

M02. Gestió de Bases de Dades

UF1. Introducció a les Bases de Dades

Autor: Robert Ventura Vall-Ilovera

Data creació: Setembre 2015

Actualització: Octubre 2018

Aquest document està creat en base el document original de Marc Nicolau Reixach i Úrsula Heredia Garcia

No està permesa la reproducció total o parcial d'aquest document, ni el seu tractament informàtic, ni la transmissió de cap forma o per qualsevol mitjà, ja sigui electrònic, mecànic, per fotocòpia, per registre o altres mètodes, sense el permís previ i per escrit del titular



Normalització



Contingut

Contingut ii

A 1. Normalització del model relacional.....	3
1.1. Introducció.....	3
1.2. Dependència funcional	5
1.2.1. Introducció i definició.....	5
1.2.2. Propietats de les dependències funcionals.....	8
1.3. Formes normals de Codd i de Boyce-Codd.....	9
1.3.1. Primera forma normal (1FN)	10
1.3.2. Segona forma normal (2FN)	12
Tercera forma normal (3FN).....	16
1.3.3. Forma normal de Boyce/Codd (BCFN)	18

A 1. Normalització del model relacional

1.1. Introducció

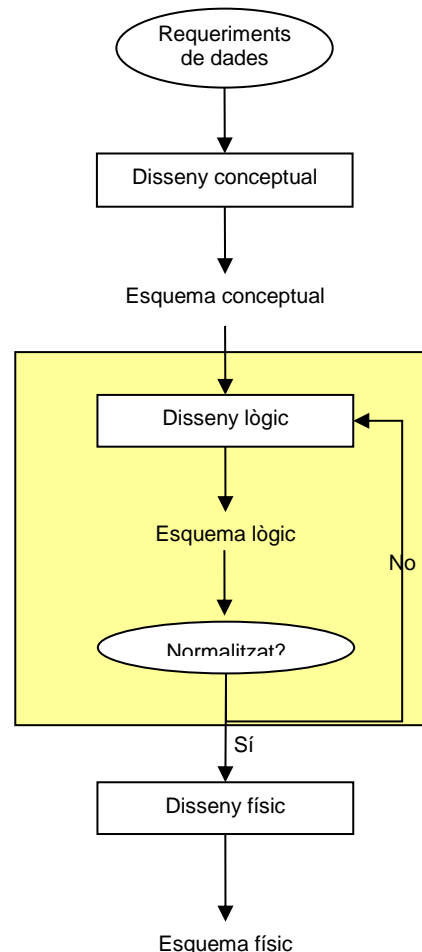
La normalització és una tècnica per dissenyar l'estructura lògica de les dades d'un sistema d'informació en el model relacional.

Va ser desenvolupada per E.F. (Edgar Frank) **Codd** al 1972.

Normalment no utilitzarem la normalització com una tècnica de disseny perquè sempre partirem d'un esquema relacional obtingut a través d'un ER o bé d'una BD existent. Normalment l'utilitzarem per normalitzar o verificar “**la qualitat**” del model relacional existent.

Tal com hem dit utilitzarem la normalització per determinar si el nostre esquema compleix certs criteris / formes normals. Comprovarem si el nostre esquema compleix certes restriccions que imposa cada forma normal a l'esquema. Realitzarem aquest procés de forma iterativa fins que el nostre esquema compleixi cert nivell de forma normal.

La tècnica consisteix en la descomposició sense pèrdua d'informació ni semàntica de la relació universal (o conjunt de taules) en noves taules en què les anomalies d'actualització (inserció, esborrat i modificació) no existeixin o siguin les mínimes.



Les raons per normalitzar una base de dades són:

- Un esquema normalitzat organitza les dades d'acord amb les seves dependències funcionals, és a dir d'acord amb les seves relacions lògiques i en **facilita la seva comprensió**.
- Un esquema normalitzat és robust i **no té redundàncies**. Les redundàncies són el mal més comú en les BD i poden provocar fallades d'integritat quan actualitzem les nostres dades.
- La normalització ens proporciona base de dades amb **esquemes flexibles** que poden estendre's amb facilitat.

Exemple: Esquema d'una BD no normalitzada que conté les dades dels professors i dels departaments d'una Universitat:

```
PROFESSOR (codi_prof, nom, data_naixement, sexe, codi_dep, nom_departament)
```

codi_professor	nom	data_naixement	Sexe	codi_departament	nom_departament
111	Pere	05/08/1973	M	D1	Informàtica
222	Joan	17/10/1974	M	D1	Informàtica
111	Pere	05/08/1973	M	D2	Matemàtiques
333	Marta	08/12/1976	F	D2	Matemàtiques
444	Maria	20/05/1980	F	D1	Informàtica
444	Maria	20/05/1980	F	D3	Investigació
555	Bernat	05/08/1973	M	D1	Informàtica

- En l'esquema anterior si volem afegir un nou departament com ho fem?
- Hi haurà molta informació redundant, per exemple el nom del departament, i el nom del professor.
- Què passa si volem modificar el nom d'un departament?

El **procés de normalització** es basa en la comprovació de totes les taules de l'esquema i si compleixen una sèrie de regles que es basen en la clau primària i les dependències funcionals.

Cada regla que es compleix augmenta el grau de normalització. Si hi ha una regla que no es compleix, la taula es pot descompondre en varies taules que si que les compleixin. La descomposició s'ha de realitzar sense pèrdua d'informació.

1.2. Dependència funcional

1.2.1. Introducció i definició

Un dels conceptes fonamentals de la normalització és la dependència funcional.

Una **dependència funcional** és una relació entre atributs de la mateixa taula. Són restriccions d'integritat que permeten conèixer quines interrelacions hi ha entre dos o més atributs.

Informalment podem dir que si X i Y són atributs de la relació/taula R , diem que Y és **funcionalment dependent** de X (notació: $X \rightarrow Y$) si cada valor de X té associat un únic valor de Y . O sigui, sempre que trobem el valor de X tenim associat el mateix valor de Y .

X o Y poden constar de varis atributs.

A X l'anomenem **determinant**, ja que determina el valor de Y .

L'atribut és **completament dependent** de X si depèn funcionalment de X i no depèn de cap subconjunt de X .

La dependència funcional és una **noció semàntica**. Si hi ha dependències funcionals entre atributs, no ho determinen unes regles abstractes, sinó els models mentals de l'usuari i les **regles de negoci de l'organització** o empresa (món real).

Dependència funcional transitiva.

La dependència funcional $X \rightarrow Z$ és transitiva si existeixen les dependències funcionals $X \rightarrow Y$, $Y \rightarrow Z$, sent X, Y, Z atributs o conjunt d'atributs d'una mateixa relació (taula).

$$\left. \begin{array}{l} X \rightarrow Y \\ Y \rightarrow Z \end{array} \right| X \rightarrow Z$$

Exemple d'una taula PROFESSORS_DEPARTAMENTS:

codi_professor	nom	data_naixement	Sexe	codi_departament	nom_departament
111	Pere	05/08/1973	M	D1	Informàtica
222	Joan	17/10/1974	M	D1	Informàtica
111	Pere	05/08/1973	M	D2	Matemàtiques
333	Marta	08/12/1976	F	D2	Matemàtiques
444	Maria	20/05/1980	F	D1	Informàtica
444	Maria	20/05/1980	F	D3	Investigació
555	Bernat	05/08/1973	M	D1	Informàtica

- **Quines característiques destaquem de les dades dels professors?**

Hi ha professors que apareixen un sol cop, n'hi ha que apareixen més d'un cop.

Apareixen més d'un cop perquè pertanyen a més d'un departament. Per tant tenim informació redundant pel simple fet d'estar relacionat en més d'un departament.

Exemple: *Pere(111) pertany als departaments d'Informàtica(D1) i Matemàtiques(D2)*

- **Es repeteixen dades dels professors?**

Tal i com hem dit anteriorment tenim dades dels professors que es repeteixen. Què passa si volem modificar el nom d'un professor que apareix més d'una vegada? Per exemple, si volem modificar el nom del professor Pere amb codi 111 haurem de modificar totes les tuples/files que continguin el codi de professor 111 i modificar l'atribut Pere.

A més, les modificacions s'hauran de fer de forma transaccional. És a dir, quan jo modifiqui el nom del professor Pere, ningú més pot modificar les altres files, perquè sinó inclouria inconsistència a les dades.

- **Què passa amb els atributs nom, data_naxiement, sexe respecte el codi_professor?**

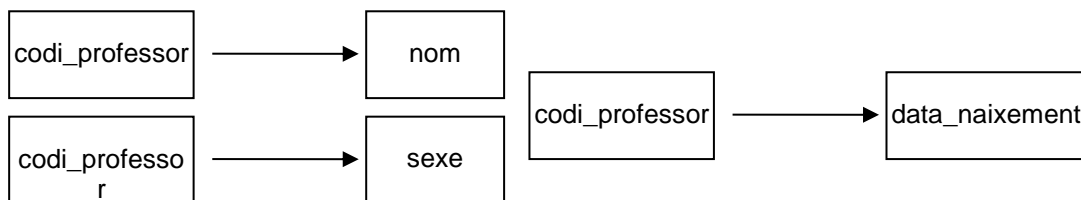
El que podem veure és que aquests atributs tenen el mateix valor pel mateix valor de codi_professor. O sigui, quan tenim el valor d'un codi_professor esperem tenir els mateixos valors en els atributs nom, data_naixement i sexe.

- **Respecte les dades dels departaments?**

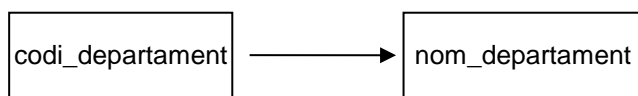
Hi ha departaments que apareixen més d'una vegada.

Al mateix amb els cas dels departaments tenim el mateix nom de departament quan tenim el mateix codi.

Qui determina el sexe d'un professor? **El codi de professor**



Qui determina el nom del departament? **El codi de departament**



Existeix una dependència funcional entre nom i sexe? **No**

Existeix una dependència funcional entre nom_departament i codi_professor? **No**

Exercici:

Troba les dependències funcionals d'aquesta taula VEHICLE:

VEHICLE(matrícula, marca, model, color)

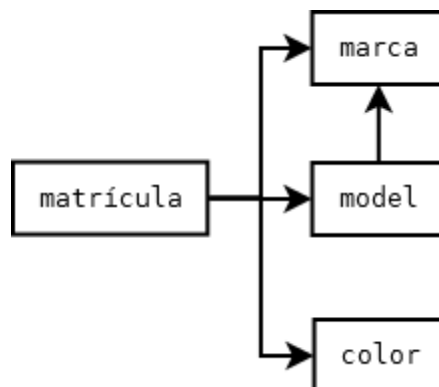
matrícula	marca	model	color
2847CBH	Ford	Ka	Vermell
1111HJG	Ford	Ka	Verd
2222FHJ	Toyota	Corolla	Blanc
3333FHJ	Fiat	Punto	Negre
4444HTP	Ford	c-Max	Vermell
5555JPT	Opel	Corsa	Blau

matrícula → marca

matrícula → model

matrícula → color

model → marca



1.2.2. Propietats de les dependències funcionals

- Reflexiva
 - Si Y pertany $X \rightarrow Y$
- Augment
 - Si $X \rightarrow Y \rightarrow XZ \rightarrow YZ$
 - $DNI \rightarrow Cognom \rightarrow DNI, Nom \rightarrow Cognom, Nom$
- Transitiva
 - Si $X \rightarrow Y$ i $Y \rightarrow Z \rightarrow X \rightarrow Z$
 - Matrícula \rightarrow Model i Model \rightarrow Marca \rightarrow Matrícula \rightarrow Marca
- Descomposició
 - Si $X \rightarrow YZ \rightarrow X \rightarrow Y$ i $X \rightarrow Z$
 - $DNI \rightarrow Nom, Cognom \rightarrow DNI \rightarrow Nom$ i $DNI \rightarrow Cognom$
- Unió
 - Si $X \rightarrow Y$ i $X \rightarrow Z \rightarrow X \rightarrow YZ$
 - $DNI \rightarrow Nom$ i $DNI \rightarrow Cognom \rightarrow DNI \rightarrow Nom, Cognom$
- Pseudotransitivitat
 - Si $X \rightarrow Y$ i $WY \rightarrow Z \rightarrow WX \rightarrow Z$
 - Matrícula \rightarrow Model i Model, Color \rightarrow Marca \rightarrow Matrícula, Color \rightarrow Marca

1.3. Formes normals de Codd i de Boyce-Codd

Cada pas correspon a una forma normal que té unes propietats. Conforme anem avançant en la normalització, les taules tenen un format més estricte (més fort) i per tant, són menys vulnerables a les anomalies d'actualització.

Les tres primeres formes normals van ser proposades per Codd al 1972, més endavant es va definir la forma Normal BCFN (Boyce/Codd) al 1974.

Al 1977 i 1979 en Ronald Fagin va proposar la quarta i la cinquena forma normal.

A mesura que avancem en la normalització incloem les Formes normals anteriors.

Per exemple, si diem que una taula està en 3FN implica que també està en 2FN i 1FN.



Vídeos:

- 1FN:
 - <https://www.youtube.com/watch?v=K7vzLrGCV50&list=PLQ9AAKW8HuJ5m0rmHKL88ZyjOIKejvrj0>
- 2FN:
 - <https://www.youtube.com/watch?v=A9sezRxNhWY&list=PLQ9AAKW8HuJ5m0rmHKL88ZyjOIKejvrj0&index=2>
- 3FN:
 - https://www.youtube.com/watch?v=GP_RcibUicQ&index=3&list=PLQ9AAKW8HuJ5m0rmHKL88ZyjOIKejvrj0
- BCFN:
 - <https://www.youtube.com/watch?v=Xo6IRbNgeXo>

1.3.1. Primera forma normal (1FN)

Una taula està en 1FN si el valor que conté cada atribut és únic i elemental (atòmic). En cadascun dels atributs només es pot incloure una dada encara que sigui composta, però no es pot incloure una llista de dades.

Per tant podem dir que la 1FN **prohibeix** els atributs **multivaluats**, atributs **compostos** i les seves **combinacions**.

Si utilitzem ER/EER → Relacional ja complim aquesta 1FN (primer forma normal)

A continuació donem un parell d'exemples d'esquemes de taules que violen la 1FN

Exemple 1 (Atribut amb un conjunt de valors atòmics):

Per exemple, Imaginem que tenim un model que ens reflecteix les pel·lícules que s'han emès en una sala de cinema.

No podem guardar en l'atribut data d'emissió, diferents dates en una mateixa "cel·la".

PEL·LICULA

<u>CodiPel·lícula</u>	Títol	Gènere	Data emissió
111	Star Trek	Ciència ficció	01/10/2015 02/10/2015 03/10/2015
222	King Kong	Fantàstic	01/10/2015 01/09/2015 03/10/2015
333	King Kong	Fantàstic	01/10/1981 05/06/1981

Solució 1: Per solucionar això caldria generar una nova relació(taula) introduint aquests atributs multivaluats.

PEL·LICULA

<u>CodiPel·lícula</u>	Títol	Gènere
111	Star Trek	Ciència ficció
222	King Kong	Fantàstic

PROJECCIONS

<u>CodiPel·lícula</u>	<u>Data emissió</u>
111	01/10/2015
111	02/10/2015
111	03/10/2015
222	01/10/2015
222	01/09/2015
222	03/10/2015

333	
333	

Exemple 2 (Atribut que és una taula):

Per exemple, imaginem que tenim un model que ens reflecteix el sistema d'un empresa que desenvolupa projectes i aquests són realitzats per els empleats els quals es vol saber el nº d'hores que hi dediquen.

EMPLEAT_PROJECTE

<u>Dni</u>	Nom	Projectes	
		Codi	Hores
27456363Z	Pere Pi	1	40
		3	15
32456783X	Marta Grau	1	25
		2	10

Per solucionar això caldria descompondre la taula en una nova taula.

EMPLEAT

<u>Dni</u>	Nom
27456363Z	Pere Pi
32456783X	Marta Grau

EMPLEAT_PROJECTE

<u>Dni</u>	<u>Codi</u>	Hores
27456363Z	1	40
27456363Z	3	15
32456783X	1	25
32456783X	2	10

Nota: Una taula en 1FN pot contenir informació redundant. La regla només contempla la necessitat de no tenir atributs multivaluats.

1.3.2. Segona forma normal (2FN)

Una taula està en segona forma normal (2FN) si, i només sí, està en 1FN i cada atribut que no forma part de la clau primària és completament dependent de la clau primària (no només d'una part).

La 2FN només s'aplica a les taules que tenen claus primàries compostes per 2 o més atributs. Si una taula està en 1FN i la seva clau primària és simple (només té un atribut), aquesta també està en 2FN.

Per passar una taula de 1FN a 2FN cal eliminar les dependències parcials de la clau primària.

Per fer això, mourem els atributs que són funcionalment dependents en una nova taula amb una còpia del seu determinant. El seu determinant estarà format pels atributs de la clau primària dels que depèn.

Exemple 1:

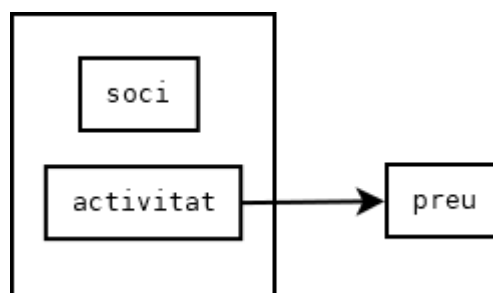
Imaginem que tenim un SI (Sistema d'Informació) a on tenim inscripcions d'activitats esportives d'un gimnàs. Resulta que tenim el següent esquema relacional per la relació INSCRIPCIÓ:

INSCRIPCIÓ(soci, activitat, preu)

activitat → preu (dependència funcional)

Troblem una dependència funcional parcial entre l'atribut activitat i preu, perquè cada activitat té un preu, independentment del soci que l'estigui realitzant.

<u>soci</u>	<u>activitat</u>	preu
27456363Z	Aeròbic	20
27456363Z	BodyPump	15
32456783X	Steps	10
32456783X	BodyPump	15



Les **anomalies** que es poden produir si es manté aquesta dependència són varies:

- No és possible conèixer el preu d'una activitat si no hi ha socis inscrits.
- Si es canvia el preu d'una activitat i no es canvia a totes les persones inscrites hi haurà un problema d'integritat ja que existiran 2 preus per una mateixa activitat. Cosa que no és desitjable. Com sabrem quin és el preu correcte?

Per passar la taula a 2FN s'ha d'eliminar l'atribut de la dependència parcial, que mou una còpia del seu determinant:

ACTIVITAT(codi, preu)

INSCRIPCIO(soci, activitat)

ON {activitat} REFERENCIA ACTIVITAT {codi}

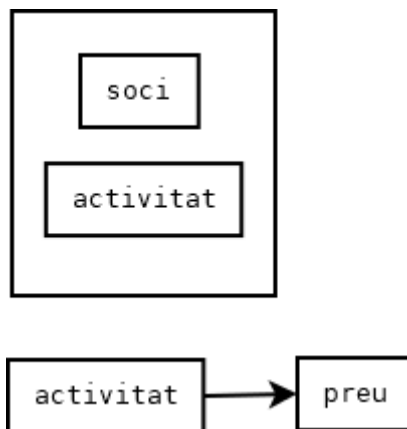
D'aquesta manera evitem els problemes citats anteriorment. Podem conèixer el preu de les activitats sense que hi hagin d'haver inscripcions, posat que el preu només està emmagatzemat una vegada i si aquest canvia serà el mateix per totes les inscripcions.

INSCRIPCIO

<u>soci</u>	<u>activitat</u>
27456363Z	Aeròbic
27456363Z	BodyPump
32456783X	Steps
32456783X	BodyPump

ACTIVITAT

<u>codi</u>	preu
Aeròbic	20
BodyPump	15
Steps	10



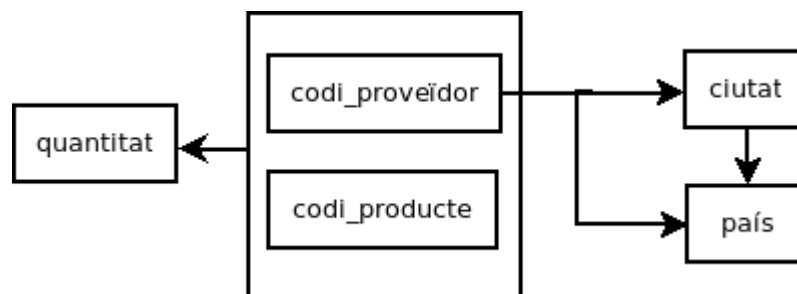
Exemple 2:

Imaginem que tenim un SI (Sistema d'Informació) que tenim guardada la informació de les comandes que hem de fer als nostres proveïdors i l'estructura de la taula és la següent:

COMANDES

<u>codi proveïdor</u>	país	ciutat	<u>codi producte</u>	quantitat
s1	Anglaterra	Londres	p1	300
s1	Anglaterra	Londres	p2	200
s1	Anglaterra	Londres	p3	400
s1	Anglaterra	Londres	p4	200
s1	Anglaterra	Londres	p5	100
s1	Anglaterra	Londres	p6	100
s2	França	Paris	p1	300
s2	França	Paris	p2	400
s3	França	Paris	p2	200
s4	Anglaterra	Bristol	p2	200
s4	Anglaterra	Bristol	p4	300
s4	Anglaterra	Bristol	p5	400

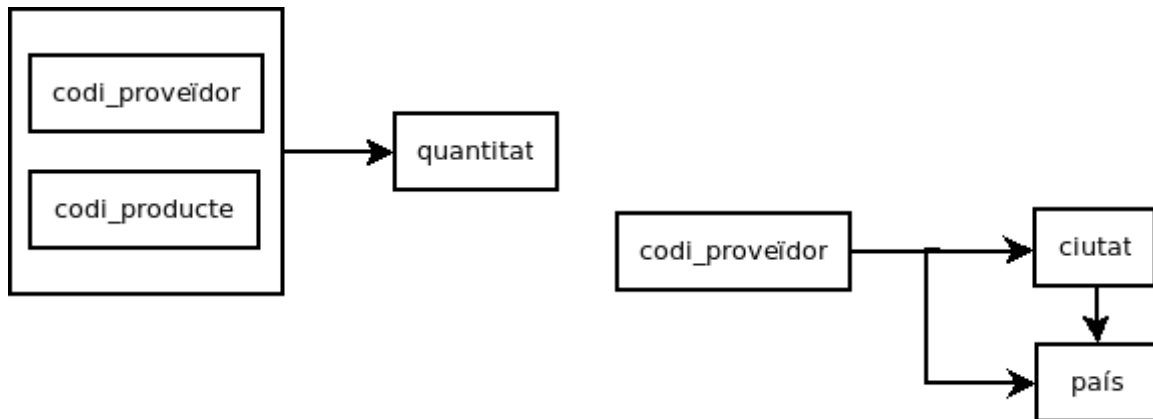
El diagrama de dependències funcionals de la taula és el següent:



En aquest exemple també podem veure una sèrie de problemes que provoca la redundància d'informació alhora de realitzar operacions de manipulació (insercions, esborrats i modificacions):

- **Inserció:** no es pot donar d'alta un nou proveïdor "s5" localitzat a "Atenes" fins que se li fa una comanda.
- **Esborrat:** si s'elimina una tupla, no només s'esborra la informació sobre el producte i la quantitat a proveir sinó també sobre el proveïdor. Per exemple el proveïdor s3.
- **Modificació:** si un proveïdor canvia de "Londres" a "Birmingham", s'han de modificar 6 tuples amb la mateixa informació i de forma transaccional.

Solució: Dividirem amb una taula nova aquelles dependències funcionals que són parcials de la clau primària.



COMANDES

<u>codi_proveïdor</u>	<u>codi_producte</u>	quantitat
s1	p1	300
s1	p2	200
s1	p3	400
s1	p4	200
s1	p5	100
s1	p6	100
s2	p1	300
s2	p2	400
s3	p2	200
s4	p2	200
s4	p4	300
s4	p5	400

PROVEÏDORS

<u>codi_proveïdor</u>	país	ciutat
s1	Anglaterra	Londres
s2	França	Paris
s3	França	Paris
s4	Anglaterra	Bristol

Una taula en 2FN encara pot presentar problemes de manipulació i redundància:

- Inserció: No podem inserir una ciutat o un país fins que no hi hagi un proveïdor a la ciutat.
- Eliminació: Si esborrem un proveïdor també esborrem la ciutat

Tercera forma normal (3FN)

Una taula està en tercera forma normal (3FN) si, i només si, està en 2FN i cada atribut que no forma part de la clau primària no depèn transitivament de la clau primària.

Una altra manera de dir-ho seria dient que no existeixen atributs que no pertanyin a la clau primària que puguin ser coneguts mitjançant un altre atribut que no formi part de la clau primària, no hi ha dependències funcionals transitives.

Per passar una taula de 2FN a 3FN cal eliminar les dependències transitives.

Per fer això, mourem els atributs que depenen transitivament en una altra taula amb còpia del seu determinant (l'atribut o atributs no clau dels que depèn).

Exemple 1:

Imaginem que estem al davant d'un SI que vol modelitzar el negoci de lloguer dels pisos d'un edifici. Regles de negoci: Un mateix inquilí no pot estar a dos edificis. El lloguer de tots els pisos d'aquest mateix edifici té el mateix preu. Resulta que tenim el següent esquema relacional per la relació/taula HABITA:

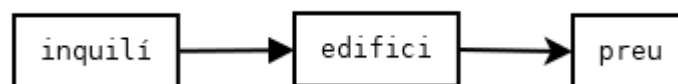
HABITA(inquilí, edifici, preu)

inquilí → edifici

edifici → preu (dependència funcional transitiva)

HABITA

<u>inquilí</u>	edifici	quantitat
i1	e1	600
i2	e2	400
i3	e1	600
i4	e1	600
i5	e2	400
i6	e1	600
i7	e1	600



Aquesta dependència existeix perquè cada edifici té un preu de lloguer, independentment de l'inquilí que l'habiti.

Mantenir aquesta dependència pot dur a diferents anomalies i/o problemes:

- No és possible conèixer el preu del lloguer si no hi ha inquilins.
- Si es modifica el preu del lloguer d'un edifici només per alguns inquilins provocarà la violació de la regla de negoci, ja que s'ha dit que els inquilins d'un mateix edifici han de pagar el mateix.

Per passar la taula a 3FN s'ha de moure l'atribut de la dependència transitiva en una nova taula emportant-nos una còpia del seu determinant:

LLOGUER(edifici, preu)

HABITA(inquili, edifici)

ON {edifici} REFERENCIA LLOGUER {edifici}

D'aquesta manera evitem els problemes citats anteriorment.

HABITA

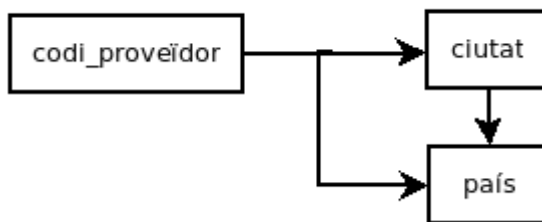
<u>inquilí</u>	edifici
i1	e1
i2	e2
i3	e1
i4	e1
i5	e2
i6	e1
i7	e1

LLOGUER

<u>edifici</u>	quantitat
e1	600
e2	400

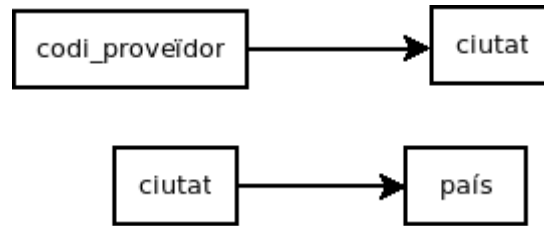
Exemple 2:

Continuant amb l'exemple anterior hem vist que la taula de PROVEÏDORS encara presenta problemes de manipulació. Verificant les regles de 3FN veiem que aquesta taula no les compleix.



Mirant el diagrama de dependències funcionals veiem que país és depenent transitivament de la clau primària a través de ciutat. En conseqüència no està en 3FN.

Per solucionar-ho agafarem l'atribut que depèn transitivament de la clau i l'afegirem a una nova taula.



PROVEÏDORS

<u>codi proveïdor</u>	Ciutat
s1	Londres
s2	Paris
s3	Paris
s4	Bristol

CIUTATS

<u>ciutat</u>	País
Londres	Anglaterra
Paris	França
Bristol	Anglaterra

Nota:

- L'avantatge de tenir un esquema de taula en 3FN és que queda eliminada informació redundant, la qual cosa comporta estalvi d'espai d'emmagatzemament i reducció d'anomalies de manipulació.
- La 3FN acostuma a ser suficient per obtenir un bon disseny dels esquemes de les taules de les bases de dades, encara que algunes aplicacions molt especialitzades poden requerir un nivell més alt de normalització.

1.3.3. Forma normal de Boyce/Codd (BCFN)

Una taula està en forma normal Boyce/Codd (FNBC) si, i només si, tots els determinants de la taula són una clau candidata de la taula, no que només en formin part.

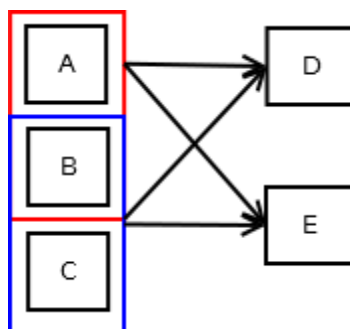
La 2FN i la 3FN eliminen les dependència parcials i les dependència transitives respectivament de la clau primària. Però aquest tipus de dependències encara poden existir sobre altres claus candidates, si aquestes existeixen.

La BCFN és més rigorosa/forta que la 3FN, per tant, tota taula en BCFN també està en 3FN, però no el revés.

La violació de la BCFN és poc freqüent ja que es dona sota certes circumstàncies que rarament es presenten.

Hem de comprovar si una taula està en BCFN si té 2 o més claus candidates compostes que tenen almenys un atribut en comú.

No és freqüent trobar relacions/taules amb més d'una clau candidata compostes i que com a mínim tinguin un atribut en comú. En cas contrari, les 3FN i FNBC són equivalents. **O sigui, Si la clau primària té un sol atribut i està en 3FN està en FNBC.**



Podem englobar-ho amb:

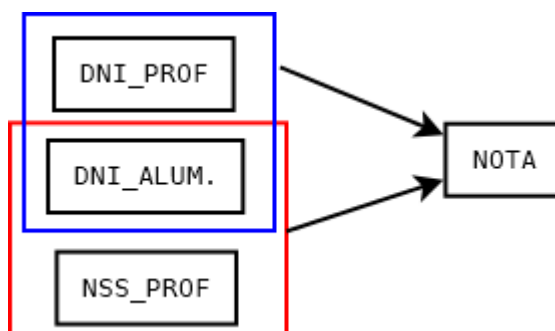
- Es busquen tots els determinants funcionals d'una taula
- Els determinants funcionals que no siguin claus candidates d'aquesta taula (ni clau principal ni clau alternativa) generaran una nova taula on:
 - La clau principal serà el determinant funcional
 - Els atributs seran els atributs no clau que depenen funcionalment de la nova clau.

Exemple:

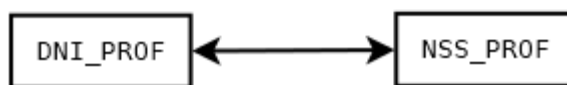
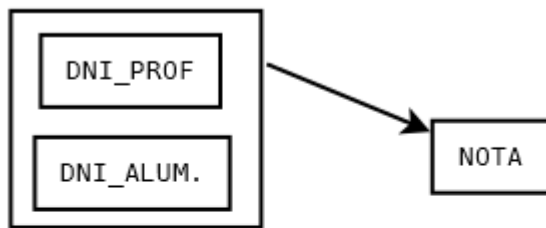
<u>dni_prof.</u>	<u>dni_estudiant</u>	nss_prof.	nota
111	AAA	1A	5
111	BBB	1A	6
111	CCC	1A	8

Quines són les claus candidates:

- dni_tutor, dni_estudiant
- nss_tutor, dni_estudiant



Per passar la taula a 3FNBC hem de moure els atributs que siguin claus candidates en una nova taula i escollir una de les claus candidates perquè sigui la clau de la taula que queda.



PROFESSORS

<u>dni_prof.</u>	nss_prof.
111	1A
222	2A
333	3A
444	4A

NOTES

<u>dni_estudiant</u>	dni_prof	nota
AAA	111	5
BBB	111	6
CCC	111	8

Exercici1

Normalitza la següent taula que intenta de guardar les comandes d'una base de dades per una empresa que es dedica a vendre productes de tennis.

COMANDES (comanda_id, data, client_id, client_nom, client_pob, article_codi, article_nom, quantitat, preu_unitat)

comanda_id	data	client_id	client_nom	client_pob	article_codi	article_nom	quantitat	preu_unitat
2301	23/02/2011	101	Pere Pi	Barcelona	3789	Xarxa	3	35,00
2301	23/02/2011	101	Pere Pi	Barcelona	4011	Raqueta	6	65,00
2301	23/02/2011	101	Pere Pi	Barcelona	9132	Paquet-3	8	4,75
2302	27/02/2011	107	Marta Mas	Girona	4011	Raqueta	2	65,00
2303	27/02/2011	110	Andreu Cau	Bilbao	5794	Paquet-6	4	5,00
2303	27/02/2011	110	Andreu Cau	Bilbao	9132	Paquet-3	8	4,75

Una botiga que ven productes de tennis ens demana que li dissenyem una base de dades pel seu negoci:

- El client ens diu que vol tenir la emmagatzemades les dades dels diferents clients que li realitzen comandes. Les dades que vol guardar de cada client són: el codi client, el nom del client, la població del client.
- Ens comenta que vol portar un control dels diferents articles que té a la botiga, aquests els identifica amb un codi de barres únic per cada producte. D'aquests també en vol saber el seu nom, el preu de venda el públic i l'estoc actual.
- Per tal de saber la facturació mensual veiem que cal guardar la informació de les diferents comandes que realitzen els clients juntament amb la quantitat de cada producte que es demana. La data d'una comanda ens determina quin és el mes de facturació per aquella comana.

