Modelos de Datos

DEFINICIONES

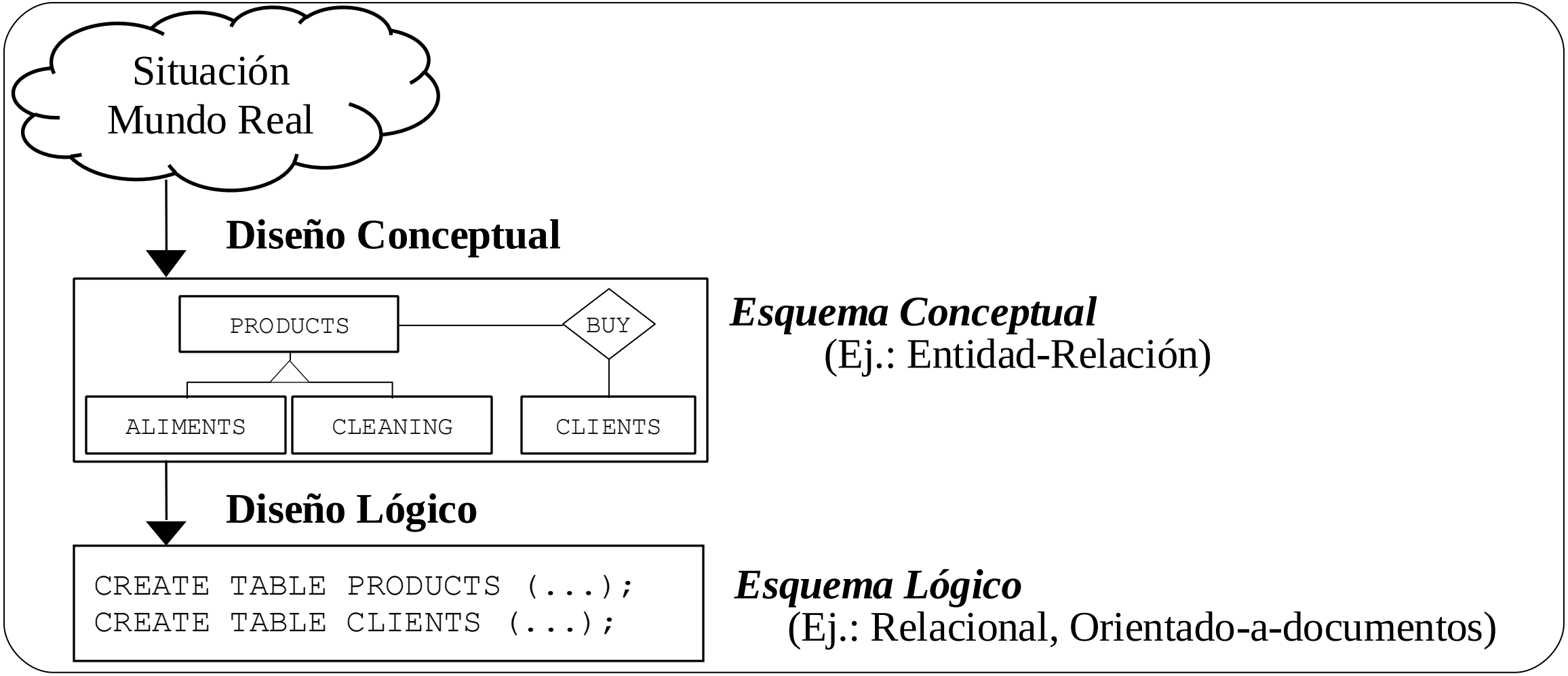
Modelos de Datos: Definición

- **¿Qué son?**
  - Lenguajes usados para especificar y manipular BDs.
  - Un Modelo de Datos permite expresar:
    - Estructuras
      - Elementos de los problemas.
        - Ej.: `CURSOS(nro_curso, nombre, horas)`
    - Restricciones
      - Reglas que deben cumplir los datos para que la base sea considerada válida.
        - Ej.: `(∀c ∃ CURSOS)(c.horas < 120)`
    - Operaciones
      - Insertar, borrar y consultar la BD.
      - Ej.: `INSERT INTO CURSOS (1911, "FBD", 90)`

Modelos de Datos: Clasificación

- **Según el nivel de abstracción:**
  - *Conceptuales*
    - Representan la realidad independientemente de cualquier implementación de BD.
    - Usado en etapa de Análisis.
  - *Lógicos*
    - Implementados en DBMSs.
    - Usado en etapas de Diseño e Implementación.
  - *Físicos*
    - Implementación de estructuras de datos.
    - Ej.: almacenamiento tabular o columnar, indexación por B Tree o Hash.

Modelos de Datos: Aplicación



Instancia de una Base de Datos

- **Conjunto de datos almacenados en una BD.**
- **Es el valor de una BD en un instante de tiempo dado.**
  - Si respetan todas las restricciones, se considera que la instancia es correcta.
- **Muy volátiles.**
  - Ej.: bases transaccionales.
- **Observar la diferencia con las ideas de instancia típicas de programación.**
  - En los lenguajes de programación tradicionales u orientados a objetos, una instancia es un elemento, aquí una instancia es un **CONJUNTO DE ELEMENTOS**.

Esquema de una Base de Datos

- **Describen qué atributos tienen los datos almacenados en la BD, cómo se relacionan esos datos entre sí, y qué restricciones de integridad deben cumplir**
  - Estructuras + Restricciones

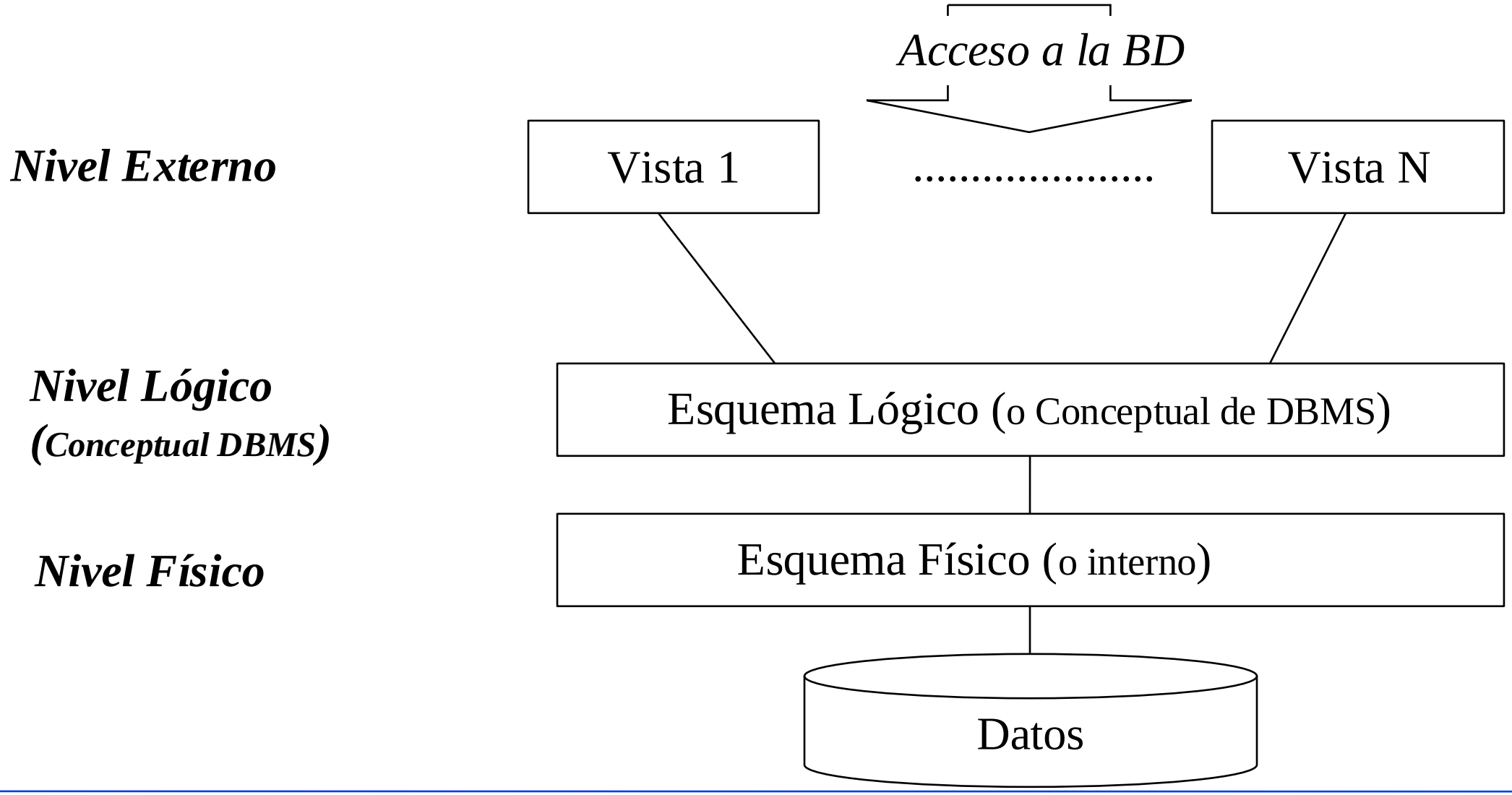
- **Por ejemplo:**
  - `CURSOS (nro_curso, nombre, horas).`
  - `ESTUDIANTES (CI, nombre, fecha_nacimiento).`
  - `TOMA_CURSO (nro_curso, CI).`

- **Muy estables.**

Arquitectura Lógica de un DBMS

- **Propiedades importantes de DBMSs:**
  - Control global único de la BD.
  - Separación entre esquema y aplicaciones.
    - Esquema: visión global de los datos de la realidad.
    - Aplicaciones: programas sobre la BD.
  - Soporte a diferentes visiones de los datos.
    - Usuarios/aplicaciones ven subconjuntos de la BD.
  - Independencia de datos.
    - Esquema lógico independiente de implementación.

Arquitectura en 3 Niveles



Independencia de Datos

- **Independencia Lógica.**
  - Independencia entre especificaciones de nivel Lógico y Externo.
  - Cambiar partes de esquema lógico sin afectar a los esquemas externos o a las aplicaciones.
    - Ej: renombrar o agregar un nuevo atributo en una tabla en una BD relacional

- **Independencia Física.**
  - Independencia entre especificaciones de nivel Lógico y Físico.
  - Cambiar implementaciones sin afectar el esquema Lógico.
    - Ej: particionamiento horizontal de una tabla en una BD relacional

Lenguajes e Interfases en Ambientes BD

- **Provistos por DBMS:**
  - Definición de esquema:
    - VDL (o SSDL) - View Definition Language.
    - SDL - Storage Definition Language.
    - DDL - Data Definition Language.
      - Suele englobar estos tres lenguajes.
  - Manipulación de la BD:
    - DML - Data Manipulation Language.
      - Modificaciones en instancias.
    - QL - Query Language.
      - Subconjunto del DML, sólo para consultas.

Lenguajes e Interfases en Ambientes BD

- **Tipos de QL:**
  - Declarativos.
    - Se especifica qué propiedad cumplen los datos.
    - No se especifica cómo se recuperan de la BD.
    - Suelen recuperar conjuntos de items (registros).
    - Es el DBMS el que define el plan de ejecución.
  - Procedurales.
    - Se especifica un algoritmo que accede a estructuras del esquema lógico y recupera los datos item por item (registro a registro).

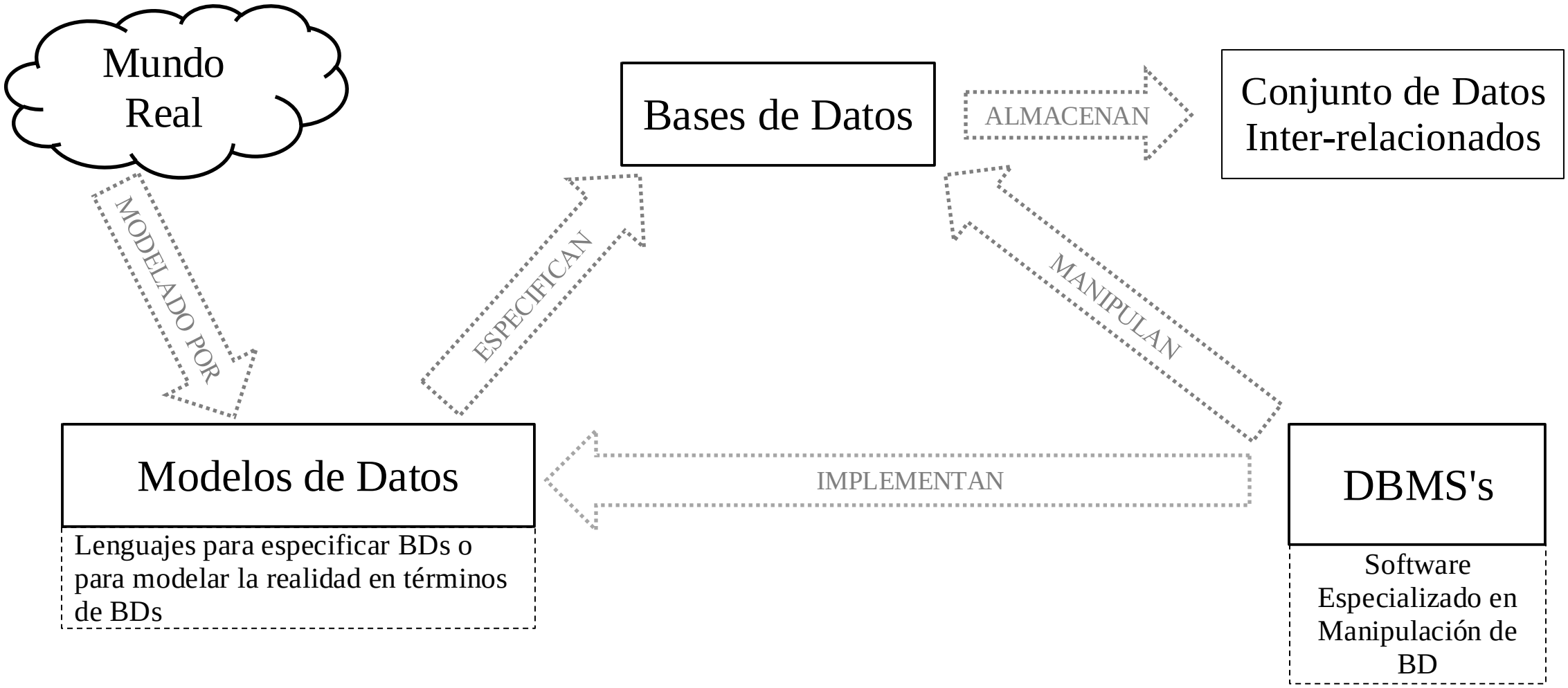
Lenguajes e Interfases en Ambientes BD

- **Lenguajes de programación:**
  - Lenguajes host (anfitrión):
    - Lenguajes de uso general (C, COBOL, Python, Java, etc) en el cual se embeben sentencias de DML.
      - a veces utilizando librerías auxiliares.
    - Se tiene un pre-procesador que traduce el programa con DML embebido en un programa “puro”.
    - PROBLEMAS: object-relational impedance-mismatch
  - Lenguajes 4GL:
    - Lenguajes procedurales orientados a acceso a BDs.
    - Conexión privilegiada con DMLs, reduce el impedance-mismatch.

Lenguajes e Interfases en Ambientes BD

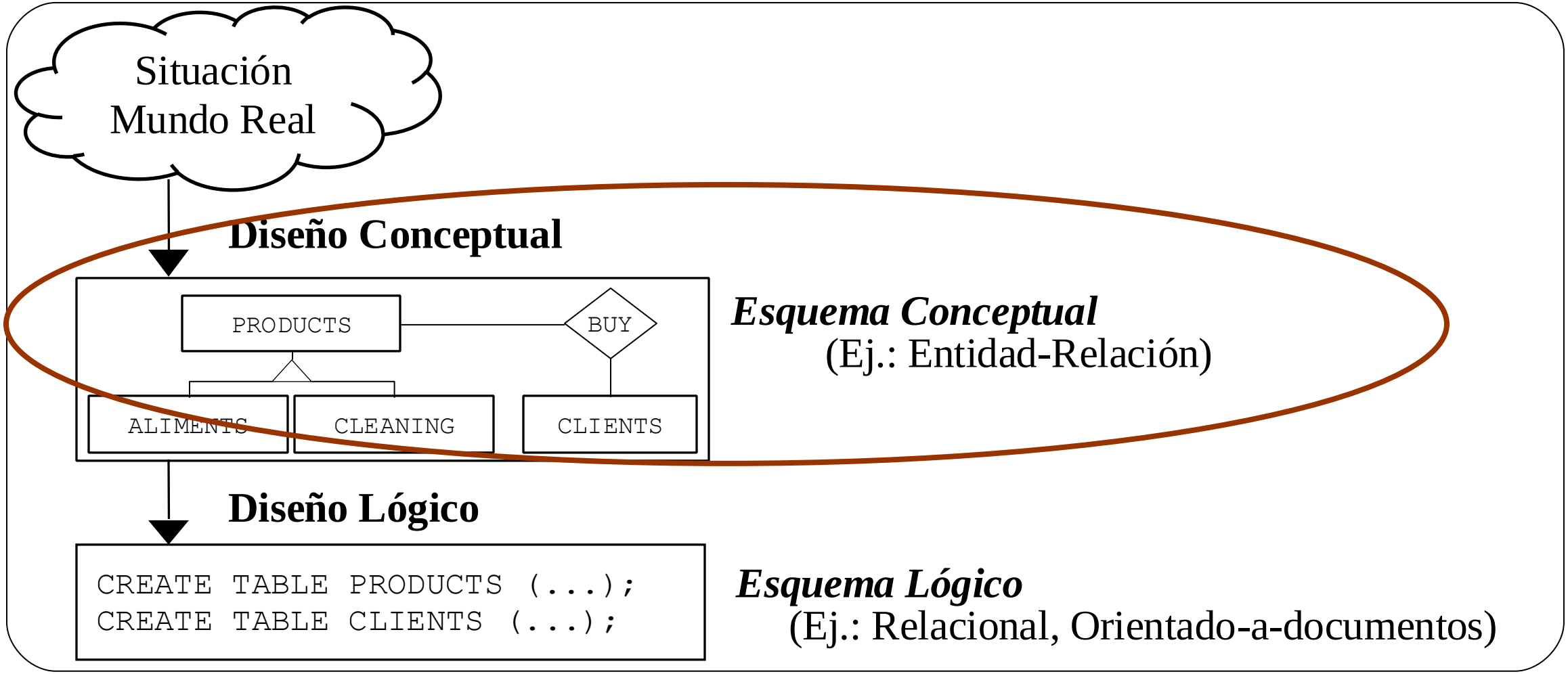
- **Interfaces especializadas:**
  - Interfaces gráficas de consulta.
    - se visualizan las estructuras en forma gráfica.
    - resultados como gráficas (torta, líneas, etc).
  - Interfaces de Lenguaje Natural.
    - se procesan frases y se traducen al QL.
  - Interfaces para Administración.
    - ambientes especializados.

Resumen de los Elementos de Bases de Datos



MODELADO CONCEPTUAL

Fases y resultados en el diseño de BDs



Modelado Conceptual

- **Qué es ?**
  - Primer etapa en diseño de una BD.
  - Actividad en la cual se construyen esquemas conceptuales de una realidad.
- **Sub-etapas:**
    - Estudio del problema real.
    - Especificación usando un lenguaje de muy alto nivel.
    - Validar resultado.
- **Resultado:**
  - Esquema Conceptual.
- **Lenguajes usados:**
  - Modelos Conceptuales.

Modelado Conceptual

- **Los Modelos Conceptuales:**
  - Modelos de datos de muy alto nivel.
    - No interesan los aspectos de implementación.
  - En general se concentran en estructuras y restricciones de integridad.
    - Se concentran en describir el dominio del problema.
  - Suelen tener una representación gráfica asociada.

- **Algunos Modelos Conceptuales de Datos:**
  - Modelo Entidad-Relación [1976].
  - Modelos ER Extendidos [´80s y ´90s].
  - Modelos Multidimensionales [2000].

Modelos Conceptuales – Conceptos Básicos

- **Elementos:**
  - Conjuntos
    - Los elementos de interés aparecen agrupados o clasificados en conjuntos de acuerdo a sus características (Ej.: Personas, Cursos, etc).
  - Relaciones entre Conjuntos
    - Conjuntos de parejas, ternas, cuaternas, etc. de elementos de los conjuntos anteriores. (Ej.: Estudiantes aprueban cursos, docentes dictan cursos, etc.).
  - Restricciones de Integridad.
    - Condiciones que indican cuando un elemento o una pareja puede o no puede pertenecer a un conjunto o relación. (Ej.: Todos los estudiantes deben ser mayores de 18 años, etc.)

Modelos Conceptuales – Términos Comunes

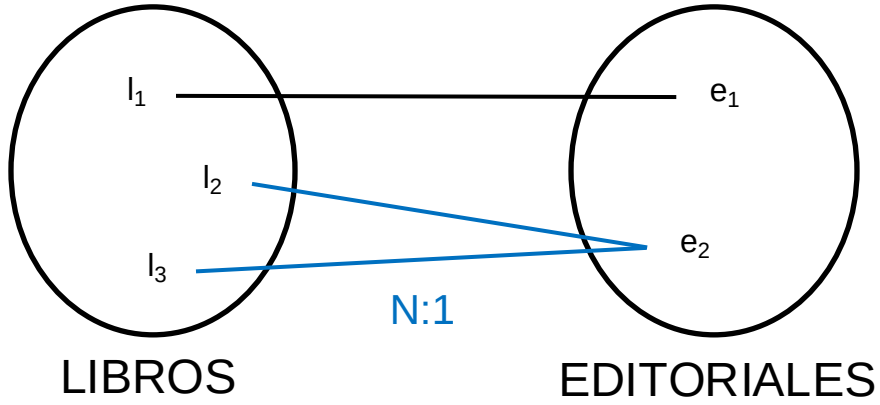
- **Atributo**
  - Característica que nos interesa de un determinado elemento de la realidad.
  - Ej.: Título de un libro
  - ¿Qué otro ejemplo se les ocurre?

Modelos Conceptuales – Términos Comunes

- **Cardinalidad**

Cuantos elementos de un conjunto pueden estar relacionados con un elemento del origen.

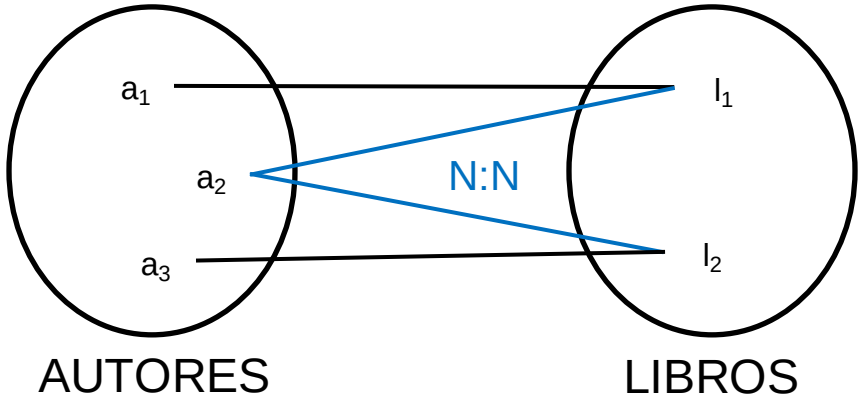
  - N:1
    - Dada una relación entre dos conjuntos A y B, se dice que tiene cardinalidad N:1 si dado un elemento cualquiera de A, puede haber en la relación sólo una pareja con ese elemento (Ej.: Libros y Editoriales)



Es una restricción de integridad

Modelos Conceptuales – Términos Comunes

- **Cardinalidad**
  - N:N
    - Dada una relación entre dos conjuntos A y B, se dice que tiene cardinalidad N:N si dado un elemento de A puede haber cualquier cantidad de elementos de B (Ej.: Libros y Autores)



Principios del Modelado Conceptual

- **Principio del 100%:**
  - El esquema conceptual asociado a un problema debe representar todos sus aspectos.

- **Principio de Conceptualización:**
  - El esquema conceptual no debe incluir ningún elemento asociado a la implementación del esquema, así como ningún elemento orientado a la performance de la futura BD.
  - Es un problema para el futuro, un desafío para los ansiosos

Modelo Entidad-Relación

- **Modelo Conceptual muy usado.**
  - Propuesto por Chen en 1976.
  - Existe una gran variedad de “dialectos” y variantes del Modelo ER.
  - Los modelos OO se inspiran y toman ideas de él, por lo que presentan similaridades.

- **Sus conceptos básicos:**
  - Entidad : elemento de la realidad.
    - Por ejemplo: Estudiantes, Cursos, Docentes.
  - Relación : asociación entre elementos.
    - Por ejemplo: Cursa, Dicta

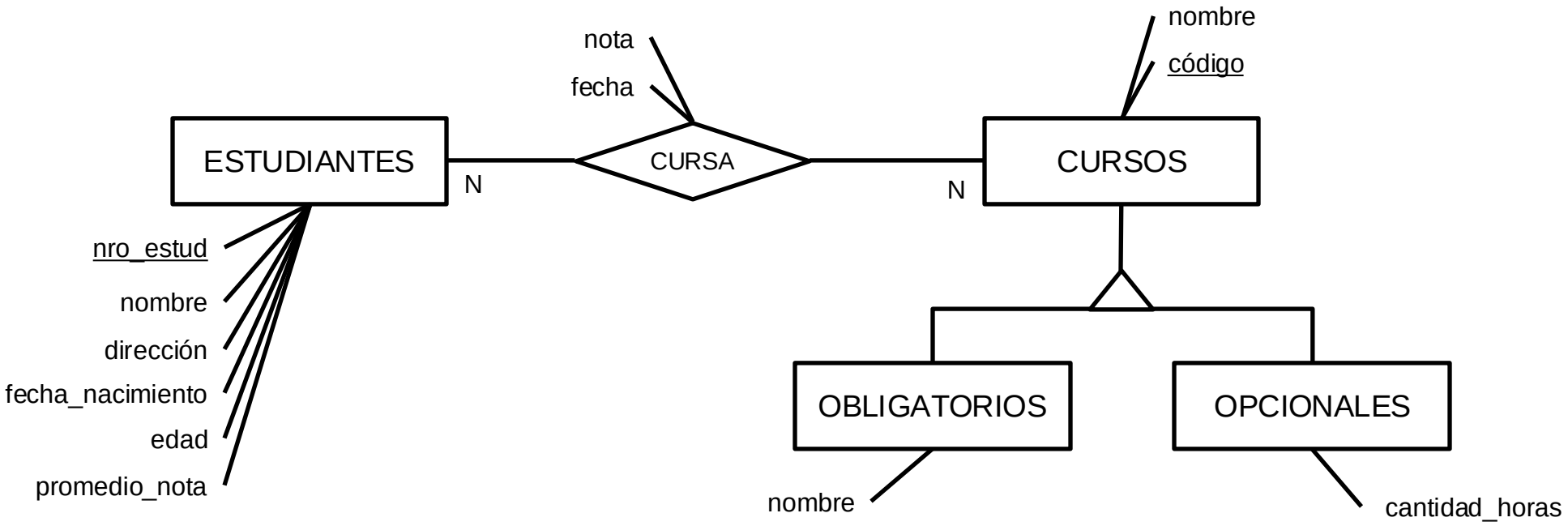
Modelo Entidad-Relación

- **¿Qué dialecto usaremos en el curso?**
  - El modelo gráfico utilizado en el curso Fundamentos de Bases de Datos
  - Los conceptos descriptos en las secciones correspondientes de Elmasri-Navathe.
  - Coincide bastante con el del Silberchatz y Korth.

- **Elementos principales:**
  - Entidades, Relaciones, Atributos
  - Generalización, Agregación, Entidad Débil.

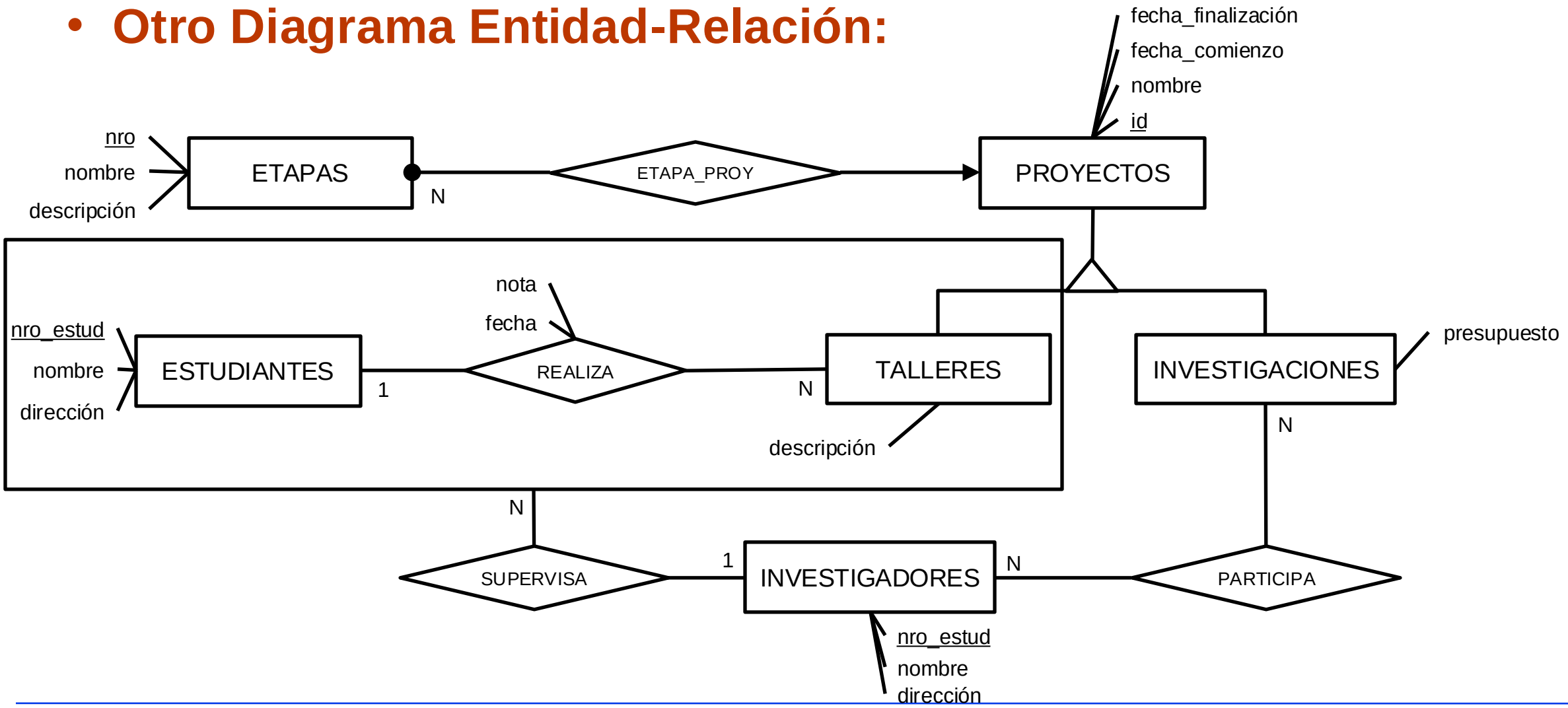
Modelización Conceptual – Ejemplo 1

- **Un Diagrama Entidad-Relación:**
  - Los estudiantes realizan cursos que pueden ser obligatorios u opcionales.



Modelización Conceptual – Ejemplo 2

- **Otro Diagrama Entidad-Relación:**



Modelo Entidad-Relación

- **Tiene un DDL Gráfico orientado a la representación de estructuras y restricciones de integridad.**

- **Tiene 2 elementos básicos:**
  - Diagrama Entidad-Relación.
    - Representa las estructuras y restricciones estructurales (Ej.: claves, entidades débiles, cardinalidades, etc).
  - Restricciones no estructurales.
    - Fórmulas lógicas o de conjuntos, representando las restricciones que no pueden ser expresadas en el diagrama por su complejidad o por falta de notación.

Modelo ER - Constructores

- **Resumen de principales constructores :**
  - Entidades:
    - modeliza conjuntos de elementos de la realidad.
  - Relaciones:
    - modeliza asociaciones entre objetos.
  - Atributos:
    - modeliza propiedades de Tipos de Entidades o de Relaciones.
  - Agregaciones:
    - representa un Tipo de Relación como un Tipo de Entidad.
  - Especializaciones (o Categorizaciones):
    - modeliza sub Tipos de Entidades.

Diagrama vs Esquema

- **No confundir el esquema ER de la base con el diagrama ER de la base:**
  - El diagrama es una representación gráfica de la estructura de los datos de la base.
  - El esquema es la estructura de datos representada por el diagrama. Esto lo vamos a ver más adelante en el curso.

- **El lenguaje tiene una semántica bien definida.**
  - Los diferentes dialectos sólo cambian los símbolos pero no el significado.

Diagrama Entidad-Relación

- Los conjuntos de entidades se presentan con un rectángulo con el nombre, del que “cuelgan” los atributos.

- Las relaciones se representan con un rombo con el nombre y que está conectado con los conjuntos de entidades que relaciona.

- Hay un conjunto grande de restricciones que se pueden imponer sobre el diagrama con diferentes notaciones.

Modelo ER – Uso Práctico

- **Cómo aplicar un modelo de datos para representar una determinada realidad se puede resumir en los siguientes pasos:**
  1. Identificar los elementos de nuestro problema.
  2. Identificar las relaciones entre los objetos.
  3. Representar las propiedades que nos interesan de nuestros objetos.
  4. Determinar otras restricciones que deseamos imponer.

Caso de Estudio

En un hospital se tiene un registro de pacientes, un registro de personal y uno de salas con funcionarios que trabajan en esas salas y con pacientes internados en esas salas.

Del personal nos interesa el número de empleado, el nombre, la dirección y el teléfono.

Sabemos que dos empleados no tienen el mismo numero.

De los pacientes nos interesa el número de registro (le es asignado cuando ingresa) y el nombre mientras que de las salas nos interesa el nombre y la cantidad de camas que tiene.

También se sabe que un empleado trabaja en una única sala y que en una sala trabajan varios empleados. Lo mismo ocurre con los pacientes.

Caso de Estudio – Identificación de Elementos y Relaciones

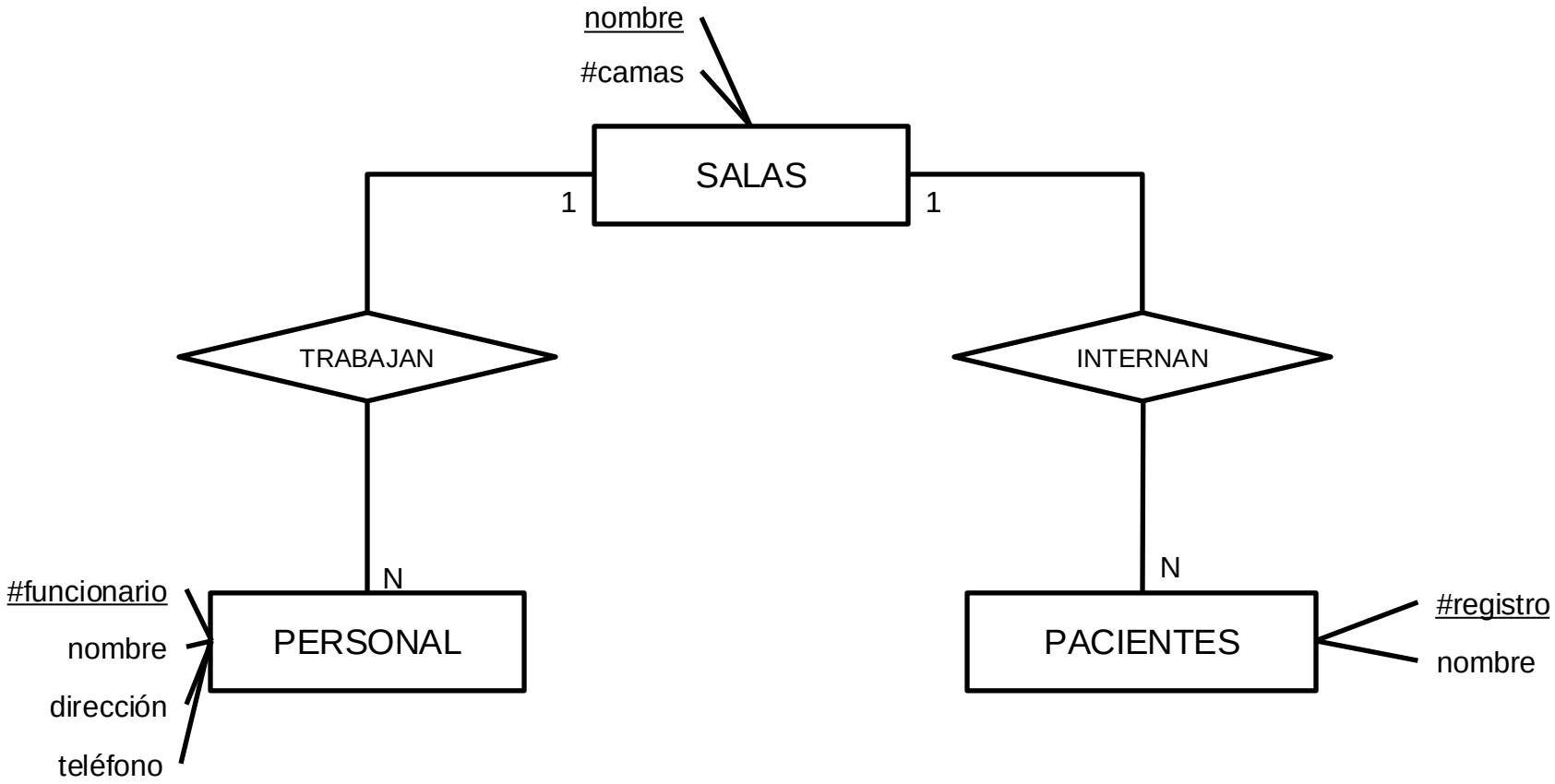
- Conjuntos de elementos de la realidad:
  - Pacientes, Salas, Personal

- Relaciones entre esos conjuntos:
  - Los Pacientes están Internados en las Salas y el Personal Trabaja en las Salas.

- Características que interesan de los objetos:
  - Personal: nro. de funcionario, nombre, dirección y teléfono
  - Pacientes: nro. de registro, nombre
  - Salas: nombre, cantidad de camas

- Restricciones:
  - Un empleado trabaja en una única sala y en una sala trabajan varios empleados. Un paciente está internado en una sola sala pero en una sala hay varios pacientes.

Caso de Estudio - Primer Nivel



Entidades

- Una Entidad es un elemento individual distinguible de nuestra realidad

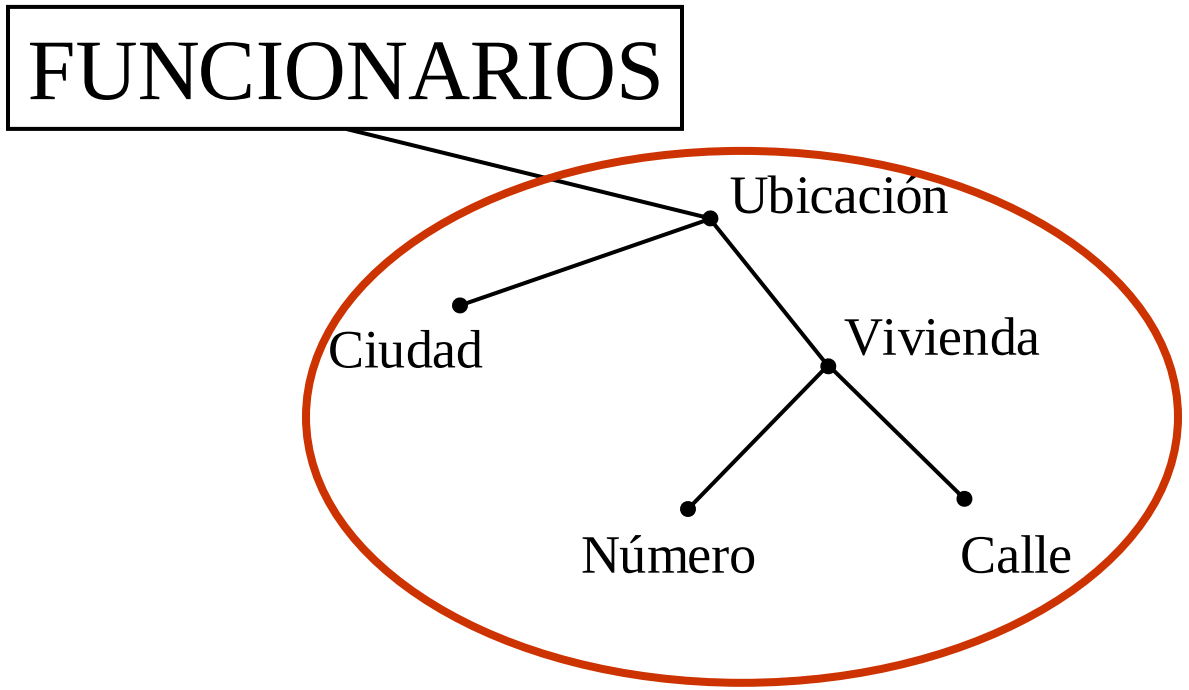
- Las entidades se agrupan en Conjuntos de Entidades o Tipos de Entidades

Atributos

- Un Atributo es una función tal que dado un elemento de un determinado conjunto de entidades devuelve un valor de un determinado conjunto de valores.

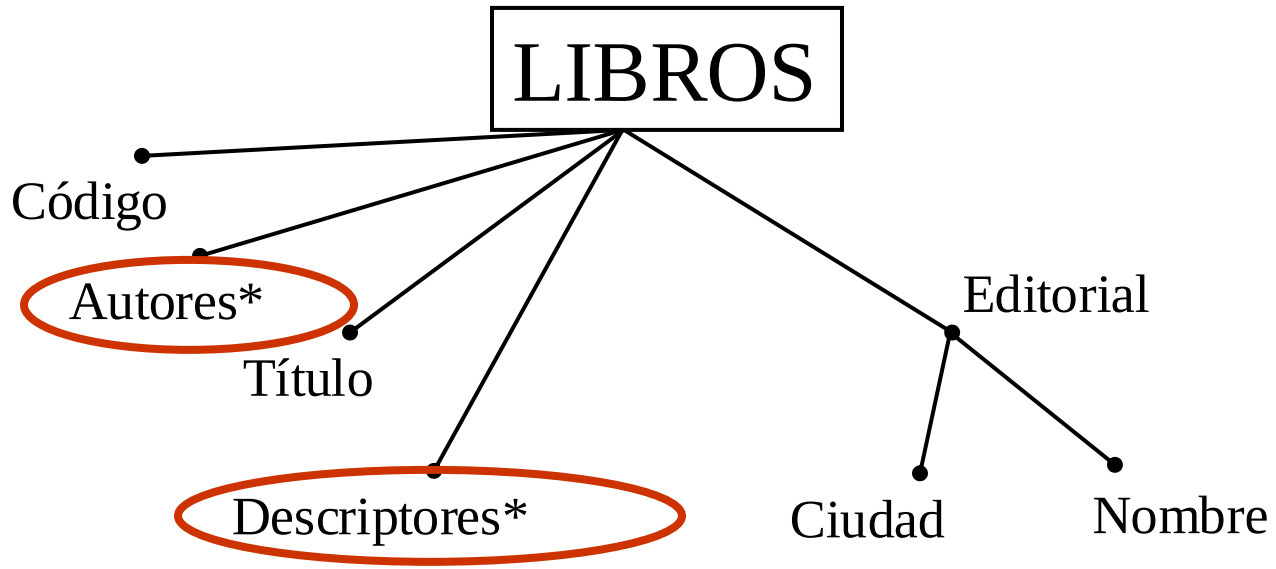
Atributos Estructurados

- Permiten representar atributos compuestos que están formados por varias partes independientes.



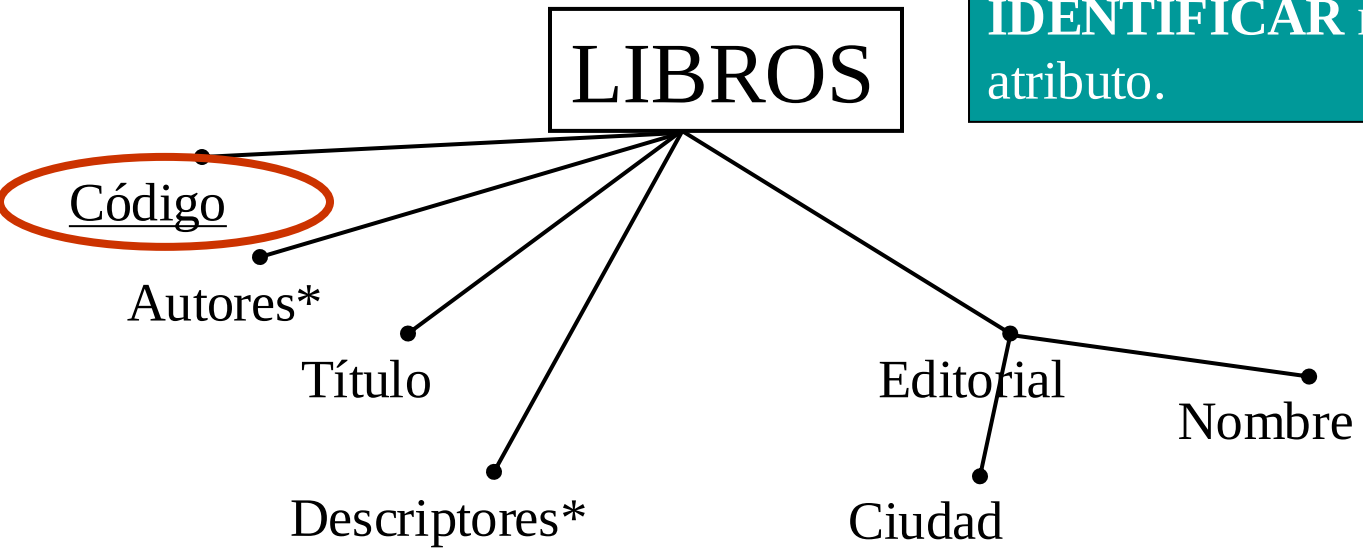
Atributos Multivalorados

- Son funciones que devuelven un valor de tipo conjunto de otro dominio.



Restricciones sobre Conjuntos de Entidades

- Se dice que un atributo es Determinante cuando no pueden existir dos entidades en el conjunto que tengan el mismo valor en ese atributo.



Las entidades del conjunto se pueden **IDENTIFICAR** mediante los valores de ese atributo.

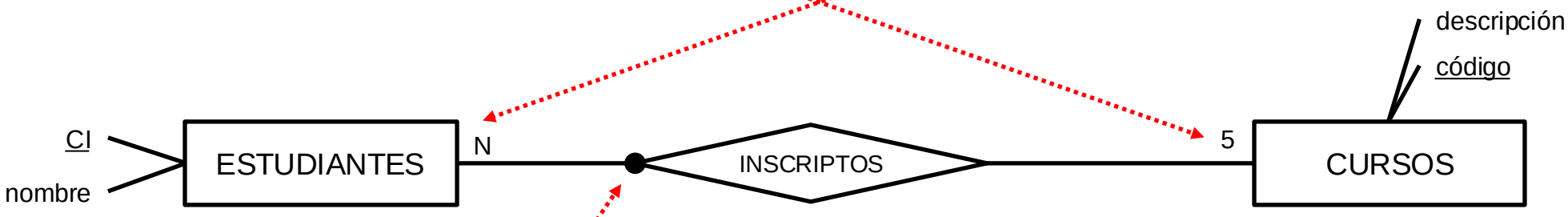
Es una restricción porque dice cuáles conjuntos de entidades son instancias válidas para este esquema y cuáles no lo son.

Restricciones sobre Relaciones

- Cardinalidad
- Totalidad

Dado un estudiante E ¿En cuántos cursos puede estar inscripto como máximo?

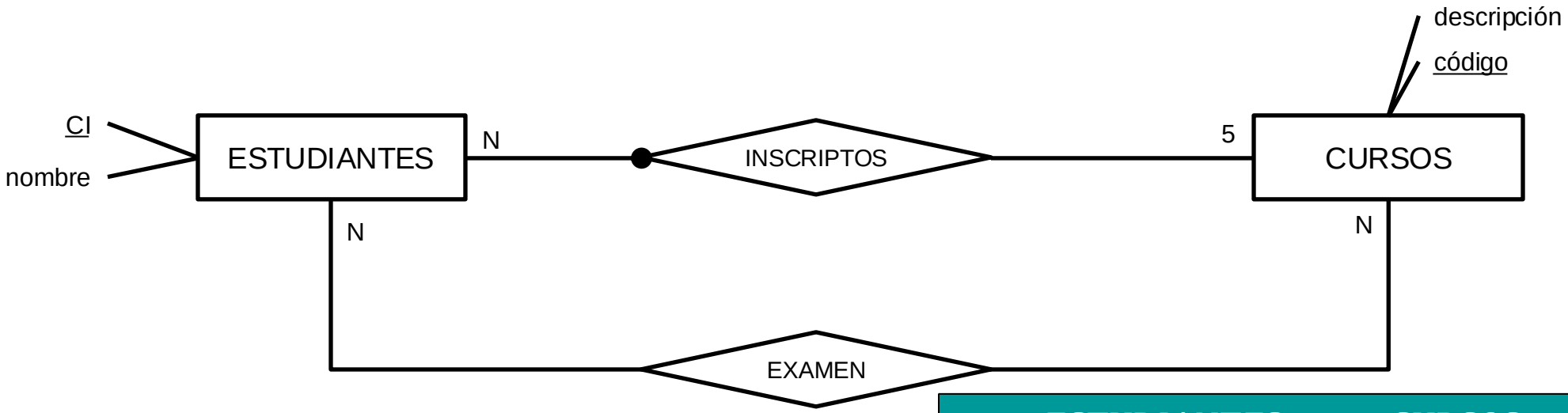
Dado un curso A ¿Cuántos estudiantes puede tener inscriptos como máximo?



Todos los estudiantes deben estar inscriptos en algún curso.

Restricciones sobre Relaciones

- Cardinalidad
- Totalidad



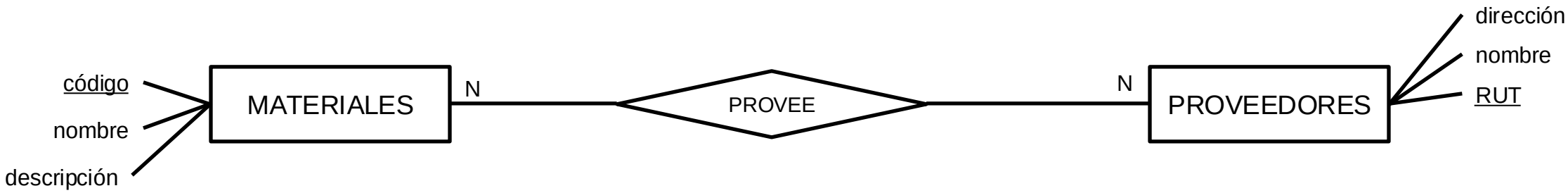
Un estudiante no puede rendir un examen de un curso en el que no está inscripto.

$\forall e \in ESTUDIANTES, \forall c \in CURSOS$
$(<e,c> \in EXAMEN \Rightarrow <e,c> \in INSCRIPTOS)$

Otra versión:
$EXAMEN \subseteq INSCRIPTOS$

Atributos de Relaciones

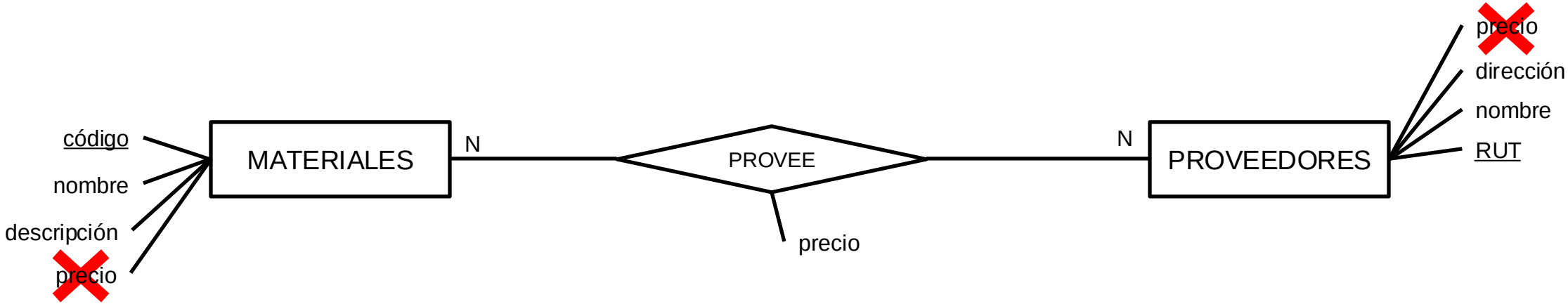
- Se tiene información de **materiales** y **proveedores** indicando qué material provee cada proveedor.
- Del material, se conoce el código que lo identifica, el nombre y una descripción.
- Del proveedor se conoce su RUT, su nombre y su dirección.
- Cualquier material puede ser provisto por cualquier proveedor.



Atributos de Relación

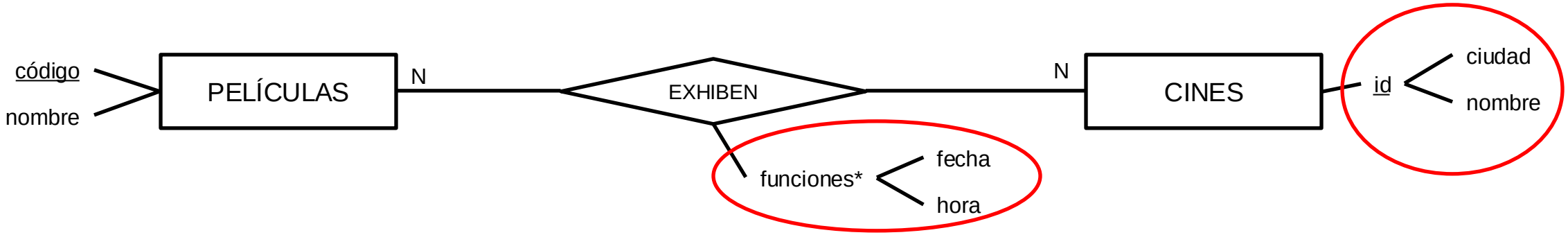
- Se conoce también el precio al que un proveedor provee un producto.
  - ¿Es un atributo del proveedor?
    - No porque depende del producto.
  - ¿Es un atributo del meterial?
    - No porque depende del proveedor.

Es un atributo de la Relación



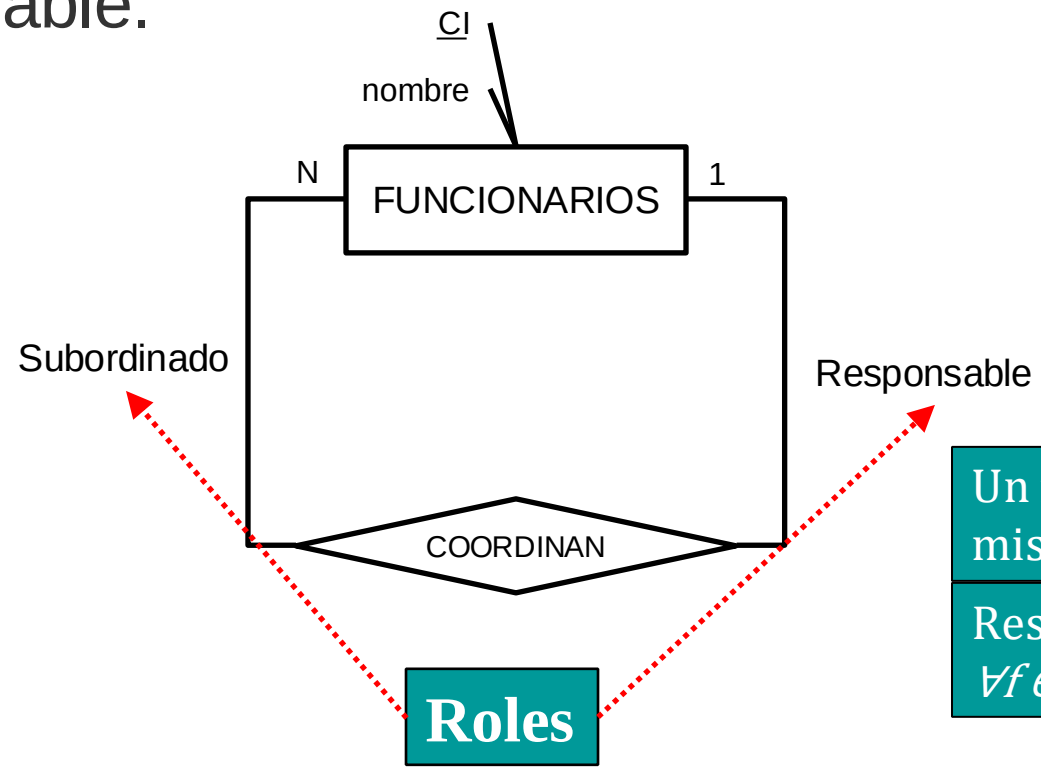
Combinación de Constructores de Atributos

- Hay un conjunto de cines de los que se conoce el nombre y la ciudad y se asume que la pareja nombre-ciudad identifica al cine.
- Existe también un conjunto de películas que se exhibe en los cines y de las que se conoce un código que la identifica y un nombre.
- Cada cine efectúa diferentes funciones de cada película, con una fecha y hora para cada una.



Autorelaciones

- En una empresa, existen **funcionarios** y se sabe que unos funcionarios son **responsables** de otros. Un responsable coordina varios funcionarios y un **subordinado** es coordinado por un único responsable.



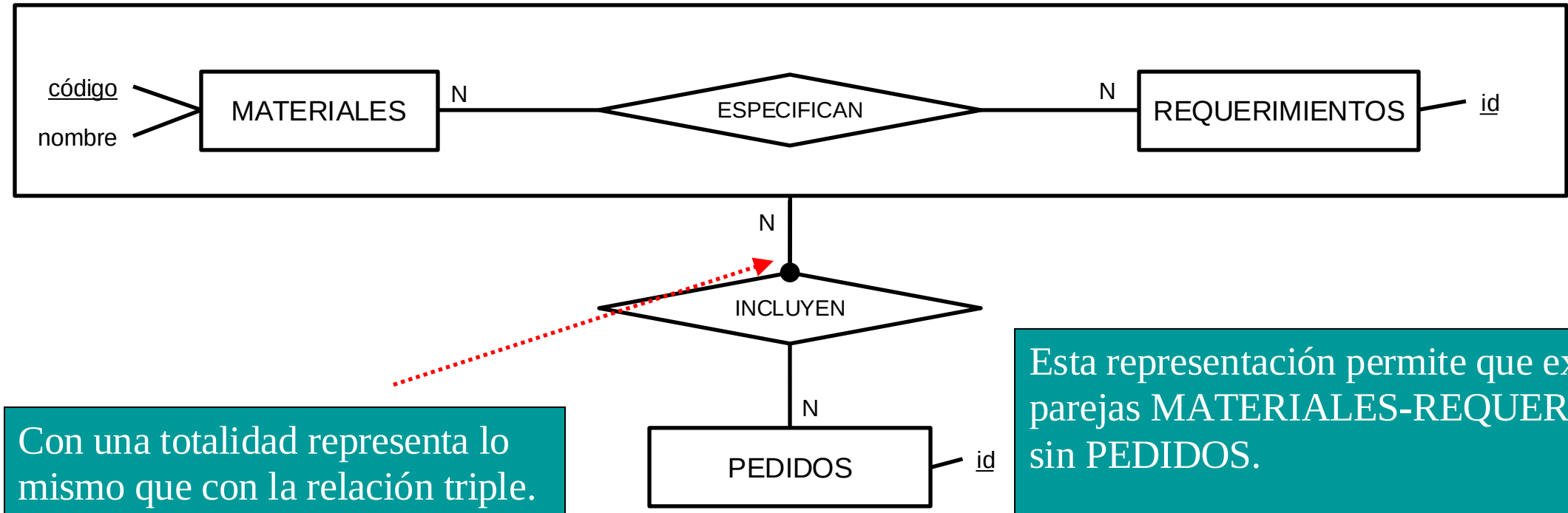
Un funcionario no puede coordinarse a si mismo

Restricción No Estructural:
$\forall f \in FUNCIONARIOS \ (<f,f> \notin COORDINAN)$

Agregaciones

- **Objetivo:**
  - Representar asociaciones entre elementos de Relaciones y de otros Conjuntos de Entidades.
  - Representar relaciones entre múltiples Conjuntos de Entidades pero manteniendo relaciones binarias.
- **Constructor:**
  - Se re-interpreta una Relación como si fuera un Conjunto de Entidades.
  - El nuevo Conjunto de Entidades se utiliza como cualquier otro.

Agregaciones



Con una totalidad representa lo mismo que con la relación triple.

Esta representación permite que existan parejas MATERIALES-REQUERIMIENTOS sin PEDIDOS.

Una relación triple no lo permite.

Especialización de Conjuntos de Entidades

- Una empresa tiene varios departamentos de los que se conoce un nombre que los identifica. Además, tiene un conjunto de funcionarios de los que se conoce su CI, su Nombre, su dirección y el departamento en que trabajan.

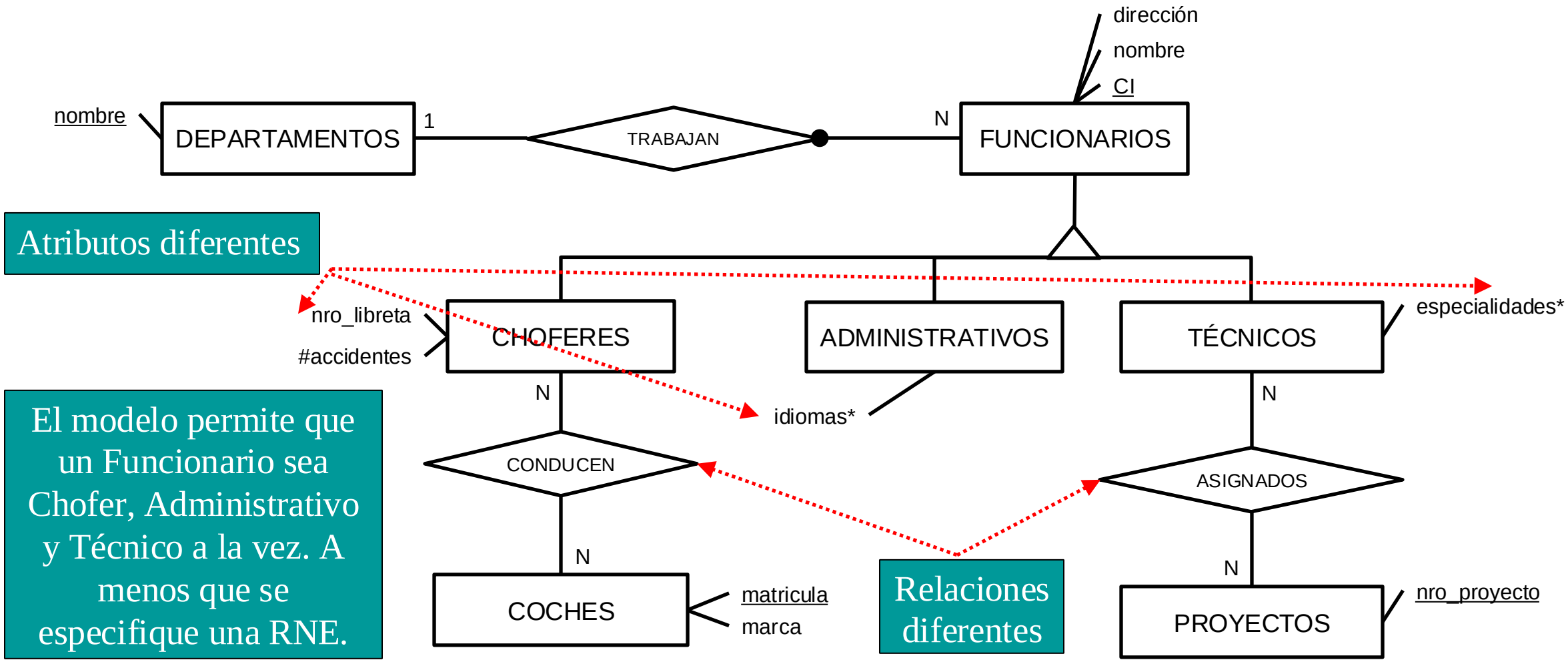
- Si el funcionario es un chofer, se conoce su nro de libreta de conducir y la cantidad de accidentes que tuvo. Si es administrativo, entonces se conocen los idiomas que habla. Si es técnico, se conoce las especialidades en que puede trabajar.

Especialización de Conjuntos de Entidades

- La empresa tiene a su vez un conjunto de coches de los que se conoce su matrícula y la marca. Cualquier coche puede ser conducido por cualquier chofer.

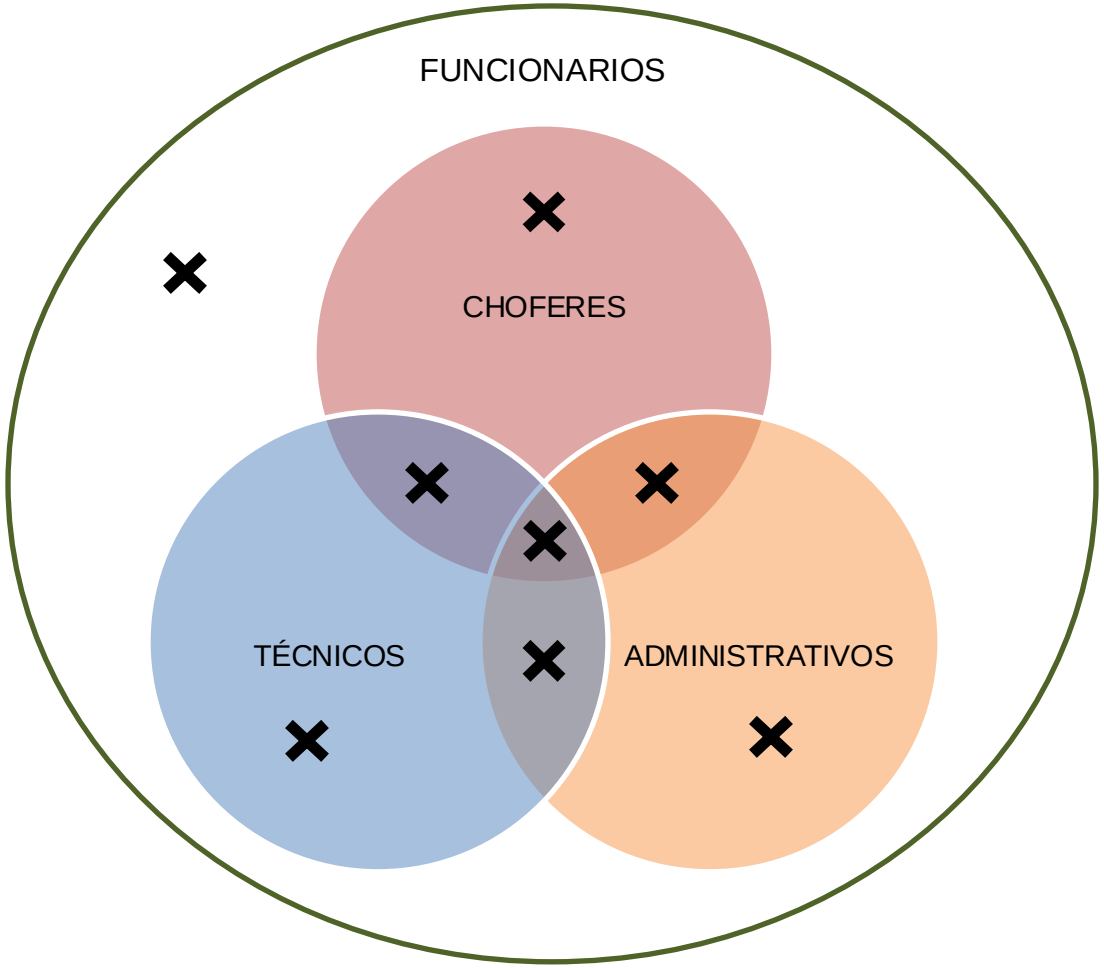
- La empresa lleva adelante un conjunto de proyectos. De cada proyecto se conoce un nro. de proyecto que lo identifica y qué técnicos trabajan en él. Un técnico puede estar asignado a varios proyectos simultáneamente y en cada proyecto pueden trabajar varios técnicos.

Especialización de Conjuntos de Entidades



Especialización de Conjuntos de Entidades

- Especialización vista como conjuntos:



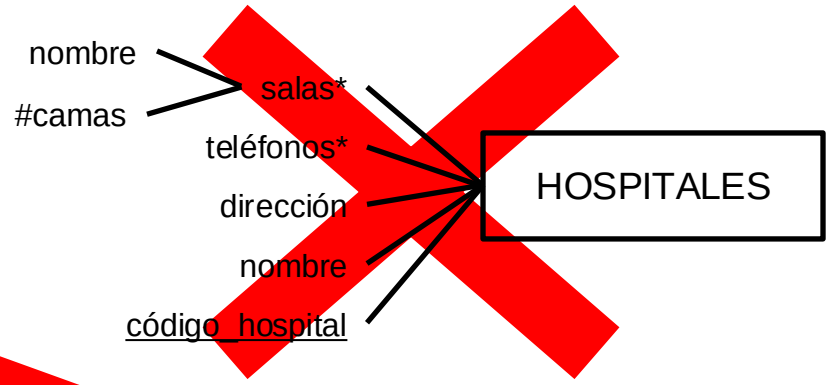
Entidad Débil

- Existe un conjunto de hospitales de los cuales se conoce el código del hospital, el nombre, la dirección y los teléfonos.

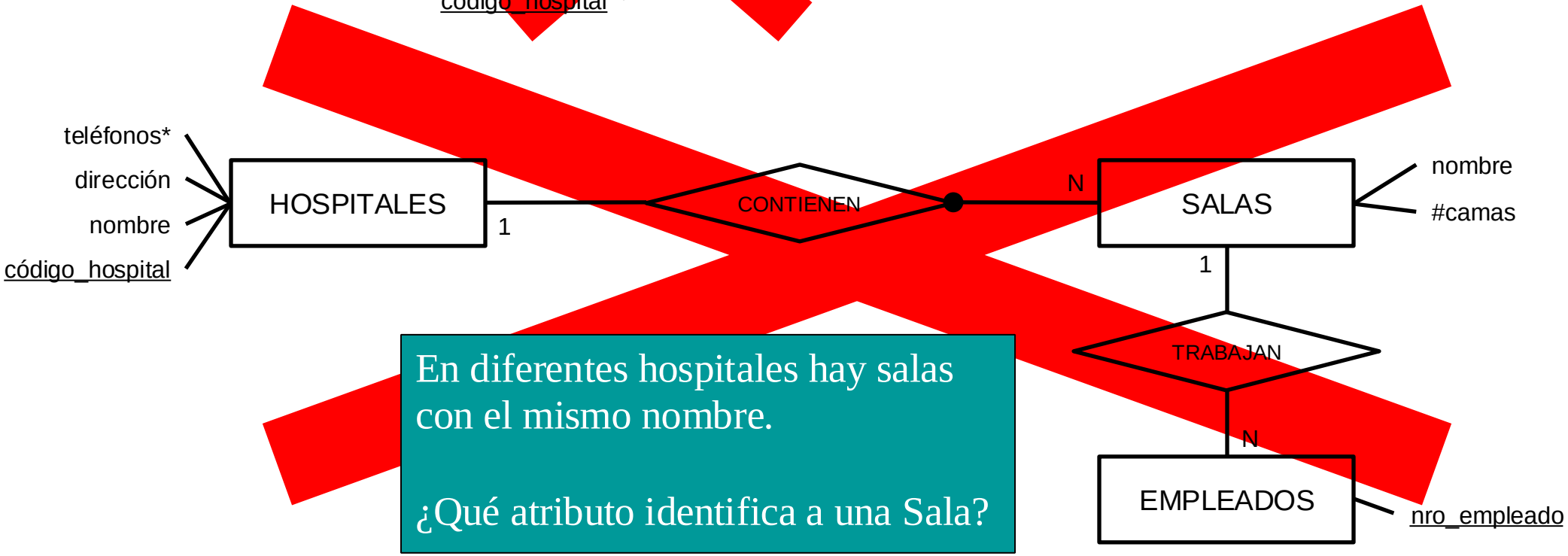
- Hay salas de las cuales se conoce el nombre de la sala y la cantidad de camas que tiene en un hospital dado. En diferentes hospitales hay salas con el mismo nombre (Ej. General, Operaciones 1) pero no dentro de un mismo hospital.

- Hay empleados que trabajan en las salas de los hospitales de los cuales se conoce el nro. de empleado entre otras informaciones. Los números de empleados no se repiten en los distintos hospitales.

Entidad Débil



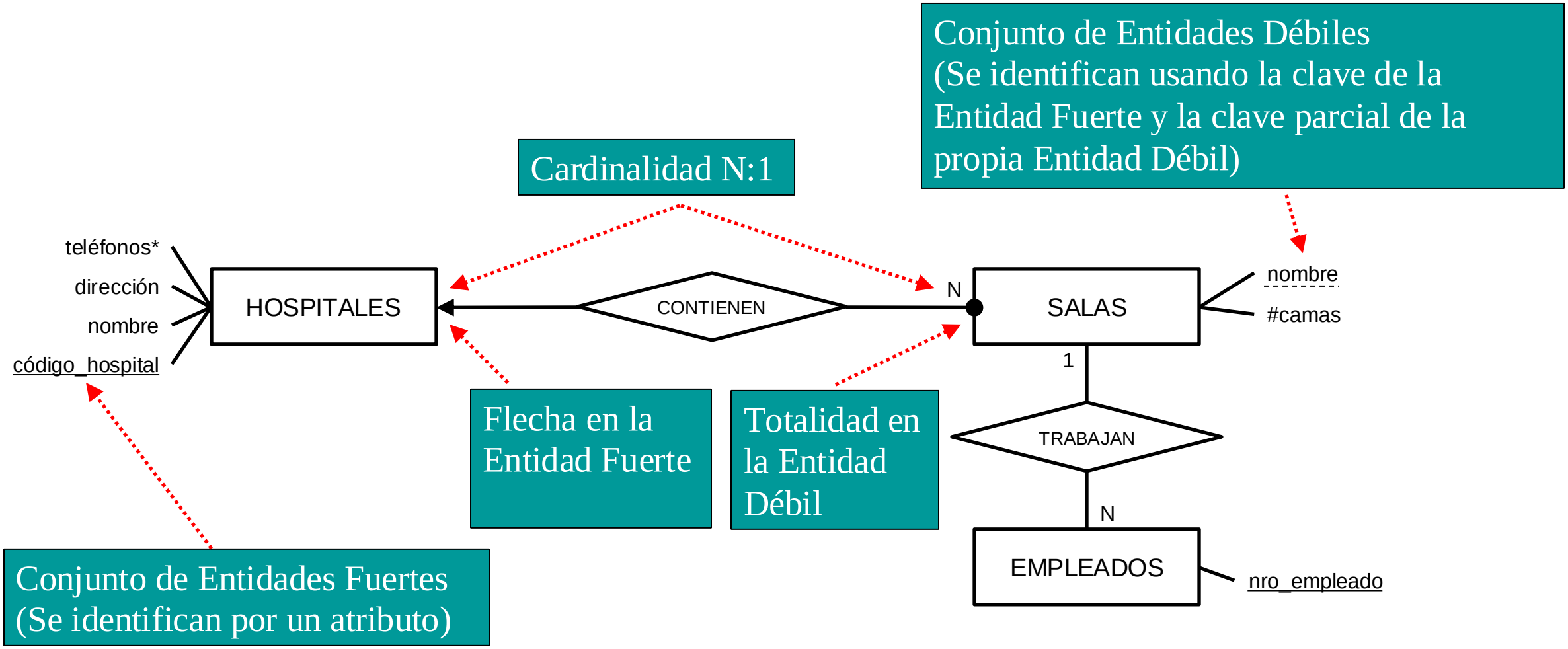
No sería posible relacionar a los empleados con las salas para saber en cuál trabajan



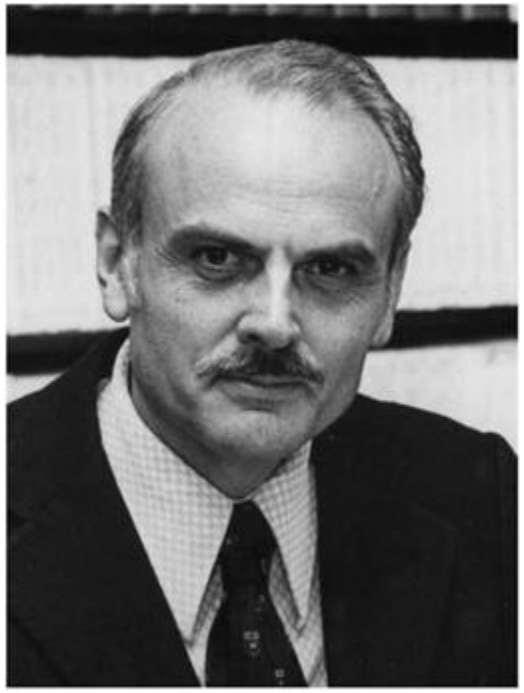
En diferentes hospitales hay salas con el mismo nombre.

¿Qué atributo identifica a una Sala?

Entidad Débil



MODELO RELACIONAL



Modelo Relacional

- **Referencia:**
  - Elmasri -Navathe. 6ª Edición. Capítulo 3

Conceptos Generales

- **Es un Modelo de Datos Lógico.**
  - Se usa como Modelo implementado por DBMS.
- **Creado por Codd en 1970.**
  - Se comenzó con una definición teórica.
  - Se proponía un modelo con fuertes elementos matemáticos para BDs.
- **Actualmente : modelo lógico dominante.**
  - Los DBMS Relacionales son la enorme mayoría.

Conceptos Generales

- **Visión Informal del Modelo.**
  - Las estructuras consisten en TABLAS,
    - Las columnas corresponden a ATRIBUTOS de tipo atómico.
    - Las filas corresponden a registros de datos.
  - Las operaciones están fundamentalmente orientadas a manejo de TABLAS, como conjuntos de registros.
  - Es un modelo de datos extremadamente simple y claro, que también ha resultado potente para la mayor parte de las aplicaciones de BDs.

Conceptos Generales

- **Dominio D.**
  - Es un conjunto de valores atómicos.

- **Esquema de relación $R(A_1,...,A_n)$.**
  - R es el nombre de relación.
  - $A_1,...,A_n$ son los atributos con dominios $D_1,...,D_n$.

- **Relación r(R).**
  - Es una instancia de un esquema de relación R.
  - Consiste en un conjunto de t-uplas (o tuplas)
    - $r = \{<a_1,...,a_n> , <b_1,...,b_n>, <c_1,...,c_n> ,.... \}$
  - También puede interpretarse a r como:
    - $r(R) \subseteq (D_1 \times ... \times D_n)$

Conceptos Generales

- **Tupla:**
  - La instancia de un esquema de relación es un **conjunto** de Tuplas:
    - Esquema: `ESTUDIANTES (CI, nombre, dirección)`
    - Instancia: `{<"1.876.543-2", "Juan", "Bvar Artigas 1232">, <…>}`
  - Una tupla es un elemento de un producto cartesiano de N dominios.
  - Puede verse como un "array":
    - `<"1.876.543-2", "Juan", "Bvar Artigas 1232">[1] = "1.876.543-2"`
  - Función del nombre de los atributos en el contenido:
    - `t: {CI, nombre, dirección} → Números U Strings`
    - `<"1.876.543-2", "Juan", "Bvar Artigas 1232">(CI) = "1.876.543-2"`

Conceptos Generales

- **Esquema de BD Relacional o Esquema Relacional:**
  - Conjunto de esquemas de relación de la BD.
  - Ejemplo:

```
EMPLEADOS (Nombre, Apellido, NSS, FechaNac, Dir, Sexo, Salario,
NSSSuper, ND)

DEPARTAMENTOS (Nombre, NumeroD, NSSGte, FechaInicGte )

LUGARES_DEPARTAMENTOS (NumeroD, LugarD)

PROYECTOS (Nombre, NumeroP, LugarP, NumeroD)

TRABAJA_EN (NSSE, NumP, Horas)
```

Ejemplo de Instancia de BD Relacional

- Fabricantes que Venden Productos:

	FABS		
	#f	Nombre	Direcc
	1	Juan	d1
	2	Pedro	d2.
	4	Maria	d3
	5	Ana	d2
	6	Pedro	d4.
	9	Pepe	d5
	10	Laura	d4
	13	Maria	d3.
	15	Pedro	d1
	16	Oscar	d3
	19	Juan	d4

PRODS	
#p	desc
1	t1
2	t2
3	t3
5	t2
6	t3
7	t4
9	t2
10	t1
11	t3
12	t2
15	t3

VENTAS		
#f	#p	precio
1	1	100
1	2	200
1	3	300
1	10	1000
1	11	1100
2	3	350
2	6	600
2	7	700
5	3	350
5	5	200
9	7	100
9	3	300
10	3	400

Características de las Relaciones

- **Es un conjunto de tuplas:**
  - No está ordenado.
  - No hay repetidos.
- **Valores de Atributos en tuplas:**
  - Son valores atómicos (indivisibles).
    - Propiedad: primera forma normal (1NF).
- **Atributos ordenados o no ?**
  - Visión “producto cartesiano” o “array” : SI .
  - Visión “tuplas como funciones”: NO .
    - t: $R \rightarrow D_1 \cup ... \cup D_n$

RI en el Modelo Relacional

- Dado $R(A_1, \ldots, A_n)$, se dice que $X \subseteq \{A_1, \ldots, A_n\}$ es **superclave** en un esquema R, si no puede existir ninguna r(R) tal que tenga dos tuplas con valores iguales de X ($t[X] = t'[X]$).

RI en el Modelo Relacional

- **Restricciones de Dominios.**
  - Restricciones de tipo en los D_i:
    - Indica a qué tipo pertenecen los valores.
    - Pueden incluir sub-rangos o enumerados.

- **Ejemplo:**
  - `FUNCIONARIO(CI, Nombre, Dir, Edad)`
    - `CI: number(9);`
    - `Nombre, Dir: String;`
    - `Edad: number(2); Edad > 18;`

RI en el Modelo Relacional

- **Clave**
  - Una ***clave*** es una *superclave* que no contiene propiamente una *superclave* (o sea, minimal).

- **Ejemplos:**
  - FABRICANTES (#código_fabricante, nombre, dirección)
  - PRODUCTOS (#código_producto, descripción)
  - VENTAS (#código_fabricante, #código_producto, precio)

RI en el Modelo Relacional

- **Claves Foráneas (Foreign Keys)**
  - Dado R, un conjunto de atributos X es una FK de R sobre S si se cumplen simultáneamente las siguientes condiciones:
    - Los atributos de X coinciden en dominio con los de una clave Y de S.
    - Los valores de X en tuplas de r(R) (para toda r) corresponden a valores de Y en la relación s(S).

RI en el Modelo Relacional

- **Integridad Referencial**
  - Se dice que existe una RI Referencial entre R y S, donde R referencia a S.
  - Es otra forma de decir que en R hay una Foreign Key sobre S.

RI en el Modelo Relacional

- **Ejemplo de RI Referenciales:**
  - Departamentos.NSSGte FK Empleados.NSS
  - Empleados.NSSSuper FK Empleados.NSS
  - Proyectos.NumeroD FK Departamentos.NumeroD

EMPLEADOS (Nombre, Apellido, NSS, FechaN, Dir, Sexo, Salario, NSSSuper, ND)

DEPARTAMENTOS (Nombre, NumeroD, NSSGte, FechaInicGte)

LUGARES_DEPARTAMENTOS (NumeroD, LugarD)

PROYECTOS (Nombre, NumeroP, LugarP, NumeroD)

TRABAJA_EN (NSSE, NumP, Horas)

RI en el Modelo Relacional

- **Una BD se considera válida si:**
  - Todas las relaciones r satisfacen las RIs.
    - Todas las instancias actuales de todas las relaciones declaradas en el esquema relacional satisfacen todas las RIs.

RI en el Modelo Relacional

- **Propiedades importantes:**
  - Las RI surgen de:
    - La observación de la realidad.
    - NO de la observación de relaciones.
  - Las RI se definen a nivel de:
    - ESQUEMA RELACIÓN
    - NO a nivel de instancia.
  - Las RI son verificadas o violadas por:
    - relaciones (instancias).
    - NO por esquemas de relación.

Operaciones de Modificación

- **INSERT**

  Sea R(A,B,C) y r(R),

  INSERT <a,b,c> **INTO** R

  incluye la tupla <a,b,c> en la relación r.

  - Las tuplas insertadas deben cumplir las RI.

Operaciones de Modificación

- **DELETE**

Sea R(A,B,C) y r(R),

DELETE FROM R **WHERE** <cond>

borra de las tuplas de r las que cumplen la condición <cond>

- Borrar tuplas puede generar violaciones a RI,
  - ¿En qué casos ?

Operaciones de Modificación

- **UPDATE**
  - Sea R(A,B,C) y r(R),

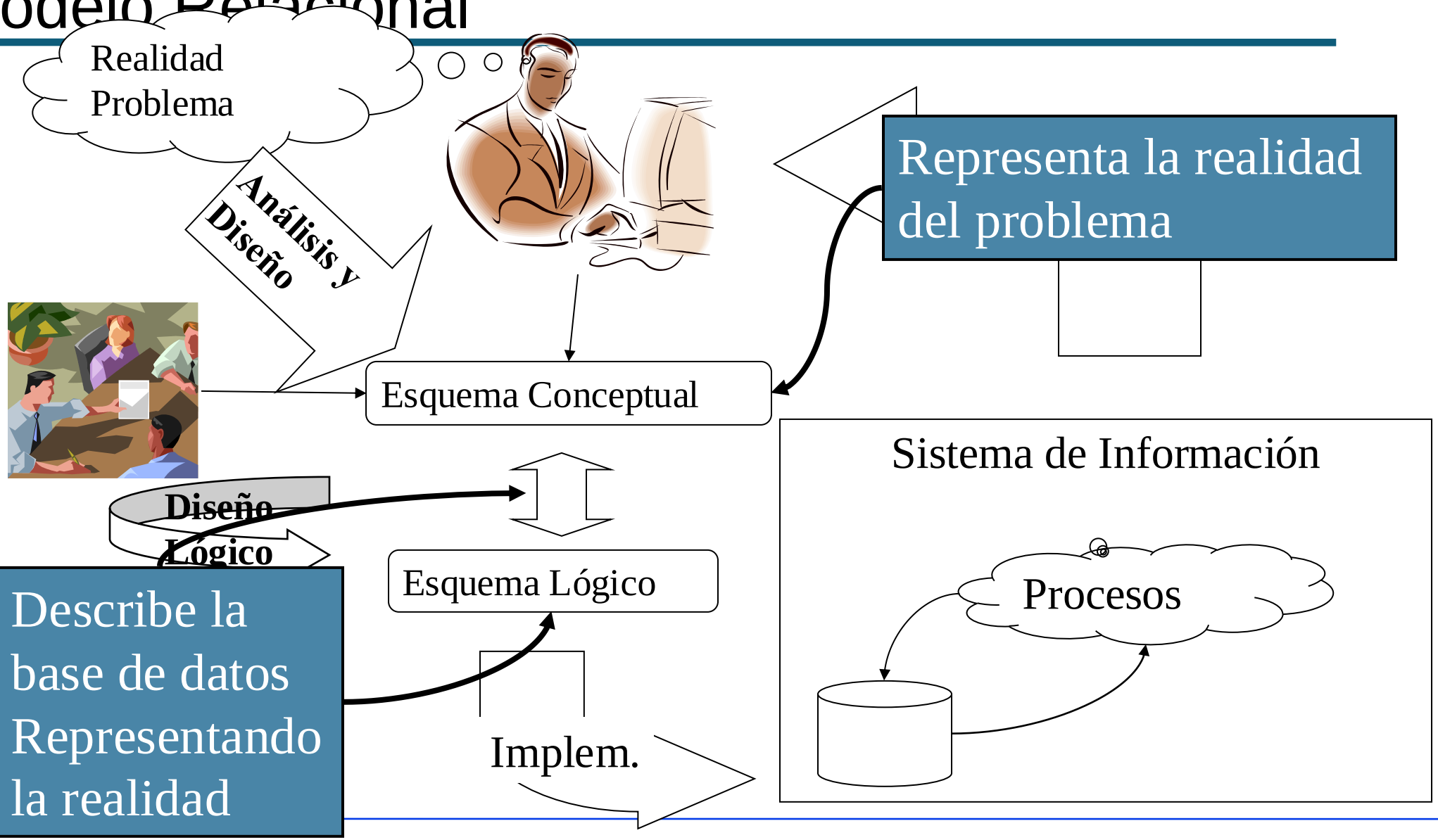
    UPDATE R **SET** <atributo> =<valor>,… **WHERE** <cond>

    modifica las tuplas de r que cumplen la condición <cond>.

  - Actualizar tuplas puede generar violaciones a RI,
    - ¿En qué casos ?

PASAJE MER A RELACIONAL

Construcción de un Sistema de Información y Modelo Relacional



INTRODUCCIÓN

Hay reglas para cada estructura del MER:

- Entidades Fuertes y Atributos
- Entidades Débiles
- Relaciones
- Agregaciones
- Categorizaciones

Trabajaremos sobre el ejemplo de los hospitales visto en el teórico de Modelo Entidad Relación.

MODELO RELACIONAL y DEPENDENCIAS DE INCLUSION

Es otra restricción sobre el Modelo Relacional.

Expresa que una proyección de ciertos atributos de una tabla debe estar incluída en la proyección de otros atributos de otra (o la misma) tabla.

Notación:

$$\Pi_{a1,a2\ldots an}(A) \subseteq \Pi_{b1,b2\ldots bn}(B)$$

Observar que las claves foráneas son un caso particular de dependencia de inclusión (pero no al revés).

ENTIDADES

Por cada entidad se crea una tabla.

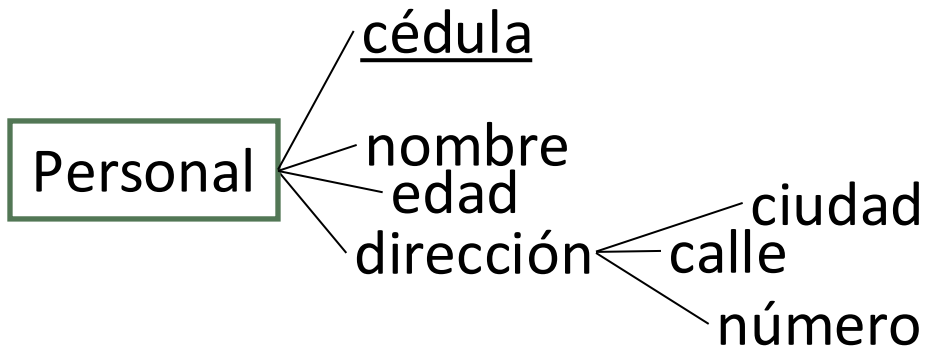
- Por cada atributo simple se crea un atributo en la tabla
- Para cada atributo estructurado se crean tantos atributos como “hojas” tenga la estructura.
- Si tiene atributos multivaluados los procesamos más adelante.

¿Cual es la clave primaria?

- Se selecciona uno de los atributos determinantes de la entidad como clave primaria de la tabla. Los restantes atributos determinantes (si los hubiere) deben ser marcados como claves alternativas.

ENTIDADES - EJEMPLO

MER



RELACIONAL

PERSONAL(cedula, nombre, edad, ciudad, calle, numero)

ATRIBUTOS MULTIVALUADOS

Por cada atributo multivaluado (ya sea de entidad o de relación) se crea una tabla.

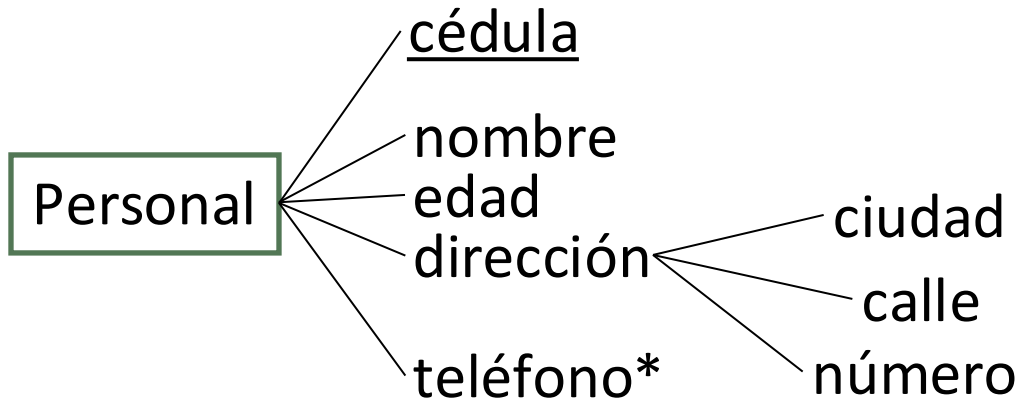
- Se crea un atributo para el multivaluado.
- Se agregan atributos que representan la clave primaria de la tabla que modela la entidad o relación a la cual pertenece el multivaluado

¿Cómo se determina la clave primaria?

- La clave está formada por todos sus atributos.

ATRIBUTOS MULTIVALUADOS - EJEMPLO

MER



RELACIONAL

PERSONAL(cedula, nombre, edad, ciudad, calle, numero)
TELEFONOS(cedula, telefono)

DEPENDENCIAS DE INCLUSION en ATRIBUTOS MULTIVALORADOS

Dado que se usan al menos dos tablas para la representación relacional, se deben restringir las relaciones entre esas tablas.

Se agregan las claves foráneas que correspondan:

$$\Pi_{\text{cedula}}(\text{TELEFONOS}) \subseteq \Pi_{\text{cedula}}(\text{PERSONAL}).$$

RELACIONES BINARIAS N:N

Para cada relación binaria con cardinalidad N:N se crea una tabla donde:

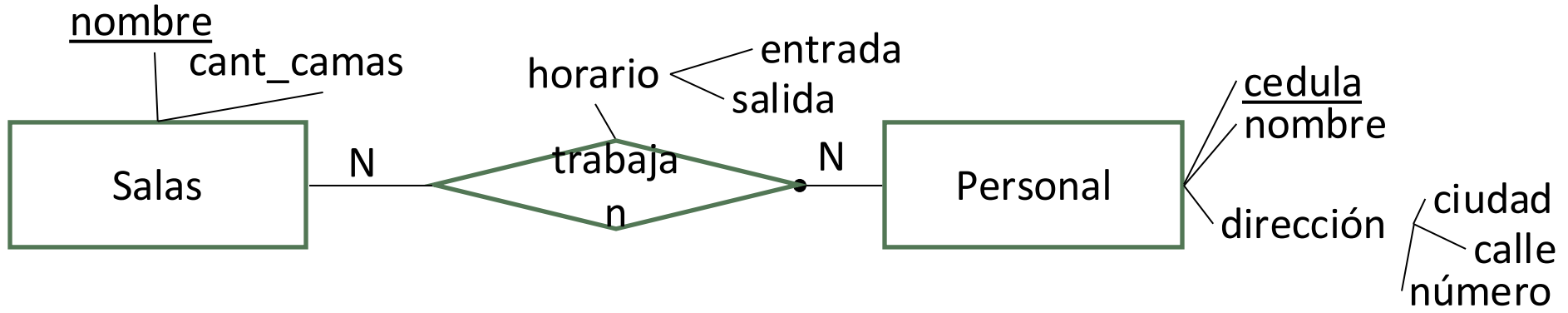
- Se colocan las claves primarias de las tablas que representan a cada una de las entidades participantes.
- Si existen atributos en la relación se tratan como si fueran los de una entidad.

¿Cómo se determina la clave primaria?

- Está formada por los atributos correspondientes a las claves primarias de las tablas que representan a las entidades participantes

RELACIONES BINARIAS N:N - EJEMPLO

MER



RELACIONAL

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

DEPENDENCIAS DE INCLUSION en RELACIONES

Por cada entidad participante en una relación se agrega una dependencia de inclusión.
Sea R la tabla de la relación y Q la tabla de un participante:

$\Pi_{q\_pk}(R) \subseteq \Pi_{q\_pk}(Q)$, donde q_pk es la clave primaria de Q en R y en Q.

Si la relación R es total sobre Q, entonces se agrega también la inclusión al revés:

$\Pi_{q\_pk}(Q) \subseteq \Pi_{q\_pk}(R)$.

EN EL EJEMPLO

$$\Pi_{nombreSala}(TRABAJAN) \subseteq \Pi_{nombre}(SALAS)$$
$$\Pi_{cedula}(TRABAJAN) \subseteq \Pi_{cedula}(PERSONAL)$$
$$\Pi_{cedula}(PERSONAL) \subseteq \Pi_{cedula}(TRABAJAN)$$

RELACIONES BINARIAS 1:N

Hay que dividir dos casos que se tratan en forma diferente:

- 1:N Sin totalidad del lado N.
- 1:N Con totalidad del lado N.

Si es sin totalidad del lado N

- Se trata como una N:N excepto por la clave primaria de la tabla de la relación, que es la clave del lado N.

Si es con totalidad del lado N

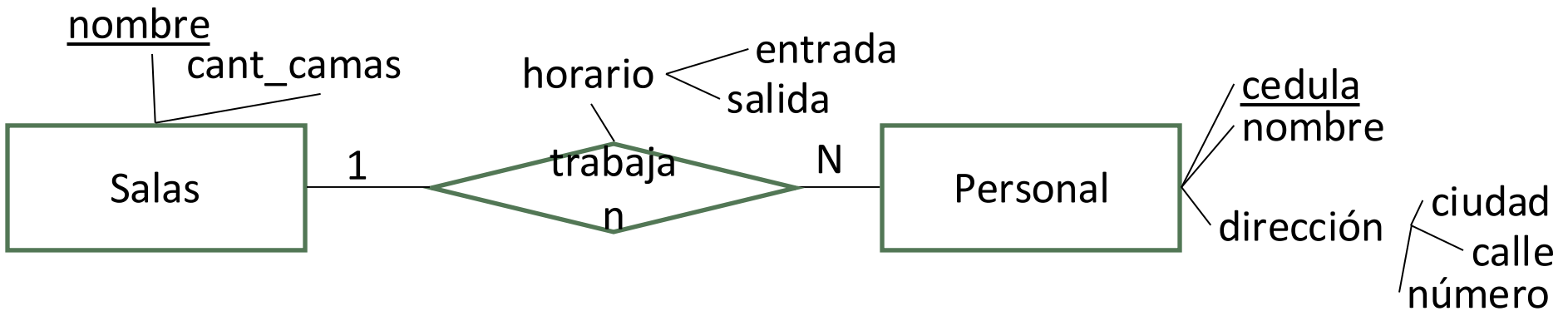
- No se crea tabla para la relación y se agrega la clave de la tabla de la entidad del lado 1 en la tabla de la entidad del lado N.
- No hay cambios en la clave primaria

RELACIONES BINARIAS 1:N (sin totalidad del lado N)

REAL.

Los funcionarios pueden trabajar o no en salas. Si trabajan en salas, lo hacen sólo en una.

MER



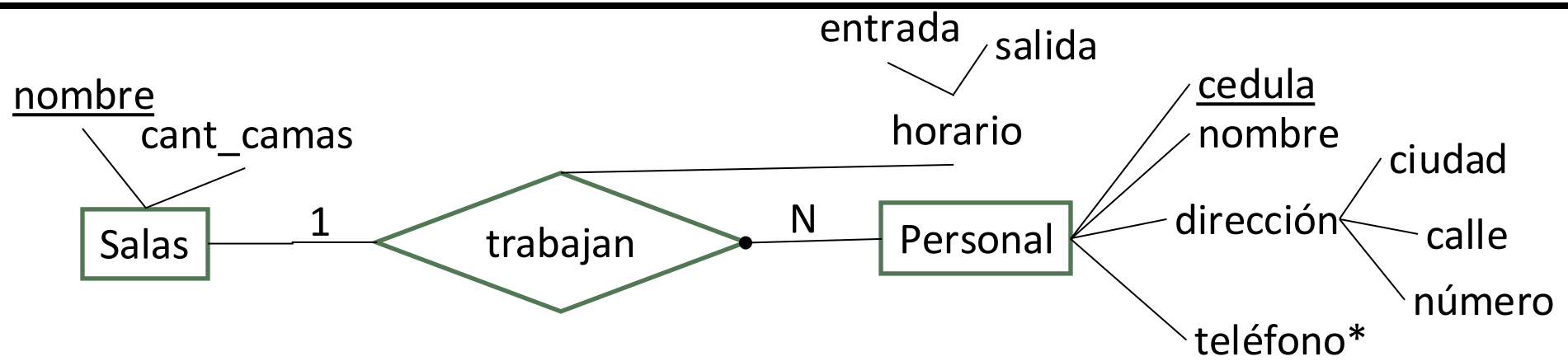
RELACIONAL

- SALAS (nombre, cant_camas)
- PERSONAL (cedula, nombre, ciudad, calle, numero)
- **TRABAJAN** (nombreSala, **cedula**, entrada, salida)
- $\Pi_{nombreSala}(TRABAJAN) \subseteq \Pi_{nombre}(SALAS)$
- $\Pi_{cedula}(TRABAJAN) \subseteq \Pi_{cedula}(PERSONAL)$

REAL.

- **Todos los funcionarios trabajan en salas. Cada funcionario trabaja en una sola sala.**

MER



RELACIONAL

SALAS (nombre, cant_camas)

PERSONAL (cedula, nombre, ciudad, calle, numero, nombreSala, hEntrada, hSalida)

Hay que agregar las dependencias de inclusión.

$\Pi_{nombreSala}(PERSONAL) \subseteq \Pi_{nombre}(SALAS)$

ENTIDADES DÉBILES

Por cada entidad débil se crea una tabla.

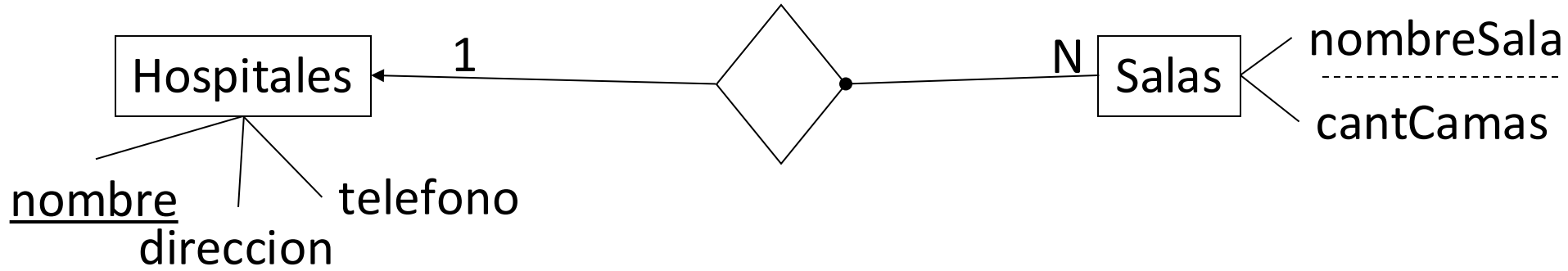
- Se procede con las relaciones 1:N y totales del lado N, no creando la tabla de la relación y agregando la clave primaria de la tabla de la entidad fuerte en la tabla de la entidad débil.

¿Cuál es la clave primaria?

- Clave primaria de la tabla que representa a la entidad fuerte + atributo/s que representa al identificador parcial

ENTIDADES DÉBILES - EJEMPLO

MER

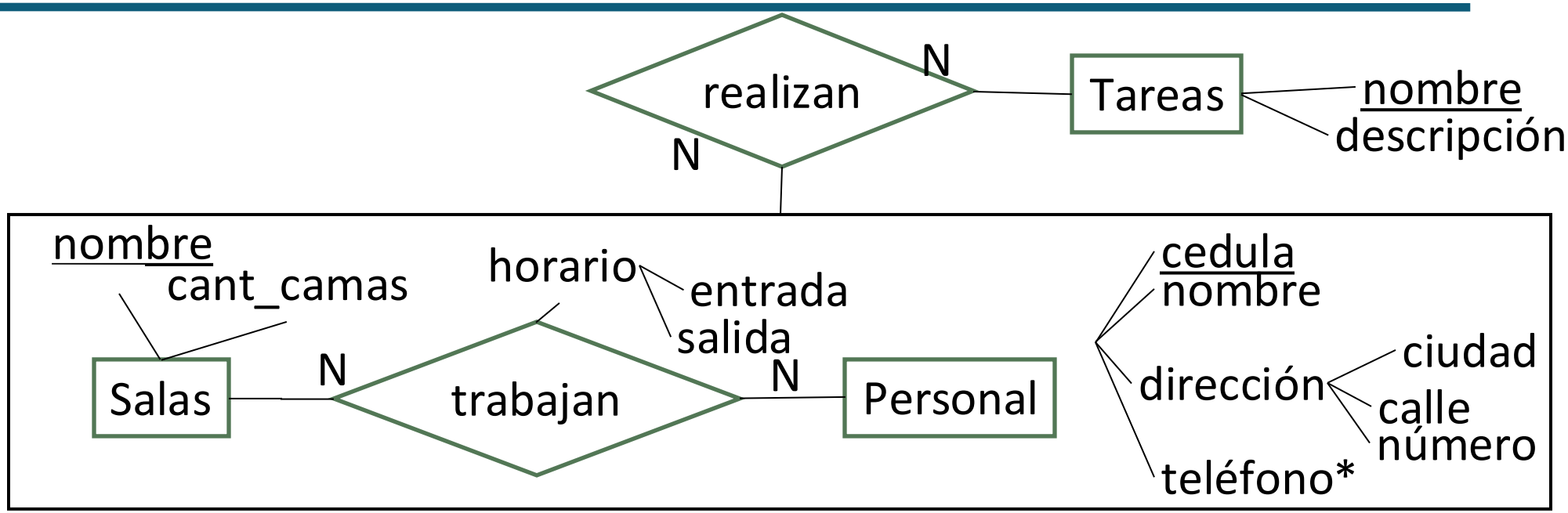


RELACIONAL

HOSPITALES(<u>nombre,</u> direccion, telefono)

SALAS(<u>nombreHospital, nombreSala</u>, cantCamas)

$\Pi_{nombreHospital}(SALAS) \subseteq \Pi_{nombre}(HOSPITALES)$

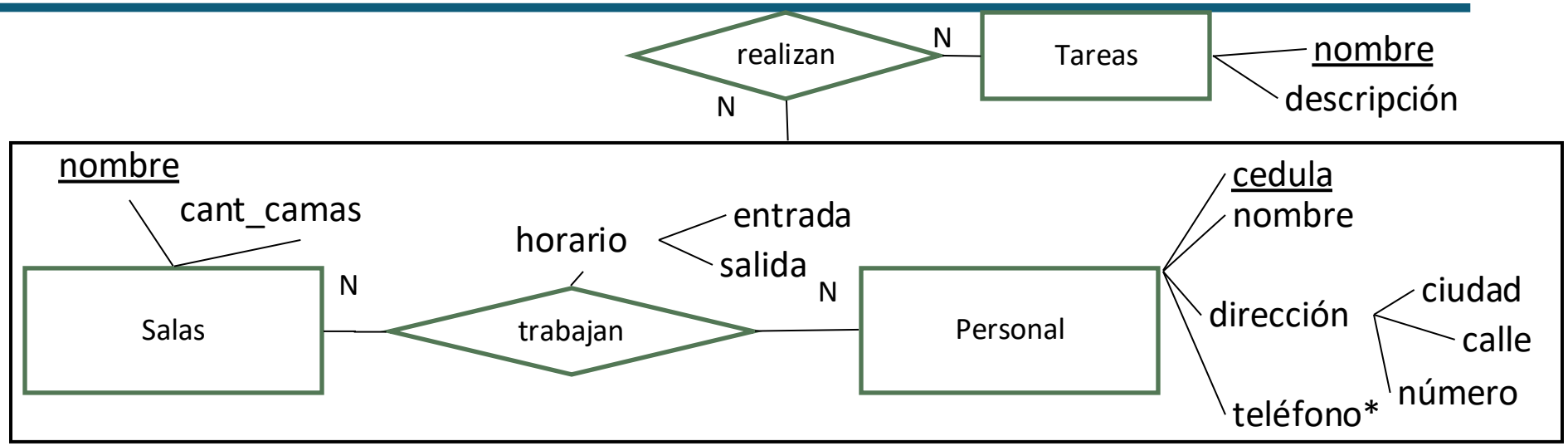


Recordemos que en MER el operador de agregación transforma a las relaciones en entidades.

En este caso las parejas de la relación TRABAJAN se relacionan con TAREAS. ¿Cómo se identifican las parejas de TRABAJAN?

AGREGACIONES (2)

MER



RELACIONAL

- TAREAS(<u>nombreTarea</u>, descripcion)
- TRABAJAN(<u>nombreSala, cedula</u>, hEntrada, hSalida)
- **REALIZAN(<u>nombreSala, cedula, nombreTarea</u>)**
  - $\Pi_{nombreTarea}(REALIZAN) \subseteq \Pi_{nombre}(TAREAS)$
  - $\Pi_{nombreSala,cedula}(REALIZAN) \subseteq \Pi_{nombreSala,cedula}(TRABAJAN)$

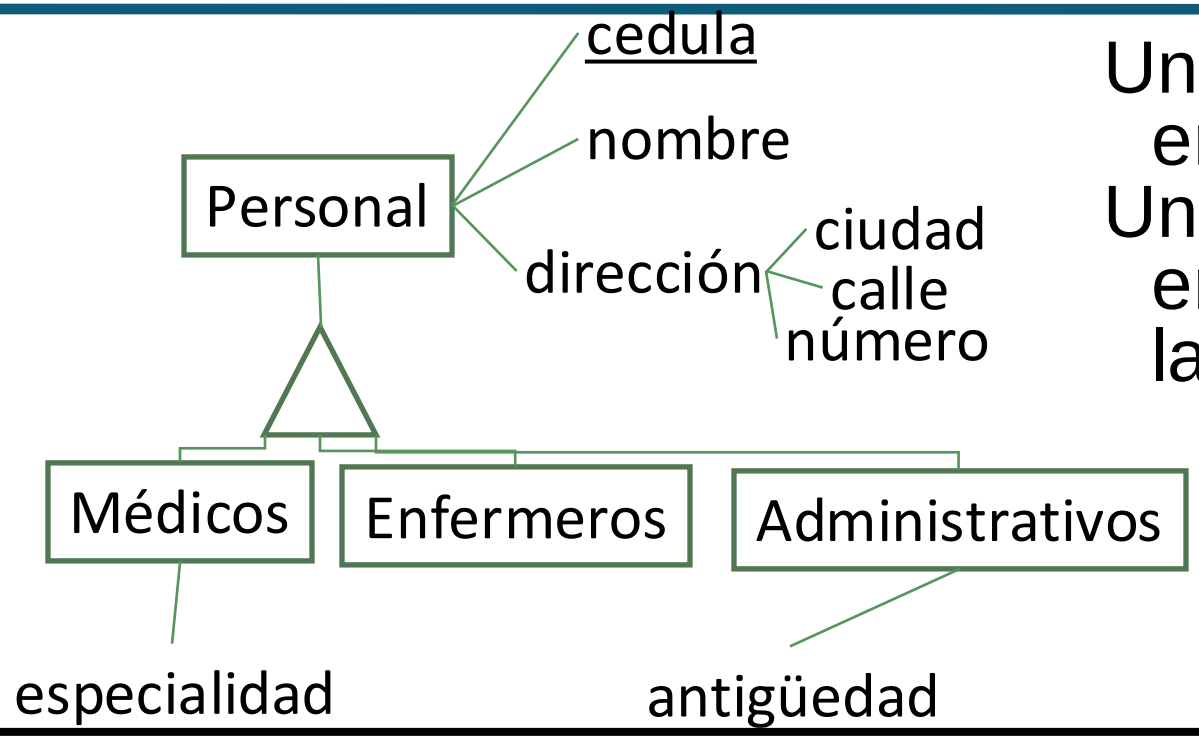
CATEGORIZACIONES

Hay que revisar diferentes opciones de implementación:

1. Por Joins: se aplica en cualquier caso, pero puede tener menos performance que el caso 4.
2. Por vistas: si es total.
3. Con atributo de tipo: si es disjunta.
4. Con atributo booleanos: también se puede aplicar en cualquier caso, pero típicamente gasta más memoria que el caso 1.

CATEGORIZACION (1)

MER



Una tabla para la super-entidad

Una tabla por cada sub-entidad con referencia a la super-entidad.

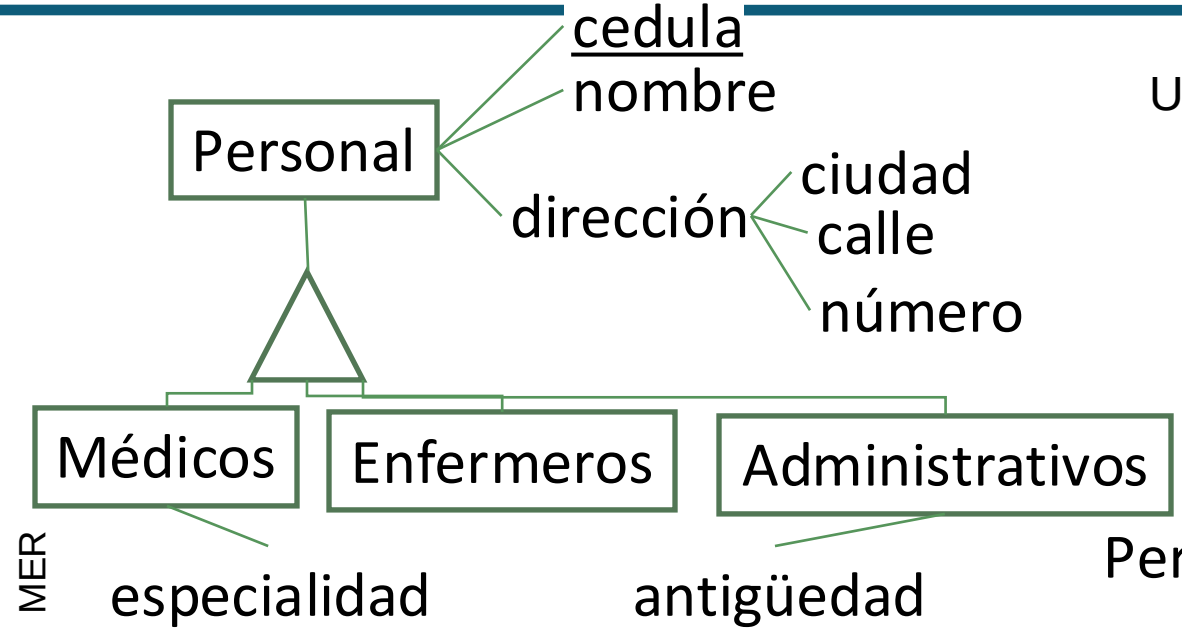
Funciona siempre !

RELACIONAL

PERSONAL(cedula, nombre, ciudad, calle, numero)
MEDICOS(cedulaPersonal, especialidad)
ENFERMEROS(cedulaPersonal)
ADMINISTRATIVOS(cedulaPersonal, antiguedad)

Faltan las deps. de inclusión

CATEGORIZACIONES (2)



MER

Una tabla por cada sub-entidad + vista

Sólo si la categorización es total

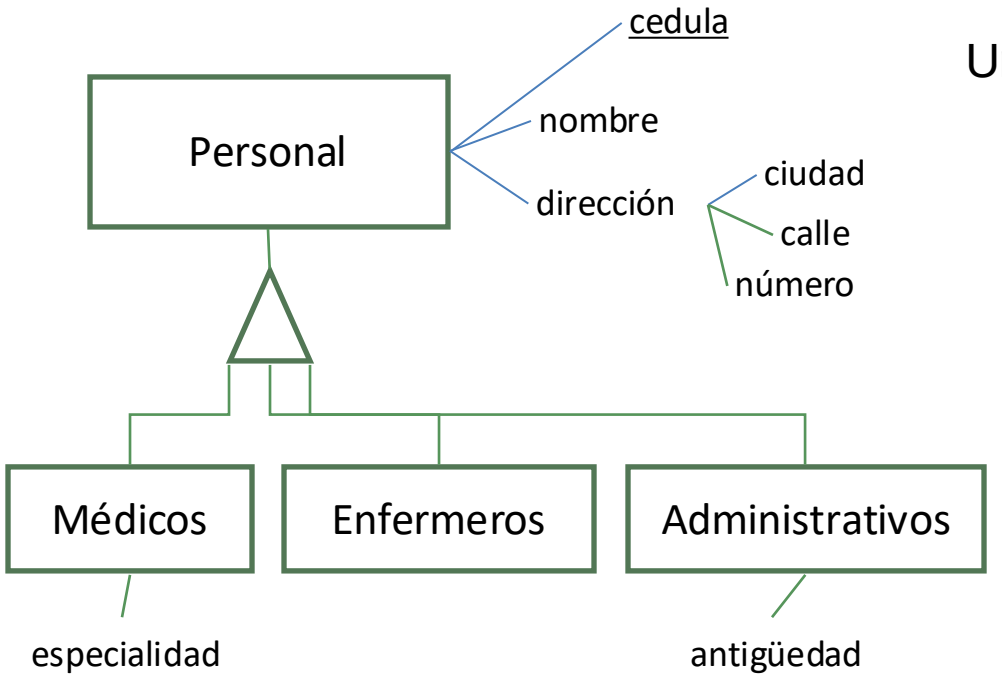
Personal = Médicos ∪ Enfermeros ∪ Administrativos

RELACIONAL

- MEDICOS(<u>cedula</u>, nombre, ciudad, calle, numero, especialidad)
- ENFERMEROS(<u>cedula</u>, nombre, ciudad, calle, numero)
- ADMINISTRATIVOS(<u>cedula,</u> nombre, ciudad, calle, numero antiguedad)
- PERSONAL ≡ {<t.cedula,t.nombre,t.ciudad,t.calle,t.numero> / MEDICOS(t) ∨ ENFERMEROS(t) ∨ ADMINISTRATIVOS(T) }

CATEGORIZACIONES (3)

MER



Una tabla con todos los atributos y un atributo de tipo.

Sólo si la categorización es disjunta

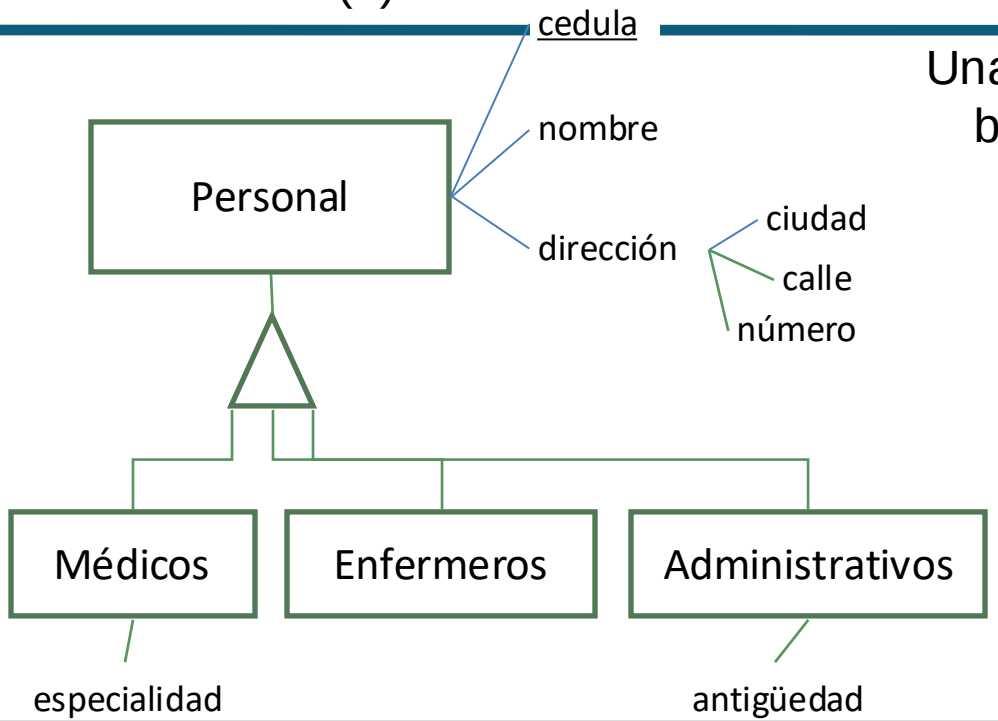
Medicos∩Enfermeros = ∅
Medicos∩ Administrativos = ∅
Enfermeros∩ Administrativos = ∅

RELACIONAL

PERSONAL(<u>cedula</u>, nombre, ciudad, calle, numero, especialidad, antigüedad, tipo)

CATEGORIZACIONES (4)

MER



Una tabla con todos los atributos y un atributo booleano por cada sub-entidad.

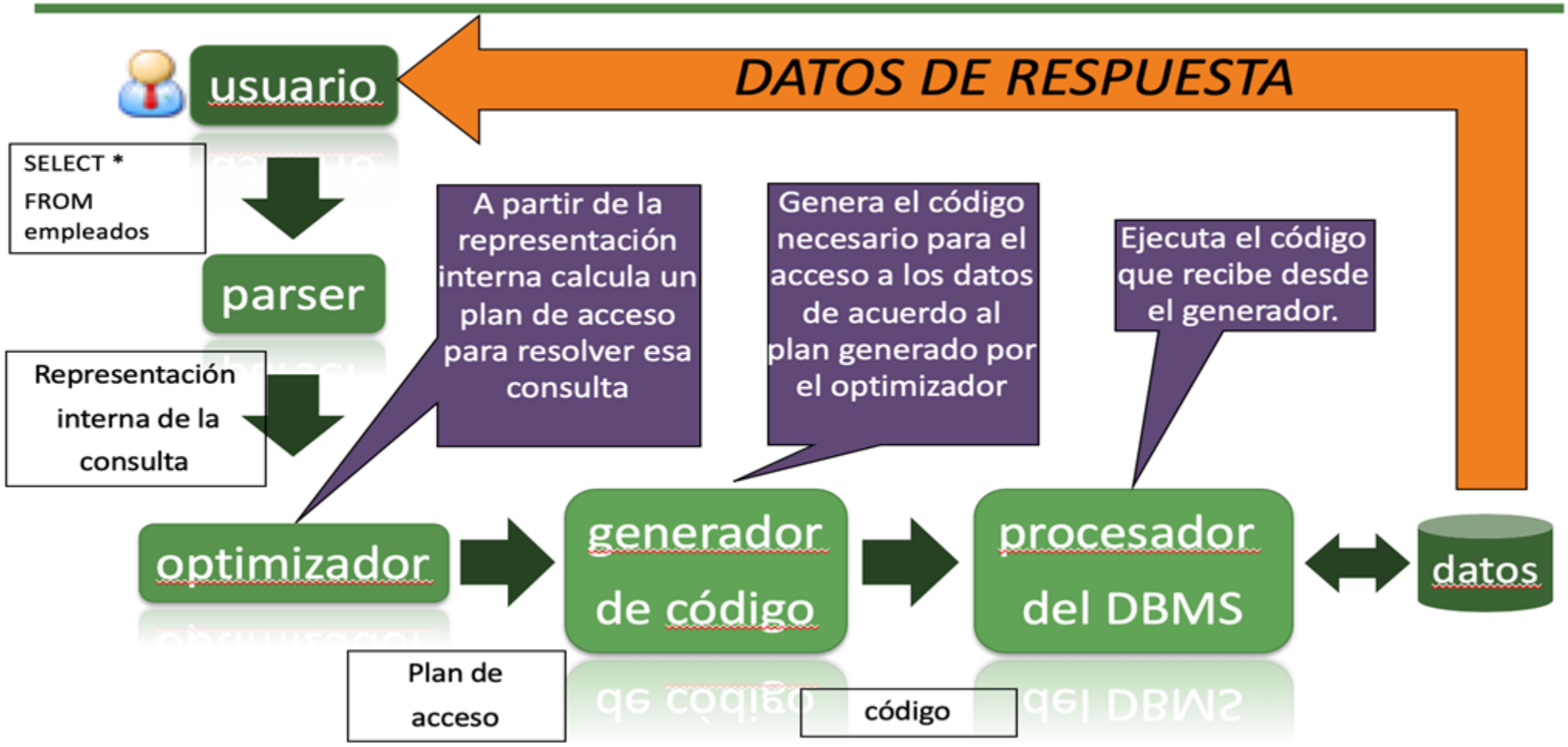
Sólo si la categorización no es disjunta

RELACIONAL

PERSONAL(<u>cedula</u>, nombre, ciudad, calle, numero, especialidad, antigüedad, esMedico, esEnfermero, esAdministrativo)

OPTIMIZACIÓN DE CONSULTAS

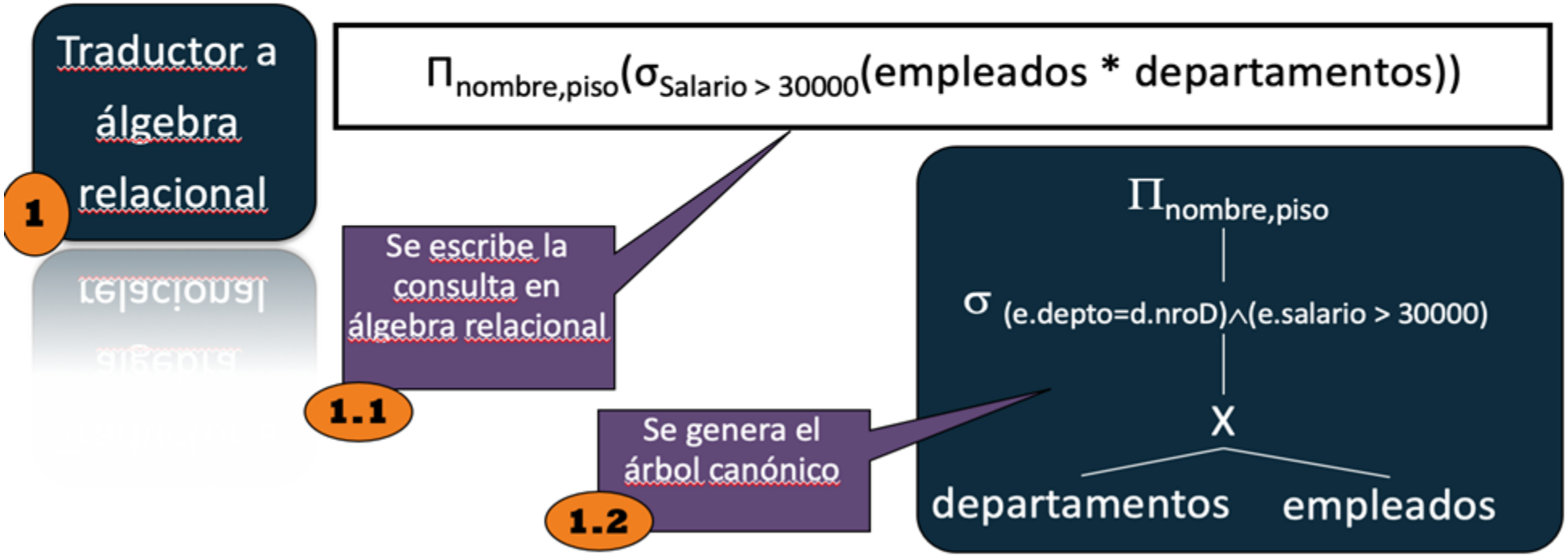
¿Cómo se resuelven las consultas?



Ejemplo de optimización (I)

- Resolvamos esta consulta sobre las tablas:
  - empleados(nombre,edad,salario,depto)
  - departamentos(nroD,nombreD,piso,gerente)

```
select  e.nombre, d.piso
from    departamentos d,
        empleados e
where   e.depto = d.nroD
        and e.salario > 30000
```

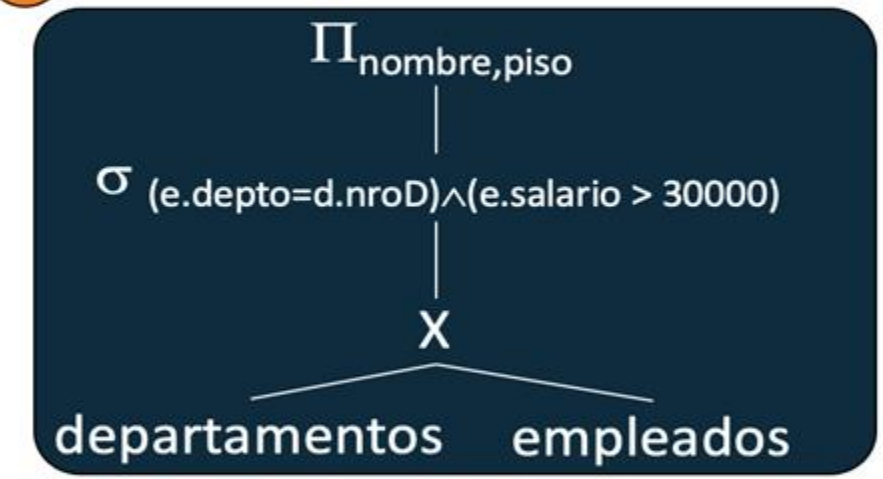


Ejemplo de optimización (II)

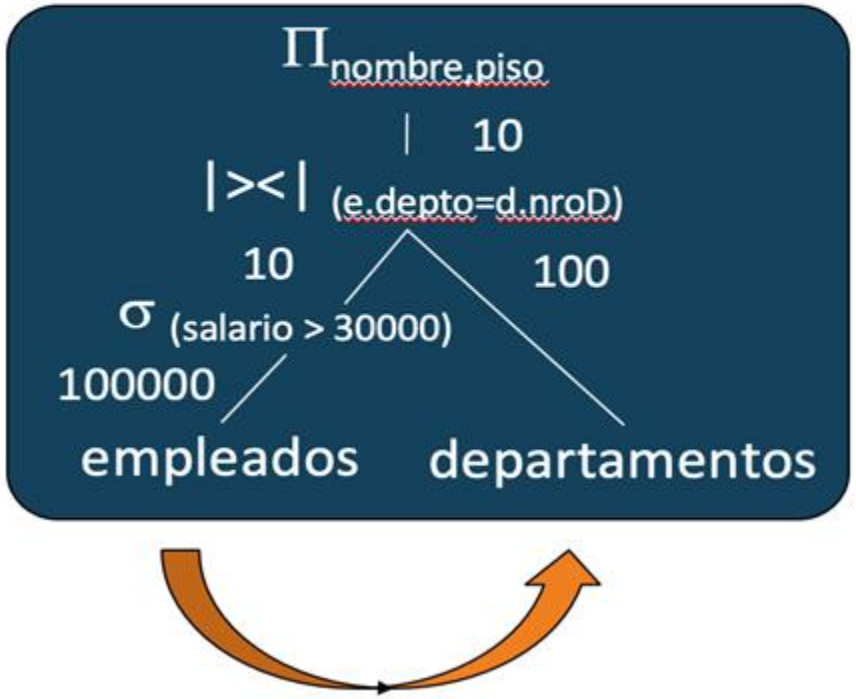
Generador de planes lógicos

2

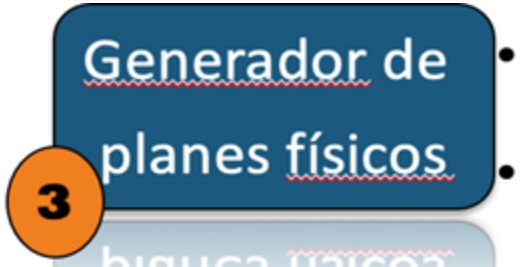
- A partir del árbol canónico se generan planes lógicos.
- Se usan heurísticas y se agregan datos de tamaño.



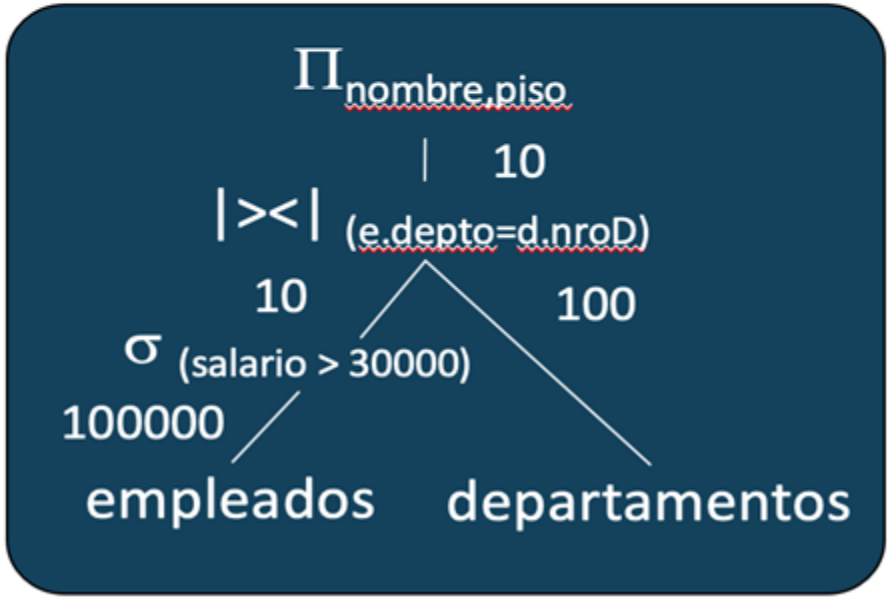
Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS (tuplas)	100
Selectividad de $\sigma_{(salario>3000)}$	1/10.000



Ejemplo de optimización (III)



- Para cada plan lógico
- Se consideran diferentes implementaciones



Parámetro	Valor
Tamaño de EMPLEADOS (tuplas)	100.000
Tamaño de DEPARTAMENTOS (tuplas)	100
Selectividad de $\sigma_{(salario>3000)}$	1/10.000
Cantidad de bloques para EMPLEADOS	2000
Cantidad de bloques para DEPTOS.	10
Índices sobre EMPLEADOS	B+ en salario
Índices sobre DEPARTAMENTOS	Hash en nroD

Operación	implementaciones		
σ	Busqueda lineal	Busqueda Binaria	Usar Indice
|X|	Loop anidado	Loop único	SortMerge

Ejemplo de optimización (IV)

Selección de plan

4

- Calculo los costos (cant. de accesos a disco)

Operador	Implementación	Costo
$\sigma_c(R)$	Busqueda Lineal	b_R
	Busqueda Binaria	$Log_2\ b_R + S_c$
	Uso de Indice	$Log_k\|R\| + S_c$
$R\|><|_cT$	Loop Anidado	$b_R+(b_R*b_T)+(js * \|R\|*\|T\|)/bfr_{RS}$

Costo del plan A = 58600

Costo del plan B = 700

Costo del plan C = 10

Plan C al Generador de Código