

Unitat 2: Disseny conceptual: Model E-R

- 1. Introducció
- 2. Diseny d'una BD
 - 2.1. Fase d'Anàlisi: Especificació de requisits Programari (E.R.S.)
 - 2.2. Fase del Disseny Conceptual: Model Entitat-Relació (MER)
 - 2.3. Fase del Diseny Lògic: Model Relacional
 - 2.4. Fase del Disseny Físic: Model Físic
- 3. Model Entitat-Relació
 - 3.1. Entitats
 - 3.2. Atributs
 - Tipus d'atributs
 - 3.3. Relacions
 - Grau de les relacions
 - Cardinalitat de les relacions
 - 3.4. Claus
- 4. Model Entitat-Relació Estés (MEER)
 - 4.1. Quan cal utilitzar relacions de jerarquia?
 - 4.2. Tipus de relacions de Jerarquia.
 - 4.3. Exemples.



Unión Europea

Fondo Social Europeo

El FSE invierte en tu futuro

Fecha	Versión	Descripción
31/08/2022	1.0.0	Versión inicial

Unitat 2: Disseny conceptual: Model E-R

1. Introducció

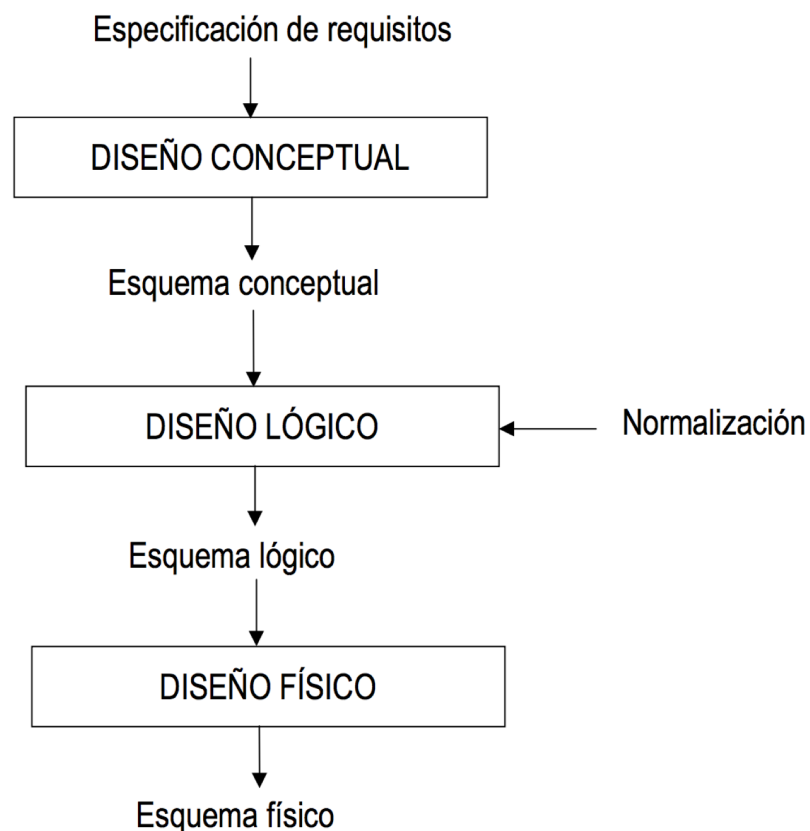
En aquest tema veurem com fer el disseny conceptual i lògic d'una base de dades. Començarem elaborant el model conceptual usant diagrames Entitat-Relació (MER) i Entitat-Relació estesos (MEER). Es tracta d'un disseny d'alt nivell, més pròxim a l'usuari i més allunyat del disseny físic de la BD.

2. Diseny d'una BD

El disseny d'una base de dades consisteix a extraure totes les dades rellevants d'un problema, per exemple, saber que dades estan implicats en el procés de facturació d'una empresa que ven articles d'informàtica, o, que dades són necessaris per a portar el control de proves diagnòstiques en un centre de radiològic.

Per a extraure aquestes dades, s'ha de realitzar una **anàlisi** en profunditat del problema, per a esbrinar quines dades són essencials per a la base de dades i descartar els que no siguen necessaris. Una vegada extrets les dades essencials comencem a construir els models adequats. És a dir, construïm, mitjançant una eina de disseny de base de dades, un esquema que expressi amb total exactitud totes les dades que el problema requereix emmagatzemar. Podríem dir que és una cosa equivalent al dibuix d'un pla previ a la construcció d'un edifici.

Previ al disseny de la nostra base de dades és necessari realitzar una primera fase denominada d'anàlisi, a partir de la qual, anirem travessant diferents fases del procés de disseny.



2.1. Fase d'Anàlisi: Especificació de requisits Programari (E.R.S.)

Abans de passar a dissenyar una BD cal tindre clar que és el que volem fer. Per a això, típicament els informàtics es reuneixen amb els futurs usuaris del sistema per a recopilar la informació que necessiten per a saber que desitgen aquests usuaris. Normalment es fa una reunió inicial a i partir d'ella s'elabora una bateria de preguntes per a entrevistar els usuaris finals en una segona reunió i obtindre d'ella una informació detallada del que s'espera de nostra BD. D'aquestes entrevistes, s'extrau el document més important de l'anàlisi, el document d'Especificació de Requisits Programari o E.R.S.

A partir d'aquesta E.R.S. s'extrau tota la informació necessària per a la modelització de dades.

2.2. Fase del Disseny Conceptual: Model Entitat-Relació (MER)

Habitualment qui realitza la modelització és un analista informàtic que no té perquè ser un expert en el problema que pretén resoldre (Comptabilitat, Gestió de Reserves hoteleres, medicina, economia, etc.). És per això que és imprescindible comptar amb l'experiència d'un futur usuari de la BD que conega a fons tots els secrets del negoci, i que, al seu torn, no tenen perquè tindre cap coneixement d'informàtica.

L'objectiu d'aquesta fase del disseny consisteix és representar la informació obtinguda de l'usuari final i concretada en l'E.R.S. mitjançant estàndards perquè la resta de la comunitat informàtica pugui entendre i comprendre el model realitzat. El model que s'utilitza en aquesta primera fase del disseny té un gran poder expressiu per a poder comunicar-se amb l'usuari que no és expert en informàtica i es denomina Model Conceptual.

El model conceptual que utilitzarem és el Model Entitat-Relació i anirem aprofundint en ell al llarg d'aquesta unitat.

2.3. Fase del Diseny Lògic: Model Relacional

Aquest model és més tècnic que l'anterior perquè està orientat al personal informàtic i generalment té traducció directa a la del model físic que entén el SGBD. S'obtenen a partir del model conceptual i dependrà de la implementació de la BD. Així, no és el mateix implementar una base de dades jeràrquica o orientada a objectes que unisca *BD relacional.

El model que s'usarà en aquest mòdul és el Model Relacional i el veurem en la següent unitat.

2.4. Fase del Disseny Físic: Model Físic

És el resultat d'aplicar el model lògic a un SGBD concret. Generalment està expressat en un llenguatge de programació de BD tipus SQL. En aquest mòdul, transformarem el Model Relacional en el model físic mitjançant el subllenguatge DDL de SQL. Això s'estudiarà en la unitat didàctica 4.

3. Model Entitat-Relació

El model Entitat-Relació és el model més utilitzat per al disseny conceptual de bases de dades. Va ser introduït per Peter Chen en 1976 i representa la realitat a través d'entitats, que són objectes que existeixen i es distingeixen d'uns altres per les seues característiques, que anomenem *atributs*. A més, aquestes entitats podran o no estar relacionades les unes amb les altres a través del que es coneix com a *relació*.

Cal tindre en compte que es tracta solament d'un model de representació, per la qual cosa no té correspondència real amb cap sistema d'emmagatzematge. S'utilitza en l'etapa d'Anàlisi i Disseny d'una Base de dades, per la qual cosa caldrà convertir-la a un altre model abans de poder començar a treballar amb ella.

A continuació veurem en detalls alguns dels elements fonamentals d'aquest model.

3.1. Entitats

Una entitat és qualsevol objecte o element sobre el qual es puga emmagatzemar informació en la BD. És a dir qualsevol element informatiu que tinga importància per a una base de dades.

Les entitats poden ser concretes com una persona o abstractes com una data. Descobrir entitats és la tasca principal del disseny d'esquemes Entitat/Relació.

Exemples d'entitats són: una persona que es diu Pedro, la factura número 32456, el cotxe matrícula 3452BCW, etc.

Les entitats es representen gràficament mitjançant rectangles i el seu nom apareix a l'interior. Un nom d'entitat només pot aparèixer una vegada en l'esquema conceptual.



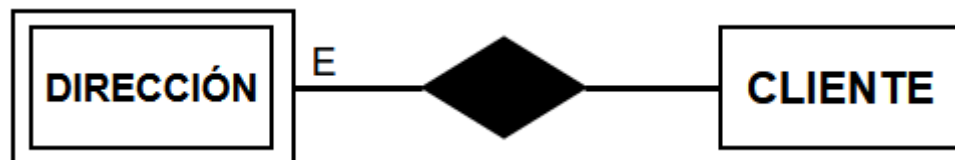
Hi ha dos tipus d'entitats:

- **Forts** (o regulars): Existeixen i s'identifiquen per si mateixes.



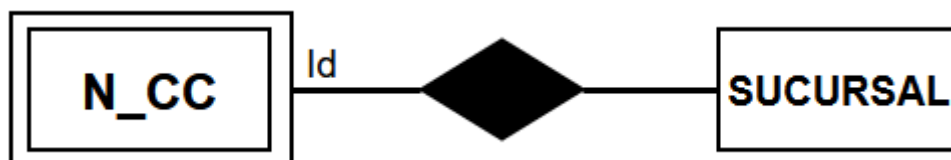
- **Febles:** Una entitat feble està unida a una entitat forta a través d'una relació de dependència. Hi ha dos tipus de relacions de dependència:
 - *D'Existència:* Es produeix quan una entitat feble necessita de la presència d'una forta per a existir. Si desapareix la tupla de l'entitat forta, la de la feble manca de sentit. Es representa per un doble rectangle i la lletra e. Són relacions poc freqüents.

(Ej: Client -> Direcció)



- *D'Identitat:* Es produeix quan una entitat feble necessita de la forta per a identificar-se. Per si sola la feble no és capaç d'identificar de manera unívoca les seues ocurrences. La clau de l'entitat feble es forma en unir la clau de l'entitat forta amb els atributs identificadors de l'entitat feble. Es representa per un doble rectangle i la lletra Id.

(Ej: Núm. de compte corrent -> Sucursal)

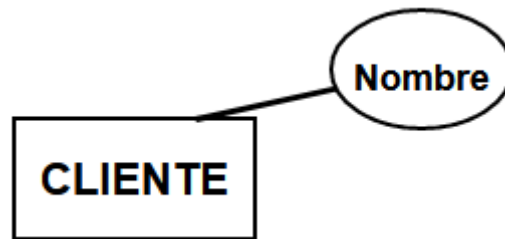


3.2. Atributs

Una entitat no és un propietat concreta, sinó un objecte que pot posseir múltiples propietats (**atributs**). És a dir "Sánchez" és el contingut de l'atribut *Primer Cognom* de l'entitat que representa a la persona *Pedro Sánchez Crespo* amb DNI 12766374,...

Una entitat es caracteritza i distingeix d'una altra pels **atributs**, en ocasions anomenades *propietats o camps*, que representen les característiques d'una entitat. És a dir, una entitat no és un propietat concreta, sinó un objecte que pot posseir múltiples propietats (**atributs**). Així "Sánchez" és el contingut de l'atribut *Primer Cognom* de l'entitat que representa a la persona *Pedro Sánchez Crespo* amb DNI 12766374,...

Els atributs se solen representar per una el·lipse, unida a l'entitat a la qual pertany, dins del la qual apareix el nom d'aquest:



Els atributs d'una entitat poden prendre un conjunt de valors permesos al qual se'l coneix com a **domini** de l'atribut. Donant valors a aquests atributs, s'obtenen les diferents *ocurrències d'una entitat*.

Exemples:

- Núm. telèfon -> Domini: cadena de 9 caràcters.
- Nomene -> Domini: cadena de "x" caràcters.
- Edat -> Domini: nombre enter.

En essència, els atributs es poden classificar en:

- **Identificadors** d'entitat (també anomenats clau primària o clau principal): són atributs que identifiquen de manera unívoca cada ocurrència d'una entitat (no es poden repetir).
- **Descriptors** d'entitat: són atributs que mostren unes característiques de l'entitat.

Sempre ha d'existir, almenys, un atribut identificatiu.

Tipus d'atributs

ATRIBUT CLAU = Atribut que identifica de manera unívoca a una entitat. Si per a identificar d'aqueixa manera a l'entitat són necessaris més d'un atribut, se subratllaria cadascun dels atributs que formen part d'aquesta clau.



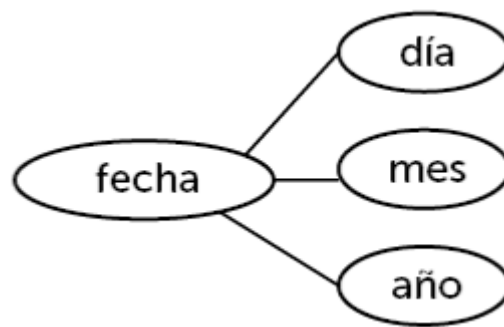
VALOR NO NUL = Atribut que no pot contindre un valor nul.



MULTIVALUAT = Atributs que posseeixen diversos valors d'un mateix domini. Per exemple, telèfons (podent emmagatzemar diversos telèfons per a un mateix empleat).



COMPOST = Moltes vegades es confonen amb els anteriors, encara que no tenen res a veure amb ells. Un atribut compost és un atribut que pot ser descompost en altres atributs pertanyents a diferents dominis. Un exemple clar és la data, composta per un valor numèric sencer per al dia, un altre per al mes i un altre per a l'any.



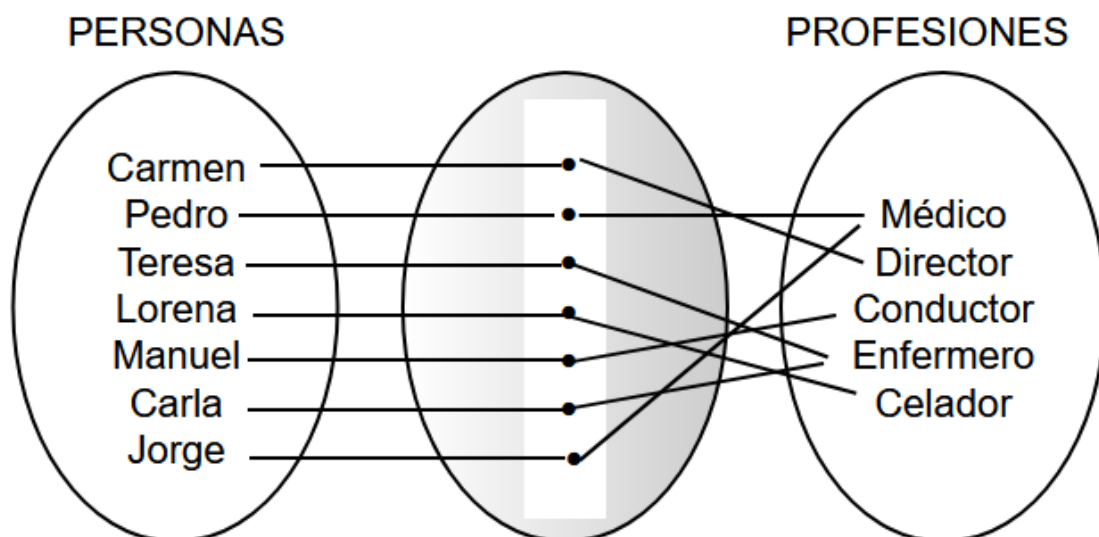
DERIVAT = Són atributs els valors dels quals es calculen a partir dels valors d'altres atributs. Per exemple podem disposar d'un atribut `data_nac` que seria un atribut descriptiu normal i calcular el valor de l'atribut `edat` a partir d'ell. El `preu_total` també podria calcular-se a partir del `preu` + % iva.



3.3. Relacions

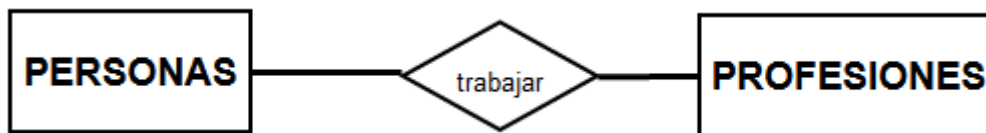
Una relació representa l'associació que existeix entre dues a més entitats. És l'element del model que permet relacionar en sí les dades d'aquest, d'una altra manera tindríem informació aïllada.

Per exemple, en el cas que tinguem una entitat **persones** i una altra entitat **professions**. Ambdues es relacionen ja que les persones treballen i els treballs són realitzats per persones.



En el cas anterior treballar podria ser el nom del conjunt de relacions entre les entitats *persones* i *professions*.

Les relacions es representen gràficament mitjançant un rombe al qual se li uneixen línies que es dirigeixen a les entitats. El nom de la relació apareix a l'interior d'aquest rombe i se sol utilitzar un verb.

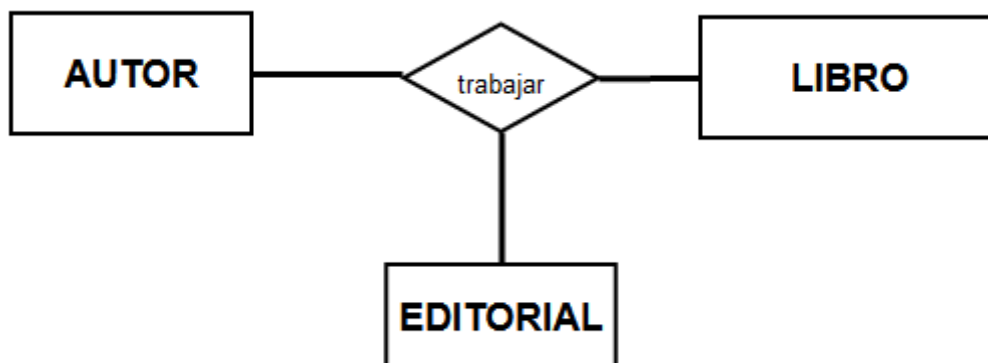


Grau de les relacions

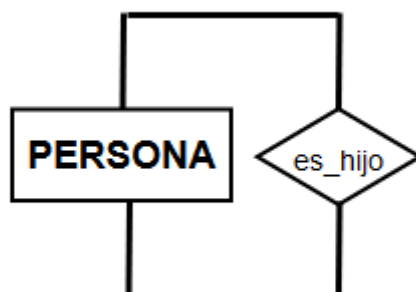
Les entitats que estan involucrades en una determinada relació es denominen entitats participants. El nombre de participants en una relació és el que es denomina *grau de la relació*.

Atenent el seu grau podem trobar:

- **Relacions Binàries:** Són les relacions típiques. Es tracta de relacions que associen dues entitats. Per exemple la relació **treballar** és de grau 2 o binària, ja que intervenen dues entitats, **PERSONES** i **PROFESSIONS**.
- **Relacions Ternàries:** Es tracta de relacions que associen tres entitats.



- **Relacions Unàries o Reflexives:** Es tracta de relacions que associen una entitat amb si mateixa.



- **Relacions n-àries:** Es tracta de relacions que associen n entitats.

Cardinalitat de les relacions

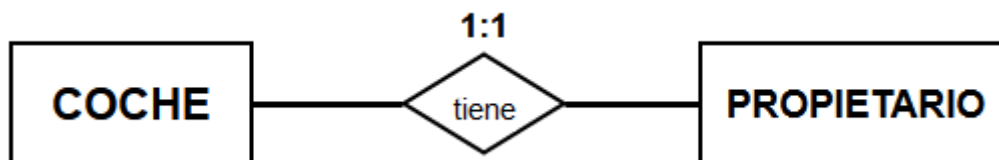
La cardinalitat d'una relació (normalment binària) és el nombre d'ocurrències d'una entitat associades a una ocurrència de l'altra entitat, mesurat en número màxim i mínim.

La **cardinalitat Màxima** indica el nombre màxim d'ocurrències en la qual pot aparèixer cada *tupla d'una entitat. Aquest pot valor podrà ser 1, un altre valor concret superior a 1 (3 per exemple) o molts (en aquest cas es representa amb una N). Normalment la cardinalitat màxima sol ser **1** o **N**.

Existeixen principalment tres tipus de cardinalitats màximes en relacions binàries:

Relació un a un 1:1

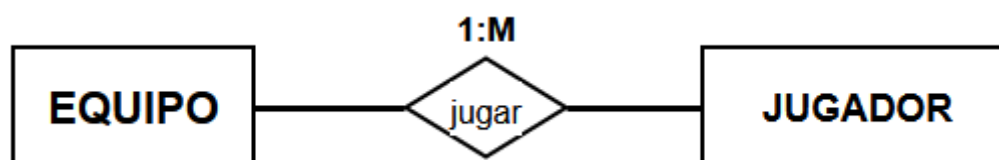
A cada element de la primera entitat li correspon un i no més d'un element de la segona entitat, i al revés. És representat gràficament de la següent manera:



La lectura s'ha de realitzar en tots dos sentits: **un cotxe te 1 propietari** i **un propietari noméste 1 cotxe**.

Relació un a molts 1:N

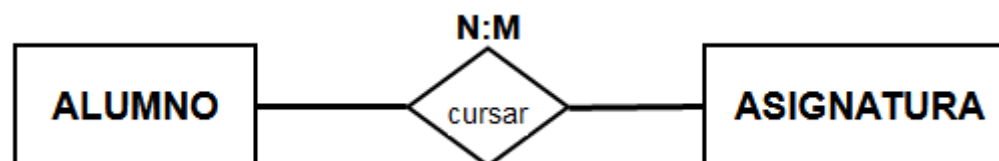
Significa que cada element d'una entitat pot relacionar-se amb qualsevol quantitat d'elements de l'altra entitat (M), i un element d'aquesta última entitat només pot estar relacionat amb un element de la primera entitat. La seua representació gràfica és la següent:



La lectura se debe realizar en ambos sentidos: **en un equip juguen molts(M) jugadors** i **un jugador només juga en 1 equip**.

Molts a molts N:M

Estableix que qualsevol quantitat d'elements d'una entitat (N) poden estar relacionats amb qualsevol quantitat d'elements de l'altra entitat (M). És representat gràficament de la següent manera:

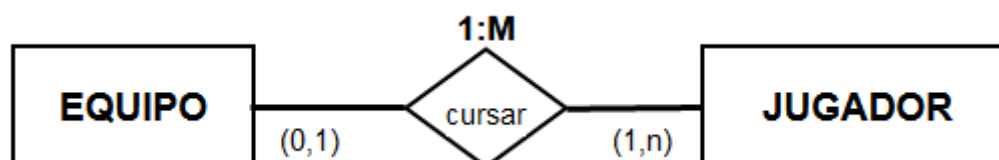


La s'ha de realitzar en tots dos sentits: **un alumne cursa moltes(M) assignatures** i **una assignatura es cursada per molts (N) alumnes**.

La **cardinalitat Mínima** indica el nombre mínim d'associacions en les quals apareixerà cada tupla de l'entitat (el valor que s'anota és de 0 o 1, encara que tinga una cardinalitat mínima de més d'un, s'indica només 1).

En els esquemes entitat-relació la cardinalitat es pot indicar de moltes formes. Potser la més completa consisteix a anotar en els extrems la cardinalitat màxima i mínima de cada entitat en la relació (min, max).

En el següent exemple un jugador té una cardinalitat mínima de 0 (pot no estar en cap equip) i una màxima d'1 (com a molt està en un equip, no pot estar en dos alhora). Cada equip té una cardinalitat mínima d'un (en realitat seria una cardinalitat mínima més alta, però s'anota un u) i una màxima de n (en cada equip hi ha molts jugadors), quedant la representació de la següent forma:



3.4. Claus

La **clau primària** és un o més atributs d'una entitat els valors de la qual són únics en cada ocurrència de l'entitat. És a dir tots els elements d'una entitat tenen en aqueix (o aqueixos) atribut, un valor diferent (i mai buit).

Aquest tipus d'atributs són fonamentals i es marquen en l'esquema subratllant el nom de l'identificador.

NIF

Perquè un atribut siga considerat un bon identificador ha de complir amb els següents requisits:

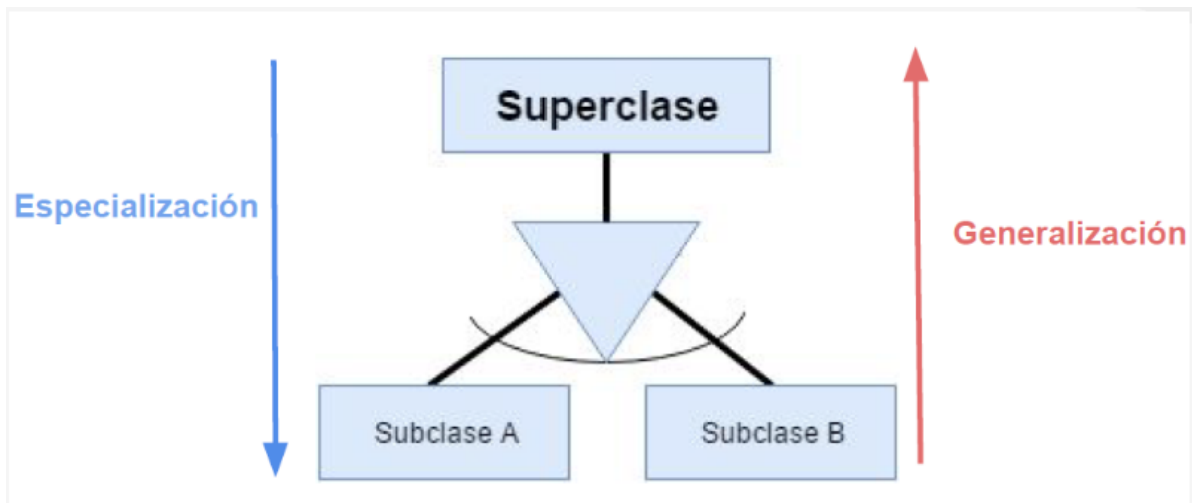
1. Han de distingir a cada exemplar de l'entitat o relació. És a dir no pot haver-hi dues ocurrències amb el mateix valor en l'identificador. Identifiquen unívocament cada ocurrència de l'entitat.
2. Tots les ocurrències d'una entitat han de tindre el mateix identificador.
3. Un identificador pot estar format per més d'un atribut (clau primària composta).
4. Pot haver-hi diversos **claus candidates**, en aqueix cas cal triar la que tinga més importància en el nostre sistema (la resta passen a ser **claus alternatives**). Les claus candidates les representarem amb un subratllat discontinu.

NIA

Totes les entitats han de tindre una clau primària, en el cas que una entitat no dispose d'un identificador (pot ocórrer, però cal ser cautelós, a vegades es tracta d'entitats que estan mal modelades) llavors cal afegir un atribut que faça d'identificador. Els futurs valors d'aquest atribut no ens interessin, l'única cosa que necessitem d'ell és que dispose d'un valor diferent per a cada exemplar de l'entitat. El nom d'aquest atribut artificial és la paraula **aneu** seguida del nom de l'entitat. Per exemple *aneu_persones*.

4. Model Entitat-Relació Estés (MEER)

El model Entitat-Relació Estés (MEER) inclou tot el vist en el MER però a més incorpora els conceptes de **Generalització** i **Especialització**, (així com el de **subclasse** i **superclasse**), la qual cosa es coneix com a **Relacions de Jerarquia**.



Una relació de jerarquia es produeix quan una entitat es pot relacionar amb unes altres a través d'una relació el rol de la qual seria «És un tipus de».

S'utilitzen per a unificar entitats agrupant-les en una entitat més general (**generalització**) o bé per a dividir una entitat en entitats més específiques (**especificació**): encara que hui dia a totes elles li les sol dir generalització o potser fins i tot més adequadament **relacions d'herència**.

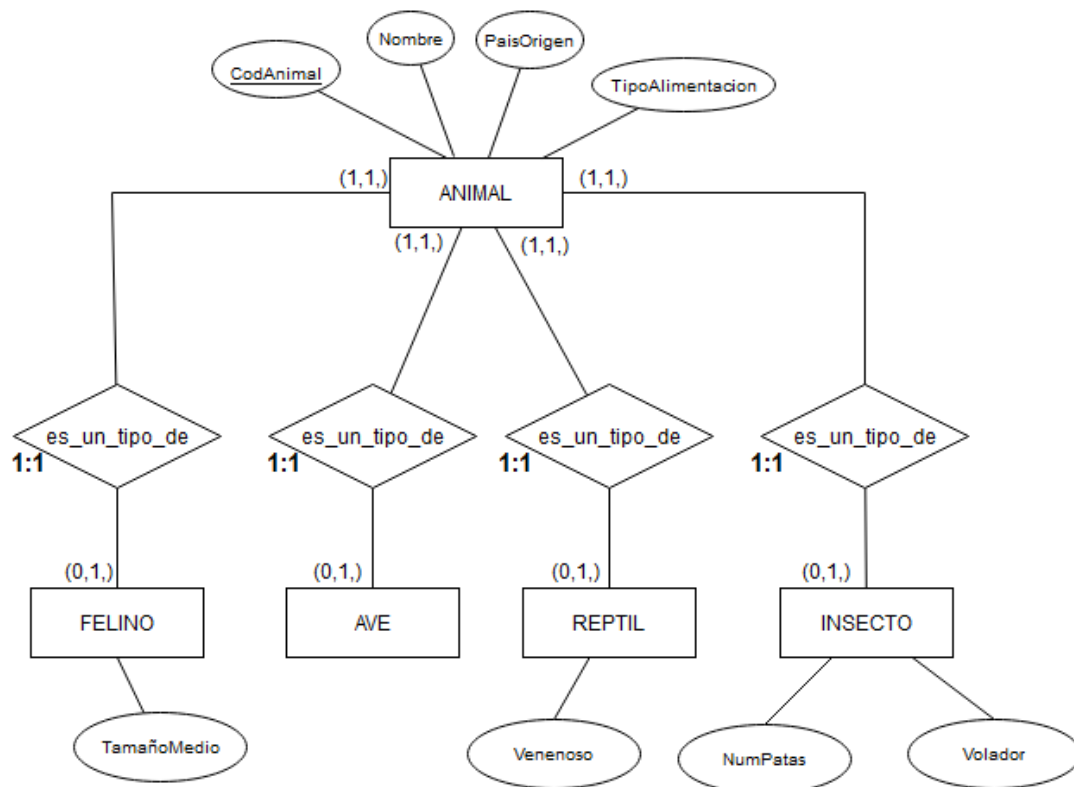
Per exemple, imaginem la següent situació:

Volem fer una BD sobre els animals d'un Zoo.

Tenim les entitats ANIMAL, FELÍ, AVE, RÈPTIL, INSECTE.

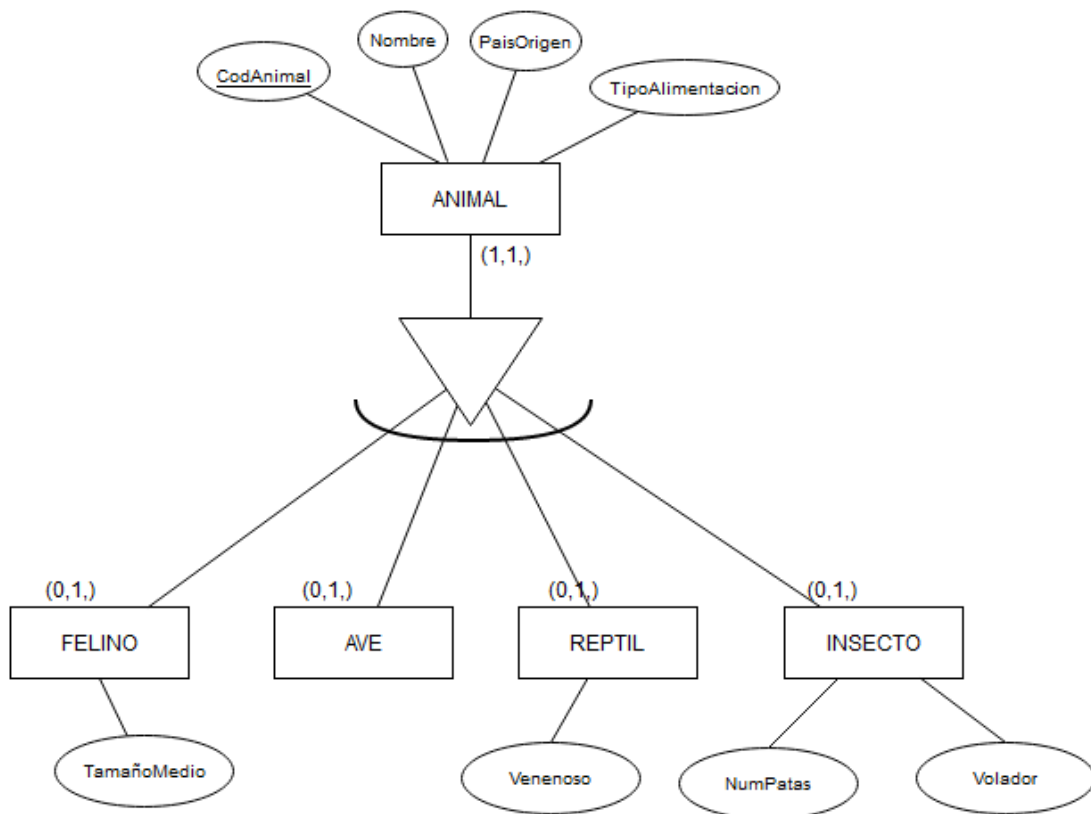
FELÍ, AVE, RÈPTIL i INSECTE tindrien el mateix tipus de relació amb ANIMAL: «són un tipus de».

*Ara bé, la seua representació mitjançant el *MER clàssic seria bastant enutjosa:*



Per a evitar haver de repetir tantes vegades el rombe de la mateixa relació, s'utilitzen uns símbols especials per a aquests casos i se substitueixen tots els rombes de relació «és un tipus de» per un triangle invertit, on l'entitats de baix són sempre un tipus de l'entitat de la de dalt i es diuen **subtipus, entitats filles o subclasses**. La de dalt es denominarà **supertipo, entitat pare o superclase**.

L'exemple anterior quedaria de la manera següent utilitzant símbols del MEER:



En la **superentidad** s'indiquen els atributs comuns a totes les subentidades, se sobreentén que les subentidades també tenen aqueixos atributs, però no s'indiquen de nou aqueixos atributs en el diagrama (CodAnimal, Nom, PaisOrigen, TipoAlimentacion). La resta són atributs Només de cada subentidad (per exemple, l'atribut Verinós pertany només als RÈPTILS).

Normalment quan tenim una especialització, les **subentidades** comparteixen clau amb la **superentidad** (a més dels atributs comuns); això és molt matisable i de fet hui dia cap dissenyador intenta distingir entre si tenim una especialització o una generalització, perquè al final ambdues impliquen el mateix esquema intern en la base de dades.

En general se solen indicar les *cardinalitats en les relacions de jerarquia*, encara que se sol sobreentendre (quan no s'indiquen explícitament) que hi ha un (0,1) damunt de cada subentidad (que significa que cada exemplar de la subentidad només pot relacionar-se com a molt amb un de la subentidad i fins i tot amb cap; un *animal* podria ser o no ser un *felí*).

Les cardinalitats que apareixen en l'esquema anterior són les habituals (de fet no se sol indicar ja que se sobreentén), però qualsevol cardinalitat seria vàlida.

4.1. Quan cal utilitzar relacions de jerarquia?

Quan els dissenyadors principiants coneixen les relacions de jerarquia, sol ocórrer un excés d'ús d'elles en els dissenys. No és convenient abusar d'aquesta mena de relacions, ja que augmenten el nombre d'entitats.

No tots els tipus d'entitats suposen l'aparició d'una generalització o una especialització. A vegades basta diferenciar els tipus d'entitat amb l'ús d'un simple atribut tipus o, com a molt, una entitat que continga els tipus d'entitats.

El recomanable és utilitzar relacions de jerarquia quan ocorre qualsevol d'aquestes situacions:

- Les subentidades tenen atributs diferents.
- Les subentidades tenen relacions diferents.

En altres casos, es pot concloure que no resulta necessària la creació d'aquesta mena de relacions.

4.2. Tipus de relacions de Jerarquia.

Els tipus de relacions de jerarquia venen definides per la manera de relacionar-se la superentidad amb les subentidad. D'aqueixa manera podem trobar-nos:

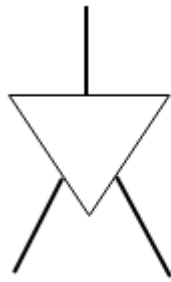
Obligatorietat: Indica si la subentidad es relacionen amb les subentidades. En aquest sentit tenim dues possibilitats:

- **Total:** Subdividim l'entitat Emprat en: Enginyer, Secretari i Tècnic i en nostra *BD no hi ha cap altre empleat que no pertanga a un d'aquests tres tipus.
- **Parcial:** Subdividim l'entitat Emprat en: Enginyer, Secretari i Tècnic però en nostra *BD pot haver-hi emprats que no pertanguen a cap d'aquests tres tipus.

Nombre de relacions: En aquest cas es mesura amb quantes subentidades es relaciona la subentidad. Possibilitats:

- **Solapada:** Subdividim l'entitat Emprat, en: Enginyer, Secretari i Tècnic i en nostra BD pot haver-hi emprats que siguin alhora Enginyers i secretaris, o secretaris i tècnics, etc.
- **Exclusiva:** Subdividim l'entitat Emprat en: Enginyer, Secretari i Tècnic. En nostra BD cap empleat pertany a més d'una subentidad.

La forma gràfica més acceptada actualment per a representar aquest tipus de relacions és la que vam mostrar a continuació:



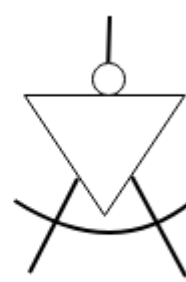
Solapada y Parcial



Solapada y Total



Exclusiva y Parcial

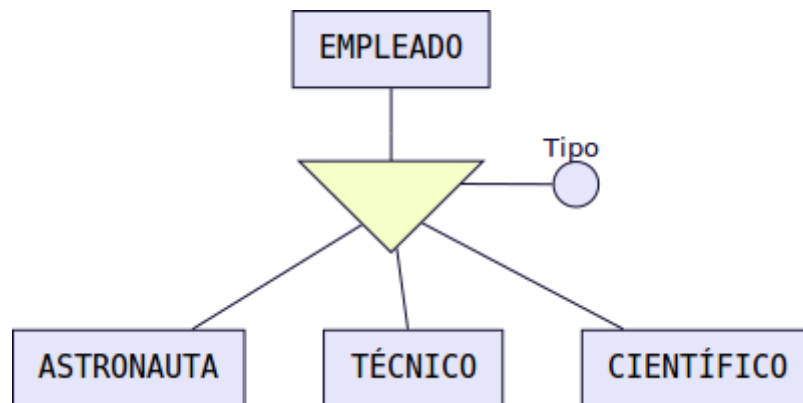


Exclusiva y Total

4.3. Exemples.

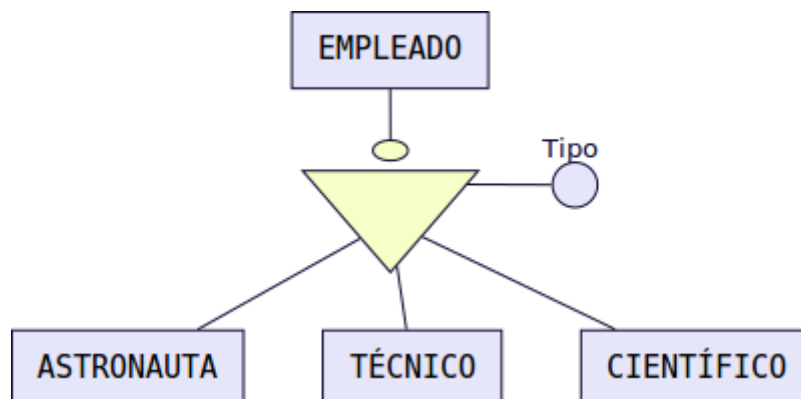
A continuació vam mostrar alguns *exemples* de cadascun d'aquests tipus de relacions de jerarquia:

Parcial i Solapada



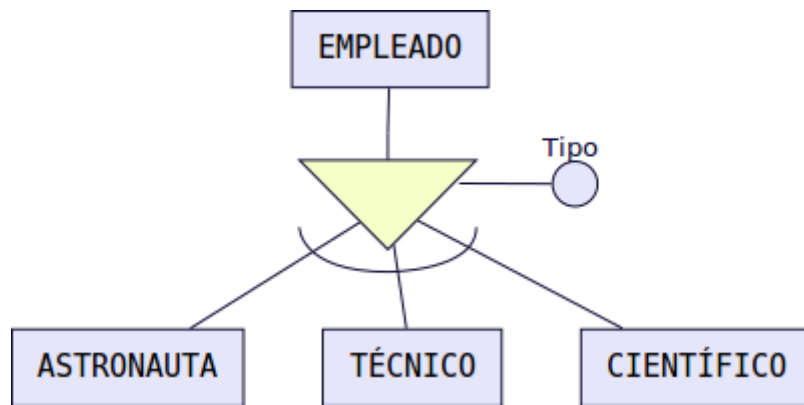
En aquesta BD un empleat podria ser simultàniament tècnic, científic i astronauta o tècnic i astronauta, etc. (solapada). A més pot ser tècnic, astronauta, científic o exercir una altra ocupació diferent (parcial).

Total i Solapada



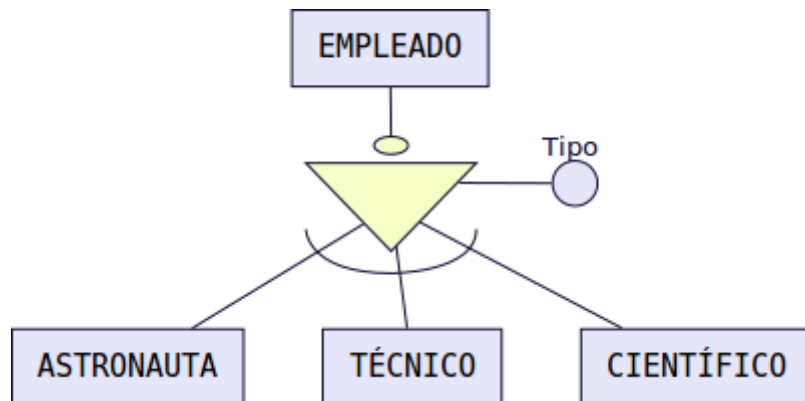
En aquesta BD un empleat podria ser simultàniament tècnic, científic i astronauta o tècnic i astronauta, etc. (solapada). A més pot ser solament tècnic, astronauta o científic (total).

Parcial i Exclusiva



En aquesta BD un empleat només pot exercir una de les tres ocupacions (exclusiva) . A més pot ser tècnic, o ser astronauta, o ser científic o també exercir una altra ocupació diferent, per exemple, podria ser FÍSIC (parcial).

Total i Exclusiva



Un empleat pot ser solament tècnic, astronauta o científic (total) i no ocupar més d'un lloc (exclusiva).