

Tema 3: Model E/R a Relacional

1er Desenvolupament d'aplicacions multiplataforma

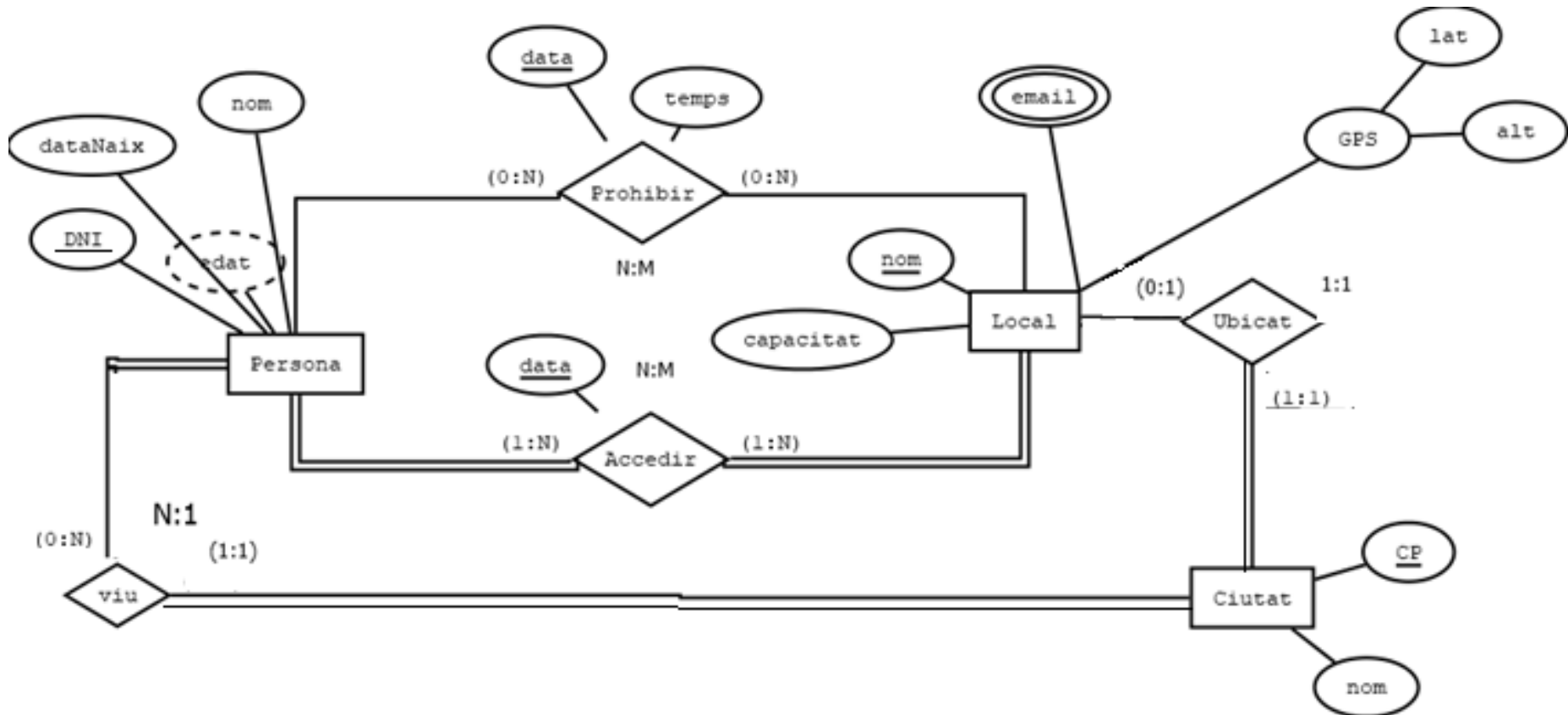
Transformació E/R a Relacional

Cal realitzar una **sèrie de transformacions** segons els atributs, els conjunts d'entitats, les relacions i el tipus de relacions. Per a explicar aquest procés s'utilitzarà el disseny per a controlar els accessos a un local segons el següent diagrama ER.

Per representar el model relacional, no requerim de cap software específic, amb un editor de text marcant què son les entitats, Claus primàries (PK) i Claus foranies (FK), en serà suficient.

Transformació E/R a Relacional

Ens basarem en aquest diagrama per realitzar la transformació a model relacional.



Conjunt d'entitats

- Tota entitat genera una nova taula.
- A mesura que avancem en el procés de transformació es podran generar taules addicionals.
- Afegirem com a camps de les nostres taules els atributs associats, **excepte els calculats, multivalors**. Dels compostos n'agafarem el detall.

Persona: {dni, nom, dataNaix}

Local: {nom, capacitat, lat, long}

Ciutat: {cp, nom}

Atributs Identificadors

- Cada taula tindrà com a clau primària els atributs identificadors de l'entitat.
- Per assenyalar la clau primària subratllem el nom del camp. Addicionalment es pot incloure el text (PK) després del nom del camp.

Persona: {dni (PK), nom, dataNaix}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Atributs Calculats/Derivats

Tots els atributs calculats s'eliminen del model relacional.

En el nostre exemple hi ha un atribut “edat” que es calcula a partir de l'atribut “dataNaix”. La taula “Persona” no precisa d'aquest atribut.

Persona: {dni (PK), nom, dataNaix, ~~edat~~}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Atributs Multivalor

- Es generarà una nova taula per cada atribut multivalor
- Aquesta taula contindrà l'atribut/s identificador/s i el propi multivalor.
- El/s atribut/s identificador/s seran alhora PK i FK.

Persona: {dni (PK), nom, dataNaix}

Email: {localNom (PK, FK), email (PK)}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Atributs Compostos

- Dels atributs compostos només es consideren en el model relacional aquells atributs que realment contenen la informació (nodes fulla).

Persona: {dni (PK), nom, dataNaix}

Email: {localNom (PK, FK), email (PK)}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Relacions 1:N / N:1

Quan entre dos conjunts d'entitats s'estableix una relació 1:N o N:1, l'atribut identificador de la taula que té cardinalitat 1 s'afegeix a la taula amb cardinalitat N.

Persona: {dni (PK), nom, dataNaix, ciutatCP (FK)}

Email: {localNom (PK, FK), email (PK)}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Relacions N:M

- Per les relacions N:M sempre cal generar una nova taula.
- La nova taula contindrà com a mínim els atributs identificadors de les entitats relacionades.
- Els atributs associats a la relació caldrà analitzar-los per determinar si passen a formar part de la clau primària o no.

Persona: {dni (PK), nom, dataNaix, ciutatCP (FK)}

Email: {localNom (PK,FK), email (PK)}

Local: {nom (PK), capacitat, lat, long}

Ciutat: {cp (PK), nom}

Accedir: {personaDNI (PK,FK), localNom (PK,FK), data (PK)}

Prohibir: {personaDNI (PK,FK), localNom (PK,FK), data (PK), temps}

Relacions 1:1

A les relacions 1:1, no hi ha una única possibilitat de transformació de taules i dependrà del cas que haurà d'analitzar-se en detall (diapositiva 13).

Persona: {dni (PK), nom, dataNaix, ciutatCP (FK)}

Email: {localNom (PK, FK), email (PK)}

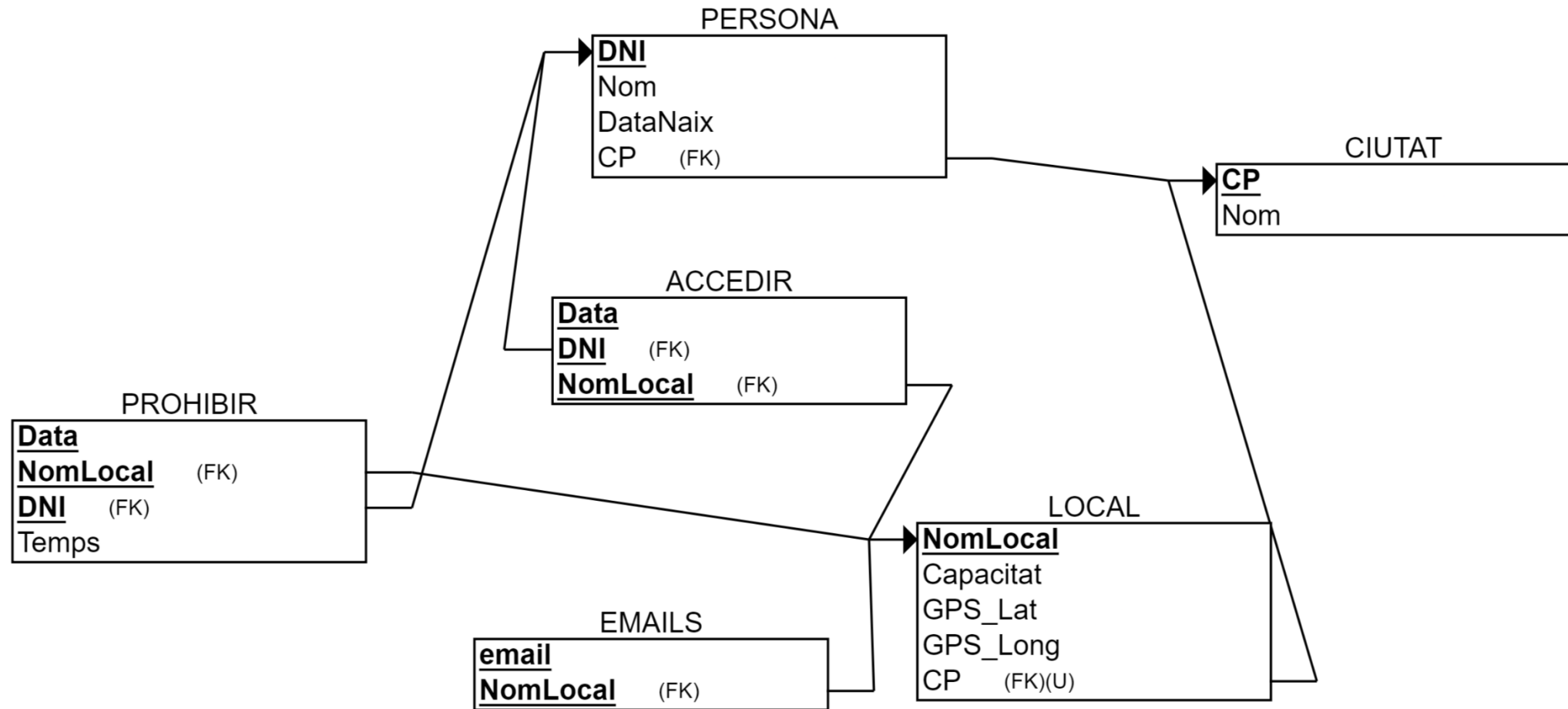
Local: {nom (PK), capacitat, lat, long, **ciutatCP (FK)**}

Ciutat: {cp (PK), nom}

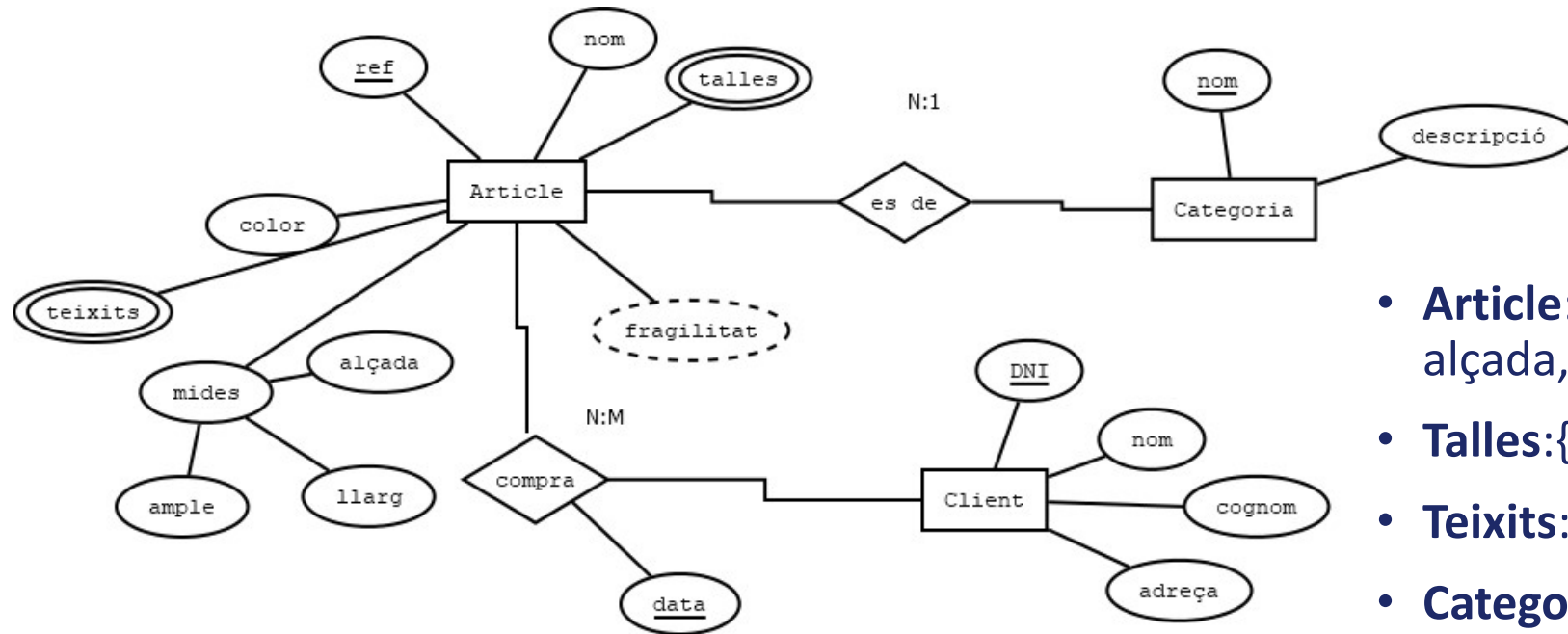
Accedir: {personaDNI (PK,FK), localNom (PK,FK), data (PK)}

Prohibir: {personaDNI (PK,FK), localNom (PK,FK), data (PK), temps}

Esquema Relacional Final

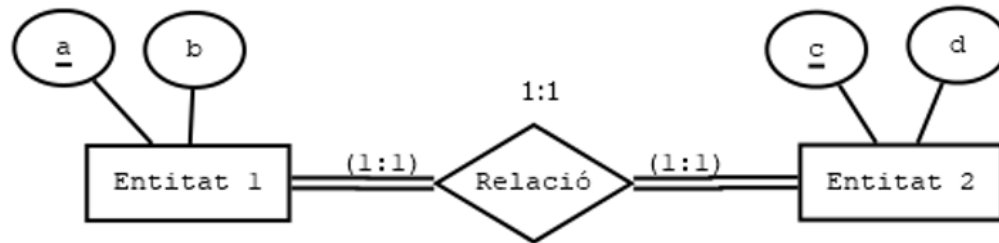


Altres exemples de resultats



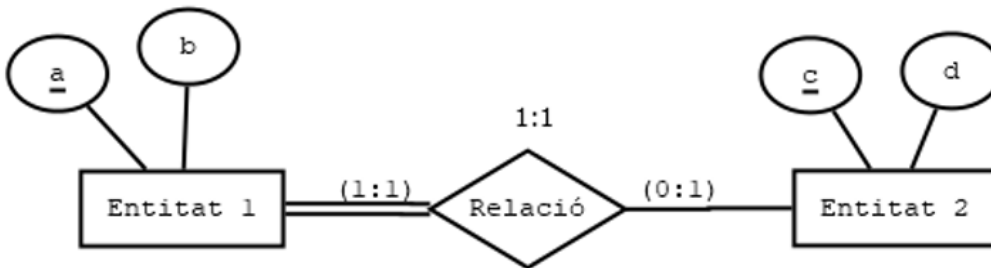
- **Article**: {ref(PK), nom, color, ample, llarg, alçada, categoriaNom(FK)}
- **Talles**: {talla(PK), articleRef(PK,FK)}
- **Teixits**: {teixit(PK), articleRef(PK,FK)}
- **Categoria**: {nom(PK), descripció}
- **Client**: {DNI(PK), nom, cognom, adreça}
- **Compra**: {data(PK), articleRef(PK,FK), clientDNI(PK,FK)}

Model relacional: Relacions 1:1



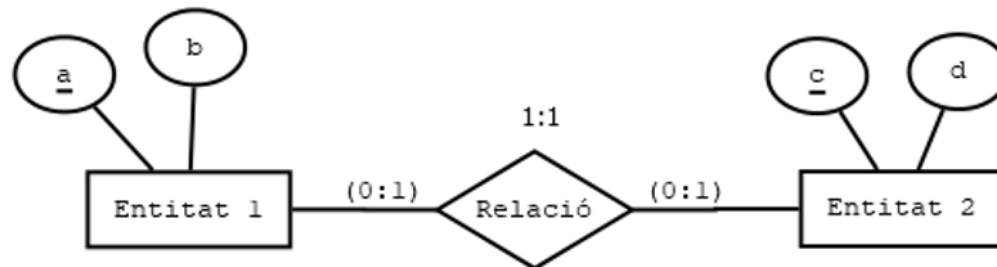
OPCIO 1:
 Entitat1Entitat2:{a(pk),b,c,d}
 Opció de crear dos Entitats
 amb un camp identificador

OPCIO 3:
 Entitat1:{a(pk),b}
 Entitat2:{c(pk),d,a(fk)}



OPCIO 2:
 Entitat1:{a(pk),b,c(fk)}
 Entitat2:{c(pk),d}

Entitat1:{a(pk), b}
 Entitat2:{c(pk),d, a(fk)}

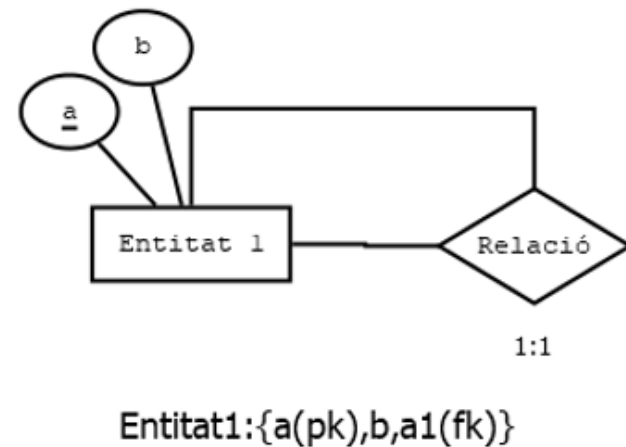


Entitat1:{a(pk),b}
 Entitat2:{c(pk),d}
 Relació:{a(pk,fk),c(fk)}

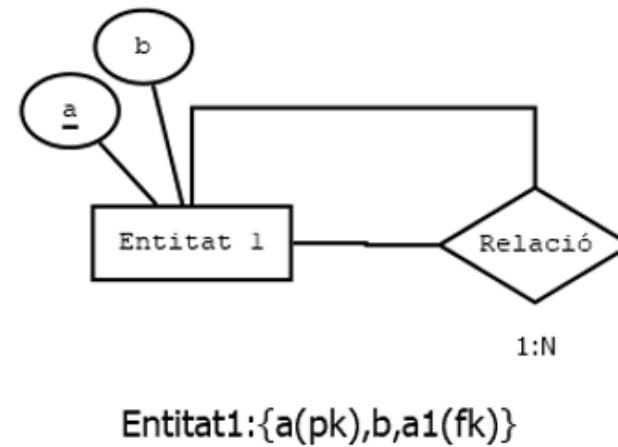
Model relacional: Relacions reflexives

Depenen del tipus de relació:

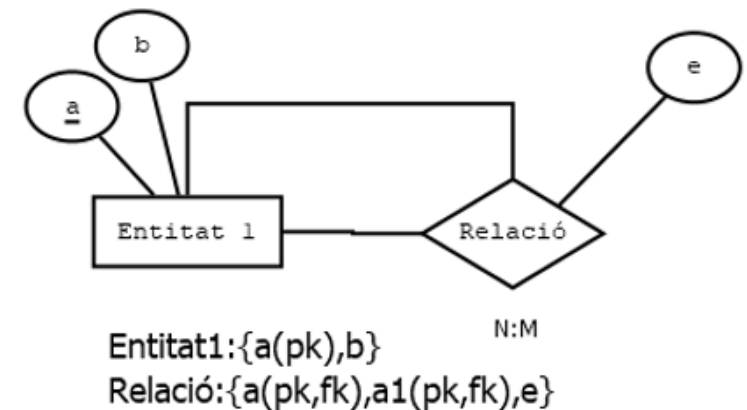
RELACIONS 1 a 1 (Reflexives)



RELACIONS 1 a N (Reflexives)

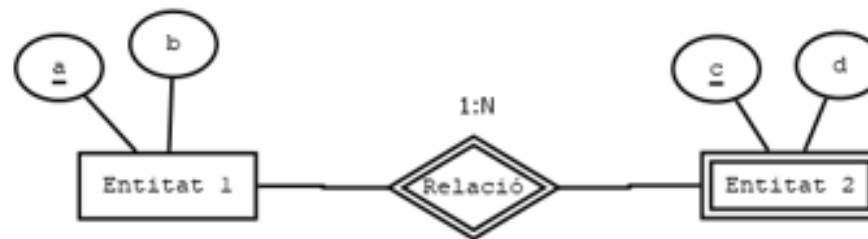


RELACIONS N a M (Reflexives)



Model relacional: Relacions amb entitats dèbil

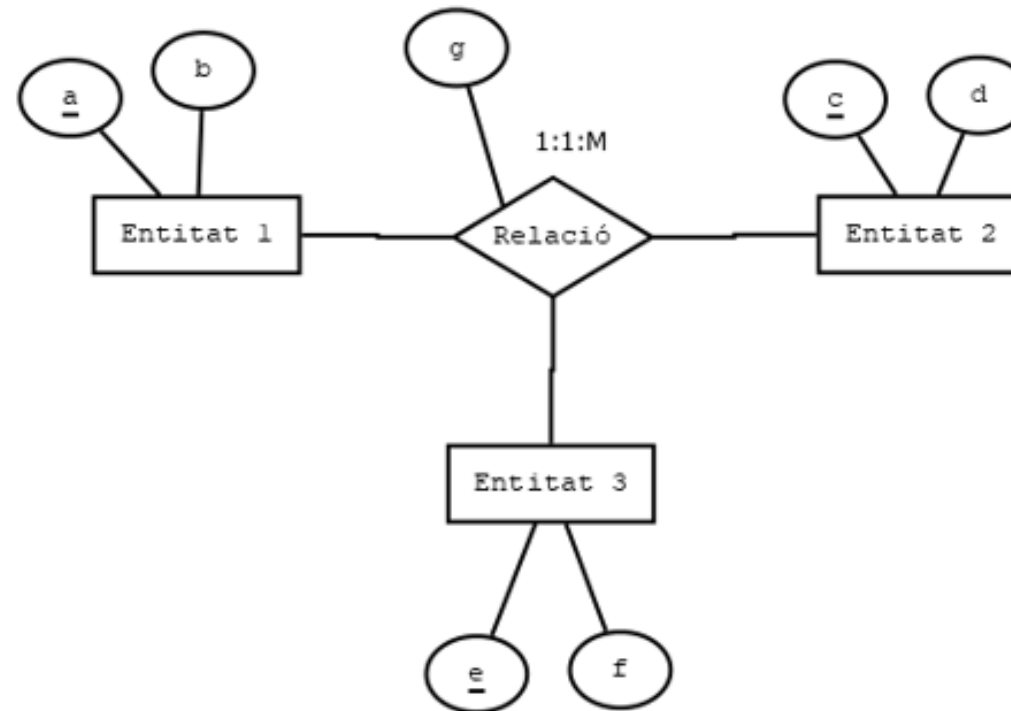
- Relació 1:N: Es comporta igual que una relació 1:N amb la diferencia que l'atribut que l'entitat dèbil rep com a foràni, és ademés atribut clau.



Entitat1: {a(pk), b}
 Entitat2: {c (pk), d, a(pk, fk)}

Model relacional: Relacions ternàries / N-àries

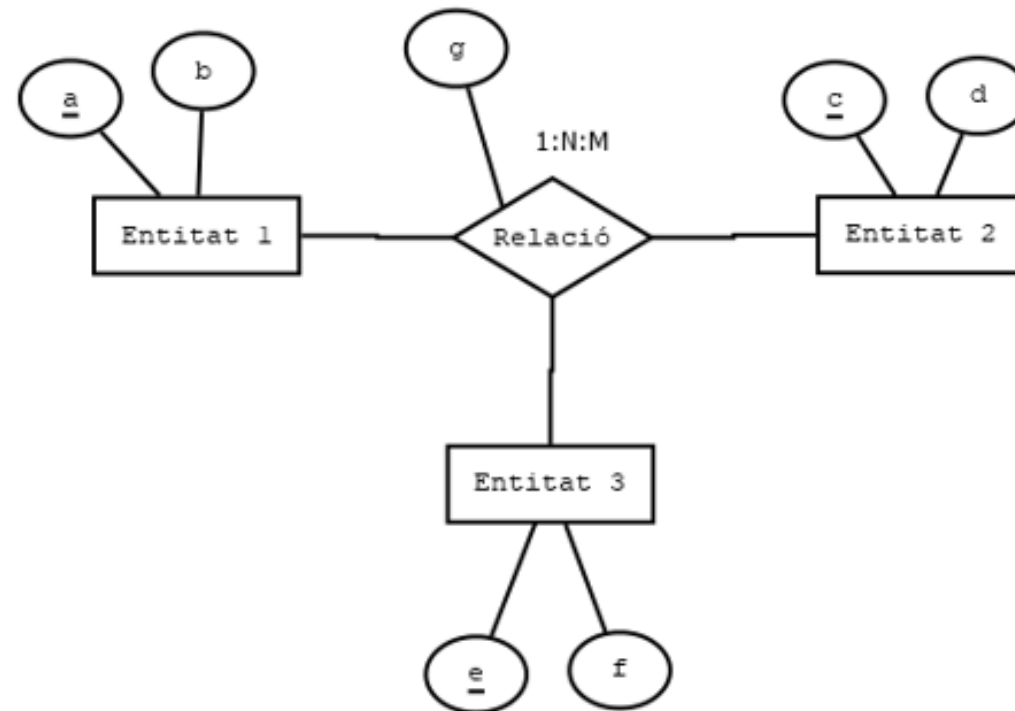
- Relació 1:1:N



Entitat1: {a(pk), b}
 Entitat2: {c (pk), d}
 Entitat3: {e(pk), f}
 Relació: {a(fk), c(fk), e(pk, fk), g}

Model relacional: Relacions ternàries / N-àries

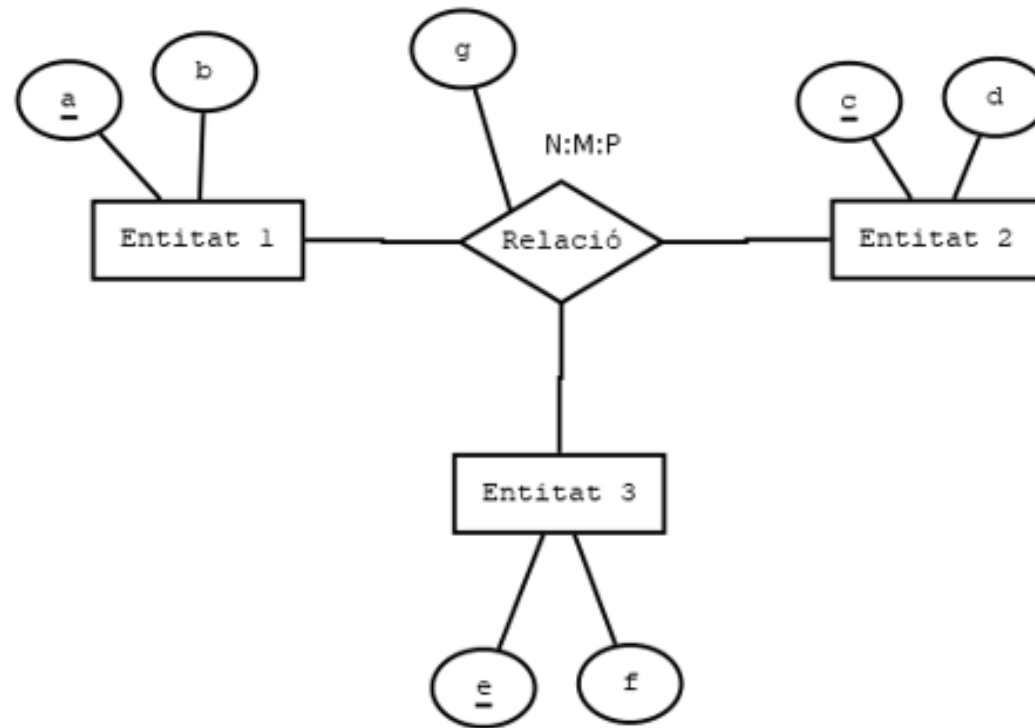
- Relació 1:N:M



Entitat1: {a(pk), b}
 Entitat2: {c (pk), d}
 Entitat3: {e(pk), f}
 Relació: {a(fk), c(pk, fk), e(pk, fk), g}

Model relacional: Relacions ternaries / N-àries

- Relacions N:M:P



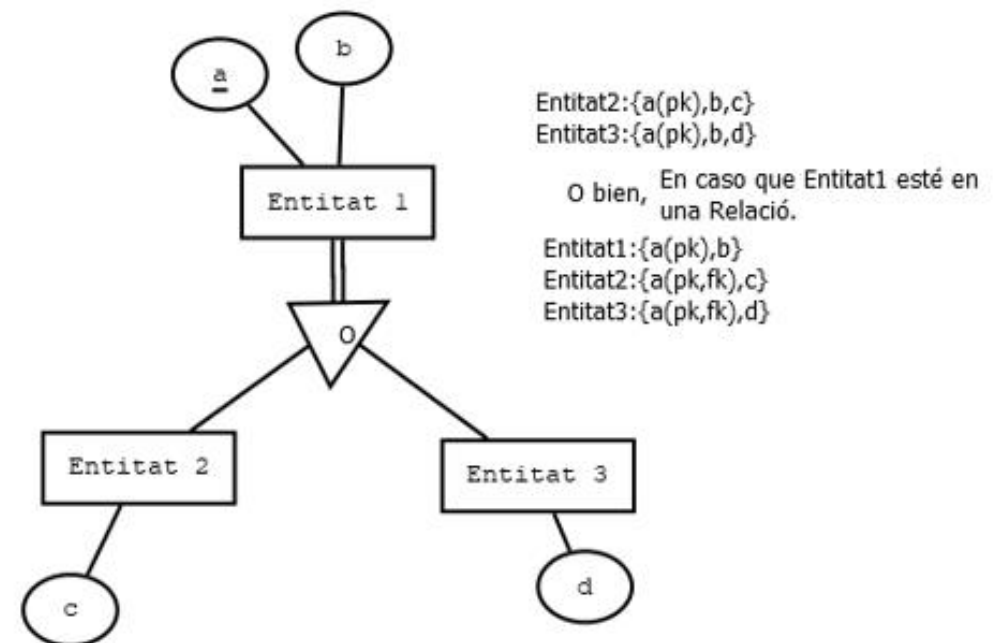
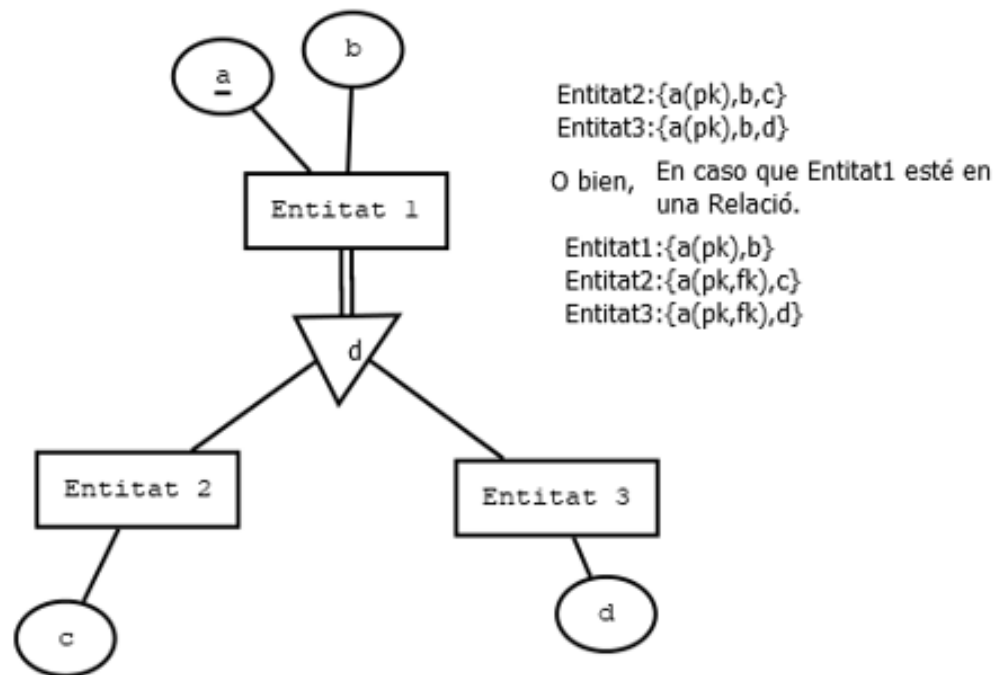
Entitat1: {a(pk), b}
 Entitat2: {c (pk), d}
 Entitat3: {e(pk), f}
 Relació: {a(pk, fk), c(pk, fk), e(pk, fk), g}

Model relacional: Jerarquies

- En el model relacional de les jerarquies, **no cal tenir en compte** si es **Disjunta** o **Solapada**.
- Només cal veure si es total o parcial.
- Cal tenir en compte si en la superclasse hi ha alguna relació que hi penja.

Relacional: Jerarquia Disjunta/Solapada Total

Sigui disjunta o solapada, en els dos casos es comporta igual.

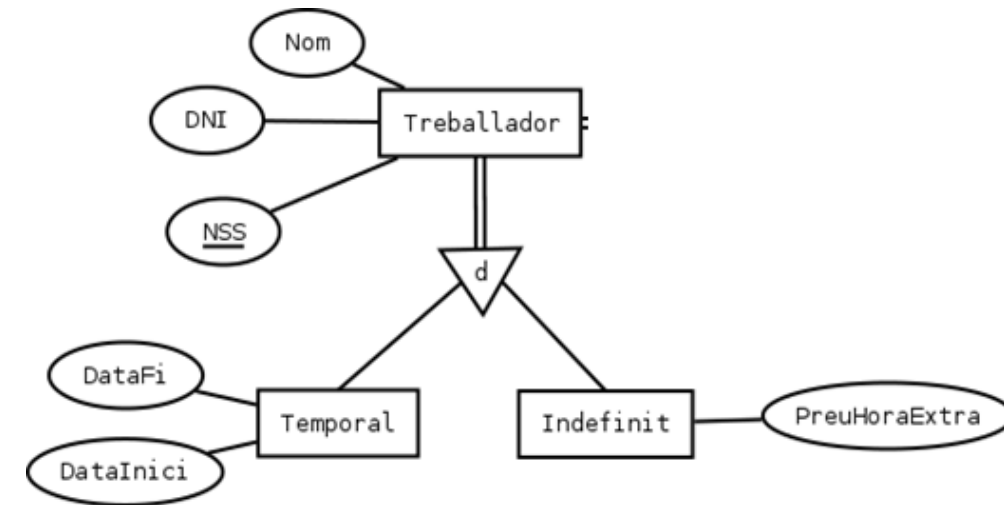


Relacional: Jerarquia Disjunta/Solapada Total (cas 1)

Creació de 2 taules (si no intervé en cap relació)

Es creen 2 taules, una pel conjunt d'entitats «Temporal» i una segona pel conjunt d'entitats «Indefinit» que són les subclasses. En cada una de les taules generades hi hauran de constar tots els atributs propis de la subclasse i els de la superclasse.

- **Temporal:** {NSS(PK), DNI, Nom, DataFi, DataInici}
- **Indefinit:** {NSS(PK), DNI, Nom, PreuHoraExtra}

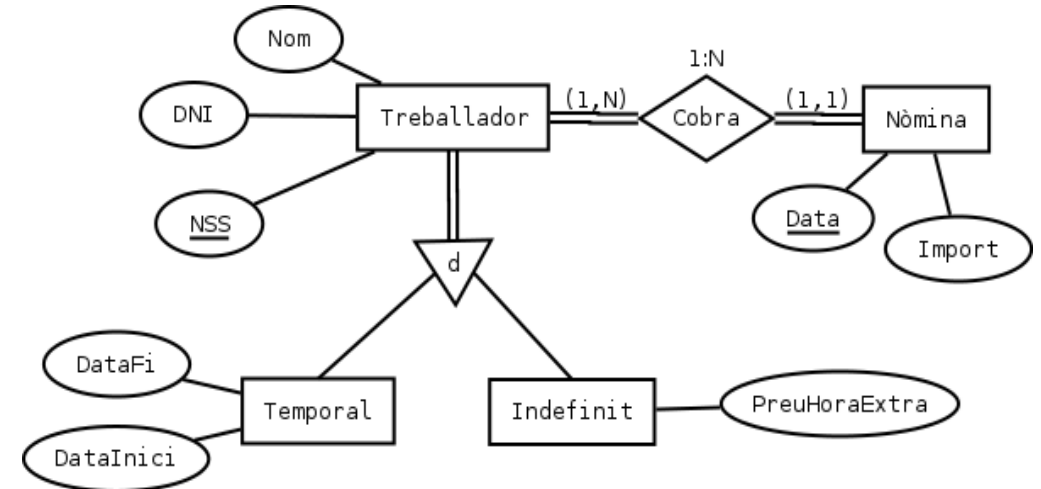


Relacional: Jerarquia Disjunta/Solapada Total (cas 2)

Creació de 3 taules

La taula generada a partir del conjunt d'entitats que és superclasse («Treballador») té tots els atributs comuns, mentre que cada taula associada a la subclasse disposaran dels seus propis atributs i l'atribut identificador de la superclasse.

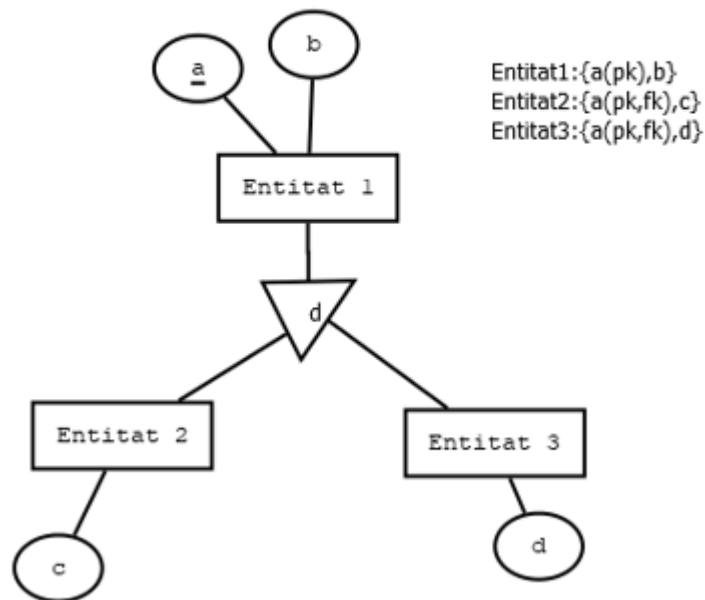
- **Treballador** {NSS(PK), DNI, Nom}
- **Temporal**: {NSS(PK,FK), DataFi, DataInici}
- **Indefinit**: {NSS(PK,FK), PreuHoraExtra}
- **Nòmina**: {Data, Import, NSS(FK)}



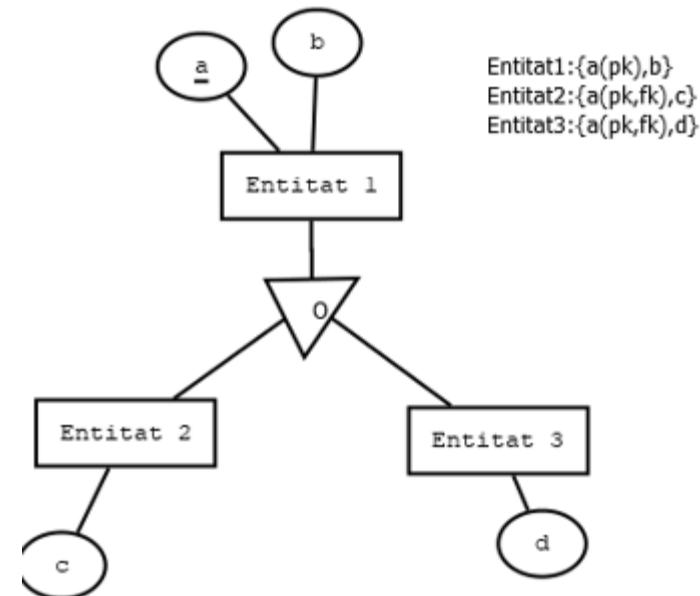
Relacional: Jerarquia Disjunta / Solapada Parcial

- Sigui disjunta o solapada, en els dos casos es comporta igual.

JERARQUIA DISJUNTA PARCIAL

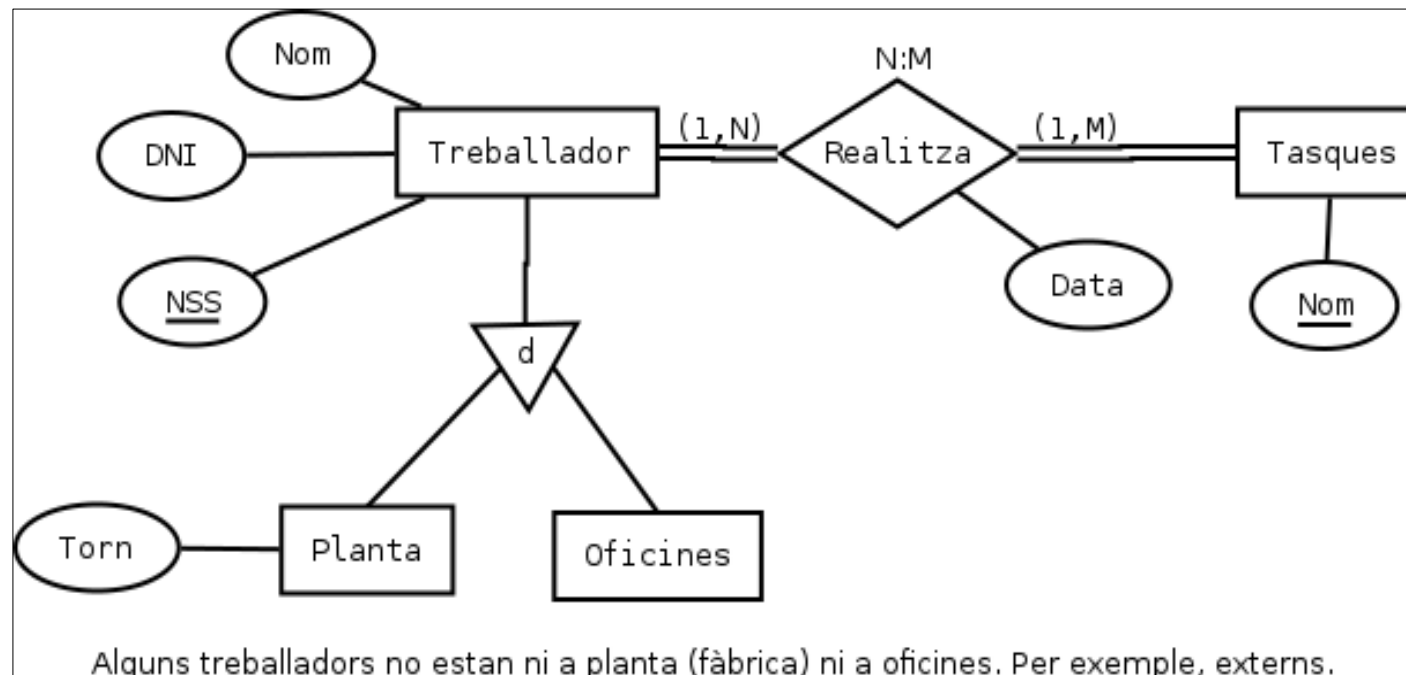


JERARQUIA SOLAPADA PARCIAL



-
- Diagrama de base de dades relacionals per a un sistema de gestió de versions de programari.
- Entitats:**
- Software**: Entitat principal amb atributs Nom (clau primària) i Fabricant.
 - Windows**: Entitat derivada de Software.
 - Linux**: Entitat derivada de Software.
 - Distribució**: Entitat amb atributs Nom i Versió (clau primària).
 - Versió**: Entitat amb atribut Nom (clau primària).
- Relacions:**
- Disponibles (Software -> Windows/Linux)**: Relació 1:N (un software té diverses versions disponibles).
 - Disponibles (Linux -> Distribució)**: Relació 1:N (una versió de Linux es distribueix en diverses formes).
 - Disponibles (Windows -> Versió)**: Relació 1:N (una versió de Windows es gestiona amb diverses versions).

Relacional: Jerarq. Disjunta Parcial (exemple)



- **Treballador:** {NSS(PK), DNI, Nom}
- **Planta:** {NSS(PK,FK), Torn}
- **Oficines:** {NSS(PK,FK)}
- **Tasques:** {Nom(PK)}
- **Realitza:** {TrebNSS(PK,FK), TasquesNom(PK,FK), Data}