Modelo Relacional

Introducción

- El modelo Relacional forma parte del Diseño Lógico de una base de datos, cuyo objetivo es transformar el esquema conceptual obtenido en la etapa anterior con el Modelo E/R, al Modelo de Datos Relacional en el que se apoya el SGBD que se va a utilizar.

Modelización del mundo real:

- Esquema Conceptual: Sería una 1ª aproximación del mundo real. Se utiliza el modelo E/R que sería como un pseudocódigo en programación.
- Modelo Lógico: Lo equiparamos a implementarlo en algún lenguaje de programación.

Tablas

Los Elementos del modelo relacional son los siguientes:

1. **Relación:** (Así se denominan las tablas en el modelo relacional)

   Se representa mediante una **tabla**, que es una estructura de datos que representa una entidad o una asociación de entidades. Una relación está formada por un conjunto de columnas llamados **atributos** y un conjunto de filas llamadas **tuplas**.

2. **Atributo**: es cada una de las columnas de la tabla. Las columnas tienen un nombre y pueden guardar un conjunto de valores. Una columna se identifica siempre por su nombre, nunca por su posición.
3. **Tupla**: representa una fila de la tabla. En una tabla no se admiten filas duplicadas.

Características que deben cumplir las tablas son:

- Una tabla no pude tener ni filas ni columnas repetidas.
- El orden de filas y columnas es irrelevante
- Las tablas deben ser planas, es decir, la intersección de una fila y una columna debe ser un único valor.
- Los valores de una columna tiene que estar dentro del mismo dominio

Ejemplo de tabla

TABLA EMPLEADO

NIF	NOMBRE	SALARIO	NUMDEP	FECHA_ALTA
13407877B	Carmen Hortal Ortega	1200	10	15/10/2003
41667891C	Juan Caballero Perea	1200	20	29/05/2002

•**Cardinalidad**: Es el número de filas de la tabla.
En el ejemplo anterior es dos.
•**Grado**: es el número de columnas de la tabla.
En el ejemplo anterior es cinco.
•**Valor**: viene representado por la intersección entre una fila y una columna. Por ejemplo, son valores de la tabla EMPLEADO 13407877B y Carmen Hortal Ortega.
•**Valor nulo**: representa la ausencia de información

Dominios

- Es el conjunto de valores que puede tomar cada atributo. Los valores contenidos en una columna pertenecen a un dominio que previamente se ha definido. Existen dos tipos de dominios:

- **Dominios generales**: son aquéllos cuyos valores están comprendidos entre un máximo y un mínimo. Un ejemplo es SALARIO, que está formado por números enteros positivos mínimo de 3 cifras.
- **Dominios restringidos**: son los que pertenecen a un conjunto de valores específico. Por ejemplo, Sexo, que puede tomar los valores H o M.
- **Dominios compuestos**: Cuando está formado por varios dominios simples

Claves

- Toda fila debe estar asociada con una clave que permita identificarla.
- A veces la fila se puede identificar por un único atributo, pero otras veces es necesario recurrir a más de un atributo.

La clave debe cumplir dos requisitos:

- Identificación unívoca: en cada fila de la tabla el valor de la clave ha de identificarla de forma unívoca.
- No redundancia: no se puede descartar ningún atributo de la clave para identificar la fila.
- No pueden tener valores nulos
- Su valor no debe estar sujeto a cambios
- Las claves dan lugar a índices pues las tablas en última instancia se almacenan en ficheros indexados.

Tipos de Claves

- En una fila puede existir más de un conjunto de atributos que cumpla los requisitos anteriores. Este conjunto se conoce como **claves candidatas**.
- Uno de esos conjuntos será elegido como **clave primaria** que identificará la fila.
- En el modelo relacional existen varios modelos de claves: superclave o identificador principal, candidatas, primarias, ajenas y artificiales.
- **Superclave o identificador principal**: es el conjunto de columnas que identifican de forma única a cada ocurrencia de la tabla. Al menos hay una superclave, la formada por todos las columnas.
- **Clave candidata**: superclave mínima.
- **Clave primaria**: o *primary key*. Es la clave candidata seleccionada por el diseñador de la Base de Datos para identificar cada ocurrencia de la tabla, siendo la columna o conjunto de columnas que permiten identificar cada una de las ocurrencias de la tabla. No puede tener valores nulos. Ha de ser sencilla, fácil de crear y no variar con el tiempo. Ejemplo: NIF. Las claves candidatas no seleccionadas se llaman claves alternativas.
- **Clave ajena**: o *foreign key*. También se la conoce como clave extranjera o foránea. Es el conjunto de columnas de una tabla que forman la clave primaria en otra tabla. Ej. NUMDEP.
- **Clave artificial**: columna creada por el equipo de diseño de la Base de Datos, cuyo contenido es aleatorio o secuencial, no repetitivo. Sirve para crear claves primarias más sencillas que una formada por la unión de varias columnas. Ha de ser única e invariable. Suele tener nombres como código, identificador, número de, etc.

Vistas

- Es una tabla ficticia cuya definición se obtiene a partir de otra u otras tablas, es decir, son vistas parciales de determinadas tablas.
- Por cada usuario tendremos una vista particular de la tabla
- **Características de las vistas**

  - La vista se define para dar una mayor seguridad del sistema o por simplicidad
  - Se pueden definir vistas sobre vistas
  - Aunque aumente el número de filas y columnas de la tabla la vista no siempre queda afectada.
  - No existe almacenamiento físico de las vistas. Únicamente se almacena su definición.

Atributos

- Atributo: Si el atributo tiene un número limitado y no muy elevado de ocurrencias forma parte de la entidad. Ejemplo: Dirección que es un atributo de empleados.
- Atributo multivaluado: En caso contrario, cuando el concepto que representa está relacionado con otras entidades podría ser una entidad. Ejemplo: Campus con profesor no interesa que sea solo un atributo, pues tiene relación con departamento , aulas,etc.
- A veces teléfono podría ser una entidad

Reglas de Codd

- En el año 1985 Codd publicó un artículo en la revista *Computerworld* en el que establecía 12 reglas que debe cumplir cualquier base de datos para que sea relacional.
- Son las siguientes:

1. **Regla de información**. Toda la información está representada a nivel lógico mediante valores en tablas.
2. **Regla de acceso garantizado**. Se puede acceder lógicamente a cada dato mediante la combinación del nombre de la tabla que lo contiene, el nombre de la columna y el valor de la clave primaria de la fila donde está.

3. **Tratamiento sistemático de valores nulos**. Un SGBDR (Sistema Gestor de Base de Datos Relacional) ha de soportar valores nulos para representar la ausencia de información de forma automática, independientemente del tipo de dato.
4. **Catálogo activo on-line basado en el modelo relacional**. La descripción de la base de datos se representa a nivel lógico como datos ordinarios, de forma que un usuario autorizado pueda utilizar el mismo lenguaje relacional para manejar sus datos y para consultar el catálogo
5. **Regla de sublenguaje completo de datos**. Un SGBDR tiene que soportar, al menos, un lenguaje cuyas sentencias tengan una sintaxis bien definida y tenga las operaciones siguientes:
   - Definición de datos
   - Definición de vistas
   - Manejo de datos (interactivo o por programa)
   - Restricciones de integridad
   - Autorizaciones
   - Gestión de transacciones

6. **Regla de actualización de vistas**. Las vistas teóricas actualizables son también actualizables por el sistema.
7. **Inserción, modificación y borrado de alto nivel**. La posibilidad de manejar una relación de la base de datos con un único operando no sólo se aplica a la recuperación de datos, sino también a la inserción, la modificación y el borrado de éstos.
8. **Independencia física de los datos**. Los cambios en la forma de almacenar los datos o en los métodos de acceso no tienen que afectar a la manipulación de éstos
9. **Independencia lógica de los datos**. Los cambios sobre los objetos de la base de datos no deben afectar a los programas y a los usuarios que acceden a esos objetos.
10. **Independencia de integridad**. Las restricciones de integridad se han de poder definir con el sublenguaje de datos relacional y almacenarse en el catálogo, no en los programas.
11. **Independencia de distribución**. El SGBD tiene un sublenguaje que permite que los programas no se alteren cuando se introduce la distribución de los datos o cuando se redistribuyen.

Traducción del modelo Entidad-Relación al modelo Relacional

- El paso de un esquema en el modelo E/R al relacional está basado en los tres principios siguientes:
  - Todo tipo de entidad se convierte en una relación.
  - Todo tipo de relación N:M se transforma en una relación.
  - Todo tipo de relación 1:N se traduce en el fenómeno de propagación de clave o se crea una nueva relación.

1) Transformación de entidades:

- Cada tipo de entidad se debe convertir a una relación, es decir, será necesario crear una tabla para cada entidad que aparezca en el diagrama E/R.

2) Transformación de atributos de entidades

- Cada atributo de una entidad se debe transformar en una columna en la relación a la que ha dado lugar la entidad
- Puesto que hay diferentes tipos de atributos, hay que indicar cómo se deben definir cada uno de ellos:
- El o los atributos principales de una entidad (es decir, los que actúan como identificador único para los elementos de la relación, subrayados en el esquema) pasan a ser la clave primaria de la relación. Se debe especificar que no son nulos.
- El resto de atributos pasan a ser columnas de la tabla, pudiendo tomar valores nulos, a no ser que se indique lo contrario.
- Los atributos multivaluados dan lugar a una nueva relación cuya clave primaria es la concatenación de la clave primaria de la Entidad en la que se sitúa el atributo más el nombre del atributo multivaluado.
- Ejemplo: Entidad recambio tiene un atributo proveedor. Se crea la relación recambio y la relación proveedor, con clave Cod_recambio y Cod_proveedor.

3) Transformación de relaciones

- Dependiendo del tipo de relación (cardinalidad) de que se trate, existen diversas formas de transformarlas:
- Relaciones N:M. Se transforman en una relación que tendrá como clave primaria la concatenación de los atributos principales de cada una de las entidades que relacionan. Estos atributos deben ser clave ajena respecto a cada una de las tablas donde ese atributo es clave primaria. Habrá que considerar lo que ocurre en los casos en los que se borre o modifique la clave primaria referenciada. Será necesario crear las restricciones de cardinalidad adecuadas a través de la sentencia CHECK de SQL.

- AUTOR (Cod-Autor...)
- LIBRO (Cod-Libro...)
- ESCRIBE (Cod-Autor, Cod-Libro...)

- Relaciones 1:N. Existen dos soluciones para transformarlas. Habrá que considerar en ambos casos las referencias ajenas, y las actuaciones en caso de borrado y modificación.
  - *Propagar* el atributo principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la relación como tal. Esta clave no permite nulos.

    Ejemplo: LIBROS (Cod-Libro, Cod-editorial...) Cod-editorial no permite nulos.

  - *Transformarla* en una relación como si fuese una de tipo N:M. Esta acción sólo es recomendable en los casos en que:

    1) cuando es posible que aparezcan muchos nulos porque existen pocos elementos relacionados,

    2) cuando se prevé que dicha relación pasará en un futuro a ser de tipo N:M,

    3) cuando la relación tiene atributos propios.

- Relaciones 1:1. Puesto que se trata de una particularización de cualquiera de los dos casos anteriores, se puede del mismo modo aplicar cualquiera de las dos reglas definidas. No obstante, es recomendable seguir las siguientes reglas:
  - Si las entidades que se asocian poseen cardinalidades (0,1) (0,1), es mejor crear una relación para evitar el tener muchos nulos como propagación de alguna de las claves a la otra relación (si se prevé que puede haber muchos nulos)
  - Si las entidades que se asocian poseen cardinalidades (0,1)(1,1), es mejor propagar la clave de la entidad (1,1) a la (0,1)
  - Si las entidades que se asocian poseen cardinalidades (1,1)(1,1) la propagación es indiferente, y se hará atendiendo a los criterios de frecuencia de acceso (consulta, modificación, inserción, etc.) a cada una de las tablas en cuestión.

4) Transformación de atributos de relaciones:

Si la relación se transforma en una tabla, todos sus atributos pasan a ser columnas de la tabla. En el caso en que alguno de los atributos de la relación sea principal, deberá ser incluido como parte de la clave primaria en dicha tabla.

5) Transformación de relaciones exclusivas:

Para soportar relaciones exclusivas debemos definir las restricciones pertinentes en cada caso. Por ejemplo, en el caso en que exista una exclusividad en la edición de un libro por parte de una editorial o de una universidad, estas dos relaciones se resuelven mediante el mecanismo de propagación de la clave, llevando las claves primarias de editorial y universidad a libro. Se utilizará entonces la cláusula CHECK de SQL para introducir las restricciones pertinentes.

```
CHECK ( (Cod_editorial IS NULL AND Cod_univers IS NOT NULL) OR
(Cod_univers IS NULL AND Cod_editorial IS NOT NULL) );
```

6) Transformación de tipos y subtipos:

Existen varias posibilidades para su transformación:

- Englobar todos los atributos de una entidad y sus subtipos en una sola relación, añadiendo el atributo que permite distinguir los subtipos. También habrá que especificar las restricciones semánticas adecuadas.
- Crear una relación para el supertipo, y tantas relaciones como subtipos existan. Esta es la mejor opción desde el punto de vista semántico, pero es menos eficiente que la opción anterior.
- Crear sólo relaciones para los subtipos, añadiendo en cada una de ellas los atributos pertenecientes al supertipo.

7) Transformación de relaciones dependientes

- Una relación dependiente en existencia se transforma en propagación de clave, de la entidad regular a la entidad débil. La clave propagada no permite blancos
- Una relación dependiente en identidad se transforma igual que la anterior pero la clave de la entidad débil esta formada por la clave de la entidad débil más la clave de la entidad regular.

8) Transformación de atributos derivados.

- Podemos transformarlo como una columna más de la tabla, con un procedimiento que vaya asociado a dicha columna para los cálculos pertinentes
- Podemos no introducir el atributo como una columna y crear un procedimiento para crear el atributo derivado.

9) Transformación de atributos de relaciones

- Todo atributo va a pasar a ser una columna de la tabla en que se transforma la relación.
- Si hay propagación de clave, los atributos se propagan con la clave. Este caso es la minoría.

10) Transformación de restricciones.

- Las restricciones del modelo E/R se van a transformar en reglas.
- Las restricciones inherentes al modelo relacional son:
    - ○No pueden existir 2 tuplas iguales en una relación
    - ○No es relevante el orden de las tuplas ni el de los atributos
    - ○Cada atributo solo puede tomar un único valor del dominio
    - ○Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo
- Restricciones semánticas o de usuario:
    - ○Restricciones de PRIMARY KEY. Permite declarar un atributo o conjunto de atributos como clave primaria.
    - ○Restricción de Unicidad (UNIQUE) Nos permite definir claves alternativas
    - ○Restricción de obligatoriedad (NOT NULL) Permite declarar si uno o varios atributos tienen que tener siempre un valor.
    - ○Restricción de FOREIGN KEY también denominada Integridad referencial. Permite enlazar relaciones de una bbdd. La integridad referencial nos indica que los valores de la clave ajena en la relación hijo deben corresponderse con los valores de la clave primaria en la relación padre o bien ser nulo si se admiten nulos. Los atributos que son clave ajena en una relación no necesitan tener los mismos nombres que los atributos de la clave primaria

EMPLEADO (NOMBRE, DNI, SALARIO, FECHA_NAC, DPTO)

DEPARTAMENTO(NUM_DPTO, NOMBRE)

La Integridad referencial asegura que los empleados de la relación EMPLEADO solo pueden pertenecer a departamentos que hayan sido dados de alta en DEPARTAMENTOS:

El modelo relacional también permite definir opciones de borrado y modificación en las claves ajenas.

Estas opciones indican las acciones que hay que llevar a cabo cuando se produce un borrado o modificación de una tupla en la relación padre referenciada por una relación hija.

Las posibilidades son:

a)Borrado/modificación en cascada (B:C, M:C)
b)Borrado/modificación restringido: No permite borrar una tupla del padre cuando tiene tuplas asociadas en la relacion hija. (B:R, M:R)
c)Borrado/modificación con puesta a nulos (B:N, M:N)
d)Borrado/modificación con puesta a valor por defecto (B:D, M:D)

- Restricciones de verificación (CHECK): Para especificar las condiciones que debe cumplir un atributo (además de NOT NULL , clave ajena, clave primaria,etc) Estas restricciones se comprueban ante un borrado, modificación e inserción.
  Siempre van ligadas a un único valor de la bbdd (por ello no tiene un nombre)
  Ejemplo: Salario tiene un check para verificar entre 600 y 6000 euros.

- Aserciones (ASSERTION): Idéntico a los Check pero la condición se establece sobre elementos de distintas relaciones y los CHECK solo de una. Por este motivo las aserciones tiene un nombre. Ejemplo: No puede haber ningún empleado del departamento de contabilidad que gane más de 800 Euros.

- Disparadores (TRIGGER) Cuando quiero especificar una acción distinta al rechazo. Nos permiten además de especificar una condición, especificar la acción que queremos se lleve a cabo si la condición se cumple. Cuando se produce un evento, si se cumple un condición se ejecuta una acción. Ejemplo: Un disparador que avise cuando un empleado gane más de 1000 Euros.