Entidades Débiles

- La **interrelación débil entre un tipo de entidad débil y el tipo de entidad del cual depende es**, obligatoriamente, **una relación de muchos a uno**.
- Sus **reglas para claves ajenas** han de ser la siguientes

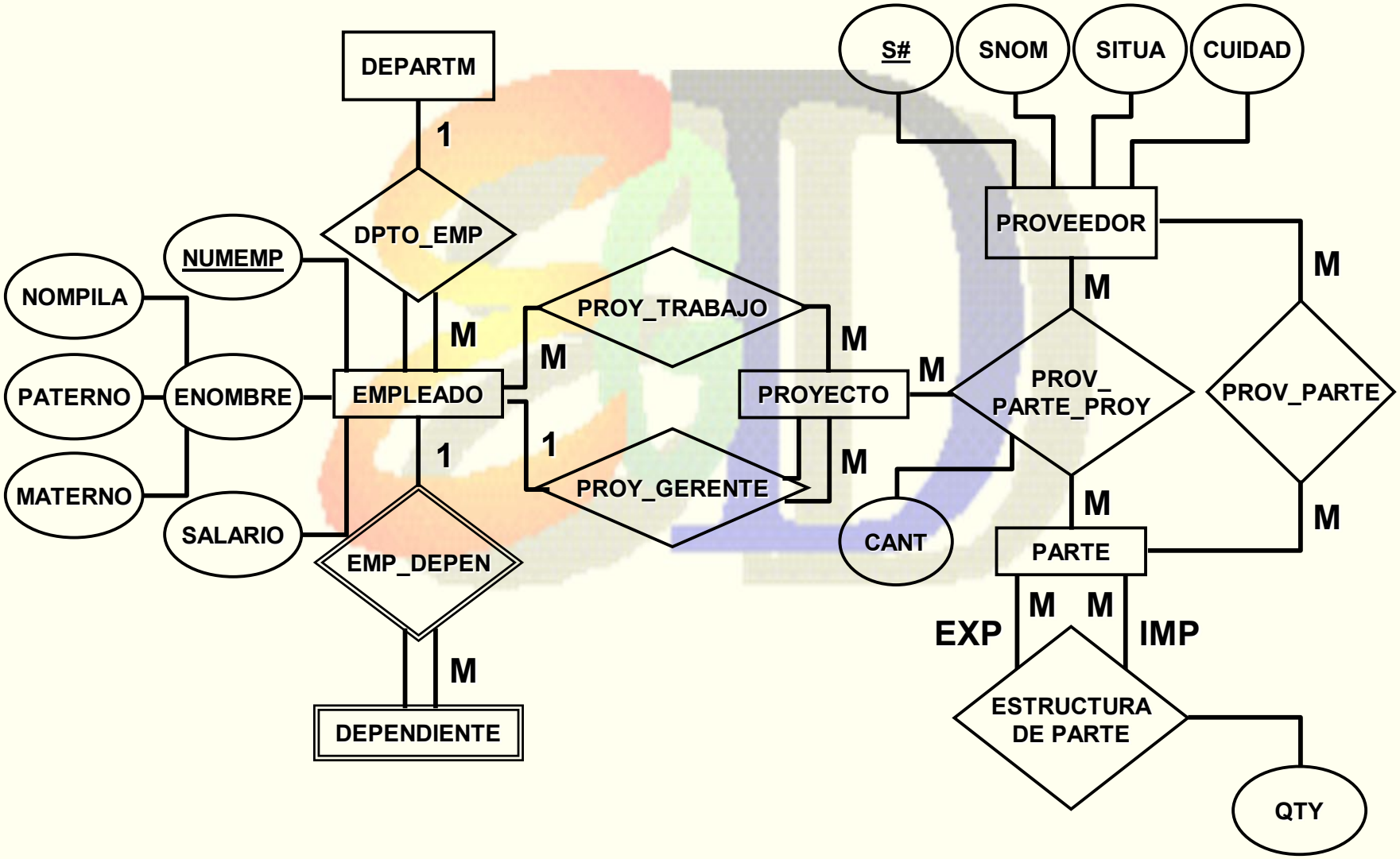
  delete . . . cascade

  update . . . cascade

- Estas dos reglas **junto a la no nulidad de la clave ajena capturan y reflejan la necesaria dependencia de existencia**.

  Sirva como ejemplo la interrelación **EMP_DEPEN**.

Entidades Débiles



Ejemplo de Entidad Débil

- Para representar la entidad débil **DEPENDIENTE** y la interrelación **EMP_DEPEN**:
  - **Habrá que utilizar una relación base DEPENDIENTE e incluir, como clave ajena de esta relación, el atributo NUMEMP, que es la clave primaria de la entidad EMPLEADO y de la cual depende la existencia de DEPENDIENTE.**

```
create table DEPENDIENTE
    (NUMEMP. . .  not null,
     NOMBRE_DEPEN. . . ,
                          . . . )
    primary key ( . . . )
    foreign key (NUMEMP) references EMP
        delete of EMP cascade
        update of EMP.NUMEMP cascade;
```

Clave Primaria de las Entidades Débiles

- Para la clave primaria:
  - Una posibilidad es tomar la **combinación de la clave ajena y el discriminante de la entidad débil del diagrama E/R**.

    Recuérdese que el discriminante de la entidad débil en el diagrama E/R proporciona unicidad **tan solo en el contexto del caso de la entidad regular diferenciada**.

    Esta posibilidad es la adecuada para el caso de dependencia en identificación.
  - Otra posibilidad sería **contar con un atributo de la entidad débil que sirva como clave primaria de la misma**. Caso que se produce cuando la **dependencia solo es en existencia**.

Ejemplo de Entidad Débil con dependencia en identificación

- En el ejemplo se ha elegido la primera posibilidad:

```
create table dependiente
        (NUMEMP. . . ,
         NOMBRE_DEPEN. . . ,
                                . . . )
        primary key (NUMEMP, NOMBRE_DEPEN)
        foreign key (NUMEMP) references EMP
            delete of EMP cascade
            update of EMP.NUMEMP cascade;
```

bajo la suposición de que **NOMBRE_DEPEN** es el **nombre del dependiente del empleado y que es el discriminante de la entidad dependiente en el diagrama E/R**, existiendo por lo tanto **dependencia en identificación**.

Ejemplo de Entidad Débil con dependencia en existencia

- Si existiese dependencia en existencia a causa de la presencia de un identificador total de los dependientes, NUMDEPEN:

```
create table dependiente
        (NUMEMP. . . not null,
         NUMDEPEN... ,
         NOMBRE_DEPEN  . . . not null,
                                        . . . )
        primary key (NUMDEPEN)
        alternative key (NUMEMP, NOMBRE_DEPEN)
        foreign key (NUMEMP) references EMP
                delete of EMP cascade
                update of EMP.NUMEMP cascade;
```

bajo la suposición de que **NOMBRE_DEPEN** sigue siendo el **nombre del dependiente del empleado y que es el discriminante de la entidad dependiente en el diagrama E/R**, aunque existe solo **dependencia en existencia** a causa de la existencia de NUMDEPEN, único entre todas las ocurrencias de dependiente.

Propiedades Multivaluadas

- Cada propiedad mostrada en el diagrama E/R corresponde a un atributo de la relación pertinente, excepto si se trata de una propiedad compuesta o también:
  - **Si la propiedad es multivaluada, se crea una nueva relación para ella a fin de respetar los principios básicos de normalización (1NF).**

    Por ejemplo, si cada proveedor tuviese un número indeterminado de líneas telefónicas se consideraría que la propiedad teléfono (TF#) sería multivaluada.

    En tal caso, **se añadiría una nueva relación STF**:

```
create table STF
        (S#  . . . ,
         TF#. . . )
        primary key (S#, TF#)
        foreign key (S#) references S
                delete of S cascade
                update of S.S# cascade;
```

Propiedades Multivaluadas

- Nótese que las reglas de claves ajenas coinciden con las de las relaciones que representan a entidades débiles.
- Sin embargo, en el caso de las propiedades multivaluadas, **sólo aparecerán dos atributos en la relación**:
  - **La clave primaria de la relación que representa a la entidad a la que pertenece la propiedad multivaluada.**
  - **La propiedad multivaluada en si.**

Propiedades Multivaluadas No Nulas

- Si la propiedad multivaluada TF# es no nula, significa que es ilegal que un proveedor no tenga teléfono.
- En tal caso, la tabulación sigue siendo esta:

```
create table STF
    (S#  . . . ,
     TF#. . . )
    primary key (S#, TF#)
    foreign key (S#) references S
        delete of S cascade
        update of S.S# cascade;
```

- y habría que añadir a la tabulación mostrada una restricción semántica adicional (**RSA**) que garantice que: *es ilegal que exista en la tabla S una fila con un valor de S# que no aparezca en niguna fila de la tabla STF*

Identificadores Multivaluados

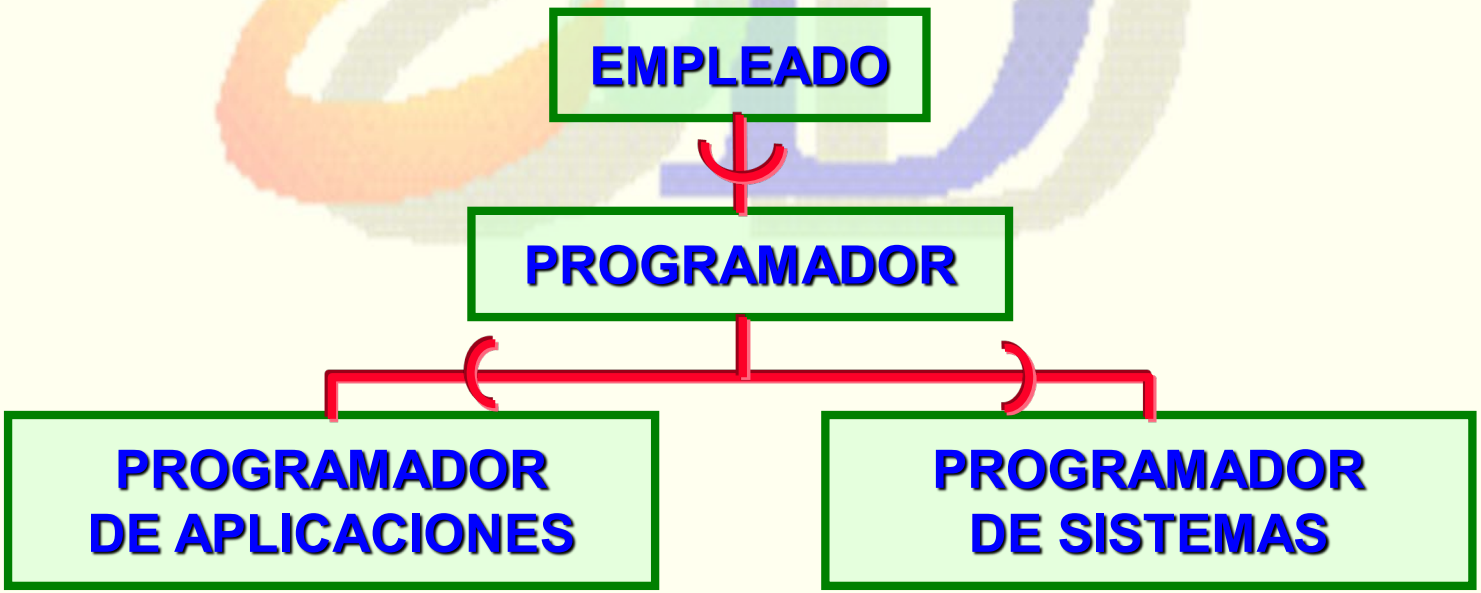
- Supóngase que la propiedad multivaluada TF# es un identificador,
  - porque es ilegal que un número de teléfono pertenezca a más de un proveedor
- En tal caso, la **clave primaria de la relación STF**, **será sólo el atributo TF#**.
  - **Ello se debe a sus características especiales de unicidad**.

```
create table STF
    (S#  . . . not null,
     TF#. . . )
    primary key (TF#)
    foreign key (S#) references S
        delete of S cascade
        update of S.S# cascade;
```

  - y, dado que se trata de un identificador y por tanto no puede ser nulo, habría que añadir a la tabulación mostrada una restricción semántica adicional (**RSA**) que garantice que: *sea ilegal que exista en la tabla S una fila con un valor de S# que no aparezca en niguna fila de la tabla STF*

Supertipos y Subtipos

- Cada uno de los tipos de entidad mostrados en la figura corresponderá a una relación base.
  - Cada una de estas relaciones **incluirá atributos correspondientes a las propiedades**.
  - **Aplicables a su posición dentro de la jerarquía de tipos** y, por lo tanto, por herencia: **a todas las posiciones subordinadas y a ninguna posición superior**.



Supertipos y Subtipos

- Para las relaciones **EMP** y **PGMR**, correspondientes a los tipos de entidad **empleado** y **programador**:

```
create table EMP                    create table PGMR
    (NUMEMP      . . . ,                (NUMEMP      . . . ,
     NUMDEPTO    . . . ,                 LENG        . . . ,
     SALARIO     . . . ,                             . . . )
                 . . . )                primary key ( . . . )
    primary key ( . . . )               . . .
    . . .
```

  - El atributo **LENG** representa la propiedad **lenguaje principal de programación**, que **sólo es aplicable** a los **empleados que son programadores**.
  - El atributo **SALARIO**, por el contrario, representa a una propiedad que **es aplicable** a **todos los empleados**.

Supertipos y Subtipos

- Las dos relaciones tienen la misma clave primaria: **NUMEMP**.
- La clave primaria de la relación del subtipo programador **funciona también como clave ajena** y **hace referencia a la clave primaria de la relación del supertipo empleado**.

```
create table EMP
      (NUMEMP      . . . ,
       NUMDEPTO    . . . ,
       SALARIO     . . . ,
                   . . . )
      primary key (NUMEMP)
      . . .
```

```
create table PGMR
      (NUMEMP      . . . ,
       LENG        . . . ,
                   . . . )
      primary key (NUMEMP)
      foreign key (NUMEMP) references EMP
            delete of EMP . . .
            update of EMP.NUMEMP . . .
```

Supertipos y Subtipos

- Las reglas para claves ajenas que aparecen en la definición de la relación que representa al subtipo constituyen **la forma lógica de representar la situación de jerarquía de tipos**.
- No deben permitirse los valores nulos ya que **claramente deben ser rechazadas las representaciones de propiedades de una ocurrencia de subtipo sin conocer a qué caso del supertipo se está refiriendo**.
- Por otro lado, cualquier eliminación o actualización de la clave primaria del supertipo **debe propagarse automáticamente a sus características como subtipo**.

```
create table PGMR
    (NUMEMP      . . . ,
     LENG        . . . ,
                 . . . , )
    primary key (NUMEMP)
    foreign key (NUMEMP) references EMP
        delete of EMP cascade
        update of EMP.NUMEMP cascade;
```

Supertipos y Subtipos

- Las relaciones **PGMR_APL** y **PGMR_SIS** **representan a los subtipos de programador**: **programador_de_aplicaciones** y **programador_de_sistemas**.

- Los atributos **ATRIBUTO1** y **ATRIBUTO2** **representan a propiedades que sólo son aplicables a los programadores de aplicaciones o a los de sistemas**, respectivamente.

```
create table PGMR_APL
    (NUMEMP . . . ,
     ATRIBUTO1      . . . ,
                    . . . , )
    primary key (NUMEMP)
    foreign key (NUMEMP) references PGMR
        delete of PGMR cascade
        update of PGMR.NUMEMP cascade;
```

```
create table PGMR_SIS
    (NUMEMP . . . ,
     ATRIBUTO2      . . . ,
                    . . . , )
    primary key (NUMEMP)
    foreign key (NUMEMP) references PGMR
        delete of PGMR cascade
        update of PGMR.NUMEMP cascade;
```