





### Objectius

Els objectius d'aquest tema són:

1. Conèixer l'estructura bàsica del model: la taula.
2. Saber descriure els camps, amb el seu domini, i saber triar la clau principal.
3. Saber definir altres restriccions (no nul, únic, ...)
4. Entendre el concepte de clau externa, i saber definir les necessàries.
5. Saber traduir correctament del Model Entitat-Relació al Model Relacional.
6. Redefinir les taules necessàries atenent a motius pràctics.



### Coneixements previs

Els coneixements previs necessaris per al tema són els del tema 2: Model Entitat-Relació. En certa manera aquest tema aclarirà conceptes d'aquell tema, ja que ens permetrà aplicar-lo a una cosa palpable com són les taules, directament implementables en un SGBD Relacional. Així podrem concretar possibles ambigüitats del esquema Entitat-Relació, i saber si valen la pena determinades restriccions.

Però entendre el Model Entitat-Relació i dominar la interpretació d'un esquema serà bàsic en el disseny d'una Base de Dades. Sempre començarem pel Model Entitat-Relació, i posteriorment el traduirem al Model Relacional, aplicant en tot cas el sentit comú.

En 1970 **E.F. Codd** va introduir la teoria matemàtica de les relacions en el camp dels SGBD. El model de dades que va crear s'anomena **MODEL RELACIONAL**, amb una base matemàtica (la de les relacions) molt sòlida, on les dades s'estructuren en forma de relacions (taules) que són estructures de dades simples i uniformes, que permeten una fàcil comprensió.

En el moment en què va sorgir, els models que funcionaven eren el jeràrquic i el de xarxa (també anomenat Codasyl, pel grup de treball que va estandaritzar-lo). Com a model els va superar perquè, com va dir el mateix Codd, *"proporciona un mitjà per a descriure les dades amb la seua estructura únicament, sense haver de superposar cap estructura addicional per a representar-se en la màquina"* (és a dir, sense punters ni històries).

A pesar d'això va estar uns anys en competència amb aquells, amb molta gent molt reticent, ja que els productes comercials que eixien no eren suficientment eficients, potser perquè la tecnologia de l'època no ho permetia. A partir dels anys 80, quan la tecnologia ho va permetre, van eixir productes millors, com per exemple l'**ORACLE** (1979), i aleshores la seua implantació va ser aplastant.

Els objectius del Model Relacional són:

- **Fidelitat**, per a originar esquemes que representen fidelment la informació (els objectes i relacions entre ells) que existeix en el domini del problema.
- **Independència física**<sup>[1]</sup>, per a que la manera de guardar les dades no influeixi en la seua manipulació lògica, i així els usuaris que accedeixen a aquestes dades no hagen de modificar els seus programes per canvis en l'emmagatzematge físic.
- **Independència lògica**, per a que les vistes externes no es vegien afectades per canvis en l'esquema conceptual de la B.D.
- **Flexibilitat**, per a poder oferir les dades a cada usuari de la forma més adequada a la seua aplicació.
- **Uniformitat**, les estructures lògiques de les dades presenten un aspecte simple i uniforme (les taules), cosa que facilita la concepció i manipulació per part dels usuaris.
- **Senzillesa**, les característiques anteriors, unides a uns llenguatges d'usuari senzills, fan que el M. Relacional siga fàcil d'entendre i d'utilitzar per l'usuari final.

Recordeu que les **relacions** del Model Relacional són les **taules**. No són el mateix que les relacions del Model Entitat-Relació. Per evitar confusions intentarem anomenar-les sempre taules.

---

<sup>[1]</sup> En realitat tant la independència física com lògica les han intentades aconseguir tots els models.

## 2. Estructura del Model Relacional

L'element bàsic del Model Relacional és la **RELACIÓ**, que serà una taula o matriu bidimensional amb unes característiques o restriccions que comentarem més avant.

Dni	Nom	Adreca	Telefon	Sou	Data_n
18.876.543	Llopis Bernat, Jaume	C/ Artana, 3	964-213243	2.000	7-12-55
18.900.111	Garrido Vidal, Rosa	C/ Herrero, 54	964-253545	2.000	25-1-58
18.922.222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	694-216191	1.500	8-6-59
18.932.165	Folch Mestre, Pilar	C/ Palància, 22	964-234567	3.000	8-6-60
18.933.333	Peris Andreu, Joan	C/ Balmes, 3	964-223344	2.000	15-3-60
18.934.567	Sebastià Broch, Ferran	C/ Magallanes, 38	964-281706	2.500	14-7-62
18.944.444	Garcia Tomàs, Alícia	C/ Amunt, 15	964-205080	2.500	10-5-64

Normalment una relació té un **NOM** (p.e. **Empleat**) encara que ocasionalment no en tindrà, per exemple una taula que siga el resultat d'una consulta poc freqüent.

Les **FILES**, on tenim la informació de les ocurrences, dels individus, també s'anomenen **TUPLES** (de vegades per similitud amb fitxers també s'anomenen **REGISTRES**).

Les **COLUMNES**, que seran característiques que ens interessin dels individus i que en cada tupla agafa un valor, les anomenarem també **ATRIBUTS** (o **CAMPS**).

El conjunt de valors possibles que pot agafar un atribut determinat s'anomena **DOMINI**. Més avant veurem que els dominis s'intentaran definir el millor possible, per prevenir errors.

L'**ESQUEMA** o **ESTRUCTURA DE LA RELACIÓ** és la definició de la relació, és a dir, atributs que té, dominis d'aquests i restriccions que podrem definir, que veurem en la següent pregunta.

L'**ESTAT DE LA RELACIÓ** és la informació que conté en un determinat moment. Normalment l'estat variarà contínuament al llarg del temps, bé perquè s'afegeixen noves tuples (augmenta la cardinalitat), bé perquè es modifica el valor d'algun atribut en alguna tupla. En canvi l'esquema difícilment canviarà.

Una **CLAU CANDIDATA** és un atribut o conjunt d'atributs que identifiquen unívocament cada tupla de la relació. En l'exemple podrien ser claus candidates **Dni**, **Nom**, fins i tot ens podríem plantejar combinacions, com el conjunt (**Nom**, **Data\_n**), ja que sembla impossible que dues persones de l'empresa es diguen igual i damunt hagen nascut el mateix dia. De entre totes les claus candidates en triarem una, que serà la **CLAU PRINCIPAL** o **CLAU PRIMÀRIA**, i servirà per a identificar de forma efectiva en el Model cadascuna de les tuples.

Podria donar-se el cas que un atribut no agafe cap valor per a una tupla determinada, per exemple, un empleat que no tinga telèfon. Aleshores li donarem el **VALOR NUL**.

Per últim, les relacions o taules poden ser **PERMANENTS** o **TEMPORALS**. Les primeres es guarden. Les segones, normalment resultat d'una consulta ocasional, no.

Representarem la taula amb el nom de la taula en majúscules seguit, entre parèntesis, en minúscules i separats per comes, pels noms dels camps, amb la clau principal subratllada. També és convenient fugir dels caràcters especials (vocals accentuades, ç, ñ, guionet, ...) per no tenir

problemes quan anem a implementar-la en un SGBD determinat (Access, Oracle, PostgreSQL, ...). Per a una millor lectura intentarem posar sempre la clau principal al principi, el o els primers camps.

```
EMPLEAT (dni, nom, adreca, telefon, sou, data_n)
```

També podem utilitzar una forma alternativa de representar-la, amb un requadre que agafa tota la taula, dalt el nom de la taula, i baix cadascun dels camps, posant la clau principal en negreta o subratllada.

EMPLEAT
♦dni
°nom
°adreca
°telefon
°sou
°data_n

Igual que en altres models de dades, en el Model Relacional existeixen restriccions, és a dir, estructures o ocurrències no permeses.

Aquestes restriccions poden ser de dos tipus fonamentals: **restriccions inherents**, que són imposades pel propi model, i **restriccions d'usuari** (també anomenades restriccions **semàntiques**) en les quals és l'usuari qui prohibeix, perquè el model li ho permet, determinades circumstàncies per a poder definir millor la Base de Dades.

Com hem dit són les que imposa el propi model. Algunes són característiques que han d'acomplir les relacions. Per tant no qualsevol taula matemàtica és una relació. Podem considerar les següents:

- **Valors atòmics:** cada valor de la taula, és a dir, qualsevol valor de qualsevol atribut de qualsevol tupla ha de ser simple, no divisible. Per tant no valen atributs compostos o repetitius.

Així, si considerem **Nom** en la relació **Empleat** com a nom de pila més cognoms, no serà divisible (no podré agafar posteriorment el nom de pila per una banda i els cognoms per una altra; si ho volguera fer, s'haurien de definir els atributs simples **Cognom1**, **Cognom2** i **Nom**).

Tampoc valen valors repetitius, per exemple un vector de 12 entrades. Queden per tant descartats els atributs **multivaluats**.

- **Tuples distintes:** no poden haver dues tuples iguals. Això és una diferència respecte a les taules matemàtiques on sí que es poden duplicar files.
- **L'ordre de les tuples no és significatiu.**
- **L'ordre dels atributs no és significatiu.**

Les anteriors són condicions, impositcions que ens dona el mateix model.

Les realment interessants per a nosaltres són les restriccions d'usuari, també anomenades restriccions semàntiques, ja que seran condicions que podrem posar nosaltres per a que l'esquema de la B.D. explique el millor possible la realitat, i evitar possibles errors en les dades.



## 3.2.1 Restricció de domini

El valor d'un atribut ha de ser un valor atòmic del domini. Definint clarament el domini ens assegurem (dins del possible) que l'atribut no pugui agafar valors incorrectes.

Per una banda, el domini serà d'un tipus determinat, triat d'una gamma prou extensa: enter curt, enter, enter llarg, real, doble precisió, caràcter, cadena de caràcters (text), data, hora, ...

Així, per exemple, definint el *Sou* com un número real impedirem que per error pugui agafar el valor 2.R00'00, o que la *data de naixement*, amb domini de tipus data, siga 15-14-1958 o 31-2-1958.

També es podran definir dominis que estiguen en un determinat interval (nota d'un examen: 0-10) o d'un tipus enumerat (nota d'avaluació: MD, IN, SUF, BE, NOT, EXC).

Es poden definir regles de verificació o validació (**CHECK**) que ens ajuden a perfilar molt bé el domini. Aquest seria un exemple d'exigir que el treballador tinga de 18 a 65 anys. La sintaxi és inventada, només per a entendre-ho, amb una hipotètica funció que trau l'any, i una altra que ens dona la data d'avui.

```
CHECK ( Any(avui - data_n) >= 18) and (Any(avui - data_n) < 65) )
```

### Nota

En **Access** la regla de validació seria:

```
AgregFecha("aaaa";18;[data_n])<=Ahora() Y AgregFecha("aaaa";65;[data_n])>=Ahora()
```

on es veu que la fórmula és afegir 18 anys a la data de naixement i comprovar que no supera la data d'avui; i afegir 65 anys a la data d'avui i comprovar que sí supera la data d'avui.

Per exemple, **Empleat** podria quedar:

```
EMPLEAT (dni:caràcter(10), nom:caràcter(30), adreca:caràcter(30), telefon:enter(9), sou:numèric(6,2), data_n:data)
```

amb la regla de validació:

```
CHECK Sou > 0
```

a banda de la mencionada anteriorment.

Si utilitzem la manera de representar alternativa, ho tindríem així:

EMPLEAT	
•dni	caràcter(10)
°nom	caràcter(30)
°adreca	caràcter(30)
°telefon	enter(9)
°sou	numèric(6,2)
sou > 0	
°data_n	data

La següent imatge mostra com es faria en **Access** la primera regla de validació, aplicada al camp **data\_n**

Base de datos1 : Base de datos (Access 2007) - Micr...

Herramientas de tabla

Inicio Crear Datos externos Herramientas de base de datos Diseño

Ver Vistas Clave principal Generador de reglas de validación Herramientas Insertar filas Eliminar filas Columna de búsqueda Hoja de propiedades Índices Mostrar u ocultar

Advertencia de seguridad Se ha deshabilitado parte del contenido de la base de datos Opciones...

Todas las ta... << Empleat

Empleat Empleat : Tabla

Nombre del campo	Tipo de datos	Descripción
dni	Texto	
nom	Texto	
adrec	Texto	
telefon	Número	
sou	Número	
data_n	Fecha/Hora	

Propiedades del campo

General Búsqueda

Formato	
Máscara de entrada	
Título	
Valor predeterminado	
Regla de validación	AgregFecha("aaaa";18;[data_n]) <= Ahora() Y AgregFecha("aaaa";65;[data_n]) >= Ahora()
Texto de validación	
Requerido	No
Indexado	No
Modo IME	Sin Controles
Modo de oraciones IME	Nada
Etiquetas inteligentes	
Alineación del texto	General
Mostrar el Selector de fecha	Para fechas

Un nombre de campo puede tener hasta 64 caracteres de longitud, incluyendo espacios. Presione F1 para obtener ayuda acerca de los nombres de campo.

Vista Diseño. F6 = Cambiar paneles. F1 = Ayuda. Bloq Num

### 3.2.2 Restricció de clau principal

Permet declarar un atribut o un conjunt d'atributs com a **CLAU PRINCIPAL** o **PRIMÀRIA** (Primary Key). Aquesta clau principal servirà per a identificar unívocament cadascuna de les files.

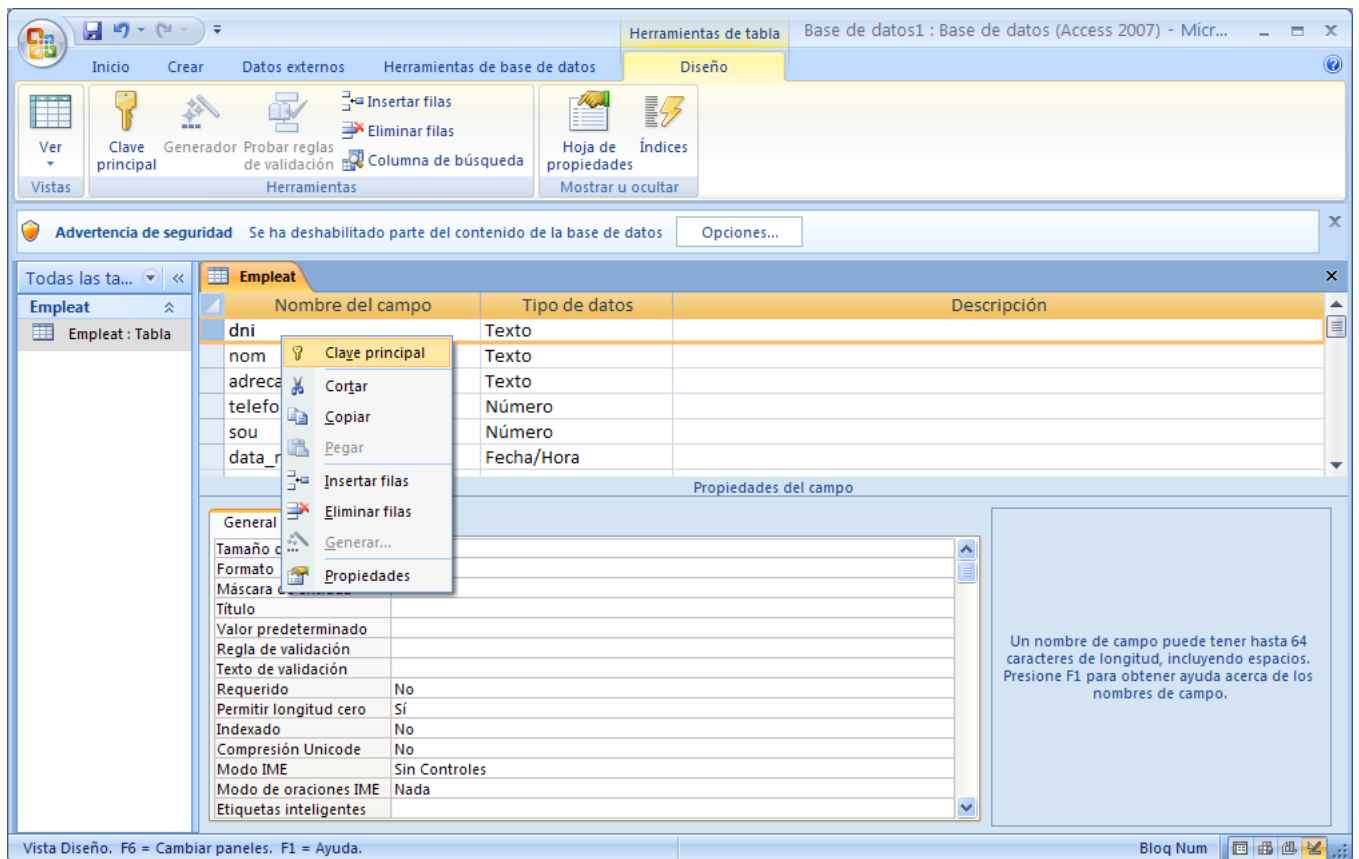
Com a conseqüència de l'anterior, la clau principal no podrà agafar valors nuls ni repetits, ja que en cas contrari no es podria assegurar la identificació de les files.

Aquestes últimes característiques també les podran tenir altres atributs, cosa que dóna lloc als tipus de restricció que es veuen en els següents punts.

Nosaltres sempre definirem una clau principal. Si tenim claus candidates, triem una d'elles com a clau principal. Si no en tenim, ens inventem un camp que servirà de clau principal (bé un número o bé un codi alfanumèric).

No és convenient que la clau principal estiga formada per un número excessiu de camps. Podríem dir que 3 és el màxim. Si la clau candidata està formada per més de 3 camps, o bé triem una altra clau candidata, o bé ens inventem una.

En **Access** es marca la clau principal amb una figura d'una clau. En la figura es veu com fer que el camp **dni** siga la clau principal de la taula **EMPLEAT**:



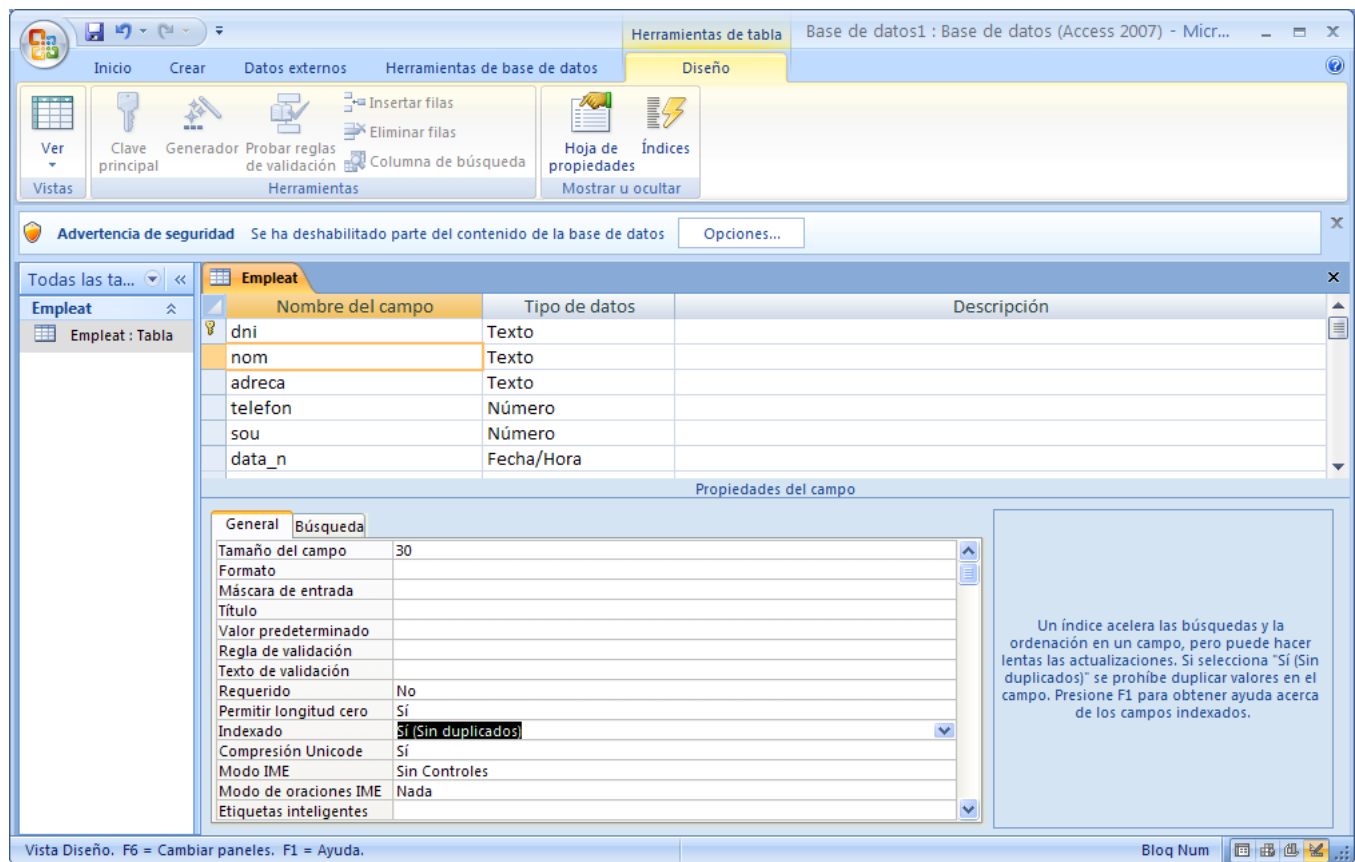
### 3.2.3 Restricció d'unicitat

Si en un camp, o en un conjunt de camps, definim la restricció d'unicitat (**UNIQUE**), això obliga a que, en cas de tenir valors el camp, no es puguin repetir. Suposem, per exemple, els alumnes d'un Institut. La clau no pot ser el DNI, ja que alguns alumnes no en tindran, però en cas de tenir-ne, és clar que no es podrà repetir.

Representarem que un camp és únic, posant **únic** entre parèntesi baix del camp. Per exemple, si considerem que el camp nom de la taula EMPLEAT ha de ser únic, ho representarem així:

```
EMPLEAT (dni, nom, adreca, telefon, sou, data_n)
         (únic)
```

En Access la restricció d'unicitat (**UNIQUE**) es defineix posant en l'apartat **Indexado** el valor **Sí (sin duplicados)**. En la figura es mostra com fer que el camp **nom** de la taula **EMPLEAT** siga únic.



### 3.2.4 Restricció de valor no nul

Obliga a que el camp agafe sempre un valor. Per exemple, el camp **Nom** és un bon candidat a ser no nul.

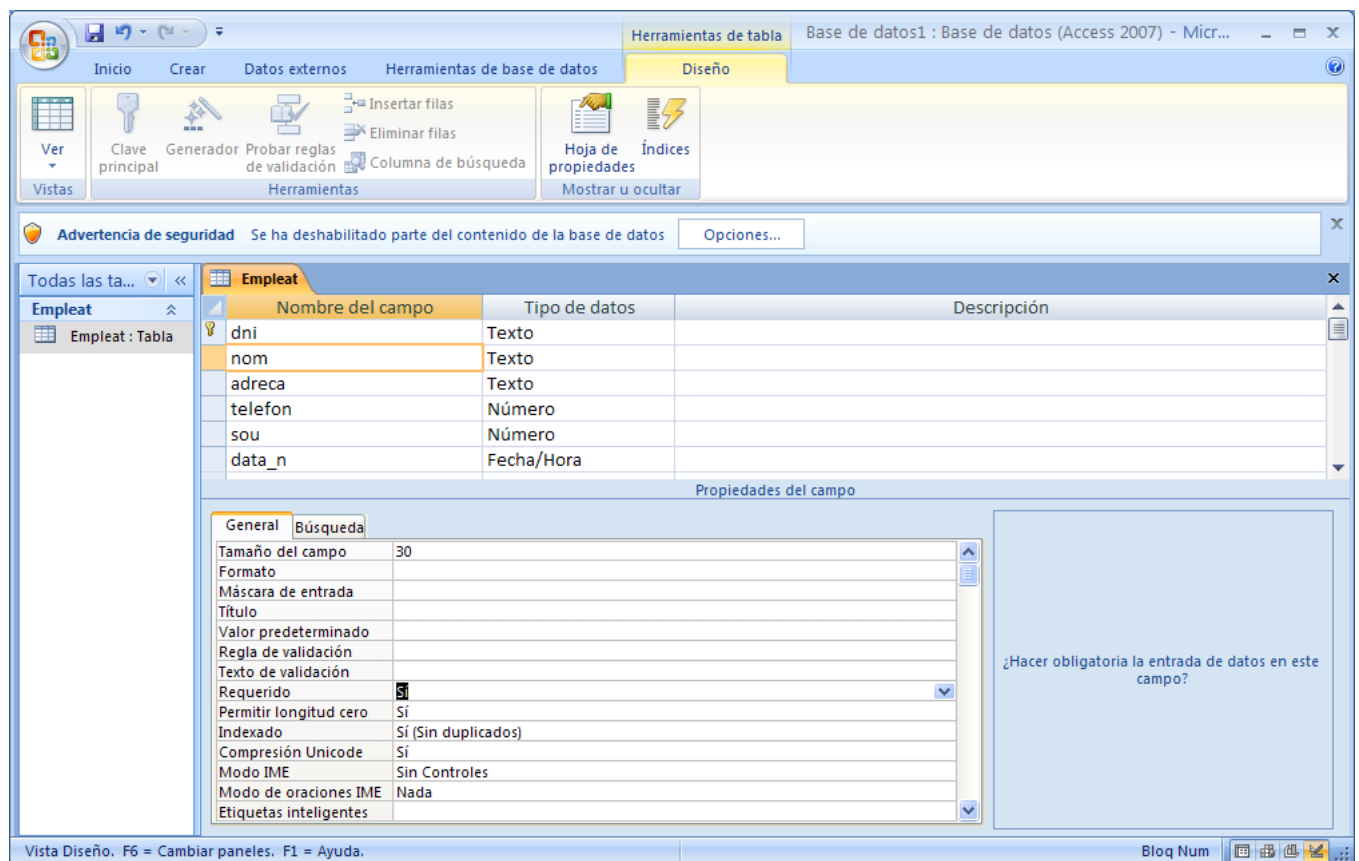
Ho representarem posant **no nul** entre parèntesi baix del camp.

```
EMPLEAT (dni, nom, adreca, telefon, sou, data_n)  
(no nul)
```

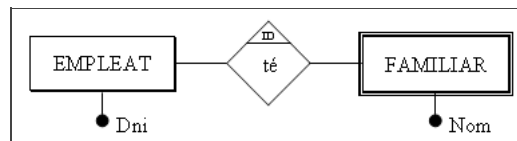
Per mig de la representació alternativa, podem marcar amb un punt negre davant del camp no nul.

EMPLEAT
*dni
*nom
*adreca
*telèfon
*sou
*data_n

En Access la manera de posar un camp com a **NOT NULL** serà posar en l'apartat **Requerido** el valor **Si**



Per poder explicar-la ens recolzarem en un exemple.



que es podria traduir en les següents taules:

EMPLEAT		FAMILIAR	
dni	nom	dni_emp	nom
18.876.543	Llopis Bernat, Jaume	18.876.543	Jaume Llopis Doménech
18.900.111	Garrido Vidal, Rosa	18.900.111	Felip Gomis Pitarch
18.922.222	Nebot Aliaga, Carme	18.932.165	Josep Serra González
18.932.165	Folch Mestre, Pilar	18.932.165	Xavier Serra Folch
18.933.333	Peris Andreu, Joan	18.933.333	Silvia Peris Prades
18.934.567	Sebastià Broch, Ferran	18.933.333	Silvia Prades Arnau
18.944.444	Garcia Tomàs, Alicia	18.944.444	Andreu Garcia Torró
		18.944.444	Laura Almela Garcia
		18.944.444	Lluís Almela Garcia
		18.944.444	Marc Almela Tomeu

Si en una taula R2 (**Familiar**) tenim un atribut (**dni\_emp**) que és clau (primària o candidata) d'una altra taula R1 (**Empleat** → **dni**), tot valor d'aquell atribut ha de concordar amb un valor de la clau de R1 (no he de poder posar en familiar un Dni que no el tinga cap empleat de l'empresa). L'atribut en R2 és, per tant, una **CLAU EXTERNA**.

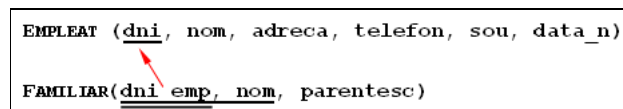
Ha de ser impossible posar en **Familiar** el Dni 18.754.321, perquè no està en l'altra, i per tant no és un Dni d'un empleat de l'empresa.

Les relacions R1 i R2 no tenen per què ser distintes, poden ser la mateixa. Així, si considerem el supervisor, aquest ha de ser de l'empresa:

EMPLEAT						
dni	nom	adreca	telefon	sou	data_n	supervisor
18.876.543	Llopis Bernat, Jaume	C/ Artana, 3	964.213243	2.000	7-12-55	18.933.333
18.900.111	Garrido Vidal, Rosa	C/ Herrero, 54	964.253545	2.000	25-1-58	18.932.165
18.922.222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	694.216191	1.500	8-6-59	18.944.444
18.932.165	Folch Mestre, Pilar	C/ Palància, 22	964.234567	3.000	8-6-60	
18.933.333	Peris Andreu, Joan	C/ Balmes, 3	964.223344	2.000	15-3-60	18.944.444
18.934.567	Sebastià Broch, Ferran	C/ Magallanes, 38	964.281706	2.500	14-7-62	
18.944.444	Garcia Tomàs, Alicia	C/ Amunt, 15	964.205080	2.500	10-5-64	18.933.333

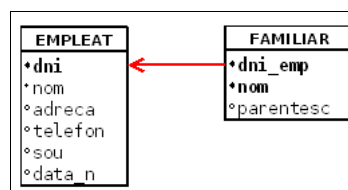
Supervisor és una clau externa, però de la mateixa taula. No tots els SGBD permeten una clau externa reflexiva.

Una manera de representar les claus externes en l'esquema és la següent:



On el doble subratllat indica una clau externa (que en aquest exemple, a més, forma part de la clau principal, però que no sempre serà així) que "apunta" a la clau principal de l'altra taula.

L'altra manera, utilitzant la forma alternativa, serà aquesta, que també és molt fàcil d'entendre:



Açò ens impedirà que introduïm valors no correctes, no existents. ¿Però què passarà si esborrem un empleat, o si modifiquem el seu Dni? ¿Què fem amb els familiars? Doncs en principi tres podrien ser les accions a realitzar, i depenent de la situació particular triaríem una o una altra:

- No deixar esborrar-lo o modificar-lo (**NO ACTION**).

Segurament no és l'opció més adequada per a l'exemple dels empleats i els familiars. Però pensem en un altre exemple, amb clients i factures. Què fem si s'esborra un client i tenim factures d'ell? Segurament el més adequat serà no fer l'acció, és a dir, no esborrar el client (sobretot si les factures estan pendents de cobrar ...).

- Esborrar també els familiars o canviar-los **en cascada (CASCADE)**.

Segurament aquesta és l'opció més adequada per al cas dels familiars, que s'eliminen automàticament.

- Canviar el valor de la clau externa al valor nul o un valor predeterminat (**SET NULL** o **SET DEFAULT**).

En l'exemple dels familiars no té sentit ja que no ens interessin els familiars dels que no són de l'empresa. Però imaginem, per exemple, un proveïdor que ens ha proporcionat uns articles. Pel fet de no treballar ja amb el proveïdor i llevar-lo de la B.D. no hauríem d'eliminar els articles. Seria suficient amb donar un valor nul al proveïdor d'aquest article<sup>[1]</sup>.

Hi ha SGBD que fins i tot permeten accions distintes per al cas d'esborrament i d'actualització de la clau, com per exemple Access.

En Access les restriccions d'integritat (claus externes) es defineixen amb les **relacions** entre taules. El següent vídeo mostra com crear la clau externa de Familiar cap a Empleat, i es pot observar com la manera de representar-lo és molt pareguda a la forma alternativa que estem utilitzant.

---

<sup>[1]</sup> Observem, però, que aquesta clau externa hauria d'admetre valors nuls. Si no ho permet, millor un valor per defecte.

A pesar de totes les restriccions anteriors, que normalment els SGBDR compleixen, hi ha d'altres que no es poden expressar per mig del Model Relacional, i per tant no poden complir directament els SGBDR. Són les **restriccions externes a l'esquema relacional**. Estaran normalment les heretades dels Model E/R, i hi hauran algunes més que sí que es podien expressar en el Model E/R, però no en el Model Relacional. Les veurem en la pregunta 4.

Els SGBDR més avançats, més potents, permetran un tractament a aquestes restriccions externes, consistents en executar un procediment definit per l'usuari després d'una actualització. Són els **Disparadors (TRIGGERS)**. Aquest concepte és molt potent, ja que dona una resposta procedimental on es pot fer qualsevol cosa.

En Access dues seran les ferramentes per a fer açò, les **Macros** i procediments en Visual Basic.



## 4. Transformació del M. E/R al M. Relacional

---

A continuació veurem les regles de transformació de l'esquema en el Model E/R al Model Relacional.

D'aquesta manera continuem el procés de disseny d'una Base de Dades. En el Tema 2 hem après a fer l'esquema en el Model Entitat-Relació. Ara el traduirem al Model Relacional, i ja es podrà implementar en qualsevol SGBD Relacional. Faltaria només el procés de Normalització (tema 4) per acabar de deixar les taules perfectament dissenyades. De tota manera les Bases de Dades que dissenyem nosaltres, amb el procés descrit anteriorment, tindran unes taules molt "normalitzades", sempre que dissenyem bé.

Tota entitat es transformarà en una taula, amb tots els seus atributs, que es consideraran com a simples. Es tria un (o un conjunt) com a clau principal, i el denotarem subratllant-lo. Les entitats dèbils les estudiarem millor més endavant.

En el nostre exemple, com teníem 4 entitats, ens eixiran de moment 4 taules:

<b>EMPLEAT</b> ( <u>dni</u> , nom, adreca, telefon, sou, data_n)
<b>DEPARTAMENT</b> ( <u>num_d</u> , nom_d)
<b>PROJECTE</b> ( <u>num_p</u> , nom_p)
<b>FAMILIAR</b> ( <u>nom</u> , data_n, parentesc)

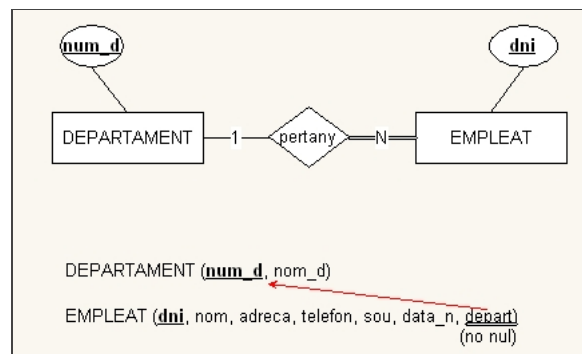
No considerarem els atributs multivaluats. Els tractarem i solucionarem en el tema següent, el de **Normalització**.

Per cada relació 1:N entre les entitats **A** i **B**, on **A** és la que participa amb grau de cardinalitat 1, i **B** amb grau N, s'inclou un nou camp en **B** (del mateix tipus que la clau principal de **A**) que a més serà **clau externa** que apuntarà a **A**, més concretament a la clau principal de **A**. En moltes ocasions al camp nou de **B** se li posa el mateix nom que a la clau principal de **A**, però no és necessari, depèn del gust de cadascú.

També s'inclouran en **B** tots els possibles atributs de la relació.

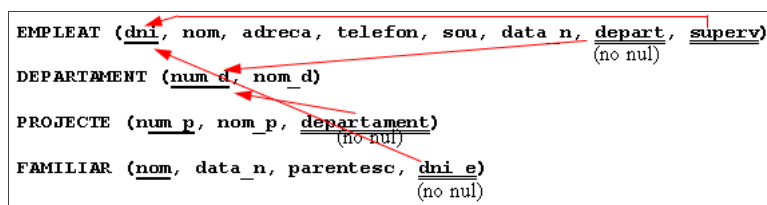
La següent animació intenta explicar-ho millor:

Si a més l'entitat que participa amb grau N ho fa de forma **total** (com en la figura de baix), la clau externa **no pot ser nula** (és a dir sempre ha de tenir un valor).



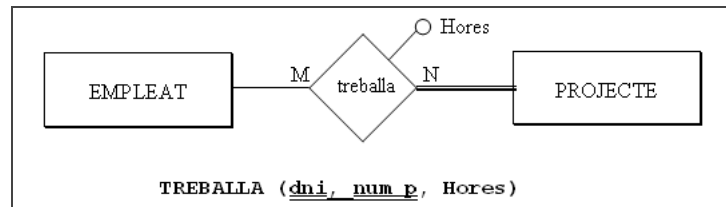
En l'exemple nostre:

- Per la relació **Pertany** (figura de dalt) inclourem l'atribut **departament** a **Empleat**, que a més haurà de ser no nul.
- Per la relació **Controla** inclourem l'atribut **departament** a **Projecte** (no nul).
- Per la relació **Supervisa** inclourem l'atribut **supervisor** a **Empleat** (és reflexiva), però aquest sí que pot ser nul. Encara que sembla estrany, un camp pot ser clau externa que apunta a la clau principal de la mateixa taula.
- Per la relació **Té** entre empleat i familiar, inclourem en **FAMILIAR** l'atribut **dni\_e**, però com que Familiar és dèbil la veurem millor un poc més endavant.



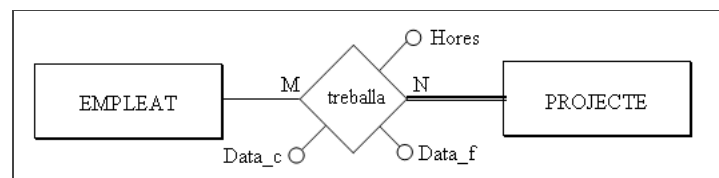
Per cada relació M:N construirem una nova taula on s'inclouran com a clau externa les claus principals de les dues entitats, i a més la seua combinació constituirà (o formarà part de) la clau principal. Inclourem també els possibles atributs de la relació.

En l'exemple:



Cada vegada que ens trobem amb una relació M:N i la traduïm per una nova taula haurem de **preguntar-nos si és suficient amb la clau principal** formada per les dues claus externes, o si fa falta afegir un altre camp. Hem de ser conscients que la clau principal no puga repetir-se, que hi haja 2 files amb la mateixa clau principal. Així, en l'exemple, ens hauríem de fer la següent pregunta: pot un empleat treballar en el mateix projecte més d'una vegada? En aquest cas la contestació és negativa, i per tant és suficient amb aquesta clau principal.

Però suposem que la resposta és que sí que pot treballar més d'una vegada al llarg del temps. En aquest cas seria una espècie d'històric, on faria falta, a més, saber quan comença i quan acaba de treballar en un projecte un determinat treballador (serien atributs de la relació):



Aleshores, com no és suficient amb la clau principal formada per les dues claus externes, inclourem un altre camp en la clau principal. Sembla que el més adequat seria **Data\_c** (ja no es pot donar el cas que el mateix treballador treballi més d'una vegada en el mateix projecte, començant el mateix dia)

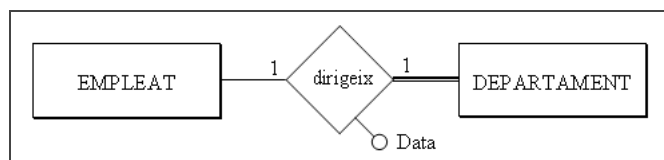
**TREBALLA (dni, num\_p, data\_c, data\_f, hores)**

Però per una altra banda, les claus principals molt llargues no són operatives, i encara que la traducció literal siga com hem dit, per motius pràctics podem canviar la clau principal. Considerarem que el nombre màxim de camps en la clau principal és de 3. 4 ja són massa, i aleshores buscarem una altra clau principal (un codi de treball, per exemple). Les claus externes continuarien sent-ho.

No hi ha una forma única de traduir aquestes relacions. Tres seran les possibles traduccions, segons la participació total o parcial de les entitats en la relació, i també segons el que ens diga el "sentit comú".

- Si de les dues entitats que entren la relació, **A** i **B**, una d'elles i només una, participa de forma total, per exemple **B**, traduirem la relació 1:1 com una **clau externa** en la taula corresponent a l'entitat que participa de forma total (**B**). Podem obligar també a que aquest camp que serà clau externa siga **no nul** (ja que totes les ocurrences de **B** entren en la relació). També podem fer que aquest camp siga **únic** (no es podrà repetir, ja que si es poguera repetir seria una relació 1:N). A més, inclourem en **B** tots els possibles atributs de la relació.

Per exemple, la relació **dirigeix**, que és 1:1 entre **EMPLEAT** i **DEPARTAMENT**:

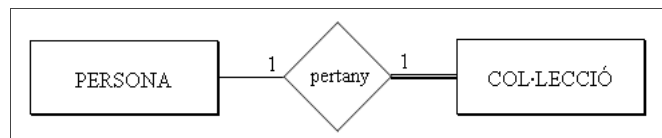


Com que l'entitat de la dreta participa de forma total, triarem **DEPARTAMENT**:

<b>DEPARTAMENT</b> ( <u>num_d</u> , nom_d, <u>director</u> , Data) <span style="margin-left: 150px;">(no nul, únic)</span>
---

Ho fem d'aquesta manera perquè tots els departaments tenen director, però no tots els empleats són directors. Si posàrem la clau externa en la taula **EMPLEAT** (s'anomenaria per exemple **dep\_que\_dirigeix**) moltes vegades estaria buit, ja que relativament són pocs els empleats que dirigeixen un departament.

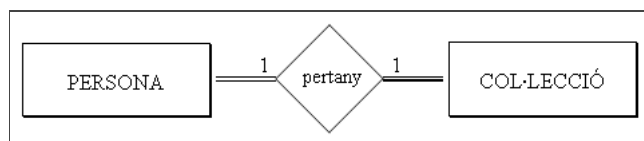
Vegem un altre exemple de relació 1:1, el de les papallones. Teniem una relació 1:1 entre **PERSONA** i **COL·LECCIÓ**



Triaríem **COL·LECCIÓ**, ja que entra de forma plena o total en la relació.

<b>COL·LECCIÓ</b> ( <u>cod_col</u> , ... (dades col·lecció)... , <u>dni-prop</u> ) <span style="margin-left: 150px;">(no nul, únic)</span>
---

- Si les dues entitats participen de forma total, es pot considerar tot (les dues entitats i la relació, amb els seus possibles atributs) com una sola taula. En la pràctica això serà prou estrany, perquè ja ho hauríem considerat una sola entitat. Per exemple, considerem que totes les persones que estudiem tenen una col·lecció:



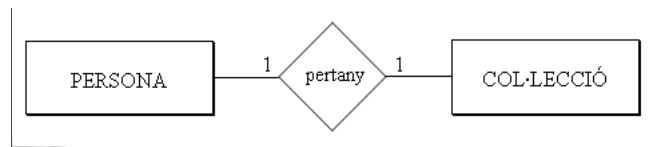
Aleshores podríem considerar una única taula, que continga les dades de la persona i de la seua col·lecció:

<b>PERSONA</b> ( <u>Dni</u> , ... (dades pers)... , ... (dades col·lecció)... )
---

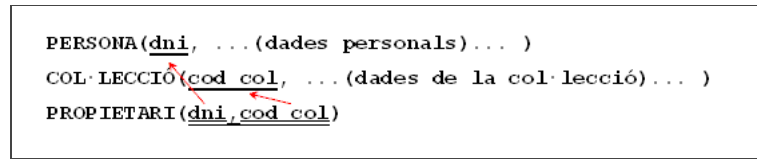
De tota manera, pot ser ens interesse (per separar clarament els dos tipus d'informació) dues taules. Aleshores podríem traduir-lo com en el primer punt, amb una clau externa (no nula) en una de les dues taules, i hauríem de triar la taula que *menys* s'utilitze.

- Si les dues entitats participen de forma parcial, com que si posem una clau externa en una de les dues, moltes vegades tindrà valor nul, podem traduir-la com una nova taula que marque la relació, on hi haurà una tupla per cada relació entre dues ocurrences. Inclouríem en la nova taula els possibles atributs de la relació.

En l'exemple podria ser que les col·leccions pertanyen a una persona particular o a una institució (no tenen propietari):



Quedaria:



Però com comentàvem al principi, haurem d'aplicar el sentit comú, ja que potser una de les dues podria participar de forma "quasi" total (per exemple, quasi totes les col·leccions són d'una persona). Aleshores podria ser millor traduir-lo com en el primer cas, posant la clau externa en la que participa de forma "quasi" total, ja que aquesta tindrà relativament pocs valors nuls, i seria més costós mantenir una altra taula. Evidentment la clau externa sí que podria ser nula, en aquest cas.

Resumint, **una relació 1:1 quasi sempre la traduirem com una clau externa en la taula que participa en la relació de forma total o quasi total** (o la que previsiblement té més ocurrences en la relació)

Les entitats dèbils, com que com a mínim depenen d'una altra, podem ser més restrictius que les altres. Una entitat sempre és dèbil a través, com a mínim, d'una relació que la comunica amb l'entitat principal. A més participa de forma total (com que no pot existir sense l'altra, tota ocurrència està en la relació).

Per tant, tota entitat dèbil tindrà una clau externa, que apunta a la principal i serà no nula. Però encara podem anar més enllà:

- **Dependència en existència:** farem que la clau externa **esborre i actualitze en cascada**, ja que si deixa d'existir la principal no té sentit la dèbil.
- **Dependència en identificació:** a més d'esborrar i actualitzar en cascada, la clau externa **formarà part de la clau principal**.

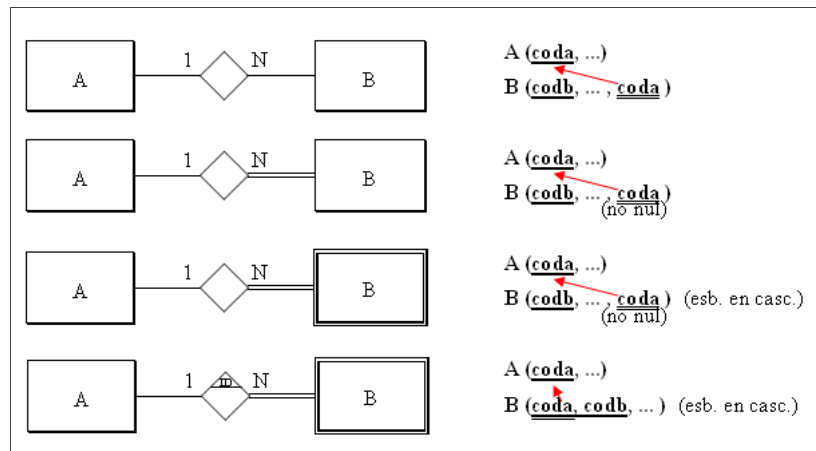
Si la relació per la qual depèn en identificació és 1:N, farà falta un altre camp en la clau principal.

Si és 1:1, amb la clau externa és suficient com a clau principal

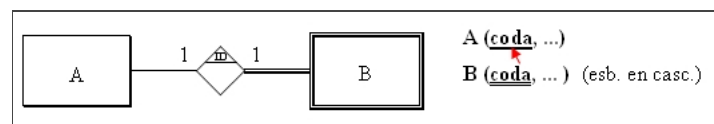
En nostre exemple quedarà així:

<b>FAMILIAR</b> ( <u><b>dni_e, nom_f</b></u> , data_n, parentesc) (esborrar en cascada)
---

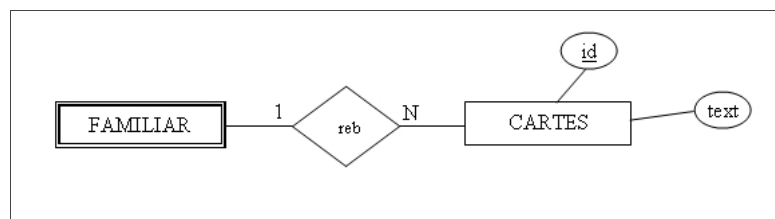
Anem a fer un quadre resum amb distints graus de dependència entre dues relacions i com es traduiria al Model Relacional:



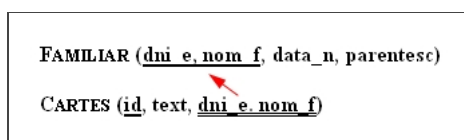
Per una altra banda, si ens trobem una entitat dèbil que depèn en identificació a través d'una relació 1:1, és suficient amb la clau principal de A com a clau principal de B



Una altra qüestió que pot dur a confusió és el cas de les claus externes formades per 2 camps. Ens basarem en l'exemple de sempre, en la taula FAMILIAR, ja que té una clau principal formada per 2 camps. Suposem que hi ha en el Model Entitat-Relació una taula que depèn d'ella, com podria ser comunicacions que se li han fet (cartes). Les entitats i la relació entre elles podria ser aquesta:



Com que la relació és de tipus 1:N la traduirem per una clau externa en CARTES. I quina serà la clau externa? Com que la taula FAMILIAR té una clau principal formada per 2 camps, haurem de posar una clau externa formada per 2 camps. Alerta! no seran 2 claus externes, sinó una clau externa formada per 2 camps. Com sempre, representarem la clau externa amb un doble subratllat, que ara agafarà als 2 camps.



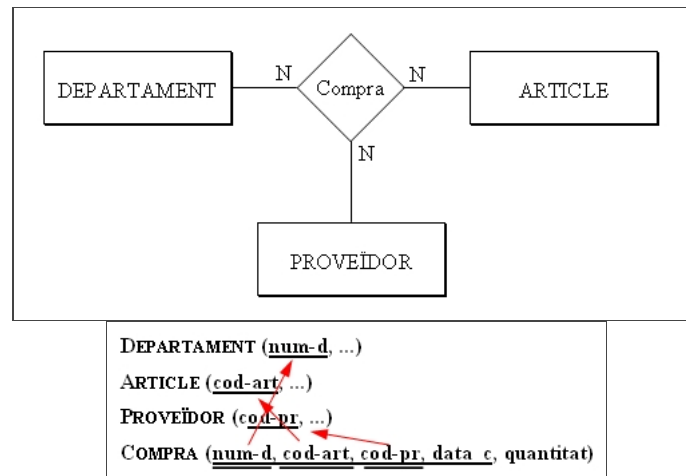


En una relació ternària o superior construïrem una nova taula, on inclourem com a claus externes les claus primàries de totes les entitats, i a més els atributs de la relació.

Habitualment, la clau principal de la nova taula serà la combinació de totes les claus principals de les entitats. Ocasionalment, si alguna entitat participa amb cardinalitat 1, la clau principal d'aquesta entitat no entraria a formar part de la clau principal de la nova taula.

Igual que en el cas de les relacions M:N, ens hauré de preguntar si és suficient amb la clau primària generada o si s'haurà d'incloure algun altre camp.

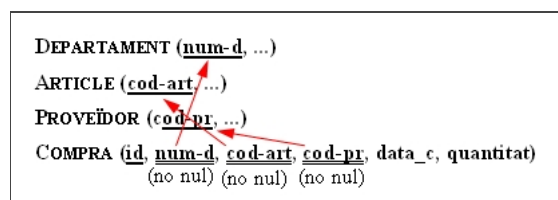
En l'exemple que vam posar de relació ternària, suposant els atributs de la relació la data de compra i la quantitat:



On hem posat totes les claus externes formant part de la clau principal, ja que totes entren amb cardinalitat N. Però no teníem prou amb aquesta clau principal, ja que el mateix departament pot comprar el mateix article al mateix proveïdor més d'una vegada. Com no teníem prou, hem posat un altre camp.

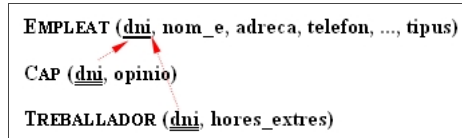
Seria moment, segurament, de substituir la clau principal que està formada per 4 camps, ja que en són massa.

Posaríem una altra clau principal, però les claus externes continuarien sent-ho, i a més serien no nules:



Aquest aspecte, com en el cas de les relacions 1:1, també té múltiples solucions. El problema de les especialitzacions és que, encara que de forma teòrica la solució siga correcta, en la pràctica suposa moltes taules i amb un cert grau de manteniment entre elles. Per tant, i aplicant el sentit comú, moltes vegades es fa una simplificació, llevant bé les subclasses, bé la superclasse, com veurem a continuació. Aquestes són les possibilitats:

- Transformar les especialitzacions en taules amb la clau principal heretada i els atributs específics (com si substituïrem l'especialització en relacions 1:1, amb les subclasses depenent en identificació de la superclasse). Estaria també bé afegir un atribut a la superclasse per a poder saber de quin tipus és. En l'exemple ens quedaria:



- Simplificar les subclasses. Aleshores tots els atributs d'aquestes haurien de passar a la superclasse. També s'haurien de passar totes les relacions que afecten a les subclasses, "retocant" les participacions totals (que ara ja no ho serien). Ara serà obligatori tenir el camp que distingeix el tipus (sinó, perdríem aquesta informació).

```
EMPLEAT (dni, nom_e, adreca, telefon, ..., tipus, opinio, hores_extres)
```

- Simplificar la superclasse. Aleshores tots els atributs d'aquesta passarien a cadascuna de les subclasses. També passarien les relacions a cadascuna d'aquestes, "retocant" les participacions totals.

```
CAP (dni, nom_e, adreca, telefon, ..., opinio)
TREBALLADOR (dni, nom_e, adreca, telefon, ..., hores_extres)
```

Ja hem comentat que restriccions externes són restriccions que no es poden expressar per mig del model de dades. Aleshores les expressarem de paraula.

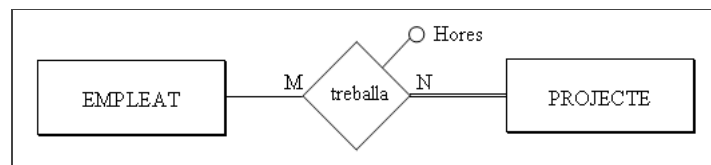
Normalment, les restriccions externes del Model E/R continuaran sent-ho en el Model Relacional, perquè tampoc es podran expressar. En el nostre exemple teníem les restriccions externes del Model E/R:

**Rex1:** El cap d'un departament ha de ser membre d'aquest.

**Rex2:** Un empleat només pot treballar en projectes coordinats pel seu departament.

Açò tampoc es pot expressar amb el Model Relacional, per tant les mantindrem.

A banda, es poden crear més restriccions externes, perquè amb el Model Relacional no es pot expressar tot el que es podia expressar amb el Model E/R. Per exemple, en la relació:



ja hem comentat que donarà lloc, a banda de les taules de les entitats, a una altra taula:

**TREBALLA** (dni, num\_p, hores)

on **dni** i **num\_p** són claus externes. Però l'entitat projecte participa de forma total en la relació, és a dir, en tot projecte ha d'haver un empleat treballant. ¿Com ho controlem això? Doncs és una nova restricció externa que la podríem formular així:

**RexR3:** Tot projecte ha de tenir com a mínim un empleat treballant-hi.

Les participacions totals que ens suposaran una restricció externa són:

- En una relació **1:N**, una participació total de l'entitat que participa amb grau 1.
- En una relació **M:N**, qualsevol participació total (si les dues participen de forma total, aleshores hi haurà dues restriccions externes).
- En una relació **1:1**, depèn de la manera de traduir-se.

La manera d'implementar les restriccions externes serà per mig d'un TRIGGER, que s'active quan hi haja una actualització (inserció, modificació o esborrat) que afecte a la restricció externa. Per exemple, en la restricció **RexR3** els moments importants són després d'inserir un nou projecte, i abans d'eliminar o modificar en la taula **Treballa** (per si un projecte es queda sense gent treballant en ell).

Les accions a desenvolupar podrien ser traure un avís, o obligar a inserir com a mínim una tupla en la taula **Treballa**, en el cas de la inserció d'un nou projecte; en el cas de modificació o eliminació en **Treballa** podria impedir-se aquesta actualització.

En el nostre exemple tindrem les restriccions externes al Model Relacional:

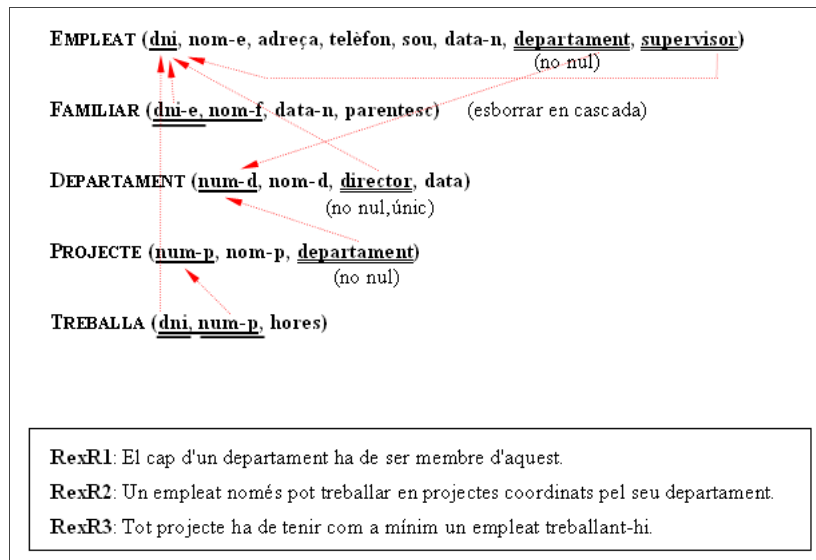
**RexR1:** El cap d'un departament ha de ser membre d'aquest.

**RexR2:** Un empleat només pot treballar en projectes coordinats pel seu departament.

**RexR3:** Tot projecte ha de tenir com a mínim un empleat treballant-hi.

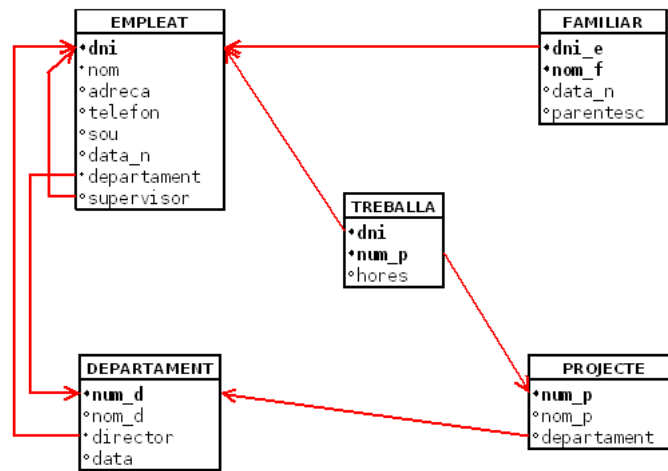
Anem a veure com quedarà definitivament la traducció de l'exemple que estem arrastrant des del Tema 2. Donarem 2 versions, tenint en compte o no l'especialització de que el treballador pot ser cap o treballador normal.

Sense tenir en compte l'especialització tindrem aquesta solució:



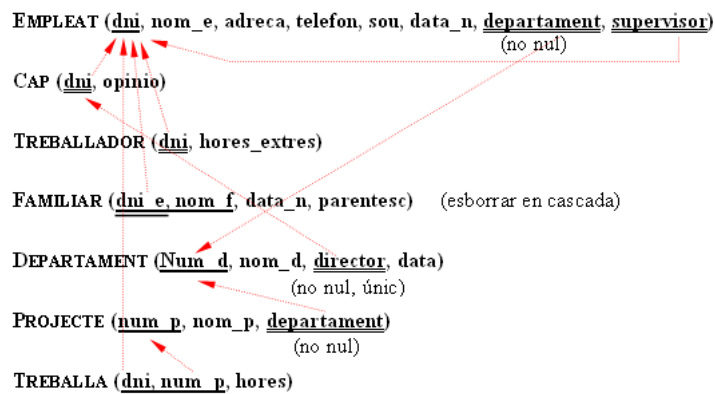
aquest vídeo explica tot el procés pas a pas:

I aquesta seria la forma alternativa de representar-lo:



- RexR1: El cap d'un departament ha de ser membre d'aquest
- RexR2: Un empleat només pot treballar en projectes coordinats pel seu departament
- RexR3: Tot projecte ha de tenir com a mínim un empleat treballant-hi

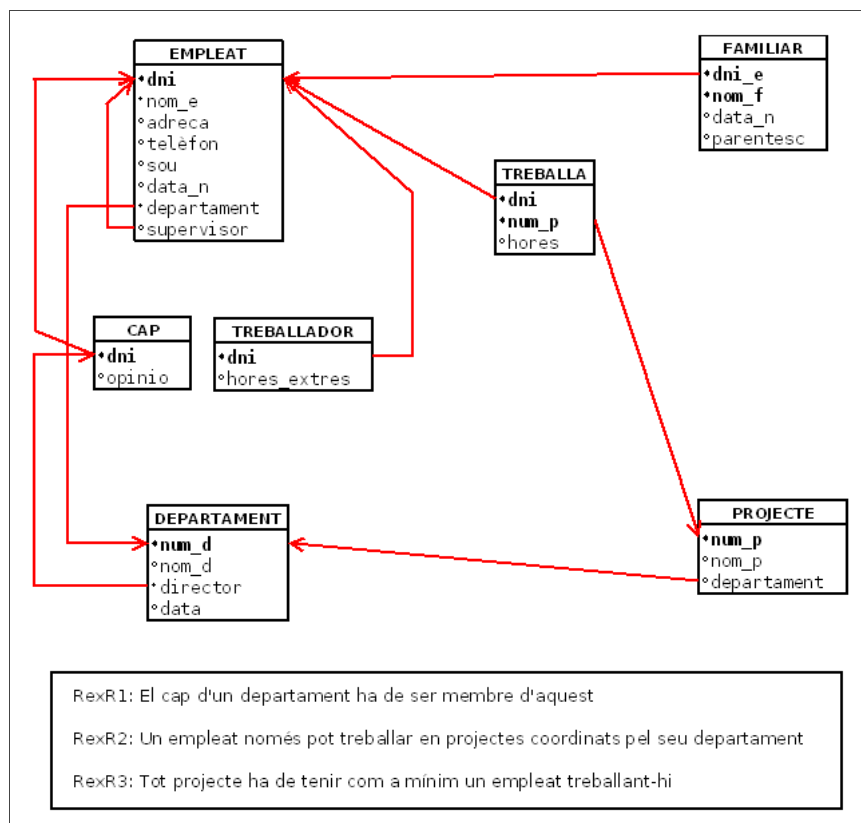
I tenint en compte l'especialització:



- RexR1: El cap d'un departament ha de ser membre d'aquest.
- RexR2: Un empleat només pot treballar en projectes coordinats pel seu departament.
- RexR3: Tot projecte ha de tenir com a mínim un empleat treballant-hi.

que aquest vídeo explica en els punts diferents a l'anterior solució

I ací tindríem la representació alternativa:



## 5. Llenguatges relacionals

Fins el moment hem dissenyat una B.D. Relacional. Però si volem la B.D. és per consultar-la, treure la informació que ens interessa, manipular-la. Ens farà falta, per tant, un llenguatge de manipulació de la B.D. (**DML**). Aquests llenguatges poden estar basats en l'Àlgebra Relacional o en el Càlcul Relacional.

- Per mig de l'**ÀLGEBRA RELACIONAL** farem operacions sobre les taules, combinant-les, seleccionant el que ens importa, ..., en definitiva manipulant-les. El resultat serà una nova taula, que serà el resultat final o servirà per a fer una altra operació. Les operacions poden ser **projecció** (agafar algunes columnes d'una taula), **selecció** (seleccionar algunes files d'una taula, les que aconsegueixen una determinada condició), **unió**, **intersecció**, **producte cartesià**, **reunió** ...

El llenguatge **SQL**, que és l'estàndard de fet i que veurem en el tema 6, està basat en l'Àlgebra relacional. Les sentències són de la forma:

```
SELECT dni, nom, sou
FROM EMPLEAT
WHERE sou > 2000
```

on estem projectant sobre els camps dni, nom i sou, i seleccionant les files que aconsegueixen la condició del final

Anem a comentar dues de les operacions esmentades abans. El **producte cartesià** de dues taules consisteix en combinar cadascuna de les files d'una taula amb cadascuna de les files de l'altra. Així, el producte cartesià de **Empleat** i **Departament** seria:

```
SELECT dni, nom, nom_d
FROM EMPLEAT , DEPARTAMENT
```

Però aquest operació no sembla tenir molt de sentit en aquest cas ¿per a què volem combinar un empleat amb tots els departaments de l'empresa? Sembla molt més lògic combinar cada empleat únicament amb el departament al qual pertany.

La **reunió** de dues taules consisteix en fer un producte cartesià i després seleccionar les files que tenen el mateix valor en dos camps determinats (un de cada taula).

```
SELECT dni, nom, nom_d
FROM EMPLEAT , DEPARTAMENT
WHERE EMPLEAT.departament = DEPARTAMENT.num_d
```

És a dir, del producte cartesià seleccionem només les files en les quals coincideixen els camp departament i num\_d, combinant cada empleat amb el seu departament.

- En el **CÀLCUL RELACIONAL**, es defineixen variables de tipus **taula**, s'utilitzen operadors entre les variables, i també uns quantificadors (**per a tot** i **existeix**). Va de forma paral·lela a l'àlgebra de manera que es poden obtenir les mateixes coses amb l'Àlgebra i amb el Càlcul.

El **QBE** (Query By Example) es basa en el càlcul relacional, i la seua particularitat és la senzillesa de fer consultes per als no experts. Per mig d'una plantilla podem col·locar els atributs que volem visualitzar, l'ordre, els criteris de selecció, etc.

Campo:	Dni	Nom	Sou	
Tabla:	Empleat	Empleat	Empleat	
Orden:		Ascendente		
Mostrar:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criterios:			>2000	
o:				

## ? Omplir els espais en blanc

Intenta omplir els espais en blanc dels següents enunciats:

L'element bàsic del model relacional és la , que no és més que una  bidimensional. Cadascuna està formada per files, també anomenades , i columnes, també anomenades .

## ? Elecció múltiple

El conjunt de valors possibles que pot agafar una columna s'anomena:

- ☐ univers  
☐ domini  
☐ camp  
☐ cap de les anteriors

Nooooooo

Molt bé

Nooooooo

Nooooooo

### Solució

1. **Incorrecte** (Realimentació)
2. **Opció correcta** (Realimentació)
3. **Incorrecte** (Realimentació)
4. **Incorrecte** (Realimentació)

## ? Veritat / fals

Indica si les següents afirmacions són correctes

Els valors de les columnes han de ser multivaluats

**Pista**

☐ Veritat ☐ Mentida

### Mentida

L'ordre de les files no és significatiu

**Pista**

☐ Veritat ☐ Mentida

### Veritat

L'ordre de les columnes sí que és significatiu, ja que suposa canviar l'estructura

**Pista**

☐ Veritat ☐ Mentida

### Mentida



## ? Elecció múltiple

Quina afirmació és la correcta?

- ☐ La clau principal només pot estar formada per un camp, que no se repetirà ni agafarà valors nuls.
- ☐ La clau principal pot estar formada per un camp o més d'un camp. En aquest últim cas algun camp pot agafar el valor nul, però no tots a l'hora.
- ☐ La clau principal està formada per unes quantes claus candidates, agafades utilitzant el "sentit comú"
- ☐ Cap de les anteriors

Incorrecte

Incorrecte

Incorrecte

Correcte

### Solució

1. Incorrecte (Realimentació)
2. Incorrecte (Realimentació)
3. Incorrecte (Realimentació)
4. Opció correcta (Realimentació)

El valor nul és equivalent a:

- ☐ el zero
- ☐ l'espai en blanc
- ☐ qualsevol de les anteriors
- ☐ cap de les anteriors

Incorrecte

Incorrecte

Incorrecte

Correcte !!

### Solució

1. Incorrecte (Realimentació)
2. Incorrecte (Realimentació)
3. Incorrecte (Realimentació)
4. Opció correcta (Realimentació)

El concepte d'integritat referencial s'aplica a través de la ...

- ☐ clau principal
- ☐ clau candidata
- ☐ clau externa
- ☐ cap de les anteriors

Incorrecte

Incorrecte

Correcte !!

Incorrecte

### Solució

1. Incorrecte (Realimentació)
2. Incorrecte (Realimentació)
3. Opció correcta (Realimentació)
4. Incorrecte (Realimentació)

## ? Com traduiries del Model E/R al Relacional

Com traduiries una entitat?

- ☐ com una nova taula
- ☐ com un camp
- ☐ depén de l'entitat
- ☐ com una clau externa

Correcte !!

Incorrecte

Incorrecte

Incorrecte

### Solució

1. Opció correcta (Realimentació)
2. Incorrecte (Realimentació)
3. Incorrecte (Realimentació)
4. Incorrecte (Realimentació)

I com traduiries una relació 1:N ?

- ☐ com una nova taula
- ☐ com un camp
- ☐ depén de la relació
- ☐ com una clau externa

Incorrecte

Incorrecte

Incorrecte

Correcte !!

### Solució

1. Incorrecte (Realimentació)
2. Incorrecte (Realimentació)
3. Incorrecte (Realimentació)
4. Opció correcta (Realimentació)

I com traduiries una relació M:N

- ☐ com una nova taula
- ☐ com un camp
- ☐ depén de la relació
- ☐ com una clau externa

Correcte !!

Incorrecte

Incorrecte

Incorrecte

### Solució

1. Opció correcta (Realimentació)
2. Incorrecte (Realimentació)
3. Incorrecte (Realimentació)
4. Incorrecte (Realimentació)

