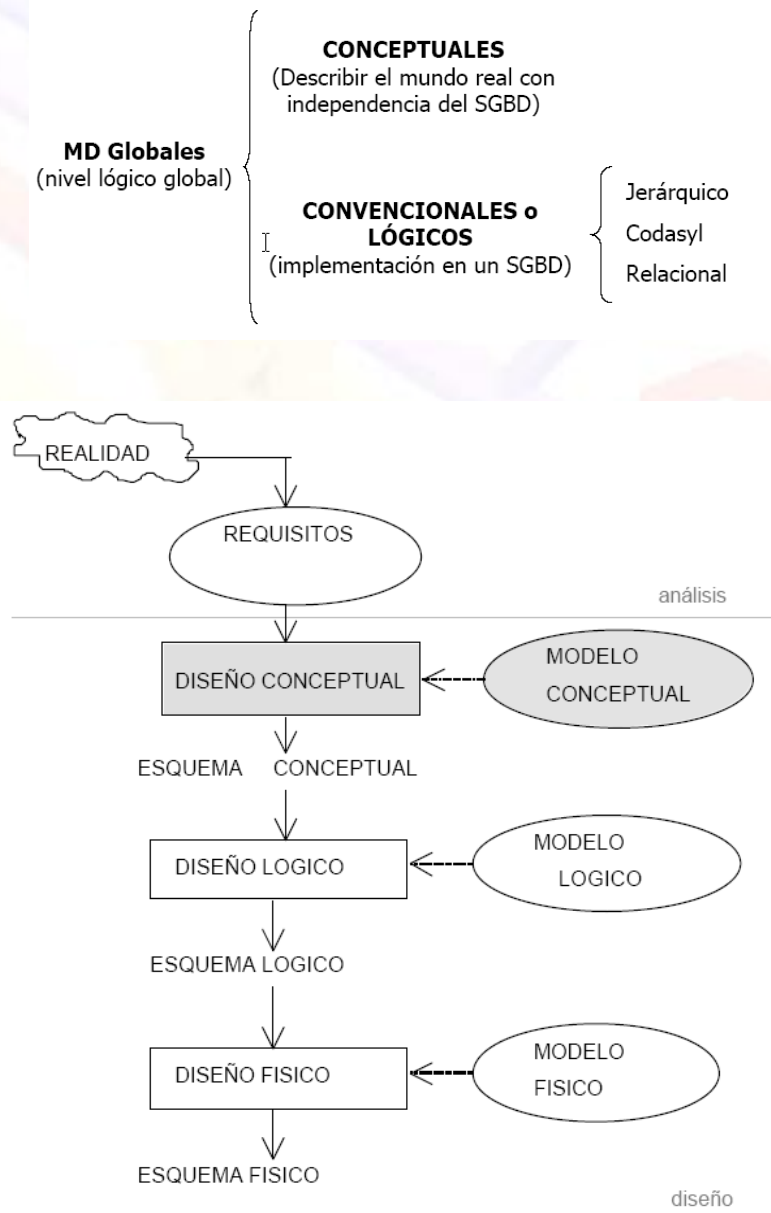


MODELOS DE DATOS (II)

I - Modelos de datos lógicos

En temas anteriores hemos estudiado el modelo conceptual, este modelo es completamente independiente del sistema gestor de base de datos.



Nuestros sistemas gestores son relacionales con lo cual el modelo a analizar será el modelo relacional.

Nuestro modelo lógico lo vamos a dividir en dos componentes:

- **modelo relacional estático** : Contempla aquellas propiedades del universo del discurso que son invariantes en el tiempo, es decir, su estructura.
- **modelo relacional dinámico** : Relativa a las propiedades del universo del discurso que varían con el tiempo. Incluye las operaciones que se aplican a los datos o valores almacenados en las estructuras.

Dentro del modelo relacional estático describiremos los siguientes elementos :

- Objetos (relaciones)
- Asociaciones entre objetos (interrelaciones)
- Propiedades o características de los objetos o asociaciones (atributos)
- Dominios (Conjuntos nominados de valores sobre los que se definen las propiedades)
- Elementos no permitidos o restricciones: Limitaciones impuestas a la estructura del esquema o a los datos que invalidan ciertas ocurrencias de la base de datos (integridad referencial, no permitir que dos filas de una tabla sean iguales).

Los elementos mencionados se representarán a través de tablas.

La componente dinámica consta de un conjunto de operadores que se definen sobre la estructura del modelo de datos relacional.

Realizaremos una serie de operaciones sobre la base de datos :

- Selección
- Creación
- actualización
- Eliminación
-

Estas operaciones en función del sistema gestor de base de datos podrán ser escritas en diversos lenguajes relacionales (por ejemplo el SQL o el GQBE). Estos lenguajes se definen como lenguajes de consulta; los lenguajes de consulta se usan para especificar las solicitudes de información.

Existen una serie de lenguajes que no son fáciles de usar ya que son la base de los lenguajes prácticos:

- álgebra relacional : forma la base del lenguaje de consulta SQL ampliamente usado.
- el cálculo relacional de tuplas
- el cálculo relacional de dominios : es la base del lenguaje QBE.

II - Componente estática del modelo relacional

Este modelo de datos es un modelo simple, potente y formal para representar la realidad. También ofrece una base firme para enfocar y analizar formalmente muchos problemas relacionados con la gestión de bases de datos, como el diseño de la base de datos, la redundancia, la distribución, etc. Son fundamentales para el diseño de las bases de datos relacionales el formalismo y una base matemática.

El elemento básico del modelo es la **relación**. El elemento relación es lo que nosotros hemos denominado tabla en el capítulo anterior (realmente la tabla es la forma de representar una relación).

Cada fila de la tabla se denomina **tupla** de la relación.

Cada columna de la tabla se denomina **atributo** de la relación.

El conjunto de todos los valores que puede adoptar un atributo en particular se denomina **dominio** de ese atributo. Un dominio estara compuesto por un nombre y un tipo de datos. Ejemplos de dominios :

- edades : numeros enteros entre 15 y 65
- localidades de Cantabria
- números de teléfono
-

La definición matemática de relación se desarrolla empezando por la noción de dominios. Un dominio es una colección de valores. Dados varios atributos, A_1, A_2, \dots, A_n , con dominios D_1, D_2, \dots, D_n , una relación es simplemente un subconjunto del producto cartesiano $D_1 \times D_2 \times \dots \times D_n$. En términos estrictos, tampoco importa el orden de las tuplas en la relación.

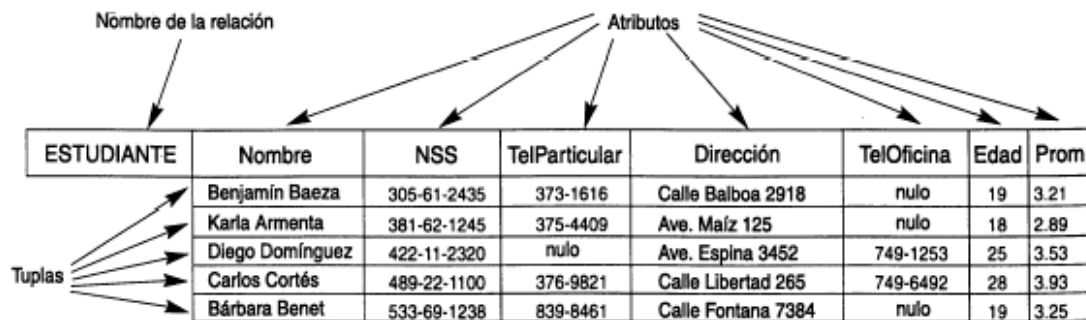


Diagram illustrating the structure of a table (relation) with labels pointing to its components:

- Nombre de la relación** points to the header cell **ESTUDIANTE**.
- Atributos** points to the header cells: **Nombre**, **NSS**, **TelParticular**, **Dirección**, **TelOficina**, **Edad**, and **Prom**.
- Tuplas** points to the rows of data.

ESTUDIANTE	Nombre	NSS	TelParticular	Dirección	TelOficina	Edad	Prom
	Benjamín Baeza	305-61-2435	373-1616	Calle Balboa 2918	nulo	19	3.21
	Karla Armenta	381-62-1245	375-4409	Ave. Maíz 125	nulo	18	2.89
	Diego Domínguez	422-11-2320	nulo	Ave. Espina 3452	749-1253	25	3.53
	Carlos Cortés	489-22-1100	376-9821	Calle Libertad 265	749-6492	28	3.93
	Bárbara Benet	533-69-1238	839-8461	Calle Fontana 7384	nulo	19	3.25

Los matemáticos definen las relaciones como subconjuntos del producto cartesiano de la lista de dominios como acabamos de definir. Esta definición se corresponde de manera casi exacta con la definición de tabla dada anteriormente. La única diferencia es que aquí se han asignado nombres a los atributos, mientras que los matemáticos sólo utilizan números, utilizando el entero 1 para denotar el atributo cuyo dominio aparece en primer lugar en la lista de dominios, 2 para el atributo cuyo dominio aparece en segundo lugar, etcétera. Como las tablas son esencialmente relaciones, se utilizarán los términos matemáticos relación y tupla en lugar de los términos tabla y fila

El **grado** de una relación es el número de columnas; la **cardinalidad** es el número de tuplas.

Un valor de dominio que es miembro de todos los dominios posibles es el **valor nulo**, que indica que el valor es desconocido o no existe.

Cuando se habla de bases de datos se debe diferenciar entre el esquema de la base de datos, o diseño lógico de la misma, y el ejemplar de la base de datos, que es una instantánea de los datos de la misma en un momento dado, es decir una ocurrencia de la base de datos.

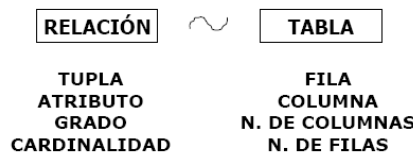
Esquema: ESTUDIANTE (NOMBRE,EDAD,SEXO)
Ocurrencia: estudiante

NOMBRE	EDAD	SEXO
Juan González	19	Varón
Antonia Sánchez	23	Hembra
Antonio Hernández	25	Varón
Alberto Redondo	20	Varón

Ejemplo de un esquema y ocurrencia de una relación

Una serie de reglas en una relación :

- No existen tuplas repetidas.
- Las tuplas no están ordenadas (de arriba hacia abajo).
- Los atributos no están ordenados (de izquierda a derecha).
- Todos los dominios de los atributos son atómicos.
- Los valores de una atributo deben ser atómicos, es decir no es divisible en componentes. Por ello en el modelo relacional no se permiten atributos compuestos ni multivaluados (primera forma normal). Los atributos multivaluados se deben representar con relaciones individuales, y los atributos compuestos se representan únicamente mediante sus atributos componentes simples.



Analicemos un poco mas el concepto de atomicidad. Un dominio es atómico si los elementos del dominio se consideran unidades indivisibles. Por ejemplo, el conjunto de los enteros es un dominio atómico, pero el conjunto de todos los conjuntos de enteros es un dominio no atómico.

III - Diferencia entre dominio y atributo

Un atributo y un dominio pueden recibir el mismo nombre, pero es preciso distinguirlos:

- Un atributo está siempre asociado a una relación, mientras que un dominio tiene existencia con independencia de las relaciones
- Un atributo representa una propiedad (que puede ser la identificación) 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

Ejemplo

Dominio: DNIs, Nombres, Edades
Relación: EMPLEADO(n_dni, nombre, edad_e)
Atributos

IV - Intención y extensión de una relación

Se habla de **intención** de una relación a su definición invariante en el tiempo; también denominada **esquema de relación**.

Se habla de **extensión** de la relación a su definición a través del conjunto de tuplas que, en un momento determinado, cumplen el esquema de relación, es decir es el conjunto de ocurrencias de la relación.

V - Claves

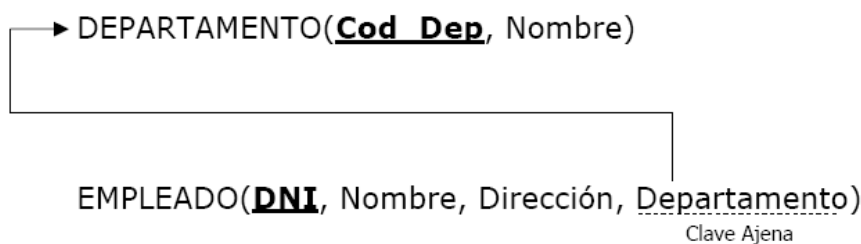
Los conceptos de superclave, de clave candidata y de clave primaria, se definen como en el modelo conceptual.

Una **superclave** de una entidad es un atributo o conjunto de atributos que permite distinguir de modo único cada entidad dentro de un conjunto de entidades.

Una **clave candidata** son las superclaves mínimas.

La **clave primaria** es la clave candidata elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades.

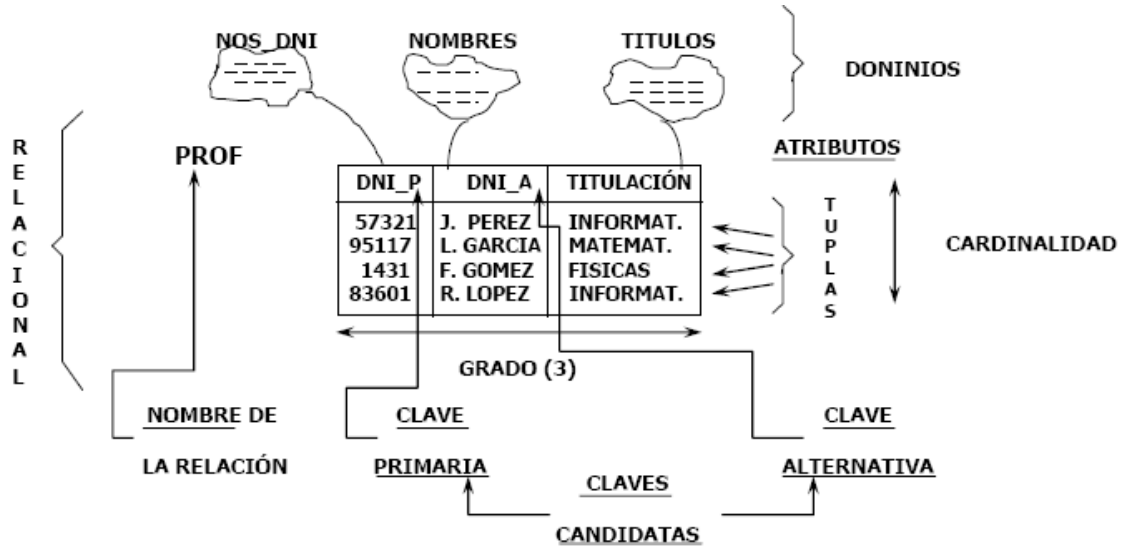
Ningún componente de la clave primaria de una relación puede en algún momento no tener valor (aceptar nulos).



En el modelo relacional existe una clave nueva definida como clave externa o foránea.

En el modelo relacional se denominan **claves ajenas o claves foráneas** a una asociación de una relación a otra, mediante su clave, es decir tendremos que un atributo de una relación es la clave principal de otra. Este concepto lo conocemos en el formalismo individual como una relación implícita.

Una Relación (R1) puede poseer como uno de sus atributos (A) una clave primaria de otra relación (R2). Este atributo (A) constituye una clave foránea en R1 y referencia a R2.



VI - Interrelaciones

Las interrelaciones, son similares a lo que denominamos como relaciones en el modelo conceptual.

La interrelaciones son asociaciones definidas entre los esquemas de relación.

Se distinguen distintos tipos según el número de tuplas que intervienen por parte de cada relación. A este número se le denomina **cardinalidad de la relación**. Puede ser:

- 1 a 1: correspondencia biunívoca : Se podría fusionar ambos esquemas en uno sólo. La clave ajena puede localizarse en cualquiera de los dos (o bien compartir clave primaria)
- 1 a n: correspondencia múltiple : Un esquema (1) es padre del otro (n). La clave ajena se sitúa en la relación que participa con múltiples tuplas.
- n a n: tiene que existir una relación intermedia (recordar como lo realizamos en ACCESS).

VII - Esquema de una relación, de una base de datos y diagrama de una relación

Como ya hemos visto el **esquema de una relación** podemos definirlo como una representación de la relación donde colocamos un nombre de relación R, un conjunto de n atributos {A1} y el dominio de cada uno de los atributos, donde cada atributo será definido sobre un dominio (finito o infinito). Además sobre el esquema de la relación se colocan también las claves principales y algunas de las restricciones que veremos en el resto del tema actual.

Ejemplos

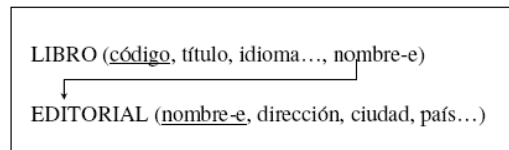
Podemos ver algunos ejemplos de esquemas de relaciones :

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

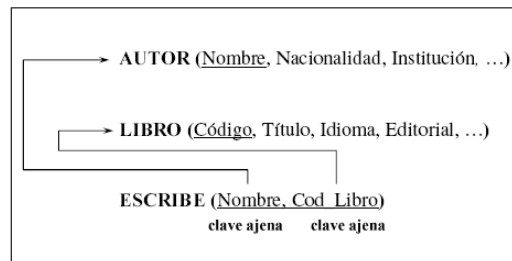
EDITORIAL (nombre-e, dirección, ciudad, país...)

El **esquema de una base de datos**, también llamado esquema relacional, es el conjunto de todos los esquemas de relaciones junto con las interrelaciones entre ellas. Además se colocaran otras restricciones, como por ejemplo la integridad referencial.

Ejemplo



Ejemplo

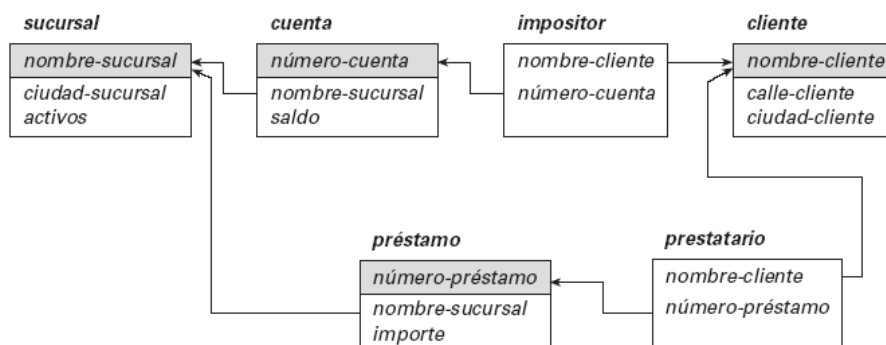


Un diagrama de relación es una mezcla entre lo que es un diagrama E/R con un grafo relacional. Es bastante utilizado en software de bases de datos.

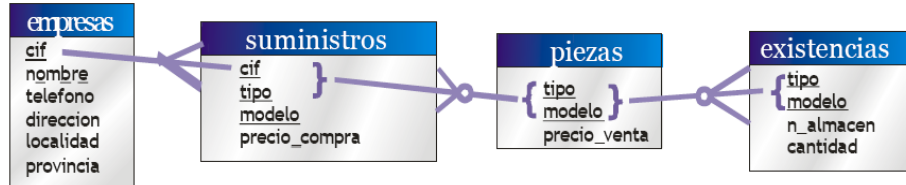
Cada relación aparece como un cuadro con los atributos listados dentro de él y el nombre de la relación sobre él. Si hay atributos clave primaria, una línea horizontal cruza el cuadro con los atributos clave primaria listados sobre ella. Las dependencias de clave externa aparecen como flechas desde los atributos clave externa de la relación referenciante a la clave primaria de la relación referenciada.

No hay que confundir un diagrama de esquema con un diagrama E-R. En particular, los diagramas E-R no muestran explícitamente los atributos clave externa, mientras que los diagramas de esquema sí.

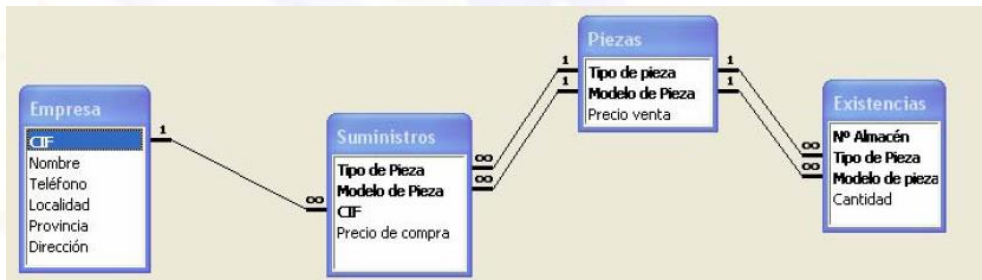
Ejemplo



Ejemplo



Ejemplo



VIII - Restricciones

Existen una serie de estructuras u ocurrencias no permitidas en el modelo relacional.

Se trata de unas condiciones de obligado cumplimiento por los datos de la base de datos.

Las hay de varios tipos.

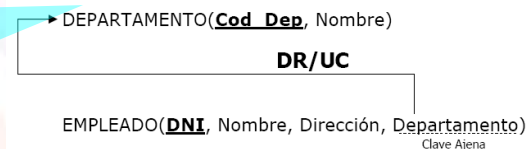
- **inherentes** : Son aquellas que no son determinadas por los usuarios, sino que son definidas por el hecho de que la base de datos sea relacional. Por ejemplo:
 - No puede haber dos tuplas iguales. Para conseguir esto es necesario que exista una clave primaria
 - El orden de las tuplas no importa
 - El orden de los atributos no importa
 - Cada atributo sólo puede tomar un valor en el dominio en el que está inscrito
 - Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo
- **semánticas** (o de usuario) : El modelo relacional permite a los usuario incorporar restricciones personales a los datos. Estas son utilidades ofrecidas por el modelo de datos con el fin de contemplar, de forma sencilla, la semántica del mundo real. Las principales son:
 - **Clave primaria**. Hace que los atributos marcados como clave primaria no puedan repetir valores ni dejarlos vacíos. La obligatoriedad de la existencia de una clave primaria es una restricción inherente; en cambio, la elección de los atributos que la componen es una restricción semántica.
 - **Unicidad**. Impide que los valores de los atributos marcados de esa forma, puedan repetirse. Define una clave alternativa.

- **Obligatoriedad.** Prohíbe que el atributo marcado de esta forma no tenga ningún valor, es decir se prohíbe colocar el valor NULL a este atributo
- **Integridad referencial.** Toda relación (relación que referencia) que tenga un atributo (o conjunto de atributos) asociado con una clave primaria de otra relación (relación referenciada), debe tomar valor de entre los definidos para la clave primaria de la relación referenciada o bien tomar valor nulo. Al descriptor de la relación que referencia se le denomina clave ajena.

IX - Reglas de integridad diferencial

Junto con las claves ajenas es necesario definir las reglas de integridad referencial, es decir, el comportamiento deseado cuando se produce una eliminación o modificación de tuplas relacionadas a través de una restricción de clave ajena.

¿Qué sucede cuando se produce una operación de borrado o modificación sobre las tuplas de la base de datos y exista integridad referencial?

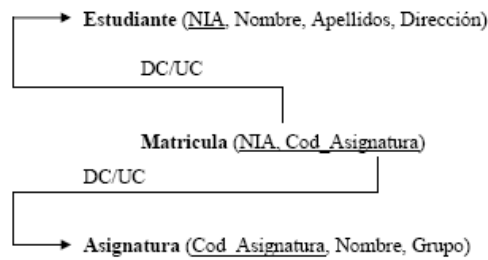


Las dos operaciones que debemos analizar son por lo tanto el borrado y la modificación.

Las reglas u opciones de integridad referencial son las siguientes :

1. Operación restringida (NO ACTION). Si la operación rompe la restricción referencial, no se lleva a cabo. Se abrevia NA o R (Restrict).
2. Operación con transmisión en cascada (CASCADE). Los valores afectados serán actualizados también en la clave ajena. Se abrevia C.
3. Operación con puesta a nulo (SET NULL). Los valores afectados serán sustituidos por el valor nulo. Se abrevia SN.
4. Operación con puesta a valor por defecto (SET DEFAULT). Los valores afectados serán sustituidos por un valor por defecto. Se abrevia SD.

Ejemplo



Ejemplo

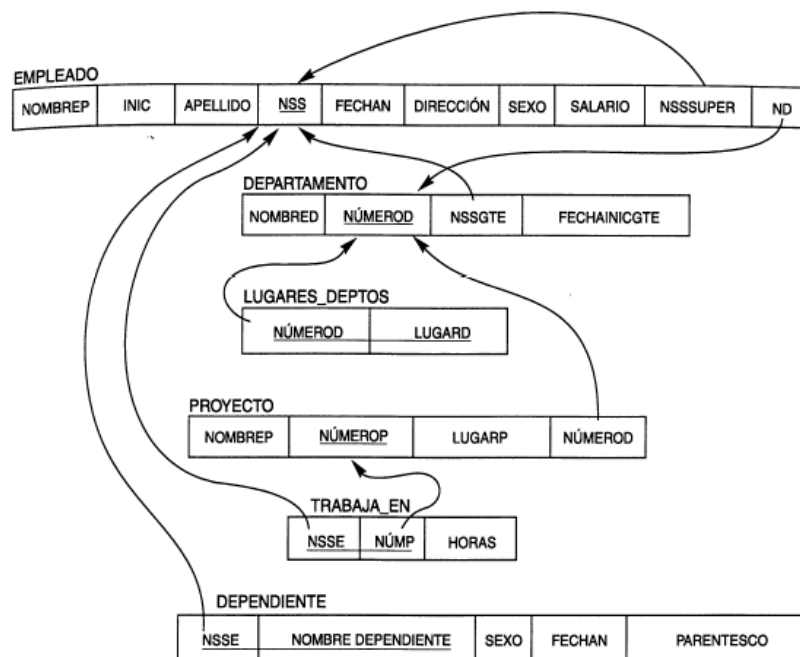
```
CREATE TABLE editorial (
  nombre_e CHAR(20) PRIMARY KEY,
  dirección CHAR(50) NOT NULL,
  ciudad CHAR (15),
  país CHAR(15);
```

Ejemplo

```
CREATE TABLE libro (
  código CHAR(3),
  título CHAR (50) UNIQUE,
  idioma CHAR(25),
  nombre_e CHAR(20),
  PRIMARY KEY (código),
  FOREIGN KEY (nombre_e) REFERENCES editorial
  ON DELETE SET NULL
  ON UPDATE CASCADE);
```

Ejercicio 1

En la siguiente base de datos comprobar si podremos realizar las operaciones descritas. En caso de no poder realizarlas, indicar porque.



Operaciones de insertar

1. Insertar <'Cecilia', 'F', 'Laguardia', '677678989', '05-ABR-50', 'Calle Viento 6357, Malinalco, MX', F, 28000, nulo, 4> en EMPLEADO.
— Esta inserción satisface todas las restricciones, así que es aceptable.
2. Insertar <'Alicia', 'J', 'Zapata', '999887777', '05-ABR-50', 'Calle Viento 6357, Malinalco, MX', F, 28000, '987654321', 4> en EMPLEADO.
— Esta inserción viola la restricción de clave porque ya existe otra tupla con el mismo valor de NSS en la relación EMPLEADO.
3. Insertar <'Cecilia', 'F', 'Laguardia', nulo, '05-ABR-50', 'Calle Viento 6357, Malinalco, MX', F, 28000, nulo, 4> en EMPLEADO.
— Esta inserción viola la restricción de integridad de entidades (nulo en la clave primaria NSS), de modo que no es aceptable.
4. Insertar <'Cecilia', 'F', 'Laguardia', '677678989', '05-ABR-50', 'Calle Viento 6357, Malinalco, MX', F, 28000, '987654321', 7> en EMPLEADO.
— Esta inserción viola la restricción de integridad referencial especificada sobre ND porque no existe ninguna tupla de DEPARTAMENTO en la que NÚMEROD = 7.

Operaciones de eliminar

1. Eliminar la tupla TRABAJA_EN con NSSE = '999887777' y NÚMP = 10.
— Esta eliminación es aceptable.
2. Eliminar la tupla EMPLEADO con NSS = '999887777'.
— Esta eliminación no es aceptable porque dos tuplas de TRABAJA_EN hacen referencia a esta tupla. Por tanto, si se elimina la tupla, se violará la integridad referencial.
3. Eliminar la tupla EMPLEADO con NSS = '333445555'.
— Esta eliminación producirá violaciones a la integridad referencial aún más graves, porque tuplas de las relaciones EMPLEADO, DEPARTAMENTO, TRABAJA_EN y DEPENDIENTE hacen referencia a la tupla en cuestión.

Operaciones de modificar

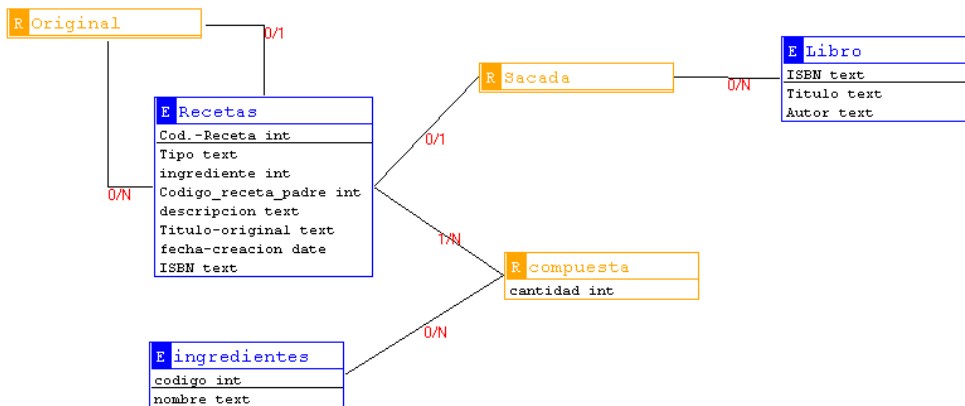
1. Modificar el SALARIO de la tupla EMPLEADO con NSS = '999887777' cambiándolo a 28000.
— Aceptable.
2. Modificar el ND de la tupla EMPLEADO con NSS = '999887777' cambiándolo a 1.
— Aceptable.
3. Modificar el ND de la tupla EMPLEADO con NSS = '999887777' cambiándolo a 7.
— Inaceptable, porque viola la integridad referencial.

Ejemplo

Tanto ha sido su interés que en la actualidad posee un gran número de recetas nuevas a modo de apuntes, imposible de organizar de una manera efectiva.

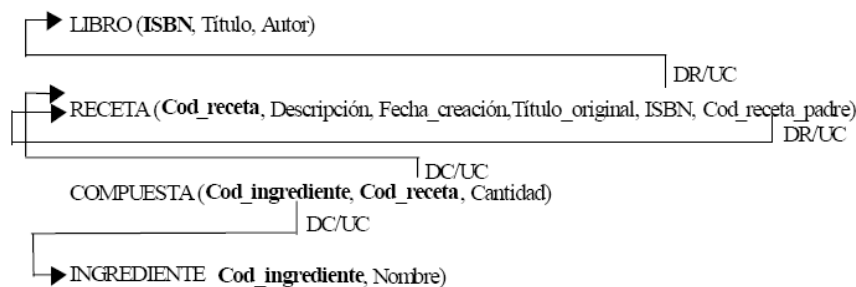
Los amigos de Antonio, en agradecimiento a tantas veladas de buena comida, vino y compañía han decidido agradecerle con una base de datos que le gestione su maravilloso gran hobby. Para ello deben controlar todas las recetas que posee, teniendo en cuenta que:

- Cada receta proviene de una idea original de un libro de cocina de la biblioteca de Antonio, y se desea almacenar su origen.
- Cada receta tiene un tipo (Sopas, Verduras, Carnes,...) e incorpora unos ingredientes de los que se desea saber su nombre y cantidad.
- Además cada receta contiene una breve explicación de cómo mezclar los ingredientes y obtener el producto final y el título de la receta original de la que proviene.
- Es bien sabido por los amigos de Antonio que de cada receta que él prueba, incorporando ciertos cambios, consigue nuevas recetas, por lo que sería interesante almacenar si cada receta es idea surgida de una receta original, o si por el contrario, proviene de una receta elaborada alguna vez ya por Antonio.
- Lo que los amigos de Antonio quieren es que cuando él quiera pueda consultar las recetas por tipo y por ingrediente además de poder localizar el libro que le dio la idea de cada una de sus recetas. También sería interesante saber qué receta proviene de alguna otra y cual no.



El diagrama E/R sería (creado mediante el programa gerwin, cuidado porque las cardinalidades están colocadas en sentido contrario a las vistas en clase):

El esquema relacional sería el siguiente :



¿Explicar las restricciones semánticas y las restricciones de integridad referencial?