

INTRODUCCIÓN A LAS BASES DE DATOS

AUTOR: MAX JOSÉ BERMÚDEZ LEÓN
DICIEMBRE: 2020



Introducción

En el entorno del mercado actual, la competitividad y la rapidez de maniobra de una empresa son imprescindibles para su éxito. Para conseguirlo existe cada vez una mayor demanda de datos y, por tanto, más necesidad de gestionarlos. Esta demanda siempre ha estado patente en empresas y sociedades, pero en estos años se ha disparado debido al acceso multitudinario a las redes integradas en Internet y a la aparición de los dispositivos móviles que también requieren esa información.

En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para un usuario. Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la manipulación de los datos. Antes de la aparición de las aplicaciones informáticas, las empresas tenían como únicas herramientas de gestión de datos los ficheros con cajones, carpetas y fichas de cartón. En este proceso manual, el tiempo requerido para manipular estos datos era enorme. Pero la propia informática ha adaptado sus herramientas para que los elementos que el usuario utiliza en cuanto a manejo de datos se parezcan a los manuales. Por eso se sigue hablando de archivos, formularios, carpetas, directorios.

Se puede entender a una organización como un sistema de información formado por diversos objetos: el capital, los recursos humanos, los inmuebles, los servicios que presta, etc.

Los sistemas de información actuales se basan en bases de datos (BD) y sistemas de bases de datos (SGBD) que se han convertido en elementos imprescindibles de la vida cotidiana de la sociedad moderna.



Tabla de contenido

Introducción.....	1
Introducción a las bases de datos	3
Fundamentos de bases de datos	3
Definición de Bases de Datos	4
Evolución de las bases de Datos	5
Evolución y tipos de bases de datos.....	5
Concepto y Origen de las BD y de los SGBD.....	6
Modelado de datos	8
Tipos de modelados	9
Base de datos relacional	16
Modelado Logicos.....	21
Normalización	22
Formas Normales	26
Conclusiones y recomendaciones	30
Referencias bibliográficas.....	31

INCLUIR TEXTO DE RELEVANTE QUE DESEA RESALTAR, DEBE INCLUIR IDEAS COMPLETAS QUE TENGAN SENTIDO POR SI MISMAS. ES POSIBLE AJUSTAR EL TAMAÑO DEL RECUADRO A LA EXTENSIÓN DEL TEXTO.

Introducción a las bases de datos

Fundamentos de bases de datos

En el entorno del mercado actual, la competitividad y la rapidez de maniobra de una empresa son imprescindibles para su éxito. Para conseguirlo existe cada vez una mayor demanda de datos y, por tanto, más necesidad de gestionarlos. Esta demanda siempre ha estado patente en

empresas y sociedades, pero en estos años se ha disparado debido al acceso multitudinario a las redes integradas en Internet y a la aparición de los dispositivos móviles que también requieren esa información.

En informática se conoce como dato a cualquier elemento informativo que tenga relevancia para un usuario. Desde su nacimiento, la informática se ha encargado de proporcionar herramientas que faciliten la manipulación de los datos. Antes de la aparición de las aplicaciones informáticas, las empresas tenían como únicas herramientas de gestión de datos los ficheros con cajones, carpetas y fichas de cartón. En este proceso manual, el tiempo requerido para manipular estos datos era enorme. Pero la propia informática ha adaptado sus herramientas para que los elementos que el usuario utiliza en cuanto a manejo de datos se parezcan a los manuales. Por eso se sigue hablando de ficheros, formularios, carpetas, directorios.

La clientela fundamental del profesional informático es la empresa. La empresa se puede entender como un sistema de información formado por diversos objetos: el capital, los recursos humanos, los inmuebles, los servicios que presta, etc.

Los sistemas de información actuales se basan en bases de datos (BD) y sistemas de bases de datos (SGBD) que se han convertido en elementos imprescindibles de la vida cotidiana de la sociedad moderna.

Definición de Bases de Datos

Cada día, la mayoría de nosotros nos encontramos con actividades que requieren algún tipo de interacción con una base de datos (ingreso en un banco, reserva de una entrada para el teatro, solicitud de una suscripción a una revista, compra de productos, ...). Estas interacciones son ejemplos de lo que se llama aplicaciones tradicionales de bases de datos (básicamente información numérica o de texto), aunque los avances tecnológicos han permitido que también existan: bases de datos multimedia, sistemas de información geográfica (GIS), almacenes de datos, sistemas de proceso analítico on-line, ...

- Una base de datos se entenderá como una colección de datos relacionados entre sí y que tienen un significado implícito.
- Por datos queremos decir hechos conocidos que pueden registrarse y que tienen un significado implícito.

Por ejemplo, una agenda con los nombres y teléfonos de un conjunto de personas conocidas es una base de datos, puesto que es una colección de datos relacionados con un significado implícito.

La definición presentada anteriormente hace referencia a dos elementos para que un conjunto de datos constituya una Base de Datos:

Para manipular y gestionar las bases de datos surgieron herramientas software denominadas: sistemas gestores de bases de datos (SGBD en lo sucesivo) sobre los que se profundizará más adelante.

Evolución de las bases de Datos

Los predecesores de los sistemas gestores de bases de datos fueron los sistemas gestores de ficheros o sistemas de archivos tradicionales.

1. Archivos tradicionales. Consiste en almacenar los datos en archivos individuales, exclusivos para cada aplicación particular. En este sistema los datos pueden ser redundantes (repetidos innecesariamente) y la actualización de los archivos es más lenta que en una base de datos.
2. Base de datos. Es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones. La base de datos es una fuente de datos que son compartidos por numerosos usuarios para diversas aplicaciones.

Así, en un Sistema de archivos tradicional la información está dispersa en varios ficheros de datos y existe un cierto número de programas que los recuperan y agrupan. Aunque los sistemas de ficheros o archivos supusieron un gran avance sobre los sistemas manuales, tienen inconvenientes bastante importantes que se solventaron, en gran medida, con la aparición de los sistemas de bases de datos.

Evolución y tipos de bases de datos

Los predecesores de los sistemas gestores de bases de datos fueron los sistemas gestores de ficheros o sistemas de archivos tradicionales.

1. Archivos tradicionales. Consiste en almacenar los datos en archivos individuales, exclusivos para cada aplicación particular. En este sistema los datos pueden ser redundantes (repetidos innecesariamente) y la actualización de los archivos es más lenta que en una base de datos.
2. Base de datos. Es un almacenamiento de datos formalmente definido, controlado centralmente para intentar servir a múltiples y diferentes aplicaciones. La base de



datos es una fuente de datos que son compartidos por numerosos usuarios para diversas aplicaciones.

Así, en un Sistema de archivos tradicional la información está dispersa en varios ficheros de datos y existe un cierto número de programas que los recuperan y agrupan. Aunque los sistemas de ficheros o archivos supusieron un gran avance sobre los sistemas manuales, tienen inconvenientes bastante importantes que se solventaron, en gran medida, con la aparición de los sistemas de bases de datos.

Concepto y Origen de las BD y de los SGBD

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de base de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización.

Una base de datos de puede definir como:

- Una colección interrelacionada de datos, almacenados en un conjunto sin redundancias innecesarias cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente.
- Conjunto de datos organizados con características afines entre sí, que identifican a un ente en especial, cualquier colección de información interrelacionada, es una base de datos.

- Conjunto autodescriptivo de registros integrados; autodescriptivo ya que puede contener información del usuario, descripción de la Base de Datos, que nos permiten representar las relaciones entre los datos. En donde la información que se tiene almacenada está disponible para todos los usuarios del sistema, en los que los datos redundantes pueden eliminarse o al menos minimizarse.

Los componentes de una base de datos son:

- Los datos:
 - El componente fundamental de una base de datos son los datos que están interrelacionados entre sí, formando un conjunto con un mínimo de redundancias.
- El software:
 - Los datos, para que puedan ser utilizados por diferentes usuarios y diferentes aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se utiliza un software o conjunto de programas que actúa de interfaz entre los datos y las aplicaciones. A este software se le denomina Sistema de Gestión de Base de Datos (SGBD). El SGBD crea y organiza la base de datos, y además atiende todas las solicitudes de acceso hechas a la base de datos tanto por los usuarios como por las aplicaciones.
- Personas:
 - Informático: son los profesionales que definen y preparan la base de datos.
 - Usuarios: Son los que tienen acceso a la base de datos para ingresar o manipular la información contenida en esta, dependiendo del nivel de usuario o control de seguridad establecido.

Modelado de datos

El modelado de datos es una manera de estructurar y organizar los datos para que se puedan utilizar fácilmente por las bases de datos.

En otras palabras, un modelo es un conjunto de herramientas conceptuales para describir datos, sus relaciones, su significado y sus restricciones de consistencia.

Las tareas relacionadas con el diseño de bases de datos abarcan todo el ciclo vital de desarrollo de software, y las tareas relacionadas con el diseño de bases de datos iniciales pueden iniciarse durante la fase inicial. En proyectos que utilizan el modelado empresarial para describir el contexto empresarial de la aplicación, el diseño de bases de datos puede iniciarse en un nivel conceptual con la identificación de actores empresariales y guiones de uso empresariales en el modelo de guiones de uso empresariales, y los trabajadores empresariales y entidades empresariales en el modelo de análisis empresarial. En proyectos que no utilizan el modelado empresarial, el diseño de bases de datos puede empezar en el nivel conceptual con la identificación de actores del sistema y guiones de uso del sistema en el modelo de guion de uso, y la identificación de clases de análisis en el modelo de análisis de las realizaciones de guiones de uso.

El modelado de datos tiene como características,

- Es el proceso de analizar los aspectos de interés para una organización y la relación que tienen unos con otros.
- Resulta en el descubrimiento y documentación de los recursos de datos del negocio.
- El modelado hace la pregunta ¿" Qué ? " en lugar de ¿" Cómo ? ", ésta última orientada al procesamiento de los datos.
- No es una tarea sencilla, tiene un nivel de complejidad muy alto, pero es una actividad necesaria y es casi uno de los procesos críticos en el análisis de un proceso de negocio.

Ahora bien, el modelado de datos, tiene como meta u obtener el beneficio de,

- Registrar los requerimientos de datos de un proceso de negocio.
- Dicho proceso puede ser demasiado complejo y se tendrá que crear un "enterprise data model", el cual deberá estar constituido de líneas individuales.
- Permite observar:
 - Patrones de datos
 - Usos potenciales de los datos

Tipos de modelados

En resumen, existen 3 tipos de modelados de datos,

- Conceptual: muy general y abstracto, visión general del negocio/institución, se puede decir que esta fase incluye la identificación de las entidades del sistema y empresariales claves de nivel superior y sus relaciones, que definen el ámbito del problema que tratará el sistema.
- Lógico: versión completa que incluye todos los detalles acerca de los datos. Esta fase incluye el perfeccionamiento de las entidades del sistema y empresariales de alto nivel de la fase conceptual en entidades lógicas más detalladas.
- Físico: Esquema que se implementara en un manejador de bases de datos (DBMS), así mismo esta fase incluye la transformación de los diseños de la clase lógica en diseños de tablas de bases de datos físicas detalladas y optimizadas. La fase física también incluye la correlación de los diseños de tablas de base de datos con espacios de tablas y con el componente de base de datos en el diseño de almacenamiento de bases de datos.

Modelado de datos Conceptual

El modelado de datos conceptuales representa la fase inicial del desarrollo del diseño de los datos permanentes y el almacenamiento de datos permanentes para el sistema. En muchos casos, los datos permanentes para el sistema los gestiona un sistema de gestión de bases de datos relacionales (RDBMS). Las entidades del sistema y empresariales que se identifican en un nivel conceptual de los modelos empresariales y los requisitos del sistema se desarrollarán durante las tareas de análisis de guiones de uso, de diseño de guiones de uso y de diseño de bases de datos hasta convertirse en diseños de tablas físicas detalladas que se implementarán en el RDBMS. Tenga en cuenta que el modelo conceptual de datos que se trata en este documento de conceptos no es un producto de trabajo independiente. Consiste en una vista compuesta de información contenida en productos de trabajo de modelado empresarial, requisitos y disciplinas de análisis y diseño que es importante para el desarrollo del modelo de datos.

Existen diferentes tipos de modelos conceptuales

- Basados en registros:
 - Jerárquico: datos en registros, relacionados con apuntadores y organizados como colecciones de árboles.
 - Redes: datos en registros relacionados por apuntadores y organizados en gráficas arbitrarias.
 - Relacional: datos en tablas relacionados por el contenido de ciertas columnas.

- Basados en Objetos:
 - Orientado a objetos: datos como instancias de objetos (incluyendo sus métodos).
 - Entidad-relación: datos organizados en conjuntos interrelacionados de objetos



(entidades) con atributos asociados.

Modelo Jerárquico

Un modelo de datos jerárquico es un modelo de datos en el cual los datos son organizados en una estructura parecida a un árbol. La estructura permite a la información que se repite y usa relaciones padres/Hijo: cada padre puede tener muchos hijos, pero cada hijo sólo tiene un padre. Todos los atributos de un registro específico son catalogados bajo un tipo de entidad.

En una base de datos, un tipo de entidad es el equivalente de una tabla; cada registro individual es representado como una fila y un atributo como una columna. Los tipos de entidad son relacionados el uno con el otro usando 1: Trazar un mapa de n, también conocido como relación de uno a varios. El ejemplo más aprobado de base de datos jerárquica modela es un IMS diseñado por la IBM.

Un ejemplo de un modelo de datos jerárquico sería si una organización tuviera los registros de empleados en una tabla (el tipo de entidad) llamada "Empleados". En la tabla habría atributos/columnas como el Nombre de pila, el Apellido, el Nombre de Trabajo y el Salario. La empresa también tiene datos sobre los hijos del empleado en una tabla separada "Hijos" llamada con atributos como el Nombre de pila, el Apellido, y la fecha de nacimiento. La tabla de Empleado representa un segmento paternal y la tabla de Hijos representa un segmento Infantil. Estos dos segmentos forman una jerarquía donde un empleado puede tener muchos hijos, pero cada hijo solo puede tener un padre.

Considere la estructura siguiente:

EmpNo	Puesto	Reporta
10	Director	



20	Senior Manager	10
30	Typist	20
40	Programmer	20

En esta tabla, "el hijo" es el mismo tipo que "el padre". La jerarquía que declara EmpNo 10 es el jefe de 20, y 30 y 40 cada informe a 20 es representado por la columna "Reporta". Llamada en la Base de datos relacional, la columna Reporta es una llave foránea, el referirse de la columna EmpNo. Si el tipo de datos "hijo" fuera diferente, estaría en una tabla diferente, pero todavía habría una llave foránea que se refiere la columna EmpNo de la tabla de empleados.

Comúnmente se conocen a estos modelos simplemente como la lista de adyacencia, fue presentado por el Doctor Edgar F. Codd.

Modelo de Redes

La mayor parte de bases de datos de objeto usan el concepto de navegación para proporcionar la navegación rápida a través de las redes de objetos, generalmente usando identificadores de objeto como indicadores "inteligentes" de objetos relacionados.

El modelo de red organiza datos que usan dos fundamental construcciones, registros y conjuntos. Los registros contienen campos (que puede ser organizado jerárquicamente, como en el lenguaje COBOL de lenguaje de programación). Los conjuntos (para no ser confundido con conjuntos matemáticos) definen de uno a varias relaciones entre registros: un propietario, muchos miembros.

Un registro puede ser un propietario en cualquier número de conjuntos, y un miembro en cualquier número de conjuntos. El modelo de red es una variación sobre el modelo jerárquico, al grado que es construido sobre el concepto de múltiples ramas (estructuras de nivel inferior) emanando de uno o varios nodos (estructuras de nivel alto), mientras el modelo se diferencia del modelo jerárquico en esto las ramas pueden estar unidas a múltiples nodos. El modelo de red es capaz de representar la redundancia en datos de una manera más eficiente que en el modelo jerárquico. Las operaciones del modelo de red son de navegación en el estilo: un programa mantiene una posición corriente, y navega de un registro al otro por siguiente las relaciones en las cuales el registro participa.

Los registros también pueden ser localizados por suministrando valores claves. Aunque esto no sea un rasgo esencial del modelo, las bases de datos de red generalmente ponen en práctica las relaciones de juego mediante indicadores que directamente dirigen la ubicación de un registro sobre el disco. Esto da el funcionamiento de recuperación excelente, a cargo de operaciones como la carga de base de datos y la reorganización.

La mayor parte de bases de datos de objeto usan el concepto de navegación para proporcionar la navegación rápida a través de las redes de objetos, generalmente usando identificadores de objeto como indicadores "inteligentes" de objetos relacionados. Objectivity/DB, por ejemplo, los instrumentos llamados 1:1, 1:muchos, muchos:1 y muchos:



muchos, llamados relaciones que pueden cruzar bases de datos. Muchas bases de datos de objeto también apoyan SQL, combinando las fuerzas de ambos modelos. En el modelo de red no existen restricciones.

Modelo Relacional

El modelo relacional, para el modelado y la gestión de bases de datos, es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos.

Su idea fundamental es el uso de relaciones. Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas. Pese a que esta es la teoría de las bases de datos relacionales creadas por Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, pensando en cada relación como si fuese una tabla que está compuesta por registros (cada fila de la tabla sería un registro o "tupla") y columnas (también llamadas "campos").

Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional

solamente indica lo que se desea devolver.

Esquema

Un esquema contiene la definición de una estructura (generalmente relaciones o tablas de una base de datos), es decir, determina la identidad de la relación y qué tipo de información podrá ser almacenada dentro de ella; en otras palabras, el esquema contiene los metadatos de la relación. Todo esquema constará de:

- Nombre de la relación (su identificador).
- Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, equivalente al tipo de dato por ejemplo character, integer, date, string...

Instancias

Una instancia de manera formal es la aplicación de un esquema a un conjunto finito de datos. En palabras no tan técnicas, se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla, como, por ejemplo:

- Ciertos caracteres y números (una sola columna de una sola fila).
- Algunas o todas las filas con todas o algunas columnas
 - Cada fila es una tupla. El número de filas es llamado cardinalidad.
 - El número de columnas es llamado aridad o grado.

Base de datos relacional

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional.

Estrictamente hablando el término se refiere a una colección específica de datos, pero a menudo se le usa, en forma errónea como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como sistema gestor de base de datos relacional (SGBD) o en inglés relational database management system (RDBMS).

Modelo Orientado a Objetos

La tecnología de bases de datos vive un momento de lenta transición del modelo relacional a otros modelos. Entre éstos se encuentra el multidimensional para sistemas OLAP, el semiestructurado para bases de datos XML de intercambio electrónico de información, el modelo dimensional para creación de Data Warehouse y el orientado a objetos. En una base de datos orientada a objetos, los componentes se almacenan como objetos y no como datos, tal y como hace una base relacional, cuya representación son las tablas.

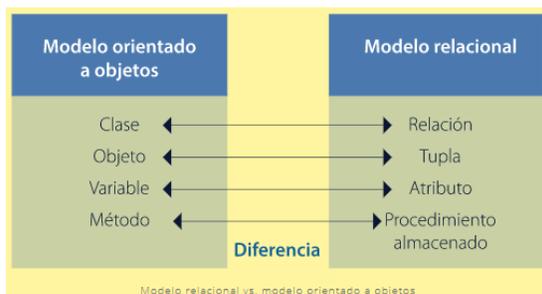
Algo importante que debemos resaltar es que hoy en día, las empresas siguen utilizando los manejadores de bases de datos relacionales y no se sabe aún si serán suplantadas por completo, ni cuándo.

Utilidad del modelo de base de datos orientado a objetos

Los administradores de base de datos (DBMS por sus siglas en inglés) evolucionan con el afán de satisfacer nuevos requerimientos tecnológicos y de información. Aunque los DBMS

relacionales (RDBMS) son actualmente líderes del mercado y brindan las soluciones necesarias a las empresas comerciales, existen aplicaciones que necesitan funciones con las que no cuentan. Las CAD/CAM, los sistemas multimedia, como los geográficos y de medio ambiente, los de gestión de imágenes y documentos y los de apoyo a las decisiones necesitan de modelos de datos complejos, difíciles de representar como tuplas de una tabla.

En general, estas aplicaciones necesitan manipular objetos y los modelos de datos deben permitirles expresar su comportamiento y las relaciones entre ellos.



Modelo Entidad – Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados entidades y de relaciones entre estos objetos.

Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos.

El modelo E-R (Entidad-Relación) además de entidades y relaciones representan las uniones que los contenidos de la base de datos deben cumplir. Una unión es la correspondencia de cardinalidades, que expresan el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones.

Conceptos básicos: Existen tres conceptos fundamentales que se emplean en el modelo de datos E-R (Entidad-Relación): conjunto de entidades, conjunto de relaciones y atributos, las cuales se definen a continuación.



Entidad

Se puede definir como Entidad a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar información. Una entidad tiene propiedades y valores que identifican a un sujeto u objeto el cual existe y es distinguible de otros objetos, se representan por un conjunto de atributos, ejemplo entidad cliente: rfc, nombre, dirección, teléfono.

Un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades, o atributos.

Atributos

Los Atributos son características o propiedades asociadas a la entidad que toman valor en una instancia particular. Ejemplo: nombre, cédula, teléfono. Cada entidad tiene un valor para cada uno de sus atributos.

Dominio del atributo: Para cada atributo hay un conjunto de valores permitidos, llamados el dominio, o el conjunto de valores, de ese atributo.

Un atributo, como se usa en el modelo E-R, se puede caracterizar por los siguientes tipos de atributo:

- Atributos simples: Un atributo simple es aquel que no se puede subdividir, por ejemplo, la edad y el sexo de una persona.
- Atributos compuestos: Un atributo compuesto, es un atributo que puede ser subdividido en otros atributos adicionales, por ejemplo, la dirección de una persona, puede subdividirse en calle, número, código postal, etc.
- Atributos Mono valorados y multivalorados.
- Atributos derivados.

Llave o clave de la relación: Es el identificador único de cada tupla.

Clave primaria: clave candidata que el diseñador elige de la base de datos como el medio

principal de identificar entidades dentro de un conjunto de entidades.

Clave compuesta: Una clave compuesta de más de un atributo.

Clave candidata: Cualquier conjunto de atributos que puede ser elegido como clave de una relación.

Clave externa: Un conjunto de atributos o un atributo, en una relación que constituyen una clave en alguna otra relación, usada para establecer enlaces lógicos entre relaciones.

Tupla

Conjunto de atributos que representan a una unidad. Valor nulo: El valor dado a un atributo en una tupla si el atributo es inaplicable o su valor es desconocido.

Cardinalidad: Numero específico de ocurrencias de una entidad, asociadas con una ocurrencia de la entidad relacionada, esto es el número máximo de instancias de un conjunto de objetos que puede estar relacionado con una sola instancia de otro conjunto de objetos.

Relación

Una relación es una asociación entre entidades, se denomina de igual modo a una tabla que se genera a partir de la relación o asociación de dos o más tablas o entidades existentes.

Restricciones

Un esquema de desarrollo E-R puede definir ciertas restricciones a las que los contenidos de la base de datos se deben adaptar. En este apartado se examina la correspondencia de cardinalidades y las restricciones de participación, que son dos de los tipos más importantes de restricciones.

La correspondencia de cardinalidades, o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

Reglas de cardinalidad:

- Cardinalidad de uno a uno: Una cardinalidad de la interrelación que es 1 en ambas direcciones. Cuando un registro de una tabla sólo puede estar relacionado con un único registro de la otra tabla y viceversa. En este caso la clave foránea se ubica en alguna de las 2 tablas.
- Cardinalidad de uno a muchos: Cuando un registro de una tabla (tabla secundaria) sólo puede estar relacionado con un único registro de la otra tabla (tabla principal) y un registro de la tabla principal puede tener más de un registro relacionado en la tabla secundaria. En este caso la clave foránea se ubica en la tabla secundaria.
- Cardinalidad de muchos a muchos: Cuando un registro de una tabla puede estar relacionado con más de un registro de la otra tabla y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos (Tabla débil o de vinculación) que incluya los pares de valores relacionados entre sí.

El nombre de tabla débil deviene de que con sus atributos propios no se puede encontrar la clave, por estar asociada a otra entidad. La clave de esta tabla se conforma por la unión de los campos claves de las tablas que relaciona.

Reglas que determinan las interrelaciones (cardinalidad).

- Regla 1. Si dos tablas tienen una interrelación de uno a uno (1 a 1), entonces el campo clave de una de las tablas debe aparecer en la otra tabla.
- Regla 2. Si dos tablas tienen una interrelación de uno a muchos (1 a *), entonces el campo clave de la tabla del (1) debe aparecer en la tabla del muchos (*).
- Regla 3. Si dos tablas tienen una interrelación de muchos a muchos (* a *), entonces debe crearse una tabla que tenga los campos claves de las dos tablas.

Modelado Logicos

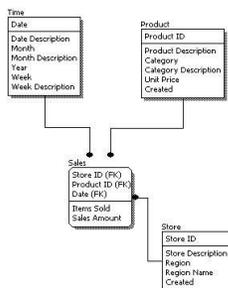
Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos.

Las características de un modelo de datos lógicos incluyen:

- Incluye todas las entidades y relaciones entre ellos.
- Todos los atributos para cada entidad están especificados.
- La clave principal para cada entidad está especificada.
- Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).
- La normalización ocurre en este nivel.

Los pasos para diseñar el modelo de datos lógicos son los siguientes:

- Especifique claves primarias para todas las entidades.
- Encuentra las relaciones entre diferentes entidades.
- Encuentra todos los atributos para cada entidad.
- Resuelva las relaciones de muchos a muchos.
- Normalización.



Ventajas y desventajas de los modelos de datos

- Ventajas
 - El objetivo principal de un modelo de datos es asegurarse de que los objetos de datos ofrecidos por el equipo funcional se representen con precisión.

- El modelo de datos debe ser lo suficientemente detallado para ser utilizado para construir la base de datos física.
- La información en el modelo de datos se puede utilizar para definir la relación entre tablas, claves primarias y externas y procedimientos almacenados.
- El modelo de datos ayuda a las empresas a comunicarse dentro y entre las organizaciones.
- El modelo de datos ayuda a documentar las asignaciones de datos en el proceso ETL
- Ayuda a reconocer las fuentes de datos correctas para poblar el modelo.
- Desventajas
 - Para desarrollar el modelo de datos se deben conocer las características físicas de los datos almacenados.
 - Incluso los cambios más pequeños realizados en la estructura requieren modificaciones en toda la aplicación.
 - No hay un lenguaje de manipulación de modelos establecido en DBMS.

Normalización

La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. Con objeto de minimizar la redundancia de datos, facilitando su gestión posterior.

Las bases de datos relacionales se normalizan para:

- Minimizar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las tablas.
- Proteger la integridad de datos.

En el modelo relacional es frecuente llamar tabla a una relación; para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

- Cada tabla debe tener su nombre único.

- No puede haber dos filas iguales. No se permiten los duplicados.
- Todos los datos en una columna deben ser del mismo tipo

Terminología

- Relación = tabla
- Registro = fila, o tupla
- Atributo = columna o campo
- Clave = llave o código de identificación
- Clave Candidata = superclave mínima
- Clave Primaria = clave candidata elegida
- Clave Externa = clave ajena o clave foránea
- Clave Alternativa = clave secundaria
- Dependencia Multivaluada = dependencia multivalor = dependencia múltiple
- RDBMS = Del inglés Relational Data Base Management System, que significa Sistema de gestión de bases de datos relacionales.
- 1FN = Significa Primera Forma Normal o 1NF, del inglés First Normal Form.

Los términos Relación, Tupla y Atributo derivan del álgebra y cálculo relacional, que constituyen la base teórica del modelo de base de datos relacional.

Todo atributo en una tabla tiene un dominio, el cual representa el conjunto de valores que el mismo puede tomar. Una instancia de una tabla puede verse entonces como un subconjunto del producto cartesiano entre los dominios de los atributos. Sin embargo, suele haber algunas diferencias con la analogía matemática, ya que algunos RDBMS permiten filas duplicadas, entre otras cosas. Finalmente, una tupla puede razonarse matemáticamente como un elemento del producto cartesiano entre los dominios.

Dependencias

El proceso de normalización se basa en relaciones que se conocen que mantienen los datos, principalmente dependencias funcionales, multivaluadas y de join.



- Dependencia funcional:
 - Una dependencia funcional es una relación entre uno o más atributos.
 - Propiedades:
 - Dependencia funcional reflexiva:
 - Si "y" está incluido en "x" entonces $x \rightarrow y$
 - A partir de cualquier atributo o conjunto de atributos siempre puede deducirse él mismo. Si la dirección o el nombre de una persona están incluidos en el DNI, entonces con el DNI podemos determinar la dirección o su nombre.

- Dependencia funcional aumentativa
 - $X \rightarrow Y$ entonces $xy \rightarrow yz$
 - DNI \rightarrow nombre
 - DNI, dirección \rightarrow nombre, dirección
 - Si con el DNI se determina el nombre de una persona, entonces con el DNI más la dirección también se determina el nombre y su dirección.

- Dependencia funcional transitiva
 - Sean X, Y, Z tres atributos (o grupos de atributos) de la misma entidad. Si Y depende funcionalmente de X y Z de Y, pero X no depende funcionalmente de Y, se dice entonces que Z depende transitivamente de X. Simbólicamente sería:
 - $X \rightarrow Y \rightarrow Z$ entonces $X \rightarrow Z$
 - FechaNacimiento \rightarrow Edad
 - Edad \rightarrow Conducir
 - FechaNacimiento \rightarrow Edad \rightarrow Conducir
 - Entonces entendemos que FechaNacimiento determina a Edad y la Edad determina a Conducir, indirectamente podemos saber a través de FechaNacimiento a Conducir (En muchos países, una persona

necesita ser mayor de cierta edad para poder conducir un automóvil, por eso se utiliza este ejemplo).

- C será un dato simple (dato no primario), B, será un otro dato simple (dato no primario), A, es la llave primaria (PK). Decimos que C dependerá de B y B dependerá funcionalmente de A.

Claves

Una clave primaria es el conjunto mínimo de columnas que identifica unívocamente a cada fila. La clave primaria es un identificador que va a ser siempre único para cada fila. Se acostumbra a poner la clave primaria como la primera columna de la tabla, pero es más una conveniencia que una obligación. Muchas veces la clave primaria es numérica auto-incrementada, es decir, generada mediante una secuencia numérica incrementada automáticamente cada vez que se inserta una fila.

En una tabla puede que tengamos más de una columna que puede ser clave primaria por sí misma. En ese caso se puede escoger una para ser la clave primaria y las demás claves serán claves candidatas.

Una clave ajena (foreign key o clave foránea) es aquella columna que existiendo como dependiente en una tabla, es a su vez clave primaria en otra tabla.

Una clave alternativa es aquella clave candidata que no ha sido seleccionada como clave primaria, pero que también puede identificar de forma única a una fila dentro de una tabla. Ejemplo: Si en una tabla clientes definimos el número de documento (id_cliente) como clave primaria, el número de seguro social de ese cliente podría ser una clave alternativa. En este caso no se usó como clave primaria porque es posible que no se conozca ese dato en todos los clientes.

Una clave compuesta es una clave que está compuesta por más de una columna.

La visualización de todas las posibles claves candidatas en una tabla ayudan a su optimización. Por ejemplo, en una tabla PERSONA podemos identificar como claves su DNI, o el conjunto de su nombre, apellidos, fecha de nacimiento y dirección. Podemos usar cualquiera de las dos opciones o incluso todas a la vez como clave primaria, pero es mejor

en la mayoría de sistemas la elección del menor número de columnas como clave primaria.

Formas Normales

Las formas normales son aplicadas a las tablas de una base de datos. Decir que una base de datos está en la forma normal N es decir que todas sus tablas están en la forma normal N. En general, las primeras tres formas normales son el mínimo que deben cubrir la mayoría de las bases de datos. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F. Codd.

- Primera Forma Normal (1FN)
 - Una tabla está en primera forma si:
 - Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
 - 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.
 - Esta forma normal elimina los valores repetidos dentro de una base de datos.

- Segunda Forma Normal (2FN)
 - Dependencia funcional. Una relación 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.
 - En otras palabras, podríamos decir que la segunda forma normal está basada

en el concepto de dependencia completamente funcional.

- Tercera Forma Normal (3FN)
 - La 3NF fue definida originalmente por E.F. Codd. La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva en los atributos que no son clave.

Reglas de Codd

Edgar Frank Codd se percató de que existían bases de datos en el mercado que decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar literalmente normalizadas dichas tablas; entonces Codd publicó doce (12) reglas que un verdadero sistema relacional debería tener, en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas.

- Regla 1: La regla de la información: Toda la información en un RDBMS está explícitamente representada de una sola manera por valores en una tabla.
- Regla 2: La regla del acceso garantizado
- Cada ítem de datos debe ser lógicamente accesible al ejecutar una búsqueda que combine el nombre de la tabla, su clave primaria y el nombre de la columna.
- Regla 3: Tratamiento sistemático de los valores nulos
- La información inaplicable o faltante puede ser representada a través de valores nulos
- Regla 4: La regla de la descripción de la base de datos
- La descripción de la base de datos es almacenada de la misma manera que los datos ordinarios, esto es, en tablas y columnas, y debe ser accesible a los usuarios autorizados.
- Regla 5: La regla del sub-lenguaje integral
- Debe haber al menos un lenguaje que sea integral para soportar la definición de datos,

manipulación de datos, definición de vistas, restricciones de integridad, y control de autorizaciones y transacciones.

- Regla 6: La regla de la actualización de vistas
- Todas las vistas que son teóricamente actualizables, deben ser actualizables por el sistema mismo.
- Regla 7: La regla de insertar y actualizar
- La capacidad de manejar una base de datos con operandos simples se aplica no sólo para la recuperación o consulta de datos, sino también para la inserción, actualización y borrado de datos'.
- Regla 8: La regla de independencia física
- El acceso de usuarios a la base de datos a través de terminales o programas de aplicación, debe permanecer consistente lógicamente cuando quiera que haya cambios en los datos almacenados, o sean cambiados los métodos de acceso a los datos.
- Regla 9: La regla de independencia lógica
- Los programas de aplicación y las actividades de acceso por terminal deben permanecer lógicamente inalteradas cuando quiera que se hagan cambios (según los permisos asignados) en las tablas de la base de datos.
- Regla 10: La regla de la independencia de la integridad
- Todas las restricciones de integridad deben ser definibles en los datos, y almacenables en el catálogo, no en el programa de aplicación.
- Regla 11: La regla de la distribución
- El sistema debe poseer un lenguaje de datos que pueda soportar que la base de datos esté distribuida físicamente en distintos lugares sin que esto afecte o altere a los programas de aplicación.
- Regla 12: Regla de la no-subversión
- Si el sistema tiene lenguajes de bajo nivel, estos lenguajes de ninguna manera pueden ser usados para violar la integridad de las reglas y restricciones expresadas en un lenguaje de alto nivel (como SQL).

Modelado Físicos

Un modelo de datos físicos es un modelo específico de bases de datos que representa objetos de datos relacionales (por ejemplo, tablas, columnas, claves principales y claves externas) y sus relaciones. Un modelo de datos físicos se puede utilizar para generar sentencias DDL que, después, se pueden desplegar en un servidor de base de datos.

El modelo de datos físicos representa cómo se construirá el modelo en la base de datos. Un modelo de base de datos física muestra todas las estructuras de tabla, incluidos el nombre de columna, el tipo de datos de columna, las restricciones de columna, la clave principal, la clave externa y las relaciones entre las tablas.

Las características de un modelo de datos físicos incluyen:

- Especificación de todas las tablas y columnas.
- Las claves externas se usan para identificar relaciones entre tablas.
- La desnormalización puede ocurrir según los requisitos del usuario.

Las consideraciones físicas pueden hacer que el modelo de datos físicos sea bastante diferente del modelo de datos lógicos.

El modelo de datos físicos será diferente para diferentes Sistemas de Gestión de Base de datos. Por ejemplo, el tipo de datos para una columna puede ser diferente entre MySQL y SQL Server.

Los pasos básicos para el diseño del modelo de datos físicos son los siguientes:

- Convertir entidades en tablas.
- Convertir relaciones en claves externas.
- Convertir atributos en columnas.
- Modificar el modelo de datos físicos en función de las restricciones / requisitos físicos.

Estas lecturas constituyen un apoyo en el proceso de aprendizaje, Así mismo, todo el material investigativo por tema que el estudiante utilice para su aprendizaje es permitido, pero deben prestar atención que los mismos no tengan mas 5 años de publicado.

Conclusiones y recomendaciones

Algunos de los aspectos aprendidos y que de gran peso es la base de datos su definición, requerimiento, ventajas y características donde podemos decir que la base de datos: Es una colección de datos o información usados para dar servicios a muchas aplicaciones al mismo tiempo.

En cuanto al requerimiento podemos decir que cumple las mismas tareas de análisis que del software y tiene como característica relacionar la información como vía organización y asociación donde la base de datos tiene una ventaja que es utilizar la plataforma para el desarrollo del sistema de aplicación en las organizaciones.

Otro aspecto importante sería el diseño y creación de la base de datos, donde existen distintos modos de organizar la información y representar las relaciones entre por datos los tres modelos lógicos principales dentro de una base de datos son el jerárquico, de redes y el relacional, los cuales tienen ciertas ventajas de procesamiento y de negocios.

Referencias bibliográficas

- Bertone, N. (2017). *Introducción a las bases de datos: fundamentos y diseño*. Pearson Educación
- Silberschatz, A. & Sudarshan, S. (2018). *Fundamentos de bases de datos* (6a. ed.). McGraw-Hill Interamericana.
- Bernal, Nieto. (2017). *Diseño de base de datos*. Universidad del Norte
- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). *Base de datos*. Grupo Editorial Patria.

<https://sites.google.com/site/marichelacatotatics2/1-3-elaboracion-de-modelos-relacionales-de-bases-de-datos-de-manera-tecnica-utilizando-una-herramienta-de-diseno-de-base-de-datos?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

<https://gestionbasesdatos.readthedocs.io/es/latest/Tema1/Teoria.html>

https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/guidances/concepts/conceptual_data_modeling_5CF78A88.html

<https://www.marcoteorico.com/curso/31/fundamentos-de-bases-de-datos/414/modelo-entidad-relacion->

<https://www.marcoteorico.com/curso/31/fundamentos-de-bases-de-datos/417/disenio-con-diagramas-e-r->



www.usanmarcos.ac.cr

San José, Costa Rica