



Clase

# **Modelado Lógico: El modelo relacional**

**(Implementable)**

2

Mundo  
Real



Modelo  
Conceptual

Esquema  
Entidad  
Relación

No  
depende  
del DBMS

Modelo  
Lógico  
(implementable)

Esquema  
Relacional

Depende  
del tipo de  
DBMS

Modelo  
Físico

Esquema  
Físico

Depende  
del DBMS

Un modelo lógico es una **descripción de la Base de Datos en términos de la estructura de datos** que pueda procesarse en un tipo de DBMS.



# El modelo Relacional

- El **modelo relacional**, (1970 - **Edgar Frank Codd**) está basado en la lógica de predicados y en la teoría de conjuntos. Se consolidó como el paradigma del modelado de base de datos, y persigue una serie de **objetivos**:

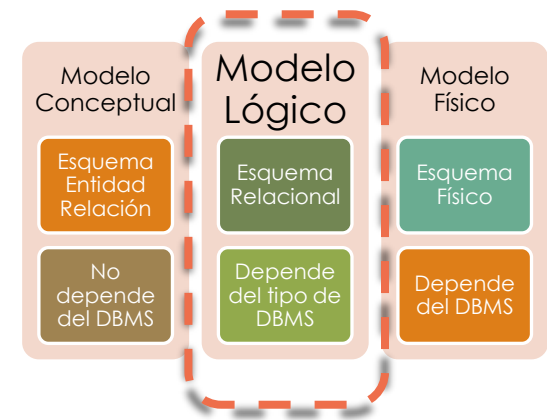
## Independencia física

- El modo cómo se almacenan los datos no debe influir en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no han de modificar sus programas por cambios en el almacenamiento físico

## Independencia lógica

- Añadir, eliminar o modificar cualquier elemento de la BD no debe repercutir en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).

# Características del modelo Relacional



## Fundamentación teórica

- Al estar el modelo definido con rigor matemático, el diseño y la evaluación del mismo puede realizarse por métodos sistemáticos basados en abstracciones.

## Flexibilidad

- Ofrecer a cada usuario los datos de la forma más adecuada a la correspondiente aplicación.

## Uniformidad

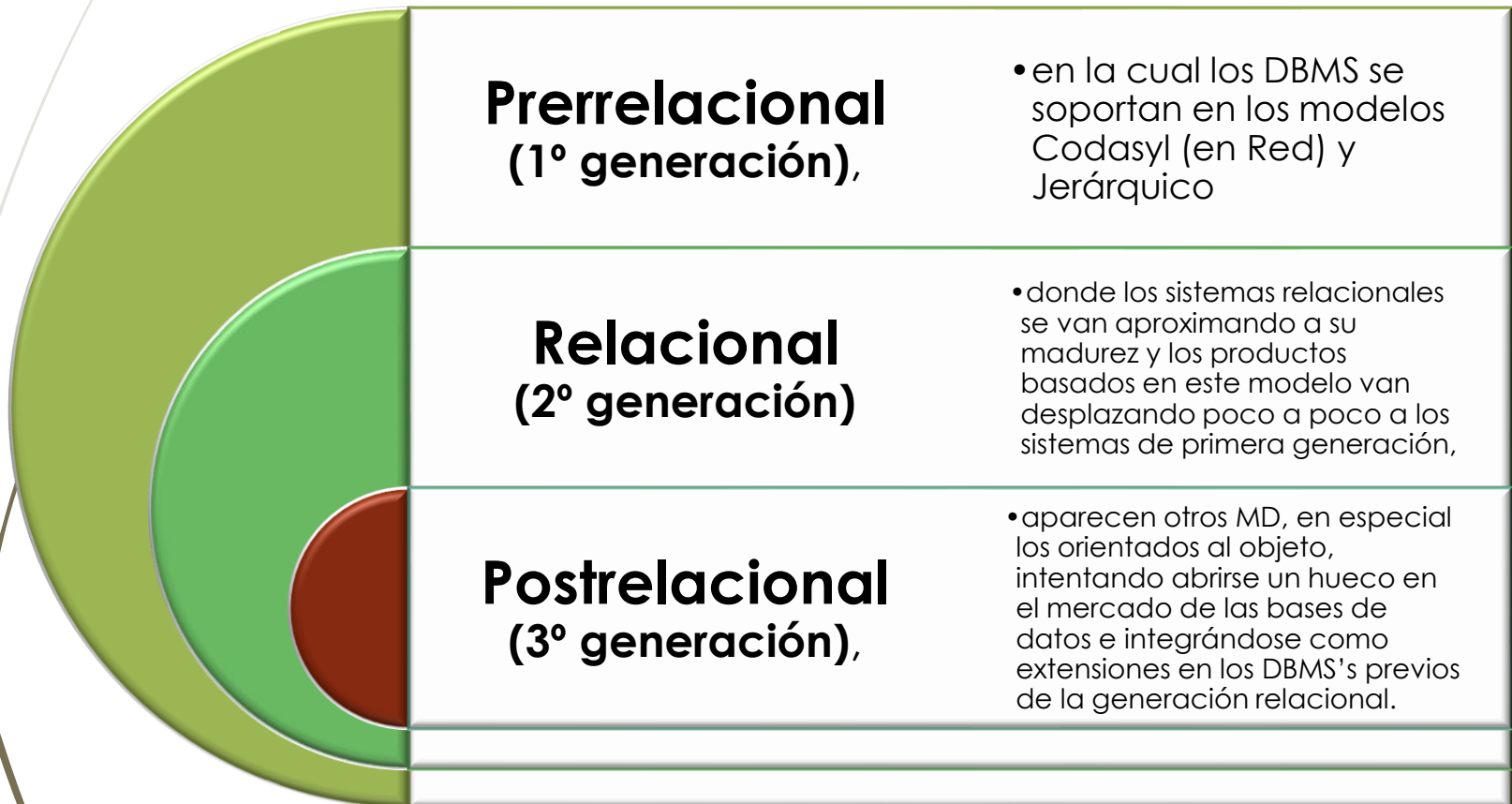
- Las estructuras lógicas de los datos presentan un aspecto uniforme (tablas), lo que facilita la concepción y manipulación por los usuarios

## Sencillez

- Las características anteriores, producen como resultado que el modelo relacional (MR) sea fácil de comprender y de utilizar por parte del usuario.

# Generaciones de DBMS's

La aparición del MR representa un hito en el desarrollo de las BD, ya que ha marcado **tres etapas diferentes**, conocidas como generaciones de los DBMS's:



# DBMS's comerciales

Las ventajas citadas han contribuido al hecho de que desde mediados de los años 80, el MR sea utilizado por prácticamente la totalidad de los **DBMS comerciales**.



Algunas de las principales empresas informáticas del mundo, son en origen, empresas de DBMS: **ORACLE**, Sybase, INFORMIX, ...

Los grandes fabricantes de software tienen "su" DBMS relacional: IBM, DB2, Microsoft SQL Server, ...

Existen varios DBMS diseñados para PCs y usuarios no expertos: Microsoft Access, etc.

El tremendo éxito real del MR ha promovido, junto al **cambio tecnológico**, la aparición de los **DBMS Objeto-Relacionales**, y fracasan, en general, los DBMS de Objetos puros

# El modelo relacional

- El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos:

- la **estructura** de datos,

- la **integridad** de los datos y

Componentes  
estáticas

- la **manipulación** de los mismos.

Componente  
dinámica

- En los últimos años, se han propuesto algunas extensiones al modelo relacional para capturar mejor el significado de los datos, para disponer de los conceptos de la orientación a objetos y para disponer de capacidad deductiva.

# Estructura de los Datos

Analizamos los siguientes elementos básicos:

## RELACIÓN O TABLA

Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.

## ATRIBUTO

Representa las propiedades de la relación. Se representa mediante una **columna**.

## DOMINIO

Es el conjunto válido de **valores** que toma un atributo.

## TUPLA

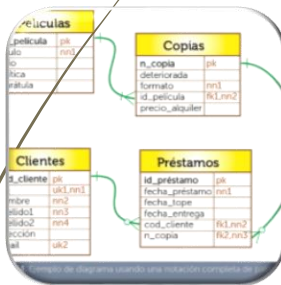
Es una ocurrencia de la relación. Se representa mediante una **fila**.



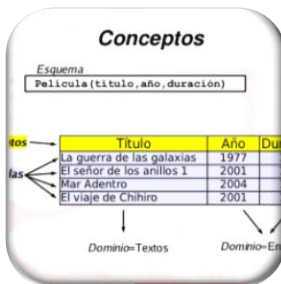
# La tabla (relación)...

El modelo relacional se basa en el concepto matemático de relación, que gráficamente se representa mediante una tabla.

**Codd**, que era un experto, utilizó terminología perteneciente a las matemáticas: **de la teoría de conjuntos y de la lógica de predicados**.



En el modelo relacional, las relaciones o TABLAS se utilizan para almacenar información sobre los objetos que se representan en la base de datos.



*Una relación es una tabla con filas y columnas.*

# Las tablas (relaciones)

- Un DBMS **sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas.**
- **No se aplica a la estructura física de la base de datos,** que se puede implementar con distintas estructuras de almacenamiento.

## Campos o Atributos

Registros  
o  
Tuplas

| Nombre    | Nacionalidad | Institución |
|-----------|--------------|-------------|
| Sofia     | España       | O.N.U.      |
| John      | EE. UU.      | O.M.S.      |
| Jacob     | Francia      | N.A.S.A.    |
| Sebastian | Chila        | UNASUR      |

# Los atributos (columnas)

- Una relación se representa gráficamente como una **tabla bidimensional** en la que las filas corresponden a tuplas individuales y **las columnas** corresponden a los **campos o atributos de esos registros**.
- Los atributos almacenan **una propiedad o característica** de la entidad de la que trata la **tabla** y pueden aparecer en la relación en cualquier orden.
- *es el nombre de una columna de una relación.*

# Un dominio...

*Un dominio es el conjunto de valores legales de uno o varios atributos.*

- Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.

**Un dominio  $D$  es un conjunto de valores atómicos. Por **atómico** se entiende que cada valor del dominio es indivisible en lo tocante al modelo relacional.**



# El dominio...

- El concepto de dominio es importante porque permite que el usuario **defina**, en un lugar común, el **significado y la fuente de los valores** que los atributos pueden tomar.
- *Esto hace que haya más información disponible para el sistema cuando éste va a ejecutar una operación relacional*, de modo que las operaciones que son semánticamente incorrectas, se pueden evitar.

# Las TUPLAS (filas)

- Los elementos tipo “filas” de una **TABLA** son las denominadas **tuplas**.
- Las tuplas de una relación no siguen ningún orden

► *Una tupla es una fila de una tabla.*

Conceptualmente constituye un:

**ejemplar, ocurrencia o instanciación.**

# Otros conceptos redefinidos

## GRADO

*El grado de una relación es el número de atributos que contiene.*

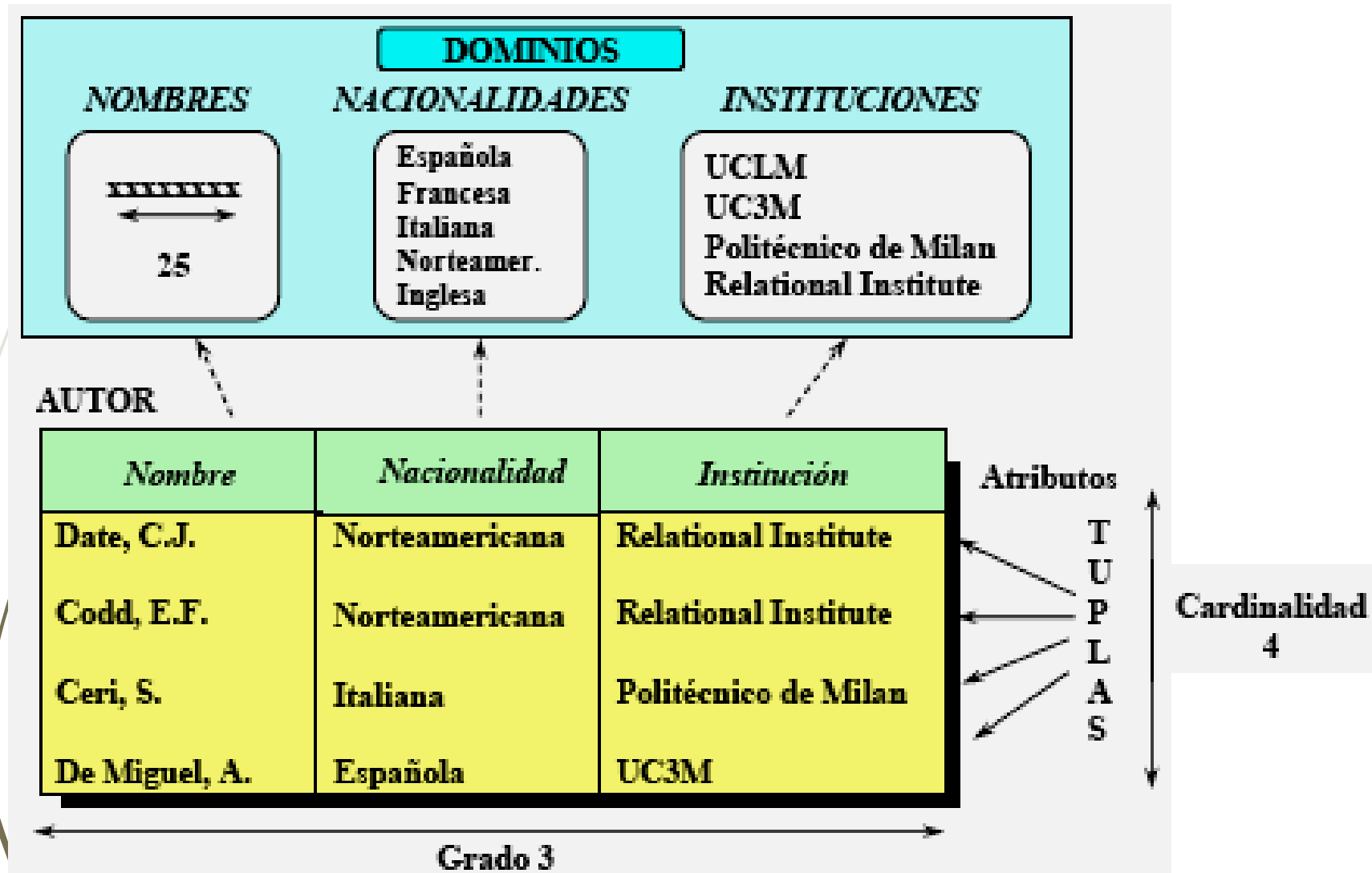
- El grado de una relación no cambia con frecuencia.

## CARDI- NALIDAD

*La cardinalidad de una relación es el número de tuplas que contiene.*

- Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de las mismas varía constantemente

# Una base de datos relacional es un conjunto de relaciones normalizadas





# En relación a la terminología...

**RELACIÓN**

TUPLA  
ATRIBUTO  
GRADO  
CARDINALIDAD

Modelo  
relacional  
(teoría)

**TABLA**

FILA  
COLUMNA  
Nº DE COLUMNAS  
Nº DE FILAS

DBMS  
relacionales  
(implementación)

**FICHERO**

REGISTRO  
CAMPO  
Nº DE CAMPOS  
Nº DE REGISTROS

Sistemas de  
Archivos  
Tradicionales



# En relación a la terminología...

## Relación – intensión vs extensión

### ***ESQUEMA (INTENSIÓN):***

**AUTOR** (*Nombre: Nombres, Nacionalidad: Nacionalidades, Institución: Instituciones*)

### ***RELACIÓN (EXTENSIÓN, ESTADO u OCURRENCIA):***

**AUTOR**

| <i>Nombre</i> | <i>Nacionalidad</i> | <i>Institución</i>   |
|---------------|---------------------|----------------------|
| Date, C.J.    | Norteamericana      | Relational Institute |
| De Miguel, A. | Española            | UC3M                 |
| Ceri, S.      | Italiana            | Politécnico de Milan |

# Propiedades de las Tablas

Cada Tabla tiene un nombre y éste es distinto del nombre de todas las demás.

Los valores de los atributos son atómicos: en cada tupla, cada atributo toma un solo valor. Se dice que las relaciones están normalizadas.

No hay dos atributos que se llamen igual.

El orden de los atributos no importa: los atributos no están ordenados, lo importante es mantener la correspondencia entre atributos y valores

Cada tupla es distinta de las demás: no hay tuplas duplicadas

El orden de las tuplas no importa: las tuplas no están ordenadas

# Revisión y nuevos conceptos

En el capítulo anterior fue definido, en referencia a los atributos, el concepto de **Identificador Clave** para responder a cierta restricción.

## Nuevo concepto: **Claves Candidatas**

- El atributo o conjunto de atributos  $K$  de la relación  $R$  es una clave candidata para  $K$  si y sólo si satisface las sig. propiedades:
  - **Unicidad:** nunca hay dos tuplas en la relación  $R$  con el mismo valor de  $K$ .
  - **Irreductibilidad:** ningún subconjunto de  $K$  tiene la propiedad de unicidad, es decir, no se pueden eliminar componentes de  $K$  sin destruir la unicidad

**Clave Primaria** es un atributo de tipo **"clave candidata"** **que se escoge para identificar sus tuplas de modo único.**

**Ya que una relación no tiene tuplas duplicadas, siempre hay por lo menos una clave candidata y, por lo tanto, la relación *siempre* tiene clave primaria.**

# Nueva Conceptualización

Una base de datos relacional es una **colección de tablas**, pero es necesario, además **ASOCIAR unas tablas con otras**, de modo de completar su noción básica de “conjunto de datos relacionados”.

## Concepto de CLAVE AJENA

Se denomina *Clave Ajena* de una relación  $R_2$  a un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de una clave candidata de una relación  $R_1$ .

# Clave AJENA o clave FORANEA

- Una **clave ajena** es un atributo o un conjunto de atributos de una **tabla** cuyos valores **coinciden con los valores de la clave primaria de alguna otra tabla** (puede ser la misma).
- **Las claves ajenas representan relaciones entre datos.**

# Modelo relacional

| equipos   |       |       |
|-----------|-------|-------|
| id_equipo | n(3)  | pk    |
| nombre    | v(50) | nn,uk |

| jugadores  |       |       |
|------------|-------|-------|
| id_jugador | n(5)  | pk    |
| nombre     | v(50) | nn,uk |
| id_equipo  | n(3)  | nn,fk |

## Ejemplo de datos

| jugadores  |             |           |
|------------|-------------|-----------|
| id_jugador | nombre      | id_equipo |
| 1          | C. Ronaldo  | 1         |
| 2          | Messi       | 2         |
| 3          | Griezmann   | 3         |
| 4          | Luis Suárez | 2         |
| 5          | Bale        | 1         |

Clave secundaria  
FK

| equipos   |                    |
|-----------|--------------------|
| id_equipo | nombre             |
| 1         | Real Madrid        |
| 2         | F.C. Barcelona     |
| 3         | Atlético de Madrid |

Clave primaria  
PK

Ilustración 6. Ejemplo de base de datos relacional de dos tablas

## Películas

|             |     |
|-------------|-----|
| id_película | pk  |
| título      | nn1 |
| año         |     |
| crítica     |     |
| carátula    |     |

## Copias

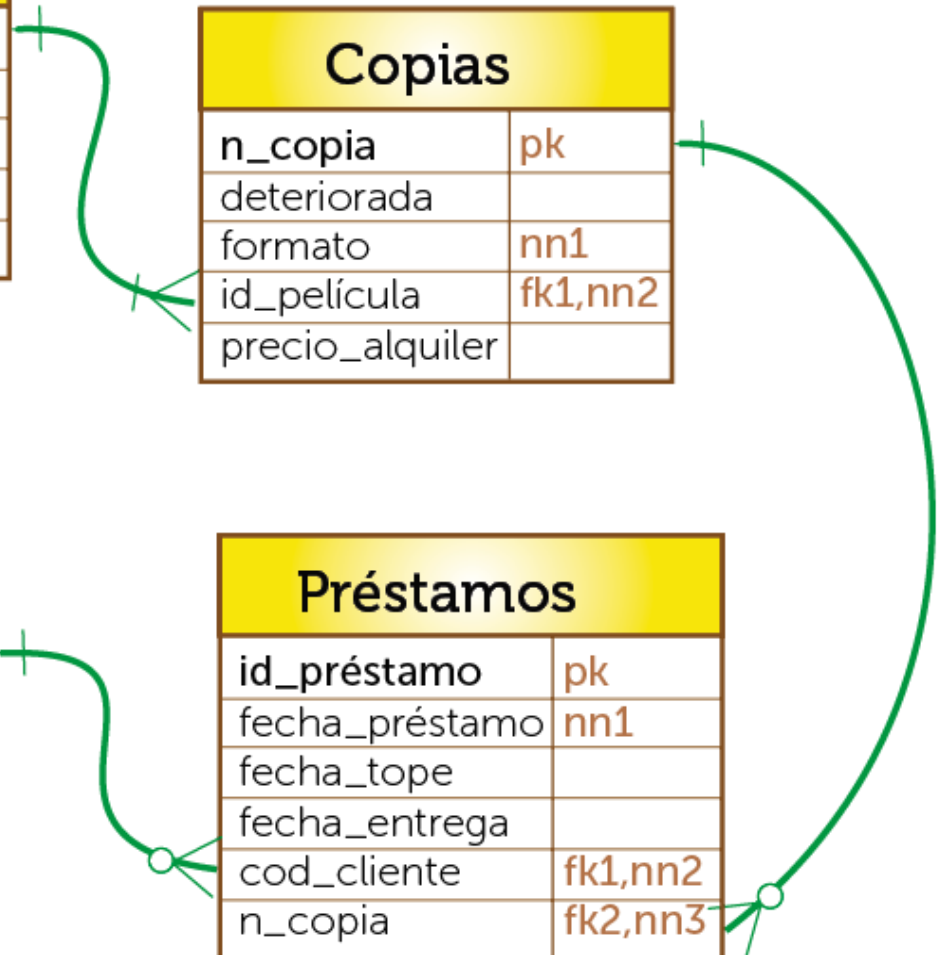
|                 |         |
|-----------------|---------|
| n_copia         | pk      |
| deteriorada     |         |
| formato         | nn1     |
| id_película     | fk1,nn2 |
| precio_alquiler |         |

## Clientes

|             |         |
|-------------|---------|
| cod_cliente | pk      |
| dni         | uk1,nn1 |
| nombre      | nn2     |
| apellido1   | nn3     |
| apellido2   | nn4     |
| dirección   |         |
| email       | uk2     |

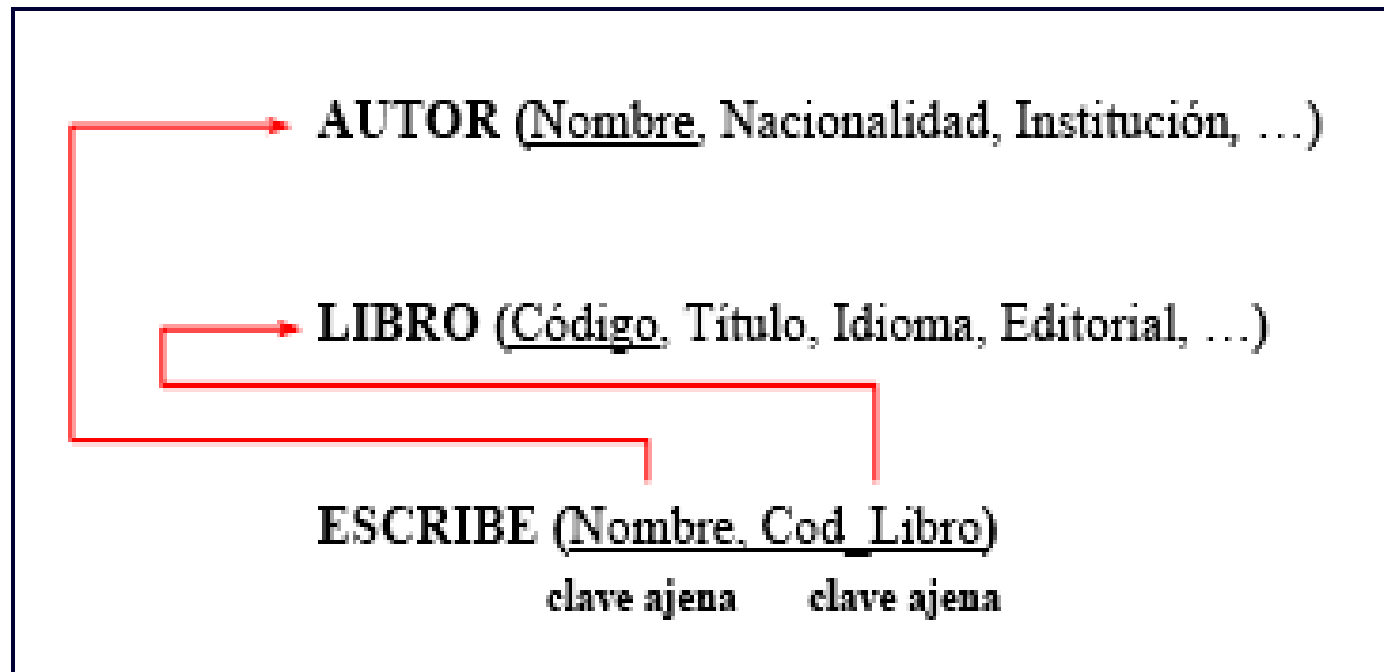
## Préstamos

|                |         |
|----------------|---------|
| id_préstamo    | pk      |
| fecha_préstamo | nn1     |
| fecha_tope     |         |
| fecha_entrega  |         |
| cod_cliente    | fk1,nn2 |
| n_copia        | fk2,nn3 |





# Claves AJENAS

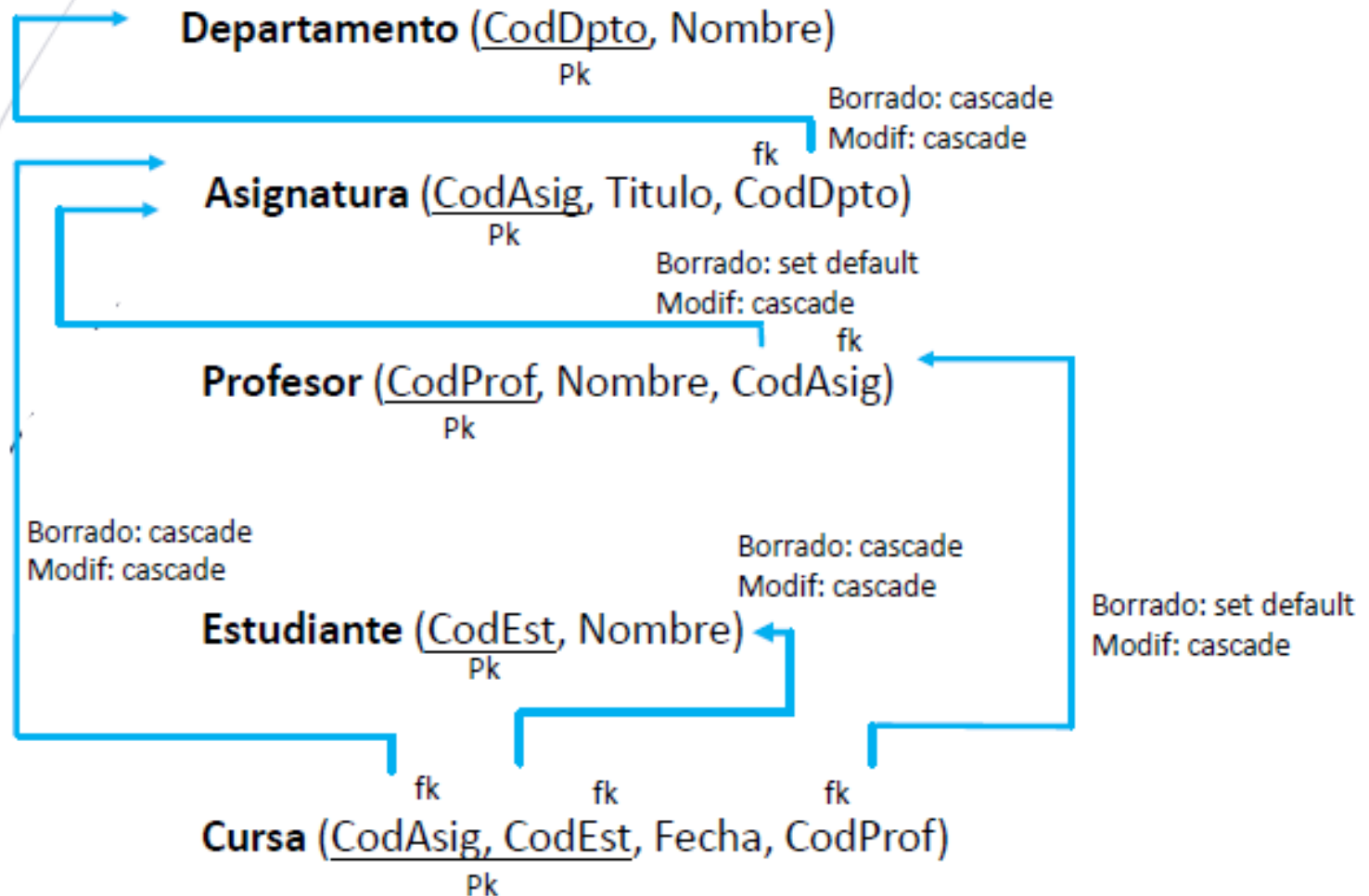


*Los atributos principales (forman la clave primaria) se subrayan*  
*Las claves ajenas se indican con una flecha*

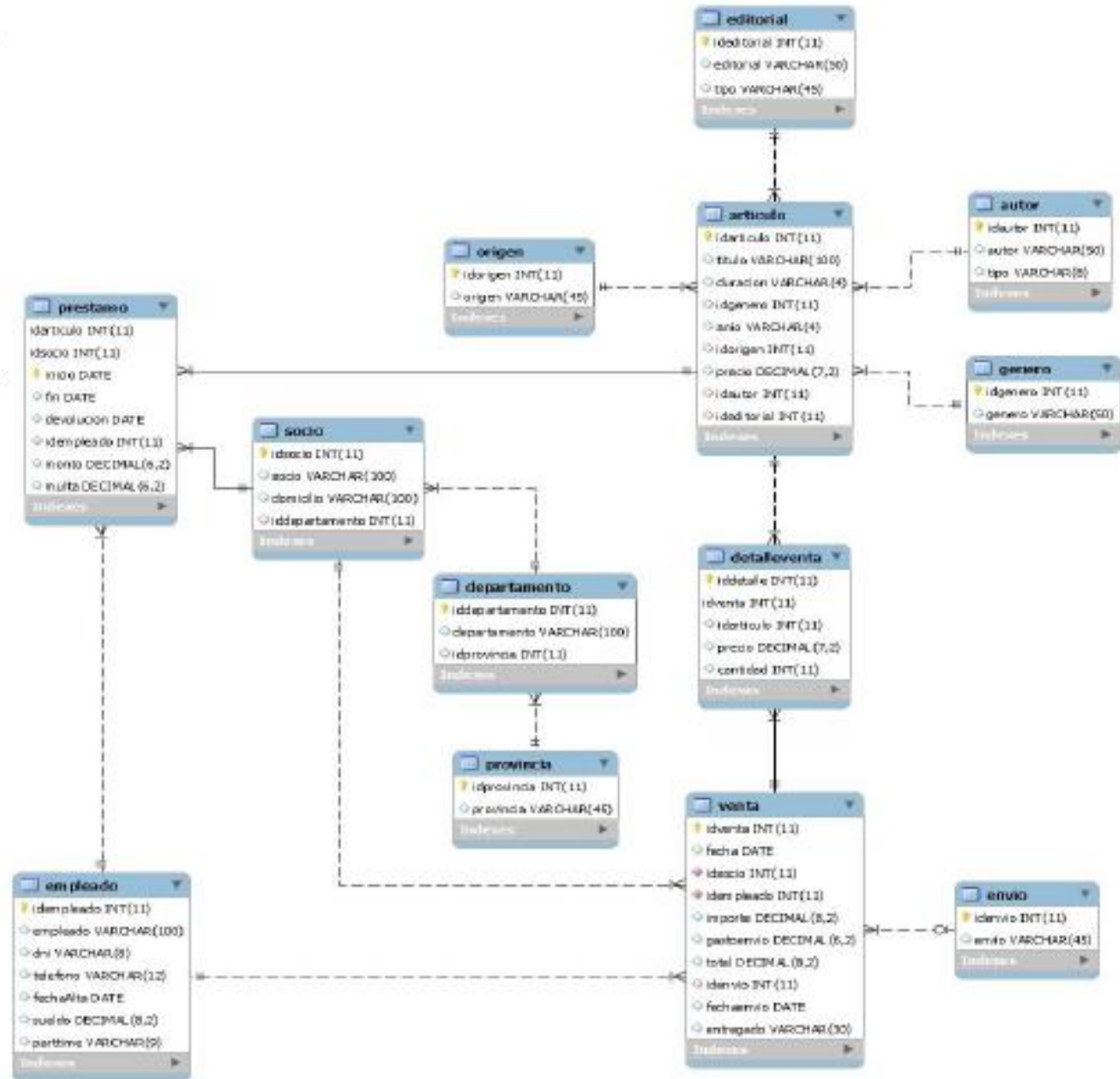
# Esquema de una BD Relacional

- Una base de datos relacional es un conjunto de relaciones normalizadas;
- Un esquema de base de datos relacional  $S$  es un conjunto de esquemas de relaciones  $S = R_1, R_2, \dots, R_n$  y un conjunto de **restricciones de integridad**.
- *Para representar el esquema de una base de datos relacional se debe dar el nombre de sus relaciones, los atributos de éstas, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajenas.*

# Notación Modelo Relacional



# Notación Modelo relacional (2)



# Restricciones al Modelo

## *Reglas de integridad*

Una vez definida la estructura de datos del modelo relacional, pasamos a estudiar las reglas de integridad que los datos almacenados en dicha estructura deben cumplir para garantizar que son correctos.

- Las **RESTRICCIONES INHERENTES** vienen impuestas por el propio Modelo de Datos.
  1. Intrínsecas
  2. Integridad de entidades
- Las **RESTRICCIONES SEMANTICAS.**

# Restricciones Inherentes Intrínsecas

En el caso del MR, una **TABLA** tiene unas propiedades intrínsecas y que se derivan de la misma **definición matemática de relación**, ya que, al ser un conjunto:



**No puede haber dos tuplas iguales**  
**=> obligatoriedad de la clave primaria**

El orden de las tuplas no es significativo.  
El orden de los atrib. no es significativo

Cada atributo sólo puede tomar un **único valor del dominio subyacente**; no se admiten grupos repetitivos (ni otro tipo de estructuras) como valores de los atributos de una tupla

# Restricciones Inherentes de Integridad de Entidades

- "Ningún atributo que forme parte de la **clave primaria de una Tabla** puede tomar un valor nulo"; es decir, un valor desconocido o inexistente.

**Nulos** → Cuando en una tupla un atributo es desconocido, se dice que es **nulo**.

Un **nulo** no representa el valor cero ni la cadena vacía, éstos son valores que tienen significado.

El **nulo** implica **ausencia de información**, bien porque al insertar la tupla se desconocía el valor del atributo, o bien porque para dicha tupla el atributo no tiene sentido

# Restricciones Semánticas

- **Son definidas por el usuario.**
- El modelo ofrece a los usuarios/diseñadores facilidades para que puedan reflejar en el esquema, lo más fielmente posible, la semántica del mundo real.
- Los tipos de restricciones semánticas permitidos en el MR son:

**Clave Primaria (PRIMARY KEY),  
Unicidad (UNIQUE),  
Obligatoriedad (NOT NULL),  
Integridad Referencial (FOREIGN KEY),  
Restricciones de Rechazo:  
Verificación (CHECK), y  
Aserción (CREATE ASSERTION).**



# Restricciones Semánticas

## 1. (PRIMARY KEY):

Permite declarar un atributo o un conjunto de atributos como **clave primaria** de una Tabla.

- => sus valores no se podrán repetir ni se admitirán los nulos (o valores "ausentes").
- Debemos distinguir entre la restricción inherente de obligatoriedad de la clave primaria y la restricción semántica que le permite al usuario indicar qué atributos forman parte de la clave primaria.

## 2. (UNIQUE):

- Los valores de un conjunto de atributos (uno o más) no pueden repetirse en una Tabla.
- Esta restricción permite la definición de **claves candidatas (alternativas)**.

## Restricciones Semánticas (2)

### 3. Obligatoriedad (NOT NULL):

El conjunto de atributos no admite valores nulos.

### 4. Integridad Referencial (FOREING KEY):

Si una relación R2 (relación que referencia) tiene un descriptor (subconjunto de atributos) CA que referencia a una clave candidata CC de la Tabla R1 (tabla referenciada), todo valor de dicho descriptor CA debe coincidir con un valor de CC o ser nulo.

- La condición puede expresarse como  $R2.CA = R1.CC$
- El descriptor CA es, por tanto, una **clave ajena** de la relación R2.
- Las relaciones R1 y R2 no son necesariamente distintas.
- La clave ajena puede ser también parte (o la totalidad) de la clave primaria de R2.
- CA puede admitir nulos o tener restricción de obligatoriedad (NOT NULL).

# Integridad Referencial

## (Modos de Borrado y Modificación)

Además de definir las **claves ajenas**, hay que determinar las **consecuencias que pueden tener ciertas operaciones** (borrado y modificación) realizadas sobre tuplas de la tabla referenciada:



**NO ACTION:** rechazar la operación de borrado o modificación. (Restringir)



**CASCADE:** Propagar la modificación (o borrado) de las tuplas de la tabla que referencia.

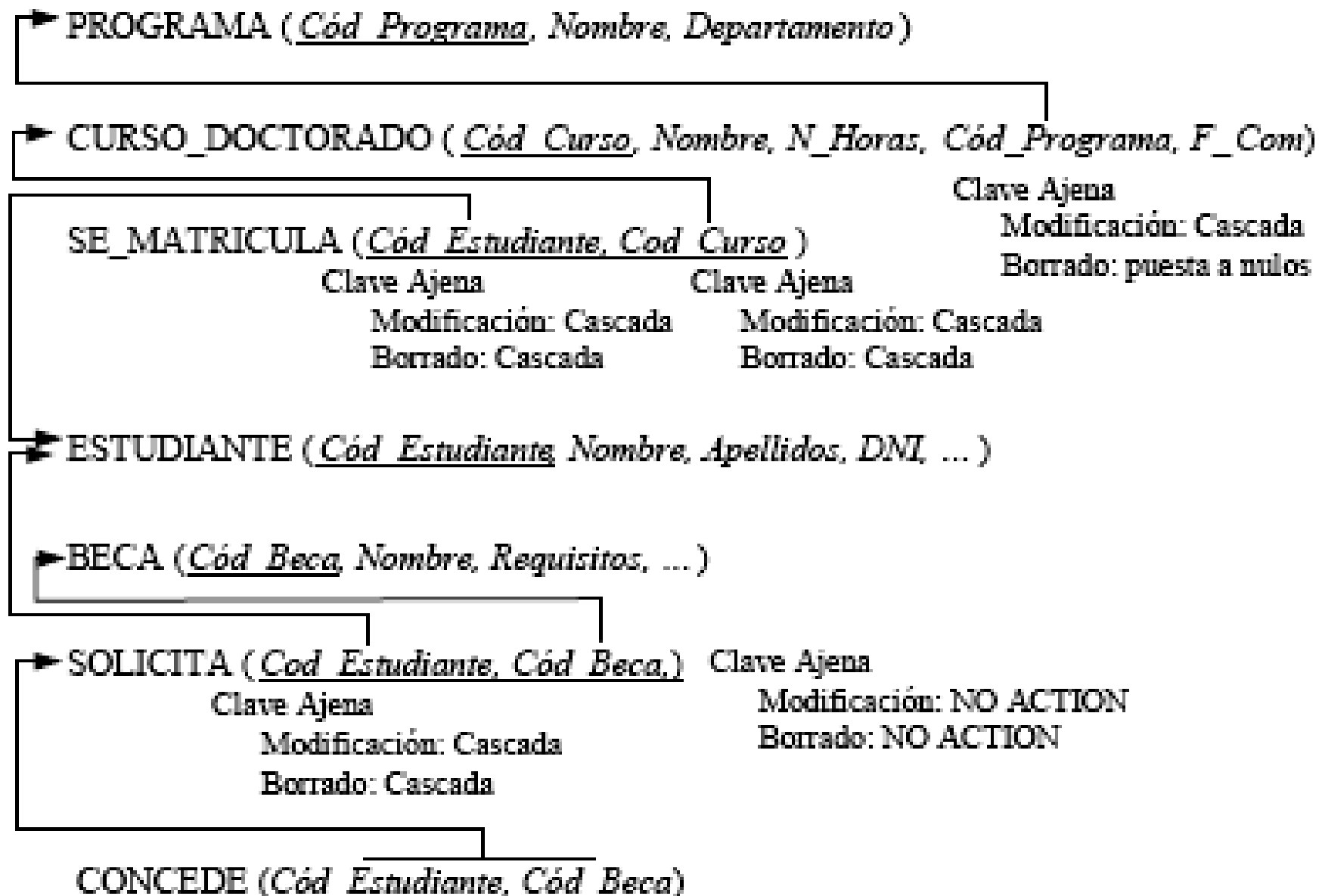


**SET NULL:** poner valor nulo en la clave ajena de la tabla que referencia.



**SET DEFAULT:** poner un valor por defecto en la clave ajena de la tabla que referencia.

## Semánticas – ejemplos de integridad referencial



## 5. Restricciones Semánticas de Rechazo

El usuario formula una **condición** mediante un predicado definido sobre un conjunto de atributos, tuplas o dominios, **que debe ser verificado** en **toda operación de actualización** para que el nuevo estado constituya una **ocurrencia válida** del esquema.

➤ Verificación (CHECK):

➤ Aserción (ASSERTION)

## 5. Restricciones Semánticas de Rechazo

### ► Verificación (CHECK):

Comprueba, en toda operación de actualización, si el predicado es **cierto o falso** y, en el segundo caso, rechaza la operación.

La restricción de verificación se define sobre **un único elemento** (dentro de un CREATE TABLE) y **puede o no** tener nombre.

**CHECK N\_HORAS > 30 en CURSO\_DOCTORADO**

## 5. Restricciones Semánticas de Rechazo

### ► Aserción (ASSERTION):

Actúa de forma idéntica a la anterior, pero se diferencia de ella en que puede afectar a varios elementos (por ejemplo, a dos tablas distintas).

Por tanto, su definición **no va unida a la de un determinado elemento** del esquema y **siempre** ha de tener un nombre.

```
CREATE ASSERTION CONCEDE_SOLICITA AS
```

```
CHECK (SELECT Cod_Estudiante, Cod_Beca  
FROM CONCEDE) IN
```

```
(SELECT Cod_Estudiante, Cod_Beca FROM  
SOLICITA));
```

# Reglas de Negocio

- Además de las dos reglas de integridad anteriores, los usuarios o los administradores de la base de datos pueden imponer ciertas restricciones específicas sobre los datos, denominadas *reglas de negocio*.
- *Por ejemplo, si en una oficina de la empresa inmobiliaria sólo puede haber hasta veinte empleados, el DBMS debe dar la posibilidad al usuario de definir una regla al respecto y debe hacerla respetar. En este caso, no debería permitir dar de alta un empleado en una oficina que ya tiene los veinte permitidos.*
- **Actualmente todo DBMS relacional permite definir este tipo de restricciones.**





# Muchas gracias

ltorresauad@herrera.unt.edu.ar