

Bases de Datos

Modelo Relacional: Conceptos Básicos

Apunte elaborado por:

- MCs. Norma Herrera
- Lic. Darío Ruano
- Prof. Paola Azar

1 Introducción

El modelo relacional fue introducido por **Edgar F. Codd** en el **año 1970** en su histórico artículo de ACM “**A Relational Model of Data for Large Shared Data Banks**”. Hasta ese momento los SGBD estaban basados en los modelos de Red y Jerárquico, cuyo principal problema era que los programas de aplicación quedaban completamente atados a la representación física de los datos. El modelo relacional fue un hito en la historia de las bases de datos porque logró aislar a los programas de aplicación de la representación interna de los datos.

El modelo relacional estableció las bases para la tecnología actual de bases de datos. Su éxito se debió principalmente a su sencillez para especificar el esquema de la base de datos, su potencial y su base en una sólida fundamentación teórica: teoría de conjuntos y lógica de primer orden. Pero a pesar de esta fuerte base matemática, la estructura básica del modelo relacional es simple, lo que hace que sea sencillo de entender desde el punto de vista intuitivo.

En la sección 1 vimos que un dato es un valor con un significado asociado y que la información surge al relacionar dos o mas datos. En el modelo relacional **los datos se modelizan con atributos y la información con relaciones**.

1.1 Dominios, Atributos y Relaciones

Un **dominio** es un conjunto de valores atómicos. Por atómicos entenderemos que cada valor del dominio es indivisible y se trabaja como una unidad.

Un **atributo** modeliza un dato y se especifica dando su nombre y su dominio. El nombre del atributo representará el significado del dato modelizado y el dominio el conjunto de valores permitidos. En el ejemplo de las cabañas de la Sección 3, habíamos visto que teníamos los datos *código, nombre, capacidad y cantidad de habitaciones* de cada cabaña. Estos datos lo podemos modelizar de la siguiente manera:

<u>Dato</u>	<u>Nombre del Atributo</u>	<u>Dominio del Atributo</u>
Código de cabaña	CodCabaña	N
Nombre de Cabaña	NombreCabaña	Strings
Capacidad la cabaña	CapacidadCabaña	{1,...,10}
Cantidad de habitaciones de la cabaña	CantHabCabaña	{1, 2, 3, 4}

Un **esquema de relación** está formado por un conjunto de atributos que determina la constitución de la relación. Al conjunto de elementos que conforman la relación en un determinado instante del tiempo se lo denomina **instancia de la relación**. Cada elemento que conforma la instancia de una relación se denomina **nupla o tupla**¹. La instancia es lo que varía en el tiempo, el esquema no. El esquema de una relación sólo debe cambiar cuando cambia alguna restricción y/o dato de la realidad que estamos modelizando.

1 Este nombre tienen que ver con la definición matemática de una relación como producto cartesiano de varios(n) dominios.

Siguiendo con el ejemplo anterior, podemos modelizar la información que se necesita guardar de cada cabaña con una relación Cabañas cuyo esquema será el siguiente:

Cabañas = { CodCabaña, NombreCabaña, CapacidadCabaña, CantHabCabaña}
 $\text{dom}(\text{CodCabaña}) = \mathbb{N}$
 $\text{dom}(\text{NombreCabaña}) = \text{Strings}$
 $\text{dom}(\text{CapacidadCabaña}) = \{1, \dots, 10\}$
 $\text{dom}(\text{CantHabCabaña}) = \{1, 2, 3, 4\}$

Una posible instancia de la relación Cabañas podría ser:

CodCabaña	NombreCabaña	CapacidadCabaña	CantHabCabaña
1000	Sol	4	2
1001	Sierra	8	3
1003	Aire	4	2
1004	Tierra	6	2

La instancia de una relación se puede visualizar por medio de una tabla donde cada columna corresponde a un atributo del esquema de la relación y cada fila es una nupla de la relación. Por esta razón es frecuente utilizar los términos *relación* y *tabla* como sinónimos. De igual manera suele utilizarse el término *fila* para las *nuplas* y *columna* para los *atributos*. Notar que cada nupla (fila) relaciona datos por lo tanto cada nupla de la relación nos da información.

El esquema de relación es una descripción de la información que pretendemos almacenar en esa relación mientras que la instancia nos indica cuál es la información en una instante dado de tiempo. La instancia es lo que varía en el tiempo y el esquema no. El esquema solo debería modificarse cuando se produce un cambio en el UD.

Notar que²:

- No pueden existir dos nuplas iguales en una instancia de relación.
- El orden en que se den las nuplas de la instancia (filas de la tabla) no es significativo, dado que la información que esas nuplas representan no depende del orden en que se dan.
- El orden en que se den los atributos que forman el esquema de la relación (columnas de la tabla) tampoco es significativo (por la misma razón que el punto anterior)

1.2 Clave de una Relación

Dado que el orden en que se dan las nupla (filas) de una instancia de relación no importa, no podemos seleccionar una nupla específica por su posición. No existe “primera fila”, ni “última fila” ni “quinta fila” en una instancia de relación. Entonces, ¿cómo podemos especificar que necesitamos una nupla específica de una relación? en nuestro ejemplo de las cabañas, ¿cómo indico que quiero la nupla que me da la información correspondiente a la cabaña 1003?

2 Estas propiedades se derivan de la definición matemática de relaciones. Las relaciones son conjuntos de nuplas y en un conjunto no importa el orden en que se den sus elementos: $\{1, 5, 2\} = \{2, 1, 5\}$

En toda relación debe existir un atributo o conjunto de atributos cuyos valores identifiquen unívocamente las nuplas de la relación; a este conjunto de atributos se lo denomina **clave o identificador** de la relación. Una clave es un atributo o conjunto de atributos cuya combinación de valores no se pueden repetir en las nuplas de la instancia.

En el ejemplo de las cabañas, teníamos el siguiente esquema de relación:

Cabañas={CodCabaña, NombreCabaña, CapacidadCabaña, CantHabCabaña}

En el relato también especificaba lo siguiente: “....El complejo cuenta con un gran número de cabañas cada una de las cuales tiene asignado un código que es único....”. Esto nos indica que en ninguna instancia de Cabañas pueden existir dos nuplas con el mismo valor para CodCabaña. Por lo tanto CodCabaña es una clave de la relación.

Puede suceder que una relación tenga más de una clave. Por ejemplo, en el relato del complejo de cabañas habíamos visto que existía una ambigüedad con respecto al nombre de cabaña porque el relato no indicaba si los nombres de cabaña se podían o no repetir. Supongamos que el administrador del complejo de cabañas nos indica que los nombres de cabaña tampoco se pueden repetir, entonces el atributo NombreCabaña es otra clave de la relación. Esto significa que en ninguna instancia de Cabañas pueden existir dos nuplas (filas) con el mismo valor para CodCabaña y/o con el mismo valor para NombreCabaña. Las claves deben especificarse al momento de especificar el esquema de la relación. Para nuestro ejemplo nos quedaría el siguiente esquema:

Cabañas={CodCabaña, NombreCabaña, CapacidadCabaña, CantHabCabaña}

dom(CodCabaña)=|N

dom(NombreCabaña)=Strings

dom(CapacidadCabaña)={1,...,10}

dom (CantHabCabaña)={1, 2, 3, 4}

Clave 1: CodCabaña

Clave 2: NombreCabaña

Veamos otro caso. El relato del complejo de cabañas dado en la Sección 3 también nos indicaba:

De cada reserva realizada se registra la cabaña reservada, el cliente que hizo la reserva, la fecha de reserva y la cantidad de días de la estadía. Se permite que un cliente haga mas de una reserva, para la misma o para distintas fechas. A cada reserva se le asigna un código que es único.

Podemos modelizar esta parte con una relación cuyo esquema es el siguiente:

Reservas={CodReserva, CodCabaña, DNICliente, Fecha, CantDias}

dom(CodReserva)=|N

dom(CodCabaña)=|N

dom(DNICliente)=|N

dom(Fecha)= TipoFecha

dom (CantDias)= |N

Para encontrar la clave de esta relación debemos analizar el relato y a partir de él determinar cuál es el atributo o conjunto de atributos cuya combinación de valores no se puede repetir en ninguna instancia de la relación.³

Claramente `CodReserva` es una clave. Pero en este caso hay otra clave más. Si bien un cliente puede realizar mas de una reserva y una cabaña puede ser reservada para distintas fechas, por el mismo cliente o por clientes distintos, nunca debería pasar que se registren dos reservas de la misma cabaña para la misma fecha, por lo que otra clave de esta relación es `CodCabaña Fecha`.

`Reservas`={`CodReserva`, `CodCabaña`, `DNICliente`, `Fecha`, `CantDias`}

`dom(CodCabaña)`=|N

`dom(DNICliente)`=|N

`dom(Fecha)`= TipoFecha

`dom (CantDias)`= |N

Clave 1: `CodReserva`

Clave 2: `CodCabaña Fecha`

Resumiendo, una relación puede tener una o varias claves las que pueden ser simples (formadas por un solo atributo) o compuestas (formadas por varios atributos). Lo que nunca puede pasar es que una relación no tenga clave.

1.3 Base de Datos Relacional

Una base de datos relacional es un **conjunto de relaciones que representan una realidad dada**. Por lo tanto, para describir una base relacional debemos dar:

- El **esquema de la base de datos** formado por un conjunto de esquemas de relaciones.
- La **instancia de la base de datos** formado por un conjunto de instancias de relaciones. La instancia de la base de datos se puede visualizar como un conjunto de tablas: una por cada esquema de relación que conforma el esquema de la base de datos.

Para obtener el esquema de una base de datos relacional existen procesos de modelización que nos aseguran que el modelo relacional obtenido tiene propiedades que son deseables en un esquema de base de datos. Básicamente existen dos caminos:

- Usar la teoría de normalización propuesta por Edgar Codd.
- Diseñar primero el modelo entidad-relación y luego realizar el pasaje al modelo relacional. En este caso se debe verifica que el modelo relacional obtenido cumple con ciertas propiedades que son deseables de una base de datos.

Este proceso de modelado lo estudiaremos mas adelante.

3 Existe un método formal para determinar claves de una relación haciendo uso de dependencias funcionales entre los datos. No lo estudiaremos en este curso.

1.4 Claves Foráneas o Foreign Key

En una base de dato muchas veces sucede que un atributo de una relación coincide en su significado con otro atributo que es clave en otra relación. En nuestro ejemplo de las cabañas, habíamos llegado al siguiente modelo relacional:

Cabañas={CodCabaña, NombreCabaña, CapacidadCabaña, CantHabCabaña}

dom(CodCabaña)=|N

dom(NombreCabaña)=Strings

dom(CapacidadCabaña)={1,...,10}

dom (CantHabCabaña)={1, 2, 3, 4}

Clave 1: CodCabaña

Clave 2: NombreCabaña

Cientes={DNICliente, NombreCliente, DirCliente, TECliente}

dom(DNI)=|N

dom(NombreCliente)= dom(DirCliente)=Strings

dom(TECliente)=|N

Clave: DNICliente

Reservas={CodReserva, CodCabaña, DNICliente, Fecha, CantDias}

dom(CodReserva)= dom(CodCabaña)=|N

dom(DNICliente)=|N

dom(Fecha)= TipoFecha

dom (CantDias)= |N

Clave 1: CodReserva

Clave 2: CodCabaña Fecha

En esta base de datos es razonable pedir que todos los valores que se carguen en la columna CodCabaña de la tabla Reservas, previamente se hayan cargado en la columna CodCabaña de la tabla Cabañas porque sino estaríamos registrando una reserva de una cabaña que no existe. De la misma manera los valores que se carguen en la columna DNICliente de la tabla Reservas, previamente deben haber sido cargados en la columna DNICliente de la tabla Cientes porque sino estaríamos registrando la reserva de un cliente que no existe en nuestra base de datos.

Estas restricciones se conocen con el nombre de **restricciones de integridad referencial** y se especifican en el modelo con la **cláusula FK** (por su siglas en inglés Foreign Key).

En nuestro modelo la relación Reservas quedaría de la siguiente manera:

$\text{Reservas} = \{\text{CodReserva}, \text{CodCabaña}, \text{DNICliente}, \text{Fecha}, \text{CantDias}\}$
 $\text{dom}(\text{CodReserva}) = \text{dom}(\text{CodCabaña}) = \mathbb{N}$
 $\text{dom}(\text{DNICliente}) = \mathbb{N}$
 $\text{dom}(\text{Fecha}) = \text{TipoFecha}$
 $\text{dom}(\text{CantDias}) = \mathbb{N}$
Clave 1: CodReserva
Clave 2: CodCabaña Fecha
 $\text{FK}(\text{CodCabaña}) \rightarrow \text{Cabañas}(\text{CodCabaña})$
 $\text{FK}(\text{DNICliente}) \rightarrow \text{Clientes}(\text{DNICliente})$

Todas las restricciones de un esquema de relación se determinan durante el proceso de modelado de la base de datos.



Bibliografía:

Fundamentos de Sistemas de Bases de Datos, 3a edición. R. Elmasri , S. Navathe.

Fundamentos de Bases de Datos, 4ta edición. A. Silberschatz H. F. Korth, S. Sudarshan.