



Unidad III Modelo relacional

- 3.1 Introducción al modelo relacional
- 3.2 Conversión de Modelo E-R a Modelo relacional
- 3.3 Esquema de la base de datos
- 3.4 Restricciones
 - 3.3.1 Integridad de entidad
 - 3.3.2 Integridad referencial
- 3.5 Integridad de dominio

3.1 Introducción al modelo relacional.- Modelo de organización y gestión de bases de datos consistente en el almacenamiento de datos en **tablas** compuestas por filas, renglones o **tuplas**, y columnas o **campos**. Se distingue de otros modelos, como el jerárquico, por ser más comprensible para el usuario inexperto, y por basarse en la lógica de predicados para establecer relaciones entre distintos datos, podemos resumir el concepto como una colección de **tablas** (relaciones). El objetivo del modelo relacional es crear un "esquema" (*schema*), lo cual consiste de un conjunto de "tablas" que representan "relaciones", relaciones entre los datos.

Y que es la Tabla: es el nombre que recibe cada una de las relaciones que se establecen entre los datos almacenados; cada nueva relación da lugar a una tabla. Están formadas por **filas**, también llamadas tuplas, donde se describen los elementos que configuran la tabla (es decir, los elementos de la relación establecida por la tabla), **columnas** o campos, con los atributos y valores correspondientes, y el **dominio**, concepto que agrupa a todos los valores que pueden figurar en cada columna.

Estas tablas, pueden ser construidas de diversas maneras:

- Creando un conjunto de tablas iniciales y aplicar operaciones de normalización hasta conseguir el esquema más óptimo.
- Convertir el diagrama e-r a tablas y posteriormente aplicar también operaciones de normalización hasta conseguir el esquema óptimo.

La primera técnica fue de las primeras en existir y, como es de suponerse, la segunda al ser más reciente es mucho más conveniente en varios aspectos:

- El partir de un diagrama visual es muy útil para apreciar los detalles, de ahí que se llame modelo conceptual.
- El crear las tablas iniciales es mucho más simple a través de las reglas de conversión.



- Se podría pensar que es lo mismo porque finalmente hay que "normalizar" las tablas de todas formas, pero la ventaja de partir del modelo e-r es que la "normalización" es mínima por lo general.
- Lo anterior tiene otra ventaja, aun cuando se normalice de manera deficiente, se garantiza un esquema aceptable, en la primera técnica no es así.

Filas o tuplas.- Cada uno de los renglones en una relación conteniendo valores para cada uno de los atributos.

Columnas, atributos o campos.- Los atributos son las columnas de un relación y describen características particulares de ella.

Dominio.- Se debe considerar que cada atributo (columna) debe ser atómico, es decir, que no sea divisible, no se puede pensar en un atributo como un "registro" o "estructura" de datos.

El diseño de una base de datos se realiza a dos niveles. El primero es el **nivel conceptual**, en la cual se contempla una estructura abstracta **Esquema**: diseño general de la base de datos a nivel lógico. Incluye el tipo de datos y las relaciones entre ellos. Es de naturaleza fija y solo se altera excepcionalmente. El esquema se define y se mantiene utilizando el lenguaje de definición de datos (DDL) Lenguaje de Definición de Datos.

El segundo es el **nivel físico**, en el cual la base de datos es ya implementable, es aquí cuando tenemos la **Instancia**: contenido concreto de la base de datos en un momento dado. Varía con el tiempo, al añadir, eliminar o modificar datos, utilizando el lenguaje de manipulación de datos (DML) Lenguaje de Manipulación de Datos.

Detalladamente, las fases del diseño de una base de datos son las siguientes:

Metodología de diseño conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual. Normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información. Cada una de estas visiones suelen corresponder a las diferentes áreas funcionales de la empresa como, por ejemplo, producción,



ventas, recursos humanos, etc. A los esquemas conceptuales correspondientes a cada vista de usuario se les denomina *esquemas conceptuales locales*.

Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores.

El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo.

Las tareas a realizar en el diseño conceptual son las siguientes:

- 1.- Identificar las entidades.**
- 2.- Identificar las relaciones.**
- 3.- Identificar los atributos y asociarlos a entidades y relaciones.**
- 4.- Determinar los dominios de los atributos.**
- 5.- Determinar los identificadores.**
- 6.- Dibujar el diagrama entidad-relación.**
- 7.- Revisar el esquema conceptual local con el usuario.**

1. Identificar las entidades. En primer lugar hay que definir los principales objetos que interesan al usuario. Estos objetos serán las entidades. Una forma de identificar las entidades es examinar las especificaciones de requisitos de usuario, en estas especificaciones se buscan los nombres nominales que se mencionan.

Por ejemplo: número de empleado, nombre de empleado, número de inmueble, dirección del inmueble, alquiler, número de habitaciones).

También se buscan objetos importantes como personas, lugares o conceptos de interés, excluyendo aquellos nombres que sólo son propiedades de otros objetos. Por ejemplo, se pueden agrupar el número de empleado y el nombre de empleado en una entidad denominada *empleado*, y agrupar número de inmueble, dirección del inmueble, alquiler y número de habitaciones en otra entidad denominada *inmueble*.



Otra forma de identificar las entidades es buscar aquellos objetos que existen por sí mismos. Por ejemplo, *empleado* es una entidad porque los empleados existen, sepamos o no sus nombres, direcciones y teléfonos. Siempre que sea posible, el usuario debe colaborar en la identificación de las entidades.

No siempre es obvio saber si un objeto es una entidad, una relación o un atributo. Por ejemplo ¿cómo se podría clasificar *matrimonio*?

Conforme se van identificando las entidades, se les dan nombres que tengan un significado y que sean obvias para el usuario. Los nombres de las entidades y sus descripciones se anotan en el diccionario de datos.

2. Identificar las relaciones. Una vez definidas las entidades, se deben definir las relaciones existentes entre ellas.

Del mismo modo que para identificar las entidades se buscaban nombres en las especificaciones de requisitos, para identificar las relaciones se suelen buscar las expresiones verbales (por ejemplo: oficina tiene empleados, empleado gestiona inmueble, cliente visita inmueble). Conforme se van identificando las relaciones, se les van asignando nombres que tengan significado para el usuario y se anotan en el diccionario de datos los nombres de las relaciones, su descripción y las cardinalidades con las que participan las entidades en ellas.

La mayoría de las relaciones son binarias (entre dos entidades), pero no hay que olvidar que también puede haber relaciones en las que participen más de dos entidades.

Es muy importante repasar las especificaciones para comprobar que todas las relaciones explícitas o implícitas, se han encontrado. Si se tienen pocas entidades, se puede comprobar por parejas si hay alguna relación entre ellas. De todos modos, las relaciones que no se identifican ahora se suelen encontrar cuando se valida el esquema con las transacciones que debe soportar.

3. Identificar los atributos y asociarlos a entidades y relaciones. Son atributos los nombres que identifican propiedades, cualidades, identificadores o características de entidades o relaciones. Lo más sencillo es preguntarse, para cada entidad y cada relación,



¿Que quiero se quiere saber de ...?

La respuesta a esta pregunta se debe encontrar en las especificaciones de requisitos. Al identificar los atributos, hay que tener en cuenta si son simples o compuestos. Por ejemplo, el atributo *dirección* puede ser simple, teniendo la dirección completa como un solo valor: 'San Rafael 45, Colonia 2 de Abril'; o puede ser un atributo compuesto, formado por la *calle* ('San Rafael'), el *número* ('45') y la *población* ('Colonia 2 de Abril').

El escoger entre atributo simple o compuesto depende de los requisitos del usuario. Si el usuario no necesita acceder a cada uno de los componentes de la dirección por separado, se puede representar como un atributo simple. Pero si el usuario quiere acceder a los componentes de forma individual, entonces se debe representar como un atributo compuesto.

Es muy útil elaborar una lista de atributos e ir eliminándolos de la lista conforme se vayan asociando a una entidad o relación. De este modo, uno se puede asegurar que cada atributo se asocia a una sola entidad o relación, y que cuando la lista se ha acabado, se han asociado todos los atributos.

Conforme se van identificando los atributos, se les asignan nombres que tengan significado para el usuario. De cada atributo se debe anotar la siguiente información:

- a) .- Nombre y descripción del atributo.
- b) .- Alias o sinónimos por los que se conoce al atributo.
- c) .- Tipo de dato y longitud.
- d) .- Valores por defecto del atributo (si se especifican).
- e) .- Si el atributo siempre va a tener un valor (si admite o no nulos).
- f) .- Si el atributo es compuesto y, en su caso, qué atributos simples lo forman.
- g) .- Si el atributo es derivado y, en su caso, cómo se calcula su valor.

4. Determinar los dominios de los atributos. El dominio de un atributo es el conjunto de valores que puede tomar el atributo. Por ejemplo el dominio de los números de oficina son las tiras de hasta tres caracteres en donde el primero es una letra y el siguiente o los dos



siguientes son dígitos en el rango de 1 a 99; el dominio de los números de teléfono y los números de fax son las tiras de 9 dígitos.

Un esquema conceptual está completo si incluye los dominios de cada atributo: los valores permitidos para cada atributo, su tamaño y su formato. También se puede incluir información adicional sobre los dominios como, por ejemplo, las operaciones que se pueden realizar sobre cada atributo, qué atributos pueden compararse entre sí o qué atributos pueden combinarse con otros.

Aunque sería muy interesante que el sistema final respetara todas estas indicaciones sobre los dominios, esto es todavía una línea abierta de investigación.

Toda la información sobre los dominios se debe anotar también en el diccionario de datos.

5. Determinar los identificadores. Cada entidad tiene al menos un identificador. Los identificadores pueden ser simples o compuestos. De cada entidad se escogerá uno de los identificadores como clave primaria en la fase del diseño lógico.

Cuando se determinan los identificadores es fácil darse cuenta si una entidad es fuerte o débil. Si una entidad tiene al menos un identificador, es *fuerte* (otras denominaciones son *padre*, *propietaria* o *dominante*).

Si una entidad no tiene atributos que le sirvan de identificador, es *débil* (otras denominaciones son *hijo*, *dependiente* o *subordinada*).

Todos los identificadores de las entidades se deben anotar en el diccionario de datos.

6. Dibujar el modelo relacional. Una vez identificados todos los conceptos, se puede dibujar el modelo relacional correspondiente a una de las vistas de los usuarios. Se obtiene así un esquema conceptual local.

7. Revisar el esquema conceptual local con el usuario

Antes de dar por finalizada la fase del diseño conceptual, se debe revisar el esquema conceptual local con el usuario.



Este esquema está formado por el diagrama entidad-relación y toda la documentación que describe el esquema. Si se encuentra alguna anomalía, hay que corregirla haciendo los cambios oportunos, por lo que posiblemente haya que repetir alguno de los pasos anteriores. Este proceso debe repetirse hasta que se esté seguro de que el esquema conceptual es una fiel representación de la parte de la organización que se está tratando de modelar.

Fuentes de información:

Korth, H. y Silbertchatz, A. Fundamentos de Bases de datos. (5ª ed.). Ed. McGraw Hill.

http://basededatos.umh.es/e_r.htm#entidades_relaciones

<http://www.marcoteorico.com/curso/31/fundamentos-de-bases-de-datos/416/diagramas-e-r->

<https://blog.es.logicalis.com/analytics/conceptos-basicos-del-modelo-relacional-en-la-gestion-de-bases-de-datos>