

Bases de datos relacionales

Hay distintos modelos de datos, uno de ellos es el modelo relacional, que da lugar a las bases de datos relacionales.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Sistema de gestión de base de datos

Los sistemas de gestión de base de datos (SGBD); (en inglés: Database management system, abreviado DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Existen distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias. Consistencia. En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Elementos principales

- Tablas: Son cada uno de los conjuntos de datos que corresponden a una entidad.
- Registros: Cada uno de los elementos individuales de la tabla (filas).
- Campos: Cada dato individual de un registro (columnas)

- **Restricción:** Es una limitación que se impone, obligando a que se cumplan ciertas condiciones en la Base de Datos.
- **Clave primaria:** Es un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. No puede haber dos registros en una tabla que tengan claves primarias idénticas.
- **Clave foránea:** es una referencia a una clave en otra tabla, y determina la relación existente entre dos tablas. Las claves foráneas no necesitan ser claves únicas en la tabla en donde están sino en aquella tabla a la que hacen referencia.

Diagrama Entidad-Relación (DER)

Entidades

El término entidad, cuando se le refiere en base de datos, es cualquier objeto sobre el que se tiene información. Una entidad está descrita por sus características. Por ejemplo, la entidad Persona lleva consigo las características de: Nombre, Apellido, Género, Estatura, Peso, Fecha de nacimiento, etc.. Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un identificador. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, empleado, etc.

Relaciones

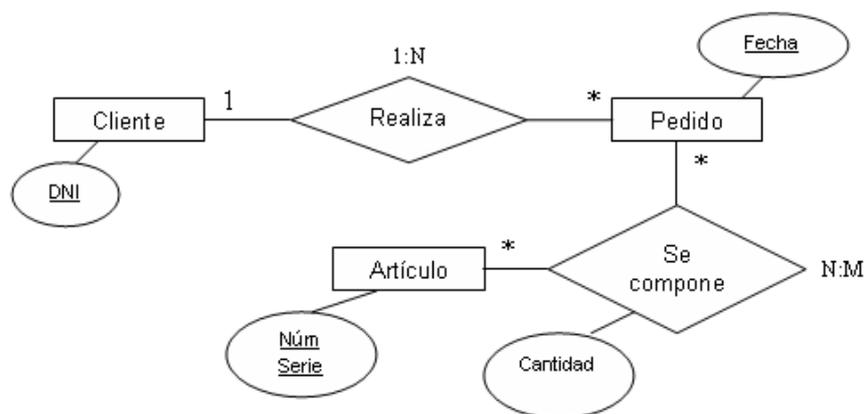
Una relación describe cierta dependencia entre entidades. Se representa mediante un rombo etiquetado en su interior con un verbo. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.

Una relación tiene sentido al expresar las entidades que relaciona. Por ejemplo: una persona (entidad) trabaja para (relación) un departamento (entidad).

Atributos

Los atributos son propiedades relevantes propias de una entidad y/o relación. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta. Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos.

Los atributos describen información útil sobre las entidades. En particular, los atributos identificativos son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta. Por ejemplo, el atributo identificativo que distingue a un empleado de otro es su número de CUIL.



Vemos que las relaciones

pueden ser, básicamente, de tres tipos:

- De cero a muchos o de uno a muchos: esto es, un registro de una entidad "A" se relaciona con muchos de otra entidad "B", pero cada registro de "B" se relaciona con uno solo (o ninguno) de la entidad "A". Ejemplo: Un cliente puede realizar muchos pedidos, pero un pedido corresponde a un único cliente.
- De muchos a muchos: Un registro de una entidad "A" se relaciona con muchos de otra entidad "B", y a su vez cada registro de "B" se relaciona con muchos de "A". Ejemplo: Un pedido se compone de muchos artículos, y a su vez un artículo puede formar parte de muchos pedidos.
- De de cero a uno: Un elemento de "A" se relaciona con cero o uno de "B", y a la vez uno de "B" se relaciona con uno de "A". Por ejemplo, una persona puede tener cero o un cónyuge, y este cónyuge no puede estar casado con nadie más. En muchas ocasiones cabe preguntarse si no es conveniente unificar ambas entidades en una sola.

Normalización de BD

Típicamente, se establece una tabla por cada entidad, y una tabla por cada relación en el caso de que ésta tenga atributos propios o sea “de muchos a muchos”.

Pero las tablas obtenidas de esta manera podrían no resultar óptimas, por lo que se las somete a una técnica llamada “normalización”.

Primera forma normal (1FN)

Una tabla está en Primera Forma Normal si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
- La tabla contiene una clave primaria única.
- La clave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Los Campos no clave deben identificarse por la clave (Dependencia Funcional)
- Debe Existir una independencia del orden tanto de las filas como de las columnas, es decir, si los datos cambian de orden no deben cambiar sus significados

Segunda forma normal (2FN)

Una tabla está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal).

El significado de “depender” en el párrafo anterior puede entenderse así: A depende de B si el hecho de conocer B es suficiente para saber A. Así, se puede decir que el nombre de una persona “depende” de su DNI, porque si se sabe su DNI se puede averiguar su nombre.

Tercera forma normal (3FN)

La tabla se encuentra en 3FN si está en 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Existen otras formas normales, que no abordaremos en la materia.

Integridad referencial

Consiste en asegurar que un campo definido como clave foránea (es decir, que referencia a la clave primaria de otra tabla), en ningún momento se encuentre referenciando a algo inexistente. En otras palabras, la integridad referencial asegura la consistencia de los datos.

Lo explicaremos con un ejemplo:

Supongamos que tenemos una tabla EMPLEADOS, y otra tabla DEPARTAMENTOS. Un empleado pertenece a un único departamento, y un departamento puede tener muchos empleados. Uno de los atributos de la tabla EMPLEADOS es el nro_departamento, que hace referencia a la clave primaria de DEPARTAMENTOS.

Ahora bien, supongamos que quiero eliminar un registro de DEPARTAMENTOS. Si ningún empleado pertenece a ese departamento, no hay problema. ¿Pero qué pasa si no es así? Básicamente, hay dos opciones:

a) RESTRINGIR: No se puede eliminar un departamento, hasta tanto no haya ningún empleado que pertenezca a él.

b) CASCADA: Al momento de eliminar el departamento, se eliminan todos los empleados que pertenezcan a él.

Algo similar sucede al momento de actualizar: Si el nro_departamento de “Ventas” es 8, y quiero modificarlo por 9, quedarán muchos empleados que figurarán como pertenecientes al departamento número 8, que ya no existe. Otra vez tenemos dos opciones:

a) RESTRINGIR: No se puede modificar el nro_departamento, hasta tanto no haya ningún empleado que pertenezca a él.

b) CASCADA: Al momento de modificar el nro_departamento, se modifica también el nro_departamento de todos los empleados que pertenezcan a él.

(Este apunte es un resumen del apunte confeccionado por la prof. Adriana Durán para la asignatura Programación II en 2010)