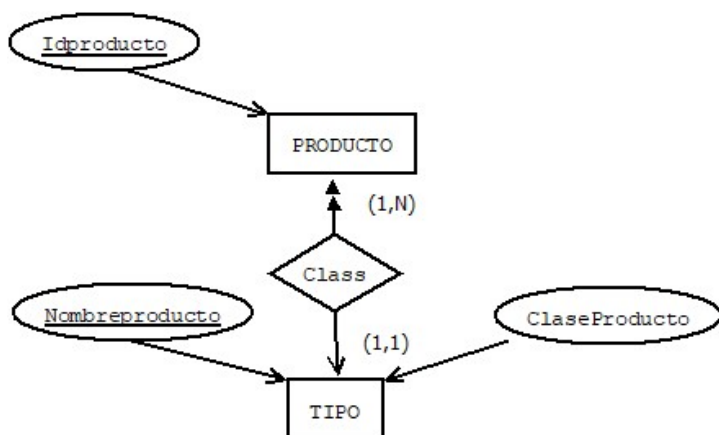
Si nos centramos en la Instancia TRABAJADOR, varios pedidos con distintos productos pueden ser vendidos por el mismo trabajador. Puede ser 0 la instancia mínima ya que hay trabajadores que no tienen que ser vendedores (por ejemplo contables o empleados de mantenimiento) y por lo tanto no vender nada.

Si nos centramos en Producto, vemos que un IDproducto puede ser instanciado por varios pedidos y por varios trabajadores, a la vez que puede no estar implicado en ninguna venta.

Si miramos el término Pedido, mínimo tiene que haber uno en cualquier venta, vemos que será ejecutado por un solo trabajador pero puede llamar a varios productos en la venta.

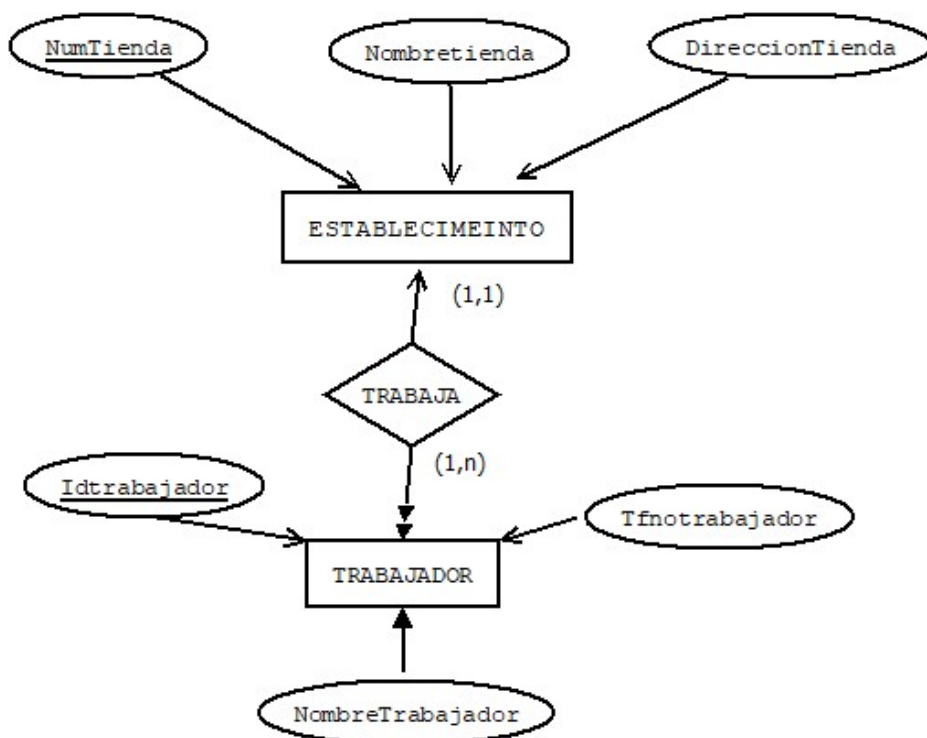
En el caso que cambiásemos Trabajador por Cliente, y la Interrelación la llamáramos Compra (por decir), lo único que cambiaría serían los mínimos y máximos sobre el Cliente que pasarían a ser de (1,n) en vez de (0,n) como el trabajador. Ya que damos por hecho que un cliente se da de alta cuando realiza al menos una compra.

b) Interrelación 1-N.



Podemos ver que la Relación lo que hace es clasificar un producto.

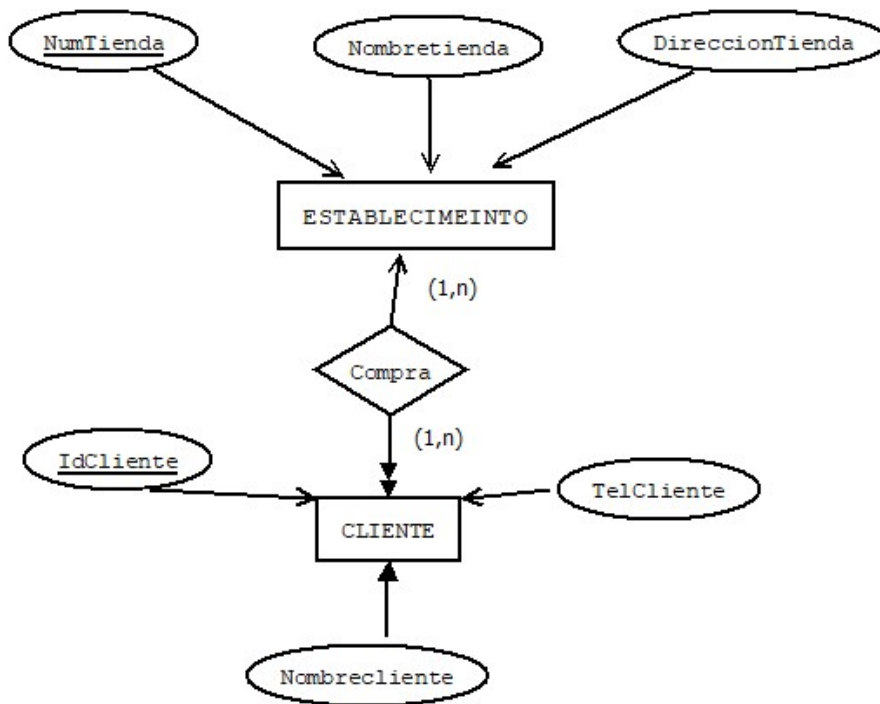
- Un Producto instanciado a Tipo sólo puede tener una clase (1,1)
- Mientras que un clase puede tener instanciados distintos productos (1,N) de manera multivaluada.



Podemos ver la relación de empleados que Trabajan en un establecimiento

- Un trabajador tiene que trabajar en un establecimiento
- Un establecimiento tiene de 1 a N empleados.

c) Relaciones N:M.



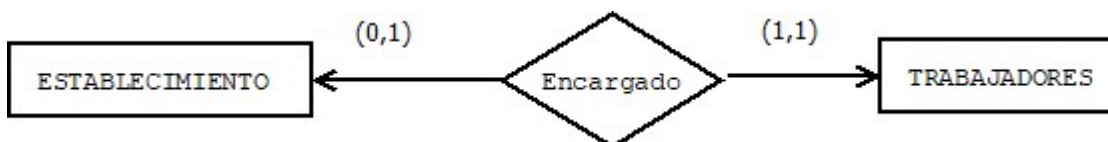
Podemos ver que :

Un cliente puede comprar en distintos establecimientos. Ponemos mínimo uno, porque damos por hecho que ha hecho al menos un pedido para ser dado de alta de cliente

Y a su vez establecimiento puede tener varios clientes.

d) Relaciones 1:1.

Podríamos establecer una interrelación entre los trabajadores y los establecimientos, mediante la relación Encargado. Definiendo a un Encargado como único responsable de un establecimiento



En este caso un trabajador puede ser o no encargado, mientras que un establecimiento ha de tener un encargado.