

Diseño Lógico

Modulo: Bases de Datos

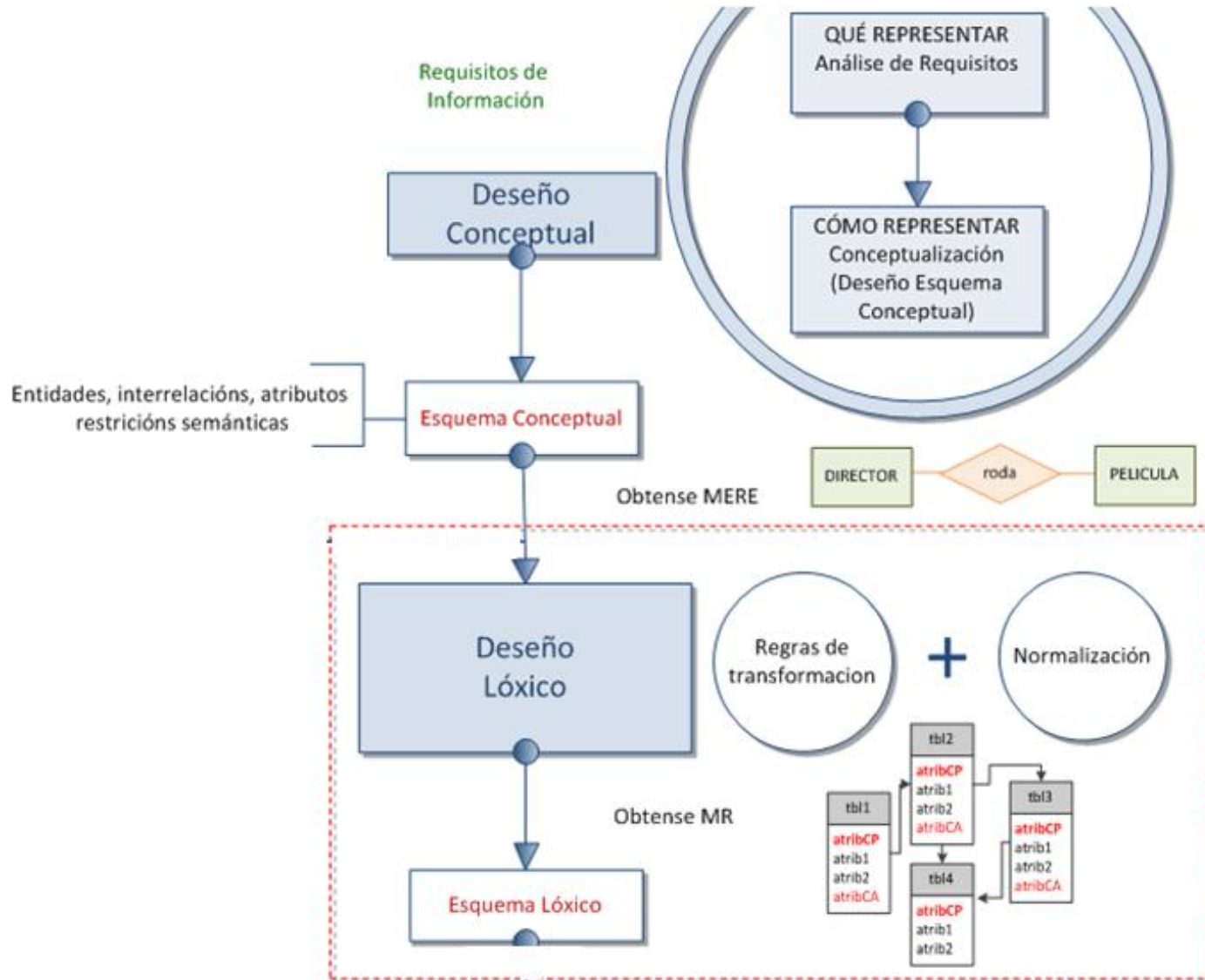
Diseño lógico

El diseño lógico es una parte (fase) del proceso de diseño de Bases de Datos.

Objetivo: Creación del **esquema logico** de la base de datos en el modelo de datos elegido (p.ej. relacional), independientemente del SGBD que se vaya a utilizar.

Un **esquema lógico** de una base de datos, es una descripción de la estructura de la base de datos que puede procesar un SGBD. El esquema lógico de base de datos depende de un tipo de SGBD (relacional, de redes, jerárquico...), pero no de un SGBD específico.

Diseño lógico



Diseño lógico. Etapas

Diseño Lógico Estándar

Parte del esquema conceptual resultante de la etapa de diseño conceptual de datos, y junto con los requisitos de usuario se construirá un “esquema lógico estándar”. El esquema lógico estándar puede expresarse empleando varias técnicas, entre las que se encuentran el diagrama de estructura de base de datos (DD) y el modelo relacional. El esquema lógico estándar se describirá empleando un lenguaje de definición de datos (DDL) estándar, habitualmente SQL-92 (estándar ISO) .

Diseño Lógico específico

A partir del esquema lógico estándar obtenido en la etapa anterior se elaborará el esquema específico a implementar en un SGBD comercial concreto empleando el propio lenguaje de definición de datos (DDL) del gestor.

La finalidad es la adaptación del diseño lógico obtenido a las características propias del SGBD. **En este contexto pueden darse los siguientes casos:**

Diseño lógico. Etapas (Continuación)

En este contexto pueden darse los siguientes casos:

- El SGBD soporta todos los elementos del modelo lógico estándar sin restricciones. La transformación del esquema estándar al específico y directa, teniendo sólo que transcribir a la sintaxis propia del SGBD empleado (normalmente SQL).
- El SGBD no soporta ciertos conceptos, o bien, los soporta pero con restricciones. En este caso se tendrán que implementar nuevas utilidades o realizar una programación complementaria o encargar a los programas el control de las restricciones no soportadas.

Modelo Relacional

Modulo: Bases de Datos

INTRODUCCIÓN

- Prouesto por E.F. Codd,1970.
- El modelo más usado en las aplicaciones comerciales de procesamiento de datos convencional.
- Dividido en 3 partes:
 1. Estructura de Datos.
 2. Integridad de Datos (características generales).
 3. Manipulación de Datos.

ESTRUCTURA DE DATOS RELACIONAL

Una base de datos estaría compuesta por un Conjunto de RELACIONES

- **Relación** representada mediante una **Tabla** con filas y columnas
 - Estructura de datos fundamental del modelo
 - Representa una entidad genérica
 - Tiene un nombre que la identifica
 - Compuesta de COLUMNAS
 - Cada columna representa un **atributo** de la entidad.
 - **Conjunto** de FILAS
 - Cada fila representa una **entidad concreta (ejemplar)**
- Modelo basado en Teoría matemática
 - Analogía entre Relación (concepto matemático) y Tabla
 - Teoría de Conjuntos y Lógica de Predicados de 1^{er} orden. (álgebra y cálculo relacional)

>> El modelo relacional tiene por lo tanto una sólida Base Formal.

Estructura de datos relacional: términos básicos

- **Relación = Tabla**
- **Tupla = Fila** $t \in R$ si la tupla t está en la relación R .
- **Cardinalidad:** nº de tuplas (filas) en una relación.
- **Atributo = Columna** (nombre único en cada relación).
- **Grado:** nº de atributos (columnas) en una relación
- **Dominio** colección de valores permitidos para ciertos atributos

Estructura de datos relacional.

Definiciones formales: DOMINIOS

- **Escalar**

- Menor unidad semántica de información: valor de un dato individual
- Valor *Atómico* (desde pto. de vista del modelo)

- **Dominio**

Conjunto de valores escalares (del mismo tipo) donde toman su valor los atributos

- Cada atributo estará definido sobre un ÚNICO dominio --- OBLIGATORIO
- Si A, B representan un mismo concepto, A y B tienen el mismo dominio
- Un dominio D puede contener valores no tomados por ningún atributo

$\{\text{valores de } a\} \subseteq \text{Dominio (A)}$

\subseteq contenido o igual

Estructura de datos relacional.

Definiciones formales: DOMINIOS

- **Definición de Dominio.** Forma parte de la Definición de la BD
 - Naturaleza conceptual: almacenado o no en la BD el *conj. real de valores*
 - *Creación*
CREAR DOMINIO Titulos CHARACTER(50);
Almacena descripción en el Catálogo del Sistema
 - *Destrucción*
DESTRUIR DOMINIO Titulos;
Elimina la definición del Catálogo del Sistema
 - *Definición de Datos*
CREAR RELACIÓN PELICULA
(titulo **Titulos,**
 duración **Tiempo,**
 director **Nombres,**
 estreno **Fechas**
 ...);

Estructura de datos relacional

Definiciones formales: RELACIONES

Una RELACIÓN **R**, sobre un conjunto de Dominios **D₁, D₂, ..., D_n** se compone de dos partes: cabecera y cuerpo

1. **Cabecera o Esquema:** Conjunto fijo de atributos (pares *Atributo: Dominio*)

$$\{(A_1:D_1), (A_2:D_2) \dots (A_n:D_n)\}$$

Cada A_j corresponde a un dominio único D_j

2. **Cuerpo:** Conjunto de tuplas variable en el tiempo

tupla = conjunto de pares *Atributo:Valor*

$$\{(A_1:V_{i1}), (A_2:V_{i2}) \dots (A_n:V_{in})\}, \text{ donde } i=1..m$$

- Cardinalidad **m** (cambia con el tiempo)
- Grado **n** (no suele variar)

Intensión de una relación: Parte definitorio y estática (no suele variar en el tiempo) de la relación, es lo que llamaremos esquema de relación.

Extensión: Conjunto de tuplas que, en un instante determinado, satisfacen el esquema de relación y se encuentran almacenadas en la base de datos; es lo que se suele llamar, simplemente, relación. La extensión varía en el transcurso del tiempo

Estructura de datos relacional

Definiciones formales: RELACIONES

- **Relación vs. Tabla**
 - Relación: Representación abstracta de un elemento o entidad de datos.
 - Tabla: Representación concreta de tal elemento abstracto
- **Modelo Relacional**
 - **Ventaja**
 - Representación muy sencilla (tabla) de su elemento abstracto básico (relación)
 - Fácil de utilizar, entender, razonar...

Estructura de datos relacional

Definiciones formales: RELACIONES

- Definición de Datos
 - Definición de cada Relación dentro de la Definición de la BD
 - *Creación*
CREAR RELACIÓN PELICULA

| | | |
|---|----------|----------|
| (| titulo | Titulos, |
| | duración | Tiempo, |
| | director | Nombres, |
| | estreno | Fechas |
| | ...) |); |
 - Almacenamiento de la descripción en el Catálogo del Sistema
 - Creación de la tabla “vacía” (sin tuplas)
 - Eliminación de una Relación
 - *Destrucción --- tuplas, tabla e información existente en el Catálogo*
DESTRUIR RELACION PELICULA;

Estructura de datos relacional

Definiciones formales: RELACIONES

- Propiedades de una Relación

1. **No existen tuplas repetidas**

2. **Las tuplas no están ordenadas**

Pq cuerpo = conjunto
matemático de tuplas

3. **Los atributos no están ordenados**

Pq cabecera = conjunto de pares *Atributo:Dominio*

4. **Los valores de atributos son Atómicos**

Pq Dominio = Conjunto de Valores Atómicos

- **Intersección** fila/columna = un solo valor (no una lista de valores)
- Si **R** cumple esta propiedad, **R** está **normalizada (en 1FN)**

Estructura de datos relacional

Definiciones formales: RELACIONES

- FORMAS NORMALES proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas. Cuanto más alta sea la forma normal aplicable a una tabla, menos vulnerable será a inconsistencias y anomalías

R está en determinada **FN** si
 cumple un conjunto de condiciones o **restricciones**,
 necesarias para estar **bien diseñada**,
 de acuerdo con el **modelo relacional** de datos.

- Toda relación ha de estar en 1FN (estructura de datos simple)

Estructura de datos relacional

BASE DE DATOS RELACIONAL

- **Base de Datos Relacional (BDR)**

- BD percibida por el usuario como una colección de relaciones *normalizadas de diversos grados y que varía con el tiempo*

- *Analogía con los conceptos de los sistemas de procesamiento de ficheros:*

| | |
|----------------|--|
| Relación ----- | Archivo |
| Tupla ----- | Registro concreto |
| | • sin orden <u>visible</u> entre registros |
| Atributo ----- | Campo |
| | • sin orden visible entre campos |
| | • valor atómico |

- Principio básico del MR:

Entidades representadas de una ÚNICA manera: por su VALOR

- Conexión lógica entre Relaciones (**Interrelación**)
representada mediante valores. No existen punteros

Estructura de datos relacional

BASE DE DATOS RELACIONAL

- En una BD distinguimos
 - ♦ **Esquema:** Diseño lógico de la BD
 - ♦ **Instancia:** Visión del conetenido de la BD en un determinado instante
- Esquema de una Relación
 - Conjunto con nombre de pares Atributos: **Dominio**

PELICULA (titulo:**Titulos**, duracion:**Tiempo**, director:**Nombres**, fecha:**Fechas**)

>> Esquema de BD = conjunto de esquemas de relación

- Esquema de Relación **no suele variar** (costoso)
 - reescritura de miles de tuplas
 - ¿valores de nuevos atributos para tuplas existentes?

Estructura de datos relacional:

INSTANCIA DE BASE DE DATOS

- Instancia de una Relación
 - Conjunto de tuplas que contiene en un instante dado (instantánea)
{("Volaverunt", ..., "ELuna, B.", ...)
("Torrente", ..., "Segura, S.")
("Barrio", ..., "León, F.", ...)}
 - >> Instancia de BD = conjunto de instancias de sus relaciones
 - Instancia de Relación **variable** en el tiempo
 - **nuevas** tuplas, **modificación** o **borrado** de existentes
- ESQUEMA de relación suele incluir, además de la cabecera, un conjunto de Reglas de Integridad

Características Generales de INTEGRIDAD de DATOS

- La Instancia de una Base de Datos **refleja la realidad**
Es un MODELO de una PORCIÓN del mundo real (UoD, universo de discurso)
- Algunas configuraciones de valores NO tienen SENTIDO pues NO REPRESENTAN ningún estado posible del mundo real modelado

>> La definición de la BD necesita **REGLAS DE INTEGRIDAD...**

- ◆ Informan al SGBD de **restricciones del mundo real**, así el SGBD evita configuraciones de datos imposible.
- ◆ Aumentan la capacidad expresiva del modelo relacional.
- ◆ Las **RI son específicas de la BD** considerada, PERO el MR incluye DOS CARACTERÍSTICAS GENERALES DE INTEGRIDAD IMPORTANTES y NECESARIAS en toda BD:
 - **Claves Candidatas y Primarias**
 - **Claves Ajenas** (o foráneas o externas)

CLAVES CANDIDATAS

- Sea R una relación $R(A_1:D_1, A_2:D_2, \dots, A_n:D_n)$
- Una **clave candidata** para R es un subconjunto **CK** de sus atributos tal que:
 1. (Prop. **Unicidad**) No existen dos tuplas distintas con igual valor para CK
 2. (Prop. **Irreductibilidad**) Ningún subconjunto de CK tiene la propiedad de unicidad
- Clave Candidata:
 - Simple: con 1 atributo
 - Compuesta: con más de 1 atributo
- Las claves candidatas deben ser irreductibles, pues cada clave candidata es una **RI** (restricción de integridad) para el sistema
- Mecanismo Básico de Direccionamiento a nivel de tuplas
- **Clave Primaria** (*Primary Key, PK*) es una clave candidata que se escoge como *identificador de las tuplas*
- **Claves Alternativas** (*AK*) es cualquier clave candidata que no sea primaria

CLAVES AJENAS (*Foreign Keys*, FK)

- Atributo (o conjunto de atributos) FK de una relación R2, tal que:
 1. Existe otra relación R1 con clave primaria PK, y
 2. Cada valor de FK en R2 es idéntico al valor de PK en alguna tupla de R1
- >> atributo (o conjunto de atributos) de una relación que hace referencia a una Clave Primaria de otra relación (o la misma)

Restricción de Integridad Referencial

Todo valor de una FK debe coincidir con un valor en la correspondiente PK

La BD no debe contener FK sin correspondencia:

Si B referencia a A, A debe existir

- Ojo: Puede existir valor de PK al que NO haga referencia ningún valor de la FK

CLAVES AJENAS (*Foreign Keys*, FK)

- Clave Ajena
 - Simple
 - Compuesta

Cada componente de una FK debe estar definido sobre el mismo DOMINIO que el correspondiente atributo de la PK a la que referencia

- Uso de Claves Ajenas facilita:
 - Eliminación de la Redundancia: Integridad entre ficheros
 - Mecanismo del MR para establecer VÍNCULOS ENTRE RELACIONES

CLAVES

- Definición de Datos

La **definición** de cada Relación incluirá...

- ❖ Clave primaria
- ❖ Claves alternativas y
- ❖ Claves ajenas

CREAR RELACION PELICULA

```
(  
    titulo Titulo,  
    duracion Tiempo,  
    director CHAR (9),  
    estreno Fechas,  
    ...  
    CLAVE PRIMARIA (titulo),  
    CLAVE AJENA (director)  
        REFERENCIA DIRECTOR (nif)  
    ...);
```

CREAR RELACION DIRECTOR

```
(  
    nif CHAR (9),  
    nombre Nombres,  
    nacionalidad Nacionalidades,  
    ...  
    CLAVE PRIMARIA (nif)  
    CLAVE ALTERNATIVA (nombre)  
    ...);
```


MANTENIMIENTO DE INTEGRIDAD REFERENCIAL: REGLAS DE CLAVES AJENAS

- Las operaciones que no satisfacen (violan) la Integridad Referencial dejan la BD en un estado incorrecto

¿Cómo evita el Sistemas esos Estados Incorrectos?

El Sistema puede...

1. **Rechazar** toda operación que pueda provocar un estado ilegal
2. **Aceptar** (y ejecutar) tales **operaciones**, pero realizar **acciones** que restauren la integridad de los datos

Diseñador define Reglas o *Políticas de Mantenimiento de la Integridad Referencial* para que la BD SIEMPRE alcance un Estado Final Legal

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL: REGLAS DE CLAVES AJENAS

R2 \rightarrow R1

POLÍTICAS

1. **BORRADO** DE UNA TUPLA **t** DE **R1** A LA QUE REFERENCIA OTRA(S) DE **R2**
 1. (Defecto) **Rechazar** la operación
Sólo permite borrar **t** si ninguna otra tupla hace referencia a **t**
 2. **Cascada**. Propagar la eliminación
 - 1º Borrar todas las tuplas de **R2** que referencian a **t**
 - 2º Eliminar **t**
 - (* 3. *Borrar a nulos – se verá después **)
2. **MODIFICACIÓN** DE UNA FK DE **R2** A UN VALOR NO EXISTENTE EN LA PK DE **R1**
 - (SIEMPRE) **Rechazar** la operación

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL: REGLAS DE CLAVES AJENAS

3. **MODIFICACIÓN** DEL VALOR DE LA CLAVE PRIMARIA DE UNA TUPLA **t** DE **R1** AL CUAL REFERENCIA UNA(S) TUPLA(S) DE **R2**
 1. (Defecto) **Rechazar** la operación
Sólo permite modificar la PK de **t** si ninguna otra tupla referencia a **t**
 2. **Cascada**. Propagar la modificación
 - Toda tupla de **R2** que referencia a **t** seguirá haciendolo:
cambiar su valor de clave ajena al nuevo de la clave primaria de **t**
 - Modificar el valor de la clave primaria de **t**

(* 3. *Modificar a nulos -- se verá después* *)
4. **INSERCIÓN** DE UNA TUPLA **t** en **R2** CUYO VALOR DE FK NO CORRESPONDE CON NINGÚNO DE LA PK EN **R1**
 - (SIEMPRE) **Rechazar** la operación

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL: REGLAS DE CLAVES AJENAS

- Diseñador de la BD indica (al SGBD) la política a seguir en la *Definición de cada Relación*, dentro de la *Defnición de la BD*

CREAR RELACION PELICULA

(

**titulo Titulos,
duracion Tiempo,
director Nombres,
estreno Fechas,**

● ● ●

CLAVE PRIMARIA (titulo),

CLAVE AJENA (director) REFERENCIA DIRECTOR (nombre)

AL BORRAR RECHAZO

AL MODIFICAR CASCADA

);

MANTENIMIENTO DE INTEGRIDAD REFERENCIAL

Operaciones Encadenadas

- Sean $R1$ y $R2$ relaciones tales que $R2 \rightarrow R1$, con Regla de Borrado en Cascada
- Borrado de una tupla de $R1 \Rightarrow$ borrado de tuplas de $R2$ que la referencian
- Si existe $R3$, tal que $R3 \rightarrow R2$ (con Regla Borrado X) entonces $R3 \rightarrow R2 \rightarrow R1$
- Borrado **directo** de tuplas de $R2 \equiv$ Borrado **provocado** en $R1$
 - a) Si $X = \text{CASCADA}$, no problema
 - b) Si $X = \text{RECHAZADO}$ y hay tuplas en $R3$ que las referencian
La operación completa FALLARÁ

****Comportamiento análogo para las modificaciones****

**Las operaciones de actualización en una BD son siempre ATÓMICAS:
se realiza “TODO o NADA”**

NULOS

- En el mundo real existe...
 - ✓ información perdida,
 - ✓ Ausencia de información o,
 - ✓ Valores no aplicables a ciertos atributos
- Necesaria una forma de representar estas situaciones en los sistemas de BD
⇒ **NULO** (NULL)
- Si una tupla tiene un atributo que contiene un NULO, significa que el valor real de tal atributo es desconocido

♣ **NULO no es un valor** en sí mismo,
sino un **indicador o marca de ausencia de información**
- * No hay dos NULOS iguales (*num_telefono NULL ≠ edad NULL*)

NULOS

- En la *definición* de cada *Relación* de la BD se indica qué atributos pueden o no contener NULO

CREAR RELACION ALUMNO

```
(  
    num_exp    Expedientes    NO NULO,  
    nombre     Nombres        NO NULO,  
    direccion  Direcciones,  
    num_telef   Telefonos      NULO,  
    ...  
    CLAVE PRIMARIA (num_exp),  
    ...  
);
```

Consecuencias de los NULOS en la INTEGRIDAD de Datos

- NULO y Claves Primarias

Restricción de Integridad de Entidad:
primaria puede contener valor nulo

Ningún atributo componente de una clave

CREAR RELACION EMPLEADO

```
(
    cod_emp    Codigos          NO NULO,
    nombre     Nombres          NO NULO,
    num_telef   Telefonos,
    dpto        Num_Departamentos,
               contener NULO?
    jefe        Codigos,
    CLAVE PRIMARIA (cod_emp),
    CLAVE AJENA depto REFERENCIA DEPARTAMENTO (numero),
    CLAVE AJENA jefe REFERENCIA EMPLEADO (cod_emp)
);
```

¿Qué pasaría si **cod_emp** pudiere

- NULO Y Claves Ajenas

El Modelo Relacional permite NULO como valor de clave ajena

Ejemplo: Empleado no asignados a ningún departamento o empleados sin jefe

Consecuencias de los NULOS en la INTEGRIDAD de Datos

- Hemos de **extender la Definición de Clave Ajena**

“Sea R2 una relación. FK es una Clave Ajena en R2 si es un subconjunto de sus atributos, tal que:

1. Existe otra relación R1 con valor de clave primaria PK y
2. En todo momento, cada valor de FK en R2
 - a) es NULO, o
 - b) es idéntico a un valor de PK en alguna tupla de R1”

Restricción de Integridad de Referencial:

La Base de Datos no debe contener valores no nulos de clave ajena sin correspondencia

Consecuencias de los NULOS en la INTEGRIDAD de Datos

- Hay que extender algunas POLÍTICAS de mantenimiento de la Integridad Referencial:

1. BORRADO DE UNA TUPLA t DE R_1 A LA QUE REFERENCIA OTRA(S) DE R_2

1. (Defecto) **Rechazar** la operación
2. **Cascada**. Propagar la eliminación
3. **Borrar a nulos.** ** Ojo: Si la FK de R_2 permite NULO
 - Las tuplas de R_2 que referencian a t pasan a contener NULL
 - Eliminación de t

Ejemplo: borrado de un departamento con empleados

3. MODIFICACIÓN DEL VALOR DE LA CLAVE PRIMARIA DE UNA TUPLA t DE R_1 AL CUAL REFERENCIA UNA(S) TUPLA(S) DE R_2

1. (Defecto) **Rechazar** la operación
2. **Cascada**. Propagar la modificación
3. **Modificar a nulos.** ** Ojo: Si la FK de R_2 permite NULO
 - Toda tupla de R_2 que referencia a t pasa a contener NULL
 - Modificar el valor de la clave primaria de t

Ejemplo: Cambio del **cod_emp** de un jefe que cambia de puesto de trabajo

RESUMIENDO ...

El **DBMS** se encarga de ...

- **Comprobar las claves candidatas** (Primarias + Alternativas)

No existen dos tuplas distintas con igual valor para la Clave Candidata

→ *Definición de BD:* indicar los Atributos Componentes de las Claves Candidatas ←

- **Comprobar la Integridad de la Clave Primaria**

Un atributo componente de una clava primaria no puede contener NULO

→ *Definición de BD:* indicar los Atributos Componentes de la Clave Primaria ←

- **Comprobar la Integridad Referencial...**

El valor de la clave ajena en cualquier tupla, o coincide con un valor de clave primaria de alguna tupla en la relación referenciada, o es NULO

→ *Definición de BD:* indicar los Atributos Componentes de las Clave Ajenas ←

- **...y mantenerla** frente operaciones que puedan violar la Integridad

→ *Definición de BD:* indicar Políticas de Mantenimiento de Integridad Referencial ←