**Preguntas Bases Tema 1**

### **Ficheros: definición, tipos principales y usos.**

Un fichero (también llamado archivo) guarda unidades de información. Pueden guardarse tanto en local como en la nube, y los hay de varios tipos: de texto plano (código fuente, csv), binarios (guardan información en binario, tal como imágenes, audio…), de documentos (mucho más elaborados que los de texto, como Word), de hojas de cálculo (tipo excel, sirve para hacer cálculos complejos), multimedia (audio, vídeo, imagen…), de bases de datos (guardan información de forma estructurada, facilitando su posterior explotación), ejecutables (para software instalable en un pc) y comprimidos (de forma que ocupan mucho menos espacio.

## **Tipos de diseño (conceptual, físico y lógico):**

El diseño conceptual consiste en traspasar las ideas/requisitos de la base de datos a un mapa conceptual, más abstracto. Se usan distintos métodos para ello, como los diagramas entidad-relación. El diseño lógico coge lo desarrollado en la fase previa y va a escoger qué tipo de modelo de datos queremos, qué sistema gestor, etc. Por último, el diseño físico crea la base de datos, estableciendo las cargas de trabajo.

### **Qué es una base de datos**

Es un conjunto de datos organizados y estructurados (o no, ya que también hay bases semi y desestructuradas) sistemáticamente para luego facilitar su consulta, transformación (edición) y eliminación posterior. Lo importante no es la estructura, es su sistematización para luego poder manipular la información. El Big Data, por ejemplo, es ingente, y no está estructurado porque sería imposible hacerlo.

### **Qué es un SGBD y las facilidades que aportan**

Es una interfaz que conecta los datos con el usuario y le permite manipularlos, consultarlos, crearlos y eliminarlos de forma mucho más visual, además de gestionar controles de acceso. Aportan organización, estructura, son capaces de gestionar gran cantidad de información. Cuando ésta se estructura en la interfaz gracias a unas reglas semánticas que codifican los mensajes para que sean legibles con una máquina. La información clasificada, sistematizada y estructurada de esta forma son los **datos**.

### **Tipos de restricciones**

El modelo de datos sistematiza los datos, describe cómo se relacionan entre sí, así como sus restricciones. Existen tres tipos de restricciones:

* Inherentes: Restricciones impuestas por el modelo.
* Opcionales: Restricciones impuestas por el diseñador de la base de datos de acuerdo a los requisitos. El SGBD ayuda a gestionarlas.
* Libres: Dependen del gusto de cada usuario.

### **Modelos de datos. Explicación. (Todos o varios de ellos). La imagen también ayuda.**

### 

* **Modelo jerárquico**: En árbol. Jerarquiza la información, con datos que están o bien por encima o son dependientes de otros.
* **En red**: Son nodos interconectados entre sí. Forman un grafo.
* **Documental**: Se estructuran en documentos, de forma que los datos están semi estructurados. JSON es un formato muy utilizado, y MongoDB es un ejemplo.
* **Relacional**: Tablas de datos. De los modelos más utilizados.
* **Multidimensional**: Para cantidades de datos ingentes.
* **Orientado a objetos**: Sigue la filosofía POO.

### 7- **Elementos en un entorno de BD. (datos,metadatos,sgbd, usuarios) Explicar.**

Los datos son el corazón de una BD, la información refinada, por así decirlo, ya que están clasificados y sistematizados, aunque estén desestructurados También forman parte de la colección de datos los del usuario. Pueden estar cifrados para más seguridad.

Los metadatos describen a los datos. Por ejemplo, un archivo jpg puede tener metadatos del modelo de cámara, especificaciones técnicas del tipo de objetivo que se ha usado, etc. Se almacenan en los llamados diccionarios de datos.

Los SGBD son las interfaces que conectan los datos con el usuario, a menudo a través de una interfaz gráfica. Permiten gestionar, manipular, crear y eliminar datos, además de permitir controles de acceso.

Los usuarios pueden ser tanto humanos como máquinas que acceden a la base de datos para recoger información. En el segundo caso, las máquinas suelen valerse de APIs para hacerle llegar la información al usuario humano. En el caso de los usuarios humanos, pueden tener distintos modos de acceso (con diferentes permisos).

### 8- **Funciones de un SGBD**

La SGBD, como ya hemos dicho, actúa como interfaz entre el usuario y los datos. Sus funciones principales son:

* **Definición y control centralizado**: Descripción de los datos y de las interrelaciones de las estructuras que los almacenan. Mantiene esto en el diccionario de datos.
* **Manipulación de los datos**: mecanismos para alterar los datos, crear otros y eliminarlos. Esto debe afectar lo menos posible a las aplicaciones que interactúan con los datos.
* **Seguridad e integridad**: : mecanismos para la gestión de accesos autorizados, el control de vistas externas de usuario y el control de las restricciones, garantizando la integridad y consistencia de los datos.
* **Administración**: Copias de seguridad, sobre todo, control de accesos.
* **Control de concurrencia**, garantizando el acceso simultáneo a los datos

### 9- **Cuentas de usuario. Tipos y explicación.(normales,avanzados,programadores, admins).**

* **Usuarios normales**: Interactúan con la base de datos indirectamente. Por ejemplo, meten datos a la base al registrarse en una web.
* **Usuarios avanzados**: Interactúan directamente con la SGBD. Serían los programadores normales.
* **Desarrolladores**: Crean los SGBD.
* **Admins**: Controlan accesos, deciden sobre la estructura de la base de datos, crean las copias de seguridad, y pueden modificar la estructura física.

### **10- Tipos de lenguaje**

* **De definición de datos**: para crear datos. Create, Alter y Drop.
* **De manipulación de datos**: Eliminar datos, actualizarlos, cambiarlos. Alter, Update, Insert, Delete.
* **De control de datos**: son los comandos que generalmente usan los administradores para controlar el acceso a los datos. Ejemplos: GRANT y REVOKE.

### **11- SGBD actuales y características.(Access,mariaDB, mongoDB….)**

Access de Microsoft: manejo muy sencillo, para bases de datos muy pequeñas; fácil de usar en webs, orientado a usuarios individuales o pequeños equipos.

Apache Derby: tamaño reducido, escrito en Java, multiplataforma, muy portable, embebido o cliente/servidor.

CouchDB: orientado a documentos, NoSQL, usa JSON, licencia Apache.

DB2 de IBM: multiplataforma, usa XML. Su versión gratuita se llama DB2 Express.

FireBird: relacional, multiplataforma, bajo consumo, concurrencia óptima. Código abierto.

Informix de IBM: sencillo, confiable, relacional y multiplataforma. Usa XML. Consume menos recursos que Oracle.

MariaDB: derivado de MySQL y con licencia GPL. Multiplataforma.

MongoDB: NoSQL, usa BSON con esquema dinámico. Multiplataforma. MySQL: comercial y libre. Es relacional, multihilo, multiusuario y multiplataforma. Orientado a entornos web...

SQL Server de Microsoft: cliente/servidor, relacional, competidor de Oracle.

Oracle: multiplataforma, confiable y seguro, cliente/servidor, relacional, uso de JSON. Ofrece una versión gratuita Express Edition. Es un producto muy competitivo.

OrientDB: orientado a documentos con licencia Apache.

PostgreSQL: sistema relacional-objetos. Es el SGBD libre más avanzado del mundo. Multiplataforma.

SQLite: Relacional, accesos muy rápidos y multiplataforma.

SyBase: Relacional, altamente escalable y de bajo costo.

### **12- Procesos de producción de software.**

La norma 12207 establece diferentes procesos para producir software.

* **Adquisición**: Son las actividades que desarrolla el cliente que contrata el producto, ya que es una actividad externa al desarrollo.
* **Suministro**: hace referencia al proceso que el proveedor del producto desarrolla. Contiene los mecanismos para dar respuestas óptimas, planificando los procedimientos, detallando los recursos que gestionar con el fin de garantizar el éxito del producto.
* **Análisis**: Analizar los requisitos del cliente.
* **Diseño:** contiene todas las etapas de diseño conceptual y lógico del producto que se va a elaborar.
* **Desarrollo**: contiene las actividades de análisis de requisitos, diseño, codificación, integración, pruebas e instalación y aceptación. Una vez que se ha diseñado el producto, se comienza con las fases relacionadas con la implementación del producto y cada uno de los mecanismos para su programación.
* **Validación y verificación**: el producto comienza a usarse como si estuviera en posproducción. Se testean todos los posibles itinerarios de ejecución.
* **Explotación y mantenimiento**: el producto entra en su ejecución real, haciéndose necesario revisiones constantes y detección de necesidades de mejoras.

### **13- Fases del ciclo de vida de una Base de datos.**

* **Planificación**: Qué se quiere conseguir, cómo se quiere recolectar datos, y con qué recursos se cuenta.
* **Definición**: se especifica el ámbito y los límites de la aplicación de bases de datos, así como los sistemas con los que interactúa. Se determinan los usuarios y las áreas de aplicación.
* **Recogida y análisis de los requisitos**: se recogen y analizan los requerimientos de los usuarios y de las áreas de aplicación a través de entrevistas, observación del funcionamiento de la organización, análisis de documentos e instrumentos de recogida de información, etcétera.
* **Diseño de la base de datos**: está dividido en tres fases importantes:
  + **Diseño conceptual**: se produce un esquema conceptual independiente del nivel físico del sistema.
  + **Diseño lógico**: Se escoge el modelo de datos y el SGBD. El esquema conceptual se pasa a diagramas.
  + **Diseño físico**: atendiendo al SGBD elegido en la fase anterior, que más se adecua al sistema de información el esquema lógico se traduce en un esquema físico. En este, se tiene en cuenta las estructuras de almacenamiento y los métodos de acceso.
* **Diseño de la aplicación**: diseño de los programas de aplicación que interactúan con la base de datos, y en muchas ocasiones es paralela al diseño de esta. Se debe asegurar que los requisitos de usuario se encuentran en el diseño de la aplicación. También se puede diseñar las interfaces de usuario.
* **Prototipado**: un prototipo es un modelo de trabajo de las aplicaciones del sistema y es útil para que los usuarios puedan usar el sistema e identificar qué aspectos son o no adecuados. El prototipado ayuda al equipo de diseño e implementación a detectar incorrectas interpretaciones de los requisitos de los usuarios.
* **Implementación**: haciendo uso de los comandos del lenguaje de definición de datos se crea el esquema de la base de datos, el sistema de almacenamiento en ficheros y el esquema de los usuarios. Se implementan los procesos que permiten comunicarse con la base de datos y trabajar con los datos. También se usan los comandos de manejo de datos que son llamados desde scripts o desde aplicaciones desarrolladas con otros lenguajes de programación.
* **Conversión y carga de datos**: cuando se actualizan sistemas y se debe reemplazar los datos antiguos por nuevos incluidos en diferentes estructuras.
* **Prueba y mantenimiento**: se prueba y valida los requisitos de los usuarios, se diseñan test con datos reales y se toman medidas sobre la fiabilidad y calidad del software desarrollado.

### **14- Servidor y cliente. Definición.**

Un servidor es una máquina que centraliza las peticiones de acceso a la información. Los clientes son otras máquinas que acceden a este servidor y solicitan los datos. Este sistema es el más sencillo y el más antiguo en cuanto a la ubicación de las bases de datos.

### **15- Tipos de arquitecturas (cliente-servidor,2 niveles, 3 niveles).**

Los programas informáticos que las organizaciones usan para comunicarse con los almacenes de datos están compuestos por cuatro componentes: las **bases de datos, la lógica de transacciones, la lógica de negocios y de la aplicación de gestión de datos y la interfaz de usuarios**. Atendiendo al modo en el que estos componentes se desarrollan en el lado del cliente y en el lado del servidor se obtienen diferentes modelos.

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**, siendo a menudo un SGBD.

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**. En el nivel del **servidor, se implantan sistemas gestores de bases de datos que proveen mecanismos para la entrada y salida de datos que requiere el nivel intermedio**. Los servidores de aplicaciones gestionan la lógica de negocios y el acceso a los datos disminuyendo la complejidad del desarrollo de aplicaciones. La conexión física de las máquinas de esta arquitectura puede estar provista a través de redes locales o redes extensas.

### 

### **16- Seguridad en las bases de datos.**

Hoy en día, proteger los datos es esencial, ya que la información es muy valiosa, más en el mundo digitalizado en el que vivimos. Se usan diferentes técnicas, como firewalls, cifrado de datos… además de control de accesos. Las vistas ocultan a los usuarios no autorizados la información confidencial. Asimismo, las copias de seguridad también son importantes en el caso de que se produzcan catástrofes para evitar que la información se pierda, así como esenciales en el mantenimiento de los datos.

Los mecanismos de protección tanto lógicos como físicos deben aumentar, al hacerlo las posibles amenazas y sus consecuentes pérdidas de disponibilidad, de confidencialidad, de privacidad o de integridad.

### **17- Copias de seguridad**

Las copias de seguridad y de restauración son tareas muy importantes en el lado del servidor. El administrador y cualquier usuario con funciones destinadas a tales efectos deben establecer protocolos exhaustivos sobre el mantenimiento de los datos, la recuperación ante desastres y operatividad con datos que no intervienen en la operatividad presente de la organización.

### **18- Alta disponibilidad**

La alta disponibilidad también es una tarea importante, a través de mecanismos de balanceo de carga, duplicidad lógica o física, sistemas físicos de redundancia de datos, etcétera. De esta forma, los datos siempre están disponibles.

### **19- Cifrado**

El cifrado o encriptado de la información protege el activo de cualquier organización: los datos. Estos pueden cifrarse a través de diferentes metodologías y algoritmos, explotación de certificados digitales y uso de protocolos seguros. Los datos almacenados en los soportes pueden estar cifrados para evitar su conocimiento en caso de cualquier intrusismo. Implantar servidores de autenticación, como por ejemplo Kerberos, también mejora la seguridad de los sistemas gestores de bases de datos.

### **20- Bases de datos distribuidas**

Juegan un papel importante en la seguridad de los datos, ya que la información se almacena en diferentes servidores en vez de en un único lugar, lo que posibilita que si un servidor cae el resto no se ve afectado. Así, la pérdida de información es mínima.

### **21- Integración web**

La integración web en la nube añade una capa extra de protección porque los datos se guardan en internet además de en servidores físicos, así que si uno cae los datos no se pierden.

### **22- Pasos para diseñar una base de datos**

Para que los datos sean fieles se debe conceptualizar a través de ideas y definiciones. Esta representación se denomina **modelo conceptual.** Partiendo de este modelo se obtendrá el **esquema relacional de datos**.

Para llevar a cabo este procedimiento es necesario disponer de un conjunto de herramientas para **diseñar los datos** y sus relaciones y otras que permitan llevar las acciones con las bases de datos. **La primera es el modelo de datos y la segunda, el sistema gestor de base de datos.**

Pasos para diseñar una BD:

* Recoger los requisitos en lenguaje natural.
* Diseñar el diagrama conceptual, el de entidad-relación.
* Elegir el modelo de datos.
* Transformar el diseño conceptual a un modelo relacional.
* Normalizar aquellos elementos que tienen defectos de diseño
* Optimizar la solución obtenida en fases anteriores según criterios de almacenamiento físico, espacio de disco, tiempo de acceso a datos, etcétera.

### **23- Elementos del diagrama entidad-relación**

**Entidad**: es aquel objeto, real o abstracto, acerca del cual se desea almacenar información en la base de datos. Van dentro de cuadrados. Existen dos clases de entidades: las regulares, que tienen existencia por sí mismas, y las débiles, cuya existencia depende de otra entidad. Serían las tablas de la base de datos.

**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 el DNI. Todas las combinaciones que siguen ese patrón forman el dominio.

**Relación**: es una asociación o correspondencia existente entre una o varias entidades. Una relación se caracteriza por su nombre único, el tipo de correspondencia y la cardinalidad. Se representan con flechas (que apuntan hacia N). Hay varios tipos de relación, denominados cardinalidades:

* + 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.

**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.

### **24- Formas normales**

Cuando creas una base de datos, puedes estructurar las tablas de varias maneras. El objetivo es que los datos estén bien organizados, sin duplicados innecesarios y fáciles de modificar. Las **formas normales** son reglas que te ayudan a organizar tus tablas para lograr esto.

**Primera forma normal**: se dice que un tipo de entidad está en primera forma normal si **todos sus atributos solo toman un único valor del dominio de valores**.

Una tabla está en 1NF si cumple con los siguientes requisitos:

* **Todos los valores son atómicos**: cada columna contiene valores indivisibles. No se permite el uso de listas o conjuntos en una sola celda, sino que se crea un registro para cada valor. Si un estudiante tiene distintas asignaturas cada registro indica una de esas, aunque la PK y el nombre del alumno se repitan.
* **No hay duplicados de filas**.
* **Cada campo tiene un valor único**: no puede haber varios valores para un mismo campo en una fila.

**Segunda forma normal**: un tipo de entidad está en segunda forma normal si está en primera y cualquier atributo que no forme parte de la clave primaria depende totalmente de todos los atributos que forman la clave primaria.

**Tercera forma normal**: un tipo de entidad está en tercera forma normal si está en segunda y los atributos dependen solo de la clave primaria.

### **25. Transformaciones. Tipos nombrarlos o explicación de 1 o 2 tipos.**

En el diseño lógico, cada uno de esos elementos del diagrama conceptual deben transformarse para conseguir un conjunto de tablas compuestas por columnas y que se relacionan entre ellas. El diseño lógico es la etapa del proceso de diseño de una base de datos en la que se obtiene la representación de su estructura en términos de almacenamiento. El modelo entidad relación se usa para el modelado conceptual y el relacional para el lógico.

* **Transformación de entidades**: Cada entidad se convierte en una tabla o relación en el modelo relacional. La tabla tendrá el mismo nombre que la entidad de la que proviene. La tabla se crea usando el comando CREATE TABLE.
* **Transformación de atributos:** Cada atributo de una entidad se transforma en una columna de una tabla.
* **Identificadores**: se transforman en una columna, que es la clave primaria de la tabla.
* **Identificadores alternativos**: se transforman en una columna a la que se le añade la restricción UNIQUE, lo cual significa que no puede haber valores repetidos en esa columna.
* **Atributos no identificadores**: se transforman en una columna de la tabla.

**Transformación de las interrelaciones**

Las interrelaciones del modelo E/R se transforman en nuevas estructuras en alguna tabla, ya sea como una nueva columna para relacionar las tablas que intervienen o la creación de una nueva tabla. Estas nuevas claves que adquieren las tablas tras resolver una interrelación se llaman claves ajenas. La transformación va a depender de la multiplicidad de la interrelación:

**Interrelación N:M:** se convierte siempre en una tabla del modelo relacional cuya clave primaria, resulta de la concatenación de los atributos principales de los tipos de entidad que unía y además se comportan como claves ajenas.

**Interrelación 1:N**: se añade a la tabla con mayor cardinalidad (en la dirección de N) las columnas que comportan su clave primaria. Estas nuevas columnas formarán una clave ajena hacia ella.

**Interrelación 1:1**: se realizan del mismo modo que las 1:N, pero teniendo en consideración la cardinalidad mínima en ambos lados. Si son iguales, se puede propagar de cualquier tabla a la otra; sin embargo, si una es menor que la otra, se propaga de la de menor a la de mayor.

### **26- Tipos de relaciones.**

1:1 -> Relación 1:1. Por ejemplo, una persona y su DNI. La FK se puede situar en cualquiera de las dos tablas.

1:N -> Uno a muchos. Por ejemplo, un profesor para muchos alumnos. La FK va en la dirección de N.

N:M -> Muchos a muchos. Por ejemplo, muchos alumnos en muchas asignaturas y viceversa. La FK sale directamente de la cardinalidad, y ambas tablas tienen su FK.

### **27- Clustering**

Una técnica que optimiza los accesos es la agrupación de tablas, o también llamada clustering. Este proceso tiene como fin agrupar tablas, también organizándose de forma física en áreas cercanas en los soportes para así acelerar los accesos a los datos que contienen.

### **28- Desnormalización de tablas**

Por motivos de implementación, 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, lo contrario a normalizarlas, que es evitar redundancias.

### 

### **29- Monitorización de tablas. Definición y objetivos.**

La monitorización de bases de datos consiste en realizar un seguimiento proactivo del estado de la BD. Se controla que todo funcione correctamente, se analiza que el código y las transacciones estén bien ejecutadas, así como se vigila la automatización de los datos y la generación de informes automáticos. También se vigila todo lo relacionado con los bugs que pueda haber.

### **30- Sistemas de soporte a la decisión DSS.**

Las bases de datos se construyen para contener una gran colección de datos. Estos datos se guardan en almacenes de datos. Los almacenes de datos constituyen la arquitectura base para el sistema de soporte a decisiones y la minería de datos. Las bases de datos constituyen el acúmulo de datos que la organización requiere para sus gestiones de cada día.

Los sistemas de soporte a decisiones son sistemas que, a través del análisis de los datos almacenados, se explotan para la toma de decisiones.

### **31- Almacenes de datos**

Los almacenes de datos reúnen los datos de varios orígenes bajo un esquema unificado en un solo sitio. A veces, es necesario aglutinar esa ingesta de datos en divisiones más pequeñas con el fin de centrarlos por temas o áreas de negocio, y así se puedan tomar decisiones más localizadas. Este subconjunto de datos de todo el DataWarehouse se denomina Data Mart. Se consultan mediante herramientas OLAP (On Line Analytical System).

Componentes:

**Modelo de datos corporativos**: los esquemas de cada base de datos deben integrarse obteniendo un único modelo de datos que identifique y estructure los requisitos de información del almacén de datos. Se obtiene un único modelo conceptual.

**Limpieza y carga de datos operativos**: se reconoce los sistemas OLTP que intervienen, se corrigen errores de inconsistencia, redundancias inútiles, se definen tipos de datos, se documenta y se cargan desde las bases de datos hacia el almacén.

**Almacén de datos**: contiene todos los datos actuales en un momento determinado. Es necesario equilibrar el alto coste de cargas operativas con la actualización de los datos.

**Extracción y recuperación de datos**: mecanismos de acceso a datos, su visualización, análisis y resultados en informes. Se usará para ello la base de datos y el diccionario de datos.

**Metadatos**: que describen la estructura de los datos y algoritmos para crear los resúmenes. Independientemente de estos componentes es necesario **establecer la arquitectura física del almacén de datos**. Esta puede ser de diferentes tipos.

**Arquitectura de un almacén de datos**

* **Arquitectura centralizada**: se usa un único servidor para guardar todo el almacén de datos. Maximiza la potencia de cálculo y facilita el mantenimiento, pero las consultas pueden consumir muchos recursos. La inseguridad aumenta.
* **Arquitectura distribuida**: los datos del almacén se reparten entre varios servidores. Se suelen distribuir por temas lógicos. La carga está distribuida, lo que mejora la ejecución de las consultas, pero el mantenimiento se hace más complejo. A cambio, la seguridad es mayor porque si un servidor cae el resto de la información no se ve afectada.
* **Distribución por niveles**: se eligen diferentes servidores en función del nivel de detalle de los datos que contienen, datos de detalle, datos resumidos y datos muy resumidos. Los datos muy resumidos se pueden repartir en varios servidores para aumentar la disponibilidad de los datos. El acceso a los datos que más se requieren es muy rápido sin penalizar las consultas a los datos de detalles.

### 

### **32- Tipos de sistemas de información.**

Se utilizan a la hora de analizar datos para tomar posteriormente decisiones varias.

* Los sistemas de procesamiento de transacciones, TPS, Transaction Processing System: almacenar y procesar las transacciones operacionales que se usarán en los sistemas de apoyo a la toma de decisiones. El concepto de **transacción** se introduce para **gestionar la complejidad que conlleva los múltiples accesos al mismo recurso de información**. Estos procesos, considerados de ejecución concurrente, **pueden acarrear errores sobre la fiabilidad de los datos**, mantenimiento y recuperación ante fallos. **Se llama transacción a un conjunto de operaciones que se deben hacer todas juntas para que no afecten al acceso de los datos**. Esta unidad lógica permite asegurar la integridad de los datos.
* Los sistemas de información administrativa, MIS, Management Information System: facilitar la información de interés sobre la situación general del negocio a la gerencia de la organización.
* Los sistemas de soporte a la decisión, DSS, Decission Support System: combinar analizar los datos para resolver problemas.
* Los sistemas de soporte a ejecutivos, EIS, Executive Information System: extraer información de interés para las metas estratégicas.
* Los sistemas para la toma de decisiones en grupo, GDSS, Group Decission Support System: compartir información para las decisiones conjuntas.
* Sistemas expertos de soportes a la toma de decisiones, EDSS, Expert Decision Support Systems: desarrollan conocimiento en áreas concretas y actúan como consultores expertos.

### **33- Sistemas OLTP (Sistemas On-Line Transaction Processing)**

Estos sistemas se caracterizan porque **sistematizan las transacciones en sus bases de datos** y usan sistemas online para la inserción de grandes volúmenes de datos denominados datos operativos.

Los sistemas OLTP tienen características que hacen que no sean tecnologías óptimas para los sistemas de toma de decisión ya que los datos proceden de fuentes muy heterogéneas, lo que provoca falta de consistencia. Además, tienden a organizarse por criterios geográficos. Se priorizan las operaciones de actualización e inserción de datos sobre la resolución de consultas. Puede ocurrir que la velocidad de entrada de datos es mayor que el tiempo de respuesta a los accesos concurrentes de los usuarios.

### **34- Componentes de un DSS**

Los sistemas de soporte a la decisión son una herramienta de BI cuyo objetivo es el análisis de datos. Supone una herramienta de explotación de bases de datos orientadas al almacenamiento Data Warehouse. corporativo o Data Mart de inmenso tamaño.

Los componentes principales de un DSS son:

* **Componente de diálogo**: el conjunto de opciones para el usuario para dirigir las acciones del sistema.
* **Componente de datos**: los datos suponen la entrada para el procesamiento de modelos y lo manejan las fuentes de datos, y los propios almacenes de datos.
* **Componente modelo**: suministra la capacidad de análisis. Los más usados son los de optimización, los descriptivos, los probabilísticos, los estratégicos, los tácticos y los operacionales.

### **35- Minería de datos**

La minería de datos, o Data Mining, es la técnica que recopila datos en la búsqueda de reglas y estructuras a partir de los datos localizados en un almacén de datos. El objeto es la búsqueda del conocimiento en las bases de datos. La plataforma con la que trabaja la minería de datos es el almacén de datos. Busca datos, patrones, intenta predecir cómo se van a comportar.

Se pretende construir patrones o categorías.

* Se incluye el análisis de información aplicado en los sistemas de soporte a la decisión.
* Se automatizan los procesos de planteamiento y descubrimiento de hechos e hipótesis.
* Se pretende predecir futuras tendencias y comportamientos.
* Se buscan relaciones y patrones globales ocultos en los almacenes de datos.

El procedimiento para resolver un problema a través de la minería de datos se divide en dos grandes etapas: la preparación de los datos y la minería de datos propiamente dicha.

Primero se parte del planteamiento del problema definiéndolo y determinando los recursos, las fuentes de información y su disponibilidad. Se seleccionan las fuentes válidas, realizando un tratamiento y estructuración de la información, eliminando la información no útil y detectando otra que falta.

**Minería de datos**: se seleccionan las técnicas, los algoritmos en búsqueda de patrones, se seleccionan los modelos y parámetros apropiados. Se interpretan los resultados en busca de modelos útiles en la toma de decisión y se evalúan usando diferentes casos de aplicación. Se cruzan datos, técnicas OLAP y se utilizan técnicas de machine learning, así como modelos multidimensionales.

* Análisis multidimensional: se cruzan datos de múltiples formas y con distintos niveles de agregación. Se basan en las bases de datos multidimensionales y técnicas OLAP.
* **Agentes inteligentes**: análisis para detectar patrones y relaciones. Se usan técnicas con metodologías basadas en sistemas expertos y el aprendizaje automático.
* **Consultas e informes**: son las metodologías tradicionales a través de consultas, y emplean la estadística para el análisis multidimensional y agentes inteligentes. Son metodologías muy útiles para bases de datos relacionales.
* **Detección de alarmas**: se ejecutan agentes para desencadenar acciones extraordinarias.

36-