UDE
UNIVERSIDAD DE LA EMPRESA

Transformación de MER a Tablas

- Licenciatura e Ingeniería en Informática
- 2do. año

UDE
UNIVERSIDAD DE LA EMPRESA

Conceptos Generales

- Las estructuras consistirán en TABLAS
  - Sus columnas corresponden a ATRIBUTOS atómicos
  - Y sus filas serán los registros de datos.
- Un esquema de relación $R(A_1,\ldots,A_n)$
  - R será el nombre de la relación
  - $A_1 \ldots A_n$ serán los atributos con dominios $D_1 \ldots D_n$
  - Manejado por el DDL
- r(R)
  - Será una instancia de un esquema de relación R.
  - Conjunto de tuplas en un cierto instante del tiempo del contenido de la Base de Datos.
  - Manejado por el DML

UDE
UNIVERSIDAD DE LA EMPRESA

Clave Primaria

- Dada una tabla T con atributos ($a_1 \ldots a_n$)
  - Diremos que un subconjunto de atributos es CLAVE de T si dada una combinación de datos $a_i \ldots a_m$ el resto de los atributos queda unívocamente determinado.
  - La clave que se utiliza para determinar una Entidad del mundo real se denomina ***Clave Primaria*** de T $Pk(T)$
- Ej. Empleados (CI, nombre, apellido)
  - Instancias serán
    - <"1.723.458-2","Ana","López">
    - <"3.723.689-3","Beatriz","García">

UDE
UNIVERSIDAD DE LA EMPRESA

Claves Foráneas (Foreign Keys)

- Dado un esquema R, un conjunto de atributos X será una FK de R si:
  - Los atributos X coinciden en dominio con los de una clave Y de otro esquema S.
  - Los valores que tomará X en las tuplas de la instancia r(R) corresponden a valores de Y en la instancia s(S).
- Se dice que B posee clave foránea hacia A si todo elemento de la tabla B **debe** tener su atributo clave dentro de un atributo clave de alguna tupla de la tabla A

Fk(B) referencia Pk(A) y Pk(A) = Fk(B)

Integridad Referencial

- Reglas estáticas de definición de estructuras para garantizar su integridad.
- Si existe una Integridad Referencial entre R y S, donde R referencia a S, decimos que en R hay una FK sobre S.
- Ej.:
  - Todo Empleado pertenece a una Sección

EMPLEADOS(CI, Nombre, Dirección, NroSecc)

SECCIONES(NroSecc, NomSecc)

  - Nrosecc debe ser clave en Secciones
  - Empleados.Nrosecc referencia a Secciones.NroSecc (Toda tupla de empleados debe tener secciones existentes en Secciones)

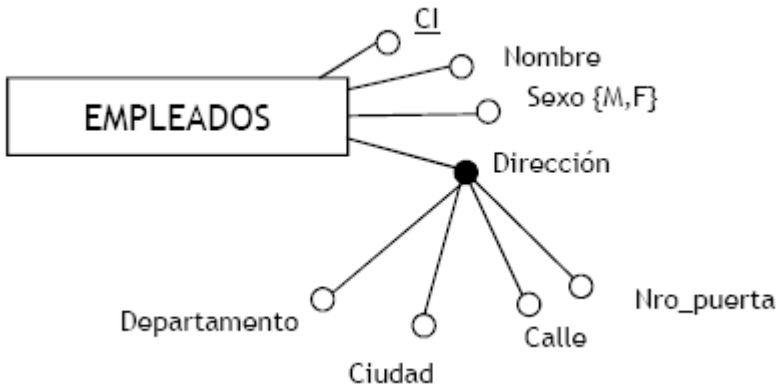
UDE
UNIVERSIDAD DE LA EMPRESA

Operaciones de modificación

- Dados el esquema $R(a_1..a_n)$ y su instancia r(R)
- Insert
  - Insert <$a_1 \ldots a_n$> into R
  - Las tuplas insertadas deben cumplir la integridad referencial.
- Delete
  - Delete from R where condicion
  - Debo considerar las posibles violaciones de la integridad referencial que pueda causar un borrado.
- Update
  - Update R set <atr>=valor,… where condicion
  - Debo considerar cuando las actualizaciones puedan violar la integridad referencial

Entidades Fuertes (I)

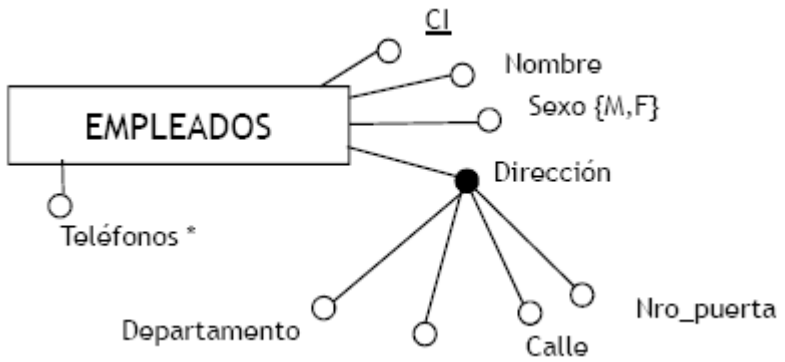
- Por cada **Entidad** se crea una Tabla
  - Cada atributo simple será un nuevo atributo de la tabla.
  - Cada atributo compuesto agregará nuevos atributos en la tabla por cada sub-atributo de la estructura.
  - El atributo determinante será clave e irá subrayada.



EMPLEADOS(CI, Nombre, Sexo, Departamento, Ciudad, Calle, Nro_puerta)

Entidades Fuertes (II)

- Los atributos multivaluados generan una nueva tabla.
- Tendrá un atributo por cada componente.
- Tendrá la clave de la entidad
- La clave serán todos sus atributos.

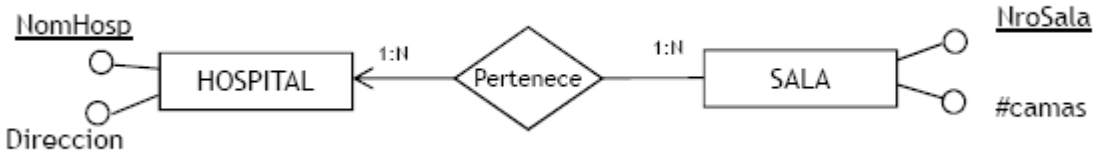


EMPLEADOS +

EMP_TELEFONOS(CI, nro_telefono)

Entidad Débil (I)

- Por cada **Entidad Débil** se crea una Tabla
  - Se incluye como atributos la clave primaria de la Tabla de la entidad fuerte.
  - No se crea una tabla para la relación
  - Debe haber una clave foránea desde la entidad débil hacia la fuerte.



HOSPITAL(NomHosp, Direccion)

SALA(NomHosp, NroSala, #camas)

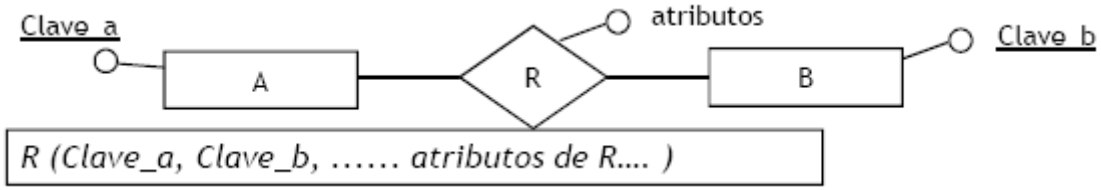
SALA.NomHosp REFERENCIA HOSPITAL.NomHosp

Entidad Débil (II)

- En el ejemplo, toda tupla de SALAS debe tener una pareja Hospital-Sala en alguna tupla de HOSPITALES con ese dato de Hospital
  - Nomhosp es clave en HOSPITALES
  - Nomhosp,Nomsala es clave en SALAS

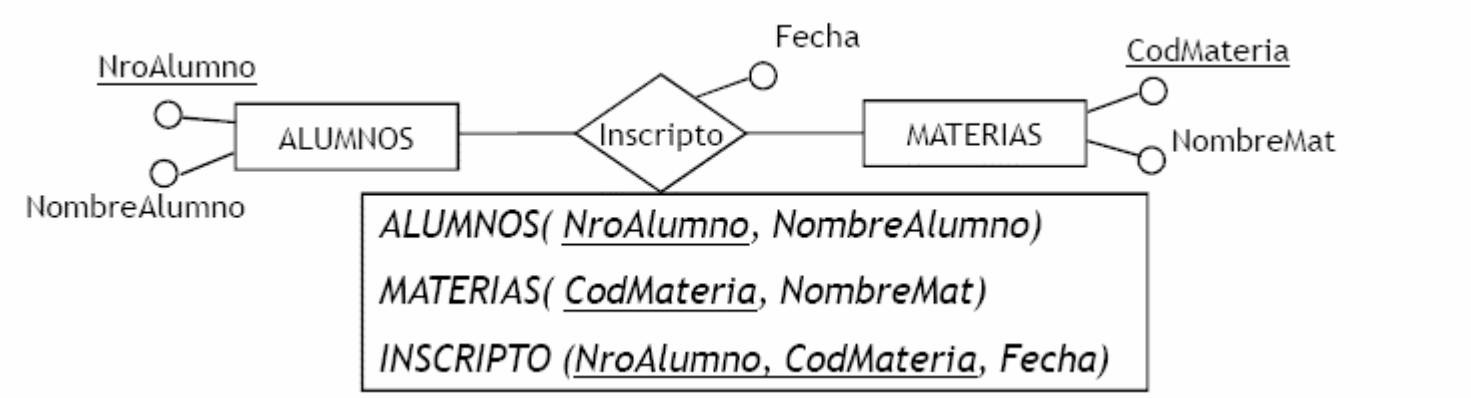
Relaciones Binarias N:N

- Para cada Relación Binaria con cardinalidad N:N se crea una tabla.
  - Se colocan las claves primarias de las tablas que representan a cada una de las entidades participantes.
  - Los atributos de la relación se tratan como si fuera una entidad.
  - La clave dependerá de la realidad, por lo general será la pareja de claves de ambas entidades.



Relaciones Binarias N:N

- Ejemplos: Un alumno se podrá inscribir a una materia varias veces.

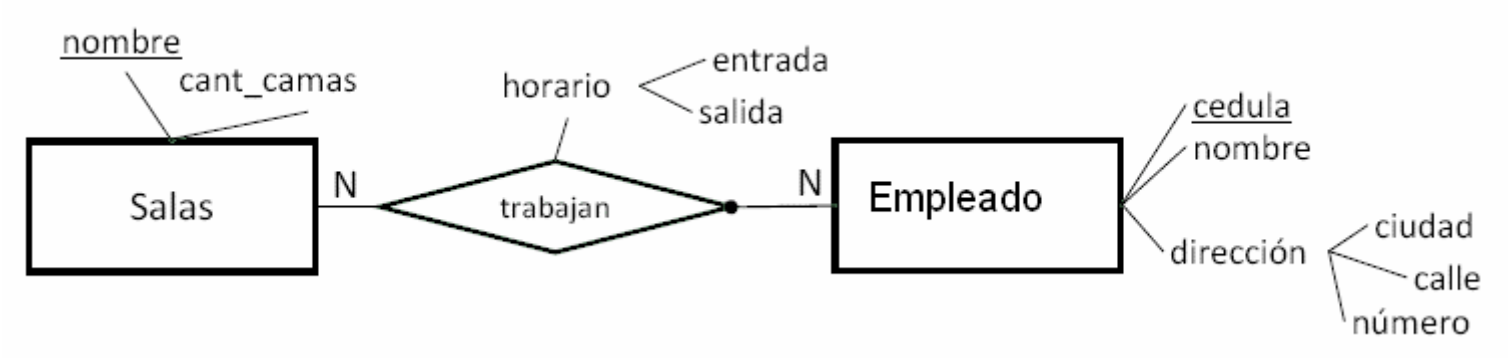


- Si tuviéramos la restricción de que se puede inscribir pero sólo una vez para cada fecha, entonces la relación INSCRIPTO sería:

INSCRIPTO (NroAlumno, CodMateria, Fecha)

Relaciones Binarias N:N

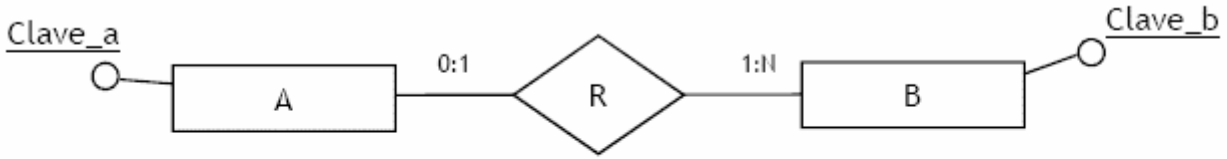
- Otro ejemplo:



- SALAS(nombre,cant_camas)
- EMPLEADO(cedula,nombre,ciudad,calle,numero)
- TRABAJAN(nombreSala,cedula,entrada,salida)

Relaciones 1:N

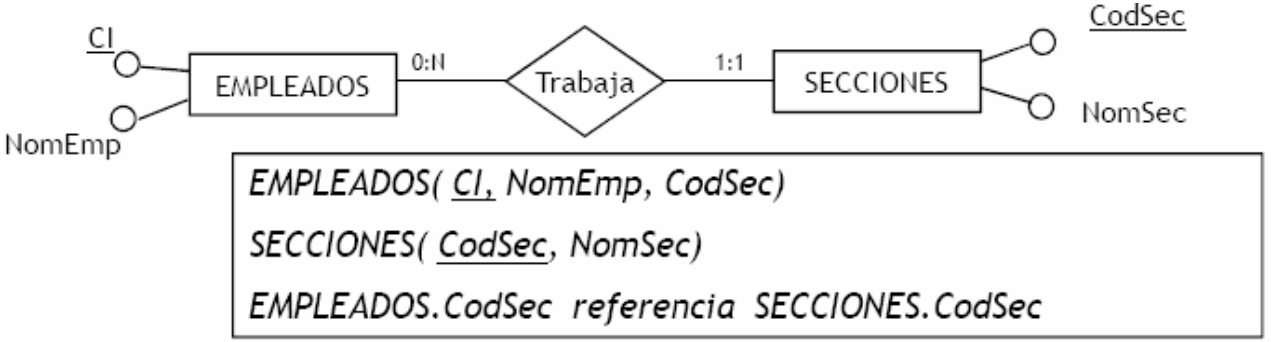
- Si la relación es 1:N puede NO CREARSE una nueva tabla
  - Se puede representar la relación en la tabla que representa a la entidad con cardinalidad N.
  - Se agrega a dicha tabla los atributos que son clave primaria de la tabla que representa la otra entidad.
  - Se agregarán también sus atributos simples o estructurados.
- Si la relación es PARCIAL sobre B, cuando un elemento de B no esté relacionado con alguno de A clave_a contendrá el valor null
- Si la relación es TOTAL sobre B, se debe definir una restricción de integridad referencial de B hacia A.



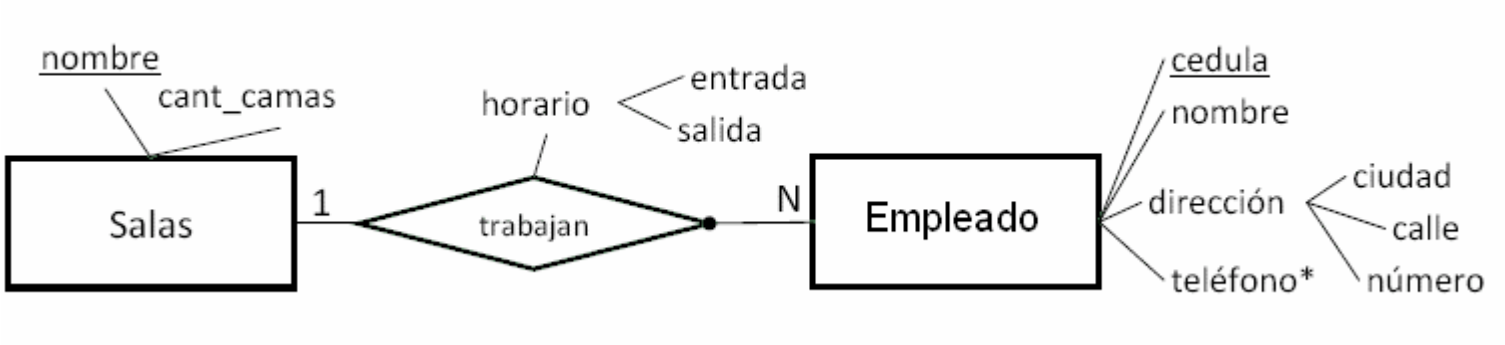
B (Clave_b, ... atributos de B, Clave_a)

Relaciones 1:N

- Ejemplos: Un empleado trabaja en una única sección



- N empleados trabajan en 1 sala



- SALAS(nombre,cant_camas)
- EMPLEADOS(cedula,nombre,ciudad,calle,numero, **nombreSala,hEntrada,hSalida**)

UDE
UNIVERSIDAD DE LA EMPRESA

Relaciones

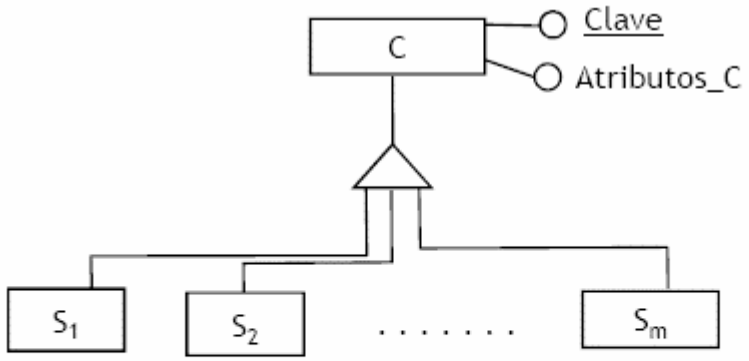
- Mínimos y máximos definidos
  - Cuando existen mínimos > 1 o cotas superiores no se pueden definir mediante restricciones estáticas.
  - Deben ser definidas por código. Ej. Triggers.

Agregaciones

- Se tratan como una relación
- Independientemente de las cardinalidades se debe implementar como una nueva tabla (ya que se relacionará con alguna otra entidad.

Categorizaciones (I)

- Existen diferentes implementaciones, dependiendo de la cantidad de atributos, solapamiento y completitud.

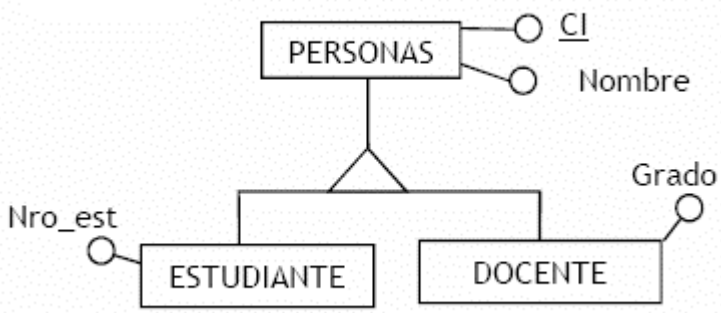


C(Clave_C, ...Atributos_C....)

S_1(Clave_C, ... Atributos_propios_S_1)

Categorizaciones (II)

- Caso 1
  - Se crea una taba para la super entidad.
  - Se crea una tabla por cada sub-entidad con referencia a la super entidad.
  - Debo considerar las restricciones de integridad referencial entre las tablas.



PERSONAS(CI, Nombre)

ESTUDIANTES(CI, Nro_est)

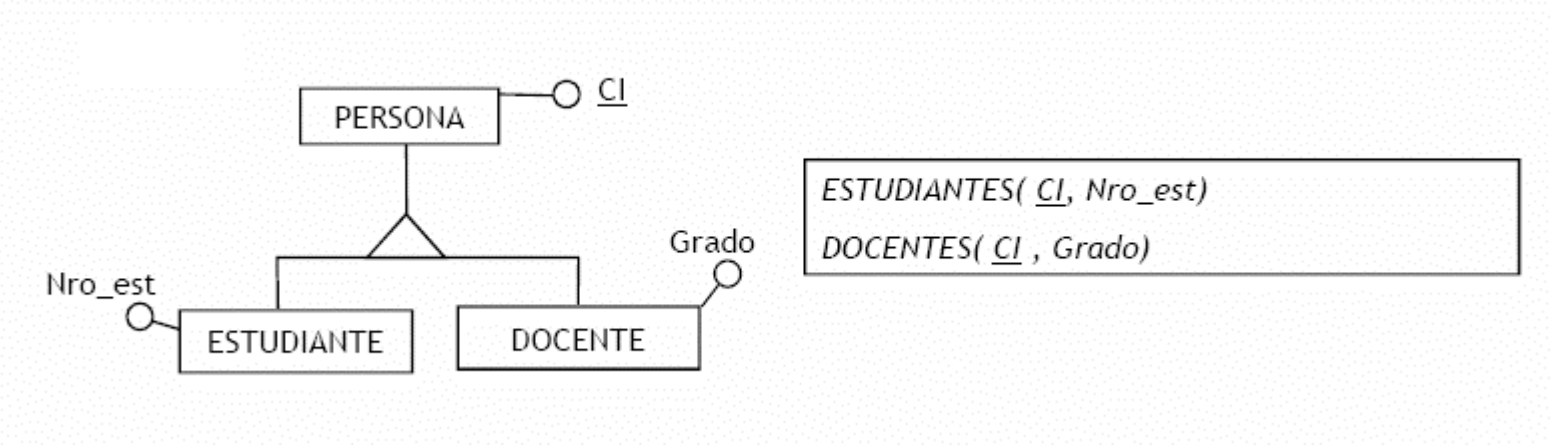
DOCENTES(CI , Grado)

ESTUDIANTES.CI referencia PERSONAS.CI

DOCENTES.CI referencia PERSONAS.CI

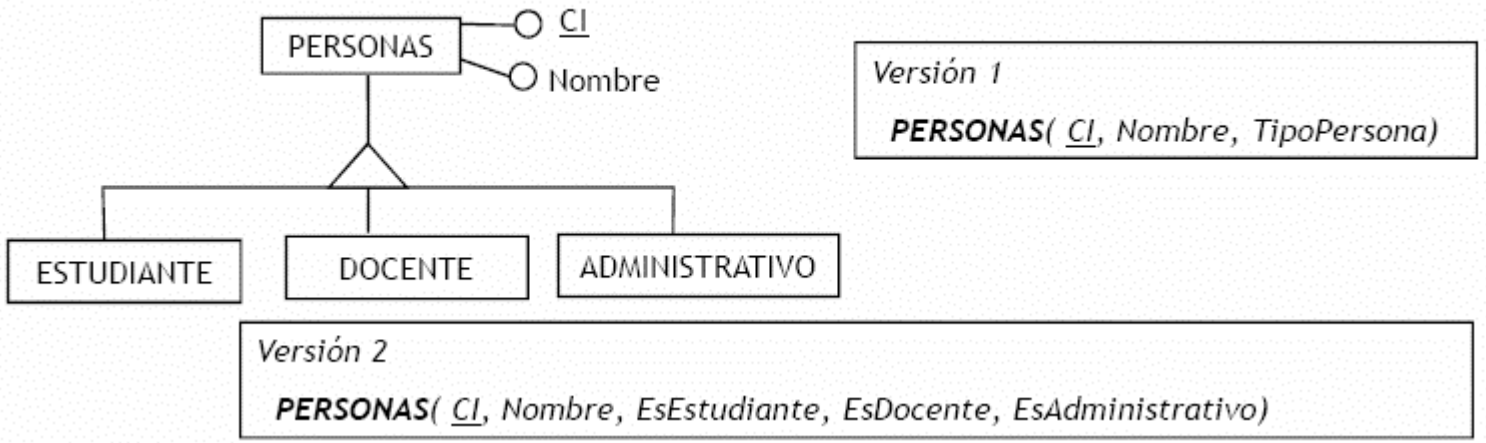
Categorizaciones (III)

- Caso 2 (relación no total y la super entidad no posee atributos propios)
  - Se crea una tabla por cada sub entidad más la clave de la super entidad.
  - No se necesita crear una tabla para la super entidad.



Categorizaciones (IV)

- Caso 3 (relación total y sub entidades sin atributos propios)
  - Dos posibles soluciones
    - Crear una tabla para la super entidad con un atributo extra de **tipo**.
    - Crear una tabla para la super entidad con atributos booleanos como sub entidades existan.



Categorizaciones (V)

- Ventajas y desventajas de las diferentes opciones:
  - Versión 1
    - No permite implementar solapamiento
    - Permite fácil agregado de nuevas sub entidades.
  - Versión 2
    - Permite solapamiento
    - Agregar nuevas sub entidades implica modificar el esquema de la super entidad.

UDE
UNIVERSIDAD DE LA EMPRESA

Categorizaciones (VI)

- Caso 4 (todas las entidades poseen atributos propios)
  - Crear una única tabla conteniendo la unión de los atributos de la super entidad más las sub entidades (considerando tipo u atributos booleanos).

    Puede contener valores nulos y exige restricción sobre datos correctos respecto a atributos particulares de las sub entidades.
  - Crear tablas para las sub entidades como entidades independientes que tendrán los atributos de la super entidad.
    Cubre solapamiento y totalidad, puede generar redundancia.
  - Crear tabla para la super entidad y tablas para las subentidades con la clave de la super entidad incluyendo restricciones de integridad desde las sub entidades a la super entidad.
    Cubre solapamiento y totalidad, evita redundancia.