Bases de Datos

Tema 02. El modelo relacional

Marta Elena Zorrilla Pantaleón

Rafael Duque Medina

DPTO. DE MATEMÁTICAS, ESTADÍSTICA Y COMPUTACIÓN

Tabla de contenido

Bibliografía

- Básica
  - Cap. 8 y 24. Elmasri, R., Navathe, S.B., Fundamentos de Sistemas de Bases de Datos, 5ª edición, Pearson Education, 2008.
  - Cap. 7. Pons, O. et al. Introducción a los sistemas de bases de datos. Paraninfo. 2008
  - Cap. 3 y 4. Silberschatz, A., Korth, H.F., Sudarshan, S., Fundamentos de Bases de Datos, 5ª edición, Madrid, 2006.
  - Cap. 3. Piattini, M., Marcos, E., Calero, C., Vela, B. Tecnología y diseño de bases de datos. Ra-ma, 2006.

- Complementaria
  - García Molina, H., Ullman, J., Widom, J. Database systems: the complete book. 2nd ed. Pearson Education International, cop. 2009.

Introducción

- En 1970, Codd publicó en ACM el trabajo "Un modelo de datos relacional para grandes bancos de datos compartidos" donde propuso un nuevo Modelo de Datos.

- Un **modelo de datos** se puede definir como un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Esto es un conjunto de conceptos, reglas y convenciones que permiten especificar datos, las relaciones entre ellos, su semántica asociada y las restricciones de integridad.

- El modelo relacional (MR) se caracteriza por:
  - Ser sencillo y uniforme (colección de tablas y lenguajes declarativos)
  - Tener una sólida fundamentación teórica: el modelo está definido con rigor matemático
  - Ser independiente del almacenamiento físico y de las aplicaciones.

Introducción (y 2)

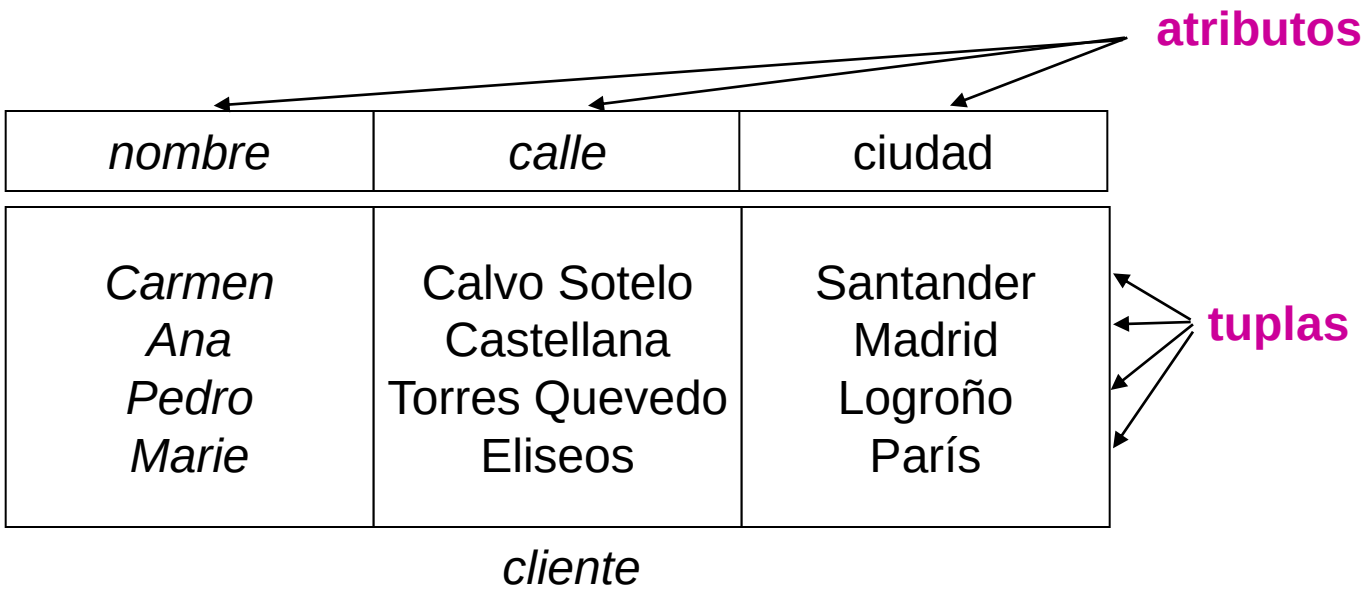
- Como ya se ha visto en el tema 1, una BD relacional consiste de un conjunto de tablas, cada una con un nombre único.
- Cada tabla tiene columnas que recogen los atributos que caracterizan cada entidad o elemento del mundo real.
- Cada fila en una tabla representa una relación entre un conjunto de valores. Como una tabla es una colección de tales relaciones, hay una correspondencia entre el concepto de tabla y el concepto matemático de relación, del cual este modelo toma su nombre.
- A continuación se explica el modelo relacional y las diferencias existentes entre el modelo matemático y el implementado en los gestores.

Elementos básicos del MR

- RELACIÓN
  - Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.
- DOMINIO
  - Es el conjunto válido de valores que toma un **atributo**. Existen con independencia de cualquier otro elemento.
- ATRIBUTO
  - Representa las propiedades de la relación. Se representa mediante una **columna**.
- TUPLA
  - Es una ocurrencia de la relación. Se representa mediante una **fila**.

Ejemplo de relación

El Universo de Discurso de una BD relacional está compuesto por un conjunto de dominios {Di} y de relaciones {Ri} definidas sobre los dominios.



Dominios

Un **dominio** es un conjunto nominado, finito y homogéneo de valores atómicos

- Un dominio =>
  - se identifica por un nombre,
  - tiene un número finito de valores,
  - todos los valores son del mismo tipo, y
  - los valores son atómicos respecto del MR
- Cada dominio puede definirse de dos maneras:
  - Extensión (dando sus posibles valores):
    - días de la semana = {lunes, martes, miércoles, … sábado, domingo}
  - Intensión (mediante un tipo de datos):
    - peso = decimal
  - A veces se asocia al dominio su unidad de medida (kilos, metros, etc.) y/o ciertas restricciones (como un rango de valores).

Atributos

Un **atributo** (A) es la interpretación de un determinado dominio en una relación, es decir el "papel" que juega en la misma.

- Notación:

  D = Dom (A) => D es el dominio de A

- Un atributo y un dominio pueden llamarse igual, pero …
  - Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia propia con independencia de las relaciones.
  - Un atributo representa una propiedad de una relación.
  - Un atributo toma valores de un dominio.
  - Varios atributos distintos (de la misma o de diferentes relaciones) pueden tomar sus valores del mismo dominio.

Relación

Una **relación** (matemáticamente) es un subconjunto del producto cartesiano de la lista de dominios $\{D_i\}$

- Esta definición no tiene en cuenta a los atributos, por eso en Bases de Datos se utiliza otra definición "un esquema de relación se compone de un nombre de relación R, un conjunto de n atributos $\{A_i\}$ y de un conjunto de n dominios (no necesariamente distintos) $\{D_i\}$ donde cada atributo será definido sobre un dominio".

- Una relación consta de los siguientes elementos:
  - Nombre de la relación
  - Cabecera: conjunto de n pares atributo-dominio
  - Cuerpo: Conjunto de m tuplas
  - Esquema: constituido por el nombre de la relación y la cabecera
  - Estado: constituido por el esquema y cuerpo.

Relación (y 2)

- Hay que diferenciar:
  - **Esquema** : conjunto de atributos $\{A_i\}$ junto con sus dominios (diseño lógico de la BD)
  - **Instancia** : conjunto de tuplas $r=\{t_1,\ldots,t_n\}$ tal que $t_i=(x_1,..,x_n)$ con $x_j \in D_j$ (instancia del esquema, esto es, datos que en un momento determinado están en la BD)

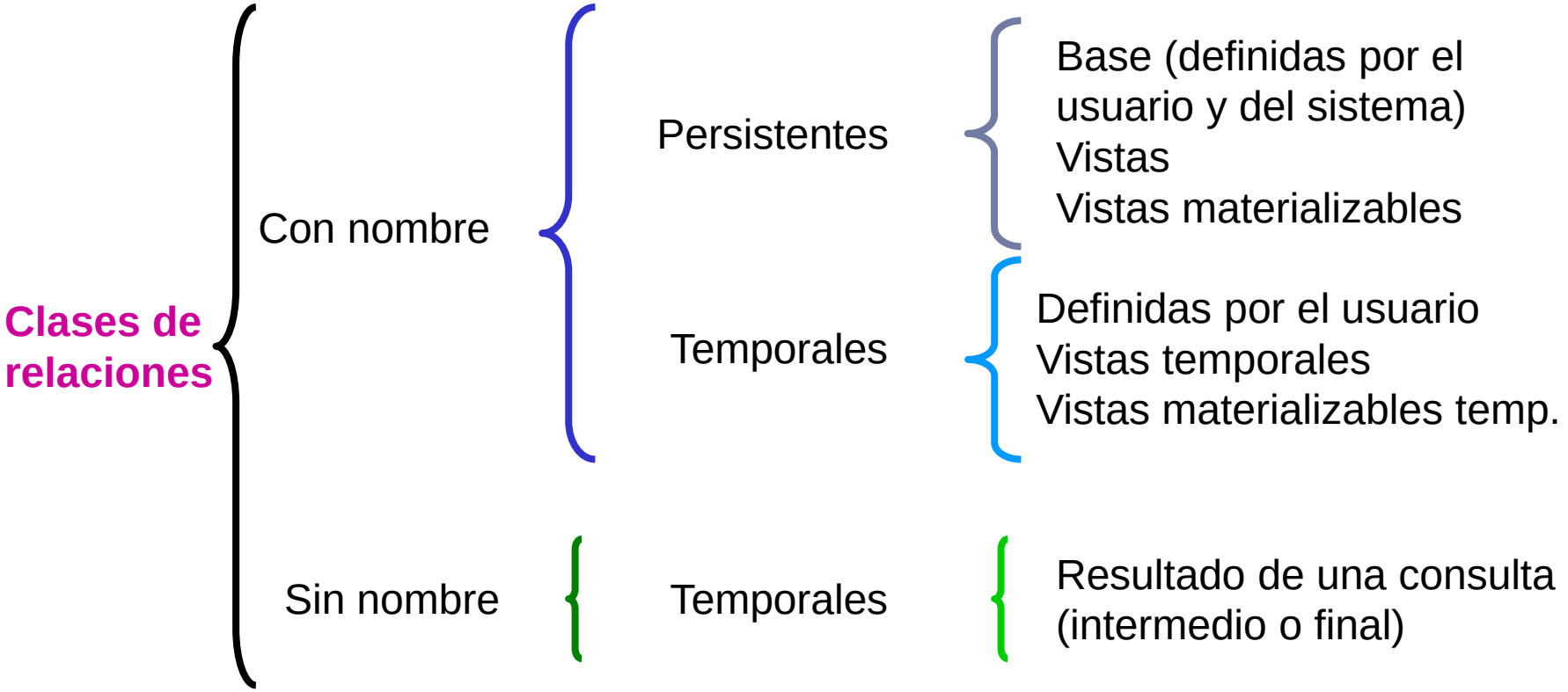
Esquema:
Persona [nombre: **Nombres**, calle: **Calles**, ciudad: **Ciudades**]

Instancia:
(Carmen, Calvo Sotelo, Santander),
(Ana, Castellana, Madrid),
(Pedro, Torres Quevedo, Logroño),
(Marie, Eliseos, Paris)

Se denomina **cardinalidad o aridad** de una relación al número de tuplas que hay en un esquema. Y **grado** al nº de atributos.

Base de datos relacional

Una **base de datos relacional** es un conjunto finito de relaciones $\{R_i\}$



Terminología

RELACIÓN		TABLA		FICHERO
TUPLA	~	FILA	~	REGISTRO
ATRIBUTO		COLUMNA		CAMPO
GRADO		Nº DE COLUMNAS		Nº DE CAMPOS
CARDINALIDAD		Nº DE FILAS		Nº DE REGISTROS

Modelo	**SGBD**	**Sistemas**
Relacional	**Relacionales**	**de Ficheros**
(teoría)	(implementación)	**Clásicos**

!CUIDADO!, una relación no es una tabla. Ni una tabla es un fichero. Existen diferencias entre los conceptos.

Restricciones inherentes

- Las restricciones inherentes vienen impuestas por el propio Modelo de Datos.

- En el caso del MR, una relación tiene unas propiedades intrínsecas que no tiene una tabla, 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 PK
  - El orden de las tuplas no es significativo.
  - El orden de los atributos no es significativo.
  - Cada atributo sólo puede tomar un único valor del dominio subyacente
    - Se dice que la relación está normalizada (en 1FN).

- Otra restricción es **la regla de integridad de entidad**:

  "Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo"

Tabla vs. relación

- Una **relación** es un concepto abstracto de origen matemático.
- Una **tabla** es una forma de representar (implementar) una relación (una estructura de datos).

- Una tabla **no** tiene las restricciones inherentes de una relación como conjunto:
  - Puede haber dos filas iguales.
  - Las filas están ordenadas en el orden de grabación física por defecto o según el valor de la clave primaria.
  - Los atributos tienen un orden según se han definido en la tabla.
  - En cada celda de una tabla puede haber uno o varios valores. Si bien en el segundo caso se puede obtener una tabla equivalente que cumple la regla de normalización.

Clave (key)

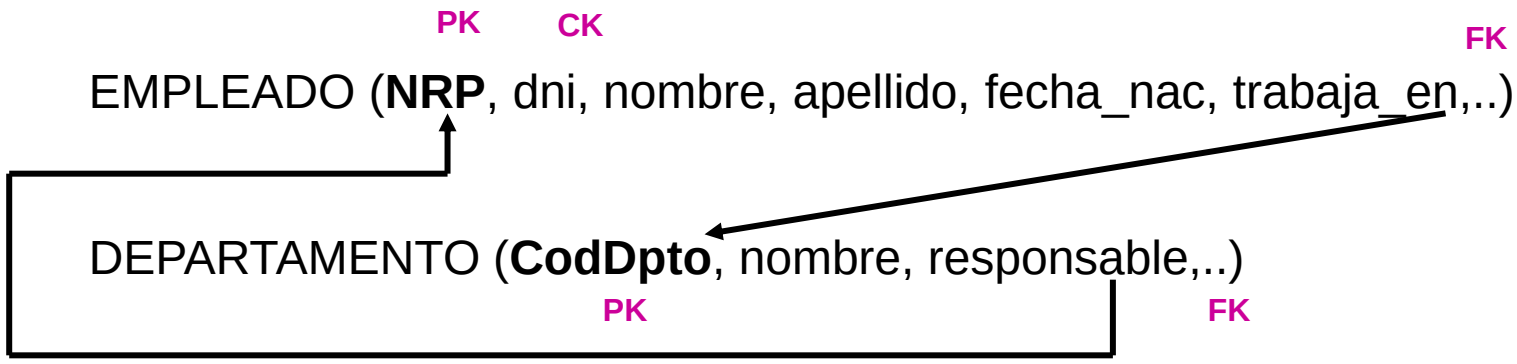
Clave Candidata (Candidate Key): conjunto de atributos que identifican unívoca y mínimamente cada tupla de la relación.

- De la definición de relación se deriva que siempre existe, al menos, una clave candidata.
- La propiedad de minimalidad implica que no se incluye ningún atributo innecesario: *CK cumple la propiedad de minimalidad si no existe un atributo X tal que {CK-X} sea clave candidata.*
- Una relación puede tener más de una clave candidata. En este caso se debe distinguir entre:
  - **Clave Primaria** (Primary Key): NRP para empleado
    - Es la clave candidata que el usuario escoge para identificar las tuplas de la relación.
    - Cuando sólo existe una clave candidata, ésta es la clave primaria.
  - **Claves Alternativas** (Alternative Key): DNI, PASAPORTE para empleado
    - Las claves candidatas que no han sido escogidas como clave primaria.

Clave (key) (y 2)

Clave Ajena (Foreign Key): Se denomina clave ajena de una relación R2 a un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de una clave candidata de una relación R1.

- R1 y R2 pueden ser la misma relación.
- La clave ajena y la correspondiente clave candidata han de estar definidas sobre el mismo dominio.



Restricciones semánticas

- Son definidas por el usuario.
- Son facilidades que el modelo ofrece a los diseñadores 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 (incorporados a SQL 92) son:
  - Clave Primaria (**PRIMARY KEY**)
  - Unicidad (**UNIQUE**)
  - Obligatoriedad (**NOT NULL**)
  - Integridad Referencial (**FOREIGN KEY**)
  - Verificación (**CHECK**)
  - Aserción (**CREATE ASSERTION**)
  - Disparador (**TRIGGER**), incluido en SQL:1999

Restricciones semánticas (y 2)

- Clave Primaria (**PRIMARY KEY**):
  - Permite declarar un atributo o un conjunto de atributos como clave primaria de una relación => sus valores no se podrán repetir ni se admitirán los nulos.
  - Ni el SQL92 ni los SGBD's relacionales obligan a la declaración de una clave primaria para cada tabla (el modelo teórico sí la impone), aunque permiten la definición de la misma.
  - Se debe 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.

- Unicidad (**UNIQUE**):
  - Los valores de un conjunto de atributos (uno o más) no pueden repetirse en una relación. Permite la definición de claves alternativas.

- Obligatoriedad (**NOT NULL**):
  - El conjunto de atributos no admite valores nulos.

Restricciones semánticas: Foreign key

- 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 relación R1 (relación referenciada), todo valor de dicho descriptor CA debe coincidir con un valor de CC o ser nulo.

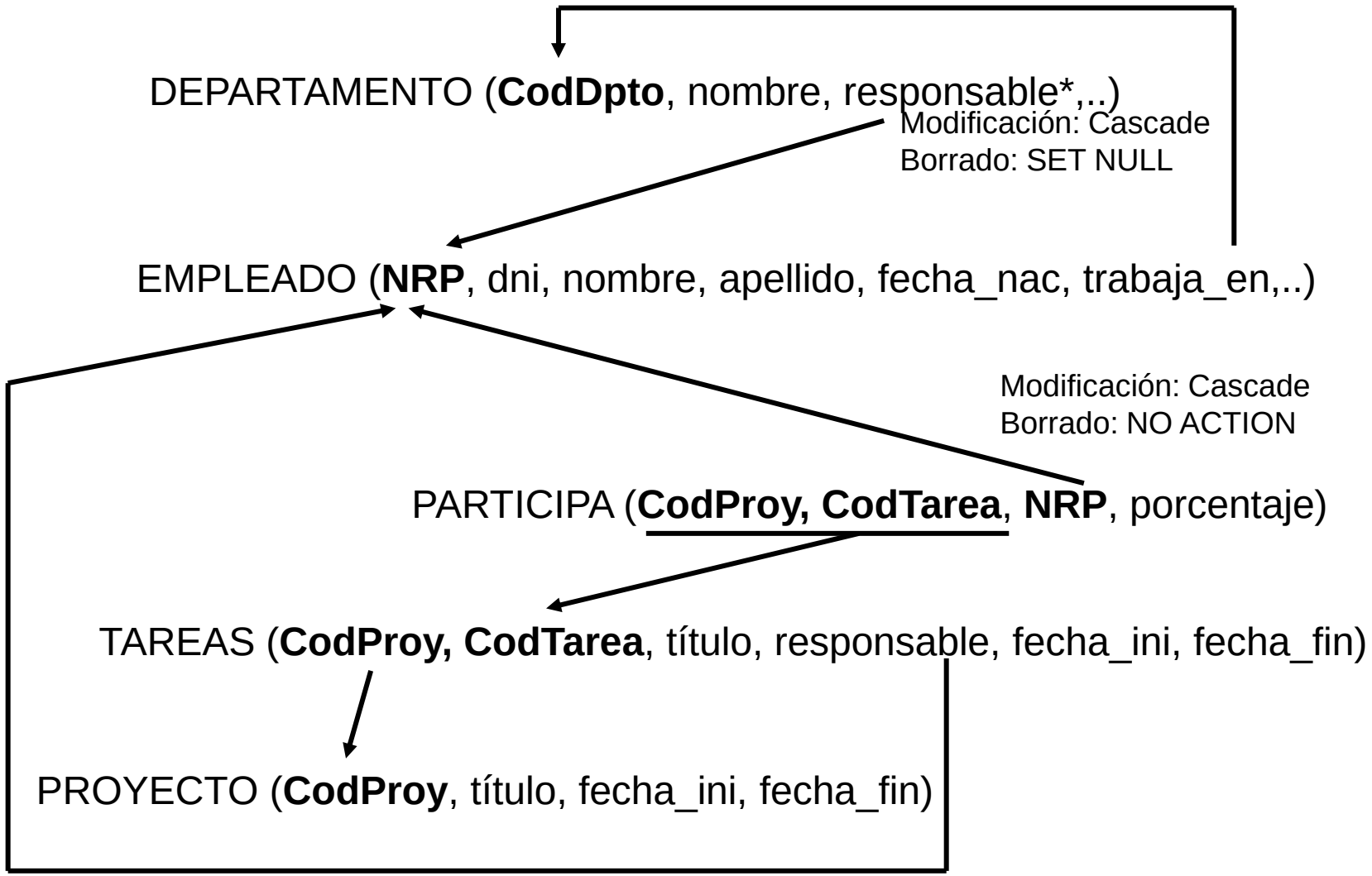
- La condición puede expresarse como R2.CA = R1.CC
  - El descriptor CA es, por tanto, una clave ajena de la relación R2.
  - Las relaciones R1 y R2 no son necesariamente distintas.
  - La clave ajena puede ser también parte (o la totalidad) de la clave primaria de R2.
  - CA puede admitir nulos o tener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

- Todo atributo de una clave primaria compuesta de una relación R2 que no está definido sobre un dominio compuesto, debe ser clave ajena de R2 referenciando a una relación R1, cuya clave primaria sea simple.

Restricciones semánticas: Foreign key (y 2)

- Integridad Referencial (FOREING KEY):
  - Además de definir las claves ajenas, hay que determinar las consecuencias que se producen al borrar o actualizar la relación referenciada. Según el estándar SQL92:
    - **NO ACTION**: rechazar la operación de borrado o modificación.
    - **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.
  - Los modos borrar y modificar son independientes, es decir, cada uno tomará una de las cuatro opciones por separado.

Ejemplo


DEPARTAMENTO (**CodDpto**, nombre, responsable*,..)
Modificación: Cascade
Borrado: SET NULL
EMPLEADO (**NRP**, dni, nombre, apellido, fecha_nac, trabaja_en,..)
Modificación: Cascade
Borrado: NO ACTION
PARTICIPA (**CodProy, CodTarea**, **NRP**, porcentaje)
TAREAS (**CodProy, CodTarea**, título, responsable, fecha_ini, fecha_fin)
PROYECTO (**CodProy**, título, fecha_ini, fecha_fin)

Restricciones semánticas: Verificación

- **CHECK**: Comprueba, en toda operación de actualización, si el predicado es cierto o falso y, en este último caso, rechaza la operación.
- La restricción de verificación se define sobre un único elemento

```
CHECK (porcentaje > 0 and porcentaje <100)
```

- O a nivel de relación

```
CHECK (fecha_fin >= fecha_ini)
```

  - Siempre dentro de un CREATE TABLE. Puede o no tener nombre.

Restricciones semánticas: Asertos

- **ASSERTION**: Actúa de forma idéntica a la anterior, pero se diferencia de ella en que puede afectar a varios elementos (por ejemplo, a dos relaciones distintas).
- Su definición no va unida a la de un determinado elemento del esquema y siempre ha de tener un nombre.

```
CREATE ASSERTION ctrl_proyecto CHECK
   (not exists (SELECT CE.trabaja_en as dpto_currito, CT.codproy, CT.codtarea
                FROM  empleado CE, participa CT   where CE.nrp = CT.nrp and
                CE.trabaja_en not in(
                    SELECT RE.trabaja_en  from empleado RE, tarea PT
                     where RE.nrp = PT.responsable and PT.codproy=CT.codproy and
                    PT.codtarea=CT.codtarea))
```

Empleados que participan en una tarea sean del mismo Dpto que su responsable

Restricciones semánticas: Disparadores

- Restricción en la que el usuario pueda especificar la respuesta (acción) ante una determinada condición.

```
CREATE TRIGGER ctrl_participa ON PARTICIPA   FOR   INSERT, UPDATE   AS
DECLARE @total  float

  SELECT  @total= count(*)  FROM   inserted
    WHERE  100 < (select sum(porcentaje) from participa
          where  participa.nrp=inserted.nrp )

  IF (@total>0)
   BEGIN
     RAISERROR (' Sobrecarga…', 16, 1)
     ROLLBACK TRANSACTION
     RETURN
   END
```

evitar la sobrecarga de los trabajadores
(disparador escrito en lenguaje TSQL)

Valores nulos

Valor nulo: utilizado para representar información desconocida, inaplicable, inexistente, no válida, no proporcionada, indefinida, etc.

- Necesidad de los valores nulos en BD:
  - Crear tuplas (filas) con ciertos atributos cuyo valor es desconocido en ese momento, p.e., la fecha de devolución de un préstamo.
  - Añadir un nuevo atributo a una relación existente; atributo que, en el momento de añadirse, no tendría ningún valor para las tuplas de la relación.
  - Atributos inaplicables a ciertas tuplas, por ejemplo, la editorial para un artículo (ya que un artículo no tiene editorial) o la profesión de un menor.
- Requiere tener cuidado en las consultas (is null)

Valores nulos (y 2)

- El tratamiento de valores nulos exige redefinir las operaciones de comparación, aritméticas, de agregación, etc. de forma específica para el caso en que un operando tome valor nulo.
- Obliga a introducir nuevos operadores especiales: **IS NULL , MAYBE**
- En las operaciones de comparación se hace necesario definir una lógica trivaluada incorporando el valor quizás (Q).

AND	C	Q	F
C	C	Q	F
Q	Q	Q	F
F	F	F	F

OR	C	Q	F
C	C	C	C
Q	C	Q	Q
F	C	Q	F

NOT	
C	F
Q	Q
F	C

- Se considera nulo el resultado de una suma, resta, multiplicación o división si alguno de los operandos toma valor nulo.
- En las agregaciones, no se consideran esas tuplas, a excepción del count(*)

Esquemas relacionales

- Ahora podemos dar una definición más completa de **esquema de una relación**:

R < A:D, S >

siendo

- R el nombre de la relación,
- A la lista de atributos,
- D los dominios sobre los que están definidos los atributos, y
- S las restricciones de integridad intraelementos (afectan a atributos y/o tuplas de una única relación).

- El **esquema de una base de datos relacional** será:

E < {Ri }, {Ii } >

siendo

- E el nombre del esquema relacional,
- {Ri} el conjunto de esquemas de relación, y
- {Ii} el conjunto de restricciones de integridad interelementos (afectan a más de una relación y/o dominio).

Esquemas relacionales (y 2)

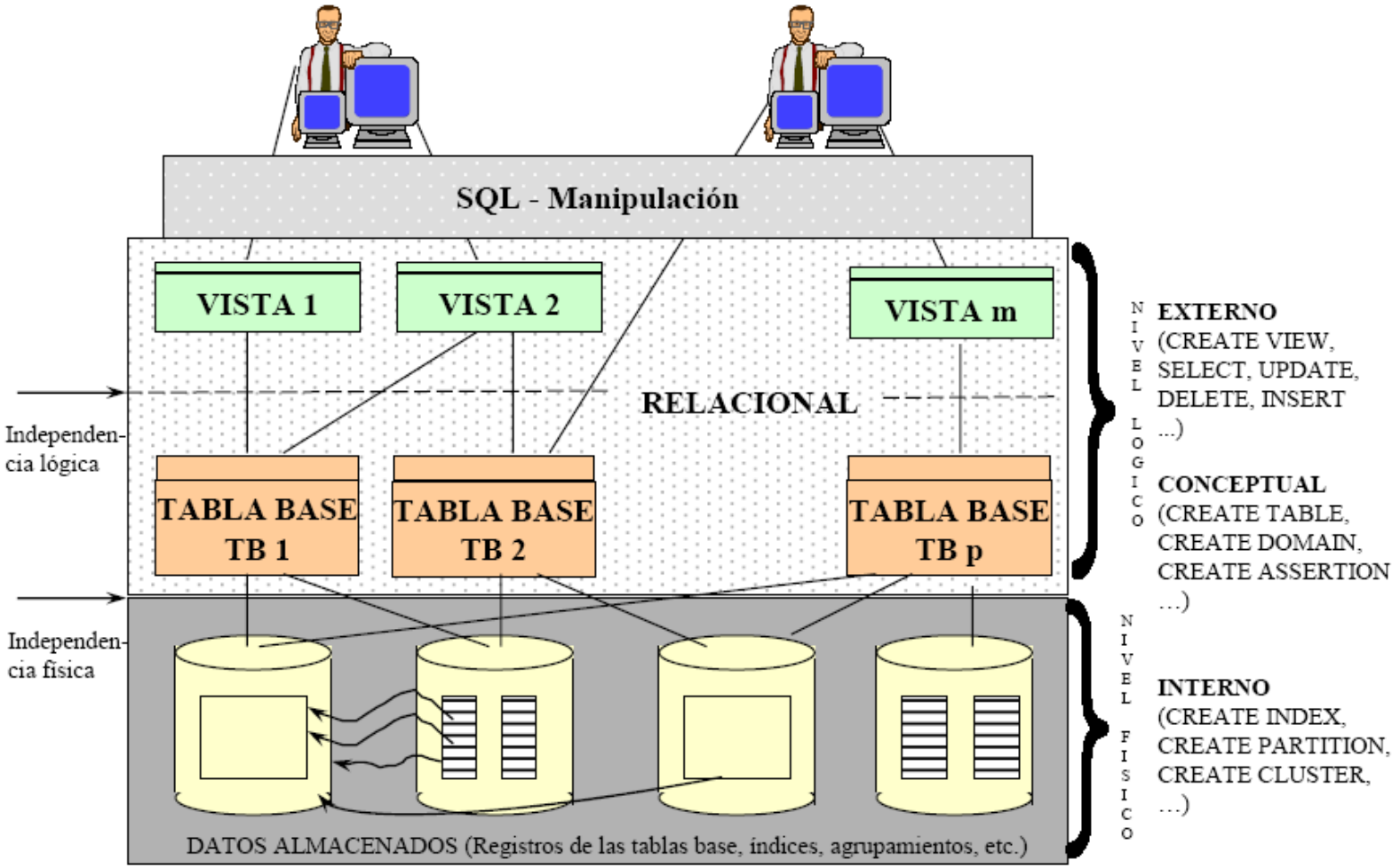
- En términos de implementación en SQL, un esquema E constará de:

E <R, D, T, V>

siendo

- R el conjunto de esquemas de relación (CREATE TABLE),
- D el conjunto de definiciones de dominios (CREATE DOMAIN),
- T el conjunto de restricciones de integridad entre relaciones y sobre dominios (CREATE ASSERTION, CREATE TRIGGER, ...), y
- V el conjunto de vistas (CREATE VIEW).

Gestores relacionales (MR vs ANSI-SPARC)



(Piattini et al. 2006)

Gestores relacionales (MR vs ANSI-SPARC) (y 2)

- El MR se adapta a la arquitectura ANSI, pero con las siguientes excepciones importantes:
  - Al usuario se le permite ver, si tiene las correspondientes autorizaciones, tanto las relaciones base como las vistas, mientras que en la arquitectura ANSI, para un usuario, la BD está limitada al esquema externo -vistas-.
  - Aunque las vistas se corresponden con los esquemas externos de ANSI y éstos pueden actualizarse, en el MR no todas las vistas son actualizables.
  - Además, en la práctica muchos SGBD relacionales no responden a la arquitectura a tres niveles, ya que las definiciones del esquema conceptual y del esquema interno no están claramente diferenciadas.

Las 12 reglas de Codd (ComputerWorld 1985)

- Cuando el MR triunfó comercialmente, muchos fabricantes que tenían productos "antiguos" no relacionales optaron por retocarlos o "camuflarlos" añadiéndoles la etiqueta relacional.
- Esto supuso una confusión que Codd intentó arreglar publicando sus 12+1 reglas, que indican las características que debe tener un SGBD para ser auténticamente relacional.
- Regla 0:
  - **Un SGBD relacional debe emplear para gestionar la BD exclusivamente sus facilidades relacionales**.
- De esta regla genérica se derivan las 12 reglas de CODD.

Reglas de Codd (y 2)

- **1. Representación de la información:** Toda la información en la Base de datos es representada de forma explícita y única a nivel lógico, por medio de valores en columnas de filas de tablas.
- **2. Acceso garantizado:** Todo dato (valor atómico) debe ser accesible mediante una combinación de tabla, un valor de su clave y el nombre de una columna.
- **3. Tratamiento sistemático de valores nulos:** El SGBD debe soportar la representación y manipulación de información desconocida y/o no aplicable, independientemente del tipo de dato.
- **4. Catálogo en línea** (diccionario de datos) basado en el modelo relacional. La descripción de la base de datos se debe representar en el nivel lógico de la misma manera que los datos ordinarios, de forma que los usuarios autorizados puedan consultarla con el mismo lenguaje con el que consultan los datos.

Reglas de Codd (y 3)

- **5. Sublenguaje de datos completo**: El SGBD debe soportar al menos un lenguaje relacional:
  - a) con sintaxis lineal.
  - b) que pueda ser usado interactivamente o en programas (embebido).
  - c) con soporte para operaciones de:
    - definición de datos (p.e. declaración de vistas).
    - manipulación de datos (p.e. recuperación y modificación de tuplas).
    - restricciones de seguridad e integridad.
    - gestión de transacciones.
- **6. Actualización de vistas**: todas las vistas teóricamente actualizables deben poder serlo en la práctica.
- **7. Inserción, modificación y borrado de tuplas de alto nivel**: todas las operaciones de manipulación de datos deben operar sobre conjuntos de filas.

Reglas de Codd (y 3)

- **8. Independencia física de los datos**: cambios en los métodos de acceso físico o la forma de almacenamiento no deben afectar al acceso lógico a los datos.
- **9. Independencia lógica de los datos:** los programas de aplicación no deben ser afectados por cambios en las tablas que preservan la integridad.
- **10. Independencia de la integridad:** Las restricciones de integridad deben estar separadas de los programas, almacenadas en el catálogo de la BD para ser editadas mediante un sublenguaje de datos.
- **11. Independencia de la distribución:** Las aplicaciones no deben verse afectadas al distribuir (dividir entre varias máquinas), o al cambiar la distribución ya existente de la Base de Datos.
- **12. Regla de no subversión:** Si el sistema posee un interfaz de bajo nivel (p.e. mediante llamadas en C), éste no puede utilizarse para saltarse las reglas de integridad y las restricciones expresadas por medio de un lenguaje de más alto nivel.

Fases del diseño e implantación de BD

- Aunque en el diseño de una aplicación de BD debe incluir además del modelo de datos, los procesos, la interfaz de usuario y la seguridad. En este apartado se aborda exclusivamente las fases para el diseño e implantación de la BD.

- Análisis de requisitos. Descripción de la información a gestionar y sus procesos. Así como información de volumen de datos, volatilidad, normas de validación, gestor de implantación, etc..
  - Técnicas: Entrevistas con usuarios y expertos. Lectura de documentación

- Diseño conceptual: traducción del análisis de requisitos al esquema conceptual. Representación generalmente gráfica de las entidades y sus relaciones.
  - Este modelo se construye con alguna de estas técnicas: ER, UML u ORM.
  - Paralelamente se pueden utilizar diagrama de flujo de datos para recoger el detalle de procesos a implementar por disparadores o procedimientos.

Fases del diseño e implantación de BD (y 2)

- Diseño lógico: traducción del modelo conceptual al LDD del gestor correspondiente. Modelo relacional, Orientado Objeto, Objeto-Relacional, schemas XML, etc.
- Diseño físico: Transformar el modelo lógico (generalmente en SQL) al físico adaptándolo a las características del gestor y al rendimiento que se espera de la BD (tiempo de respuesta, usuarios concurrentes, Nº de transacciones, volumen de datos, procesos sobre tablas…)
- Carga de datos y pruebas: Carga inicial y pruebas de verificación y validación de los requisitos del sistema (tratar de violar las reglas de integridad, comportamiento ante valores límite de los tipos de dato, tiempos de respuesta en consultas frecuentes y en consultas complejas, etc.)
- Operación:
  - Puesta en marcha
  - Tareas de mantenimiento y monitorización

Clasificación de modelos de datos

- Conceptuales: enfocados a describir el mundo real con independencia de la máquina.
- Lógicos: Implementados por los gestores

Modelo conceptual	Modelo lógico
Independientes del SGBD	Dependen del SGBD
Mayor nivel de abstracción	Más próximos al ordenador
Más enfocados al diseño de alto nivel	Más enfocados a la implementación
Mayor capacidad semántica	Poca capacidad semántica
Interfaz usuario informático	Interfaz informático/sistema
Ej.: ER, UML, ORM	Ej: relacional, jerárquico, Objeto-Relacional, Orientado a Objetos, Schemas XML, etc.

Herramientas CASE (Computer Aided/Assisted Software/System Engineering)

- Para ayudar al analista/programador en la fase del diseño conceptual y su paso al lógico y físico, existen herramientas CASE como ERWin, PowerDesigner, EasyCASE, Oracle designer (Discoverer), Visio (Microsoft), etc.

- Estas herramientas permiten diseñar el modelo de datos conceptual siguiendo alguna de las técnicas mencionadas, generalmente ER o UML y obtener el esquema lógico, generalmente relacional escrito en lenguaje SQL estándar, o bien físico, el mismo modelo pero llevado al lenguaje del gestor (Oracle, SQL Server, MySQL, etc.)

Herramientas CASE (y 2)

- Ventajas que aportan:
  - Ayudan al diseño (*verificación de errores, validación respecto a la técnica, etc.)*
  - Reducen el tiempo de desarrollo
  - Facilitan el mantenimiento del esquema de datos (ingeniería directa e inversa)
  - Reducen el tiempo de la documentación
  - Facilitan la portabilidad a otros gestores

Herramientas CASE (y 3)

- Deficiencias:
  - Generalmente no recogen toda la riqueza semántica del modelo de datos.
  - Falta de un modelo de restricciones que genere las reglas de negocio en automático.
  - No ayuda a especificar el modelo físico adecuado, lo indica el diseñador, pero no le da pautas o medidas de rendimiento.
  - No ofrecen la posibilidad de diseñar en entornos distribuidos, OO, activas, … no hay modelo que permita representarlo.
  - Los atributos derivados pueden estar en el conceptual por razones semánticas y en el físico por razones de eficiencia, el problema es que la regla por la que se genera no se puede modelizar.

Ejercicio

Carretera (IdCarretera, nombre)

Area (IdArea, nombre)

Tramo (IdCarretera, NroTramo, Area)

Pasa (IdCarretera, NroTramo, CodMunicipio, PtokmEntra, PtoKmSale)

Municipio (CodMunicipio, nombre, localidad)

1. ¿Qué recoge esta base de datos relacional?
2. Identifique las claves principales, claves candidatas y claves ajenas
3. En las restricciones de referenciabilidad indique las consecuencias del borrado y la actualización
4. ¿Añadirías alguna restricción de integridad?