

## MODEL RELACIONAL

El model relacional és un model de **representació de dades** basat en dues disciplines matemàtiques: [la lògica de predicats](#) i [la teoria de conjunts](#).

El model relacional va ser proposat originàriament per **Edgar Frank Codd** en el seu treball *A Relational Model of Data for Large Shared Data Banks* l'any 1970.

Aquest és possiblement el model més utilitzat per als SGBD actuals. Els sistemes gestors de bases de dades anomenats relacionals com Oracle, MySQL, PostgreSQL, DB2, SqlServer... l'utilitzen.

El model relacional representa la base de dades com un conjunt de relacions que s'organitzen en taules.

[!Important] En aquest model quan parlem de relacions no estem parlant de les relacions del model Entitat/Relació. NO CONFONDRE-HO!

## CONCEPTES BÀSICS DEL MODEL RELACIONAL

Les relacions poden ser concebudes com una representació tabular de les dades. Una relació representa un conjunt d'entitats.

Exemple de la relació Alumne:

Dni	Nom	Cog1	Cog2	Telèfon	Població	N.Matric
40123564	Pere	Pl	Vidal	972456789	Girona	10345672
35357853	Josep	Lopez	Lopez	972222325	Girona	10432345
77900700	Maria	Puig	Vidal	972265641	Olot	10403475
----	----	----	----	----	----	-----

On:

Anomenem **files**, **tuples** o **registres** a les entitats del model ER. Anomenem **columnes**, **atributs** o **camp**s als atributs del model ER.

## ALTRES CONCEPTES

- **Domini d'un atribut:** conjunt finit de valors d'un atribut.

Exemple: Domini de l'atribut DNI, qualsevol combinació de 8 dígits numèrics + 1 lletra.

- **Esquema d'una relació:** nom que la identifica i conjunt d'atributs que la formen.

Exemple: ALUMNE( DNI, Nom, Cognom, Telefon )

- **Grau d'una relació:** El nombre d'atributs que la formen.

Exemple: GRAU(ALUMNE) = 4

- **Extensió d'una relació:** valor de totes les tuples emmagatzemades en la relació.

Exemple:

40123564	Pere	Pl	Vidal	972456789	Girona	10345672
35357853	Josep	Lopez	Lopez	972222325	Girona	10432345
77900700	Maria	Puig	Vidal	972265641	Olot	10403475

- Cardinalitat d'una relació: nombre de tuples que conté l'extensió de la relació.

Exemple:  $CARDINALITAT(ALUMNE) = 3$

## CARACTERÍSTIQUES DE LES RELACIONS

- L'esquema de la relació no acostuma a canviar, mentre que l'extensió sí.
- Les tuples de la relació no tenen cap ordre establert.
- El valor d'un atribut és un valor simple (no es permeten atributs multivaluats).
- Els esquemes d'una relació corresponen a **Entitats** o **Relacions** del model ER.
- L'esquema de relació tindrà sempre una **clau primària** formada per un o més atributs.

## TIPUS DE CLAU

Per tal de resultar útil, l'emmagatzematge de la informació ha de permetre la identificació de les dades. En l'àmbit de les bases de dades relacionals, les tuples de les relacions s'identifiquen mitjançant les anomenades **superclaus**.

[!Important] Per definició, el model relacional no permet tuples repetides.

**Superclau:** qualsevol subconjunt d'atributs que fan que no hi hagi tuples repetides.

Exemple: Donada la Relació ALUMNE( Codi, NIF, NSS, Nom, Cognoms, Telefon ) Superclaus possibles: {Codi, NIF, NSS, Nom, Cognoms, Telefon}, {Codi, NIF, NSS, Nom, Cognoms}, {Codi, NIF, NSS, Nom }, {Codi, NIF, NSS}, {Codi, NIF}, {Codi }, {NIF}, {NSS, Nom}...

El problema de les superclaus és que podem tenir atributs que no són necessaris per tal d'identificar una tupla de manera única.

**Claus candidates:** totes les superclaus mínimes d'una relació que permet identificar les tuples de manera única.

Exemple:

Claus candidates de la relació ALUMNE(Codi, NSS, NIF, Nom, Cognom ) -> {Codi}, {NSS}, {NIF}

[!Important] **Clau primària:** és la clau candidata escollida pel dissenyador de la base de dades per identificar de manera única una tupla.

**Claus alternatives:** la resta de claus candidates descartades pel dissenyador.

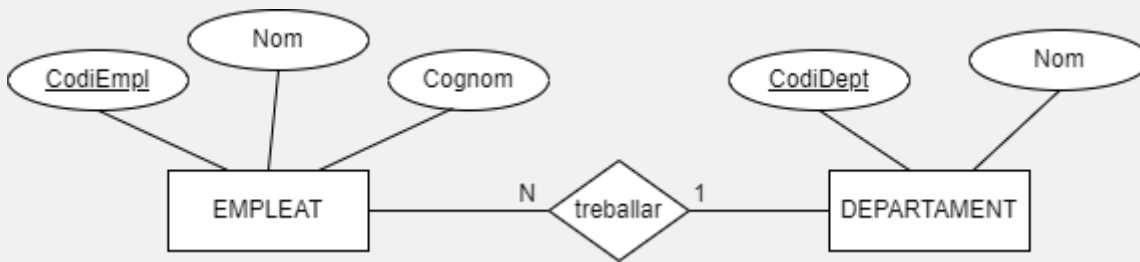
[!Important] **Claus foranes:** Una clau forana està constituïda per un atribut, o per un conjunt d'atributs, de l'esquema d'una relació, que serveix per relacionar les seves tuples amb les tuples d'una altra relació de la base de dades (o amb les tuples d'ella mateixa, en alguns casos).

- Tota clau forana sempre fa referència a una clau primària i ha de tenir el mateix nombre d'atributs que la clau primària a la qual referencia.
- Els dominis dels atributs de la clau forana han de coincidir amb el domini dels atributs de la clau primària referenciada.
- Entre els atributs de l'esquema d'una clau forana i els de la clau primària respectiva s'ha de poder establir una correspondència (concretament, una **bijecció**). Això vol dir que una clau forana sempre farà

referència a una clau primària existent.

Exemple:

Diagrama ER:



Model relacional:

EMPLEAT(CodiEmpl, Nom, Cognom, CodiDept) ON {CodiDept} REFERENCIA DEPARTAMENT  
 DEPARTAMENT (CodiDept, Nom)

Representació tabular

## OPERACIONS AMB LES RELACIONS

Les operacions habituals que realitzarem amb les relacions, i per tant amb les dades de la base de dades són les que identifiquem amb l'acrònim **CRUD** (de l'anglès Create, Read, Update i Delete).

### CONSULTA (READ)

Obtenció parametritzada de dades **sense que es vegin alterades** les emmagatzemades en la base de dades.

### ACTUALITZACIÓ

Les operacions d'actualització realitzen canvis en les tuples que queden reflectits en les relacions de la base de dades.

- **INSERCIÓ (CREATE):** Afegeix una o més tuples a una relació determinada.
- **MODIFICACIÓ (UPDATE):** Modifica el valor d'un o més atributs d'una o més tuples d'una relació determinada.
- **ESBORRAT (DELETE):** Elimina una o més tuples d'una relació determinada.

## REGLES D'INTEGRITAT

Anomenem **integritat** la propietat de les dades que permet representar correctament les situacions del món real que modelitzen.

Per tal que les dades siguin íntegres, cal garantir que siguin correctes, i també que estiguin senceres.

**Integritat de domini:** especifica que el valor de cada atribut A ha de ser un valor simple i del domini (A).

**Unicitat de clau primària:** No hi pot haver dues o més tuples amb la mateixa combinació de valors en els atributs que formen la clau primària d'una relació.

**Entitat de la clau primària:** Cap atribut que formi part de la clau primària pot contenir valors nuls\*.

[!Important]

**Integritat referencial:** la clau forana ha de coincidir amb la clau primària a la que referencia, o bé ser

nul·la.

## Integritat referencial

EL SGBD vetllarà en tot moment per mantenir la integritat referencial, especialment en els casos següents:

- Operacions sobre claus foranes:
  - Insercions de noves tuples -> Verificar la integritat.
  - Actualitzacions sobre atributs que formin part d'alguna clau forana -> Verificar la integritat.
- Operacions sobre claus primàries referenciades:
  - Esborrat de tuples -> Restricció, actualització en cascada o anul·lació.
  - Actualitzacions sobre atributs que formin part d'alguna clau primària referenciada -> Restricció, actualització en cascada o anul·lació.

\* Els SGBD proporcionen mecanismes per indicar que un cert atribut pot contenir un valor especial anomenat NUL. Significa que l'atribut no té un valor específic. Per exemple l'atribut NOTA d'un alumne en un mòdul ha de poder contenir un valor NUL, ja que al principi de curs no sabem quina nota tindrà aquell alumne. El valor NUL en aquest cas significaria: "Aquest alumne encara no té nota assignada."

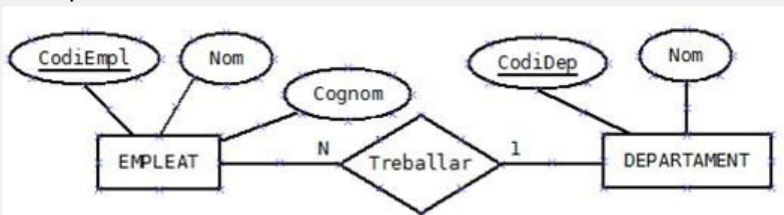
## PAS A MODEL RELACIONAL

A partir de l'**esquema conceptual** que hem obtingut amb el diagrama entitat-relació obtindrem l'**esquema lògic** amb el model relacional. Hem de tenir clar el concepte de clau primària i clau forana. Per transformar un diagrama ER a Model Relacional seguirem les següents regles:

### ENTITATS FORTES

Per cada entitat forta del diagrama ER crearem un nou esquema de relació amb els atributs de cada entitat. Indicarem quin o quins atributs formen la clau primària.

Exemple:



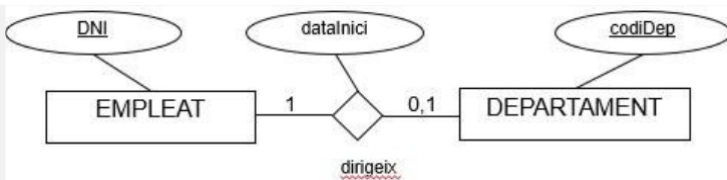
EMPLEAT (CodiEmpl, Nom, Cognom,...

DEPARTAMENT( CodiDep, Nom,...

### RELACIONS 1:1

En les relacions 1:1 escollim un dels esquemes de relació i **afegim com a clau forana la clau primària de l'altre**. Si una de les entitats té participació total escollirem aquesta per afegir-hi l'atribut de clau forana i els atributs de la interrelació si en tenim.

Exemple:



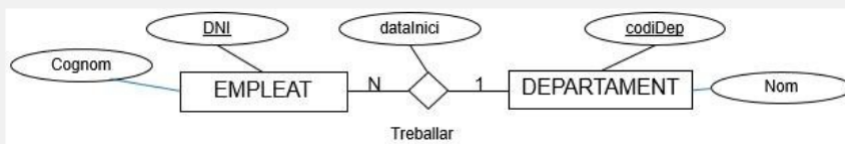
EMPLEAT( DNI, Nom, Cognom, ...)

DEPARTAMENT (codiDep, NomDep, DNIDirector, dataInici ) ON {DNIDirector} REFERENCIA EMPLEAT {DNI} En aquest cas escollim afegir a DEPARTAMENT la clau forana degut a que tot departament sempre té un empleat que el dirigeix. D'aquesta manera evitem tenir valors nulls. Si ho féssim al revés (afegir a la relació EMPLEAT la clau forana) ens trobaríem que molts empleats no dirigeixen cap departament, de manera que aquesta clau forana contindria molts valors nulls. Si la relació fos 1:1 podríem escollir indistintament una de les dues relacions per afegir la clau forana.

## RELACIONS 1:N

En les relacions 1:N afegim una clau forana a l'esquema de relació al costat N que fa referència a la clau primària de l'entitat del costat 1. Afegirem també els possibles atributs de la interrelació com atributs de l'esquema de relació del costat N.

Exemple:



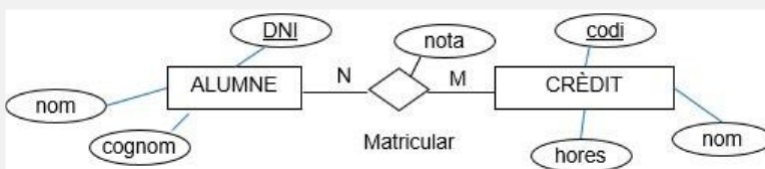
EMPLEAT (DNI, Cognom, codiDep, data Inici) ON {codiDep} REFERENCIA DEPARTAMENT { codiDep }  
DEPARTAMENT (codiDep, Nom)

## RELACIONS N:M

En les relacions N:M, afegim un nou esquema de relació.

La clau primària d'aquest nou esquema serà la combinació de les claus primàries de les entitats que intervenen en la relació. Cadascuna d'elles serà alhora clau forana dels esquemes de relació corresponents. Afegirem també els possibles atributs de la relació com atributs del nou esquema de relació.

Exemple:



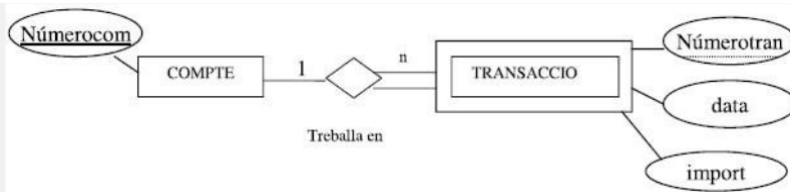
ALUMNE (DNI, nom, cognom) CREDIT

(codi, nom, hores) MATRICULA (DNI, codiCredit, Nota) ON {DNI} REFERENCIA ALUMNE {DNI} I {codiCredit} REFERENCIA CREDIT {codi}

## ENTITATS FEBLES

Per cada entitat feble crearem un esquema de relació que contingui tots els atributs de l'entitat. Afegim com a clau forana la clau primària de l'entitat forta de qui depèn. La clau primària d'aquesta relació es forma amb la combinació de l'entitat forta més l'atribut discriminant de la feble.

Exemple:

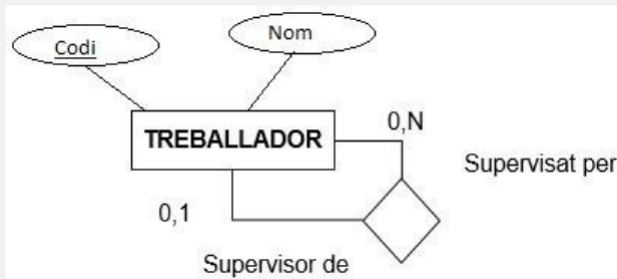


COMPTE (Númerocom) TRANSACCIÓ  
( Númerocom, Númerotran, data, import) ON {Númerocom} REFERENCIA COMPTE {Númerocom}

## RELACIONS REFLEXIVES

Aquest tipus de relacions es tracten igual que si fossin relacions de tipus 1:1, 1:N o N:M, amb la particularitat que només intervé una entitat.

Exemple:



TREBALLADOR (Codi, Nom, CodiSupervisor) ON {  
CodiSupervisor} REFERENCIA TREBALLADOR {Codi}

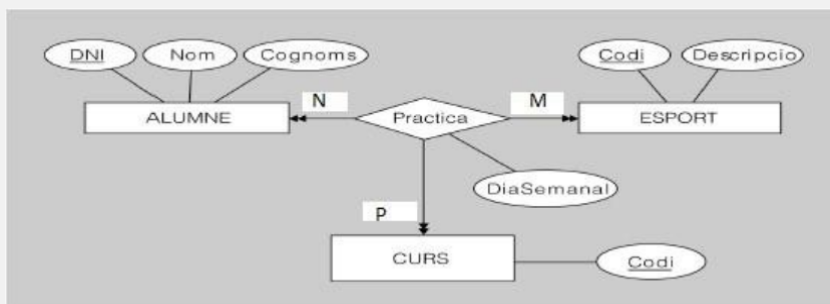
## RELACIONS TERNÀRIES

Tractarem el cas de les relacions ternàries per ser les més comunes, però és extensible a qualsevol relació amb  $n \geq 3$ .

### RELACIONS N:M:P

Es crea una nova interrelació on la clau primària està formada per tots els atributs que formen les claus primàries de les entitats relacionades. En cas que la relació contingui atributs s'afegeixen a la nova relació.

Exemple:



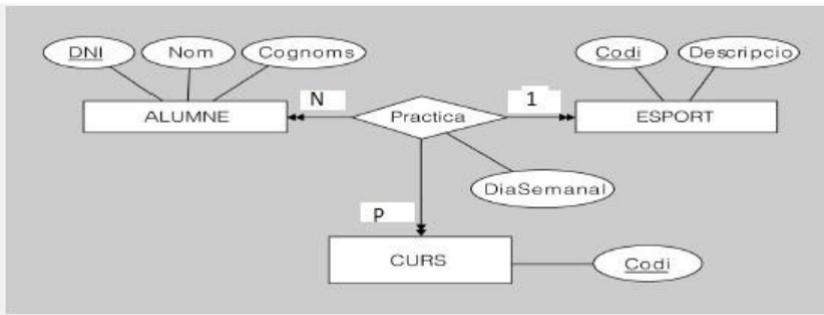
ALUMNE (DNI, Nom, Cognoms)  
ESPORT (Codi, Descripció) CURS (Codi) PRACTICAR ( DNIALumne, CodiCurs, CodiEsport, DiaSemana)  
ON {DNIALumne} REFERENCIA ALUMNE {DNI} I {CodiCurs} REFERENCIA CURS {Codi} I {CodiEsport}  
REFERENCIA ESPORT {Codi}

### RELACIONS 1:N:M

Es crea una nova relació que conté tots els atributs de les entitats relacionades. La clau primària està composta per tots els atributs que formen les claus primàries de les dues entitats que són N-M. En cas que la

relació contingui atributs s'afegeixen a la nova relació.

Exemple:



ALUMNE (DNI, Nom, Cognoms)

ESPORT (Codi, Descripció)

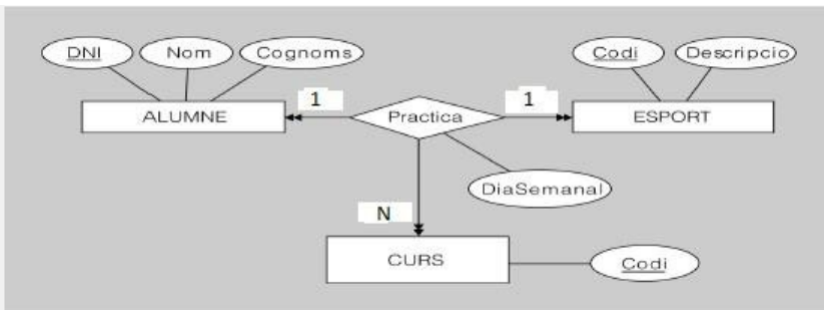
CURS (Codi)

PRACTICAR ( DNIALumne, CodiCurs, CodiEsport, DiaSemana) ON {DNIALumne} REFERENCIA ALUMNE {DNI} I {CodiCurs} REFERENCIA CURS {Codi} I {CodiEsport} REFERENCIA ESPORT {Codi}

## RELACIÓ 1:1:N

Es crea una nova relació que conté tots els atributs de les entitats relacionades. La clau primària està composta per l'atribut que forma la clau primària del costat N. En cas que la relació contingui atributs s'afegeixen a la nova relació.

Exemple:



ALUMNE (DNI, Nom, Cognoms)

ESPORT (Codi, Descripció)

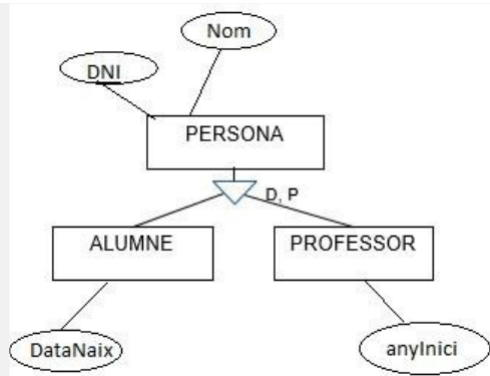
CURS (Codi)

PRACTICAR ( DNIALumne, CodiCurs, CodiEsport, DiaSemana) ON {DNIALumne} REFERENCIA ALUMNE {DNI} I {CodiCurs} REFERENCIA CURS {Codi} I {CodiEsport} REFERENCIA ESPORT {Codi}

## MODEL ER ESTÈS

Per cada tipus d'entitat superclasse de l'esquema ER estès es crea una nova relació que contingui tots els seus atributs. Per cada tipus d'entitat subclasse es crea una nova relació que conté tots els seus atributs específics i s'hi afegeix com a clau forana la clau primària de la superclasse de la que en depèn. Aquesta clau també actuarà com a clau primària.

Exemple:



PERSONA (DNI, Nom)

ALUMNE ( DNI, DataNaix) ON {DNI} REFERENCIA PERSONA {DNI}

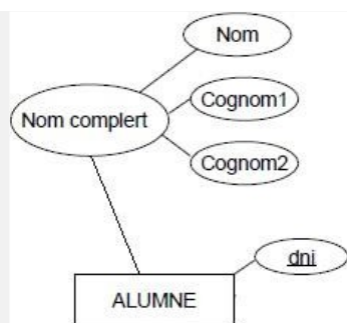
PROFESSOR ( DNI, anyInici) ON {DNI} REFERENCIA PERSONA {DNI}

## ATRIBUTS COMPOSTOS, MULTIVALUATS I DERIVATS

### ATRIBUTS COMPOSTOS

Es desglossen i es posen com a atributs simples en l'esquema relacional.

Exemple:

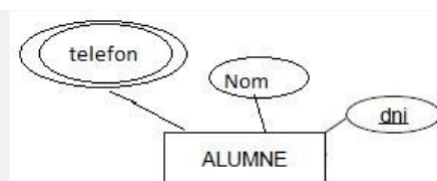


ALUMNE ( DNI, Nom, Cognom1, Cognom2)

### ATRIBUTS MULTIVALUATS

En cas d'utilitzar atributs multivaluats es crea una nova relació que conté com a atributs la clau primària de l'entitat i l'atribut simple. La clau primària estarà formada per ambdós atributs. A la relació forta no es posa l'atribut multivaluat.

Exemple:



ALUMNE (DNI, Nom )

ALUMNE\_TELEFON (DNI, Telefon) ON {DNI} REFERENCIA ALUMNE { DNI }

### ATRIBUTS DERIVATS



Com que el valor dels atributs derivats es pot obtenir a partir d'una altre atribut de la relació en aquest cas els atributs derivats no es representen en l'esquema de la relació, per tant, es queden en el model ER com a informació de les especificacions del model.

## NORMALITZACIÓ

El disseny d'una base de dades pot ser una tasca molt complexa. Hi ha diferents metodologies que permeten abordar el problema de trobar l'esquema relacional que representi millor la realitat que es vol modelitzar.

Coneixem el model ER i el procés de traducció a un model relacional. Si es segueix aquesta metodologia el model final serà el correcte. Però no sempre serà així ja que a vegades hi ha dissenyadors que no voldran perdre el temps en un model ER o perquè s'ha hagut de modificar la base de dades per noves funcionalitats.

Situacions derivades de mals dissenys:

- Repetició de la informació.
- Impossibilitat de representar certa informació.

Situacions derivades de bons dissenys:


- Emmagatzemar tota la informació necessària amb el mínim d'informació redundant.
- Millorar la consultabilitat de les dades.
- Minimitzar els problemes d'actualització.

La teoria de la **normalització** és un mètode que permet assegurar si un disseny relacional (tant si prové de la traducció d'un diagrama ER com si s'ha efectuat directament en el model relacional) és més o menys correcte.

La teoria de la normalització defineix les formes normals com a indicadors per avaluar el grau de normalitat de les relacions, i es diu que una relació està en una forma normal determinada quan satisfà un conjunt determinat de condicions.

## GRAUS DE NORMALITZACIÓ

Hi ha diferents graus de normalitat i, per tant, de formes normals, les quals compleixen la relació d'inclusió, a mesura que augmenta el nivell de la forma normal, la relació ha de complir un conjunt de condicions més restrictiu i, per tant, continua verificant les condicions de les formes normals de nivell inferior.

 Diagrama formes normals

En aquest curs veurem fins a la 3<sup>a</sup> forma normal.

### 1FN - PRIMERA FORMA NORMAL

Una relació està en primera forma normal (1FN) **si tots els atributs són atòmics**.


[!Note] Un atribut és atòmic si els elements del domini son simples e indivisibles.

[!Important]


El procés que s'ha de seguir per assolir una 1FN és afegir tantes files com sigui necessari per a cadascun dels diferents valors del camp o camps que tinguin valors no atòmics.

Exemple:

Relació amb atributs multivalor, no està en 1FN.

 Exemple 1FNa

Relació actualitzada a 1FN

 Exemple 1FNb

## 2FN - SEGONA FORMA NORMAL

Una relació està en segona forma normal (2FN) **si està en 1FN i tot atribut que no pertany a la clau té dependència funcional total de la clau**. És a dir, tots els atributs que no son clau principal han de dependre únicament de la clau principal.

[!Note]

Una dependència funcional  $x \rightarrow y$  és completament funcional si en eliminar els atributs A de X significa que la dependència no es mantinguda, això és que:


$$A \in X, X - \{A\} \not\rightarrow Y$$

[!Important]


El procés que s'ha de seguir per assolir una 2FN és dividir la relació (conservant la informació i les dependències) en tantes relacions com sigui necessari de manera que cada relació verifiqui que els seus atributs no-clau tenen dependència funcional total de la clau.


Exemple:

Relació anterior en 1FN:

 Exemple 1FNb

Relacions obtingudes en el procés de normalitzar a 2FN:  Exemple 2FNa

 Exemple 2FNb

 Exemple 2FNc

## 3FN - TERCERA FORMA NORMAL

Una relació està en tercera forma normal (3FN) **si està en 2FN i cap atribut que no pertany a la clau depèn transitivament de la clau**.


[!Note] Una dependència funcional  $X \rightarrow Y$  en un esquema de relació R és una dependència transitiva si existeix un conjunt d'atributs Z que no és un subconjunt d'alguna clau de R, on es manté  $X \rightarrow Z$  i  $Z \rightarrow Y$ .

[!Important] El procés que s'ha de seguir per assolir una 3FN és dividir la relació (conservant la informació i les dependències) en noves relacions més simples, de manera que cada relació verifiqui que cap dels seus atributs no clau depèn transitivament de la clau.

Exemple:

Analitzem la relació Comanda de l'exemple anterior, està en 2FN però conté el país del proveïdor PaisProv, aquest atribut depèn transitivament de Num a través de NomProv, per tant no està en 3FN. Així doncs per passar a 3FN dividirem la relació creant noves relacions de manera que en cada una d'elles cap atribut no clau depengui transitivament de la clau.

Aquestes serien les relacions obtingudes en 3FN:

 Exemple 3FN

## DESNORMALITZACIÓ

La desnormalització és pot definir com la introducció de redundàncies de forma controlada en una bases de dades, per tal de fer més eficients alguns processos que, altrament, farien que globalment el rendiment del sistema resultés poc òptim.

El que hem vist de disseny de bases de dades és des d'un punt de vista acadèmic. Moltes vegades haurem de "renunciar" a un disseny normalitzat per millorar rendiments o per complir requisits legals.