DAM

Bases de Datos

Tema 1: Características básicas

Las bases de datos guardan y explotan datos. Guardan información y luego podemos trabajar con ella. **Es una colección de datos, organizados o no, que son almacenados sistemáticamente**. Se actualizan y recuperan datos, eliminan, y también se añaden archivos y registros con operaciones CRUD (Create, Read, Update, Delete).

Los datos pueden ser estructurados (tablas), semi-estructurados o no estructurados (big data). Excel es una base de datos.

**SGBD**: Sistemas de gestión de bases de datos. Administran, gestionan y explotan datos. MySQL es una SGBD.

**Proceso de diseño de una base de datos**

* ***Análisis de requisitos***: Se analiza lo que el cliente quiere.
* ***Diseño conceptual***: Se utiliza UML para pasar los datos del análisis a un esquema conceptual.
* ***Diseño lógico***: Se empieza a modelar las tablas. Se escoge el sistema gestor.
* ***Diseño físico***: Se diseña la base de datos.

**Modelos de Bases de Datos**

* **Jerárquico**: en árbol. Típico de cosas como XML, donde las etiquetas pueden anidarse unas dentro de otras y forman una jerarquía.
* **En red**: La información se presenta con grafos interconectados entre sí.
* **Relacional**: Tablas.
* **Orientada a Objetos**: PostgresSQL.
* **Documental**: JSON, estilo MongoDB.

**Usos de SGBD**

Sirven como interfaz y conexión entre la app y el sistema físico de almacenamiento. Gestionan eficientemente los datos y aumentan su consistencia.

**Elementos**

* Datos y metadatos: Colección de datos y descripción complementaria.
* SGBD: Apps que permiten la gestión de datos.
* Usuarios: personas y apps que acceden a las Bases de Datos.

**Funciones SGBD**

* Gestión completa: Descripción de los datos, manipulación y acceso, y seguridad (restricciones, permisos). También debería haber concurrencia (habilidad de consultar dos tablas a la vez).

**Usuarios**

* Normales: Interactúan con la BD mediante apps, pero indirectamente. Al registrarse en una web están metiendo datos a la BD.
* Avanzados: Interactúan directamente con la SGBD (programadores que no programan SGBD).
* Programadores: Crean las SGBD.
* Admins: Control técnico con permisos y demás, también copias de seguridad.

BD distribuidas (están en distintos servidores, pero se manejan con la misma SGBD) y no distribuidas (están en un mismo servidor).

**Principales Bases de Datos**

* Access -> Microsoft. Buena para bases pequeñas y pocos users.
* Derby -> Apache. Pequeña, escrita en Java e integrada en una IDE.
* MySQL -> Oracle. Soporta BD más grandes.
* CouchDB -> Apache. Documental. Usa JSON.
* DB2 -> IBM. Escrita en XML.
* FirebirdBD. Relacional. Consume poco. Buena concurrencia.
* Informix -> IBM. Informes. Relacional. Ocupa poco.
* MariaDB -> Va por comandos. Open Source.
* MongoDB -> Usa BSON, que es JSON optimizado. No relacional.
* SQLServer -> Microsoft.
* MySQL -> MySQL. Libre y comercial.
* OrientDB -> Oracle. Orientada a documentos, pero en red.
* PostgresSQL -> Relacional y orientada a objetos. Open Source.
* SQLite -> Relacional.
* Sysbase -> De SAP. Relacional, escalable y rápida.

Las normas para hacer software están en la norma 12207.

**Ciclo de vida de las bases de datos**

* **Planificación**: Definir acciones, recursos y técnicas para recolectar datos.
* **Definición y análisis**: Límites y requerimientos.
* **Diseño y prototipado**: Desarrollo teórico de esquemas.
* **Implementación y mantenimiento**: Creación de esquemas, testeo y validación.

**Arquitecturas de bases de datos**

* **Cliente/Servidor**: Centralizado. Acceso de un cliente a un servidor.
* **A dos ( y tres) niveles**: La primera tiene entornos de cliente (lo que el usuario ve) y servidor por separado. La de tres niveles separa usuario, negocio (que antes estaba en cliente) y lógica. En la arquitectura de dos niveles, el nivel del cliente y el nivel del servidor, el primero se encarga de

la presentación de los datos al usuario, es decir, la interfaz de usuario y la lógica de negocio, y el segundo de soportar los datos a través de herramientas destinadas para esta función, los sistemas gestores de bases de datos. En la arquitectura de tres niveles se ofrece mayor escalabilidad al aportar un nivel más que separa el nivel de interfaz de usuario con el nivel de lógica de negocio y procesamiento de datos. En el lado del cliente se elaboran los aspectos asociados a la interfaz de usuario y que ejecuta el usuario final. En el nivel intermedio se emplea un servidor de aplicaciones para procesar los datos y llevar a cabo los aspectos asociados con la lógica de negocio.

* **Distribuidas**: los datos están en servidores diferentes.

**1. Arquitectura Cliente-Servidor**

**Ejemplo:** **Microsoft SQL Server**

* **Descripción:** En esta arquitectura, el servidor de bases de datos (SQL Server en este caso) se encarga de gestionar los datos y las peticiones, mientras que los clientes (que pueden ser aplicaciones o interfaces de usuario) envían consultas al servidor y reciben los resultados. El cliente y el servidor pueden estar en máquinas diferentes, pero se comunican a través de una red.

En una instalación típica de WordPress, el servidor de bases de datos (como MySQL o MariaDB) se encarga de almacenar la información del sitio web, incluyendo publicaciones, usuarios y configuraciones. El cliente es el navegador web del usuario que solicita páginas del sitio. En este caso, el navegador envía solicitudes al servidor web, el cual consulta la base de datos y devuelve el contenido al usuario. Aquí, la comunicación entre el servidor de bases de datos y el servidor web ocurre de manera directa.

**2. Arquitectura de Dos Niveles**

**Ejemplo:** **MySQL con una aplicación web básica**

* **Descripción:** En esta arquitectura, se tiene una base de datos en un servidor (MySQL) y una aplicación cliente que interactúa directamente con esta base de datos. El cliente puede ser una aplicación web, una aplicación de escritorio o móvil. La aplicación cliente envía solicitudes directamente al servidor de bases de datos y recibe respuestas. Aquí no hay intermediarios adicionales; la comunicación es directa entre la aplicación cliente y el servidor de bases de datos.

**3. Arquitectura de Tres Niveles**

**Ejemplo:** **Aplicación web con PostgreSQL y un servidor de aplicaciones**

* + **Descripción:** En esta arquitectura, se introducen tres niveles:
  + **Capa de Presentación (Cliente):** La interfaz de usuario (por ejemplo, una aplicación web o móvil) que interactúa con el usuario.
  + **Capa de Aplicación (Servidor de Aplicaciones):** Un servidor que contiene la lógica de negocio y procesa las solicitudes del cliente. Por ejemplo, una aplicación desarrollada en Java EE, Node.js, o ASP.NET que maneja la lógica y las reglas del negocio.
  + **Capa de Datos (Servidor de Bases de Datos):** El servidor de bases de datos (como PostgreSQL) que almacena y gestiona los datos.

En esta arquitectura, el cliente se comunica con el servidor de aplicaciones, que a su vez se comunica con el servidor de bases de datos. Esto permite una mayor escalabilidad y flexibilidad, ya que la lógica de negocio y el almacenamiento de datos están separados y gestionados de manera independiente.

**Ejemplo:** **Redes Sociales como Facebook**

En una plataforma como Facebook, la arquitectura de tres niveles es muy evidente:

* 1. **Capa de Presentación (Cliente):** La interfaz de usuario en el navegador web o en la aplicación móvil de Facebook.
  2. **Capa de Aplicación (Servidor de Aplicaciones):** Los servidores que manejan la lógica de negocio de Facebook, procesan las solicitudes de los usuarios, gestionan las interacciones y la lógica de los feeds, notificaciones, etc.
  3. **Capa de Datos (Servidor de Bases de Datos):** Los servidores de bases de datos (como MySQL, HBase, u otros sistemas especializados) que almacenan todos los datos del usuario, publicaciones, mensajes, etc.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamenteLa recuperación de preproducción consiste en añadir extras.

**Seguridad en Bases de Datos**

La seguridad es cada vez más importante para asegurar la integridad de los datos en el mundo actual. Muchas SGBD tienen sistemas de control de accesos y autorización..

* **Física y lógica:** cortafuegos, control de accesos.
* **Integridad y disponibilidad** de los datos.
* **Cifrado, autenticación**
* **Mantenimiento y recuperación (backup).** Copias de seguridad, uso del balanceo de carga (alta disponibilidad), duplicidad de la información… por si acaso suceden desastres. Fundamental para el mantenimiento de los datos.
* **Bases de datos distribuidas**: Los datos están en distintos servidores, así que si uno cae los demás están a salvo.
* **Integración web**: Basada en la nube.

**Power BI**

Power BI es una plataforma unificada y escalable de inteligencia empresarial (Business Intelligence). Es un conjunto de técnicas para recopilar, analizar y presentar datos empresariales.

**Usos**

* **KPI (Key Performance Indicator):** Indicadores clave de rendimiento que miden el éxito y el progreso hacia los objetivos comerciales.
* **Análisis de ventas**: Examinar datos relacionados con las ventas de una empresa, como ingresos, conversiones y métricas de rendimiento de ventas.
* **Análisis de marketing**: Estudiar datos de campañas publicitarias, ROI, análisis de audiencia y conversiones para mejorar las estrategias de marketing.
* **Gestión de proyectos**: Supervisar y controlar el progreso de proyectos, asignar recursos y gestionar eficazmente las tareas y entregables.
* **Recursos humanos**: Analizar datos relacionados con la gestión del personal, como contrataciones, capacitación y retención de empleados.
* **Análisis financiero**: Examinar datos financieros, como estados financieros, presupuestos y métricas clave para tomar decisiones financieras sólidas.
* **Visualización de datos**: Representar datos complejos de manera gráfica y comprensible para facilitar la comunicación y la toma de decisiones.
* Interfaz de usuario gráfica, Aplicación

  Descripción generada automáticamente**Análisis en tiempo real**: Procesar y analizar datos a medida que se generan en tiempo real, lo que permite la toma de decisiones instantáneas.

**Tipos de fichero (también llamados archivos)**

Los ficheros almacenan datos en una ubicación específica. Los ficheros se utilizan para guardar y organizar datos de manera estructurada, lo que permite su posterior acceso, edición y recuperación.

* **Ficheros de texto**: Almacenan datos en formato de texto sin formato, como documentos de texto, código fuente de programación, archivos CSV (valores separados por comas). Se editan con programas simples, como el bloc de notas.
* **Ficheros binarios**: Se guardan en binario, así que son legibles por una máquina (e ilegibles para nosotros). Archivos ejecutables, imágenes, audio…
* **Ficheros de hojas de cálculo**: Contienen datos organizados en filas y columnas. Se usan para realizar cálculos y análisis de datos.
* **Ficheros de bases de datos**: Estos ficheros almacenan datos estructurados en un formato que permite la búsqueda, recuperación y organización eficiente de la información. Ejemplos incluyen bases de datos SQL (como MySQL o PostgreSQL), bases de datos NoSQL (como MongoDB o Redis) y ficheros de bases de datos locales (como SQLite).
* **Fichero de imágenes**: almacena imágenes o gráficos. Pueden ser estáticas o animadas. PNG, JPG
* **Ficheros de audio**. MP3, WAV.
* **Ficheros de vídeo**: MP4, AVI.
* **Ficheros de programas y ejecutables**: Juegos, software… .exe.
* **Ficheros comprimidos:** Almacenan múltiples ficheros en un solo archivo para ahorrar espacio y facilitar la transferencia. Ejemplos incluyen archivos ZIP, RAR y 7z.

**Diseño de bases de datos**

Para que los datos sean fieles deben conceptualizarse primero a través de ideas. Se trata del modelo conceptual.

**Pasos**

* Se describen los requisitos en lenguaje natural.
* Diagrama conceptual de la primera fase.
* Diseño lógico: se elige el sistema gestor y se modelan las tablas.
* Diseño físico: Se implementa la BD.

El modelo entidad-relación describe relaciones entre entidades. Se compone de:

Diagrama

Descripción generada automáticamente

* **Entidad**: Se representan dentro de cuadrados. Objeto real o abstracto donde se va a guardar información. Serían las tablas.
* **Atributo**: Serían las columnas de la tabla, son las propiedades de la entidad, y están definidas sobre un dominio. Se representan con círculos.
* **Dominio:** Conjunto de valores que toma una entidad. Cada atributo toma x valores, el conjunto de estos valores es el dominio. Por ejemplo, un atributo puede ser DNI. Todas las combinaciones que siguen ese patrón forman el dominio.
* **Relación**: Asociación entre entidades. Se representan con flechas. Hay varios tipos:
  + 1:1 -> Relación 1:1. Por ejemplo, una persona y su DNI.
  + 1:N -> Uno a muchos. Por ejemplo, un profesor para muchos alumnos.
  + N:M -> Muchos a muchos. Por ejemplo, muchos alumnos en muchas asignaturas y viceversa.
* **Cardinalidad**: Cuántos valores toma la entidad. Los tipos de relación encajan aquí. Es importante decir que **las flechas indican la dirección de N**.
* Diagrama

  Descripción generada automáticamente**Generalización**: Entidad padre con hijos. Los hijos heredan las burbujas del padre. Por ejemplo, una entidad puede ser vehículos. Pueden tener color, marca… y las entidades hijas (como moto o coche) heredan esos atributos.

**Transformación de interrelaciones**

* **Clave primaria (Primary Key)**: Un solo atributo por entidad sirve para distinguir registros. Por ejemplo, dos profesores pueden llamarse igual pero distinguirse por DNI, que sería la clave primaria.
* **Clave externa** **(Foreign Key):** Sirve para conectar unas tablas con otras. La PK de una tabla se pasa como atributo a otra tabla para que estén conectadas entre sí.

La recuperación preproducción consiste en añadir extras.

**Seguridad en Bases de Datos**

* **Física y lógica:** cortafuegos, control de accesos.
* **Integridad y disponibilidad** de los datos.
* **Cifrado, autenticación**
* **Mantenimiento y recuperación (backup).** Copias de seguridad, uso del balanceo de carga, duplicidad de la información…
* **Bases de datos distribuidas**: Los datos están en distintos servidores, así que si uno cae los demás están a salvo.

**Diagrama

Descripción generada automáticamente**En las relaciones 1:N, FK va a donde apunta N. En el diagrama, la PK de B pasa a A, porque N apunta a A, y se convierte también en FK.Diagrama

Descripción generada automáticamente

Imagen que contiene interior, texto, tabla, colgando

Descripción generada automáticamenteEn 1:1, la FK es la PK de la otra tabla y puede ir en cualquier lado.

En relaciones N:M salen FKs de las dos relaciones, que también son PKs.

Diagrama

Descripción generada automáticamente

**Seguridad en Bases de Datos**

**·       Física y lógica:** cortafuegos, control de accesos.

**·       Integridad y disponibilidad de los datos.**

**·       Cifrado, autenticación**

**·       Mantenimiento y recuperación (backup).** Copias de seguridad, uso del balanceo de carga, duplicidad de la información…

**·       Bases de datos distribuidas:** Los datos están en distintos servidores, así que si uno cae los demás están a salvo.

**Fases del diseño conceptual**

* **Modelo conceptual:** Representa los elementos que se quieren almacenar info y sus relaciones.
* **Transformación a tablas:** Cada entidad se convierte en una tabla o relación en el modelo relacional.
* **Transformación atributos y relaciones:** Atributos pasan a columnas y relaciones a FKs.
* **Normalización del esquema.** Normalizar es limpiar datos antes de pasarlas a la base.
* **Seguridad:** Agrupar tablas, usar índices, planificar el uso de la nube.

**Diseño físico**

En esta etapa, la última en el proceso de desarrollo de la base de datos, se pretende conseguir un esquema interno creado a través del sistema gestor de bases de datos elegido. Este esquema interno debe funcionar según los requerimientos que los usuarios plantearon en la etapa de análisis. Además de cumplir con estos requisitos, el diseño físico contemplará los mecanismos que permitan disminuir tanto el tiempo de respuesta como el espacio de almacenamiento usado. Los procesos asociados con la seguridad también son aspectos asociados al diseño físico.

**También se buscará la definición de las estructuras de archivos, la indexación de tablas, el uso del espacio de memoria principal, los roles de seguridad y los objetos de gestión.** Estos pasos serán testeados y puestos a evaluación hasta su refinación. **Este proceso de mejora se conoce como ajuste de la base de datos o proceso tunning**. El tunning o afinamiento es el proceso sistemático de búsqueda de origen, naturaleza y corrección de los problemas de rendimiento de un sistema de bases de datos. El diseño físico de las consultas es una tarea importante en este proceso.

Una técnica que optimiza los accesos es la **agrupación de tablas**, o también llamada **clustering.**

En la etapa física se pueden tomar decisiones sobre la redundancia de datos con el fin de aumentar la eficiencia temporal en las acciones de accesos a datos. La redundancia de datos evita buscar información extra en tablas gigantes, lo que aumentaría el tiempo en las operaciones asociadas a estas tablas. Para evitar esos accesos, los datos que se requieren pueden estar incluidos en las tablas iniciales. Este proceso se puede llamar desnormalización de tablas.

Los usuarios de la base de datos accederán de la totalidad de esta solo a aquella parte que se considere necesaria, ni más ni menos, con el fin de aumentar su seguridad. Esta visión parcial de todo el esquema se consigue a través de un elemento denominado vista. Las vistas son mecanismos que permiten construir la visión que tiene un usuario de toda la base de datos, es decir, su esquema lógico. Junto a esto, existirán unas reglas que definan el modo en que un usuario puede explotar la arquitectura, los roles, y el esquema lógico, los perfiles.

La monitorización de bases de datos consiste en realizar un seguimiento proactivo del estado y del rendimiento de la misma.

El diseño físico no se debe concebir como un proceso secuencial que finaliza cuando se ha encontrado una configuración válida. Las bases de datos pueden sufrir variaciones a lo largo del tiempo, tanto en su tamaño como en su esquema lógico global o en el uso que se desee hacer de ellas, lo que obligará a una monitorización continua de su rendimiento (para comprobar si se va degradando) y a la introducción de ajustes que permitan adaptarse a los cambios sufridos por la base de datos o solventar las pérdidas de eficiencia que se hayan producido. Cada proceso de ajuste supone en cierta medida una reconstrucción del diseño físico. que se haya realizado, por lo cual esta etapa de diseño volvería a comenzar de nuevo.