

BASES DE DATOS

TEMA 4

DISEÑO DE BASES DE DATOS RELACIONALES

El modelo relacional se basa en dos ramas de las matemáticas: la teoría de conjuntos y la lógica de predicados de primer orden. El hecho de que el modelo relacional esté basado en la teoría de las matemáticas es lo que lo hace tan seguro y robusto. Al mismo tiempo, estas ramas de las matemáticas proporcionan los elementos básicos necesarios para crear una base de datos relacional con una buena estructura, y proporcionan las líneas que se utilizan para formular buenas metodologías de diseño.

La teoría matemática proporciona la base para el modelo relacional y, por lo tanto, hace que el modelo sea predecible, fiable y seguro. La teoría describe los elementos básicos que se utilizan para crear una base de datos relacional y proporciona las líneas a seguir para construirla. El organizar estos elementos para conseguir el resultado deseado es lo que se denomina diseño.

En 1970, el modo en que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco) para relacionar registros de distintos ficheros. Codd demostró que estas bases de datos limitaban en gran medida los tipos de operaciones que los usuarios podían realizar sobre los datos.

El modelo relacional representa la segunda generación de los DBMS. En él, todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. El modelo relacional, como todo modelo de datos, tiene que ver con tres aspectos de los datos: Estructura de datos, Integridad de datos, Manejo de datos.

4.1 Definición del problema

Es el proceso por el que se determina la organización de una DB, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar. Éste se considera ahora una disciplina estable, con métodos y técnicas propios.

Para definir correctamente lo primero es realizar diseño conceptual, que parte de las especificaciones de los requisitos del usuario y su resultado es el esquema conceptual de la base de datos que corresponderá a un Modelo E / R. **Un esquema conceptual** es una descripción de alto nivel de la estructura de la base de datos, independientemente del DBMS que se vaya a utilizar para manipularla. **Un modelo conceptual** es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del **diseño conceptual** es describir el contenido de los datos de la DB y no las estructuras de almacenamiento que se necesitarán para manejar esta información. El modelo relacional puede considerarse como un lenguaje de programación abstracto, orientado de manera específica hacia las aplicaciones de bases de datos. En términos tradicionales una relación se asemeja a un archivo, una tupla a un registro, y un atributo a un campo. Pero estas correspondencias son aproximadas. Las características principales de los “archivos” relacionales son:

- Cada archivo contiene solo un tipo de registros.
- Los campos no tienen un orden específico, de izquierda a derecha.
- Los registros no tienen un orden específico, de arriba hacia abajo.
- Cada campo tiene un solo valor.
- Los registros poseen un campo identificador único llamado clave primaria.

Actualmente algunos de los manejadores de DB, utilizan un sistema de búsqueda con algoritmos de árboles. Pero las búsquedas que se pueden realizar con estos algoritmos son

sólo para memoria principal. Los algoritmos implementados para realizar búsquedas con listas salteadas o por bloques son eficientes para realizar búsquedas en memoria secundaria.

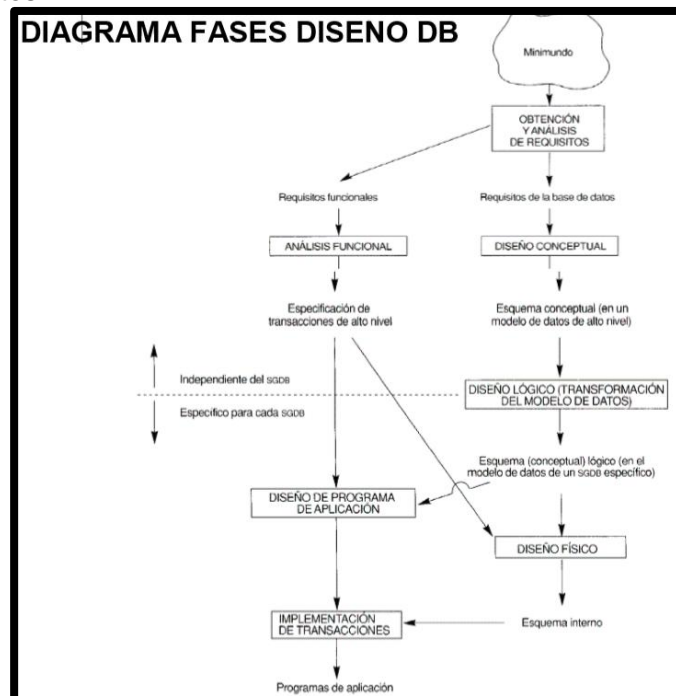
El primer paso para la definición del diseño de una base de datos es la producción del esquema conceptual. Normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información.

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. Determinar las jerarquías de generalización (si las hay).
7. Dibujar el diagrama entidad-relación.
8. Revisar el esquema conceptual local con el usuario.

El principal concepto del modelo ER es la **entidad**, que es una "cosa" en el mundo real con existencia independiente. Una entidad puede ser un objeto físico (una persona, un auto, una casa o un empleado) o un objeto conceptual (una compañía, un puesto de trabajo o un curso universitario). Cada entidad tiene propiedades específicas, llamadas **atributos**, que la describen. Una entidad particular tiene un valor para cada uno de los atributos. Cada uno de los atributos de una entidad posee un **dominio**, el que corresponde al tipo del atributo.

Para todo conjunto de valores de una entidad, debe existir un atributo o combinación de atributos, que identifique a cada entidad en forma única el cual se denomina llave (primaria). Una **relación** se puede definir como una asociación entre entidades. Una relación también puede tener atributos.



Tipos de relaciones

Un tipo de relación es un subconjunto del producto cartesiano $E_1 \times E_2 \times \dots \times E_n$. Un tipo de relación podría también interpretarse como un conjunto de pares ordenados, en este caso: (e_1, d_1) , (e_2, d_2) , (e_3, d_1) , (e_4, d_2) . Según el número de entidades relacionadas (o razón de cardinalidad), se pueden definir tres tipos de relaciones:

1. Relaciones Uno a Uno (1:1). Una entidad A está asociada a lo más con una entidad B, y una entidad B a lo más con una entidad A. Ejemplo: "Ser jefe de" es una relación 1:1 entre las entidades empleado y departamento.
2. Relaciones Uno a Muchos (1:n). Una entidad A está asociada con una o varias entidades B. Una entidad B, sin embargo, puede estar a lo más asociada con una entidad A. Ejemplo: "Ser profesor" es una relación 1:n entre profesor y curso, suponiendo que un curso sólo lo dicta un profesor.
3. Relaciones Muchos a Muchos (n:m). Una entidad A está asociada con una o varias entidades B, y una entidad B está asociada con una o varias entidades A. Ejemplo: "Estar inscrito" es una relación n:m entre las entidades alumno y curso.

Tipos de atributos

- ✓ *Atributos compuestos* se pueden dividir en sub-partes más pequeñas, que representan atributos más básicos con significados propios. Por ejemplo, una "dirección" puede sub-dividirse en: dir-calle, colonia, ciudad, región.
- ✓ *Atributos no sub-dividibles* se llaman atómicos o simples. Si no hay necesidad de referirse a los elementos individuales de una dirección, entonces la dirección completa puede considerarse un atributo simple, tienen un sólo valor para una entidad particular.
- ✓ Atributos multivalorados pueden tener un conjunto de valores para una misma entidad. Eje.: "títulos profesionales" (una persona puede tener uno, dos o más).

En algunos casos una entidad particular puede no tener valores aplicables para un atributo. Para estas situaciones tenemos un valor especial llamado nulo. También, si no se conoce el valor. Un tipo de entidad define un conjunto de entidades con los mismos atributos. En los diagramas E-R, un tipo de entidad se representa como una caja rectangular, los nombres de los atributos como elipses y las relaciones como rombos. Los atributos multivalorados se representan con elipses dobles.

Un tipo de atributo usualmente tiene un atributo cuyos valores son distintos para cada entidad individual (atributo clave o llave) y sus valores se usan para identificar cada entidad unívocamente. Cada atributo simple tiene un conjunto de valores o dominio asociado, que especifica el conjunto de valores que puede asignarse a cada entidad individual.

Este modelo considera la DB como una colección de relaciones. De manera simple, una relación representa una tabla, en que cada fila representa una colección de valores que describen una entidad del mundo real. Cada fila se denomina tupla.

Dominios, tuplas, atributos, relaciones

- ✓ *Un dominio D* es un conjunto de valores atómicos. Atómico quiere decir que cada valor en el dominio es indivisible. Es útil dar nombres a los dominios.
- ✓ *RUTs*: números de 8 dígitos más un carácter que puede ser del 0 al 9 o K Nombres: el conjunto de nombres de personas Notas: valores entre 1.0 y 7.0.
- ✓ Un *esquema de relación R*, denotado $R(A_1, A_2, \dots, A_n)$ está constituido por un nombre de relación R y una lista de atributos A_1, \dots, A_n . Se usa para describir una relación. R es el nombre de esta relación. El grado de una relación es el número n de atributos del esquema de la relación.

- ✓ Una *instancia de relación* refleja sólo las tuplas válidas que representa un estado particular del mundo real. A medida que este cambia, también lo hace la relación, transformándose en otro estado de relación.
- ✓ *Notación* Las letras Q, R, S denotan nombres de relación, las letras q, r, s denotan estados de relación, las letras t, u, v denotan tuplas. Los nombres de atributos se califican con el nombre de relación a la cual pertenecen.

Restricciones

Las restricciones de dominios especifican que el valor de cada atributo A debe ser un valor atómico del dominio $\text{dom}(A)$. Una relación se define como un conjunto de tuplas. Por definición todos los elementos de un conjunto son distintos. Luego todas las tuplas de una relación deben ser distintas. Esto implica que dos tuplas no pueden tener la misma combinación de valores para todos sus atributos. Pero puede haber otros subconjuntos de atributos de un esquema de relación R con la propiedad de que no haya dos tuplas en una instancia de relación r de R que tengan la misma combinación de valores para esos atributos.

El valor de un **atributo clave** se usa para identificar unívocamente una tupla en una relación. El hecho que un conjunto de atributos constituya una clase es una propiedad del esquema de la relación, y es invariante en el tiempo.

En general, un esquema de relación puede tener más de una clave, y en ese caso, cada una de las llaves es una **llave candidata**. Una de las llaves candidatas se designa como llave primaria de la relación. Usamos la convención de que los atributos que forman la llave primaria de un esquema de relación se subrayan.

Base de datos relacional

Es un conjunto de esquemas de relación $S = (R_1, R_2, \dots, R_m)$ y un conjunto RI de restricciones de integridad.

Una instancia de DB relacional DB de S es un conjunto de instancias de relación $DB = \{r_1, \dots, r_n\}$ tal que cada r_i es una instancia de R_i y tal que las relaciones r_i satisfacen las restricciones de integridad especificadas en RI.

La **restricción de integridad de entidad** establece que ningún valor de llave primaria puede ser nulo. Esto es porque ellas identifican tuplas de la relación.

La **restricción de integridad referencial** se especifica entre dos relaciones y se usa para mantener la consistencia entre tuplas de las dos relaciones. Informalmente, una tupla en una relación que hace referencia a otra relación debe referirse a una tupla existente en esa relación.

Operaciones de actualización

La operación **Insert** provee una lista de valores de atributos para una nueva tupla t que se va a insertar en una relación. El DBMS puede hacer dos opciones cuando hay violación de restricciones. Una es rechazar la inserción. La otra es intentar corregir la razón de rechazo de la inserción.

Operaciones de borrado

La operación **Delete** borra tuplas de una relación. Es posible que se viole la integridad referencial cuando una tupla que se quiere borrar es referenciada por las llaves foráneas de otras tuplas de la BD. Las tuplas a borrar se especifican a través de condiciones sobre los atributos de la relación. Hay tres opciones con respecto a una operación de borrado que causa una violación. La primera es rechazar la operación. La segunda es intentar propagar el borrado. Una tercera opción es modificar los valores de los atributos referenciantes que causan la violación (cada uno de estos valores puede ser puesto en nulo

o cambiado para referenciar otra tupla válida). Hay que observar que si un atributo referenciante que causa una violación, es parte de la llave primaria, no puede ser nulo, pues se violaría la integridad de entidad.

Operaciones de modificación

La operación **Modify** se usa para cambiar valores a uno o más atributos en una tupla (o tuplas) de una relación R. Es necesario especificar una condición sobre los atributos de R para seleccionar la o las tuplas a modificar. El modificar un atributo que no es llave primaria ni llave foránea no tiene problemas. El modificar una llave primaria es similar a borrar una tupla e insertar otra en su lugar. Por tanto, es relevante la discusión anterior (Insert y Delete). Si se modifica un atributo de una llave foránea, el DBMS debe asegurarse que el nuevo valor se refiere a una tupla existente en la relación referenciada.

Terminología relacional equivalente

Columna Nombre de la tabla: Trabajo

Código	Nombre	Posición	Salario
1	Edgardo Trujillo	Gerente	19000
2	Lidimarie Fonsi	Empleada	12000
3	Jean Piaget	Empleado	13500
4	Jerome Bruner	Empleado	14000

Fila

- ✓ Clave Primaria = clave candidata elegida
 - ✓ Clave Ajena = clave externa o clave foránea
 - ✓ Clave Alternativa = clave secundaria
 - ✓ Dependencia Multivaluada = dependencia multivalor
 - ✓ RDBMS = Del inglés Relational Data Base Manager System que significa, SistemaGestor de Bases de Datos Relacionales.
 - ✓ 1FN = Significa, Primera Forma Normal o 1NF del inglés First Normal Form.
- ✓ Relación = tabla o archivo
 - ✓ Tupla = registro, fila o renglón
 - ✓ Atributo = columna o campo
 - ✓ Clave = llave o código de identificación
 - ✓ Clave Candidata = superclave mínima

Los términos Relación, Tupla y Atributo derivan de las matemáticas relacionales, que constituyen la fuente 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. Finalmente, una tupla puede razonarse matemáticamente como un elemento del producto cartesiano entre los dominios.

4.2 Solución de Problemas

Redundancia e inconsistencia de datos.

Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea más de una vez en un dispositivo de almacenamiento. Esto aumenta los costos de almacenamiento y acceso a los datos, además de que puede originar la inconsistencia de los datos - es decir diversas copias de un mismo dato no concuerdan entre sí.

Dificultad para tener acceso a los datos.

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos.

Aislamiento de los datos.

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

Anomalías del acceso concurrente.

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente. En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

Problemas de seguridad.

La información de toda empresa es importante, aunque unos datos lo son más que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autenticación y protección de los datos.

Problemas de integridad.

Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación. Lo anterior es originado por:

- ✓ Redundancia
- ✓ Anomalías
 - Actualización
 - Inserción
 - Borrado

Creadas durante:

- ✓ Mantenimiento
- ✓ Creación
- ✓ Modificación

Donde la Solución es:

- ✓ Normalización