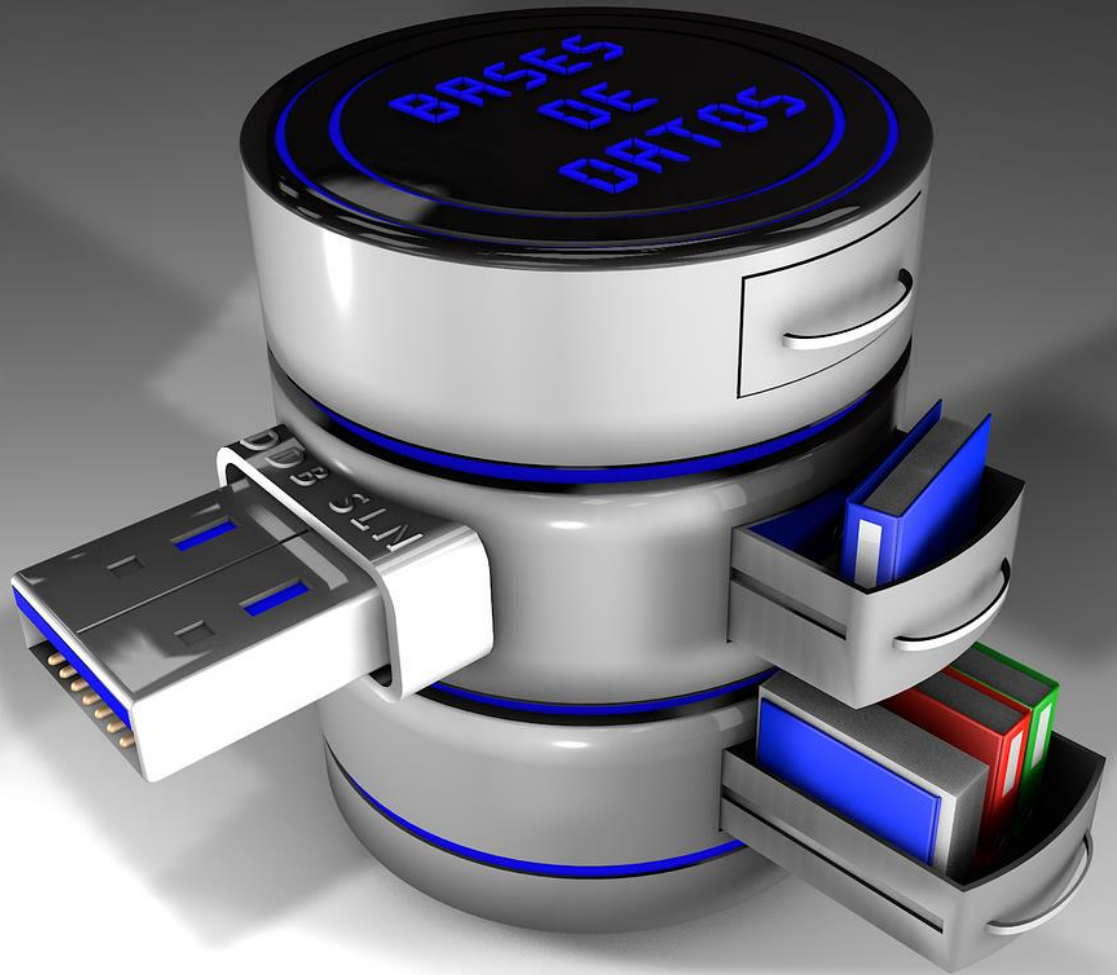


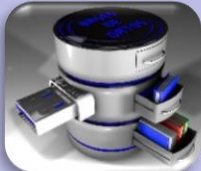
UNITAT DIDÀCTICA 7

Formes normals



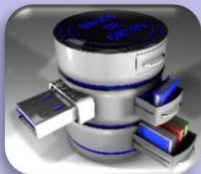
Índex

1. Introducció
2. Teoria de la normalització.
3. Primera Forma Normal (1FN)
4. Segona Forma Normal (2FN)
5. Tercera Forma Normal (3FN)
6. Forma Normal Boyce/Codd



Índex

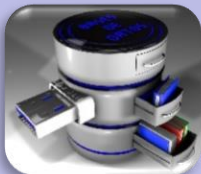
- 7. Quarta Forma Normal (4FN)
- 8. Quinta Forma Normal (5FN)
- 9. Pros i contres de la normalització
- 10. Exemple normalitzat (3FN)



Introducció

Quan dissenyem una base de dades relacional, en general pretendrem obtenir un conjunt d'esquemes de relació que ens permeten emmagatzemar la informació sense redundàncies i de manera que puguem recuperar aquesta informació fàcilment.

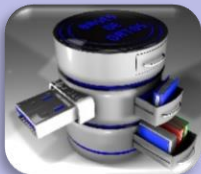
Fins ara hem après a dissenyar un model conceptual de la base de dades, model entitat relació, i establir el seu disseny lògic, model relacional, obtenint així una sèrie de taules finals que són les candidates a formar la nostra base de dades.



Introducció

No obstant això, aquestes taules han sigut obtingudes a partir d'un disseny conceptual elaborat sense cap mena de regles, per la qual cosa podem obtenir un disseny de taules més o menys heterogeni.

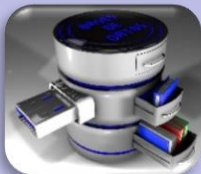
La **teoria de la normalització** que consisteix en un conjunt de **regles formals** que ens permeten confeccionar un disseny lògic, o assegurar que un disseny lògic que ja hem confeccionat complisca una sèrie de propietats, corregint l'estructura de les dades de les taules.



Introducció

S'ha d'evitar una sèrie de problemes com ara:

- ✓ Incapacitat d'emmagatzemar uns certs fets.
- ✓ Redundàncies i, per tant, possibilitat d'inconsistències.
- ✓ Ambigüitats.
- ✓ Pèrdua d'informació.
- ✓ Aparició en la B.D. d'estats no vàlids en el món real, és el que es diu anomalies d'inserció, esborrat i modificació.

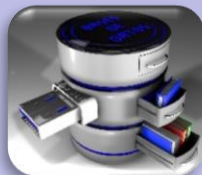


Teoria de la normalització

La teoria de la normalització, desenvolupada per Edgar F. Codd en 1972 permet millorar el disseny lògic d'un sistema d'informació.

Les regles formals en les quals es fonamenta la teoria de la normalització són conegudes amb el nom de Formes Normals, que són un conjunt de restriccions que han de complir les relacions.

Existeixen sis formes normals, de manera que quan la base de dades compleix les regles de la primera forma normal es considera que està en primera forma normal (1FN), quan passen la segona, que està en segona forma normal (2FN), etc. A més, una base de dades de la qual s'afirme que està en 2FN, està també en 1FN, perquè les formes normals s'apliquen de manera successiva.

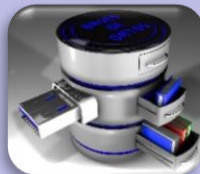


Teoria de la normalització

En principi, Codd va definir la 1FN, 2FN i 3FN, amb la idea que era més desitjable que una relació estiguera en 2FN que en 1FN, i al seu torn, era millor que estiguera en 3FN que en 2FN.

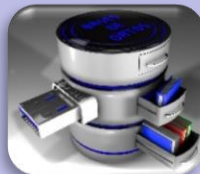
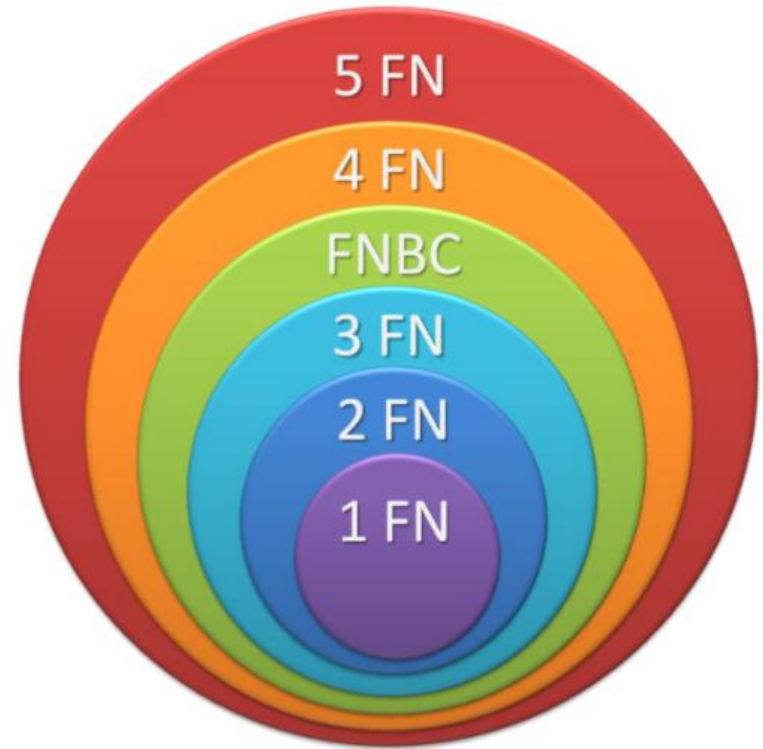
Amb posterioritat (1974), i atés que la 3FN no contemplava alguns casos particulars, Raymond F. BoyceBoyce va ajudar a Codd a redefinir la 3FN i va ser el que es va dir la forma normal de BOYCE/CODD (FNBC).

En 1977 i 1979, Ronald Fagin va introduir la 4FN i 5FN respectivament.



Teoria de la normalització

De les sis formes normals, generalment només s'apliquen sobre les bases de dades les tres primeres i/o la de Boyce/Codd, considerant que una base de dades que està en 3FN o en FNBC és una base de dades correctament dissenyada.

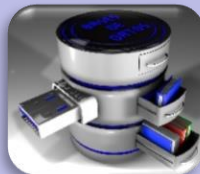


Teoria de la normalització

Les formes normals es basen en el concepte de dependència, que comprén les restriccions definides sobre els atributs d'una relació. Tenim tres tipus de dependències:

Univers de les relacions

- Dependències funcionals. S'usa per a la 1FN, 2FN, 3FN i FNBC.
- Dependències multivaluadas (4FN)
- Dependències de Join o de combinació (5FN)

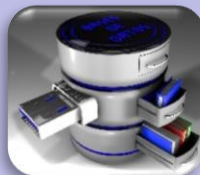


Teoria de la normalització

Les formes normals es basen en el concepte de dependència, que comprén les restriccions definides sobre els atributs d'una relació. Tenim tres tipus de dependències:

Univers de les relacions

- Dependències funcionals. S'usa per a la 1FN, 2FN, 3FN i FNBC.
- Dependències multivaluadas (4FN)
- Dependències de Join o de combinació (5FN)



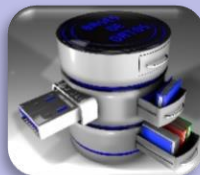
Teoria de la normalització

Dependència funcional

Donada una relació R , l'atribut Y de R depén funcionalment de l'atribut X de R ($R.X \rightarrow R.Y$) si i només si un valor d' Y en R està associat a cada valor X en R (en qualsevol moment donat).

Òbviament, si l'atribut X és una clau candidata de la relació R (o és la clau primària), aleshores tots els atributs Y de la relació han de dependre funcionalment d'eixe atribut X .

Per exemple, si tenim una relació amb clients d'un hotel i dos dels seus atributs són el número de client i el seu nom, podem afirmar que el nom té una dependència funcional del número de client; és a dir, cada vegada que en una tupla el número de client, és segur que el nom de client serà sempre el mateix.



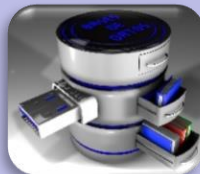
Teoria de la normalització

Dependència funcional completa

Es diu que l'atribut Y de la relació R és per complet dependent funcionalment de l'atribut X de R si depén funcionalment de X i no depén funcionalment de cap subconjunt propi de X.

En aquest cas X ha d'estar format per més d'un atribut.

Si tenim una clau formada pel codi de la factura i la data. Les dades del client tenen dependència del codi de la factura, ja que aqueix codi de la factura correspon només a aqueix client, però aqueixa data pot correspondre a més d'un client. Per tant les dades del client, no tenen dependència completa amb la clau.



Teoria de la normalització

Dependència funcional transitiva

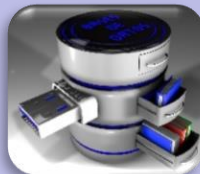
És una dependència d'un atribut no principal d'un altre no principal.

Cuando se tienen tres atributos X, Y y Z se cumple que $X \rightarrow Y$, $Y \rightarrow Z$ pero no $Y \rightarrow X$. Por tanto Z tiene dependencia transitiva con respecto a X, a través de Y.

Per exemple tenim la següent taula:

Pedido (NumPedido, DataPedido, article, descripción, cantitat, preu, data_prevista, NomProv, Pais_prov)

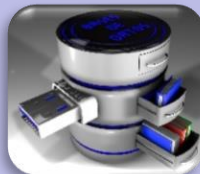
NumPedido \rightarrow NomProv	}	PaisProv \rightarrow NumPedido
NomProv \rightarrow PaisProv		



Primera Forma Normal (1FN)

És una forma normal inherent a l'esquema relacional, que compleix tota taula realment relacional. Per a que una taula estiga en **primera forma normal** cada columna d'una taula ha de ser atòmica, és a dir, que cada atribut ha de contindre un únic valor del domini.

ALUMNAT		
NIA	Nom	Assignatures
1212	Andrés García Pérez	Bases de Dades, Anglès
2233	Andrea Martínez Aguado	Fonaments de Maquinari, Llenguatge de marques, Formació i Orientació Laboral

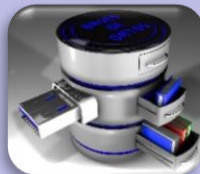


Primera Forma Normal (1FN)

Per a solucionar-ho, podem crear noves columnes (separant els cognoms) o noves taules amb les corresponents claus alienes

ALUMNAT			
NIA	Nom	Cognom1	Cognom2
1212	Andrés	García	Pérez
2233	Andrea	Martínez	Aguado

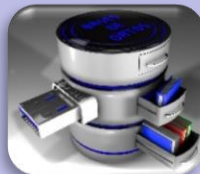
MATRICULAT	
Cod_aL	Assignatura
1212	Bases de Dades
1212	Anglès
2233	Fonaments de Maquinari
2233	Llenguatge de marques
2233	Formació i Orientació Laboral



Segona Forma Normal (2FN)

Una taula està en **segona forma normal** (2FN) quan està en 1FN i a més tots els atributs que no formen part de la clau principal tenen dependència funcional de la clau completa i no de part d'ella.

Significa, per tant, que en una relació només s'ha d'emmagatzemar informació sobre un tipus d'entitat, i que es tradueix en què els atributs que no aporten informació directa sobre la clau principal han d'emmagatzemar-se en una relació separada.



Segona Forma Normal (2FN)

Per exemple: En la taula PRODUCTIVITAT recollim els temps que utilitza cada empleat a realitzar determinades reparacions en relació amb els temps previstos, per a calcular la productivitat d'aqueix empleat.

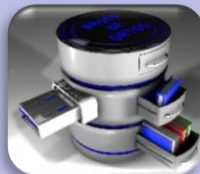
PRODUCTIVITAT (NumReparacio, CodEmpleat, DataActuació, NomEmpleat, Lloc_treball, SalariBase, NumHores, DescriReparacio, TempsEstimat, ImportReparacio, TempsEmpleat)

Dependències:

NumReparacio, CodEmpleat, DataActuació → TempsEmpleat

CodEmpleat → NomEmpleat, Lloc_treball, SalariBase

NumReparacio → DescriReparacio, TempsEstimat, ImportReparacio



Segona Forma Normal (2FN)

Tenim dependències parcials, és a dir, dependències basades en una part de la clau primària que produiran redundàncies en insertar noves files en la taula. Per a eliminar eixes dependències **crearem noves taules** que recullen les dependències:

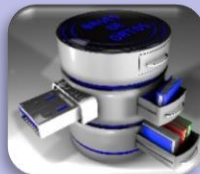
PRODUCTIVITAT (NumReparacio, CodEmpleat, DataActuació, TempsEmpleat)

EMPLEATS (CodEmpleat, NomEmpleat, Lloc_treball, SalariBase)

REPARACIONS (NumReparacio, DescriReparacio, TempsEstimat, ImportReparacio)

Aquestes taules estan en 2FN ja que:

- ✓ Estan en 1FN
- ✓ Cap atribut depén solament d'una part de la clau primària.



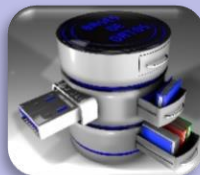
Tercera Forma Normal (3FN)

Una base de dades està en tercera forma normal(3FN) si està en 2FN i no existeixen atributs que no pertanyen a la clau primària que puguin ser coneguts mitjançant un altre atribut que no forme part de la clau primària. És a dir, que no existisquen dependències funcionals transitives.

Seguint amb l'exemple anterior veiem que existeixen dependències transitives:

EMPLEATS (CodEmpleat, NomEmpleat, Lloc_treball, SalariBase)

Lloc_treball → SalariBase



Tercera Forma Normal (3FN)

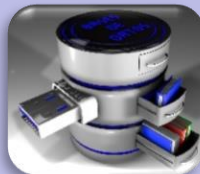
Para eliminar esta dependència guardem eixos atributs en una nova taula. No obstant això deixem el determinant Lloc_treball) en la taula original com a clau aliena per a poder relacionar les dues taules.

PRODUCTIVITAT (NumReparacio, CodEmpleat, DataActuació, TempsEmpleat)

REPARACIONS (**NumReparacio**, DescriReparacio, TempsEstimat, ImportReparacio)

EMPLEATS (CodEmpleat, NomEmpleat, Lloc_treball)

LLOCS (Lloc_treball, SalariBase)



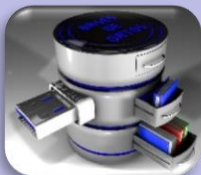
Tercera Forma Normal (3FN)

Tot i que la normalització de bases de dades implica un major esforç de programació, la 3FN és considerada com l'estàndard per als esquemes relacionals i només es descarta baix comptades excepcions.

Una d'elles seria la denormalizació de bases de dades que estan en la 3FN a la segona forma normal. Això es fa perquè els Joins que enllacen diverses taules en bases de dades molt grans tarden molt temps.

Denormalizando la base de dades s'espera reduir el nombre de taules i amb això la duració de la consulta.

La major part de les vegades, la normalització acostuma a finalitzar en la tercera forma normal.

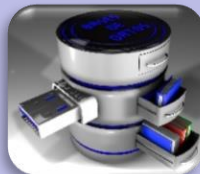


Forma Normal Boyce/Codd(FNBC)

El problema de la 3FN és que no treballa amb relacions que:

- Tenen diverses claus candidates,
- Eixes claus candidates són compostes, i
- Les claus candidates tenen almenys un atribut en comú.

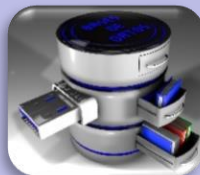
Per això, es defineix la FNBC per al cas en el qual existisca més d'una clau candidata, i que es complisquen les condicions anteriors. En el cas en el qual no es donen aquestes condicions, o bé no existisca més d'una clau candidata (només la clau primària) la FNBC és completament equivalent a la 3FN.



Forma Normal Boyce/Codd(FNBC)

Per tant, una forma senzilla de comprovar si una relació es troba en FNBC consisteix a comprovar, a més de que estiga en 3FN, el següent:

1. Si no existeixen claus candidates compostes (amb diversos atributs), està en FNBC.
2. Si existeixen diverses claus candidates compostes i aquestes tenen un element comú, pot no estar en FNBC.



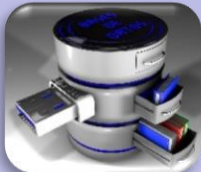
Quarta Forma Normal (4FN)

Una taula està en 4FN si:

- ✓ està en FNBC
- ✓ No hi ha dependències multivaluades llevat que siguin trivials.

Dependència multivaluada: si dos atributs sense relació entre sí depenen del mateix atribut.

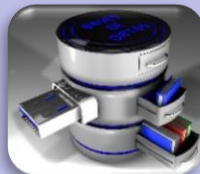
Dependència trivial: Si un atribut es dependent funcionalment de sí mateix.



Quarta Forma Normal (4FN)

Pot veure's que el client amb el número 234 ha demanat els articles 1-0023-D i 2-0023-D, que s'han d'entregar en la seua adreça amb el codi postal 12345. Per al client 567, els articles 1-0023-D, 3-0023-D, 4-0023-D i 5-0023-D s'entregaran en el codi postal 56789.

Nº cliente	Nº artículo	CP
234	1-0023-D	12345
234	2-0023-D	12345
567	1-0023-D	56789
567	3-0023-D	56789
567	4-0023-D	56789



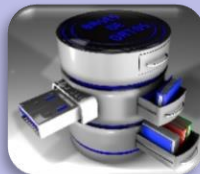
Quarta Forma Normal (4FN)

Els registres només poden identificar-se amb una superclave composta pels tres atributs (núm. client, núm. article i codi postal).

Al no donar-se cap atribut no-clau la taula està en 3FN.

Tampoc presenta dependències transitives ni trivials, de manera que també compleix amb la FNBC.

No obstant això, sí que conté dependències multivaluadas, ja que l'atribut núm. d'article i l'atribut codi postal depenen de núm. de client però no guarden relació entre si.

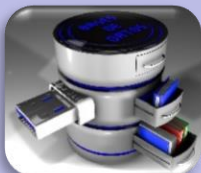


Quarta Forma Normal (4FN)

L'inconvenient d'aquest disseny és que cada vegada que es registre un nou article per a un client, també serà necessari incloure el codi postal, de manera que hi haurà dades redundants.

Si transformem la taula a la 4FN, aquestes repeticions poden reduir-se. Per a això, s'ha de fragmentar la taula de tal manera que no presenti cap dependència o, almenys, només dependències multivaluades trivials.

Crearem , dos taules separades, la qual cosa és possible perquè el número d'article i el codi postal no estan relacionats.



Quarta Forma Normal (4FN)

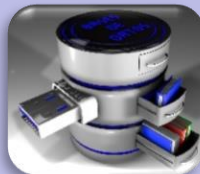
Article

Nº cliente	Nº artículo
234	1-0023-D
234	2-0023-D
567	1-0023-D
567	4-0023-D

Lloc_entrega

Nº cliente	CP
234	12345
567	56789

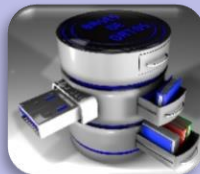
Com veiem, la quarta forma normal elimina les redundàncies produïdes per les dependències multivaluades, en aquest cas, en la columna CP



Quinta Forma Normal (5FN)

Una taula està en la 5FN quan satisfà les condicions de la 4FN i compleix, a més, aquesta condició:

La taula no pot fragmentar-se més sense que es perdi informació.

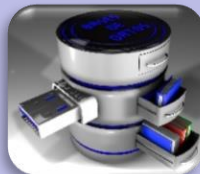


Quinta Forma Normal (5FN)

Suposem una empresa que administra una pàgina web TYPO3 i una botiga en línia *Magento. Tres empleats són els encarregats de gestionar-les: Pérez, García i González, i cadascun d'ells aporta coneixements de diferent índole.

Coneixements_per_empleat_projecte

Empleado	Proyecto	Conocimientos
Pérez	Magento	PHP
Pérez	Magento	SQL
Pérez	TYPO3	JavaScript
Pérez	TYPO3	SQL
García	Magento	PHP
García	TYPO3	JavaScript
González	TYPO3	PHP



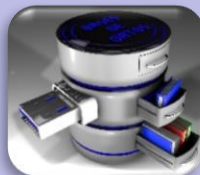
Quinta Forma Normal (5FN)

L'empleat Pérez aporta al projecte Magento els seus coneixements en PHP i SQL i per a la pàgina web TYPO3 recorre a SQL i a Javascript.

García també s'encarrega de la programació PHP de la botiga en línia i treballa amb Javascript en la pàgina web.

Finalment, González només participa en el projecte TYPO3, encarregant-se en solitari de la programació PHP.

D'això es conclou que per a utilitzar Magento es requereix experiència en PHP i SQL, mentre que un projecte de TYPO3 dona per descomptat tindre coneixements en PHP, SQL i Javascript.

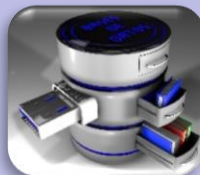


Quinta Forma Normal (5FN)

La taula només posseeix una **clau composta per tots els atributs**, complint així amb la 3FN i la FNBC. Al no donar-se dependències entre els tres atributs també seria una representant de la 4FN.

Es tractarà ara de comprovar si també està en la 5FN. Per a això fragmentarem la taula original Coneixements_per_empleat_projecte en tres taules:

- Participació en projecte
- Coneixements per empleat
- Requeriments per projecte



Quinta Forma Normal (5FN)

Participació en projecte

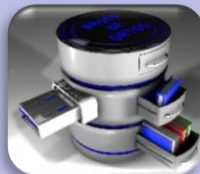
Proyecto	Empleado
Magento	Pérez
Magento	García
TYPO3	Pérez
TYPO3	García
TYPO3	González

Coneixements per empleat

Empleado	Conocimientos
Pérez	PHP
Pérez	SQL
Pérez	JavaScript
García	PHP
García	JavaScript
González	PHP

Requeriments per projecte

Proyecto	Cualificación
Magento	PHP
Magento	SQL
TYPO3	JavaScript
TYPO3	SQL
TYPO3	PHP

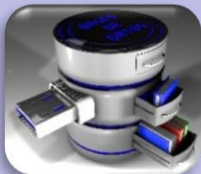


Quinta Forma Normal (5FN)

A primera vista, l'escissió de la taula original aporta claredat.

Amb tot, les taules que com aquesta resulten de la normalització igualen en quantitat d'informació a la taula original?

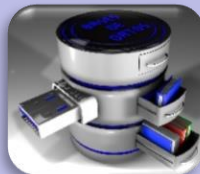
Per a esbrinar-ho hem de dur a terme un Join, una consulta a la base de dades que implique a les tres taules. El resultat és sorprenent



Quinta Forma Normal (5FN)

En reconstruir la taula original hem de donar per descomptat que cada empleat implicat en el projecte aporta les seues qualificacions si el projecte les requereix. La informació que González s'ha encarregat ell només de programar el projecte TYPO3 en PHP s'ha perdut. Això vol dir que la **taula original no pot fragmentar-se** sense pèrdues, per tant està en 5FN.

Empleado	Proyecto	Conocimientos
Pérez	Magento	PHP
Pérez	Magento	SQL
Pérez	TYPO3	JavaScript
Pérez	TYPO3	SQL
Pérez	TYPO3	PHP
García	Magento	PHP
García	TYPO3	JavaScript
García	TYPO3	PHP
González	TYPO3	PHP

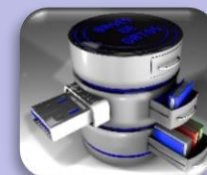


Quinta Forma Normal (5FN)

taula original

Empleado	Proyecto	Conocimientos
Pérez	Magento	PHP
Pérez	Magento	SQL
Pérez	TYPO3	JavaScript
Pérez	TYPO3	SQL
García	Magento	PHP
García	TYPO3	JavaScript
González	TYPO3	PHP

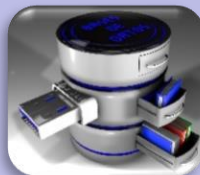
Empleado	Proyecto	Conocimientos
Pérez	Magento	PHP
Pérez	Magento	SQL
Pérez	TYPO3	JavaScript
Pérez	TYPO3	SQL
Pérez	TYPO3	PHP
García	Magento	PHP
García	TYPO3	JavaScript
García	TYPO3	PHP
González	TYPO3	PHP



Pros i contres de la normalització

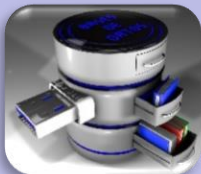
L'objectiu de la normalització és la **reducció dels valors duplicats** i si es normalitza a una base de dades en alguna de les formes normals descrites, la taula resultant presentarà l'avantatge de comptar amb menys redundància que l'original. Així, **la normalització simplifica el manteniment dels bancs de dades**.

D'altra banda, normalitzar una base de dades implica sempre separar els atributs en taules independents. Això requereix probablement integrar claus alienes i pot conduir a redundàncies de claus. Però el seu major inconvenient és que en una base de dades normalitzada les dades que formen un tot lògic ja no es guarden junts. Si es vol reunir les dades que figuren en taules separades és necessari executar un Join.



Pros i contres de la normalització

Les consultes a les bases de dades amb Joins permeten filtrar dades complexes; però dur-les a terme requereix un esforç major que una consulta simple, al que ve a sumar-se la lentitud de l'execució de la consulta quan els Joins impliquen un gran nombre de taules.

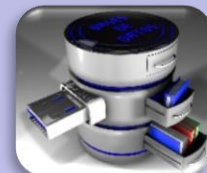


Exemple normalitzat (3FN)

En el nostre cas, totes les normalitzacions que farem sern fins a la **3FN**

Tenim una taula anomenada *orden* que conté informació dels pedidos

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad	Id_art	Nom_art	Cant	Precio
2301	23/02/11	101	Martin	Riobamba	3786	Red	3	35,00
2301	23/02/11	101	Martin	Riobamba	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Riobamba	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Ambato	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Quito	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Quito	3141	Funda	2	10,00

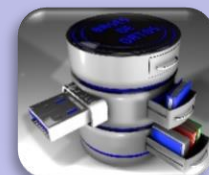


Exemple normalitzat (3FN)

1FN

- Totes les dades son atòmiques. → **OK**
- Alguns autors recomanen, en la 1FN desfer-se de les dades repetides → **NO OK**

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad	Id_art	Nom_art	Cant	Precio
2301	23/02/11	101	Martin	Riobamba	3786	Red	3	35,00
2301	23/02/11	101	Martin	Riobamba	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Riobamba	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Ambato	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Quito	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Quito	3141	Funda	2	10,00

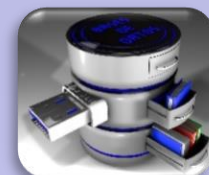


Exemple normalitzat (3FN)

1FN

Passos a seguir:

- ✓ hem d'eliminar els grups repetits
- ✓ hem de crear una nova taula amb la CP de la taula original (ordenes) i el grup repetit



Exemple normalitzat (3FN)

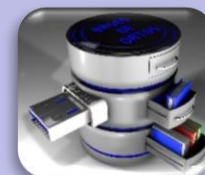
1FN

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad
2301	23/02/11	101	Martin	Riobamba
2302	25/02/11	107	Herman	Ambato
2303	27/02/11	110	Pedro	Quito

Artículos_ordenes

Id_orden	Id_Art	Nom_art	Cant	Precio
2301	3786	Red	3	35,00
2301	4011	Raqueta	6	65,00
2301	9132	Paq-3	8	4,75
2302	5794	Paq-6	4	5,00
2303	4011	Raqueta	2	65,00
2303	3141	Funda	2	10,00

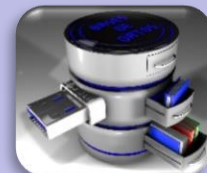


Exemple normalitzat (3FN)

2FN

Hem d'eliminar qualsevol columna no clau que no depenga de la clau prevaldre de la taula.

- ✓ Determinar les dependències funcionals completes
- ✓ Eliminar aqueixes columnes de la taula original
- ✓ Crear una nova taula

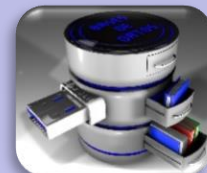


Exemple normalitzat (3FN)

2FN

La taula ordenes està en 2FN, ja que qualsevol valor únic de ID_ORDEN determina un només valor per a cada columna. Per tant totes les columnes són dependents de la clau primària.

La taula Articulos_ordenes no està en 2FN ja que les columnes PRECIO i NOM_ART són dependents de ID_ART, però no són dependents de ID_ORDRE. El que farem és eliminar aquestes columnes i crear una nova taula amb aquestes columnes i la clau primària de la qual depenen



Exemple normalitzat (3FN)

2FN

Artículos_ordenes

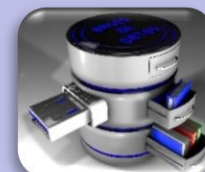
Id_orden	Id_Art	Cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

Artículos

Id_Art	Nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00

ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Ciudad
2301	23/02/11	101	Martin	Riobamba
2302	25/02/11	107	Herman	Ambato
2303	27/02/11	110	Pedro	Quito



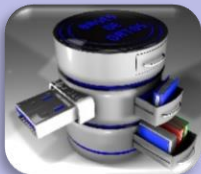
Exemple normalitzat (3FN)

3FN

La 3FN es complix quan està en 2FN i no existeixen atributs que no pertanyen a la clau primària que puguin ser coneguts mitjançant un altre atribut que no forme part de la clau primària. És a dir, que no existisquen dependències funcionals transitives.

Per tant els passos a seguir són:

- ✓ determinar les columnes dependents
- ✓ eliminar aqueixes columnes
- ✓ crear una nova taula

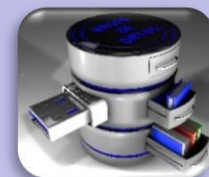


Exemple normalitzat (3FN)

3FN

Tant la taula articles com la taula articles_ordenes es troben en 3FN. No obstant això la taula ordenes no ho està, ja que nom_cliente i ciudad són dependents de l'id_cliente i eixa columna no és clau principal.

Crearem una nova taula amb les dades dels clients



Exemple normalitzat (3FN)

3FN

ordenes

Id_orden	Fecha	Id_cliente
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

clientes

Id_cliente	Nom_cliente	Ciudad
101	Martin	Riobamba
107	Herman	Ambato
110	Pedro	Quito

artículos_ordenes

Id_orden	Id_Art	Cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

articulos

Id_Art	Nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
3141	Funda	10,00

