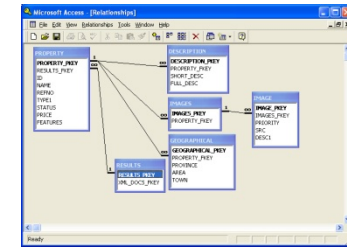


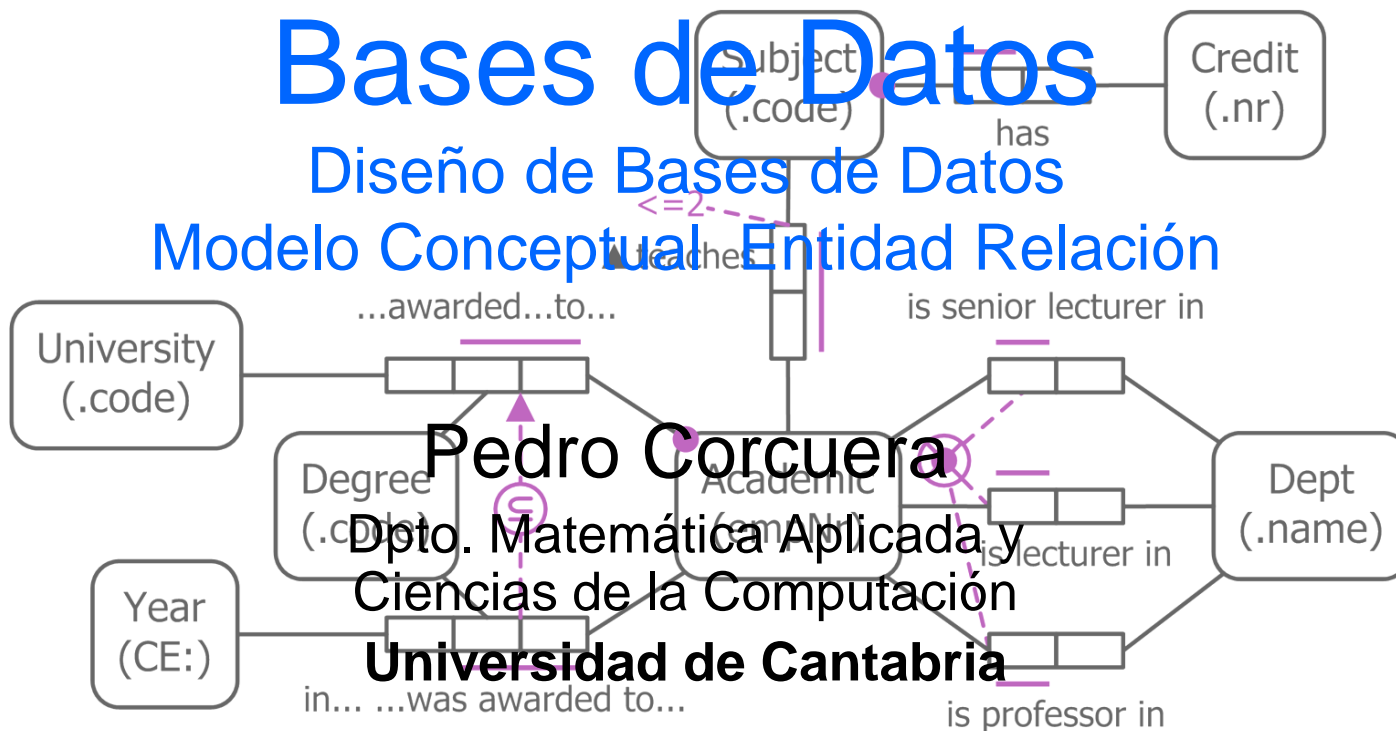
SQL



# Bases de Datos

## Diseño de Bases de Datos

### Modelo Conceptual Entidad Relación



**Pedro Corcuera**  
Dpto. Matemática Aplicada y  
Ciencias de la Computación  
Universidad de Cantabria

corcuerp@unican.es

- Presentar los conceptos y técnicas del diseño de las bases de datos
- Presentar la técnica de modelado entidad-relación

- Modelos de datos
- Fases de diseño
- Modelo Entidad-Relación

- Colección de herramientas conceptuales que se emplean para especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad
  - Los modelos de datos describen las relaciones entre los datos que forman una base de datos
  - No se refieren en ningún momento a los valores específicos que un elemento de datos debe tomar
  - Tratan a los datos como grupos genéricos, que pueden tomar cualquier conjunto de valores específicos

- Un **modelo de datos** es una colección de conceptos para describir a los datos.
- Un **esquema** es una descripción de una colección particular de datos usando un modelo de datos específico.
  - Un SGBD soporta un modelo de datos, que es usado para describir el esquema de la base de datos a utilizar.
- Existen varios modelos de datos. En Bases de Datos se han usado tradicionalmente tres:
  - Jerárquico
  - Redes
  - Relacional

- Otros modelos incluyen:
  - Modelo Entidad-Relacion (ER)
  - Modelo Entidad Relacion Extendido (EER)
  - Modelo Orientado a Objetos
    - Lenguaje de Modelacion Unificado (UML)
  - Otros modelos semánticos (ORM)

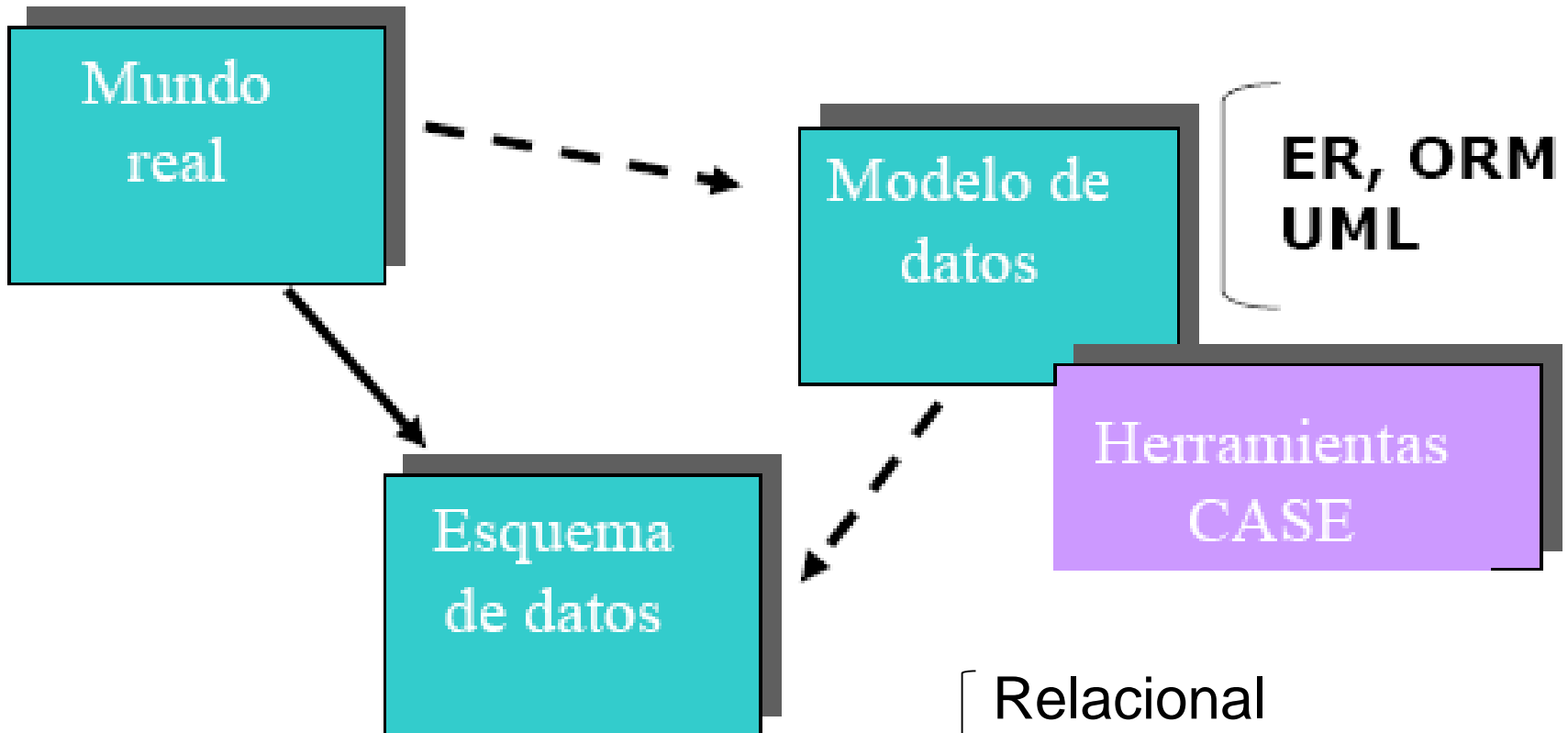


- Fase inicial: análisis de requisitos. Descripción de la información a gestionar y sus procesos. Entrevistas con usuarios y expertos.
  - Análisis de requisitos. Especificación funcional
- Diseño conceptual: traducción del análisis de requisitos al esquema conceptual. Representación generalmente gráfica de las entidades y sus relaciones.
  - Modelo ER, modelo UML, ORM
  - DFD, diagrama de casos, diagramas de colaboración, de secuencia, etc.

- Implantación en el gestor:
  - Diseño lógico: traducción del modelo conceptual al LDD del gestor correspondiente. Modelo relacional, OO, OR
  - Diseño físico: determina la organización de archivos y las estructuras de almacenamiento interno.



# Modelo, Esquema y Ejemplar



**Ejemplar del esquema:** instancia del esquema, esto es, datos que en un momento determinado están en el esquema

- Características:
  - Independientes del SGBD
  - Mayor nivel de abstracción
  - Mayor capacidad semántica
  - Más enfocados al diseño de alto nivel
  - Interfaz usuario/informático

- Desarrollado por Peter Chen ( M.I.T.) en los 70's
- Es un Modelo Conceptual de alto nivel
- Se usa comúnmente para modelar aplicaciones de Bases de datos y en investigación de Bases de Datos
- Representa gráficamente y de manera lógica toda la información y como los datos se relacionan entre sí.
- Es independiente del DBMS en el cual se vaya a implementar

- Diseño conceptual. (ER y EER se usan en este nivel de abstracción)
  - ¿Cuáles son las entidades y relaciones en la empresa?
  - ¿Qué información acerca de esas entidades y relaciones deben almacenarse en la BD?
  - ¿Cuáles son las restricciones de integridad (o reglas de negocio) que se deben mantener?
  - Un esquema de la base de datos en estos modelos se puede representar gráficamente (Diagramas ER)
  - Los diagramas ER se pueden traducir a esquemas relacionales

- Entidad (entity)
  - Objeto que existe y se distingue de los demás
  - Se representa con una serie de atributos
  - Pueden ser concretos. pe.: un libro, una persona,...
  - O abstractas. pe.: préstamo, pedido,...
- Atributo (attribute)
  - Propiedades que caracterizan a las entidades.
  - Clave primaria: atributos que identifican a la entidad. pe.: ISBN (PK), título, idioma, ... para entidad libro

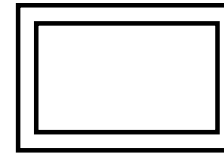
- Dominio (domain)
  - Conjunto de valores permitidos para un atributo
  - P. ej: indicando el tipo de datos (por intención)
  - P. ej: sexo-> M o F (por extensión)
- Entity Extension
  - Conjunto de elementos del mismo tipo. Los conjuntos no necesariamente deben ser disjuntos
- Relación (Relationship)
  - Conexión semántica entre dos o más entidades
- Relationship Instance
  - Conjunto de relaciones del mismo tipo

- Existen dos categorías de tipos de entidades:
  - Regulares o fuertes, que son aquellas cuyos ejemplares tienen existencia por sí mismos
    - Caso préstamos de la biblioteca: LIBRO y AUTOR
  - Débiles, en las cuales la existencia de un ejemplar depende de que exista un cierto ejemplar de otro tipo de entidad
    - Caso del EJEMPLAR que depende de LIBRO

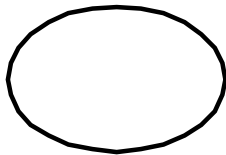
# Elementos gráficos del modelo Entidad - Relación



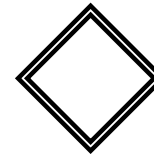
Entidad



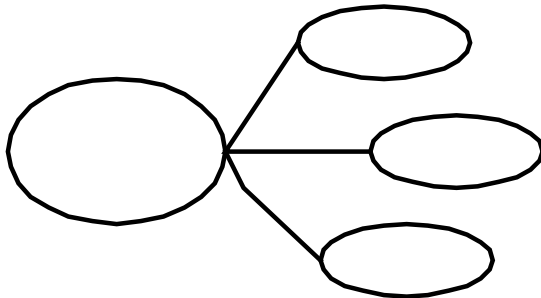
Entidad  
Débil



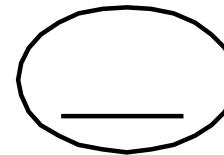
Atributo



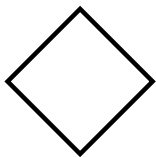
Relación de  
Identificación



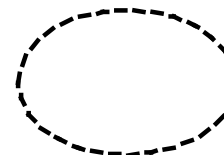
Atributo  
Compuesto



Atributo  
Llave



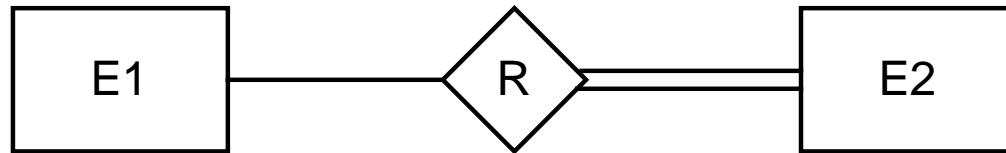
Relacion



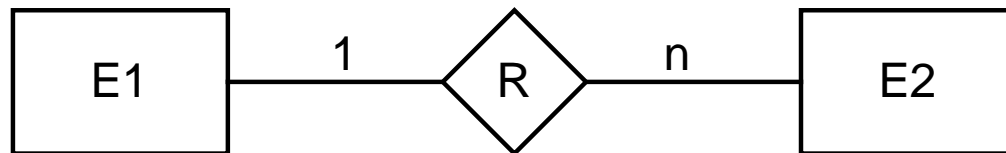
Atributo  
Derivado

————— Enlace

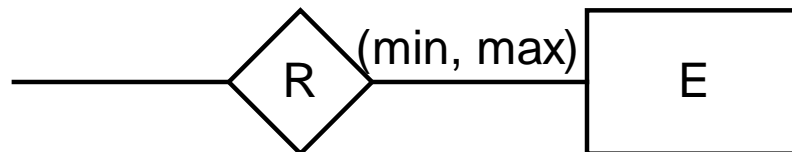




Participacion Total de E2 en R



Relación de Cardinalidad 1:n para E1:E2 en R

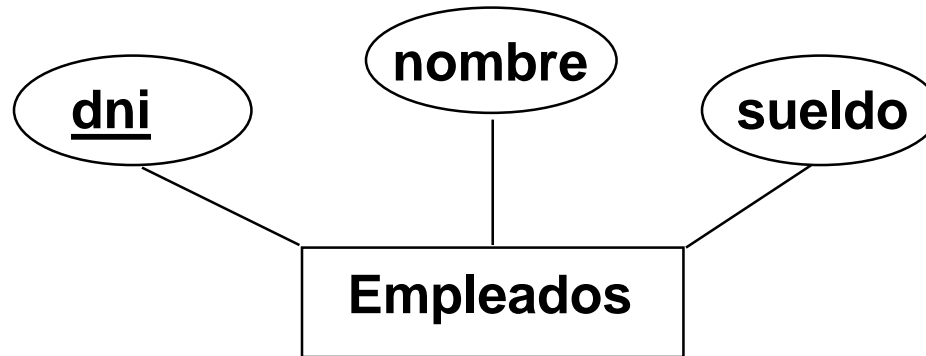


Restriccion Estructural (min, max) en la participacion de E en R

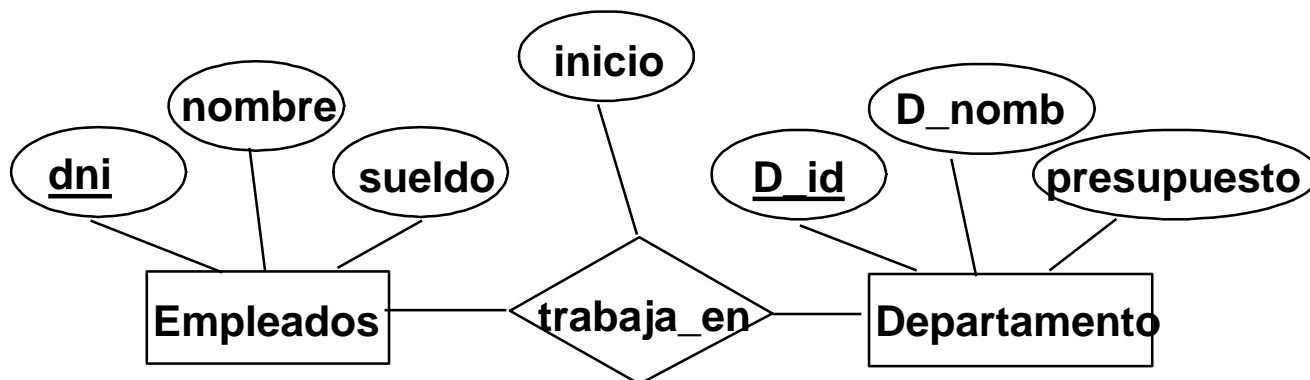
- 
- Tres elementos básicos: entidades, atributos y relaciones.
    - ENTIDAD: Es todo objeto de datos que es diferenciable de otros objetos, ya sean abstractos o concretos.
    - ATRIBUTO: Permite describir a una entidad. Los atributos describen propiedades o características de una entidad
    - RELACIÓN : Describe la conexión o asociación existente entre **dos** o **más** entidades.

- 
- Cada atributo de una entidad tiene un nombre
  - El conjunto de valores permitidos para cada atributo se denomina **dominio** del atributo
  - Se requiere (normalmente) que los valores de los atributos sean **atómicos**; esto es, indivisibles
    - Los valores de los atributos multivalorados son no atómicos
    - Los valores de los atributos compuestos son no atómicos
  - El valor especial *null* es aceptable pero crea complicaciones en algunas operaciones

- ENTIDAD
- ATRIBUTO



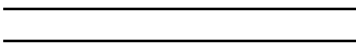

- RELACIÓN



- Cardinalidad: n<sup>o</sup> máximo de unidades de un conjunto que se conecta o relaciona con una entidad de otro y viceversa
- Cardinalidad 1:1
  - Una instancia de la entidad *A* está asociada con 0 o 1 instancia de la entidad *B*
  - Una instancia de la entidad *B* está asociada con 0 o 1 instancia de la entidad *A*

- Cardinalidad 1:N
  - Una instancia de la entidad *A* está asociada con 0 o más instancias de la entidad *B*
  - Una instancia de la entidad *B* está asociada con 0 o 1 instancia de la entidad *A*
- Cardinalidad M:N
  - Una instancia de la entidad *A* está asociada con 0 o más instancias de la entidad *B*
  - Una instancia de la entidad *B* está asociada con 0 o más instancias de la entidad *A*

- Participación total
  - Se representa con doble línea del lado de la entidad en la que todas sus instancias deben estar asociadas con alguna instancia de la otra entidad.

- Restricción de estructura = cardinalidad + restricción de participación
- **Cardinalidad:** Como se relaciona 1 instancia de una entidad con respecto a las instancias de la otra entidad, Ej., 1:1, 1:N, M:N
- **Participación:** Determina la dependencia de una instancia de una entidad con respecto a las instancias de la otra entidad.
  - Total: Ej. Un empleado debe trabajar en un departamento (doble línea en el diagrama ER) 
  - Parcial: Ej. Un empleado puede ser administrador (línea sencilla en el diagrama ER) 



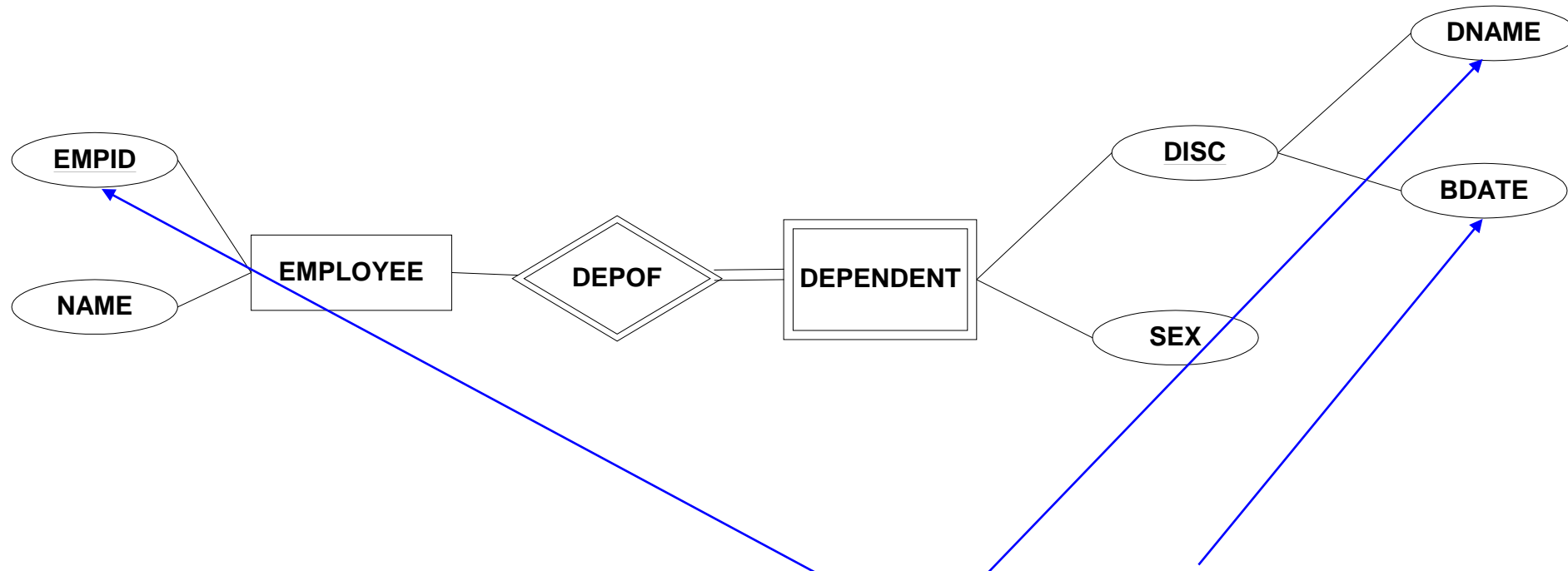
- Una instancia de una entidad debe distinguirse de otras instancias. Se debe especificar una llave (KEY)
  - **Llave (KEY):**  
conjunto de atributos de una entidad que identifica de manera única una instancia de dicha entidad.
  - **Super llave (Superkey):**  
conjunto de uno o más atributos de una entidad que identifica de manera única una instancia de dicha entidad
  - **Llave candidata (Candidate Key):**  
Super llave para la que ningún subconjunto propio es una superllave (contiene los atributos mínimos para identificar la tupla)

- **Primary Key:**  
llave seleccionada por el DBA como el principal medio para identificar de manera única las instancias de una entidad
- En el modelo ER se subraya el atributo que es la llave primaria para esa entidad

- **Strong Entity**: entidad que tiene una llave Primaria
- **Weak Entity**: entidad que no tiene suficientes atributos para formar una llave primaria
- La llave primaria (primary key) de una entidad débil se forma con la llave primaria de la entidad fuerte (string) de la cual depende, concatenada con un discriminador (discriminator). El discriminador es el conjunto de atributos que permite distinguir las instancias de la entidad débil.

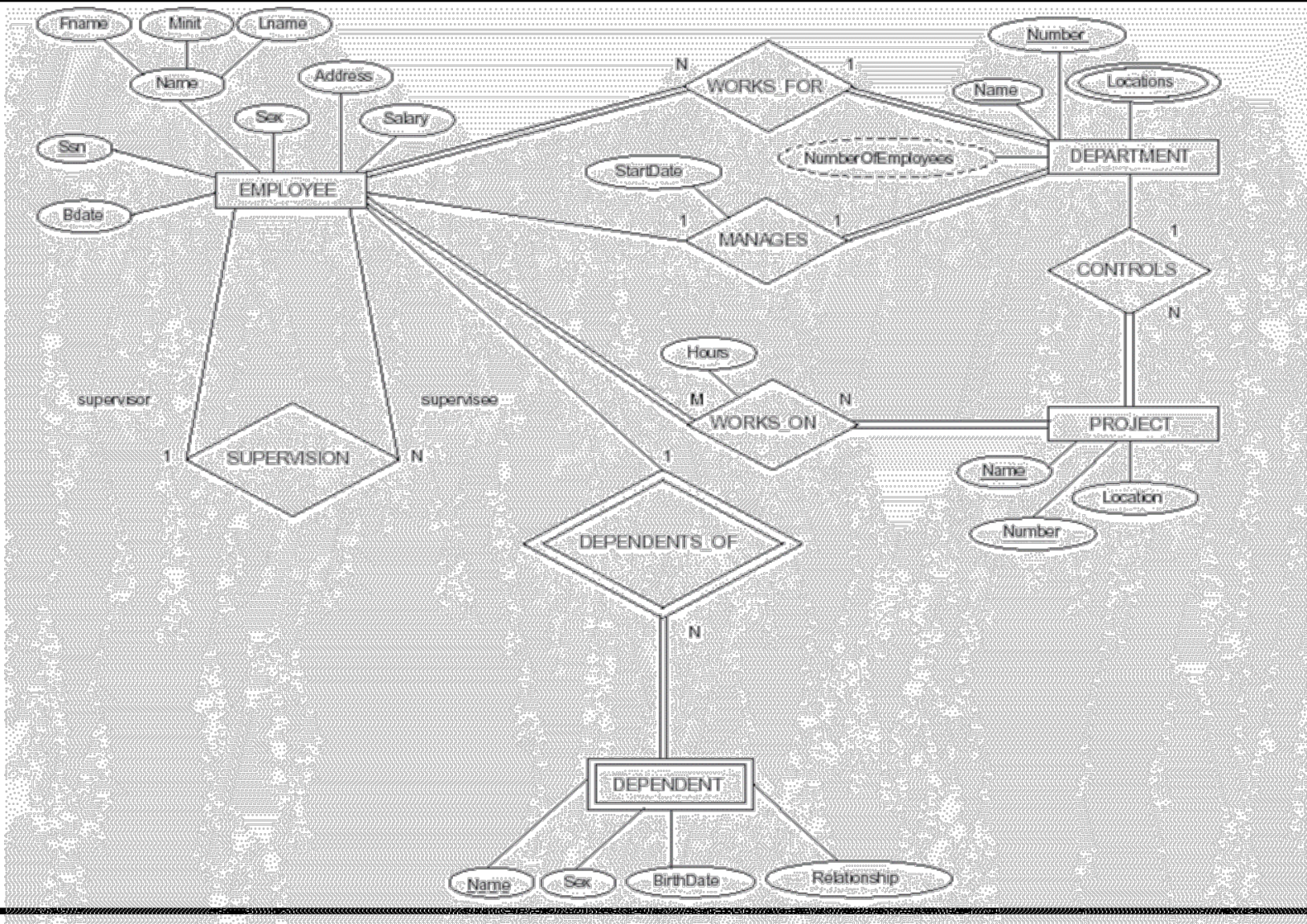
# Weak Entity Example

- Entidades débiles: Una entidad débil solo puede ser identificada de manera única al considerar la llave primaria de otra entidad (dueña)



Dependent's primary key: {empid, dname, birthdate}

## Análisis diagrama ER



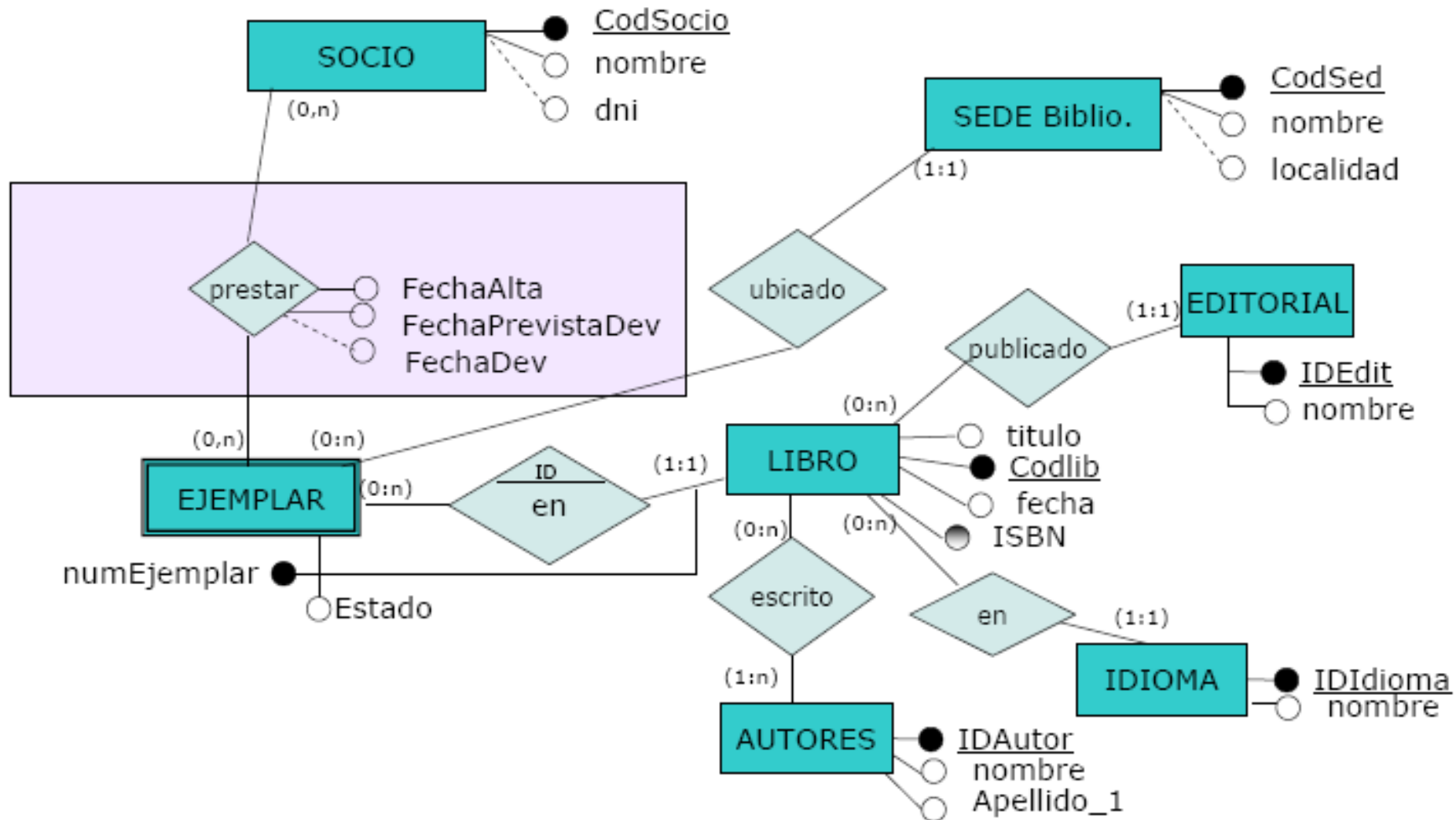
- **Empleado**
  - Trabaja exactamente en un dept.
  - Puede o no administrar un solo departamento.
  - Trabaja en al menos 1 proyecto y máximo en  $n$  proyectos
  - Puede o no tener dependientes pero máximo  $n$  dependientes
  - Puede o no ser supervisor y si supervisa, supervisa a máximo  $n$  empleados
  - Puede o no ser supervisado por un solo supervisor.
- **Departamento**
  - Tiene al menos 1 empleado (max  $N$ )
  - Tiene exactamente un administrador
  - Puede o no controlar proyectos (max  $N$ )
- **Proyecto**
  - Controlado por exactamente 1 departamento
  - Tiene al menos 1 empleado asignado (max  $N$ )
- **Dependiente**
  - Depende de exactamente un empleado

## Ejemplo: Gestión de préstamos

---

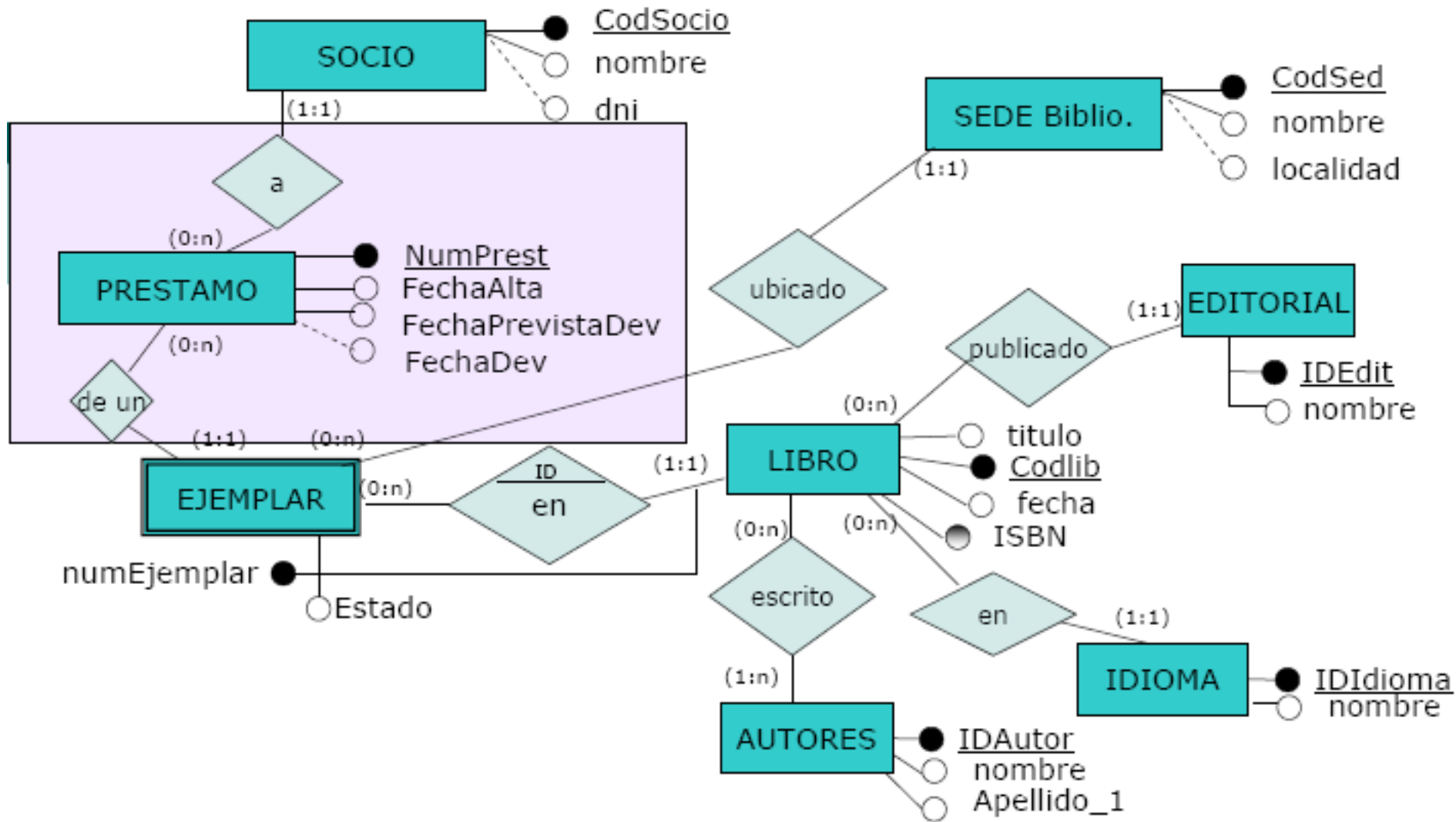
- Una biblioteca está interesada en automatizar la gestión de **préstamos** cuyo funcionamiento es:
    - Registrar el **socio** que se lleva el ejemplar del **libro**, fechas de entrega, devolución prevista y de devolución
    - La biblioteca está organizada en diversas **sedes** y el socio puede coger libros de cualquiera de ellas
    - Del socio se tienen los datos personales básicos
    - Y de los libros, todos los campos descriptivos que los caracterizan (**título, idioma, autores, editorial**, fecha,...).
    - Además de cada ejemplar se querrá conocer el estado en el que se encuentra (prestable, en reparación, fuera de circulación)
-

## Ejemplo: Gestión de préstamos





## Ejemplo: Gestión de préstamos

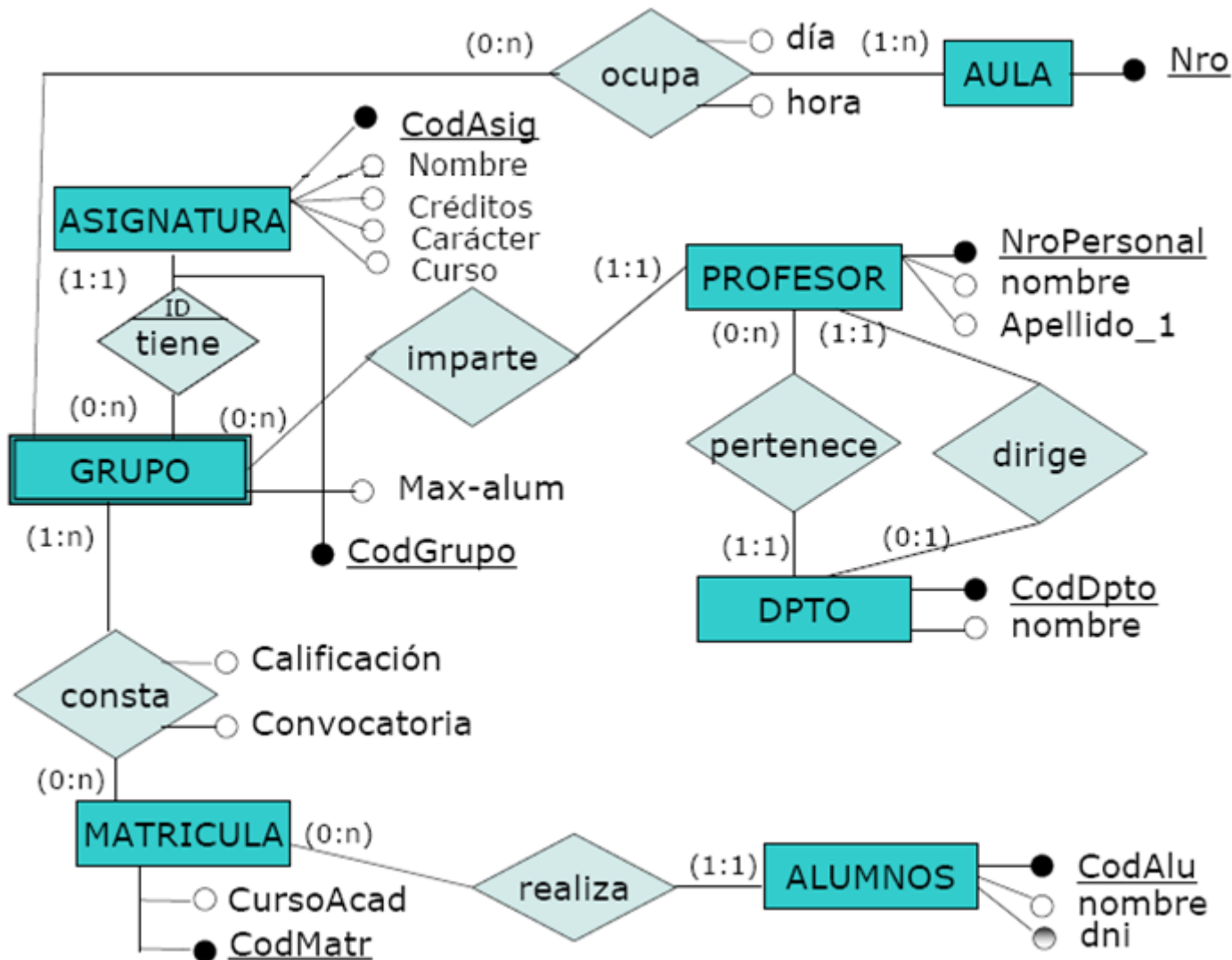


## Ejemplo: Gestión docente

---

- Cada **profesor** pertenece a un sólo **departamento** y debe pertenecer a uno
- El profesor puede impartir varios **grupos** de la misma o distinta **asignatura**, y un grupo debe ser enseñado por un profesor
- Los **alumnos** se matriculan de varias asignaturas (al menos una) cada curso académico pero han de hacerlo en un grupo. A su vez un grupo tendrá varios alumnos matriculados. Cada grupo tendrá asignado un **aula** para cada día y hora de la semana.
- La **matrícula** dará opción a dos convocatorias de examen con su respectiva calificación.
- Todo departamento debe tener un director, que es profesor
- Los atributos de cada entidad son los que cabría esperar

## Ejemplo: Gestión docente

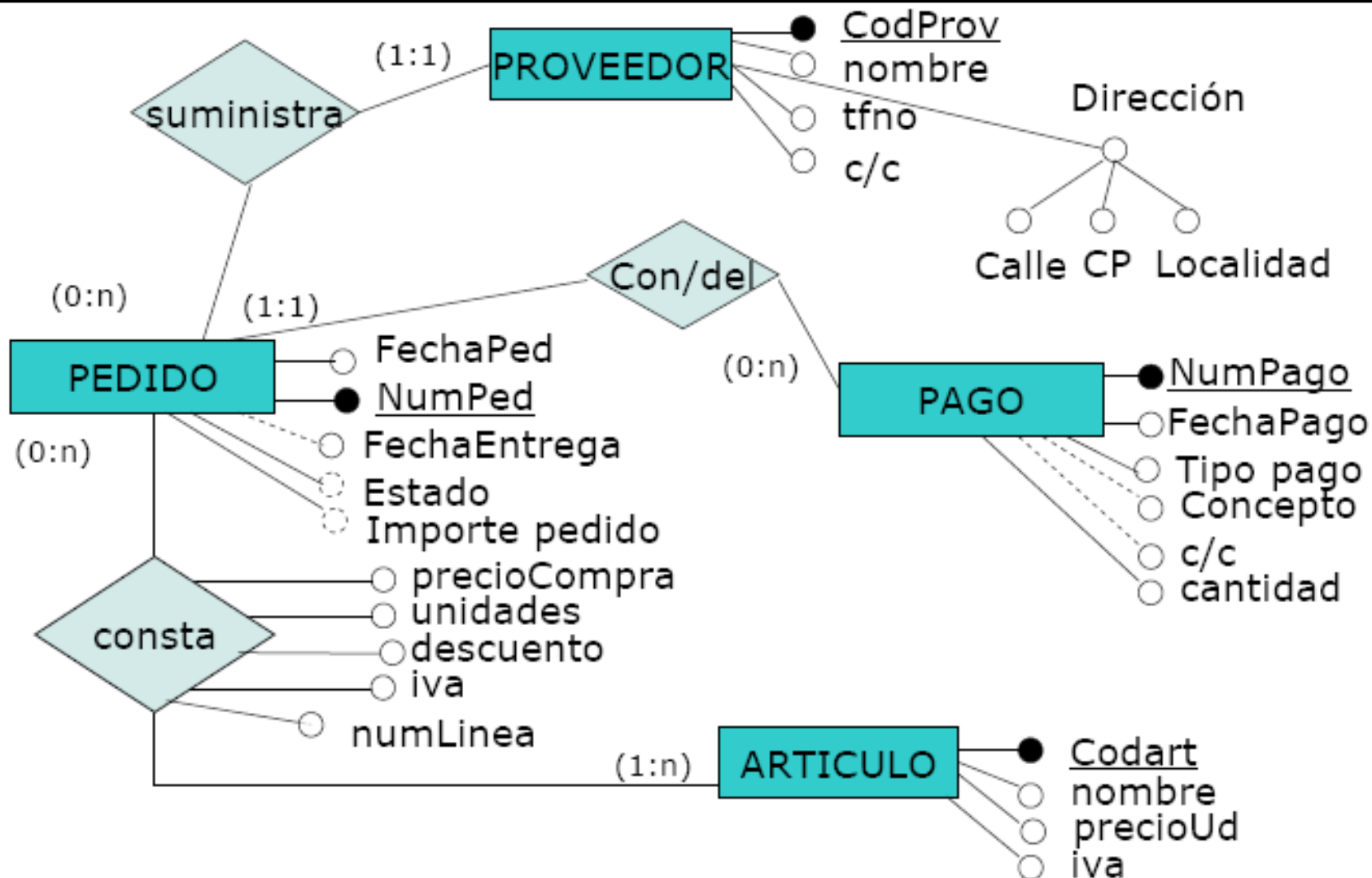


## Ejemplo: Gestión de compras

---

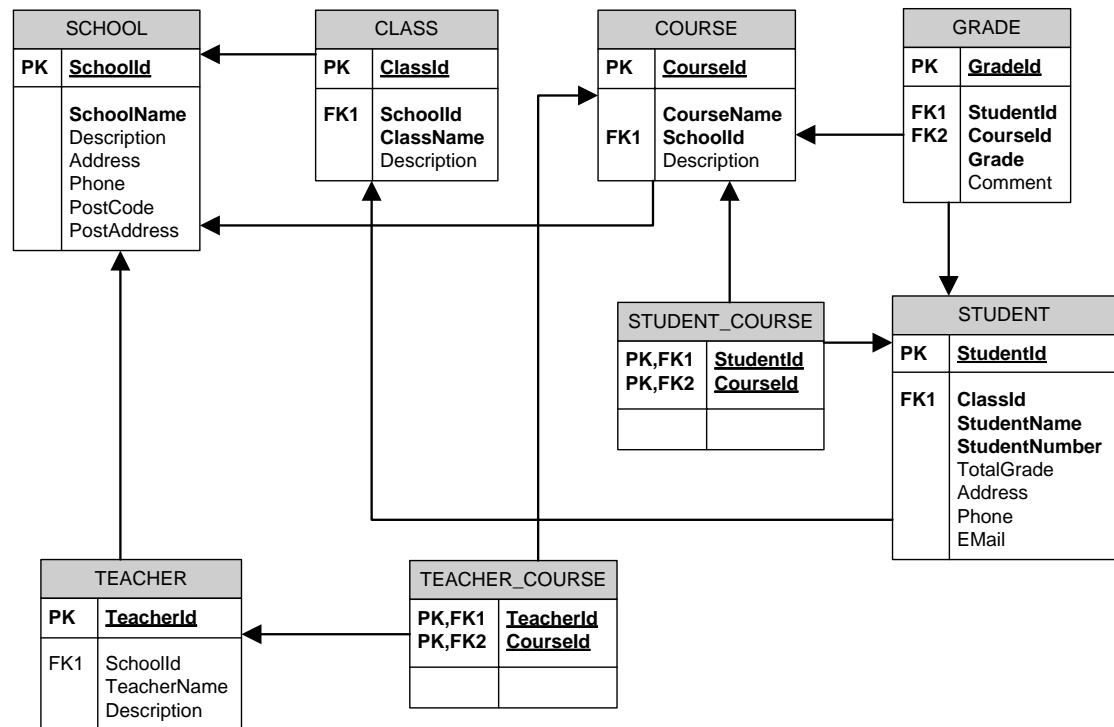
- Una empresa está interesada en automatizar su proceso de compras cuyo flujo de funcionamiento es:
  - Requiere registrar la hoja del pedido que realiza a un determinado proveedor en una determinada fecha
  - En la hoja del pedido queda constancia del número de unidades que compra de cada artículo y el precio de compra, y en caso de que el proveedor o bien por volumen o por promoción, le realiza un descuento, también lo anota
  - Los productos que compran tienen distinto IVA
  - Generalmente el paga a sus proveedores al mes de recibir la mercancía y por transferencia, aunque lo puede hacer a plazos
  - Los atributos de cada entidad son los que cabría esperar

## Ejemplo: Gestión de compras



- MS Visio
- PowerDesigner
- ER/Studio
- Toad Data Modeler
- MySQL Workbench

## Ejemplo de diseño en MS Visio



- Las bases de datos relacionales se basan en el uso de tablas (también llamadas relaciones).
- Las tablas se representan gráficamente como una estructura rectangular formada por filas y columnas.
- Cada fila (tuplas o registros) posee una ocurrencia de la instancia o relación representada por la tabla.
- Cada columna almacena información sobre una propiedad (atributo) determinada de la tabla. Cuando no se conoce el valor de un atributo se le asigna el valor nulo.

- Relación (**Tabla**). Son los objetos principales sobre los que debe recogerse la información. Una relación tiene un nombre, un conjunto de atributos y está formada por un conjunto de tuplas con los valores de cada uno de los atributos. Se representa mediante una tabla bidimensional donde las columnas representan los atributos y las filas representan las tuplas o registros.
- Tupla o **registro**. Corresponde a una fila de la tabla. El número de tuplas se denomina *cardinalidad*, la cardinalidad varía con el tiempo.



- **Dominio.** Es una colección de valores, de los cuales uno o más atributos obtienen sus valores reales. Pueden ser finitos o infinitos.
- **Atributo.** Corresponde a una columna de la tabla (equivalente a un campo de un registro) y se definen sobre dominios. El número de atributos se llama *grado*. El grado no varía con el tiempo, si añadimos un atributo a una relación, podemos considerar que se trata de otra relación nueva.

- **Clave candidata** es un atributo  $K$  (o conjunto de atributos) de una relación  $R$  que cumple dos propiedades:
  - **Unicidad:** No existen dos tuplas en  $R$  con el mismo valor de  $K$ .
  - **Minimalidad:** Si  $K$  es compuesto, no será posible eliminar ningún componente de  $K$  sin destruir la propiedad de unicidad.

## Claves de una relación

---

- **Clave primaria.** Es posible que una relación posea más de una clave candidata, en ese caso, se escoge una de ellas como clave primaria y el resto se denominan claves alternativas.
- Toda relación tiene una clave primaria y suele representarse subrayando y/o añadiendo el carácter # al atributo (o conjunto de atributos) correspondiente.

# Restricciones Inherentes del Modelo Relacional

---

- No existen tuplas repetidas (obligatoriedad de clave primaria).
- El orden de las tuplas y el de los atributos no es relevante.
- Cada atributo de cada tupla solo puede tomar un único valor sobre el dominio sobre el que está definido.
- Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo (regla de integridad de entidad)

# SQL Restricciones Semánticas o de Usuario

---

- Restricción de Clave Primaria (PRIMARY KEY), permite declarar un atributo o conjunto de atributos como la clave primaria de una relación.
- Restricción de Unicidad (UNIQUE), permite que una clave alternativa o secundaria pueda tomar valores únicos para las tuplas de una relación. Se entiende que la clave primaria siempre tiene esta restricción.
- Restricción de Obligatoriedad (NOT NULL), permite declarar si uno o varios atributos de una relación debe tomar siempre un valor.

# SQL Restricciones Semánticas o de Usuario

---

- Restricción de Integridad Referencial o de Clave Foránea (FOREIGN KEY), se utiliza para que mediante claves foráneas podamos enlazar relaciones de una base de datos.
- Restricción de Valor por Defecto (DEFAULT), permite que cuando se inserte una tupla o registro en una tabla, para aquellos atributos para los cuales no se indique un valor exacto se les asigne un valor por defecto.

# SQL Restricciones Semánticas o de Usuario

---

- Restricción de Verificación o Chequeo (CHECK), en ocasiones puede ocurrir que sea necesario especificar una condición que deben cumplir los valores de determinados atributos de una relación de la BD.
- Aserciones (ASSERTION): generaliza a la anterior, lo forman las aserciones en las que la condición se establece sobre elementos de distintas relaciones.
- Disparadores (TRIGGERS), puede interesar especificar una acción cuando no se cumple una determinada restricción semántica.

# SQL(Structure Query Language)

---

- SQL es un lenguaje estándar diseñado para la gestión de datos de una Sistema gestor de bases de datos relacional.
- SQL proporciona métodos para definir la base datos, para manipular la información y para gestionar los permisos de acceso a dicha información.
- Para que un gestor sea relacional, debe soportar SQL.



# SQL(Structure Query Language)

---

- Las instrucciones SQL se dividen en tres grupos:
  - DDL (Data Definition Language) que se encarga de la definición de bases de datos, tablas, vistas e índices entre otros.
    - Son comandos propios de este lenguaje: CREATE
  - DML (Data Manipulation Language), instrucciones para manipular los datos y construir consultas
    - Comandos típicos: SELECT, UPDATE, INSERT, DELETE
  - DCL (Data Control Language) Se encarga del control de accesos y privilegios entre los usuarios.
    - Comandos típicos: CREATE USER, GRANT, REVOKE

- Consulta que devuelve todos los productos con un descuento mínimo del 25 por ciento:

```
SELECT * FROM Products
WHERE ProductID IN
(SELECT ProductID FROM OrderDetails
WHERE Discount >= .25);
```

- Consulta que devuelve los nombres de los empleados cuyos salarios son iguales o mayores que el salario medio de todos los empleados que tienen el mismo cargo.

```
SELECT LastName, FirstName, Title, Salary
FROM Employees AS T1
WHERE Salary >= (SELECT Avg(Salary) FROM Employees
WHERE T1.Title = Employees.Title) Order by Title;
```

---

- La normalización de bases de datos es un proceso por el cual un esquema existente se modifica para traer sus tablas componentes hacia el cumplimiento a través de una serie de formas normales progresivas.
- Fue introducida como procedimiento por E. Codd, en 1970.
- La normalización conlleva beneficios porque reduce el espacio de almacenamiento mediante la categorización inteligente de los datos, permitiendo realizar búsquedas mejores y más rápidas.

- El proceso de normalización de base de datos consiste en la obtención de datos para ajustarse a las formas estándares progresivas (hasta 5).
  - 1NF, los valores de cada columna de una tabla son atómicas; lo que significa que son únicos, y no contiene conjuntos de valores.
  - 2NF, reduce los datos superfluos en una tabla seleccionándolos, poniéndolos en nuevas tablas y estableciendo relaciones entre ellos. Se trata de las relaciones entre las columnas de clave compuesta y las columnas que no son clave, de forma que las columnas sin clave tienen que depender de la clave compuesta completa.
  - 3NF, requiere que todas las columnas dependan directamente de la clave primaria.