

2020-Final_19-02-2020

2.a)Diferencias entre una funcion y procedure

Diferencias entre funciones de usuario y stored procedures:

- Un stored procedure sí puede actualizar datos de una o más tablas. Una función de usuario no puede actualizar nada (sí puede realizar una consulta de datos).
- Un stored procedure no puede usarse en un SELECT. Una función de usuario sí puede usarse en un SELECT, en un UPDATE o bien en un DELETE.
- Un stored procedure puede devolver 1 único parámetro como salida. Una función de usuario debe devolver sí o sí 1 único parámetro como salida.

2.b)Desarrollar un algoritmo de ordenamiento sobre un vector

```
if down < up
then begin {si down>=up, el vector es ordenado }
    { y se retorna al programa que }
    { ha hecho la llamada }
rearrange(down,up,j);
    { reorganiza los elementos del }
    { subvector tal que uno de los }
    { elementos (posiblemente x[down] }
    { está ahora en x[j] (j es un }
    { parámetro de salida) y: }
    { 1. x[i]<=x[j] para down<=i<j }
    { 2. x[i]<=x[j] para j<i<=up }
    { x[j] está ahora en su }
    { posición final }
quick(lc,j-1);
    { ordena el subvector entre }
    { x[down] y x[j-1] }
quick(j+1,up)
    { ordena el subvector entre }
    { x[j+1] y x[up] }
end {termina then begin}
```

2019-Final_26_de_Febrero

a) Explique el concepto, la funcionalidad y para que se utilizan los triggers en una base de datos relacional

Triggers: Los triggers son un mecanismo de la base de datos opcionales asociados a una tabla que ejecutan un conjunto de sentencias SQL cuando un evento ocurre. Los eventos pueden ser INSERTS, UPDATES o DELETES. Puede definirse el momento en que se ejecutará el mecanismo : AFTER (luego de que el evento ocurra), BEFORE (antes del evento), INSTEAD OF (en vez de ejecutar el evento) o FOR EACH ROW (por cada línea). Pueden utilizarse para realizar deletes en cascada, para replicación automática de datos y con respecto a la seguridad: realizar autorizaciones de seguridad y registrar logs de auditoría.

b) Explique que significa la sigla ACID y ejemplifique cada concepto

Las siglas ACID refieren a 4 propiedades que deben cumplir las transacciones del DBMS. Estas siglas se refieren a Atomicidad (capacidad de una transacción de ejecutarse en su totalidad o no ejecutarse), Consistencia (sólo pueden guardarse datos válidos), Isolation (cada transacción está aislada de las demás, es decir no puede afectarla directamente), Durabilidad (una vez que se realiza el commit de la transacción, ésta perdurará en el tiempo).

2019-Final_25_de_Septiembre

a) Desarrolle 2 objetos de una base de datos que sirven para brindar seguridad

Vistas: Las vistas son un conjunto de columnas ya sean reales o virtuales, pertenecientes a una misma tabla o no, con un filtro determinado o no. Éstas toman el resultado de una consulta y lo tratan como si fuera una tabla pero a diferencia de las últimas, no contienen datos almacenados ni ocupan espacio de almacenamiento. Se pueden utilizar para ocultar la complejidad de los datos, para simplificar sentencias al usuario, para presentar los datos desde una perspectiva diferente y con respecto a la seguridad: restringir el acceso a un número determinado de filas y/o columnas.

Triggers: Los triggers son un mecanismo de la base de datos opcionales asociados a una tabla que ejecutan un conjunto de sentencias SQL cuando un evento ocurre. Los eventos pueden ser INSERTS, UPDATES o DELETES. Puede definirse el momento en que se ejecutará el mecanismo : AFTER (luego de que el evento ocurra), BEFORE (antes del evento), INSTEAD OF (en vez de ejecutar el evento) o FOR EACH ROW (por cada línea). Pueden utilizarse para realizar deletes en cascada, para replicación automática de datos y con respecto a la seguridad: realizar autorizaciones de seguridad y registrar logs de auditoría.

2019-Final_12_de_Febrero

a) Defina el concepto de transaccion en un Sistema de Base de Datos

Las transacciones son un bloque de sentencias SQL que están delimitadas por el inicio de transacción y un commit. Este bloque debe cumplir con las propiedades ACID (Atomicidad, Consistencia, Isolation, Durabilidad) que se refieren a que la transacción debe ejecutarse en su totalidad o no ejecutarse, que sólo debe guardar datos correctos, cada transacción debe estar aislada de las demás y a que una vez que se llega al COMMIT, los datos persistirán en el tiempo.

b) Desarrolle el concepto de Grafos, su posible utilizacion, clasificacion y representacion computacional de los mismos.

Es una pareja $G=(V,A)$, donde V es un conjunto de vértices (o nodos) y A es un conjunto de pares que une dichos vértices, llamadas aristas. Sirve para establecer relaciones. Pueden mantener o no jerarquía entre sí, y en caso de hacerlo, dicha jerarquía se representa poniendo un sentido al arco.

Los grafos pueden ser dirigidos o no dirigidos, y pueden ser restrictos o irrestrictos.

La representación computacional se puede dar por estructuras estáticas como dinámicas. Entre las estáticas se encuentran la matriz de adyacencia, y la matriz de incidencia. Entre las dinámicas se pueden representar por Listas, como PFaltz, Siklosi, o listas de adyacencias. La estructura estática es menos conveniente en casi todos los casos, porque ocupa más espacio ($O(V^2)$) y no es flexible para el crecimiento del grafo, pero tiene menos costo computacional. La estructura dinámica es más conveniente debido a que ocupa menos espacio ($O(V+A)$), pero tiene un costo computacional mayor.

2019-Final_04_de_Diciembre

a)Desarrolle dos objetos de bases de datos que puedan garantizar la integridad de la misma

Constraints: Los constraints son utilizados para asegurar que se cumplan distintos tipos de restricciones y las reglas de integridad. Para garantizar la integridad de entidades (cada fila de una tabla debe tener una única forma de identificarse) se utilizan los PK Constraints, para garantizar la integridad referencial (asegurar la coherencia de datos entre dos tablas) se utilizan los FK Constraints y para garantizar la integridad semántica (asegura que los datos tengan una correcta configuración) se utilizan los Constraints CHECK, NOT NULL, UNIQUE, DEFAULT y DATA TYPE.

Triggers: Los triggers son un mecanismo de la base de datos opcionales asociados a una tabla que ejecutan un conjunto de sentencias SQL cuando un evento ocurre. Los eventos pueden ser INSERTS, UPDATES o DELETES. Puede definirse el momento en que se ejecutará el mecanismo : AFTER (luego de que el evento ocurra), BEFORE (antes del evento), INSTEAD OF (en vez de ejecutar el evento) o FOR EACH ROW (por cada línea). Pueden utilizarse para realizar deletes en cascada, para replicación automática de datos, realizar autorizaciones de seguridad, registrar logs de auditoría y con respecto a la integridad utilizarlos para asegurar la integridad relacional entre dos tablas.

b)Explique el concepto de ISOLATION LEVEL. Mencione las diferencias entre cada nivel.

Se puede determinar el nivel de aislamiento de las transacciones, es decir, el grado en que éstas se afectarán unas a otras. Se pueden determinar cuatro niveles: READ UNCOMMITTED que es el más permisivo de todos permite lecturas sucias, no repetibles y fantasmas. No tiene bloqueos; READ COMMITTED es menos permisivo que el anterior ya que no permite lecturas sucias pero si no repetibles y fantasmas. Éste establece un bloqueo ante una lectura pero lo desbloquea apenas obtiene los datos

commiteados; REPEATABLE READ es menos permisivo que CR. No permite lecturas sucias ni no repetibles pero si lecturas fantasmas. Establece un bloqueo ante una lectura y desbloquea cuando se hace el commit de la transacción; y por último SERIALIZABLE READ que no permite ni lecturas sucias, ni no repetibles ni fantasmas. Éste se bloquea ante una lectura o escritura y se desbloquea al commitear la transacción.

2018-Final_17_de_Julio

a)Relacione y explique los conceptos de transaccion, lockeos y aislamiento

Las transacciones son un bloque de sentencias SQL que están delimitadas por el inicio de transacción y un commit. Este bloque debe cumplir con las propiedades ACID (Atomicidad, Consistencia, Isolation, Durabilidad). Se puede determinar el nivel de aislamiento con el que van a trabajar las transacciones de la forma READ UNCOMMITTED, READ COMMITED, REPEATABLE READ y SERIALIZABLE READ, que dependiendo de cada una permitirán o no lecturas sucias, no repetibles y/o fantasmas. Para que se cumplan dichas propiedades de aislamiento se utilizan las herramientas de "lockeo" que pueden ser a nivel fila, bd, página o tabla y además pueden ser exclusivos, compartidos o promovibles.

b)Explique la Arquitectura Interna de un Sistema de Base de Datos

La arquitectura interna puede ser definida mediante el modelo ANSI/SPARC. Éste define a la arquitectura en 3 capas: Nivel Externo o Vistas, Nivel Conceptual y Nivel Interno o Físico. El Nivel Externo es la percepción que tiene el usuario sobre la base de datos, en el Nivel Conceptual define una forma 'entendible' de la información de la base de datos (definición de tipos de datos, restricciones, reglas de integridad) y el Nivel Interno es una representación de bajo nivel de la BD que especifica cómo se almacenarán los datos en disco.

2018-Final_21_de_Febrero

a)Explique el objeto vista, su definicion, para que casos lo usaria, relacionelo con la seguridad e integridad

Las vistas son objetos de las bases de datos. Son conjuntos de columnas, ya sean reales o virtuales, que no ocupan (alocan) espacio de almacenamiento, no contienen datos almacenados, sino que están definidas por una consulta que implica una o varias tablas. Se utilizan para suministrar un nivel adicional de seguridad restringiendo el acceso a un conjunto de datos, ocultar complejidad de datos, simplificar sentencias. Al crear una view el usuario debe tener permisos de select sobre las columnas de las tablas involucradas. No se puede adicionar a una view las cláusulas de order by o union, tal como tampoco se pueden definir triggers sobre una vista.

Se pueden utilizar para casos en los que se ven involucradas tablas con información sensible. Por ejemplo tengo la tabla cliente que tiene el nombre, numero de cuenta bancaria, cuanto dinero dispone en ella y su sucursal principal; y otra tabla que tiene la información de las sucursales. Si yo quisiese que un grupo de usuarios puedan ver que clientes tienen asignadas ciertas sucursales, podría crear una vista que se encargue de traer esta información ocultando los datos sensibles. Obteniendo así que los usuarios consulten sobre la vista, y sobre ella puedan hacer sus sub búsquedas siendo que en ningún momento ellos tuvieron conocimiento de la información sensible ni la lógica o complejidad que hubo por detrás para obtener dicha información.

b) Explique el concepto de Dominio y las reglas de Integridad de Entidad, Referencial y Semántica

El dominio son todos los valores reales posibles que pueden tomar los registros.

Integridad de Entidad: es usada para garantizar que los datos pertenecientes a una misma tabla tienen una única manera de identificarse. Que tenga una primary key única y no nula. Se valida unicidad y existencia.

Integridad Referencial: es usada para garantizar que la coherencia entre datos de dos tablas. El uso de foreign keys hace que dado una o más columnas se estará referenciando a la primary key de otra. Si vos quieres borrar un registro que está siendo referenciado por otra tabla, ocurrirá un error.

Integridad Semántica: es usada para garantizar el sentido y configuración de los datos que vamos a almacenar, y que estos respeten las restricciones de dominio o sobre los atributos. Ej, DATA TYPE, DEFAULT, UNIQUE, NOT NULL, CHECK

2017-Final_12_de_Diciembre

a) Determinar que recursos brindan en líneas generales los motores de bases de datos para optimizar la performance de las queries.

Indices. Los índices son una estructura asociada a una tabla o vista. La principal función es aumentar el rendimiento cuando se hacen consultas a esa tabla. Es decir, lo que posibilita el índice es que el DBMS no recorra toda la tabla en búsqueda de los datos pedidos, sino que a través del índice, se recorran un subconjunto de los datos. La principal utilización es la de acelerar las consultas a las bases de datos.

2017-Final_25_de_Julio

b) Comparar procesos OLTP vs OLAP (DDS) en cuanto a su utilización de bases de datos

OLTP y OLAP son los dos sistemas de tratamiento transacciones online. OLTP es un procesamiento transaccional mientras que OLAP es un sistema de procesamiento analítico.

OLTP es un sistema que gestiona aplicaciones orientadas a transacciones en Internet, por ejemplo, ATM. OLAP es un sistema en línea que reporta a consultas analíticas multidimensionales como informes financieros, pronósticos, etc. La diferencia básica entre OLTP y OLAP es que OLTP es un sistema de modificación de base de datos online, mientras que OLAP es un sistema de respuesta de consulta de base de datos online.

2017-Final_11_de_Julio

a)Desarrolle el concepto de indice. Tecnicas de creacion de los mismos.

Ventajas y desventajas de cada una de ellas.

Los índices son una estructura asociada a una tabla o vista. La principal función es aumentar el rendimiento cuando se hacen consultas a esa tabla. Es decir, lo que posibilita el índice es que el DBMS no recorra toda la tabla en búsqueda de los datos pedidos, sino que a través del índice, se recorran un subconjunto de los datos. La principal utilización es la de acelerar las consultas a las base de datos.

Mientras que hashing permite mayor velocidad de acceso directo, árbol B entrega mejor performance a la hora de realizar accesos secuenciales. Ambos permiten claves duplicadas, pero solo árbol B acepta una clave que no sea completa. Mientras que árbol B desperdicia mucho espacio en nodos y punteros, hashing solamente necesita al menos una tabla de índices.

b)Desarrolle el concepto de Funcion y Store procedure en una Base de Datos Relacional. Indique usos y diferencias entre ambos objetos de base de datos.

Una función definida por el usuario es una rutina de Transact-SQL o Common Language Runtime (CLR) que acepta parámetros, realiza una acción, como un cálculo complejo, y devuelve el resultado de esa acción como un valor.

Los stored procedures (procedimientos almacenados) son los equivalentes a las rutinas, procedimientos, o funciones disponibles en casi todos los lenguajes de programación, allí se utilizan los datos que se encuentran en las tablas, se realizan operaciones aritméticas o lógicas sobre esos datos y se devuelve el resultado de ese procesamiento.

Diferencias entre funciones de usuario y stored procedures:

- Un stored procedure sí puede actualizar datos de una o más tablas. Una función de usuario no puede actualizar nada (sí puede realizar una consulta de datos).
- Un stored procedure no puede usarse en un SELECT. Una función de usuario sí puede usarse en un SELECT, en un UPDATE o bien en un DELETE.
- Un stored procedure puede devolver 1 único parámetro como salida. Una función de usuario debe devolver sí o sí 1 único parámetro como salida.

2017-Final_12_de_Mayo

b) Explique cada uno de los diferentes niveles de aislamiento.

Estos son los cuatro niveles de aislamiento para transacciones:

READ UNCOMMITTED: No tiene bloqueos por tabla. Es el más performante, pero sufre de lecturas sucias, no repetibles y fantasmas, por lo cual, ante una misma consulta puedo obtener filas que no fueron commiteadas en una transacción.

READ COMMITTED: Tiene bloqueos compartidos por tabla. Soluciona lecturas sucias, pero no sufre de lecturas no repetibles y fantasmas, por lo que ante una misma consulta puedo obtener filas distintas con valores desactualizados.

REPEATABLE READ: Tiene bloqueos exclusivos por tabla, pero no son por rango, por lo cual soluciona lecturas no repetibles, pero no así las fantasmas, por lo que ante una misma consulta puedo obtener filas de más o de menos. Puede causar deadlock.

SERIALIZABLE: Tiene bloqueos exclusivos por tabla y por rangos, por lo no tiene problemas de lectura, es el más consistente, y las transacciones parecen ocurrir en serie. Pero puede causar deadlock: un hilo de ejecución 1 quiere lo que tiene el hilo de ejecución 2 y viceversa, pero no puede acceder al recurso porque están bloqueados simultáneamente.

2016-Final_20_de_Diciembre

a) Explicar la utilidad de obtener el plan de ejecución de un select de consulta de datos

Plan de Ejecución de un SELECT

Antes de ejecutar una consulta, el motor de base de datos la optimiza. El resultado de esa optimización es lo que se conoce como plan de ejecución. Las decisiones que toma el motor para optimizar incluyen: qué algoritmo de join usar, qué algoritmo de ordenamiento usar, qué algoritmo de agrupación usar o si recorrer la tabla de forma secuencial o usando índices. Todas estas decisiones mejoran la performance de la consulta. Por ejemplo, no es lo mismo utilizar un algoritmo de join especializado para tablas de tamaño pequeño que usarlo para tablas de tamaño mayor

Una de las principales características de revisar y analizar el plan de ejecución de una consulta es poder ver cuales son los motivos por el cual la misma se demora en devolver los resultados. Básicamente te permite detectar la falta de una primary key, o un índice en una columna, ya que la misma te regresa un "Table scan". A grandes rasgos la principal utilidad de el plan es detectar esos pequeños errores de diseño de base de datos.

El optimizador de consultas es el componente del sistema de gestión de base de datos que intenta determinar la forma más eficiente de ejecutar una consulta SQL.

Se decide cuál de los posibles planes de ejecución de una consulta dada es el más eficiente.

Los optimizadores basados en costos asignan un costo (operaciones de E/S requeridas, CPU, etc.) a cada uno de esos planes y se elige el que tiene menor costo.

El optimizador no puede ser accedido directamente por los usuarios, sino que, una vez enviadas las consultas al servidor, pasan primero por el analizador y recién entonces llegan al optimizador.

Una consideración muy importante es el orden de los "join".

b) Que objetos y funciones dispone un motor de base de datos para el manejo de seguridad de los datos.

Para el manejo de seguridad, un motor de base de datos dispone de:
triggers, permisos, vistas, procedimientos

2016-Final_14_de_Mayo

a) Compare los metodos de Hashing y Arbol B en cuanto a su eficiencia en la recuperacion de datos

Hashing no utiliza espacio extra para su implementación, a diferencia de Arbol B que almacena la estructura en memoria.

Hashing busca establecer relacion entre valor de los datos y valor de la clave con una función hash, en cambio Arbol B trabaja con el concepto de Arboles n-arios de busqueda, guarda en memoria la estructura y es mas lento para la creación de indices

b) Explique la funcionalidad de un ETL y el concepto de staging area en una implementacion de Data Warehouse

Extract, Transform and Load («extraer, transformar y cargar», frecuentemente abreviado ETL) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data mart, o data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

ODS (Operational Data Store) o Staging Area: es la capa intermedia que servirá como almacenamiento entre los sistemas fuente y el Data Warehouse. Su misión es ofrecer la posibilidad de gestionar los datos con el formato y estructura de origen para facilitar la integración y transformación hacia el modelo desnormalizado que poseen los Datamarts y el Data Warehouse. Hay que tener en cuenta que los sistemas fuente presentan toda la información separada por tablas, mientras que en destino la información se compila en una estructura común, que facilita en gran medida, no sólo las búsquedas, sino también la comprensión de los datos que contienen.

2016-Final_01_de_Marzo

a) Explicar los conceptos de Data Warehouse y Data Mining y el objetivo de utilizar cada uno de ellos.

Un concepto muy importante dentro de la Inteligencia de Negocios es el de minería de datos, también conocido como Data Mining. Este se refiere al proceso de explorar grandes cantidades de información en busca de patrones o tendencias que den luz sobre un comportamiento de la data en un contexto determinado. Para llevar a cabo este proceso, las empresas requieren de herramientas especializadas, como por ejemplo, Data Warehouse.

Data Warehouse: Se trata del lugar donde toda la data de una compañía es almacenada. Consiste en un sistema computarizado con una gran capacidad de almacenamiento, esencial para reunir y organizar la información proveniente de los distintos departamentos de la organización.

2018-Final_30_de_Julio

a) Explique el concepto de índice de HASH

2015-Final_15_de_Diciembre

a) Defina el concepto de lectura sucia, repetible y fantasma

Una lectura sucia ocurre cuando se le permite a una transacción la lectura de una fila que ha sido modificada por otra transacción concurrente pero todavía no ha sido cometida.

Las lecturas sucias funcionan de modo similar a las lecturas no repetibles; sin embargo la segunda transacción no necesita ser cometida para que la primera dé un resultado diferente. Lo único que se puede prevenir en el nivel de aislamiento LECTURAS NO COMETIDAS es que las actualizaciones aparezcan en desorden en el resultado; esto es, que las primeras actualizaciones siempre aparecerán antes que las actualizaciones posteriores.

Una *lectura no repetible* ocurre cuando en el curso de una transacción una fila se lee dos veces y los valores no coinciden.

El efecto de *lecturas no repetible* puede ocurrir en una implementación de concurrencia mediante bloqueos cuando no se efectúan éstos al hacer un SELECT, o cuando los bloqueos se liberan nada más terminar la operación SELECT. Cuando se usa el método MVCC, las *lecturas no repetibles* pueden aparecer cuando se relaja el requisito de que al cometer una transacción afectada por un conflicto ésta deba deshacerse.

Una *lectura fantasma* ocurre cuando, durante una transacción, se ejecutan dos consultas idénticas, y los resultados de la segunda no son iguales a los de la primera.

Esto puede ocurrir cuando no se realizan *bloqueos de rango* al realizar una operación **SELECT ... WHERE**.

La anomalía de las *lecturas fantasma* es un caso particular de las *lecturas no repetibles* cuando la transacción 1 repite una consulta acotada en rango **SELECT ... WHERE** y, entre ambas operaciones la transacción 2 crea (i.e. **INSERT**) nuevas filas (en la misma tabla) que entran dentro de esa cláusula **WHERE**.

b) Que es un data mart, explique por que motivo posee datos desnormalizados.

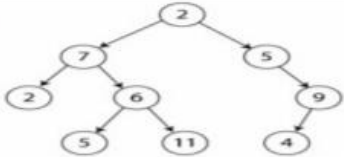
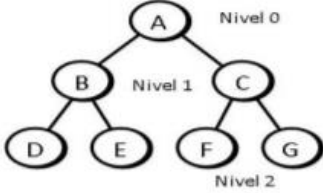
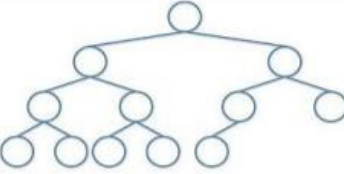
2015-Final_28_de_Julio

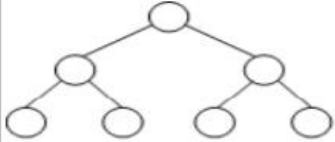
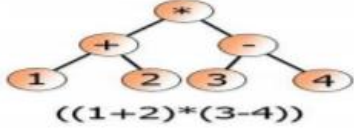
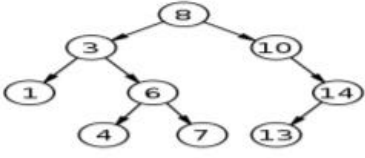
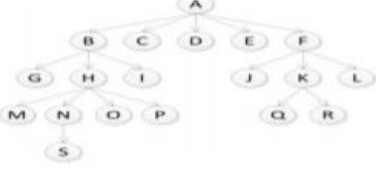
a) Explique los conceptos de Fact Table y Dimension Table en DW

2014-Final_16_de_Diciembre

a) Describa los diferentes conceptos que se relacionan directamente con la funcionalidad de concurrencia.

En Resumen

Nombre	Definición	Propiedades	Imagen
Árbol Binario	Un árbol binario es un árbol en el que ningún nodo puede tener más de dos hijos, a los cuales se los distingue como izquierdo y derecho.	Altura promedio: $O(\sqrt{N})$	
Árbol Binario Perfecto	Árbol binario con las siguientes propiedades: 1) Si $h=0$, entonces $T_L=\emptyset$ y $T_R=\emptyset$. 2) Por el contrario, si $h>0$, ambos T_L y T_R son árboles binarios perfectos de altura $h-1$.	Una de las propiedades más importantes de estos árboles es que un árbol binario perfecto de altura h tiene exactamente $2^{(h+1)} - 1$ nodos internos. Peor caso de búsqueda: $O(\log n)$.	
Árbol Binario Completo	Árbol binario en el que todos los niveles están llenos, excepto posiblemente el último nivel, el cual es completado de izquierda a derecha.	Un árbol binario completo de altura $h \geq 0$ contiene al menos 2^h nodos y a lo sumo $2^{(h+1)} - 1$ nodos.	

Árbol Binario Balanceado	Un árbol binario está completamente balanceado si está vacío, o ambos subárboles están completamente balanceados y tienen la misma altura. Por lo tanto cualquier camino desde la raíz a una hoja tiene la misma longitud.	Los únicos árboles que están perfectamente balanceados son los árboles binarios perfectos. En todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha. Complejidad de búsqueda: $O(\log n)$.	
Árboles de expresión	Árboles binarios utilizados para representar expresiones algebraicas.	Los nodos terminales o hojas son las variables o constantes en la expresión. Los no terminales son los operadores.	 ((1+2)*(3-4))
Árbol Binario de Búsqueda	Árbol que soporta las operaciones de búsqueda, inserción y eliminación de forma eficiente. Árbol que satisface la propiedad de búsqueda ordenada.	Las claves siguen un criterio de orden que determina donde una determinada clave puede ubicarse en el árbol en relación con las otras claves en ese árbol. Tiempo de búsqueda medio: $O(\log n)$ Tiempo de búsqueda peor: $O(N)$	
Árboles de búsqueda M-arios	Consiste de n subárboles y n-1 claves. Un nodo que tiene M hijos contiene exactamente M-1 claves. Hay $K=(M-1)n$ claves.	Las claves en cada nodo son distintas y están ordenadas. $h \geq \lceil \log_M(K-1) \rceil - 1$. Si elegimos M grande, un nodo puede ocupar un bloque de disco completo.	
Árboles B	Árbol vacío o árbol M-ario que cumple: 1) La raíz de T tiene al menos dos subárboles y a lo sumo M subárboles. 2) Todos los nodos internos de T (menos la raíz) tienen entre $\lceil M/2 \rceil$ y M subárboles. 3) Todos los nodos externos de T están al mismo nivel.	Obj. principal: minimizar las operaciones de entrada y salida hacia el disco. Al imponer la condición de balance, el árbol es restringido de manera tal que se garantice que la búsqueda, la inserción y la eliminación de nodos sean todos de tiempo $O(\log N)$. Principales operaciones: búsqueda, inserción y eliminación.	