

# Diseño lógico - Modelo relacional

Luis Valencia Cabrera (lvalencia@us.es)

Research Group on Natural Computing  
Departamento de Ciencias de la Computación e Inteligencia Artificial  
Universidad de Sevilla

24-09-2025, Bases de Datos

# Índice

- 1 Introducción
- 2 Propiedades
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional
- 6 Bibliografía

# Índice

- 1 **Introducción**
- 2 Propiedades
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional
- 6 Bibliografía

# Introducción al Modelo Relacional

- Fue introducido por **Edgar Frank Codd** en **1970**, aunque los *primeros SGBD relacionales* no aparecieron hasta *los 80*.
- Supuso una **revolución en el diseño lógico de BD**, dando lugar a la *segunda generación de SBGD*.
- Es el **modelo lógico más extendido** (ORACLE, SQL Server, PostgreSQL, MySQL, Access, dBaseIV, Base...)

# Introducción al Modelo Relacional

- Los **datos** se **estructuran lógicamente** en forma de **relaciones** (tablas).

# Introducción al Modelo Relacional

- Los **datos** se **estructuran lógicamente** en forma de **relaciones** (tablas).
- Nota: mientras en los Diagramas E/R hablamos de entidades e interrelaciones, en el modelo relacional solamente tenemos un tipo de estructura, **todo son relaciones**.

# Introducción al Modelo Relacional

- Los **datos** se **estructuran lógicamente** en forma de **relaciones** (tablas).
- Nota: mientras en los Diagramas E/R hablamos de entidades e interrelaciones, en el modelo relacional solamente tenemos un tipo de estructura, **todo son relaciones**.
- Intuitivamente, una Base de datos relacional es un **conjunto de tablas bidimensionales enlazadas** entre sí.

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información**

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información** relacionada con el sistema objetivo *a partir del esquema conceptual*,

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información** relacionada con el sistema objetivo *a partir del esquema conceptual*, en base a un **modelo lógico** de base de datos

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información** relacionada con el sistema objetivo *a partir del esquema conceptual*, en base a un **modelo lógico** de base de datos e **independiente del SGBD concreto** que se vaya a utilizar.

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información** relacionada con el sistema objetivo *a partir del esquema conceptual*, en base a un **modelo lógico** de base de datos e **independiente del SGBD concreto** que se vaya a utilizar.

- El diseño conceptual trataba de obtener un esquema conceptual completo y expresivo.

# Diseño Lógico

## Diseño Lógico

Proceso de **construcción de un esquema de la información** relacionada con el sistema objetivo *a partir del esquema conceptual*, en base a un **modelo lógico** de base de datos e **independiente del SGBD concreto** que se vaya a utilizar.

- El diseño conceptual trataba de obtener un esquema conceptual completo y expresivo.
- El diseño lógico perseguirá obtener una representación que estructure lo anterior con **estructuras lógicas** que capturen *eficientemente* los datos y sus restricciones.

# Diseño Lógico

- Tanto el diseño conceptual como el diseño lógico son **procesos iterativos**, se van refinando continuamente a partir de la idea inicial.

# Diseño Lógico

- Tanto el diseño conceptual como el diseño lógico son **procesos iterativos**, se van refinando continuamente a partir de la idea inicial.
- Constituyen un proceso de **aprendizaje**

# Diseño Lógico

- Tanto el diseño conceptual como el diseño lógico son **procesos iterativos**, se van refinando continuamente a partir de la idea inicial.
- Constituyen un proceso de **aprendizaje** en el que el diseñador va comprendiendo las *necesidades* y el *significado* de los datos que maneja.

# Diseño Lógico

- Tanto el diseño conceptual como el diseño lógico son **procesos iterativos**, se van refinando continuamente a partir de la idea inicial.
- Constituyen un proceso de **aprendizaje** en el que el diseñador va comprendiendo las *necesidades* y el *significado* de los datos que maneja.
- Si la base de datos no es una **representación fiel** de la necesidad a cubrir, será difícil definir todas las vistas de los usuarios o mantener la *integridad* de la misma.

# Diseño Lógico

- Tanto el diseño conceptual como el diseño lógico son **procesos iterativos**, se van refinando continuamente a partir de la idea inicial.
- Constituyen un proceso de **aprendizaje** en el que el diseñador va comprendiendo las *necesidades* y el *significado* de los datos que maneja.
- Si la base de datos no es una **representación fiel** de la necesidad a cubrir, será difícil definir todas las vistas de los usuarios o mantener la *integridad* de la misma.
- **Objetivo** de esta etapa: obtener el **esquema lógico**, que estará formado por el **conjunto de relaciones** de la base de datos a partir de la especificación realizada en la etapa del diseño conceptual.

# El diseño lógico basado en modelo relacional

## Conceptos fundamentales

- **Relación**: tabla bidimensional, a nivel lógico
- **Registro** o tupla: fila de la tabla
- **Campo**: columna de la tabla

# El diseño lógico basado en modelo relacional

## Conceptos fundamentales

- **Relación:** tabla bidimensional, a nivel lógico
- **Registro** o tupla: fila de la tabla
- **Campo:** columna de la tabla

Ejemplo: relación ESCRITOR (2 registros, 4 campos)

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
56123009	Luis Gómez	Feria,2	5/5/1961

Las relaciones se *enlazan mediante campos con contenido común*.

# Modelo relacional

## Definiciones básicas

Una relación de grado  $m$  consta de dos partes:

- **Cabecera:** conjunto fijo de  $m$  campos.
  - Cada campo esta definido por su **nombre** y su **dominio** (que indica el tipo de valores para dicho campo).
  - $\{(Nombre_1 : Dominio_1), \dots, (Nombre_m : Dominio_m)\}$
- **Cuerpo:** conjunto variable de registros.
  - Cada registro es un conjunto de  $m$  valores:
    - $R_1 \rightarrow \{(Nombre_1 : Val_{1,1}), \dots, (Nombre_m : Val_{1,m})\}$
    - ...
    - $R_n \rightarrow \{(Nombre_1 : Val_{n,1}), \dots, (Nombre_m : Val_{n,m})\}$

# Modelo relacional

## Definiciones básicas - Notas

- Cada relación tiene asociado un **nombre** que la identifica.
- Una relación de **grado**  $m$  puede representarse mediante una tabla bidimensional de  $m$  columnas y tantas filas como registros aparezcan en la relación.
- Cada **valor** de un registro **debe pertenecer** al correspondiente **dominio** especificado en la **cabecera**.

# Modelo relacional

## Definiciones básicas - Ejemplo

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
56123009	Luis Gómez	Feria,2	5/5/1961

La cabecera de la relación ESCRITOR es:

- **{ (DNI:Numérico), (Nombre:Texto), Direccion:Texto), (Fecha:Fecha/Hora) }**

El cuerpo de la misma está formado por 2 registros:

- **{ (DNI:56123009), (Nombre:'Luis Gómez'), (Direccion:'Feria,2'), (Fecha:5/5/1961) }**
- **{ (Fecha:9/5/1960), (DNI:44345789), (Direccion:'Sol,17'), (Nombre:'Ana Pérez') }**

# Índice

- 1 Introducción
- 2 Propiedades**
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional
- 6 Bibliografía

# Cabecera

Cada relación tiene asociada, como vimos, una cabecera formada por un número fijo de campos.

- Notación: **NOMBRE1.Nombre2** denota el campo *Nombre2* de la cabecera de la relación **NOMBRE1**.
- Dos campos pertenecientes a la cabecera de la misma relación no pueden tener el mismo nombre.
- El orden de los campos en la cabecera de una relación no importa.

Campos de relaciones distintas sí pueden tener el mismo nombre:

- **ESCRITOR.DNI** denota el campo DNI de la relación ESCRITOR.
- **CLIENTE.DNI** denota el campo DNI de la relación CLIENTE.

# Cuerpo

- Todos los registros del cuerpo de una relación deben tener el *mismo número de campos*, aunque *alguno* este vacío. En este caso, dicho campo vacío toma el valor NULL.
- Los **valores de los campos son atómicos**: fijado un registro, cada campo toma un único valor (no se admiten campos multivaluados).
- **No** se admiten registros **duplicados**. Dos registros de una relación deben diferir, al menos, en el valor de un campo.
- El orden de los registros en el cuerpo de una relación no importa.

# Cuerpo - Ejemplos

Observe el contenido de estas dos relaciones:

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
40876100	José Ruíz	Luna,1	1/1/1972
56123009	Luis Gómez	Feria,2	NULL

Dirección	Nombre	Fecha	DNI
Feria,2	Luis Gómez	NULL	56123009
Luna,1	José Ruíz	1/1/1972	40876100
Sol,17	Ana Pérez	9/5/1960	44345789

¿Qué le sugieren?

## Cuerpo - Ejemplos

Observe el contenido de estas dos relaciones:

DNI	Nombre	Dirección	Fecha
44345789	Ana Pérez	Sol, 17	9/5/1960
40876100	José Ruíz	Luna,1	1/1/1972
56123009	Luis Gómez	Feria,2	NULL

Dirección	Nombre	Fecha	DNI
Feria,2	Luis Gómez	NULL	56123009
Luna,1	José Ruíz	1/1/1972	40876100
Sol,17	Ana Pérez	9/5/1960	44345789

¿Qué le sugieren? Como habrá supuesto, **en el modelo relacional ambas relaciones son idénticas**, pues solo difieren en el orden de campos y registros.

# Cuerpo - Ejemplos

¿Qué puede decir sobre este ejemplo?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

# Cuerpo - Ejemplos

¿Qué puede decir sobre este ejemplo?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

¿Puede ser una relación según el modelo relacional? ¿por qué?

## Cuerpo - Ejemplos

¿Qué puede decir sobre este ejemplo?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

¿Puede ser una relación según el modelo relacional? ¿por qué?  
Como habrá supuesto, la relación anterior **NO es válida en el modelo relacional**, ya que...

## Cuerpo - Ejemplos

¿Qué puede decir sobre este ejemplo?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

¿Puede ser una relación según el modelo relacional? ¿por qué?  
Como habrá supuesto, la relación anterior **NO es válida en el modelo relacional**, ya que...

- posee campos multivaluados

## Cuerpo - Ejemplos

¿Qué puede decir sobre este ejemplo?

Nombre y Apellido	Edad	Estudios
Juan Pérez	41	Lcdo. Química
Ana Sánchez	37	Lcdo. Medicina Lcdo. Física
Juan Pérez	41	Lcdo. Química
Félix González	32	NULL

¿Puede ser una relación según el modelo relacional? ¿por qué?  
Como habrá supuesto, la relación anterior **NO es válida en el modelo relacional**, ya que...

- posee campos multivaluados
- posee registros repetidos

# Campos de la relación

Cada campo debe poseer un **nombre** (relacionado con los datos que contendrá) y debe tener asociado un **tipo** de dato. Algunos tipos posibles (no los únicos) serían:

- Texto: cadenas de caracteres, ya sean letras, números con los que no realizar operaciones o símbolos.
- Numérico: números sobre los que tiene sentido realizar operaciones.
- Fecha/hora: almacena fechas, horas o ambas.
- Sí/No: datos que solo tengan dos posibilidades (verdadero-falso).
- Autonumérico: valor numérico (1,2,...) que el SGBD incrementa de modo automático cuando se añade un registro.

# Campos de la relación

Un campo puede poseer **opcionalmente** las siguientes propiedades:

- **Descripción:** texto breve que aclara el contenido o la finalidad del campo.
- **Tamaño:** indica el tamaño máximo permitido (aplicable a campos de texto o numéricos).
- **Rango** de valores posibles, dentro de una lista de valores permitidos.
- **Requerido** o **NOT NULL:** no se permiten valores nulos para dicho campo.
- **Predeterminado:** se fija un valor por defecto para el campo.

## Campos de la relación - Ejemplo

La cabecera de una relación gráficamente presentada así:

NIF	Nombre	Fecha	Nacionalidad	Dirección
-----	--------	-------	--------------	-----------

se podría describir como:

```
{
  (NIF:Texto(9),NOT NULL),
  (Nombre:Texto(50),NOT NULL,
   Descripción="Nombre y apellidos del cliente"),
  (Fecha:Fecha,
   Descripción= "Fecha de nacimiento del cliente"),
  (Nacionalidad:Texto(20),Predeterminado="Española"),
  (Direccion:Texto)
}
```

# Claves y datos derivados

Junto con la cabecera y los campos de las relaciones, se necesita indicar:

- La **clave primaria**, y si se ha de generar automáticamente.
- Las posibles claves alternativas.
- Las **claves ajenas** (se definirán en una diapositiva posterior) y sus *reglas de comportamiento* ante el borrado y la modificación de la clave primaria a la que referencian.
- Si alguna columna es un dato derivado (su valor se calcula a partir de otros datos de la base de datos), indicar cómo se obtiene su valor.

# Restricciones

Además de lo anterior, en nuestra fase de diseño lógico basado en modelo relacional, hay que completar el esquema lógico<sup>1</sup>:

---

<sup>1</sup>Por cuestiones de tiempo, en la asignatura veremos esencialmente el primero de estos ítems

# Restricciones

Además de lo anterior, en nuestra fase de diseño lógico basado en modelo relacional, hay que completar el esquema lógico<sup>1</sup>:

- Especificando **restricciones a nivel de fila** de cada tabla, si las hay. Estas restricciones son aquellas que *involucran a una o varias columnas dentro de una misma fila*. Por ejemplo: fecha de fin mayor o igual a la de inicio.

---

<sup>1</sup>Por cuestiones de tiempo, en la asignatura veremos esencialmente el primero de estos ítems

# Restricciones

Además de lo anterior, en nuestra fase de diseño lógico basado en modelo relacional, hay que completar el esquema lógico<sup>1</sup>:

- Especificando **restricciones a nivel de fila** de cada tabla, si las hay. Estas restricciones son aquellas que *involucran a una o varias columnas dentro de una misma fila*. Por ejemplo: fecha de fin mayor o igual a la de inicio.
- Especificando otras restricciones no expresadas antes (aquellas que *involucran a varias filas de una misma tabla o a filas de varias tablas a la vez*).

---

<sup>1</sup>Por cuestiones de tiempo, en la asignatura veremos esencialmente el primero de estos ítems

# Restricciones

Además de lo anterior, en nuestra fase de diseño lógico basado en modelo relacional, hay que completar el esquema lógico<sup>1</sup>:

- Especificando **restricciones a nivel de fila** de cada tabla, si las hay. Estas restricciones son aquellas que *involucran a una o varias columnas dentro de una misma fila*. Por ejemplo: fecha de fin mayor o igual a la de inicio.
- Especificando otras restricciones no expresadas antes (aquellas que *involucran a varias filas de una misma tabla o a filas de varias tablas a la vez*).
- Reflejando las reglas de negocio (si las hay), que serán aquellas *acciones* que se deba llevar a cabo de forma *automática* como *consecuencia de actualizaciones* que se realicen sobre la base de datos.

<sup>1</sup>Por cuestiones de tiempo, en la asignatura veremos esencialmente el primero de estos ítems

# Índice

- 1 Introducción
- 2 Propiedades
- 3 Claves primarias y ajenas**
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional
- 6 Bibliografía

# Clave primaria de una relación

- Clave: *conjunto de campos* cuyos valores **determinan unívocamente a cada registro** de la relación. Dicho conjunto de campos debe ser **minimal**, *i.e.*, ningún subconjunto propio de la clave puede actuar también como clave.
- Clave candidata: cada uno de los campos o combinaciones de campos que pueden actuar como clave de la relación.
- **Clave primaria** (PK=Primary Key): clave candidata elegida por el diseñador de la BD para la relación. El resto de candidatas se denominan claves alternativas.

Nota: En el modelo relacional, *toda relación posee clave primaria*.

# Clave primaria de una relación

## Ejemplos

- Relación ALUMNO:

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

# Clave primaria de una relación

## Ejemplos

- Relación ALUMNO:

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

Claves candidatas:

- (Alumno.NIF)
- (Alumno.Codigo)

# Clave primaria de una relación

## Ejemplos

- Relación ALUMNO:

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

Claves candidatas:

- (Alumno.NIF)
  - (Alumno.Codigo)
- Relación CURSAR:

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

# Clave primaria de una relación

## Ejemplos

- Relación ALUMNO:

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

Claves candidatas:

- (Alumno.NIF)
  - (Alumno.Codigo)
- Relación CURSAR:

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

Claves candidatas:

- (Cursar.NIF, Cursar.Asignatura, Cursar.Año)

# Claves ajenas de una relación

**Clave ajena** o secundaria (FK=Foreign Key): *campo o combinación de campos de una relación* (relación hija) *que funciona como clave primaria de otra relación* de la BD (relación referenciada o relación padre para la clave ajena).

# Claves ajenas de una relación

**Clave ajena** o secundaria (FK=Foreign Key): *campo o combinación de campos de una relación* (relación hija) *que funciona como clave primaria de otra relación* de la BD (relación referenciada o relación padre para la clave ajena).

Ejemplo:

- ALUMNO (PK Alumno.NIF):

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

- CURSAR (PK (Cursar.NIF, Cursar.Asignatura, Cursar.Año)):

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

# Claves ajenas de una relación

**Clave ajena** o secundaria (FK=Foreign Key): **campo o combinación de campos de una relación** (relación hija) **que funciona como clave primaria de otra relación** de la BD (relación referenciada o relación padre para la clave ajena).

Ejemplo:

- ALUMNO (PK Alumno.NIF):

Nombre	NIF	Código	Fecha	Dirección
--------	-----	--------	-------	-----------

- CURSAR (PK (Cursar.NIF, Cursar.Asignatura, Cursar.Año)):

NIF	Asignatura	Año	Repetidor
-----	------------	-----	-----------

El campo **Cursar.NIF** es una clave ajena de la relación CURSAR y enlaza con la relación ALUMNO, *apuntando* al campo **Alumno.NIF**.

# Claves ajenas de una relación

## Ejemplos

- Relación EDITORIAL:

Nombre	Dirección	Ciudad	País
LaÑ	Sol,5	Sevilla	España

- Relación ESCRITOR:

Nombre	DNI	Nacionalidad
Ana Ruíz	56234111	Chilena

- Relación LIBRO:

Código	Título	Autor	Nombre-Ed
1256AB	Volver	56234111	LaÑ

- Claves primarias:

# Claves ajenas de una relación

## Ejemplos

- Relación EDITORIAL:

Nombre	Dirección	Ciudad	País
LaÑ	Sol,5	Sevilla	España

- Relación ESCRITOR:

Nombre	DNI	Nacionalidad
Ana Ruíz	56234111	Chilena

- Relación LIBRO:

Código	Título	Autor	Nombre-Ed
1256AB	Volver	56234111	LaÑ

- Claves primarias:

- Relación EDITORIAL: PK = (Nombre:Texto)
- Relación ESCRITOR: PK = (DNI:Texto)
- Relación LIBRO: PK = (Codigo:Texto)

- Claves ajenas en LIBRO:

# Claves ajenas de una relación

## Ejemplos

- Relación EDITORIAL:

Nombre	Dirección	Ciudad	País
LaÑ	Sol,5	Sevilla	España

- Relación ESCRITOR:

Nombre	DNI	Nacionalidad
Ana Ruíz	56234111	Chilena

- Relación LIBRO:

Código	Título	Autor	Nombre-Ed
1256AB	Volver	56234111	LaÑ

- Claves primarias:

- Relación EDITORIAL: PK = (Nombre:Texto)
- Relación ESCRITOR: PK = (DNI:Texto)
- Relación LIBRO: PK = (Codigo:Texto)

- Claves ajenas en LIBRO:

- FK = (Nombre-Ed:Texto) (→ Editorial)
- FK = (Autor:Texto) (→ Escritor)

# Claves ajenas de una relación

## Propiedades

- Las **claves ajenas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.

# Claves ajenas de una relación

## Propiedades

- Las **claves ajenas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.
- Una clave ajena y la clave primaria de la relación referenciada asociada han de estar **definidas** sobre los **mismos dominios**.

# Claves ajenas de una relación

## Propiedades

- Las **claves ajenas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.
- Una clave ajena y la clave primaria de la relación referenciada asociada han de estar **definidas** sobre los **mismos dominios**.
- *Una relación puede poseer más de una clave ajena* (tendrá una clave ajena por cada relación referenciada de la cual dependa).

# Claves ajenas de una relación

## Propiedades

- Las **claves ajenas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.
- Una clave ajena y la clave primaria de la relación referenciada asociada han de estar **definidas** sobre los **mismos dominios**.
- *Una relación puede poseer más de una clave ajena* (tendrá una clave ajena por cada relación referenciada de la cual dependa).
- *Una relación puede no poseer ninguna clave ajena.*

# Claves ajenas de una relación

## Propiedades

- Las **claves ajenas** son **esenciales** en el Modelo Relacional, ya que permiten **enlazar las relaciones** de la BD.
- Una clave ajena y la clave primaria de la relación referenciada asociada han de estar **definidas** sobre los **mismos dominios**.
- *Una relación puede poseer más de una clave ajena* (tendrá una clave ajena por cada relación referenciada de la cual dependa).
- *Una relación puede no poseer ninguna clave ajena.*
- *Una clave ajena puede enlazar una relación consigo misma* (relaciones reflexivas).

# Índice

- 1 Introducción
- 2 Propiedades
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional**
- 5 Paso del DER al esquema relacional
- 6 Bibliografía

# Valores nulos en el Modelo Relacional

## Valor nulo (NULL)

Marca para representar *información desconocida o no aplicable*.

El valor de un campo puede ser nulo por dos razones:

# Valores nulos en el Modelo Relacional

## Valor nulo (NULL)

Marca para representar ***información desconocida o no aplicable***.

El valor de un campo puede ser nulo por dos razones:

- Existencia de registros con campos desconocidos en ese momento.

# Valores nulos en el Modelo Relacional

## Valor nulo (NULL)

Marca para representar ***información desconocida o no aplicable***.

El valor de un campo puede ser nulo por dos razones:

- Existencia de registros con campos desconocidos en ese momento.
- Existencia de campos inaplicables a ciertos registros.

# Valores nulos en el Modelo Relacional

## Valor nulo (NULL)

Marca para representar **información desconocida o no aplicable**.

El valor de un campo puede ser nulo por dos razones:

- Existencia de registros con campos desconocidos en ese momento.
- Existencia de campos inaplicables a ciertos registros.

Ejemplo: relación OBRA

Código	Título	Tipo	Editorial	Año
123A	La huida	Libro	LaÑ	2002
678V	El infinito	Libro	NULL	NULL
564B	Azul	Cuadro	NULL	1975

# Valores nulos en el Modelo Relacional

## Valor nulo (NULL)

Marca para representar **información desconocida o no aplicable**.

El valor de un campo puede ser nulo por dos razones:

- Existencia de registros con campos desconocidos en ese momento.
- Existencia de campos inaplicables a ciertos registros.

Ejemplo: relación OBRA

Código	Título	Tipo	Editorial	Año
123A	La huida	Libro	LaÑ	2002
678V	El infinito	Libro	NULL	NULL
564B	Azul	Cuadro	NULL	1975

Los valores nulos del registro '678V' lo son por información desconocida, mientras que el valor nulo del registro '564B' representa un campo no aplicable (un cuadro no posee editorial).

# Restricciones de integridad

## Integridad de entidad

Diremos que una relación cumple la **restricción de integridad de entidad** si *ningún campo que forme parte de la clave primaria de la relación puede tomar valores nulos*.

# Restricciones de integridad

## Integridad de entidad

Diremos que una relación cumple la **restricción de integridad de entidad** si *ningún campo que forme parte de la clave primaria de la relación puede tomar valores nulos*.

- Nota: Para conseguir la integridad de entidad, basta declarar como 'Requerido' (NOT NULL) todos los campos que formen parte de la PK de cada relación de la BD.

# Restricciones de integridad

## Integridad de entidad

Diremos que una relación cumple la **restricción de integridad de entidad** si *ningún campo que forme parte de la clave primaria de la relación puede tomar valores nulos*.

- Nota: Para conseguir la integridad de entidad, basta declarar como 'Requerido' (NOT NULL) todos los campos que formen parte de la PK de cada relación de la BD.
- Por convenio, fijamos que cualquier campo que forme parte de una PK posee la propiedad adicional 'Requerido'; así no será necesario declararlo explícitamente.

# Restricciones de integridad

## Integridad referencial

Si una relación R2 posee una clave ajena que la enlaza con la relación R1, entonces diremos que cumple la **restricción de integridad referencial** si todo valor de dicha clave ajena de R2 cumple una de las dos condiciones:

- (a) coincide con algún valor de la clave primaria en la relación R1
- (b) toma el valor nulo (NULL)

# Restricciones de integridad

## Integridad referencial - Ejemplo

Relación ESCRITOR,  
PK = (DNI:Texto)

DNI	Nombre	Fecha	País
67543198	Luis Ruíz	1/1/1965	Chile
89564123	Ana Pérez	2/7/1977	España

Relación OBRA,  
PK = (Código:Texto)

Código	Título	Autor	Fecha
345	La huida	67543198	1993
111	El fin	33567900	1982
654	NULL	NULL	2001

FK = (Autor:Texto) (→ ESCRITOR)

¿Algo reseñable en estas claves y el contenido de estas relaciones?

# Restricciones de integridad

## Integridad referencial - Ejemplo

Relación ESCRITOR,  
PK = (DNI:Texto)

DNI	Nombre	Fecha	País
67543198	Luis Ruíz	1/1/1965	Chile
89564123	Ana Pérez	2/7/1977	España

Relación OBRA,  
PK = (Código:Texto)

Código	Título	Autor	Fecha
345	La huida	67543198	1993
111	El fin	33567900	1982
654	NULL	NULL	2001

FK = (Autor:Texto) (→ ESCRITOR)

¿Algo reseñable en estas claves y el contenido de estas relaciones?

La BD anterior **NO cumple** la restricción de integridad referencial. El valor del campo Autor del segundo registro de la tabla Obra (33567900) NO se corresponde con ningún valor del campo DNI de la tabla Escritor.

# ¿Como mantener la integridad referencial? I

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

- **Inserción:** el SGBD solo permite insertar un nuevo registro en la relación R2 cuando el valor del campo C para ese registro coincida con algún valor de la PK de R1 que aparezca en la relación (o bien el campo C se deje nulo).

# ¿Como mantener la integridad referencial? I

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

- **Inserción:** el SGBD solo permite insertar un nuevo registro en la relación R2 cuando el valor del campo C para ese registro coincida con algún valor de la PK de R1 que aparezca en la relación (o bien el campo C se deje nulo).
- **Modificación:**
  - actualización en **cascada:** si modificamos el valor de la PK de un registro de la relación padre R1, el SGBD modifica automáticamente dicho valor en todos los registros de la relación R1 que estén relacionados con él.

# ¿Como mantener la integridad referencial? I

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

- **Inserción:** el SGBD solo permite insertar un nuevo registro en la relación R2 cuando el valor del campo C para ese registro coincida con algún valor de la PK de R1 que aparezca en la relación (o bien el campo C se deje nulo).
- **Modificación:**
  - actualización en **cascada:** si modificamos el valor de la PK de un registro de la relación padre R1, el SGBD modifica automáticamente dicho valor en todos los registros de la relación R1 que estén relacionados con él.
  - actualización **restringida:** si intentamos modificar el valor de la PK en un registro de R1 referenciado por otros en otra relación R2, el SGBD nos lo impedirá.

# ¿Como mantener la integridad referencial? II

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

- **Borrado:**

- eliminación en **cascada**: si eliminamos un registro de la relación padre R1, el SGBD elimina automáticamente todos los registros de la relación R2 que están relacionados con dicho registro.

# ¿Como mantener la integridad referencial? II

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

## ■ Borrado:

- eliminación en **cascada**: si eliminamos un registro de la relación padre R1, el SGBD elimina automáticamente todos los registros de la relación R2 que están relacionados con dicho registro.
- eliminación **restringida**: si intentamos eliminar un registro de la relación padre R1, el SGBD no nos deja si hay registros de la relación R2 que están relacionados con dicho registro.

# ¿Como mantener la integridad referencial? II

La relación R2 está enlazada con la relación padre R1 mediante una clave ajena C. Para mantener la integridad referencial...

## ■ Borrado:

- eliminación en **cascada**: si eliminamos un registro de la relación padre R1, el SGBD elimina automáticamente todos los registros de la relación R2 que están relacionados con dicho registro.
- eliminación **restringida**: si intentamos eliminar un registro de la relación padre R1, el SGBD no nos deja si hay registros de la relación R2 que están relacionados con dicho registro.

Nota: es habitual que, además de estas dos opciones, exista otra opción en los SGBD relacionales para poner a nulos los campos que actúen como claves ajenas en los registros que posean el valor de la PK que se está eliminando o modificando.

# Índice

- 1 Introducción
- 2 Propiedades
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional**
- 6 Bibliografía

# Paso del DER al Esquema Lógico Relacional

Existe un **procedimiento general** para obtener el Esquema Relacional (**conjunto de relaciones**, tablas relacionales) a partir del Diagrama Entidad/Relación.

Veremos cómo tratar:

- Entidades
- Interrelaciones
- Jerarquías de generalización/especialización
- ...

# Entidades fuertes

Por cada **entidad fuerte** del diagrama E-R, se creará *una relación en el esquema relacional*, con “tantos campos como atributos posea la entidad”.

# Entidades fuertes

Por cada **entidad fuerte** del diagrama E-R, se creará *una relación en el esquema relacional*, con “tantos campos como atributos posea la entidad”.

- La PK de la relación creada es la misma que la clave primaria de la entidad.

# Entidades fuertes

Por cada **entidad fuerte** del diagrama E-R, se creará *una relación en el esquema relacional*, con “tantos campos como atributos posea la entidad”.

- La PK de la relación creada es la misma que la clave primaria de la entidad.
- Ejemplo: la entidad fuerte ALUMNO(DNI,Nombre,Dirección,Fecha) genera la relación ALUMNO, definida por:

DNI	Nombre	Dirección	Fecha
-----	--------	-----------	-------

# Entidades fuertes

Por cada **entidad fuerte** del diagrama E-R, se creará *una relación en el esquema relacional*, con “tantos campos como atributos posea la entidad”.

- La PK de la relación creada es la misma que la clave primaria de la entidad.
- Ejemplo: la entidad fuerte ALUMNO(DNI,Nombre,Dirección,Fecha) genera la relación ALUMNO, definida por:

DNI	Nombre	Dirección	Fecha
-----	--------	-----------	-------

- PK = DNI

# Entidades débiles

Débiles en **existencia**:

- Se crea la relación como en las entidades fuertes.

# Entidades débiles

## Débiles en **existencia**:

- Se crea la relación como en las entidades fuertes.
- Se añaden a la relación tantos campos como tenga la PK de la relación de la que depende, de forma que actúen en la nueva relación como clave ajena:  
FK = campos FK (→ relación *padre* (PK ent. padre))

# Entidades débiles

## Débiles en **existencia**:

- Se crea la relación como en las entidades fuertes.
- Se añaden a la relación tantos campos como tenga la PK de la relación de la que depende, de forma que actúen en la nueva relación como clave ajena:  
FK = campos FK (→ relación *padre* (PK ent. padre))

## Débiles en **identificación**:

- Se creará una relación con un campo por cada atributo de la entidad, y **se añaden los campos que forman la PK de la entidad *padre* de la que depende**.
- PK = PK de la entidad *padre* + discriminador de la entidad débil

# Entidades débiles

## Débiles en **existencia**:

- Se crea la relación como en las entidades fuertes.
- Se añaden a la relación tantos campos como tenga la PK de la relación de la que depende, de forma que actúen en la nueva relación como clave ajena:  
FK = campos FK (→ relación *padre* (PK ent. padre))

## Débiles en **identificación**:

- Se creará una relación con un campo por cada atributo de la entidad, y **se añaden los campos que forman la PK de la entidad *padre* de la que depende**.
- PK = PK de la entidad *padre* + discriminador de la entidad débil
- Se añade una clave ajena a la relación, como en el caso anterior:  
FK = PK de la entidad *padre* (→ relación *padre*)

# Entidades débiles

## Ejemplo

Relación CUENTA(Código, Titular, Fecha, Saldo)

Relación OPERACIÓN(Número, Descripción, Cantidad)

La entidad fuerte CUENTA, con PK = Código, genera la tabla:

Código	Titular	Fecha	Saldo
--------	---------	-------	-------

# Entidades débiles

## Ejemplo

Relación CUENTA(Código, Titular, Fecha, Saldo)

Relación OPERACIÓN(Número, Descripción, Cantidad)

La entidad fuerte CUENTA, con PK = Código, genera la tabla:

Código	Titular	Fecha	Saldo
--------	---------	-------	-------

La entidad débil en identificación OPERACIÓN genera la relación:

Número	Código-cuenta	Descripción	Cantidad
--------	---------------	-------------	----------

- PK = (Código-Cuenta, Número)
- FK = (Código-Cuenta) (→ Cuenta)

# Interrelaciones

Supongamos una interrelación IR que asocia las entidades E1,E2. La relación se tratará de forma distinta según el tipo de la interrelación:

- Muchas a muchas (N:M)
- Una a muchas (1:N)
- Una a una (1:1)

Además, habrá que prestar atención especial al tratamiento de las relaciones reflexivas y las de grado superior.

# Muchas a muchas (N:M)

Creamos una nueva relación con los siguientes campos:

- los campos de la PK de la entidad E1,
- los campos de la PK de la entidad E2,
- los campos correspondientes a los atributos propios de la interrelación (si los hubiese).

---

<sup>2</sup>Ojo, si la vinculación entre dos instancias dada por la interrelación se puede dar más de una vez, requiriendo atributos más genéricos junto con otros específicos de cada repetición, podría implicar la creación de dos relaciones

# Muchas a muchas (N:M)

Creamos una nueva relación con los siguientes campos:

- los campos de la PK de la entidad E1,
- los campos de la PK de la entidad E2,
- los campos correspondientes a los atributos propios de la interrelación (si los hubiese).

Con respecto a las claves:

- La **PK** contiene *generalmente* a las claves de E1 y E2

---

<sup>2</sup>Ojo, si la vinculación entre dos instancias dada por la interrelación se puede dar más de una vez, requiriendo atributos más genéricos junto con otros específicos de cada repetición, podría implicar la creación de dos relaciones

# Muchas a muchas (N:M)

Creamos una nueva relación con los siguientes campos:

- los campos de la PK de la entidad E1,
- los campos de la PK de la entidad E2,
- los campos correspondientes a los atributos propios de la interrelación (si los hubiese).

Con respecto a las claves:

- La **PK** contiene *generalmente* a las claves de E1 y E2
- Esta **podrá incluir más campos o excluir alguno**, dependiendo de la *semántica*.<sup>2</sup>

---

<sup>2</sup>Ojo, si la vinculación entre dos instancias dada por la interrelación se puede dar más de una vez, requiriendo atributos más genéricos junto con otros específicos de cada repetición, podría implicar la creación de dos relaciones

## Muchas a muchas (N:M)

Creamos una nueva relación con los siguientes campos:

- los campos de la PK de la entidad E1,
- los campos de la PK de la entidad E2,
- los campos correspondientes a los atributos propios de la interrelación (si los hubiese).

Con respecto a las claves:

- La **PK** contiene *generalmente* a las claves de E1 y E2
- Esta **podrá incluir más campos o excluir alguno**, dependiendo de la *semántica*.<sup>2</sup>
- Se añaden dos claves ajenas a la nueva relación:
  - FK = PK de E1 (→ E1)
  - FK = PK de E2 (→ E2)

<sup>2</sup>Ojo, si la vinculación entre dos instancias dada por la interrelación se puede dar más de una vez, requiriendo atributos más genéricos junto con otros específicos de cada repetición, podría implicar la creación de dos relaciones

## Una a muchas (1:N)

- **NO** se creará ninguna **relación** nueva. En su lugar, modificaremos la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.

## Una a muchas (1:N)

- **NO** se creará ninguna **relación** nueva. En su lugar, modificaremos la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.
- Suponiendo que E2 es quien participa con cardinalidad  $(1, n)$  o  $(0, n)$ , modificaremos la relación asociada a la entidad E2 como sigue:

## Una a muchas (1:N)

- **NO** se creará ninguna **relación** nueva. En su lugar, modificaremos la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.
- Suponiendo que E2 es quien participa con cardinalidad  $(1, n)$  o  $(0, n)$ , modificaremos la relación asociada a la entidad E2 como sigue:
  - añadimos como campos nuevos los atributos que forman la PK de la entidad E1

## Una a muchas (1:N)

- **NO** se creará ninguna **relación** nueva. En su lugar, modificaremos la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.
- Suponiendo que E2 es quien participa con cardinalidad  $(1, n)$  o  $(0, n)$ , modificaremos la relación asociada a la entidad E2 como sigue:
  - añadimos como campos nuevos los atributos que forman la PK de la entidad E1
  - añadimos los campos correspondientes a los atributos propios de la relación (si los hubiese)

## Una a muchas (1:N)

- **NO** se creará ninguna **relación** nueva. En su lugar, modificaremos la relación asociada a la entidad que participa con cardinalidad máxima “muchas”.
- Suponiendo que E2 es quien participa con cardinalidad  $(1, n)$  o  $(0, n)$ , modificaremos la relación asociada a la entidad E2 como sigue:
  - añadimos como campos nuevos los atributos que forman la PK de la entidad E1
  - añadimos los campos correspondientes a los atributos propios de la relación (si los hubiese)
  - añadimos una nueva clave ajena:  
FK = PK de E1 ( $\rightarrow$  E1)

## Una a una (1:1)

**NO** se creará ninguna **relación** nueva. Se tratan como las relaciones 1:N en ese sentido. Ahora bien, puesto que las dos entidades participan con cardinalidad máxima 1, tenemos dos opciones:

## Una a una (1:1)

**NO** se creará ninguna **relación** nueva. Se tratan como las relaciones 1:N en ese sentido. Ahora bien, puesto que las dos entidades participan con cardinalidad máxima 1, tenemos dos opciones:

- añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación 1:1, o bien

# Una a una (1:1)

**NO** se creará ninguna **relación** nueva. Se tratan como las relaciones 1:N en ese sentido. Ahora bien, puesto que las dos entidades participan con cardinalidad máxima 1, tenemos dos opciones:

- añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación 1:1, o bien
- añadir a la tabla asociada a E1 la PK de E2 y los atributos propios de la relación 1:1.

# Una a una (1:1)

**NO** se creará ninguna **relación** nueva. Se tratan como las relaciones 1:N en ese sentido. Ahora bien, puesto que las dos entidades participan con cardinalidad máxima 1, tenemos dos opciones:

- añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación 1:1, o bien
- añadir a la tabla asociada a E1 la PK de E2 y los atributos propios de la relación 1:1.

Nota: si una entidad participa con cardinalidad (1,1) y la otra con (0,1), optaremos por modificar la relación correspondiente a la entidad que aparece en el lado (0,1)

# Una a una (1:1)

**NO** se creará ninguna **relación** nueva. Se tratan como las relaciones 1:N en ese sentido. Ahora bien, puesto que las dos entidades participan con cardinalidad máxima 1, tenemos dos opciones:

- añadir a la tabla asociada a E2 la PK de E1 y los atributos propios de la relación 1:1, o bien
- añadir a la tabla asociada a E1 la PK de E2 y los atributos propios de la relación 1:1.

Nota: si una entidad participa con cardinalidad (1,1) y la otra con (0,1), optaremos por modificar la relación correspondiente a la entidad que aparece en el lado (0,1) (ventaja: evitamos valores nulos, ya que todo registro de la entidad de este lado tendrá asociado siempre 1 registro del lado (1,1)).

# Interrelaciones reflexivas

Se tratarán de forma distinta según su tipo (N:M) o (1:N)

# Interrelaciones reflexivas

Se tratarán de forma distinta según su tipo (N:M) o (1:N)

- **Tipo (N:M):** se creará una nueva relación como vimos antes para N:M, pero la PK de la entidad que participa aparecerá dos veces (*con nombres distintos según el rol con el que participe en la relación reflexiva*).

# Interrelaciones reflexivas

Se tratarán de forma distinta según su tipo (N:M) o (1:N)

- **Tipo (N:M):** se creará una nueva relación como vimos antes para N:M, pero la PK de la entidad que participa aparecerá dos veces (*con nombres distintos según el rol con el que participe en la relación reflexiva*).
- **Tipo (1:N):** NO se creará una nueva relación, como vimos antes para 1:N. Ahora bien, en la tabla asociada a la entidad que participa en la relación reflexiva aparecerá dos veces su PK (con nombres distintos):

# Interrelaciones reflexivas

Se tratarán de forma distinta según su tipo (N:M) o (1:N)

- **Tipo (N:M):** se creará una nueva relación como vimos antes para N:M, pero la PK de la entidad que participa aparecerá dos veces (*con nombres distintos según el rol con el que participe en la relación reflexiva*).
- **Tipo (1:N):** NO se creará una nueva relación, como vimos antes para 1:N. Ahora bien, en la tabla asociada a la entidad que participa en la relación reflexiva aparecerá dos veces su PK (con nombres distintos):
  - una vez como PK de la tabla, con el nombre sin rol, y

# Interrelaciones reflexivas

Se tratarán de forma distinta según su tipo (N:M) o (1:N)

- **Tipo (N:M):** se creará una nueva relación como vimos antes para N:M, pero la PK de la entidad que participa aparecerá dos veces (*con nombres distintos según el rol con el que participe en la relación reflexiva*).
- **Tipo (1:N):** NO se creará una nueva relación, como vimos antes para 1:N. Ahora bien, en la tabla asociada a la entidad que participa en la relación reflexiva aparecerá dos veces su PK (con nombres distintos):
  - una vez como PK de la tabla, con el nombre sin rol, y
  - otra vez como FK de la tabla que la enlaza consigo misma, con el nombre de rol del extremo de cardinalidad máxima 1.

# Interrelaciones de grado $k \geq 3$

Se debe analizar la relación y estudiar la mejor opción en cada caso.

## Interrelaciones de grado $k \geq 3$

Se debe analizar la relación y estudiar la mejor opción en cada caso.  
Se propone una **solución general** en el caso de que al menos dos de las entidades participen con cardinalidad máxima  $n$ :

## Interrelaciones de grado $k \geq 3$

Se debe analizar la relación y estudiar la mejor opción en cada caso. Se propone una **solución general** en el caso de que al menos dos de las entidades participen con cardinalidad máxima  $n$ :

- Se trata como una relación binaria (N:M). Esto es, se crea una nueva tabla para la relación siguiendo los pasos descritos para las relaciones de tipo (N:M).

## Interrelaciones de grado $k \geq 3$

Se debe analizar la relación y estudiar la mejor opción en cada caso. Se propone una **solución general** en el caso de que al menos dos de las entidades participen con cardinalidad máxima  $n$ :

- Se trata como una relación binaria (N:M). Esto es, se crea una nueva tabla para la relación siguiendo los pasos descritos para las relaciones de tipo (N:M).
- Ahora bien: en lugar de dos, habrá que añadir  $k$  claves ajenas en la tabla creada, y la PK de la nueva tabla no tiene por que contener a la suma de las PK de las entidades participantes. Puede que haya que eliminar alguno de los campos o añadir alguno de los atributos propios, según la semántica.

# Interrelaciones de grado $k \geq 3$ - Ejemplo

Ejemplo: Relación IMPARTE entre GRUPO, ASIGNATURA, AULA y PROFESOR, con atributo propio Horario.

Grupo	Cod-asig	Cod-aula	NIF-Prof	Horario
-------	----------	----------	----------	---------

Clave primaria:

## Interrelaciones de grado $k \geq 3$ - Ejemplo

Ejemplo: Relación IMPARTE entre GRUPO, ASIGNATURA, AULA y PROFESOR, con atributo propio Horario.

Grupo	Cod-asig	Cod-aula	NIF-Prof	Horario
-------	----------	----------	----------	---------

Clave primaria:

- PK = (Cod-asig, Grupo)

Claves ajenas:

## Interrelaciones de grado $k \geq 3$ - Ejemplo

Ejemplo: Relación IMPARTE entre GRUPO, ASIGNATURA, AULA y PROFESOR, con atributo propio Horario.

Grupo	Cod-asig	Cod-aula	NIF-Prof	Horario
-------	----------	----------	----------	---------

Clave primaria:

- PK = (Cod-asig, Grupo)

Claves ajenas:

- FK = Cod-asig (→ ASIGNATURA)
- FK = Grupo (→ GRUPO)
- FK = Cod-aula (→ AULA)
- FK = NIF-Prof (→ PROFESOR)

# Interrelaciones de grado $k \geq 3$

En el caso de que solamente una de las entidades participe con cardinalidad máxima  $n$ , **por lo general**:

# Interrelaciones de grado $k \geq 3$

En el caso de que solamente una de las entidades participe con cardinalidad máxima  $n$ , **por lo general**:

- El tratamiento será equivalente a tener una serie de interrelaciones binarias 1:N.

## Interrelaciones de grado $k \geq 3$

En el caso de que solamente una de las entidades participe con cardinalidad máxima  $n$ , **por lo general**:

- El tratamiento será equivalente a tener una serie de interrelaciones binarias 1:N.
- No se creará por tanto ninguna relación adicional.

## Interrelaciones de grado $k \geq 3$

En el caso de que solamente una de las entidades participe con cardinalidad máxima  $n$ , **por lo general**:

- El tratamiento será equivalente a tener una serie de interrelaciones binarias 1:N.
- No se creará por tanto ninguna relación adicional.
- Se añadirán a la relación correspondiente a la entidad que participa con cardinalidad máxima  $n$  todas las FKs necesarias para enlazarla con el resto de relaciones (una FK por relación, cada una con el número de campos que corresponda).

## Interrelaciones de grado $k \geq 3$

En el caso de que solamente una de las entidades participe con cardinalidad máxima  $n$ , **por lo general**:

- El tratamiento será equivalente a tener una serie de interrelaciones binarias 1:N.
- No se creará por tanto ninguna relación adicional.
- Se añadirán a la relación correspondiente a la entidad que participa con cardinalidad máxima  $n$  todas las FKs necesarias para enlazarla con el resto de relaciones (una FK por relación, cada una con el número de campos que corresponda).
- Hay que analizar bien la semántica, pues a veces lo anterior no será suficiente, requiriendo nuevas relaciones.

# Jerarquías de generalización/especialización

No existe una solución general para el paso de una jerarquía de generalización en el Diagrama Entidad/Relación a un conjunto de tablas en el Diagrama Relacional.

Hay que analizar las ventajas e inconvenientes en cada caso.

Propondremos tres soluciones:

- Tabla única.
- Orientada a Objetos.
- Directo del Diagrama E/R.

# Jerarquías de generalización/especialización

## Relación/tabla única

Se crea una única relación para representar la jerarquía, con:

- Nombre: nombre de la entidad padre.
- Clave primaria: PK de la entidad padre.

# Jerarquías de generalización/especialización

## Relación/tabla única

Se crea una única relación para representar la jerarquía, con:

- Nombre: nombre de la entidad padre.
- Clave primaria: PK de la entidad padre.

### Campos:

- atributos de la entidad padre,
- la unión de los atributos de los subtipos; y
- un nuevo campo *Tipo* para indicar a qué subtipo de la jerarquía pertenece cada registro.

# Jerarquías de generalización/especialización

## Relación/tabla única

Se crea una única relación para representar la jerarquía, con:

- Nombre: nombre de la entidad padre.
- Clave primaria: PK de la entidad padre.

### Campos:

- atributos de la entidad padre,
- la unión de los atributos de los subtipos; y
- un nuevo campo *Tipo* para indicar a qué subtipo de la jerarquía pertenece cada registro.

### Inconvenientes:

- Aparición de muchos valores nulos.
- Pérdida de información si existen en el DER relaciones en las que no participa la entidad padre sino un cierto subtipo.

# Jerarquías de generalización/especialización

## Orientada a Objetos

Se añade una nueva relación por cada subtipo y se consideran que son entidades distintas (no se incluye una relación para la entidad padre a menos que la jerarquía sea parcial).

# Jerarquías de generalización/especialización

## Orientada a Objetos

Se añade una nueva relación por cada subtipo y se consideran que son entidades distintas (no se incluye una relación para la entidad padre a menos que la jerarquía sea parcial). Para **cada subtipo**, su relación contendrá los campos:

- atributos de la entidad padre (la PK de la entidad padre será la PK de la relación); y
- atributos propios del subtipo en cuestión.

# Jerarquías de generalización/especialización

## Orientada a Objetos

Se añade una nueva relación por cada subtipo y se consideran que son entidades distintas (no se incluye una relación para la entidad padre a menos que la jerarquía sea parcial). Para **cada subtipo**, su relación contendrá los campos:

- atributos de la entidad padre (la PK de la entidad padre será la PK de la relación); y
- atributos propios del subtipo en cuestión.

### Inconvenientes:

- Una interrelación del DER en la que participa la entidad padre ha de clonarse para cada subtipo (aparecen muchas relaciones).
- Información redundante (los atributos de la entidad padre se repiten para cada subtipo de la jerarquía).
- Mucha repetición en jerarquías solapadas.

# Jerarquías de generalización/especialización

## Directo del DER

Solución intermedia: se añaden relaciones para la entidad padre y subtipos, y se enlazan mediante claves ajenas. En la relación padre:

- Campos: atributos de la entidad padre.
- PK = PK de la entidad padre.

# Jerarquías de generalización/especialización

## Directo del DER

Solución intermedia: se añaden relaciones para la entidad padre y subtipos, y se enlazan mediante claves ajenas. En la relación padre:

- Campos: atributos de la entidad padre.
- PK = PK de la entidad padre.

Para **cada subtipo**, su relación contendrá los campos:

- atributos de la entidad subtipo + PK de la entidad padre.
- PK = PK de la entidad padre.
- FK = PK entidad padre (→ relación padre).

# Jerarquías de generalización/especialización

## Directo del DER

Solución intermedia: se añaden relaciones para la entidad padre y subtipos, y se enlazan mediante claves ajenas. En la relación padre:

- Campos: atributos de la entidad padre.
- PK = PK de la entidad padre.

Para **cada subtipo**, su relación contendrá los campos:

- atributos de la entidad subtipo + PK de la entidad padre.
- PK = PK de la entidad padre.
- FK = PK entidad padre (→ relación padre).

### Inconvenientes:

- Se repiten registros. Cada registro de la jerarquía aparece dos veces: una en la tabla padre y otra en el subtipo correspondiente.
- Muchas claves ajenas. Puede ralentizar las consultas en la BD.

# Índice

- 1 Introducción
- 2 Propiedades
- 3 Claves primarias y ajenas
- 4 Restricciones sobre el esquema relacional
- 5 Paso del DER al esquema relacional
- 6 Bibliografía**

# Bibliografía



Adoración de Miguel, Mario Piattini, Esperanza Marcos

*Diseño de Bases de Datos Relacionales.*

RA-MA Editorial (1999)

ISBN: 978-84-7897-385-9



Ramez Elmasri, Shamkant B. Navathe

*Fundamentos de Sistemas de Bases de Datos.*

Pearson Educación S.A. - Addison Wesley Quinta Ed. (2007)

ISBN: 978-84-7829-085-7



Mercedes Marqués

*Apuntes de Bases de Datos.*

Universidad Jaume I en Castellón (2011)

ISBN: 978-84-693-0146-3