



Componente formativo

Requisitos de producto

Breve descripción:

Los datos están en constante evolución y transformación en todo el mundo. Ha introducido nuevos mecanismos de mejora para todos los sistemas, también ha hecho que su operación sea más desafiante. El registro, la supervisión y el almacenamiento de datos precisos ayuda a abordar de una manera más eficaz estos desafíos y necesidades que contemplan los sistemas.

Área ocupacional:

Ciencias Naturales, aplicadas y relacionadas

Junio 2023

Tabla de contenido

Introducción.....	4
1. Evolución de las bases de datos.....	5
1.1 Tipos de bases de datos.....	11
1.2 Herramientas de “software” más conocidas para bases de datos.....	15
2 Fases del proceso de desarrollo de bases de datos.....	16
3 Entidades, relaciones y atributos	20
4 Modelo ER en base de datos.....	21
4.1 Entidad.....	22
4.2 Tipo de relación y conjunto de relaciones.....	24
5 Procedimientos más importantes en bases de datos.....	30
5.1 Índice de base de datos.....	31
5.2 Vistas en base de datos.....	36
5.3 Reportes o informes en bases de datos	40
5.4 Permisos, roles y control de acceso de usuarios.....	41
5.5 Optimización de bases de datos.....	46
5.6 Limpieza de la base de datos	47
5.7 Pruebas en la base de datos	49
6 MySQL.....	55
Síntesis	56
Material complementario	57
Glosario.....	58

Referencias bibliográficas	61
Créditos	62

Introducción

Los datos son una colección de una pequeña unidad de información distinta. Se puede usar en una variedad de formas como texto, números, medios, bytes, etc., se puede almacenar en hojas de papel o en memoria electrónica, etc. La palabra “datos” se origina a partir de la palabra “datum” que significa 'pieza única de información'. Es plural de la palabra datum. En informática, los datos son información que se puede traducir a una forma para un movimiento y procesamiento eficientes. Los datos son intercambiables.

Una base de datos es una colección organizada de datos, de modo que se puede acceder fácilmente y gestionar. Los datos se pueden organizar en tablas, filas, columnas e indexarlos para facilitar la búsqueda de información relevante. Los gestores de bases de datos crean una base de datos, de tal manera que solo un conjunto de programas de “software” proporciona acceso a los datos a todos los usuarios. El propósito principal de la base de datos es operar una gran cantidad de información almacenando, recuperando y administrando datos.

En la actualidad, existen muchos sitios web dinámicos en la “World Wide Web” que se manejan a través de bases de datos, por ejemplo, un modelo que comprueba la disponibilidad de habitaciones en un hotel. Es un ejemplo de un sitio web dinámico que utiliza una base de datos. Hay muchas bases de datos disponibles como “MySQL”, “Sybase”, “Oracle”, “MongoDB”, “Informix”, “PostgreSQL”, “SQL Server”, etc. Las bases de datos modernas son administradas por el sistema de administración de bases de datos (DBMS).

El SQL o lenguaje de consulta estructurado se utiliza para operar con los datos almacenados en una base de datos. Este depende del álgebra relacional y del cálculo relacional de tuplas. Para mostrar la imagen de una base de datos se utiliza una estructura cilíndrica como se muestra en la siguiente figura.

1. Evolución de las bases de datos

La base de datos ha completado más de 50 años de viaje desde su evolución del sistema de archivos planos a los sistemas relacionales y relacionales de objetos. Ha pasado por varias generaciones y esta evolución ha sido así:

1. Basado en archivos

En 1968 fue el año en que se introdujeron las bases de datos basadas en archivos. En las bases de datos basadas en archivos, los datos se mantenían en un archivo plano. Aunque los archivos tienen muchas ventajas, existen varias limitaciones. Una de las principales ventajas es que el sistema de archivos tiene varios métodos de acceso, por ejemplo, secuencial, indexado y aleatorio. Requiere una programación extensa en un lenguaje de tercera generación como COBOL, BASIC.

2. Modelo de datos jerárquico

En 1968-1980 fue la era de la base de datos jerárquica. El prominente modelo de base de datos jerárquico fue el primer DBMS de IBM. Se llamó IMS (Sistema de Gestión de la Información). En este modelo, ¿los archivos están relacionados de manera padre / hijo? Al igual que el sistema de archivos, este modelo también tenía algunas limitaciones, como una implementación compleja, falta de independencia estructural, no puede manejar fácilmente una relación de muchos a muchos, etc.

3. Modelo de datos de red

Charles Bachman desarrolló el primer DBMS en "Honeywell" llamado "Integrated Data Store" (IDS). Fue desarrollado a principios de la década de 1960; pero fue estandarizado en 1971 por el grupo CODASYL ("Conference on Data Systems Languages"). En este modelo, los archivos están relacionados como propietarios y miembros, como en el modelo de red común. El modelo de datos de red identificó los siguientes componentes:

1. Esquema de red (organización de la base de datos).
2. Subesquema (vistas de la base de datos por usuario).
3. Lenguaje de gestión de datos (procedimental).

Este modelo también tenía algunas limitaciones como la complejidad del sistema y era difícil de diseñar y mantener.

4. Base de datos relacional

En 1970 – presente, es la era de las bases de datos relacionales y la gestión de bases de datos. En 1970 EF Codd propuso el modelo relacional. El modelo de base de datos relacional tiene dos terminologías principales llamadas instancia y esquema. La instancia es una tabla con filas o columnas. El esquema especifica la estructura como nombre de la relación, tipo de cada columna y nombre. Este modelo utiliza algún concepto matemático como la teoría de conjuntos y la lógica de predicados. La primera aplicación de base de datos de Internet se creó en 1995. Durante la era de la base de datos relacional se habían introducido muchos más modelos como el modelo orientado a objetos, el modelo relacional de objetos, etc.

5. Base de datos en la nube

Le permite almacenar, administrar y recuperar sus datos estructurados y no estructurados a través de una plataforma en la nube. Se puede acceder a estos datos a través de Internet. Las bases de datos en la nube también se denominan base de datos como servicio (DBaaS) porque se ofrecen como un servicio administrado. Algunas de las mejores opciones en la nube son:

1. AWS (servicios “web” de “Amazon”).
2. Computación de copo de nieve.
3. Servicios de “Oracle Database Cloud”.
4. Servidor “Microsoft SQL”.
5. Llave de nube de Google.

6. Ventajas de la base de datos en la nube

1. Costos más bajos: generalmente, el proveedor de la empresa no tiene que invertir en bases de datos. Puede mantener y dar soporte a uno o más centros de datos.
2. Automatizado: las bases de datos en la nube están enriquecidas con una variedad de procesos automatizados, como recuperación, conmutación por error y escalado automático.
3. Mayor accesibilidad: puede acceder a su base de datos basada en la nube desde cualquier ubicación en cualquier momento. Todo lo que necesita es una conexión a Internet.

7. Base de datos NoSQL

Es un enfoque para diseñar tales bases de datos que pueden adaptarse a una amplia variedad de modelos de datos. NoSQL significa "no solo SQL". Es una alternativa a las bases de datos relacionales tradicionales en las que los datos se colocan en tablas y el esquema de datos está perfectamente diseñado antes de crear la base de datos. Las bases de datos NoSQL son útiles para un gran conjunto de datos distribuidos. Algunos ejemplos de sistema de base de datos NoSQL con su categoría son:

1. MongoDB, CouchDB, Cloudant (basado en documentos).
2. Memcached, Redis, Coherence (almacén de valores-clave).
3. HBase, mesa grande, acumulación (tabular).

8. Ventaja de NoSQL

1. Alta escalabilidad: NoSQL puede manejar una gran cantidad de datos debido a su escalabilidad. Si los datos crecen, la base de datos NoSQL la escala para manejar esos datos de manera eficiente.
2. Alta disponibilidad: NoSQL admite la replicación automática. La replicación automática lo hace altamente disponible porque en caso de falla, los datos se replican al estado consistente anterior.

9. Desventaja de NoSQL

1. Fuente abierta: NoSQL es una base de datos de código abierto, por lo que todavía no existe un estándar confiable para NoSQL.
2. Desafío de gestión: la gestión de datos en NoSQL es mucho más complicada que las bases de datos relacionales. Es muy difícil de instalar y aún más complicado de administrar a diario.
3. La GUI no está disponible: las herramientas de GUI para la base de datos NoSQL no están fácilmente disponibles en el mercado.
4. Respaldo: la copia de seguridad es un gran punto débil para las bases de datos NoSQL. Algunas bases de datos como MongoDB no tienen enfoques potentes para la copia de seguridad de datos.

10. Las bases de datos orientadas a objetos

Contienen datos en forma de objetos y clases. Los objetos son la entidad del mundo real y los tipos son la colección de objetos. Una base de datos orientada a objetos es una combinación de características del modelo relacional con principios orientados a objetos. Es una implementación alternativa a la del modelo relacional. Las bases de datos orientadas a objetos mantienen las reglas de la programación orientada a objetos. Un sistema de gestión de bases de datos orientado a objetos es una aplicación híbrida. El modelo de base de datos orientada a objetos contiene las siguientes propiedades.

11. Propiedades de programación orientada a objetos

1. Objetos.
2. Clases.
3. Herencia.
4. Polimorfismo.
5. Encapsulamiento.

12. Propiedades de la base de datos relacional

1. Atomicidad.
2. Consistencia.
3. Integridad.

4. Durabilidad.
5. Concurrencia.
6. Procesamiento de consultas.

13. Bases de datos de gráficos

Es una base de datos NoSQL. Es una representación gráfica de datos. Contiene nodos y aristas. Un nodo representa una entidad y cada borde representa una relación entre dos bordes. Cada nodo de una base de datos gráfica representa un identificador único. Las bases de datos de gráficos son beneficiosas para buscar la relación entre los datos porque resaltan la relación entre los datos relevantes. Las bases de datos de gráficos son muy útiles cuando la base de datos contiene una relación compleja y un esquema dinámico. Se utiliza principalmente en la gestión de la cadena de suministro, identificando la fuente de telefonía IP.

DBMS (Sistema de gestión de bases de datos)

El sistema de gestión de bases de datos es un “software” que se utiliza para almacenar y recuperar la base de datos, por ejemplo, Oracle, MySQL, etc.; estas son algunas herramientas populares de DBMS.

- a. DBMS proporciona la interfaz para realizar las diversas operaciones como creación, eliminación, modificación, etc.
- b. DBMS permite al usuario crear sus bases de datos según sus necesidades.
- c. DBMS acepta la solicitud de la aplicación y proporciona datos específicos a través del sistema operativo.
- d. DBMS contiene el grupo de programas que actúa de acuerdo con las instrucciones del usuario.
- e. Proporciona seguridad a la base de datos.

Ventaja de DBMS

- a. Controla la redundancia: almacena todos los datos en un solo archivo de base de datos, por lo que puede controlar la redundancia de datos.

- b. Compartir datos: un usuario autorizado puede compartir los datos entre varios usuarios.
- c. Respaldo: proporciona un subsistema de copia de seguridad y recuperación. Este sistema de recuperación crea datos automáticos a partir de fallas del sistema y restaura los datos si es necesario.
- d. Varias interfaces de usuario: proporciona un tipo diferente de interfaces de usuario como GUI, interfaces de aplicaciones.

Desventaja de DBMS

- a. Tamaño: ocupa un gran espacio en disco y una gran memoria para funcionar de manera eficiente.
- b. Costo: DBMS requiere un procesador de datos de alta velocidad y una memoria más grande para ejecutar el “software” DBMS, por lo que es costoso.
- c. Complejidad: DBMS crea requisitos y complejidad adicionales.

RDBMS (Sistema de gestión de bases de datos relacionales)

Se representa como una tabla que contiene filas y columnas. RDBMS se basa en el modelo relacional; fue presentado por EF Codd. Una base de datos relacional contiene los siguientes componentes:

- a. Mesa.
- b. Grabar / tupla.
- c. Campo / nombre de columna / atributo.
- d. Ejemplo.
- e. Esquema.
- f. Teclas.

Un RDBMS es un DBMS tabular que mantiene la seguridad, integridad, precisión y consistencia de los datos.

1.1 Tipos de bases de datos

Hay varios tipos de bases de datos que se utilizan para almacenar diferentes variedades de datos. A través del siguiente recurso se invita a explorar los tipos de bases de datos.

Ventajas de la base de datos NoSQL

- a. Permite una buena productividad en el desarrollo de aplicaciones, ya que no es necesario almacenar datos en un formato estructurado.
- b. Es una mejor opción para administrar y manejar grandes conjuntos de datos.
- c. Proporciona alta escalabilidad.
- d. Los usuarios pueden acceder rápidamente a los datos de la base de datos a través de valores clave.

Base de datos en la nube:

Un tipo de base de datos donde los datos se almacenan en un entorno virtual y se ejecutan sobre la plataforma de computación en la nube. Proporciona a los usuarios varios servicios de computación en la nube (SaaS, PaaS, IaaS, etc.) para acceder a la base de datos.

Existen numerosas plataformas en la nube, pero las mejores opciones son:

- a. Servicios “web” de “Amazon” (AWS).
- b. “Microsoft Azure”.
- c. “Kamatera”.
- d. “PhonixNAP”.
- e. “ScienceSoft”.
- f. “Google Cloud SQL”, etc.

Bases de datos orientadas a objetos

El tipo de base de datos que utiliza el enfoque del modelo de datos basado en objetos para almacenar datos en el sistema de base de datos. Los datos se representan y almacenan como objetos que son similares a los objetos utilizados en el lenguaje de programación orientado a objetos.

Bases de datos jerárquicas

Es el tipo de base de datos que almacena datos en forma de nodos de relación padre-hijo. Aquí, organiza los datos en una estructura en forma de árbol. Los datos se almacenan en forma de registros que están conectados a través de enlaces. Cada registro secundario del árbol contendrá solo un padre. Por otro lado, cada registro principal puede tener varios registros secundarios.

Bases de datos de red

Es la base de datos que normalmente sigue el modelo de datos de red. Aquí, la representación de los datos se realiza en forma de nodos conectados a través de enlaces entre ellos. A diferencia de la base de datos jerárquica, permite que cada registro tenga varios nodos secundarios y principales para formar una estructura gráfica generalizada.

Base de datos relacional

Esta base de datos se centra en el modelo de datos relacionales, que almacena datos en forma de filas (tupla) y columnas (atributos), y juntos forman una tabla (relación). Una base de datos relacional utiliza SQL para almacenar, manipular y mantener los datos. EF Codd inventó la base de datos en 1970. Cada tabla de la base de datos lleva una clave que hace que los datos sean únicos, los demás ejemplos de bases de datos relacionales son "MySQL", "Microsoft SQL Server", "Oracle", etc.

Propiedades de la base de datos relacional: existen las siguientes cuatro propiedades comúnmente conocidas de un modelo relacional denominadas propiedades ACID, donde:

A de “atomicity” que significa atomicidad: asegura que la operación de datos se completará con éxito o con fracaso. Sigue la estrategia de “todo o nada”, por ejemplo, una transacción se confirmará o se abortará.

C de “consistency” que significa consistencia: si se realiza alguna operación sobre los datos se debe conservar su valor antes y después de la operación, por ejemplo, el saldo de la cuenta antes y después de la transacción debe ser correcto, es decir, debe permanecer conservado.

I de “isolation” que significa aislamiento: puede haber usuarios concurrentes para acceder a los datos al mismo tiempo desde la base de datos. Por lo tanto, el aislamiento entre los datos debe permanecer aislado, por ejemplo, cuando ocurren múltiples transacciones al mismo tiempo los efectos de una transacción no deben ser visibles para las otras transacciones en la base de datos.

D de “durability” que significa durabilidad: asegura que una vez que completa la operación y confirma los datos, los cambios de datos deben quedar permanentes.



Base de datos NoSQL

NonSQL / Not Only SQL es un tipo de base de datos que se utiliza para almacenar una amplia gama de conjuntos de datos. No es una base de datos relacional, ya que almacena datos no solo en forma tabular, sino de varias formas diferentes. Surgió cuando aumentó la demanda de crear aplicaciones modernas, por lo tanto, NoSQL presentó una amplia variedad de tecnologías de bases de datos en respuesta a las demandas; se puede dividir aún más una base de datos NoSQL en los siguientes cuatro tipos:

- Almacenamiento de clave-valor: es el tipo más simple de almacenamiento de base de datos donde almacena cada elemento individual como una clave (o nombre de atributo) que mantiene su valor juntos.
- Base de datos orientada a documentos: un tipo de base de datos que se utiliza para almacenar datos como documentos similares a JSON. Ayuda a los desarrolladores a almacenar datos utilizando el mismo formato de modelo de documento que se utiliza en el código de la aplicación.
- Bases de datos de gráficos: se utiliza para almacenar grandes cantidades de datos en una estructura similar a un gráfico. Por lo general, los sitios web de redes sociales utilizan la base de datos de gráficos.
- Almacenes de columna ancha: es similar a los datos representados en las bases de datos relacionales. Aquí, los datos se almacenan juntos en columnas grandes, en lugar de almacenarse en filas. Ver figura 1.

Figura 1. Bases de datos SQL y NoSQL



Base de datos personal

La recopilación y el almacenamiento de datos en el sistema del usuario define una base de datos personal. Dicha base de datos está diseñada básicamente para un solo usuario.

Ventaja de la base de datos personal

- a. Es simple y fácil de manejar.
- b. Ocupa menos espacio de almacenamiento, ya que es de tamaño pequeño.

Base de datos operativa

El tipo de base de datos que crea y actualiza la base de datos en tiempo real. Básicamente está diseñada para ejecutar y manejar las operaciones diarias de datos en varios negocios, por ejemplo, una organización utiliza bases de datos operativas para administrar transacciones diarias.

Base de datos empresarial

Las grandes organizaciones o empresas utilizan esta base de datos para gestionar una gran cantidad de datos. Ayuda a las organizaciones a aumentar y mejorar su eficiencia. Una base de datos de este tipo permite el acceso simultáneo a los usuarios.

Ventajas de la base de datos empresarial

- a. Los procesos múltiples son compatibles con la base de datos empresarial.
- b. Permite ejecutar consultas paralelas en el sistema.

1.2 Herramientas de “software” más conocidas para bases de datos

A continuación, puede ver las herramientas más comunes.



2 Fases del proceso de desarrollo de bases de datos

Serie de fases en las que cada una de ellas tiene su propia experiencia. Las fases principales son la planificación, el análisis de requisitos y la implementación; cada proceso se divide en pasos. El diagrama de estas fases se muestra como:



Se invita a explorar el siguiente componente donde hallaras las fases del proceso de desarrollo de las bases de datos.

Fase 1 - Planificación

La fase de planificación comienza cuando un cliente solicita desarrollar un proyecto de base de datos. Es un conjunto de tareas o actividades que deciden el recurso requerido en el

desarrollo de la base de datos y el tiempo límite de las diferentes actividades. Durante la fase de planificación se realizan cuatro actividades principales:

- a. Revise y apruebe la solicitud del proyecto de la base de datos.
- b. Priorice el proyecto de la base de datos.
- c. Asigne recursos como dinero, personas, herramientas, etc.
- d. Organice un equipo de desarrollo para desarrollar el proyecto de la base de datos.

Fase 2 - Análisis de requisitos

El análisis de requisitos se realiza para comprender un problema que no se resuelve. Es una actividad muy importante para el desarrollo del sistema de base de datos en el ciclo de vida de desarrollo de la base de datos. La persona responsable de la identificación de análisis de requisitos se llama analista. En la fase de análisis de requisitos se recopilan y analizan los requisitos y las expectativas de los usuarios. Los requisitos recopilados ayudan a comprender el sistema lo que aún no existe en la forma requerida. Hay dos actividades principales en el análisis de requisitos. Estos son:

- a. Comprensión o análisis de problemas.
- b. Especificación de requerimiento.

Pasos del análisis

Existen los siguientes pasos incluidos en el análisis de requisitos:

- a. Desarrollar un modelo de datos conceptual.
- b. Desarrollar modelo de proceso.
- c. Diagrama de flujo de datos.

Fase 3 - Diseño

El diseño es la fase principal del proceso de desarrollo de bases de datos. En esta fase, el modelo de información que desarrolla durante el análisis se utiliza para diseñar un

esquema conceptual para la base de datos y para diseñar la aplicación. Es una fase importante de las fases del proceso de desarrollo de bases de datos.

En el diseño del esquema conceptual se examinan los requisitos de datos recopilados en la fase de análisis de requisitos y se produce un esquema de base de datos conceptual que es más importante en la fase de diseño del proceso de desarrollo de la base de datos. En el diseño de la sección y de la aplicación se examina la aplicación de la base de datos analizada en la fase de análisis de requisitos y se produce la especificación de esta aplicación. Los dos pasos principales en la fase de diseño son:

- a. Diseño de base de datos.
- b. Diseño de procesos.

Fase 4 - Selección DBMS

En esta fase se selecciona el DBMS apropiado para apoyar el sistema de información. Varios factores están involucrados en la selección de DBMS. Pueden ser factores técnicos y económicos.

Factor técnico: los factores técnicos están relacionados con la idoneidad del DBMS para el sistema de información. Se consideran los siguientes factores técnicos:

- a. Tipo de DBMS como relacional, orientado a objetos, etc.
- b. Estructura de almacenamiento y métodos de acceso que admite el DBMS en la base de datos.
- c. Interfaces disponibles para usuarios y programadores.
- d. Tipos de lenguaje de consulta.
- e. Herramientas de desarrollo.

Factores financieros: están relacionados con el costo del DBMS y su operación. En este caso se consideran los siguientes costos

- a. Costo del "software".

- b. Costo de mantenimiento.
- c. Costo de "hardware".
- d. Costo de creación y mantenimiento de la base de datos.
- e. Costo de formación.

Fase 5 - Implementación

Después de la fase de diseño y la selección de un DBMS adecuado, el sistema de base de datos se implementa o se instala para su uso. El propósito de esta fase es instalar el sistema de información de acuerdo con el plan y diseño descrito en la fase anterior. La implementación implica una serie de pasos que conducen al sistema de información operativo, que incluye la creación de definiciones de base de datos, desarrollo de aplicaciones, prueba del sistema, desarrollo de procedimientos operativos y documentación, capacitación de usuarios y llenado de bases de datos en el mismo archivo de datos de la misma organización. Es una fase importante de fases en el proceso de desarrollo de bases de datos. La implementación implica dos pasos, estos son:

- a. Definición de base de datos.
- b. Creando aplicación.

Fase 6 - Mantenimiento

Una vez implementada la base de datos comienza la fase de mantenimiento del sistema de base de datos. El mantenimiento es el proceso de seguimiento y mantenimiento del sistema de base de datos. El mantenimiento incluye actividades como agregar nuevos campos, cambiar el tamaño del archivo existente, agregar nuevas tablas, etc.

A medida que cambia el requisito del sistema de la base de datos es necesario agregar una tabla nueva o eliminar tablas existentes y reconocer algunos archivos cambiando los métodos de acceso primario o eliminando índices antiguos y construyendo otros nuevos. Es posible que algunos requisitos se reescriban para un mejor rendimiento. El ajuste o la reorganización de la base de datos, continúa a lo largo de la vida de la base de datos, mientras que los requisitos siguen cambiando.

3 Entidades, relaciones y atributos

Por medio del siguiente recurso explore todo sobre entidades, relaciones y atributos.

1. La entidad

La entidad de la base de datos es una cosa, persona, lugar, unidad, objeto o cualquier elemento sobre el cual los datos deben capturarse y almacenarse en forma de propiedades, flujo de trabajo y tablas. Si bien el flujo de trabajo y las tablas son opcionales para la entidad de base de datos, las propiedades son obligatorias (porque la entidad sin propiedades no es una entidad)

2. Propiedades de la entidad

Las propiedades de la entidad son atributos clave en una base de datos. La diferencia entre entidad y propiedad se determina según si el elemento es de interés directo para la base de datos. Por ejemplo, en una base de datos de correo electrónico, la dirección de correo electrónico es relevante, mientras que, en una base de datos de clientes, el cliente es la entidad principal y la dirección de correo electrónico es una propiedad asociada al cliente.

Algunas de las propiedades de la entidad pueden representar relaciones de entidad de base de datos, es decir, relacionar una entidad con otra entidad, por ejemplo, la entidad “trabajo” puede tener la propiedad “cliente” que se refiere a la entidad “cliente”, para que cuando cree un nuevo trabajo pueda desplegar la lista de clientes y seleccionar un cliente para el que se realizará ese trabajo.

3. Tablas de entidades

Son una especie de subentidad dentro de la entidad principal (porque la tabla de entidad tiene sus propias propiedades), por ejemplo, la entidad “pedido” que tiene propiedades como ID de pedido, fecha de pedido, etc., puede tener una tabla “artículos” que tiene sus propias propiedades como nombre de producto, cantidad, precio, etc.

Además, a veces puede ser difícil decidir si es necesario crear una tabla de entidad o agregar varias propiedades a la propia entidad, por ejemplo, para la entidad “contacto”

necesita capturar información de dirección que incluya propiedades “calle”, “ciudad”, “estado”, etc. Si cada contacto tiene solo una dirección, entonces estas propiedades pueden pertenecer a la entidad, mientras que, si el contacto puede tener varias direcciones, estas propiedades deben pertenecer a la tabla de entidades.

4. Flujo de trabajo de la entidad

Es una especie de ciclo de vida de la entidad que consta de estados, transiciones y razones por las que la entidad pasa de un estado a otro, por ejemplo, cuando los usuarios cambian el estado de la entidad “tarea” de “en curso” a “cerrado”, pueden seleccionar uno de los motivos disponible: “completado” o “cancelado”.

Además, el flujo de trabajo puede tener acciones realizadas automáticamente cuando tiene lugar la transición entre los estados, por ejemplo, si el estado de la entidad “oportunidad” cambia de “abierto” a “cerrado”, el sistema ingresará automáticamente la fecha y hora actuales en el campo “fecha de cierre”.

Para mayor comprensión diríjase al:

Para mayor comprensión, diríjase al Anexo 1 - Relaciones en una base de datos. [Clic aquí.](#)

4 Modelo ER en base de datos

Se utiliza para modelar la vista lógica del sistema desde la perspectiva de los datos, que consta de estos componentes:

- a. Entidad.
- b. Tipo de entidad.
- c. Conjunto de entidades.

4.1 Entidad

Puede ser un objeto con existencia física - una persona, automóvil, casa o empleado en particular - o puede ser un objeto con existencia conceptual - una empresa, un trabajo o un curso universitario.

Una entidad es un objeto de tipo de entidad y el conjunto de todas las entidades se denomina conjunto de entidades, por ejemplo, E1 es una entidad que tiene el tipo de entidad estudiante y el conjunto de todos los estudiantes se denomina conjunto de entidades. En el diagrama ER, el tipo de entidad se representa como:



Atributos:

Son las propiedades que definen el tipo de entidad, por ejemplo, No. ID, nombre, DOB, edad, dirección, No. de teléfono son los atributos que definen el tipo de entidad estudiante. En el diagrama ER el atributo está representado por un óvalo.



Atributo clave:

Identifica de manera única a cada entidad en el conjunto de entidades, por ejemplo, No. ID será único para cada estudiante. En el diagrama ER el atributo clave está representado por un óvalo con líneas subyacentes.



Atributo compuesto:

Está compuesto por muchos otros atributos, por ejemplo, el atributo dirección del tipo de entidad del estudiante consta de calle, ciudad, departamento y país. En el diagrama ER, el atributo compuesto está representado por un óvalo que consta de óvalos.



Atributo multivalor:

Consta de más de un valor para una entidad determinada, por ejemplo, el número de teléfono (puede ser más de uno para un estudiante determinado). En el diagrama ER, el atributo multivalor está representado por un óvalo doble.



Atributo derivado:

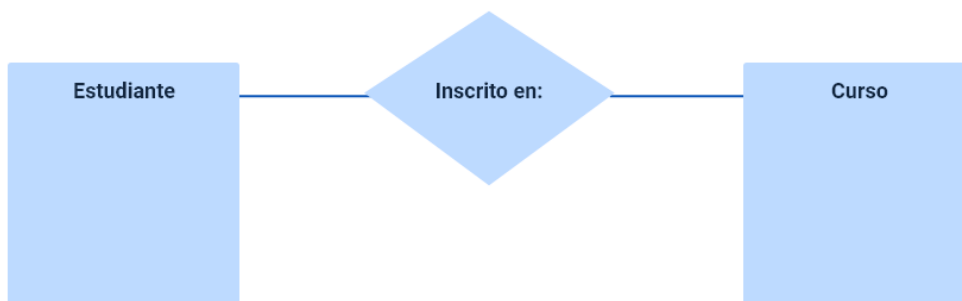
Un atributo que puede derivarse de otros atributos del tipo de entidad, por ejemplo, edad (se puede derivar de DOB). En el diagrama ER, el atributo derivado está representado por un óvalo punteado.



4.2 Tipo de relación y conjunto de relaciones

A partir de este recurso podrá descubrir los tipos de relación y el conjunto de relaciones.

Un tipo de relación representa la asociación entre tipos de entidad, por ejemplo, “inscrito en” es un tipo de relación que existe entre el tipo de entidad estudiante y curso. En el diagrama ER, el tipo de relación está representado por un diamante y conecta las entidades con líneas.



Representación de Tipo de relación según diagrama ER.

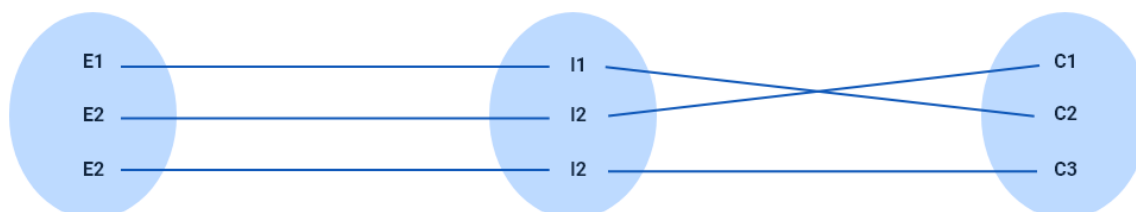
Estudiante

Inscrito en:

Curso

Representación de Tipo de relación según diagrama ER.

Un conjunto de relaciones del mismo tipo se conoce como conjunto de relaciones. El siguiente conjunto de relaciones muestra que S1 está inscrito en C2, S2 está inscrito en C1 y S3 está inscrito en C3.

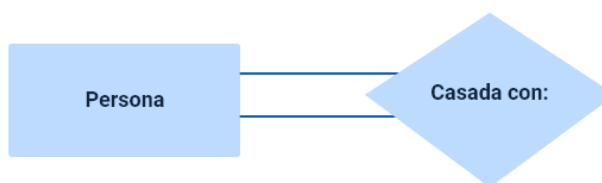


Representación de Conjunto de relación según diagrama ER.

Representación de Conjunto de relación según diagrama ER.

Grado de un conjunto de relaciones: el número de conjuntos de entidades diferentes que participan en un conjunto de relaciones se denomina grado de un conjunto de relaciones.

Relación unaria: cuando solo hay un conjunto de entidades que participa en una relación, la relación se denomina relación unaria, por ejemplo, una persona está casada con una sola persona.



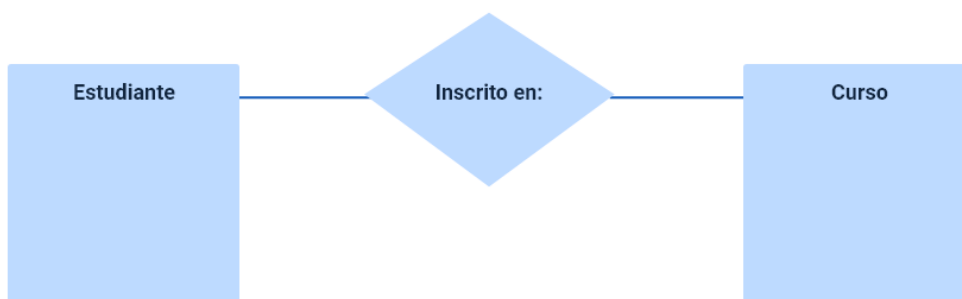
Representación de Relación unaria según diagrama ER.

Persona

Casada con:

Representación de Relación unaria según diagrama ER.

Relación binaria: cuando hay dos entidades establecidas que participan en una relación, la relación se denomina relación binaria, por ejemplo, el alumno está inscrito en el curso.



Representación de Relación binaria según diagrama ER.

Estudiante

Inscrito en:

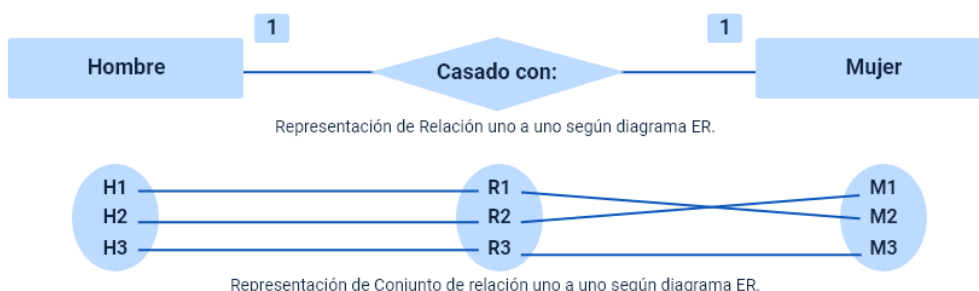
Curso

Representación de Relación binaria según diagrama ER.

Relación n-aria: Cuando hay un conjunto de “n” entidades que participan en una relación, la relación se denomina relación n-aria.

Cardinalidad: El número de veces que una entidad de un conjunto de entidades participa en un conjunto de relaciones se conoce como cardinalidad. La cardinalidad puede ser de diferentes tipos:

Uno a uno: cuando cada entidad de cada conjunto de entidades puede participar sólo una vez en la relación, la cardinalidad es uno a uno. Supongamos que un hombre puede casarse con una mujer y una mujer puede casarse con un hombre. Entonces la relación será uno a uno.

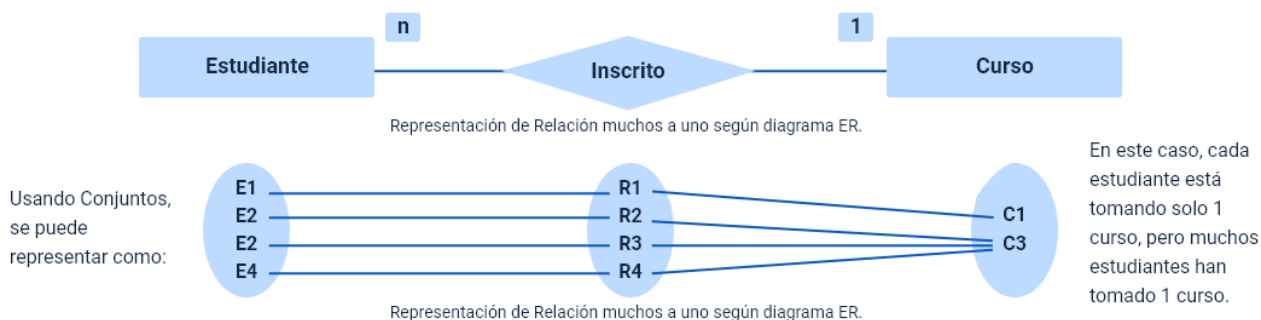


Hombre 1 Casado con: 1 Mujer

Representación de Relación uno a uno según diagrama ER.

Representación de Conjunto de relación uno a uno según diagrama ER.

Muchos a uno: Cuando las entidades de un conjunto de entidades solo pueden participar una vez en el conjunto de relaciones y las entidades de otro conjunto de entidades pueden participar más de una vez en el conjunto de relaciones, la cardinalidad es de muchos a uno. Supongamos que un estudiante solo puede tomar un curso, pero muchos estudiantes pueden tomar un curso. Entonces, la cardinalidad será “n” a 1. Significa que para un curso puede haber “n” estudiantes, pero para un estudiante, solo habrá un curso.



Estudiante n Inscrito 1 Curso

Representación de Relación muchos a uno según diagrama ER.

Usando conjuntos de puede representar como:

E1, E2, E2, E4

R1,R2,R3,R4

C1,C3

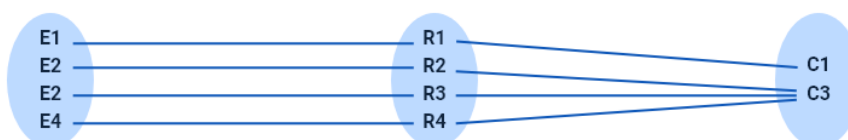
En este caso, cada estudiante esta tomado solo 1 curso, pero muchos estudiantes han tomado 1 curso.

Representación de Relación muchos a uno según diagrama ER.

De muchos a muchos: cuando las entidades de todos los conjuntos de entidades pueden participar más de una vez en la relación, la cardinalidad es de muchos a muchos. Suponga que un estudiante puede tomar más de un curso y que muchos estudiantes pueden tomar un curso. Entonces, la relación será de muchos a muchos.



Usando conjuntos se puede representar como:



En este ejemplo, el estudiante S1 está inscrito en C1 y C3 y el curso C3 está inscrito en S1, S3 y S4. Así que son relaciones de muchas a muchas.

Estudiante m Inscrito en: n Curso

Usando conjuntos de puede representar como:

E1, E2, E2, E4

R1,R2,R3,R4

C1,C3

En este ejemplo, el estudiante S1 está inscrito en C1 y C3 y el curso C3 está inscrito en S1, S3 y S4 así que son relaciones de muchas a muchas.

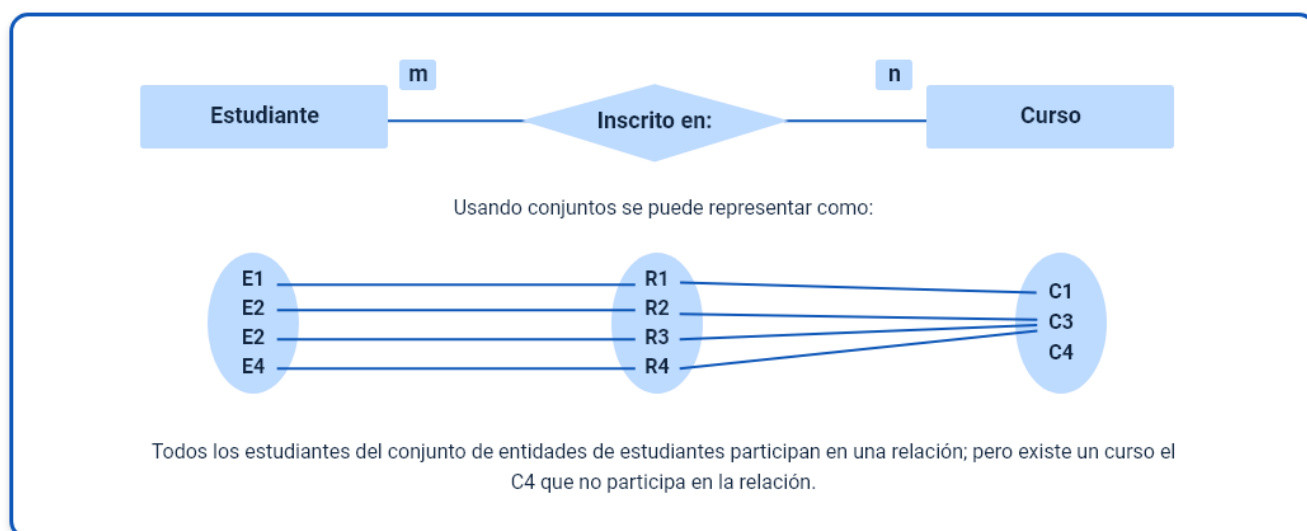
Restricción de participación

Se aplica a la entidad que participa en el conjunto de relaciones.

- Participación total:** cada entidad del conjunto de entidades debe participar en la relación. Si cada alumno debe matricularse en un curso, la participación del alumno será total. La participación total se muestra con una línea doble en el diagrama ER.
- Participación parcial:** la entidad en el conjunto de entidades puede o no participar en la relación. Si algunos cursos no están matriculados por ninguno de los alumnos, la participación del curso será parcial.

El diagrama muestra el conjunto de relaciones “inscrito en” con el conjunto de entidad de estudiante con participación total y el conjunto de entidad del curso con participación parcial.

Figura 2. Diagrama conjunto de relaciones.



Estudiante m Inscrito en: n Curso

Usando conjunto se puede representar como:

E1, E2, E2, E4

R1,R2,R3,R4

C1,C3

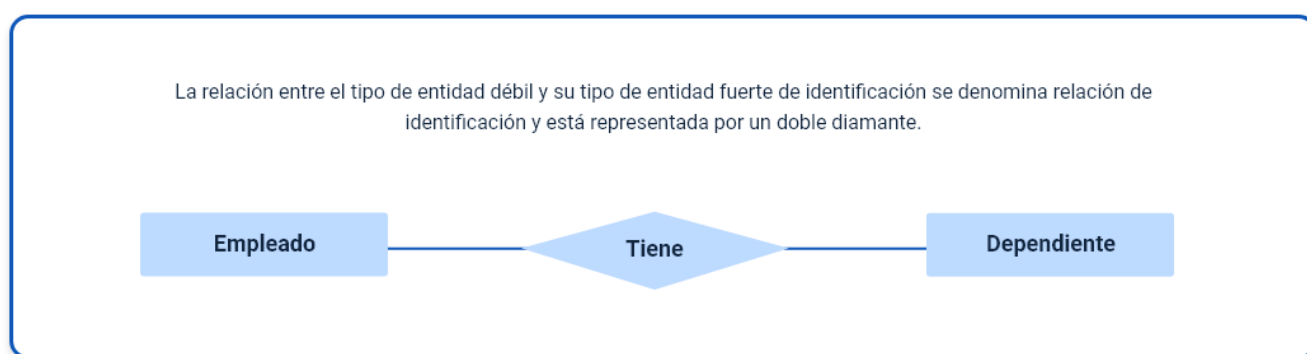
Tolos los estudiantes del conjunto de entidades de estudiantes en una relación, pero existe un curso en el C4 que no participa en la relación.

Tipo de entidad débil y relación de identificación

Como se discutió anteriormente, un tipo de entidad tiene un atributo clave que identifica de manera única a cada entidad en el conjunto de entidades; pero existe algún tipo de entidad para la cual no se puede definir un atributo clave, este se denomina tipo de entidad débil, por ejemplo, una empresa puede almacenar la información de los dependientes (padres, hijos, cónyuge) de un empleado; pero los dependientes no tienen existencia sin el empleado. Por lo tanto, el dependiente será un tipo de entidad débil y el empleado será el tipo de entidad de identificación para el dependiente.

Un tipo de entidad débil está representado por un rectángulo doble, la participación del tipo de entidad débil es siempre total.

Figura 3. Diagrama relación de identificación.



5 Procedimientos más importantes en bases de datos

Los procedimientos de la base de datos son compilados, almacenados y administrados por el servidor DBMS. Los procedimientos de la base de datos se pueden

utilizar junto con las reglas para hacer cumplir la integridad de la base de datos o para realizar operaciones que se repiten con frecuencia. Cuando se crea el procedimiento se guarda su plan de ejecución, reduciendo la sobrecarga de ejecución.

Los procedimientos de la base de datos se pueden crear de forma interactiva o en un programa integrado. Un procedimiento de base de datos se puede ejecutar en un programa de lenguaje host, en el monitor de terminal, en otro procedimiento de base de datos o en un programa 4GL. Los procedimientos de la base de datos también pueden invocarse mediante reglas.



5.1 Índice de base de datos

En este espacio conocerá todo sobre el índice de las bases de datos.

1. Índice de base de datos

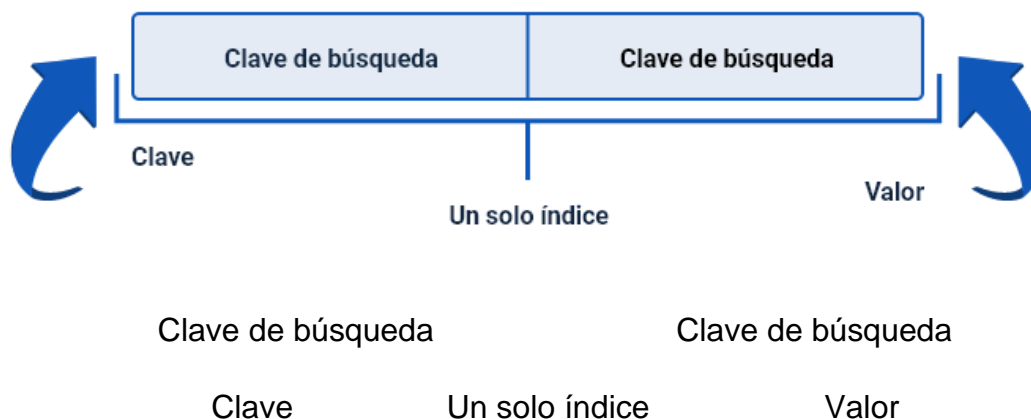
La indexación es una forma de optimizar el rendimiento de una base de datos minimizando el número de accesos al disco necesarios cuando se procesa una consulta. Es una técnica de estructura de datos que se utiliza para localizar y acceder rápidamente a los datos en una base de datos.

2. Los índices se crean utilizando algunas columnas de la base de datos

a. La primera columna es la clave de búsqueda que contiene una copia de la clave principal o clave candidata de la tabla. Estos valores se almacenan en orden para que se pueda acceder rápidamente a los datos correspondientes.

Nota: los datos pueden o no almacenarse en orden.

- b. La segunda columna es la referencia de datos o puntero que contiene un conjunto de punteros que contienen la dirección del bloque de disco donde se puede encontrar ese valor de clave en particular.



3. La indexación tiene varios atributos

- a. Tipos de acceso: se refiere al tipo de acceso, como búsqueda basada en valores, acceso a rango, etc.
- b. Tiempo de acceso: se refiere al tiempo necesario para encontrar un elemento de datos o un conjunto de elementos en particular.
- c. Tiempo de inserción: se refiere al tiempo que se tarda en encontrar el espacio adecuado e insertar un nuevo dato.
- d. Tiempo de eliminación: tiempo necesario para encontrar un elemento y eliminarlo, así como para actualizar la estructura del índice.
- e. Sobrecarga de espacio: se refiere al espacio adicional requerido por el índice.

En general, existen dos tipos de mecanismos de organización de archivos que son seguidos por los métodos de indexación para almacenar los datos.

4. Organización de archivo secuencial o archivo de índice ordenado

En este, los índices se basan en un orden de los valores. Por lo general, son un mecanismo de almacenamiento rápido y más tradicional. Estas organizaciones de archivos ordenadas o secuenciales pueden almacenar los datos en un formato denso o disperso:

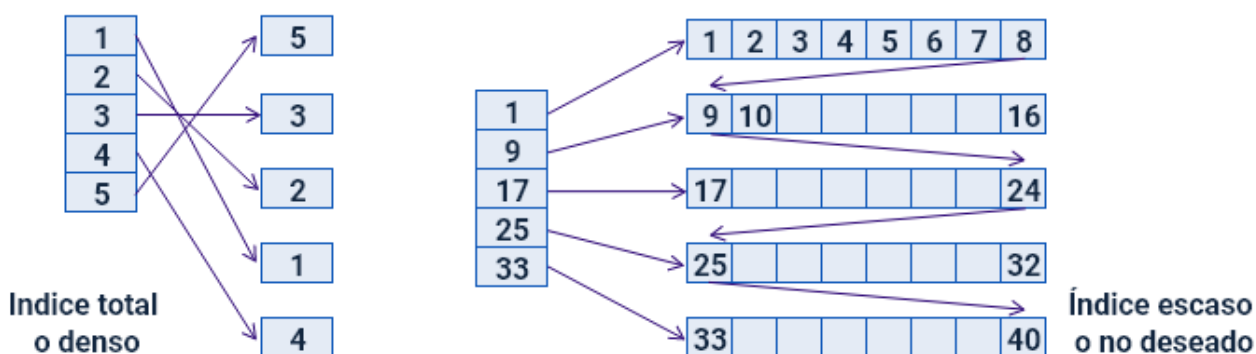
Índice denso:

- Para cada valor de clave de búsqueda en el archivo de datos hay un registro de índice.
- Este registro contiene la clave de búsqueda y también una referencia al primer registro de datos con ese valor de clave de búsqueda.

5. Índice disperso

El registro de índice aparece solo para algunos elementos en el archivo de datos. Cada elemento apunta a un bloque como se muestra.

- Para localizar un registro se busca el registro de índice con el valor de clave de búsqueda más grande, menor o igual al valor de clave de búsqueda que se está buscando.
- Comienza en ese registro al que apunta el registro de índice y se sigue con los punteros en el archivo, (es decir, secuencialmente) hasta encontrar el registro deseado.



6. Organización del archivo hash

Los índices se basan en los valores que se distribuyen uniformemente en una variedad de depósitos. Los depósitos a los que se asigna un valor se determinan mediante una función denominada función hash. Hay principalmente tres métodos de indexación:

- a. Indexación agrupada.
- b. Indexación secundaria o no agrupada.
- c. Indexación multinivel.

7. Indexación agrupada:

Cuando se almacenan más de dos registros en el mismo archivo, estos tipos de almacenamiento se conocen como indexación agrupada. Al usar la indexación de clústeres, se puede reducir el costo de búsqueda, ya que se almacenan múltiples registros relacionados con lo mismo en un solo lugar y también brinda la unión frecuente de más de dos tablas (registros). Básicamente, los registros con características similares se agrupan y se crean índices para estos grupos.

Índice agrupado ordenado según el nombre (tecla de búsqueda)

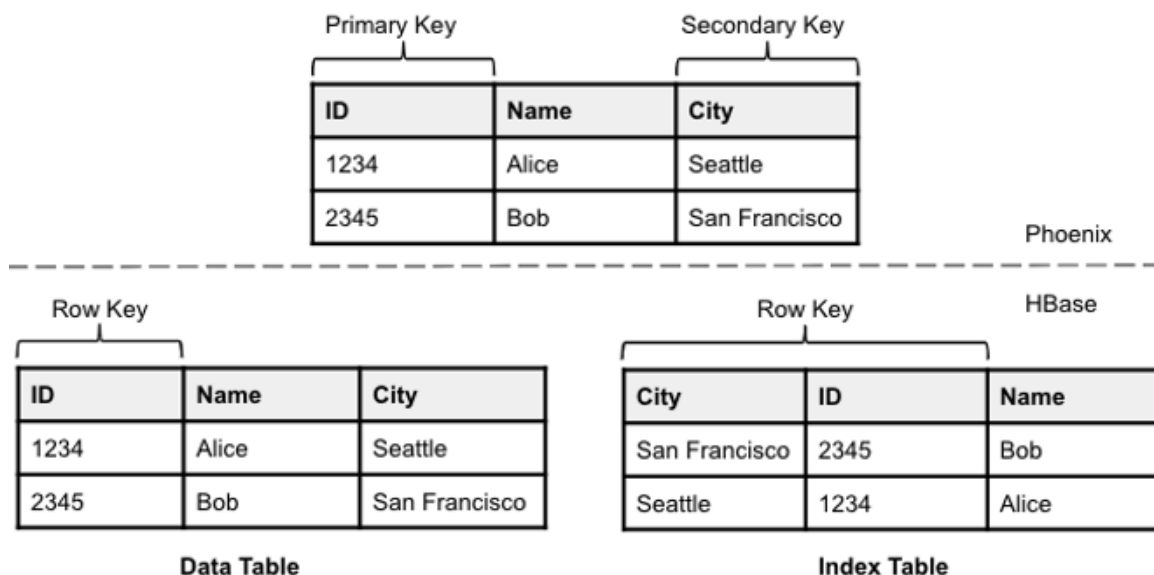
Indexación principal: este es un tipo de indexación agrupada en el que los datos se ordenan de acuerdo con la clave de búsqueda y la clave principal de la tabla de la base de datos y se utiliza para crear el índice. Es un formato predeterminado de indexación donde induce la organización secuencial de archivos; como las claves primarias son únicas y se almacenan ordenadas, el rendimiento de la operación de búsqueda es bastante eficiente.



Indexación secundaria o no agrupada: un índice no agrupado simplemente dice dónde se encuentran los datos, es decir, da una lista de punteros virtuales o referencias a la ubicación donde se almacenan realmente los datos. Los datos no se almacenan físicamente en el orden del índice. En cambio, los datos están presentes en los nodos hoja, por ejemplo, la página de contenido de un libro; cada entrada da el número de página o la ubicación de la información almacenada. Los datos reales aquí (información en cada página del libro) no están organizados; pero se tiene una referencia ordenada (página de contenido) de dónde se encuentran realmente los puntos de datos. Solo se puede tener un orden denso en el índice no agrupado, ya que el orden disperso no es posible porque los datos no están organizados físicamente en consecuencia.

Requiere más tiempo en comparación con el índice agrupado porque se realiza una cierta cantidad de trabajo adicional para extraer los datos siguiendo más el puntero. En el caso de un índice agrupado, los datos están directamente delante del índice.

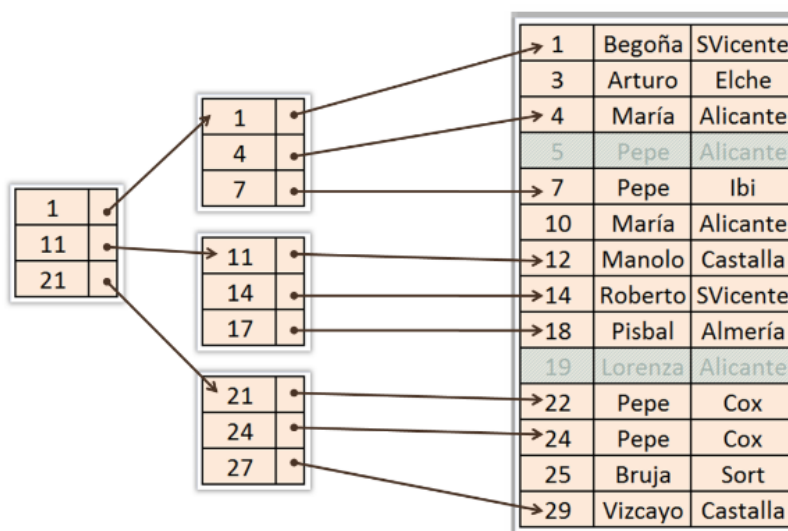
Figura 4. Índice agrupado.



Indexación multinivel: con el crecimiento del tamaño de la base de datos, los índices también crecen. Como el índice se almacena en la memoria principal, un índice de un solo nivel puede llegar a ser de un tamaño demasiado grande para almacenarlo con múltiples accesos al disco. La indexación multinivel segrega el bloque principal en varios bloques más

pequeños para que el mismo pueda almacenarse en un solo bloque. Los bloques externos se dividen en bloques internos que, a su vez, apuntan a los bloques de datos. Esto se puede almacenar fácilmente en la memoria principal con menos gastos generales. Ver figura5.

Figura 5. Índice multinivel.



Para mayor comprensión diríjase al:

Anexo 2 - Índices basados en varios campos utilizando Microsoft Access. [Clic aquí.](#)

Para mayor comprensión diríjase al:

Anexo 3 - Crear un índice y utilizarlo para mejorar el rendimiento en Microsoft Access. [Clic aquí.](#)

5.2 Vistas en base de datos

Se invita a conocer un poco más sobre las bases de datos, en esta ocasión descubra todo sobre las vistas en las bases de datos.

Vistas en base de datos

Una vista puede unir información de varias tablas o puede decir que las vistas son útiles para ocultar información no deseada, por ejemplo, para agregar el campo de esmalte a

la información del pedido. La vista de base de datos es un subconjunto de la base de datos ordenada y mostrada de una manera particular. Una vista de la base de datos muestra uno o más registros de la base de datos en la misma página. Una vista puede mostrar algunos o todos los campos de la base de datos.

Las vistas tienen filtros para determinar qué registros muestran. Las vistas pueden ordenarse para controlar el orden de los registros y agruparse para mostrar registros en conjuntos relacionados. Las vistas tienen otras opciones, como totales y subtotales. Una consulta devuelve información de una tabla o un conjunto de tablas que coincide con criterios particulares. La mayoría de los usuarios interactúan con la base de datos utilizando las vistas de la base de datos. Una clave para crear una base de datos útil es un conjunto de vistas bien elegido. Afortunadamente, aunque las vistas son poderosas, también son fáciles de crear. Cree vistas personalizadas de una base de datos para organizar, filtrar y ordenar registros.

Puede utilizar vistas para:

- a. Centrarse en los datos que les interesan y en las tareas de las que son responsables. Los datos que no son de interés para un usuario pueden quedar fuera de la vista.
- b. Definir las combinaciones, proyecciones y selecciones de uso frecuente como vistas para que los usuarios no tengan que especificar todas las condiciones y calificaciones cada vez que se realiza una operación en esos datos.
- c. Mostrar diferentes datos para diferentes usuarios, incluso cuando estén usando los mismos datos al mismo tiempo. Esta ventaja es particularmente importante cuando los usuarios de diferentes intereses y niveles de habilidad comparten la misma base de datos.

Ventajas

- a. Proporcione un nivel adicional de seguridad de la tabla al restringir el acceso a un conjunto predeterminado de filas o columnas de una tabla.

b. ocultar la complejidad de los datos: por ejemplo, una sola vista se puede definir con una combinación, que es una colección de columnas o filas relacionadas en varias tablas; sin embargo, la vista oculta el hecho de que esta información en realidad se origina en varias tablas.

c. Simplifique las declaraciones para el usuario: las vistas permiten a los usuarios seleccionar información de varias tablas sin saber realmente cómo realizar la unión.

d. Presentar datos en una perspectiva diferente: las columnas de vistas se pueden renombrar sin afectar las tablas en las que se basan las vistas.

e. Aislar las aplicaciones de los cambios en las definiciones de las tablas base. Si una vista hace referencia a tres columnas de una tabla de cuatro columnas, si se agrega una quinta columna o se cambia la cuarta columna, la vista y las aplicaciones asociadas no se ven afectadas.

f. Consulta expresa que no se puede expresar sin usar una vista, por ejemplo, se puede definir una vista que une un grupo por vista con una tabla o se puede definir una vista que une una vista UNION con una tabla.

g. Ahorro de consultas complejas.

Desventajas

a. Las filas disponibles a través de una vista no se ordenan, ni tampoco se ordenan.

b. No se pueden utilizar operaciones DML en una vista.

c. Cuando la tabla se descarta, la vista se vuelve inactiva, depende de los objetos de la tabla.

d. Afecta al rendimiento, la consulta desde la vista lleva más tiempo que la consulta directa desde la tabla.

Tipos de vistas

a. Vistas definidas por el sistema: son vistas predefinidas que ya existen en la base de datos maestra de SQL Server. También se utilizan como vistas de plantilla para todas las

bases de datos recién creadas. Estas vistas del sistema se adjuntarán automáticamente a cualquier base de datos definida por el usuario. Se tienen los siguientes tipos de vistas definidas por el sistema.

b. Vista de esquema de información: en SQL Server se tienen veinte vistas de esquemas diferentes. Se utilizan para mostrar información de una base de datos, como tablas y columnas.

c. Vista de catálogo: se introdujeron con SQL Server 2005. Se utilizan para mostrar información de autodescripción de la base de datos.

d. Vista de gestión dinámica: se introdujeron en SQL Server 2005. Estas vistas brindan al administrador información de la base de datos sobre el estado actual de la máquina SQL Server. Estos valores ayudan al administrador a analizar problemas y ajustar el servidor para un rendimiento óptimo. Estos son de dos tipos:

e. Vista de administración dinámica con ámbito de servidor: estos se almacenan solo en la base de datos maestra.

f. Vista de administración dinámica con ámbito de base de datos: estos se almacenan en cada base de datos.

Vistas definidas por el usuario

Estos tipos de vista los definen los usuarios. Se tiene dos tipos de vistas definidas por el usuario.

a. Vista simple: cuando se crea una vista en una sola tabla, se denomina vista simple. En la vista simple se puede insertar, actualizar, eliminar datos. Solo se puede insertar datos en una vista simple si se tiene una clave principal y todos los campos no nulos en la vista.

b. Vista compleja: cuando se crea una vista en más de una tabla, se denomina vista compleja. Solo se puede actualizar datos en una vista compleja. No se puede insertar datos en una vista compleja.

5.3 Reportes o informes en bases de datos

En el siguiente recurso podrá revisar los reportes o informes en las bases de datos, cómo realizarlos y aplicarlos.

Creación

Se crean a partir de datos visualizados para el análisis, el descubrimiento de datos y la toma de decisiones, y contienen datos útiles para la toma de decisiones y el análisis. La mayoría de las aplicaciones comerciales vienen con una herramienta de informes incorporada, que es una interfaz de “front-end” que llama a consultas de bases de datos de “back-end”.

Lenguaje de consulta

Para extraer datos, una consulta debe ejecutarse con varias herramientas que llamen al menos a un lenguaje de consulta, incluyendo:

- a. SQL (lenguaje de consulta estructurado).
- b. HTSQL (lenguaje de consulta estructurado de hipertexto).
- c. Lenguaje de consulta “Polycarp”.
- d. Protocolo SPARQL.

Beneficios de los informes de bases de datos

Tener una herramienta de informes de bases de datos en su lugar facilita la creación de informes y gráficos a partir de bases de datos.

Informes y cuadros de mando detallados

Crear informes y paneles para medir el rendimiento y obtener información sobre la organización. También utilizar informes para hacer seguimiento a los KPIs. Ordenar, buscar y desglosar informes para profundizar y comprender el rendimiento empresarial. En una organización, los gerentes tienden a tener requisitos particulares de presentación de informes. Los informes se pueden personalizar de acuerdo con el usuario objetivo y como resultado, se puede obtener información rápidamente y tomar mejores decisiones.

Optimice recursos y tiempo

Acelere el lanzamiento al mercado y reduzca el esfuerzo en desarrollo, permitiendo que su empresa se enfoque en las capacidades centrales del producto. Las herramientas de informes de bases de datos son intuitivas y no requieren conocimientos de SQL, lo que facilita la creación rápida de informes sin aprender “software” complejo. Generar informes de cualquier tipo es sencillo con una interfaz de arrastrar y soltar.

Visualizaciones de datos avanzadas

Cree visualizaciones de datos avanzadas desde gráficos de burbujas hasta gráficos de viñetas, detecte patrones y descubra hallazgos útiles en grandes cantidades de datos, incluyendo macrodatos. Genere informes resumidos o detallados para obtener una visión del estado de su negocio. Estos informes basados en la web se actualizan en tiempo real, brindando los datos más recientes y eficientes en términos de rendimiento.

Proteja sus datos confidenciales

Las hojas de cálculo son propensas a errores humanos y datos en la celda incorrecta, o si alguien comete un error al ingresar valores numéricos puede llevar su negocio al caos.

Examinar numerosas opciones

Los valores que se analizarán se pueden eliminar, mientras que se pueden crear gráficos poderosos en segundos. Es posible personalizar sus informes con información sobre herramientas adicional y contrastarlos con el período anterior o generar alertas para que sus informes sean más productivos y precisos.

5.4 Permisos, roles y control de acceso de usuarios

Se refiere a las diferentes autorizaciones que se deben establecer en una base de datos, en otras palabras, es la función de especificar derechos / privilegios de acceso a los recursos relacionados con la seguridad de la información. Más formalmente, "autorizar" es definir una política de acceso; con la gestión de identidad controlada, el control de acceso de usuario se concede en función de los permisos asignados a un rol.

Cada aplicación protegida por el habilitador de la base de datos puede definir un conjunto de permisos, es decir, un conjunto de cosas que se pueden hacer dentro de la aplicación que gestiona la base de datos, por ejemplo, dentro de la aplicación, la capacidad de enviar un comando para desbloquear una puerta inteligente. De manera similar, la capacidad de enviar un comando para hacer sonar la alarma y la capacidad de alterar los precios.

Estos permisos se agrupan en una serie de roles, por ejemplo, el desbloqueo de la puerta y la de activar una alarma podrían asignarse al rol de seguridad, lo que significa que los usuarios a los que posteriormente se les otorgue ese rol obtendrían ambos permisos. Los permisos pueden superponerse y asignarse a múltiples roles, también se asignan al rol de administración.

A su vez, los usuarios u organizaciones serán asignados a uno o más roles: cada usuario obtendrá la suma de todos los permisos para cada rol que tenga. El concepto de un rol es desconocido para un usuario: solo conocen la lista de permisos que se les ha otorgado, no cómo se dividen los permisos dentro de una determinada aplicación. En resumen, los permisos son todas las acciones posibles que se pueden realizar a los recursos dentro de una aplicación, mientras que los roles son grupos de acciones que puede realizar un tipo de usuario de esa aplicación.

Conceptos estándar de gestión de identidades

a. Usuario

Cualquier usuario registrado capaz de identificarse con un correo electrónico y una contraseña. A los usuarios se les puede asignar derechos individualmente o como grupo.

b. Aplicación

Cualquier aplicación asegurable que consta de una serie de microservicios.

c. Organización

Grupo de usuarios a los que se les puede asignar una serie de derechos. La alteración de los derechos de la organización afecta el acceso de todos los usuarios de esa organización.

d. Organización de roles

Los usuarios pueden ser miembros o administradores de una organización, los administradores pueden agregar y eliminar usuarios de su organización, los miembros simplemente obtienen los roles y permisos de una organización. Esto permite que cada organización sea responsable de sus miembros y elimina la necesidad de que un súper administrador administre todos los derechos.

e. Rol

Es un depósito descriptivo para un conjunto de permisos. Un rol se puede asignar a un solo usuario o a una organización. Un usuario que ha iniciado sesión obtiene todos los permisos de todos sus propios roles, además de todos los roles asociados a su organización.

f. Permiso

La capacidad de hacer algo en un recurso dentro del sistema.

Protección de los datos dentro de la base de datos

La autorización es un permiso otorgado al usuario, programa o proceso para acceder a un objeto o conjunto de objetos. El tipo de acceso a datos otorgado a un usuario puede ser de solo lectura o lectura y escritura. Los privilegios especifican el tipo de operaciones del lenguaje de manipulación de datos (DML) como seleccionar, insertar, actualizar, eliminar, etc., que el usuario puede realizar con los datos. Los dos métodos mediante los cuales se realiza el control de acceso son mediante el uso de privilegios y roles.

Privilegios: son permisos para acceder a un objeto con nombre de una manera prescrita, por ejemplo, permiso para consultar una tabla. Se pueden otorgar privilegios para permitir que un usuario en particular se conecte a la base de datos (crear una sesión);

seleccionar filas de la tabla de otra persona; o ejecutar el procedimiento almacenado de otra persona.

Privilegios de la base de datos: un privilegio es un derecho a ejecutar un tipo particular de declaración SQL o acceder al objeto de otro usuario. Algunos ejemplos de privilegios incluyen:

- a. El derecho a conectarse a la base de datos (crear una sesión).
- b. El derecho para crear una mesa.
- c. El derecho a seleccionar filas de la tabla de otro usuario.
- d. El derecho a ejecutar el procedimiento almacenado de otro usuario.

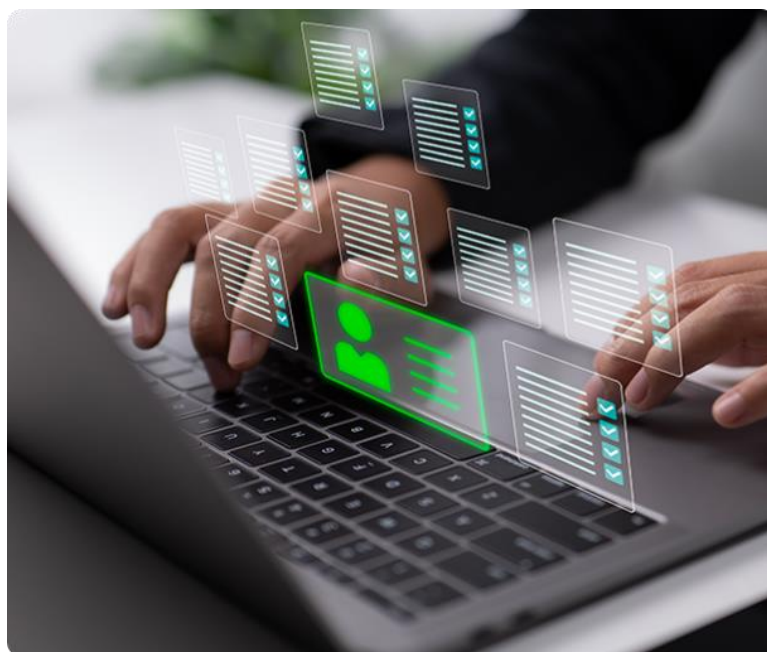
Se otorgan privilegios a los usuarios para que puedan realizar las tareas necesarias para su trabajo. Debe otorgar un privilegio solo al usuario que lo requiera, absolutamente para realizar el trabajo necesario. La concesión excesiva de privilegios innecesarios puede comprometer la seguridad.

Privilegios del sistema: un privilegio del sistema es el derecho a realizar una acción en particular, en un tipo particular de objeto, por ejemplo, los privilegios para crear tablas y eliminar el OWS de cualquier tabla en una base de datos son privilegios del sistema. En muchos sistemas comerciales de administración de bases de datos hay cientos de privilegios de sistema distintos.

Privilegios de objeto: un privilegio de objeto es un privilegio o derecho para realizar una acción particular en una tabla, vista, secuencia, procedimiento, función o paquete específico, por ejemplo, el privilegio de eliminar filas de la tabla DEPT es un privilegio de objeto. Los privilegios de objeto otorgados para una tabla, vista, secuencia, procedimiento, función o paquete se aplican tanto si se hace referencia al objeto base por su nombre como si se utiliza un sinónimo.

Roles: son mecanismos que se pueden utilizar para proporcionar autorización. Se puede otorgar un rol o un grupo de roles a una sola persona o un grupo de personas. Se

puede otorgar un rol a su vez a otros roles. Al definir diferentes tipos de roles, los administradores pueden administrar los privilegios de acceso con mucha más facilidad.



Los sistemas de administración de bases de datos brindan una administración de privilegios fácil y controlada a través de roles. Los roles son grupos con nombre de privilegios relacionados que otorga a los usuarios u otros roles. Los roles están diseñados para facilitar la administración del sistema del usuario final y los privilegios de objetos. Las siguientes propiedades de los roles permiten una administración de privilegios más sencilla dentro de una base de datos:

- a. **Administración de privilegios reducidos:** en lugar de otorgar explícitamente el mismo conjunto de privilegios para varios usos, puede otorgar los privilegios a un grupo de usuarios relacionados con un rol. Entonces, solo se debe otorgar el rol a cada miembro del grupo.
- b. **Administración dinámica de privilegios:** si los privilegios de un grupo deben cambiar, solo se deben modificar los privilegios del rol.
- c. **Disponibilidad selectiva de privilegios:** puede habilitar o deshabilitar selectivamente los roles otorgados a un usuario. Esto permite un control específico de los privilegios de un usuario en cualquier situación dada.

- d. **Seguridad específica de la aplicación:** puede proteger el uso de roles con una contraseña. Las aplicaciones se pueden crear específicamente para habilitar un rol cuando se proporciona la contraseña correcta. Los usuarios no pueden habilitar el rol si no conocen la contraseña.

Para mayor comprensión diríjase al:

Anexo 4 - Concesión y revocación de privilegios y funciones usando comandos. [Clic aquí.](#)

Para mayor comprensión diríjase al:

Anexo 5 - Gestionar permisos de usuario en Microsoft Access. [Clic aquí.](#)

5.5 Optimización de bases de datos

Por medio de la siguiente infografía sabrá cómo optimizar las bases de datos.

La optimización de la base de datos se refiere a una variedad de estrategias para reducir el tiempo de respuesta del sistema de la base de datos. Siga leyendo para obtener más información sobre la optimización de bases de datos y los profesionales que administran bases de datos en una amplia gama de industrias.

Tener en cuenta cómo se realiza la optimización de la base de datos para mejorar la velocidad, un ejemplo de esto es Magneto.

Las bases de datos pueden almacenar enormes cantidades de información, “petabytes” de información. Se accede a bits específicos de datos mediante consultas escritas en un lenguaje de interfaz particular, como SQL. La optimización de la base de datos implica maximizar la velocidad y la eficiencia con la que se recuperan los datos.

Los diseñadores, administradores y analistas de bases de datos trabajan juntos para optimizar el rendimiento del sistema a través de diversos métodos. El diseño cuidadoso que aborda cuidadosamente las necesidades funcionales es la base de la mejora del rendimiento.

La construcción de consultas también puede producir resultados más rápidos. Los administradores y analistas buscan mejorar los métodos de acceso a los datos de sus

servidores y los tiempos de recuperación a través de técnicas de diseño, análisis estadístico y monitoreo del tráfico del sistema.

En esta función se esfuerza por aprovechar al máximo el potencial de su servidor al poseer un sólido conocimiento de la estructura de los datos, las aplicaciones instaladas en el servidor y el impacto que tienen las diversas tareas en el rendimiento general de la base de datos.

Para mayor comprensión diríjase al:

Anexo 6 - Usar el analizador de rendimiento para optimizar una base de datos de “Microsoft Access”. [Clic aquí.](#)

Para mayor comprensión diríjase al:

Anexo 7 - Aumentar la velocidad de ejecución de “Microsoft Access”. [Clic aquí.](#)

5.6 Limpieza de la base de datos

En este apartado descubrirá todo sobre la limpieza de una base de datos.

1. Limpieza de la base de datos

La cantidad de datos disponibles ha seguido aumentando y por, lo tanto, existen oportunidades de error. Como resultado, se confía en la limpieza de datos para optimizar nuestros procesos de gestión de base datos. La limpieza de los datos de una base de datos aumenta la integridad y relevancia de los datos al reducir las inconsistencias, eliminar errores y permitir que las empresas tomen decisiones precisas e informadas.

2. ¿Qué es la limpieza de datos?

Es el proceso de identificación y resolución de datos corruptos, inexactos o irrelevantes. Esta etapa crítica del procesamiento de datos, también conocida como depuración o limpieza de datos, aumenta la coherencia, la confiabilidad, la gestión de las bases de datos y el valor de los datos de una empresa u organización.

3. Las inexactitudes comunes en los datos

Incluye valores perdidos, entradas extraviadas y errores tipográficos. En algunos casos, la limpieza de datos, requiere que se completen o corrijan ciertos valores, mientras que en otros casos, los valores deberán eliminarse por completo.

4. Datos sucios o con errores

Los datos que contienen este tipo de errores e inconsistencias se denominan “datos sucios” y sus consecuencias son reales. Se estima que solo el 3% de los datos cumplen con los estándares básicos de calidad y que los datos sucios les cuestan a las empresas u organizaciones en los EE. UU. Más de \$3 billones de dólares cada año.

5. El poder de los datos limpios

Una decisión es tan buena como los datos que la informan, y con cantidades masivas de datos que fluyen desde múltiples fuentes, una herramienta de limpieza de datos es más importante que nunca para garantizar la precisión de la información, la eficiencia de los procesos e impulsar la ventaja competitiva de una empresa u organización.

6. Toma de decisiones mejorada

La calidad de los datos es fundamental porque afecta directamente la capacidad de su empresa para tomar decisiones acertadas y calcular estrategias efectivas. Ninguna empresa puede permitirse perder tiempo y energía corrigiendo errores provocados por datos sucios.

7. Considere una empresa que se basa en datos generados por el cliente

Para desarrollar cada nueva generación de sus sistemas de pedidos en línea y móviles como “AnyWare” de “Domino's Pizza”. Sin un programa de limpieza de datos, es posible que los cambios y revisiones de la aplicación no se basen en información precisa o exacta. Como resultado, la nueva versión de la aplicación puede perder su objetivo y no cumplir con las necesidades o expectativas del cliente.

8. Eficiencia mejorada

Utilizar datos limpios no solo es beneficioso para las necesidades externas de su empresa, sino que también puede mejorar la eficiencia y la productividad internas. Cuando la información se limpia adecuadamente, revela información valiosa sobre las necesidades y los procesos internos, por ejemplo, una empresa puede usar datos para rastrear la productividad de los empleados o la satisfacción laboral en un esfuerzo por predecir y reducir la rotación. La limpieza de los datos, de las revisiones de desempeño, los comentarios de los empleados y otros documentos de recursos humanos relacionados puede ayudar a identificar rápidamente a los empleados que tienen un mayor riesgo de desgaste.

9. Ventaja competitiva

Cuanto mejor satisfaga una empresa las necesidades de sus clientes, más rápido se elevará por encima de sus competidores. Una herramienta de limpieza de datos ayuda a proporcionar información completa y confiable para que pueda identificar las necesidades cambiantes de los clientes y mantenerse al tanto de las tendencias emergentes. La limpieza de datos puede producir tasas de respuesta más rápidas, generar clientes potenciales de calidad y mejorar la experiencia del cliente.

10. Limpieza de datos paso a paso

Una herramienta de limpieza de datos puede automatizar la mayoría de los aspectos del programa de limpieza de datos general de una empresa; pero una herramienta es solo una parte de una solución continua y a largo plazo para la limpieza de datos.

5.7 Pruebas en la base de datos

A continuación puede consultar las pruebas correspondientes.

En el desarrollo de “software”, el procesamiento y almacenamiento de datos en diferentes estados refleja las reglas comerciales sobre las que se basa una aplicación. El corazón y el alma de cualquier aplicación de “software” son los datos que se conservan en las bases de datos para su recuperación y procesamiento posterior. El sistema de base de

datos (SQL o no SQL) elegido para una aplicación debe satisfacer las necesidades de almacenamiento y procesamiento de datos requeridos por la aplicación.

¿Qué son las pruebas de bases de datos?

Se ha analizado el código de prueba y también cómo probar correctamente las API, pero ¿qué pasa con la capa de datos de esta aplicación?, ¿de qué se trata la prueba de base de datos?

Las pruebas de bases de datos se componen principalmente de la construcción de consultas SQL para afirmar y validar diferentes operaciones, estructuras y atributos de la base de datos requeridos por la aplicación para la que es la base de datos.

Este proceso puede incluir validar el esquema, probar operaciones y transacciones CRUD para asegurarse de que la base de datos esté configurada correctamente. Estas pruebas pueden ser completamente automatizadas, completamente manuales o un enfoque híbrido, utilizando una combinación de procesos manuales y automatizados, por ejemplo, en una prueba completamente manual, podría ingresar al sistema de administración de la base de datos y ejecutar consultas para validar las suposiciones. O puede probar la interfaz de usuario de la aplicación para ver que devuelve los datos correctos después de una operación.

Los probadores de bases de datos trabajan con los desarrolladores de aplicaciones para probar adecuadamente los escenarios en los que debe operar la base de datos. Un evaluador de bases de datos debe comprender completamente las reglas comerciales de la aplicación, además de estar familiarizado con la estructura de la base de datos.

¿Por qué son importantes las pruebas de bases de datos?

Cuando la integridad de los datos en una aplicación se ve comprometida, puede provocar enormes pérdidas monetarias que podrían provocar el colapso de todo el negocio. La gravedad de estas consecuencias varía desde características menos sensibles a los datos, como la cantidad de “retweets” en un “tweet”, hasta el valor real de las acciones en una aplicación comercial.

Si bien las pruebas de la base de datos no pueden evitar todas las violaciones de datos o los compromisos de integridad, las peores consecuencias se pueden mitigar probando minuciosamente el sistema de base de datos utilizado para una aplicación. Algunas de las razones comunes para probar bases de datos son:

- a. Para prevenir ataques como ataques de inyección SQL.
- b. Asegurarse de que los problemas de la red o los cortes de energía no afecten la integridad de los datos, ni provoquen la pérdida o la corrupción de datos.
- c. Eliminar errores de la base de datos para mantener la calidad de los datos.
- d. Para verificar que todos los datos se almacenan en el formato correcto.
- e. Para validar relaciones entre entidades dentro de la base de datos.

¿Qué se debe probar?

Una vez que comprenda de qué se tratan las pruebas de bases de datos y por qué son importantes, puede aplicar lo que sabe para asegurarse de que todos los casos de prueba están bien cubiertos.

CRUD

El primer conjunto de casos de prueba es para las operaciones de creación, lectura, actualización y eliminación (CRUD) que realiza su aplicación. Esto es para garantizar que los datos se almacenen y recuperen de manera que se asegure la integridad de los datos. Un evaluador puede hacer esto directamente desde la interfaz de la aplicación o usar un sistema de administración de bases de datos (DBMS) para ejecutar consultas, usando comandos de lenguaje de manipulación de datos SQL (DML). Los comandos SQL maliciosos que pueden dañar los datos de la base de datos también se pueden ejecutar en un entorno de prueba para garantizar que la base de datos esté protegida contra ataques de inyección SQL.

Actas: las transacciones son operaciones sensibles de la base de datos porque requieren que se realicen diferentes consultas con una recuperación adecuada de las fallas. Una transacción de base de datos estándar es seguir los principios de ACID.

- a. A de “atomicity” - atomicidad: debe completar todas las operaciones o no realizar ninguna operación en absoluto.
- b. C de “consistency” - consistente: el estado de la base de datos siempre debe ser válido y se deben cumplir todas las restricciones.
- c. I de “isolation” - aislamiento: cada transacción debe realizarse de forma aislada de otras transacciones que se ejecutan al mismo tiempo. El estado final de la base de datos debe ser como si las transacciones se ejecutaran secuencialmente.
- d. D de “durability” - durabilidad: sin pérdida de datos por ningún motivo, una vez que se confirma la transacción.

El probador de la base de datos elabora consultas SQL para verificar y validar estas propiedades en las operaciones de transacción en la aplicación. Este tipo de prueba es especialmente importante para probar aplicaciones financieras.

Esquema: es el modelo de la estructura de la base de datos y los evaluadores deben estar muy familiarizados con él. Los evaluadores pueden usar el comando DESC de SQL para revelar el esquema de la base de datos y asegurarse de que se ajuste a lo que espera la aplicación. Los evaluadores también usan expresiones regulares para validar los nombres de los campos de la tabla y verificar que el valor se ajuste al tipo de datos esperado.

Disparadores: los desencadenadores son como controladores de eventos en el código. Cuando ocurre una operación en una tabla (por ejemplo, cuando se agrega una nueva fila) se puede configurar un disparador para ejecutar un fragmento de código o consulta en respuesta a la operación que acaba de ocurrir. Los disparadores crean un efecto de cascada en algunas operaciones, por ejemplo, eliminar a un usuario de una base de datos puede provocar que las publicaciones del usuario también se eliminen. Como evaluador desea asegurarse de que esta operación se realice con precisión y no elimine las publicaciones de otro usuario.

Un evaluador puede ejecutar una consulta SQL para iniciar la operación de origen que desencadena la cascada. El probador también podría realizar la operación desde la interfaz de la aplicación para examinar su efecto en los registros de la base de datos.

Otros: las operaciones que describí anteriormente no son el único conjunto de operaciones que se pueden probar, son solo algunas de las más importantes para la mayoría de las aplicaciones sensibles a los datos. Otras operaciones y atributos que se pueden probar son:

- a. Restricciones de la base de datos.
- b. Procedimientos almacenados.
- c. Vista.
- d. Índices.

¿Cómo probar las bases de datos?

Ahora que se sabe qué probar, ¿cómo se deben realizar las pruebas?

Sorprendentemente, ejecutar las pruebas de las bases de datos no es tan diferente a ejecutar pruebas en aplicaciones. Como se mencionó anteriormente, estas pruebas se pueden realizar manualmente realizando operaciones desde la interfaz de usuario de la aplicación para asegurarse de que se devuelvan los datos correctos después de cada operación o verificando los registros de la base de datos para ver los resultados de la operación. También puede utilizar un DBMS para ejecutar sus consultas de prueba y validar los resultados. Si se prefiere no realizar pruebas manualmente, puede sentirse aliviado al saber que las pruebas automatizadas de bases de datos son muy similares a las pruebas automatizadas en el código de la aplicación. La principal diferencia es que esta vez sus pruebas ejecutan consultas en lugar de código de aplicación, también se aplican los siguientes pasos que se encuentran en el código de la aplicación de prueba:

- a. Prepare su entorno de prueba.
- b. Ejecute sus pruebas con un corredor de prueba (por ejemplo, “Selenium”).
- c. Verifique y valide sus resultados.
- d. Informe sus afirmaciones.



Esta es la razón por la que el mejor atributo de un probador de bases de datos es poder crear consultas SQL para cada caso de prueba requerido.

Herramientas de prueba de las bases de datos

Existen muchas herramientas de prueba de bases de datos. La herramienta que elija dependerá de uno o más de los siguientes:

- a. Estrategia de prueba (manual, automatizada o híbrida).
- b. Tipo de base de datos (SQL o No-SQL).
- c. Proveedor de base de datos (MySQL, MSSQL, Oracle u otro).

La mayoría de los sistemas de prueba de bases de datos se componen de más de una herramienta, por ejemplo, “Selenium” se puede utilizar con “TestNG” para pruebas de bases de datos en aplicaciones Java. De manera similar, “SeLite” usa “Selenium” para probar bases de datos SQLite, mientras que “SQL Server” viene con herramientas para bases de datos de pruebas unitarias.

Para mayor comprensión diríjase al:

Anexo 8 - Lista de herramientas de prueba de bases de datos. [Clic aquí.](#)

6 MySQL

MySQL es un verdadero servidor de base de datos SQL multiproceso y multiusuario. SQL es el lenguaje de bases de datos más popular del mundo. MySQL es una implementación cliente / servidor que consta de un dominio de servidor MySQLD y muchos programas y bibliotecas cliente diferentes.

SQL es un lenguaje estandarizado que facilita el almacenamiento, la actualización y el acceso a la información, por ejemplo, puede utilizar SQL para recuperar información de productos y almacenar información de clientes para un sitio web. MySQL también es lo suficientemente rápido y flexible como para permitirle almacenar registros e imágenes.

Los principales objetivos de MySQL son la velocidad, la solidez y la facilidad de uso. MySQL se desarrolló originalmente porque en TcX se necesitaba un servidor SQL que pudiera manejar bases de datos muy grandes en un orden de magnitud más rápido de lo que cualquier proveedor de bases de datos podría ofrecer. Ahora se ha estado usando MySQL desde 1996 en un entorno con más de 40 bases de datos que contienen 10.000 tablas, de las cuales más de 500 tienen más de 7 millones de filas. Se trata de unos 100 gigabytes de datos de misión crítica. La base sobre la que se construye MySQL es un conjunto de rutinas que se han utilizado en un entorno de producción muy exigente durante muchos años. Aunque MySQL todavía está en desarrollo, ya ofrece un conjunto de funciones poderosas y muy útil.

Para mayor comprensión diríjase al:

Anexo 9 - Más sobre MySQL. [Clic aquí.](#)

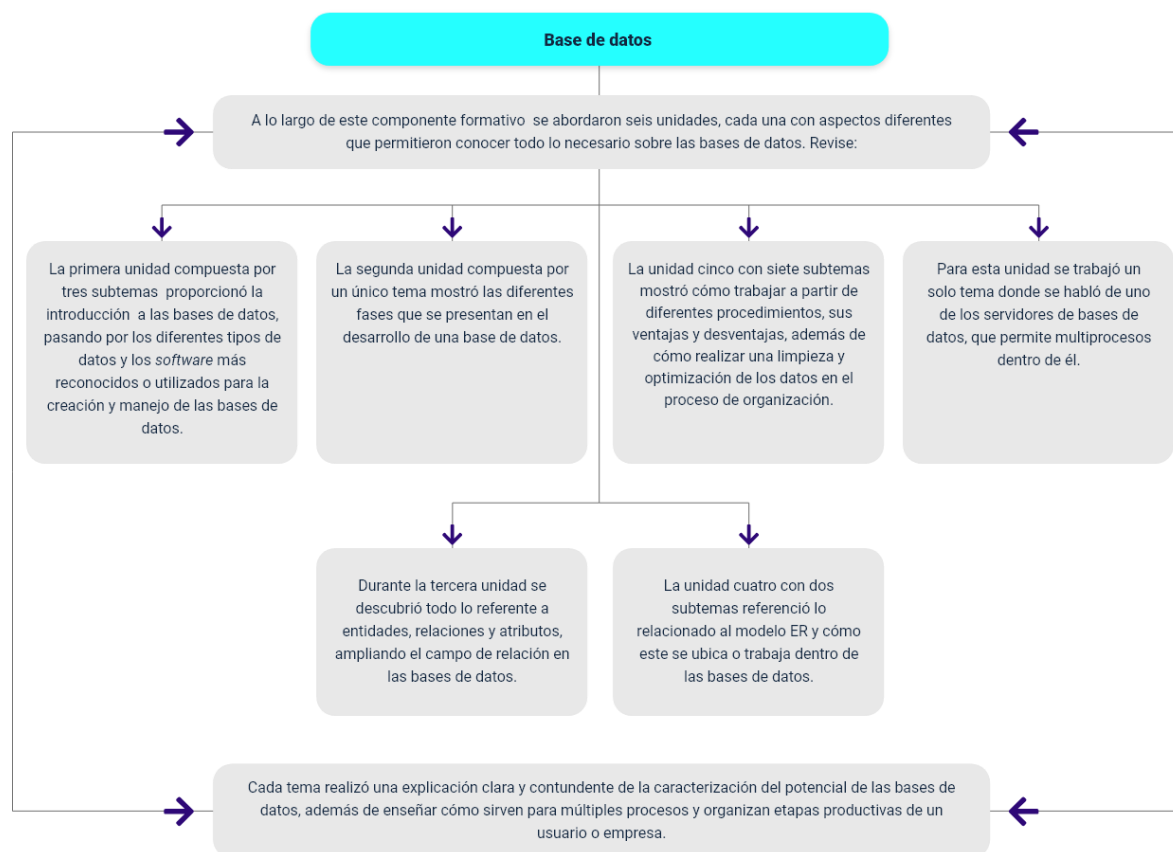
En el material complementario de este componente se recomiendan una serie de videos donde se explica paso a paso el diseño de una base de datos completa desde cero utilizando la aplicación Microsoft Access. Este material de apoyo amplía mucho más el conocimiento sobre base de datos y permite generar una gran experiencia de diseño.

Síntesis

Tecnología en gestión eficiente de la energía

Síntesis: Base de datos

El componente formativo bases de datos abarca diversos aspectos clave para su comprensión y aplicación. En primer lugar, se explora la introducción a las bases de datos, incluyendo la evolución y los diferentes tipos existentes, así como las herramientas de software más conocidas utilizadas en este ámbito. Luego, se analizan las fases del proceso de desarrollo de bases de datos, centrándose en la conceptualización de entidades, relaciones y atributos, y el modelo ER. Se examinan también los procedimientos fundamentales en bases de datos, como los índices, las vistas, los reportes o informes, los permisos y roles de usuario, la optimización y la limpieza de la base de datos, así como las pruebas necesarias. Finalmente, se destaca la importancia de MySQL, una popular herramienta de gestión de bases de datos ampliamente utilizada en la industria.



Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Evolución de las bases de datos	Edteam. (2020, 13, febrero). ¿Que son las bases de datos? [Archivo de video].	Video	https://www.youtube.com/watch?v=knVwokXITGI
Evolución de las bases de datos	Edteam. (2022, 9, septiembre). ¿SQL o NoSQL? ¿Cuál base de datos es mejor? [Archivo de video].	Video	https://www.youtube.com/watch?v=8K1PKSqHFRQ

Glosario

Acido: el acrónimo de las propiedades mantenidas por sistemas de gestión de bases de datos estándar, que significa atomicidad, consistencia, aislamiento y durabilidad.

Atomicidad: la propiedad de una transacción que garantiza que todos o ninguno de los cambios realizados por la transacción se escriben en la base de datos.

AVL-Árbol: es un árbol de búsqueda binaria autoequilibrado.

Caché: la memoria de la computadora que se reserva para contener una parte de los datos de la base de datos a los que el programa de aplicación de la base de datos accedido más recientemente.

Cascada: un atributo de clave externa que migra automáticamente los cambios realizados a una referencia (es decir, clave primaria), tabla a todas las referencias (clave externa), filas de la tabla.

Columna: una sola unidad de datos con nombre que tiene un tipo de datos en particular (por ejemplo, número, texto o fecha). Las columnas solo existen en tablas.

Compilador: una herramienta de desarrollo de “software” que traduce programas de lenguaje de alto nivel a instrucciones en lenguaje de máquina, que un procesador en particular puede comprender y ejecutar.

DBMS: un acrónimo de sistema de administración de base de datos.

Depurador: una herramienta utilizada para probar y depurar “software”. Un depurador remoto típico se ejecuta en una computadora host y se conecta al destino a través de un puerto serie o en una red.

DML: lenguaje de manipulación de bases de datos. En SQL, declaraciones como “update”, “insert” y “delete” se consideran DML.

Durabilidad: la propiedad de una transacción en la que el DBMS garantiza que todas las transacciones comprometidas sobrevivirán a cualquier tipo de falla del sistema.

E / S: de entrada y salida. Para un DBMS, normalmente se trata de una unidad de disco que se utiliza para crear una base de datos, durabilidad.

Gota: abreviatura de “Binary Large Object”. En SQL, BLOB puede ser un término general para cualquier dato de tipo varbinary largo, varchar largo o largo varchar.

IEC: Comisión Electrotécnica Internacional. Junto con ISO, IEC controla el estándar SQL (ISO / IEC 9075) y muchos otros también.

IIOT: abreviación de Internet industrial de las cosas.

Índice: una estructura separada que permite un acceso rápido a las filas de una tabla en función de los valores de datos de las columnas utilizadas en el índice.

Java: un lenguaje de programación multiplataforma, orientado a objetos, similar a C ++, que está disponible gratuitamente para todos y cada uno de los desarrolladores de “software”.

JDBC: API de conectividad de base de datos de Java. JDBC proporciona una API estándar de manipulación y acceso a la base de datos para programas Java.

JSON: una representación de datos ofrecida como una alternativa más compacta; pero aún legible por humanos para XML.

“Little-Endian”: es un tipo de direccionamiento que se refiere al orden de los datos almacenados en la memoria.

Marcas: marcas comerciales, nombres comerciales, marcas de servicio o logotipos identificados en el sitio web y/o material impreso de una empresa.

Metadatos: "Datos sobre datos". En un contexto DBMS, los datos almacenados en columnas de una tabla tienen ciertos atributos, como el tipo, largo, descripción u otras características que permitan al DBMS procesar los datos de manera significativa o permitir que los usuarios los comprendan mejor.

Mesa: una colección de columnas estrechamente relacionadas. Una tabla consta de filas, cada una de las cuales comparte las mismas columnas; pero varían en los valores de las columnas.

Pila: es una estructura conceptual que consta de un conjunto de elementos homogéneos y se basa en el principio último en entrar, primero en salir (LIFO).

Replicación: un proceso en el que las modificaciones seleccionadas en una base de datos maestra se replican (reproducen) en otra base de datos.

RTOS: abreviatura común de sistema operativo en tiempo real. "Raima Database" Manager se ejecuta en la mayoría de RTOS, como "VxWorks", "Integrity", "Linux" integrado y QNX.

Servidor ("software"): un asiento que reside en una sola máquina servidor y es capaz de aceptar conexiones desde uno o más asientos que residen en las máquinas cliente.

Sincronización: el método de implementación (con frecuencia semáforos) mediante el cual la ejecución simultánea de múltiples subprocesos o procesos informáticos pueden acceder de forma segura y actualizar los datos compartidos.

SQL: es el lenguaje comúnmente aceptado que se utiliza para definir, consultar y manipular una base de datos relacional. La etimología de "SQL" no está clara, posiblemente una progresión de "QueL" (Query Language) a "SeQueL" y a "SQL".

XML: lenguaje de marcado extensible. Los documentos XML se utilizan mucho en la "World Wide Web" de "Internet"; pero también se utilizan en muchos contextos informáticos en los que es necesario compartir datos.

Referencias bibliográficas

Aguilar, L. (s.f.). *Big data Análisis de grandes volúmenes de datos en organizaciones*. Alfaomega Grupo Editor.

Cowart, R. (1993). *El ABC de base*. Ventura Ediciones.

Cuadra, D. (s.f.). *Desarrollo de bases de datos. Casos prácticos desde el análisis a la implementación*. RA-MA.

Godoc, E. (s.f.). *Los fundamentos del lenguaje*. Eni.

Orbegozo, A. (2016). *Access 2016: Manual práctico paso a paso*. Alfaomega Grupo Editor.

Peña, R. (2013). *Access 2013: manual práctico para todos*. Alfaomega Grupo Editor.

Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizabal	Responsable del Equipo	Dirección General
Norma Constanza Morales Cruz	Responsable de línea de producción	Regional Tolima - Centro de Comercio y Servicios
Jaime Mauricio Peñaloza Trespacios	Experto técnico	Regional Distrito Capital - Centro Electricidad, Electrónica y Telecomunicaciones
Leidy Carolina Arias Aguirre	Diseñadora instruccional	Regional Distrito Capital - Centro de Diseño y Metrología
Carolina Coca Salazar	Revisora metodológica y pedagógica	Regional Distrito Capital - Centro de Diseño y Metrología
Rafael Neftalí Lizcano Reyes	Responsable Equipo desarrollo curricular	Regional Santander – Centro Industrial del Diseño y la Manufactura
Julia Isabel Roberto	Correctora de estilo	Regional Distrito Capital – Centro de Diseño y Metrología
Juan Gilberto Giraldo Cortés	Diseñador instruccional	Regional Tolima – Centro de Comercio y Servicios
María Inés Machado López	Metodólogo	Regional Tolima – Centro de Comercio y Servicios
José Yobani Penagos Mora	Diseñador Web	Regional Tolima - Centro de Comercio y Servicios
Davison Gaitán Escobar	Desarrollador Fullstack	Regional Tolima - Centro de Comercio y Servicios
Gilberto Junior Rodríguez Rodríguez	Storyboard e Ilustración	Regional Tolima - Centro de Comercio y Servicios
Nelson Iván Vera Briceño	Producción audiovisual	Regional Tolima - Centro de Comercio y Servicios

Oleg Litvin	Animador	Regional Tolima - Centro de Comercio y Servicios
Francisco Javier Vásquez Suarez	Actividad Didáctica	Regional Tolima - Centro de Comercio y Servicios
Jorge Bustos Gómez	Validación y vinculación en plataforma LMS	Regional Tolima - Centro de Comercio y Servicios
Gilberto Naranjo Farfán	Validación de contenidos accesibles	Regional Tolima - Centro de Comercio y Servicios