





## Objectius

El procés de normalització consisteix en optimitzar les taules per eliminar tota redundància i mal disseny possible.

Hem de ser conscients, però, que si la Base de Dades està ben dissenyada des del principi, estarà pràcticament normalitzada, més si tenim en compte que la Forma Normal Boyce-Codd elimina una redundància molt rara de trobar, i que la major part de les ocasions no ens molestarà aquesta miqueta de redundància (abans al contrari, ens suposaria més molèstia tenir una altra taula).

La utilitat real de la normalització serà quan partim d'una Base de Dades directament importada d'un sistema de fitxers (un fitxer → una taula), on ara sí normalment estarà molt mal dissenyat.

Els objectius mínims seran, per tant:

- Entendre el problema de la redundància i el mal disseny d'una Base de Dades.
- Detectar quan una taula no està en Primera Forma Normal.
- Saber passar una taula no normalitzada a Primera Forma Normal.
- Detectar totes les dependències funcionals entre els atributs d'una taula.
- Saber passar una taula en Primera Forma Normal a Tercera Forma Normal (directament).

Els objectius d'ampliació (i que no ens interessin tant) seran:

- Saber posar en Forma Normal Boyce-Codd

# 1. Introducció

---

El procés de normalització s'encarrega de seguir una sèrie de passos o normes. Després d'aplicar-les totes, s'obtenen les dades agrupades en diferents taules, de manera que és l'estructura òptima per a la seua implementació, gestió i explotació des de diferents aplicacions futures.

La normalització es basa en que les dades han de ser independents de les aplicacions que les utilitzen, i per tant no seran optimitzades per a una aplicació determinada sinó que seran optimitzades, en general, per a qualsevol aplicació futura. El seu objectiu és obtenir el **major nombre de taules possibles**. Cadascuna d'aquestes taules estarà formada pels atributs imprescindibles per a representar l'entitat o la relació entre entitats.

## Avantatges que s'obtenen després de la normalització:

- **Facilitat d'ús.** Les dades estan agrupades en taules que identifiquen clarament una entitat o una relació.
- **Flexibilitat.** La informació que necessiten els usuaris es pot obtenir de les taules relacionals per mig de les operacions de l'àlgebra relacional. No es tenen unes estructures rígides de dades sinó que es poden combinar de manera que s'obtinguen múltiples vistes.
- **Facilitat de Implementació.** Les taules resultants són simples.
- **Claredat.** La representació de la informació és clara i senzilla per a l'usuari. Són taules senzilles.
- **Redundància mínima.** La informació no estarà duplicada innecessàriament dins de les estructures.
- **Màxim rendiment de les aplicacions.** Només es tracta aquella informació que serà d'utilitat a cada aplicació.

Dins de tot el procés de creació d'una Bases de Dades, la normalització es faria després de construir tot l'esquema relacional, és a dir, primer es construiria l'esquema amb el Model E/R, després es traduiria al Model Relacional, i posteriorment es normalitzarien aquestes taules de l'esquema relacional. La veritat, però, és que normalment les taules que ens eixiran en l'esquema relacional estaran prou normalitzades, encara que pot ser molt convenient analitzar totes les taules.

**La vertadera utililitat i necessitat de la normalització està en la conversió o actualització de sistemes d'informació antics.** Així, si volem actualitzar una aplicació basada en fitxers tradicionals, la normalització de les taules serà absolutament necessària.

## 2. Primer Forma Normal (1FN)

Una taula està en **1FN** si i només si els valors que componen cada atribut d'una tupla són atòmics. És a dir, en un atribut no han d'aparèixer valors repetitius.

Per exemple, en la següent taula hi ha una sèrie de tipus de materials existents en una ferreteria. Un material té un codi que l'identifica, la seua descripció i les seues mides (la clau principal és el camp subratllat)

### MATERIALS

<u>COD-MAT</u>	DESCRIPCIÓ	MIDES
039	Cargol	3,5 - 5 - 7 - 9
067	Arandella	2 - 5
461	Broca	2,5 - 3 - 3,5

Els problemes que planteja són els següents:

- La falta de espai en el camp per als valors que puguin aparèixer o, pel contrari el desaprofitament de l'atribut quan existeixen pocs valors.
- La dificultat del tractament per a actualitzacions, consultes i recerques d'un valor determinat.

### Posar en 1FN

Per a passar a 1FN una taula que no ho estava **es descompon** en **dues** distintes:

**A)** La **primera taula** serà la projecció de la taula original sobre els següents atributs:

- La clau de la taula original.
- Els atributs atòmics (els que contenen valors únics).

### MATERIALS

<u>COD-MAT</u>	DESCRIPCIÓ
039	Cargol
067	Arandella
461	Broca

**B) La segona taula** serà la projecció de la taula original sobre els següents atributs:

- La clau de la taula original.
- Els atributs que tenen valors múltiples, distribuint aquests valors múltiples en tuples distintes i per tant en una fila existirà un únic valor elemental.

La clau d'aquesta segona taula estarà formada per tots els atributs.

### MAT-MIDES

<u>COD-MAT</u>	<u>MIDA</u>
039	3,5
039	5
039	7
039	9
067	2
067	5
461	2,5
461	3
461	3,5

### Nota

També haurem de ser capaços de detectar que no està en 1FN quan tenim uns atributs multivaluats "encoberts". Per exemple, una variant de l'exemple anterior podria ser:

### MATERIALS

<u>COD-</u>	DESCRIPCIÓ	MIDA1	MIDA2	MIDA3	MIDA4

<u><b>MAT</b></u>					
039	Cargol	3,5	5	7	9
067	Arandella	2	5		
461	Broca	2,5	3	3,5	

Es veu que es tracta d'una manera de "dissimular" els atributs multivaluats. Estem davant del mateix cas que en l'exemple de dalt i tindrem els mateixos problemes, i per tant la solució és la mateixa.



## 3. Dependència Funcional

---

### 3.1 DEPENDÈNCIA FUNCIONAL

Un atribut o conjunt d'atributs **Y** depèn funcionalment de l'atribut o conjunt de atributs **X**, i es representa com  $X \rightarrow Y$ , si i només si cada valor de X es correspon amb un únic valor de Y.

Per exemple, entre DNI i NOM existeix una dependència funcional, ja que el valor DNI es correspon amb un únic nom.

**DNI  $\rightarrow$  NOM**

En aquest cas (suposant que el NOM és únic i no existeixen dos noms iguals) s'acompleix també la dependència

**NOM  $\rightarrow$  DNI**

i ho podríem abreviar per

**DNI  $\longleftrightarrow$  NOM**

No sempre es dona de forma biunívoca la dependència funcional entre dos atributs, és més, en pocs casos succeeix. Per exemple, entre els atributs DNI i ADREÇA existeix una dependència funcional ja que una persona identificada pel seu DNI viu en una única ADREÇA.

**DNI  $\rightarrow$  ADREÇA**

Però en aquest cas no es dona la dependència en sentit invers, perquè en una ADREÇA viuen unes quantes persones. A més, l'adreça no ens diu ni tan sols la ciutat on es troba (per exemple, el "C/ Major, 7" es troba en moltes ciutats)

Hi ha atributs que no tenen entre ells una dependència funcional com és el cas de ADREÇA i DATA\_NAIXEMENT.

En moltes ocasions, per a determinar un únic valor d'un atribut, no és suficient amb conèixer el valor d'un altre atribut, sinó que és necessari trobar els valors d'uns quants atributs. Això és el que passa, si tenim els atributs: DNI, EMPRESA i SOU, i sabem que una persona pot treballar en més d'una empresa. Entre els atributs DNI i SOU no existeix cap dependència funcional, ja que un individu pot guanyar sous distints en empreses distintes. Però si coneixem l'empresa en la qual treballa, sí que podem dir que:

$$\text{DNI} . \text{EMPRESA} \rightarrow \text{SOU}$$

L'operador punt "." representa l'expressió "junt amb" o "i" entre els dos atributs i l'operador barra "|" fa referència a l'expressió "o també".

Per tant, podem dir que DNI "junt amb" EMPRESA determinen el SOU. I per a la dependència:

$$\text{DNI} \rightarrow \text{NOM} \mid \text{ADREÇA}$$

direm que amb el DNI es coneix el NOM "o també" l'ADREÇA.

## 3.2 DEPENDÈNCIA FUNCIONAL TOTAL

Es diu que l'atribut **Y** té una **dependència funcional total** de l'atribut **X** si té una dependència funcional de X i NO depèn funcionalment de cap subconjunt de X.

Per exemple, una dependència funcional seria:

$$\text{DNI} . \text{EMPRESA} \rightarrow \text{NOM}$$

Però, lògicament, aquesta dependència no és total ja que NOM depèn funcionalment de DNI. Per això, a aquesta dependència s'anomena **dependència parcial**.

La dependència funcional total seria:

$$\text{DNI} . \text{EMPRESA} \rightarrow \text{SOU}$$

Ara sí, el SOU no depèn funcionalment de cap subconjunt.

És evident que si X està format únicament per un atribut, la dependència funcional serà total.

Les dependències que ens interessen per a la normalització són sempre les dependències funcionals totals.

### 3.3 GRAF DE LES DEPENDÈNCIES FUNCIONALS

Serveix per a mostrar gràficament la relació existent entre tots els atributs i és una forma clara de tenir una visió general de les dades i de la cohesió existent entre elles. La clau principal es representa dins d'una caixa de línies contínues amb els seus atributs primaris. La resta dels atributs es representa fora de la caixa.

Si existeixen dependències d'un conjunt d'atributs que no són la clau, aniran dins d'una caixa de línies discontinües per a no confondre-ho amb la clau principal de la taula.

Exemple:

**$A . B . C \rightarrow M \mid S$**

**$M \rightarrow N$**

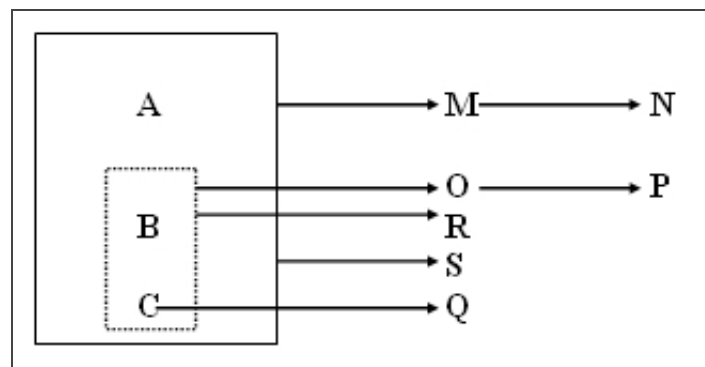
**$B . C \rightarrow O \mid R$**

**$O \rightarrow P$**

**$C \rightarrow Q$**

**(clau =  $A . B . C$ )**

El graf associat és:





## 4. Segona Forma Normal (2FN)

Una taula es diu que està en 2FN si i només si compleix dues condicions:

- Es troba en 1FN.
- Tot atribut secundari (aquells que no pertanyen a la clau principal, els que es troben fora de la caixa) depèn totalment (té una dependència funcional total) de la clau completa i, per tant, no d'una part d'ella.

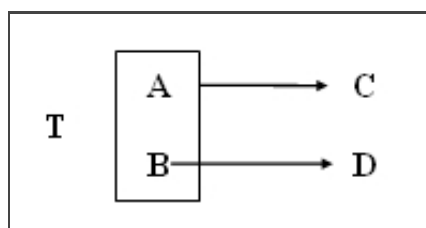
Aquesta forma normal només es considera si la clau principal és composta, i per tant, està formada per diversos atributs.

Si una taula **T** té com a atributs **A, B, C, D** i la clau és **A . B** complint-se les dependències:

**A . B  $\rightarrow$  C**

**B  $\rightarrow$  D**

S'observa que la taula no es troba en 2FN ja que l'atribut D no té una dependència funcional total amb la clau completa A . B , sinó amb una part de la clau (B). El graf de les dependències funcionals seria:

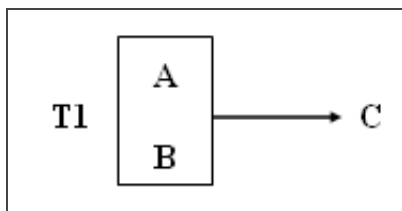


Si existeix una fletxa que ix de l'interior de la caixa que engloba la clau, aleshores la taula no està en 2FN.

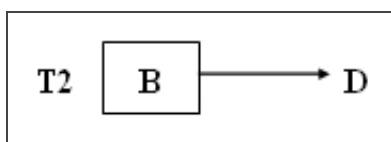
## Posar en 2FN

Per a convertir una taula que no està en segona forma normal a 2FN, es realitza una projecció i es crea:

**A) Una primera taula** amb la clau i totes les seues dependències totals amb els atributs secundaris afectats:



**B) Una segona taula** amb la part de la clau que té dependències, i els atributs secundaris implicats:



La clau de la nova taula T2 serà l'antiga part de la clau.

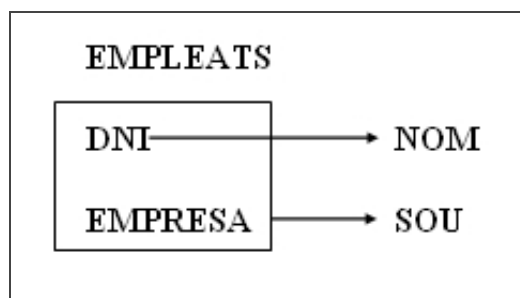
**Exemple:** Taula amb les persones que treballen en diverses empreses amb el sou corresponent, amb els atributs: **DNI** , **NOM** , **EMPRESA** , **SOU**

Entre els atributs existeixen les dependències:

**DNI → NOM**

**DNI . EMPRESA → SOU**

El graf que mostra les dependències és el següent:



És evident que la taula no es troba en 2FN, després de normalitzar s'obté:

## EMPLEATS

DNI

EMPRESA



SOU

## PERSONES

DNI



NOM

## 5. Dependència Funcional Transitiva

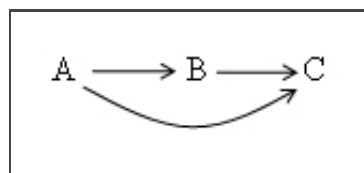
La dependència funcional transitiva s'aplica per a analitzar les taules en tercera forma normal (3FN). Consisteix bàsicament a considerar que **un atribut no primari només ha de conèixer-se a través de la clau principal o claus secundàries**. En un altre cas, estarà produint redundància d'informació amb les anomalies típiques que porta amb ella.

Suposem tres subconjunts distints d'atributs A , B i C que pertanyen a una taula T, de manera que es compleixen les condicions:

**$A \rightarrow B$  i  $B \not\rightarrow A$**

Es diu que C té una **dependència funcional transitiva** amb A o que és transitivament dependent de A si es compleix que  **$B \rightarrow C$**

Gràficament es pot mostrar:



Per tant, un atribut C és transitivament dependent d'un altre A si es coneix per diferents vies, una directament, i una altra a partir d'un altre atribut intermedi B.

Per exemple, considerem tres atributs que formen part de la taula ALUMNES:

NUMMAT = núm. de matrícula.

GRUP = Grup assignat.

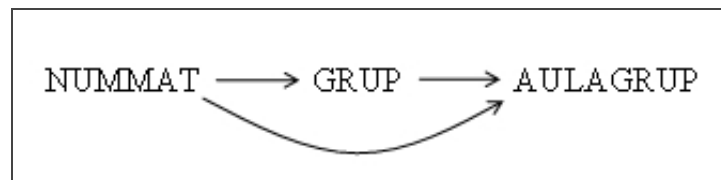
AULAGRUP = Aula assignada al grup.



**NUMMAT → GRUP | AULAGRUP**

**GRUP → AULAGRUP**

L'atribut AULAGRUP és transitivament dependent de NUMMAT, ja que es pot conèixer per mig de l'atribut NUMMAT i a través de l'atribut GRUP



## 6. Tercera Forma Normal (3FN)

Una taula es diu que està en 3FN si i només si es compleixen dues condicions:

- Es troba en 2FN.
- No existeixen atributs no primaris (atributs que no formen part de la clau principal) que són transitivament dependents de cada clau candidata de la taula.

Açò vol dir que un atribut secundari només es pot conèixer a través de la clau principal o claus candidates de la taula i no per mig d'un altre atribut no primari.

En el graf de dependències només han de mostrar-se les dependències transitives i no aquelles dependències funcionals a partir de les claus candidates, perquè se sap que per ser claus ja coneixen tots els atributs.

Exemple: A és la clau principal, B és una clau candidata i es donen les següents dependències:

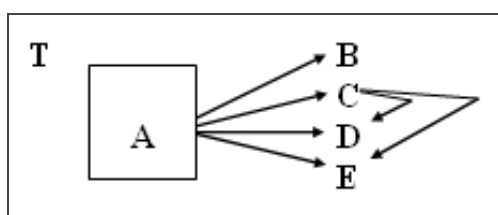
$A \rightarrow B$   $B \rightarrow A$   $C \rightarrow D$

$A \rightarrow C$   $B \rightarrow C$   $C \rightarrow E$

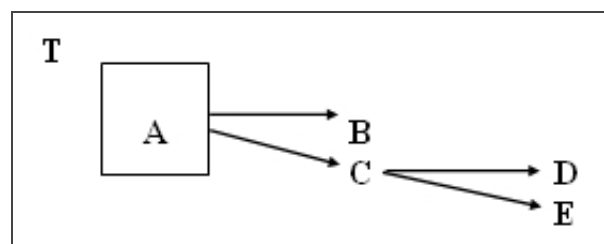
$A \rightarrow D$   $B \rightarrow D$

$A \rightarrow E$   $B \rightarrow E$

El graf queda del següent mode:



O bé



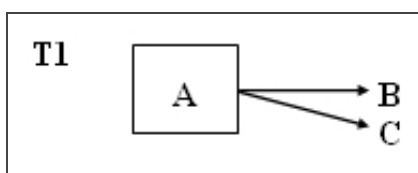
Les fletxes que mostren les dependències funcionals que té la clau candidata B no es representen (com hem dit anteriorment) perquè són evidents i no simplifiquen la visió del graf. A més a més, per a la normalització, no es necessiten per a res; pel contrari, solen complicar l'anàlisi.

La taula T no està en 3FN ja que els atributs D i E són transitivament dependents respecte de la clau A.

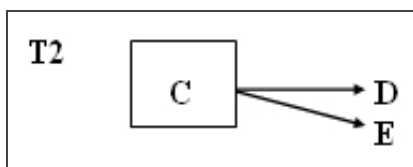
## Posar en 3FN

Per a normalitzar una taula que no estiga en tercera forma normal, és a dir, que tinga dependències transitives, descompondrem la taula en més d'una taula:

**A) Una primera taula** amb la clau principal més els atributs que no depenen transitivament

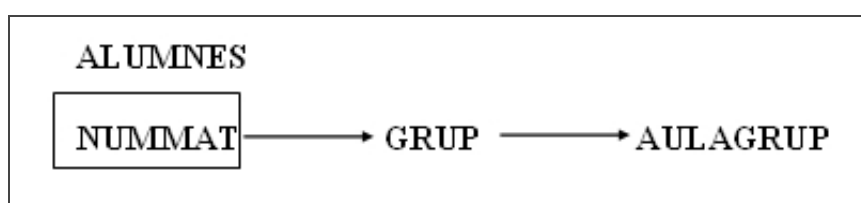


**B) Una segona taula** amb els atributs que depenen transitivament, més l'atribut de qui depenen, que serà clau principal

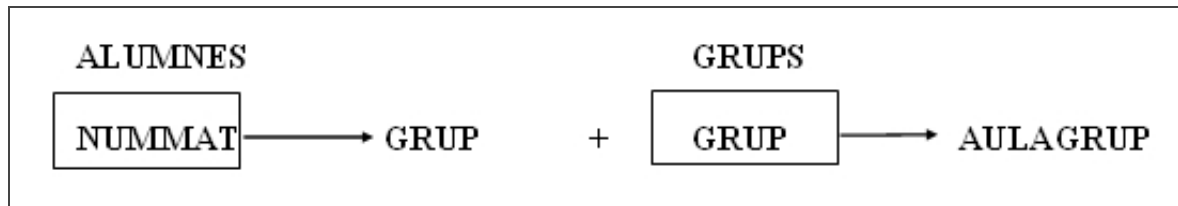


Es farà una descomposició per cada dependència transitiva que hi haja que afecte a camps distints.

Per a l'exemple dels atributs NUMMAT, GRUP i AULAGRUP tenim el següent graf:



Una vegada descomposta la taula en dos segons l'algoritme anterior, tindrem dues taules

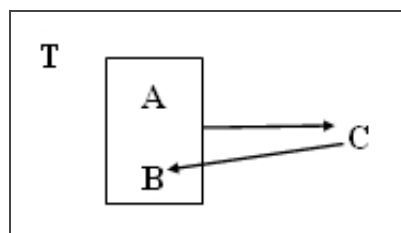


Les dues taules resultants sí que es troben en 3FN.

## 7. Forma Normal Boyce-Codd (FNBC)

Després de la creació de la 3FN es va observar, posteriorment, que podien haver anomalies que no eren abordades. De tota manera són unes redundàncies ja molt estranyes, i que de vegades no valdrà la pena considerar-les.

Són casos de taules que encara que estan en 3FN, mantenen una dependència d'un atribut secundari amb part de la clau. És l'únic cas de dependència transitiva que se'ns podia haver escapat. Gràficament és el següent cas:



**Una taula T està en FNBC si i només si està en 1FN i les úniques dependències funcionals elementals són aquelles en les quals la clau principal (i claus candidates) determinen un atribut.**

La definició engloba la 3FN ja que les dependències transitives existeixen per mig d'atributs secundaris que no eren clau.

Si la clau està formada per un únic atribut i ja estava en 3FN, la taula està en FNBC (com succeïa amb la 2FN).

**Exemple:** Taula d'una guia de carrers

### GUIA DE CARRERS

<u>ADREÇA</u>	<u>CIUTAT</u>	<u>CODPOST</u>
C/ Pez, 2	Benicarló	12580
C/ Luz, 5	Benicarló	12580
C/ Mar, 4	Castelló	12005

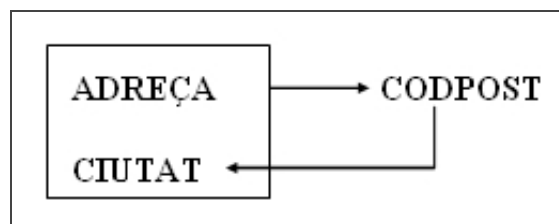
C/ Sol, 4	Vinaròs	12500
C/ Sal, 9	Castelló	12004
C/ Mar, 4	Vinaròs	12500

Les dependències funcionals que ens trobem són:

**ADREÇA . CIUTAT → CODPOST**

**CODPOST → CIUTAT**

Gràficament:

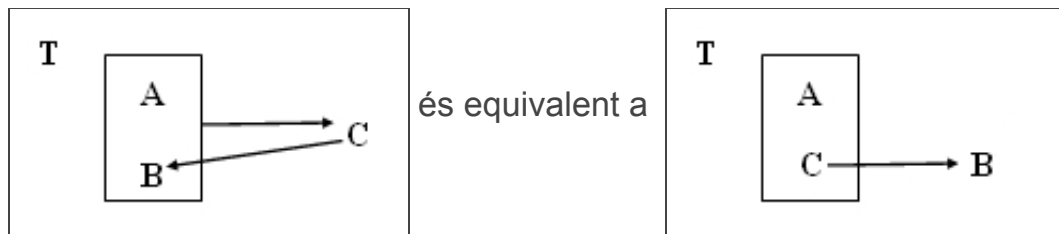


Si observem atentament les tuples d'una taula com aquesta, veurem que per a un mateix codi postal existeixen multitud de tuples que es corresponen amb la mateixa ciutat (tantes com adreces hi haja diferents), per tant existeix informació duplicada.

Si la informació, una vegada que es dona d'alta no varia, és més rendible que la dependència funcional **CODPOST → CIUTAT** es trobe en una altra taula i que existisca una sola tupla per a cada codi postal.

A més a més, què succeeix si s'elimina la tupla amb adreça "C/ Sol, 4" de "Vinaròs" i la tupla "C/ Mar, 4" de "Vinaròs"? El que ocorre és que desapareix la relació entre el codi postal "12500" i "Vinaròs" i potser aquestes dades haurien de mantenir-se.

Si analitzem amb més detall la taula, veuríem que en realitat es pot substituir la clau principal per **A + C** (en l'exemple **ADREÇA + CODPOSTAL**), ja que si A + B ja era clau principal, com per cada valor de C només podem tenir un de B, la combinació A + C també podrà identificar unívocament cada ocurrència de la taula. Per tant, si substituïrem la clau principal, ja no tindríem dubtes de com normalitzar la taula, que serà justament com veurem a continuació:



## Posar en FNBC

L'algoritme de descomposició que s'aplica a una taula que no està en FNBC és el següent:

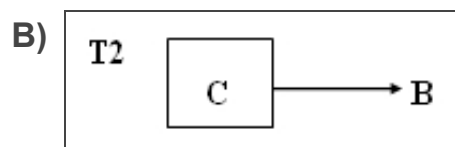
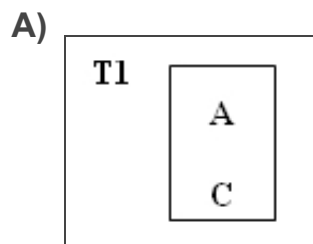
Si tenim una dependència funcional  $C \rightarrow B$  on **C** i **B** són disjunts, **C** és un atribut no primari, i **B** forma part de la clau.

S'obtenen les projeccions:

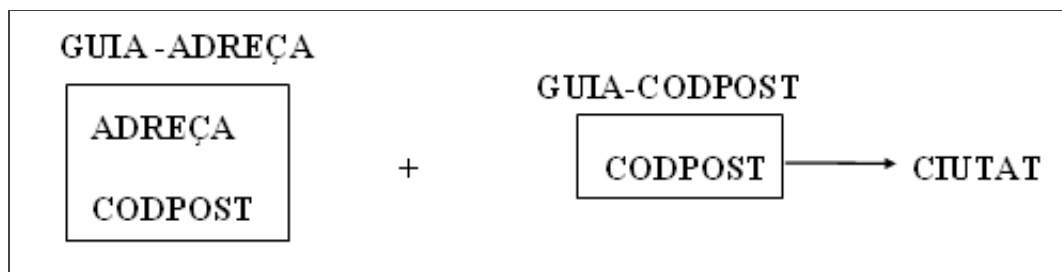
**A)** Una **primera taula T1** amb tots els atributs, excepte **B** (el que formava part de la clau principal); ara formarà part de la clau principal **C**.

**B)** Una **segona taula T2** amb els atributs **C** i **B**, i serà la clau principal **C**

En l'exemple inicial d'aquesta pregunta quedarà:



En l'exemple de la GUIA DE CARRERS:



I quedarien amb la següent informació:

**GUIA-ADREÇA**

<u>ADREÇA</u>	<u>CODPOST</u>
C/ Pez, 2	12580
C/ Luz, 5	12580
C/ Mar, 4	12005
C/ Sol, 4	12500
C/ Sal, 9	12004
C/ Mar, 4	12500

**GUIA-CODPOST**

<u>CODPOST</u>	<u>CIUTAT</u>
12004	Castelló
12005	Castelló
12500	Vinaròs
12580	Benicarló

Per últim, observem les taules que ens queden. Voldrem tenir una taula de codis postals? Si el disseny és per a Correus o Telefònica, o una empresa gran que tinga molts client i els vol tenir distribuïts per codis postals, doncs segur que sí.

Però si es tracta d'una empresa no massa gran, i que tampoc interessa massa la distribució per codis postals, segurament mantenir una taula de codis postals pot semblar fins i tot ridícul. Aleshores, mantenir la taula en 3FN i assumir la miqueta de redundància que suposa no tenir-la en FNBC, pot ser fins i tot saludable. Per això s'ha comentat des del principi del tema la importància de normalitzar fins la 3FN, i la FNBC té una importància relativa.



<b>1FN</b>
------------

**Una taula està en 1FN si i només si els valors que componen els atributs d'una tupla són atòmics.**

Es descompon la taula en dos

1a.- Projectió de la clau junt amb els atributs que tenen valors atòmics.

2a.- Nova clau amb els atributs que tenen valors múltiples (cal idear una nova clau).

<b>2FN</b>
------------

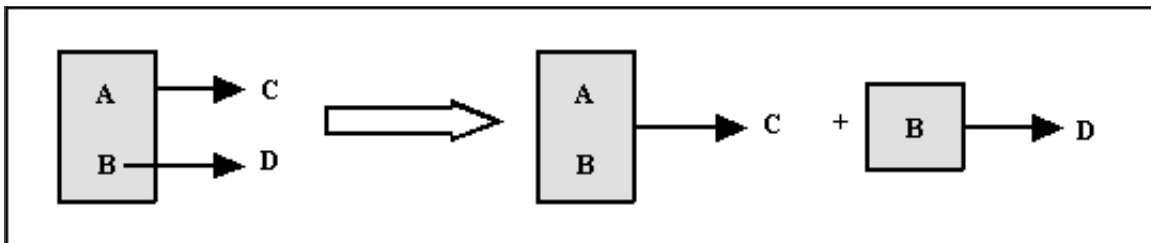
**Una taula està en 2FN si i només si compleix dues condicions:**

- **Està en 1FN.**
- **Tot atribut secundari (els que no pertanyen a la clau principal) depèn totalment (té una dependència funcional total) de la clau completa, i per tant, no d'una part d'ella.**

Es descompon la taula en dos

1a.- Una taula amb la clau i totes les seues dependències totals.

2a.- Una altra taula amb la part de la clau que té dependències, i els atributs secundaris implicats.



### 3FN

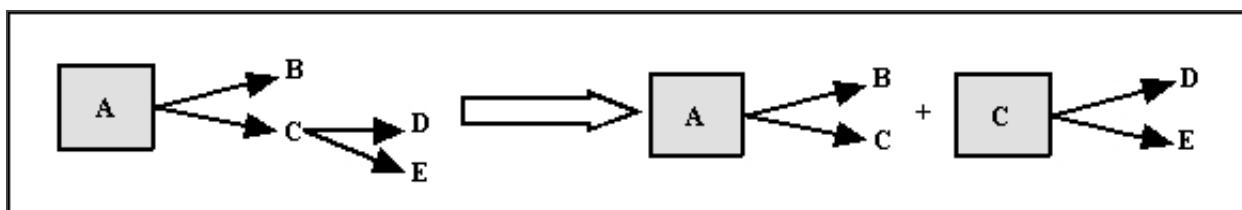
Una taula es troba en 3FN si i només si es compleixen dues condicions:

- Està en 2FN.
- No existeixen atributs no primaris (que no formen part de la clau) que són transitivament dependents d'una clau candidata (cada possible clau de la taula).

Es descompon la taula en dos

1a.- Una taula amb la clau i tots els atributs no primaris que no són transitius.

2a.- Una altra taula amb els atributs transitius i l'atribut no primari (que serà la clau de la nova taula).



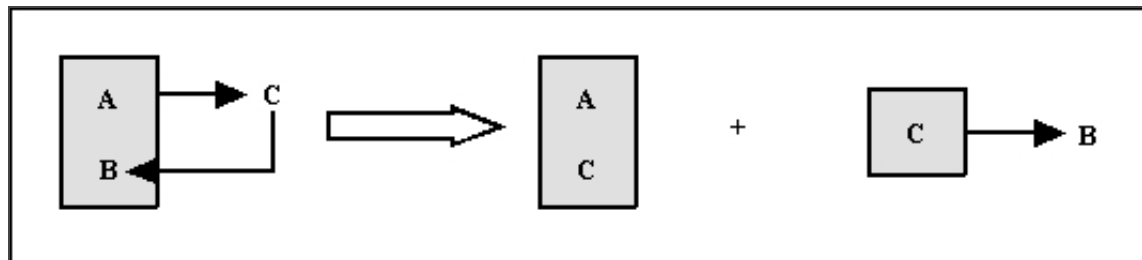
### FNBC

Una taula està en FNBC si i només si està en 1FN i les úniques dependències funcionals són aquelles en què la clau principal (i claus candidates) determinen un atribut.

Es descompon la taula en dos

1a.- Una taula amb tots els atributs menys la part de la clau dependent de l'atribut secundari. La clau està formada per la resta de la clau i l'atribut secundari del que depenia part de la clau.

2a.- Una altra taula en què l'atribut de què depèn part de la clau serà la nova clau i aquesta part de la clau com a atribut secundari.



Llicenciat sota la [Llicència Creative Commons Reconeixement NoComercial  
SenseObraDerivada 3.0](#)