

UF2 - B1

Sistemes ERP-CRM. Explotació i adequació - I

Annex 09

Disseny d'una many2many amb atributs

Isidre Guixà i Miranda

Institut Milà i Fontanals d'Igualada

Sistemes de gestió empresarial

10 de novembre de 2015
Isidre Guixà i Miranda
Institut Milà i Fontanals
C/. Emili Vallès, 4
08700 - Igualada

Aquest material està protegit sota llicència de [Creative Commons BY-NC-SA 3.0](#) i és la maqueta dels materials en desenvolupament per l'Institut Obert de Catalunya, que una vegada publicats seran accessibles a l'apartat recursos de la web <http://ioc.xtec.cat>

En cas de suggeriment i/o detecció d'error, podeu informar a iguixa@xtec.cat

Castellano

Este material está protegido bajo licencia [Creative Commons BY-NC-SA 3.0](#) y es la maqueta de los materiales en desarrollo por el Institut Obert de Catalunya, que una vez publicados seran accesibles en el apartado recursos de la web <http://ioc.xtec.cat>

En caso de sugerencia y/o detección de error, dirigir-se a iguixa@xtec.cat

Euskara

Material hau [Creative Commons BY-NC-SA 3.0](#) babesa eta eredu materialak garatzen Institut Obert de Catalunya, behin argitaratutako web <http://ioc.xtec.cat> iradokizuna bada eta / edo errore detekzioa, iguixa@xtec.cat zinen joan

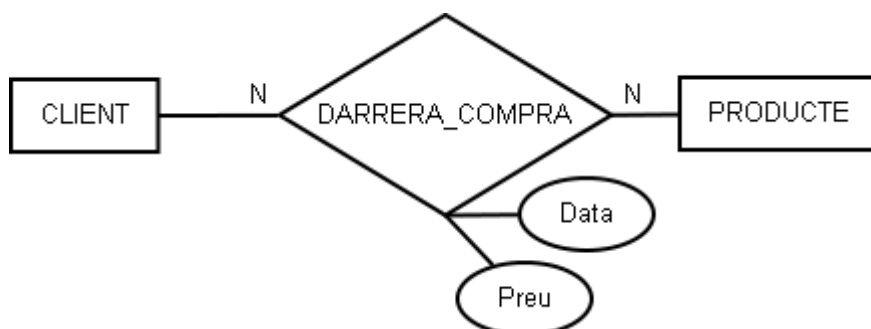
Galego

Este material está protexido baixo [Creative Commons BY-NC-SA 3.0](#) eo modelo é o desenvolvemento de materiais para o Institut Obert de Catalunya, que unha vez publicado será accesible a sección de recursos do <http://ioc.xtec.cat> web.
Se suxestión e / ou detección de erros, indica iguixa@xtec.cat

Com cal dissenyar una relació many2many entre classes quan aquesta relació té atributs? Aquest és un tema que està en l'ambit del DOO (Disseny Orientat a Objectes) i no té res a veure amb OpenERP.

Imaginem-nos que hem de dissenyar una solució informàtica per gestionar la llista dels productes que ens compren els clients, emmagatzemant la data i el preu de la darrera compra de cada producte. No interessa pas un històric de totes les compres que ens ha efectuat.

En cas d'efectuar un disseny entitat-relació per a la BD, el disseny seria similar a:



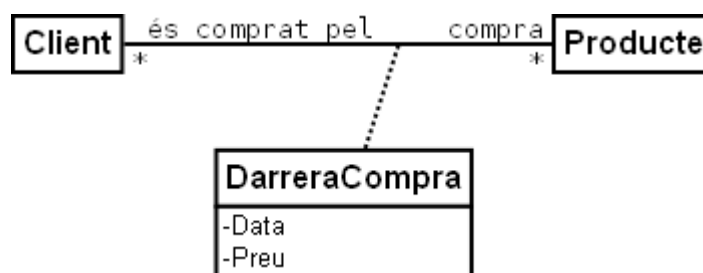
No s'inclouen els atributs de les entitats CLIENT i PRODUCTE i observem que la relació N:N entre les dues entitats té els atributs Data i Preu. La traducció al model relacional seria:

CLIENT (#codi, ...)

```
PRODUCTE (#codi, ...)
```

```
DARRERA_COMPRA (#client, #producte, data, preu)
                FK CLIENT      FK PRODUCTE
```

En cas d'efectuar un disseny UML de classes, el disseny seria similar a:



Fixem-nos que apareix la classe associativa `DarreraCompra`, necessària per poder incorporar els atributs de la relació. Aquesta classe no seria necessària si no calgués gestionar els atributs `Data` i `Preu`.

Una possible implementació de les classes `Client` i `Producte` en Java, en cas que NO calqués gestionar els atributs `Data` i `Preu`, seria:

```
class Client {
    ...
    List<Producte> productes;
}

class Producte {
    ...
    List<Client> clients;
}
```

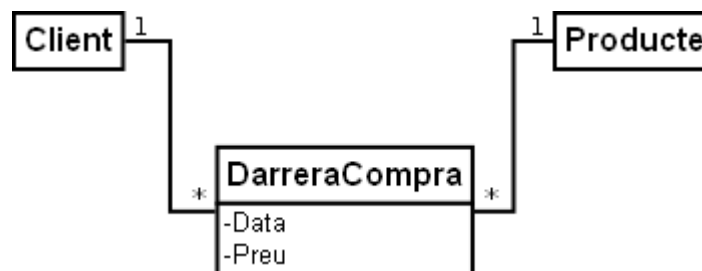
i la persistència d'aquestes classes en una BDR seria:

```
CLIENT (#codi, ...)
PRODUCTE (#codi, ...)
DARRERA_COMPRA (#client, #producte)
                  FK CLIENT      FK PRODUCTE
```

Però... en cas d'haver de gestionar els atributs Data i Preu, apareix la nova classe DarreraCompra i la implementació en Java seria similar a:

```
class Client {
    ...
    List<DarreraCompra> compres;
}
class Producte {
    ...
    List<DarreraCompra> compres;
}
class DarreraCompra {
    Client client;
    Producte producte;
    Date data;
    double preu;
}
```

que també correspondria al disseny UML següent:



La persistència d'aquestes classes en una BDR seria:

```
CLIENT (#codi, ...)
PRODUCTE (#codi, ...)
DARRERA_COMPRA (#client, #producte, data, preu)
                  FK CLIENT      FK PRODUCTE
```

Així doncs, la implementació d'una relació *: * entre classes amb atributs en la relació, necessita d'una classe associativa i la relació *: * es converteix en dues relacions one2many.

I OpenERP no és diferent... Els camps many2many en OpenERP estan pensats per donar resposta a relacions many2many entre classes sense atributs a la relació. En el moment en que la relació té atributs, cal pensar en dues relacions one2many.