INDEXACION

Indices

- El índice de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla.
- Al aumentar la velocidad de acceso, se suelen usar sobre aquellos campos sobre los cuales se hagan frecuentes búsquedas.

Indices

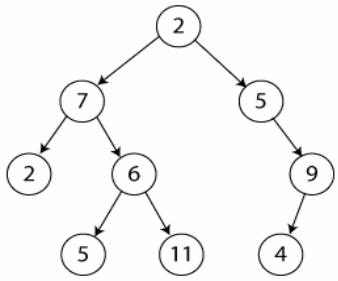
- Los índices guardan:
  - ✓ El elemento que se desea indexar.
  - ✓ La posición en la tabla.
- Para buscar un elemento que esté indexado, sólo hay que buscar en el índice dicho elemento para una vez encontrado, devolver el registro que se encuentre en la posición marcada por el índice.

Indices

- Los índices pueden ser creados usando una o más columnas, proporcionando la base tanto para búsquedas rápidas al azar como de un ordenado acceso a registros eficiente.
- Los índices son construidos sobre árboles B y B+.

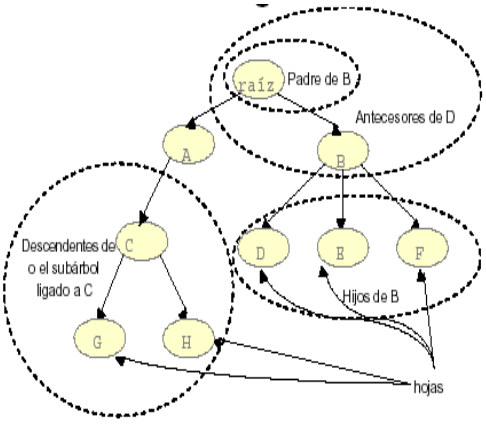
Arbol

- Un árbol es una estructura de datos que imita la forma de un árbol (un conjunto de nodos conectados)
- Un nodo es la unidad sobre la que se construye el árbol y puede tener cero o más nodos hijos conectados a él



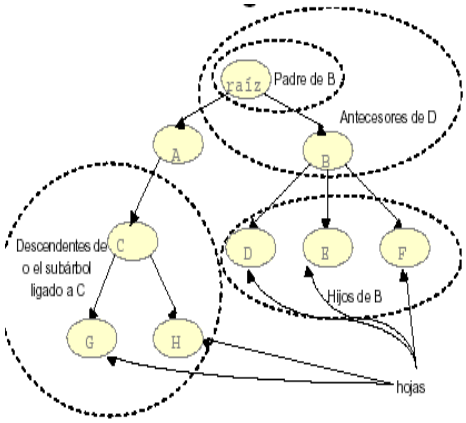
Terminología en Arboles

- Un árbol consta de *nodos* conectados.
- Cada árbol (salvo un *árbol vacío* degenerado) cuenta con un nodo distinguido llamado *raíz*.
- No puede haber rutas circulares en las conexiones de un árbol, de tal forma que sólo puede existir una ruta única desde cada nodo hasta la raíz.



Terminología de Árbol

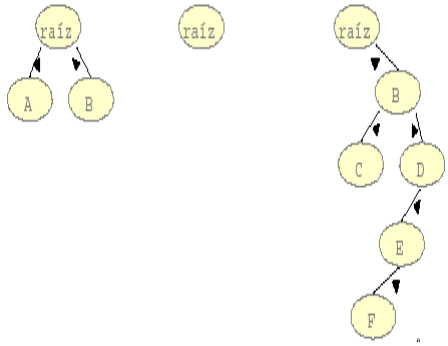
- Todos los nodos conectados a un nodo concreto son *hijos* o bien el *padre* de dicho nodo.
- Si el nodo conectado se encuentra en la única ruta a la raíz, dicho nodo recibe el nombre de padre. Todos los nodos, salvo la raíz, tienen un único padre.
- El resto de nodos conectados a un nodo concreto son los hijos del nodo.



Arbol Binario

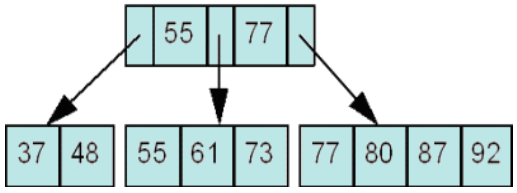
- Un tipo de árbol habitual y de gran utilidad es el llamado *árbol* binario, que permite que un nodo tenga, al menos, dos hijos.
- A continuación se incluye una definición formal del árbol binario que hace hincapié en el carácter recursivo del árbol:

  Un árbol binario es un árbol vacío, o bien un nodo raíz con subárboles formados por árboles binarios a la derecha y a la izquierda.



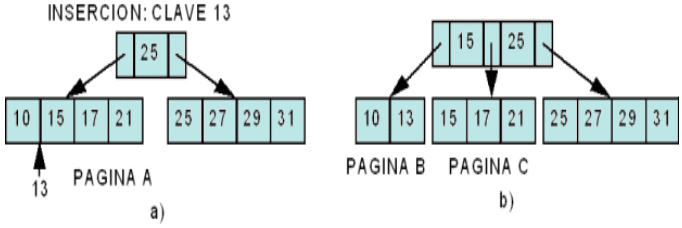
Arbol B+

- Los árboles B+ se han convertido en la técnica mas utilizada para la organización de archivos con índices. La principal característica de estos árboles es que todas las claves se encuentran en las hojas y por lo tanto cualquier camino desde la raíz hasta alguna de las claves tienen la misma longitud



Arbol B+

- **La dificultad se presenta cuando desea insertarse una clave en una pagina que se encuentra llena (m = 2d). En este caso, la pagina afectada se divide en 2, distribuyéndose las m + 1 claves de la siguiente forma:**
- **" las d primeras claves en la pagina de la izquierda y las d + 1 restantes claves en la pagina derecha ". Una copia de la clave del medio sube a la pagina antecesora**



Indices

- **Mejoran el acceso a los datos almacenados**
- **Empleados para clasificación y recuperación de datos**

Tabla Alumnos

#Alumno	Al_Apellido	Al_Nombre	Especialidad
100	Pérez	Juan	Ingeniería de Sistemas
200	López	María	Ingeniería de Sistemas
300	Sánchez	Elena	Ingeniería Electrónica
400	Pérez	Andrés	Ingeniería Electrónica
500	Valdivia	Carlos	Ingeniería de Sistemas
600	Zumaeta	José	Ingeniería Electrónica
700	Montes	Fernando	Ingeniería de Sistemas

Al_Apellido	#Alumno
López	200
Montes	700
Pérez	100, 400
Sánchez	300
Valdivia	500
Zumaeta	600

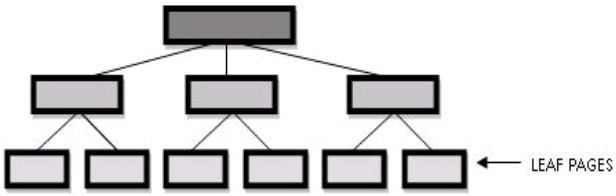
Indice por Apellido

Indice por Especialidad

Especialidad	#Alumno
Ingeniería de Sistemas	100, 200, 500, 700
Ingeniería Electrónica	300, 400, 600

Indices

- El diseño y creación de índices (indexes) es muy importante cuando se quiere mejorar la performance del Servidor de Base de datos



Indices

- Los índices no son considerados como parte del diseño lógico de la base de datos, por eso, pueden se adicionados, removidos y cambiados sin afectar el esquema de la BD

- Tipos de índices:
  - Clustered
  - Non Clustered

Indices - Tipos

Indice Clustered

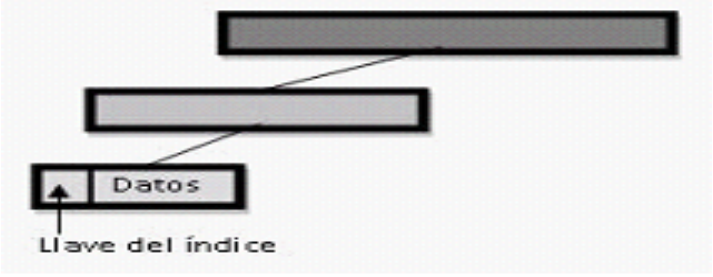
- Es un índice en el cual el orden físico de las filas corresponde al orden de las filas en el índice
- Sólo se puede tener un índice clustered por tabla
- Las operaciones de UPDATE y DELETE son mas rápidas con el uso de estos índices

Indices - Tipos

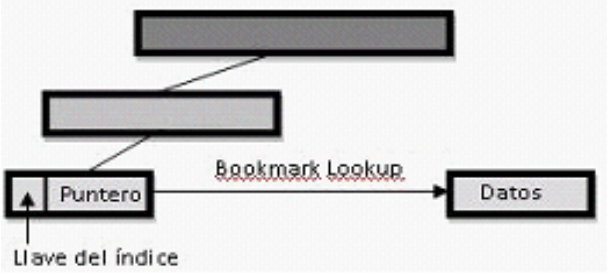
Indices Non-Clustered

- En este índice, se especifica el orden lógico de una tabla para poder acceder a los datos de forma eficiente
- El orden físico de una tabla no corresponde al orden de estos índices
- Se puede definir múltiples nonclustered indexes por tabla

Búsqueda en Indices Clustered



Búsqueda en Indices No Clustered



Creación de Indice

CREATE
[UNIQUE] [CLUSTERED | [NONCLUSTERED]
INDEX índice
ON tabla(campo [ASC|DESC][, campo [ASC|DESC], ...])

Ejemplos: CREATE INDEX

- En el ejemplo siguiente se crea un índice en la tabla Clientes con el campo IdCliente. Dos registros no pueden tener los mismos datos en el campo IdCliente y no se permiten los valores Nulos.

```
CREATE UNIQUE INDEX DNIClien
ON Clientes (DNICliente)
WITH DISALLOW NULL;
```

Ejemplos: CREATE INDEX

- En el ejemplo siguiente se crea un índice en la tabla Clientes con el campo IdCliente. Dos registros no pueden tener los mismos datos en el campo IdCliente y no se permiten los valores Nulos.

```
CREATE CLUSTERED INDEX IdClien
ON Clientes (IdCliente)
WITH DISALLOW NULL;
```

Ejemplos: CREATE INDEX

- En el ejemplo siguiente se crea un índice, que consta de los campos TeléfonoTrabajo y Extensión, en la tabla Empleados:

```
CREATE NONCLUSTERED
 INDEX ÍndiceNuevo
 ON Empleados (TelTrabajo,
  Extensión);
```

Operaciones de Indexación Online

- La opción de indexado online permite modificaciones concurrentes (actualizaciones, borrados e inserciones) en las tablas subyacentes o datos con índices cluster y de cualquier índice asociado durante la ejecución de DDL de indexación
- Por ejemplo, mientras se está reconstruyendo un índice cluster, se puede seguir haciendo actualizaciones a los datos y consultas sobre estos datos

OPTIMIZACION DE CONSULTAS

FACULTAD de INGENIERÍA de
SISTEMAS E INFORMÁTICA

Objetivos

- Revisar las consultas y resolver problemas de rendimiento
  - Revisar un plan de ejecución
  - Identificar consultas mal elaboradas
  - Revisar un plan de ejecución
  - Mantener y optimizar índices

Problema

- **En una institución internacional con muchos miembros a nivel mundial se realizan búsquedas utilizando las primeras letras de sus apellidos**
- **Los datos de la columna Apellidos de la tabla Miembros utilizan mayúsculas y minúsculas combinándolas y para encontrar a los miembros se utilizaba la función UPPER en la consulta por Apellidos**

Consulta Problema

```
SELECT Apellidos
FROM Miembros
WHERE UPPER(Apellidos) LIKE 'MAN%'
```

- Premisa, no se puede modificar el juego de caracteres del servidor que es sensitivo

- ¿ Porque esta consulta es notablemente lenta ?

Una Solución al Problema

```
ALTER TABLE Miembros
ADD ApellidoMayus AS UPPER(Apellidos)
GO
CREATE NONCLUSTERED INDEX
IXMiembrosApellidoMayus
ON Miembros(ApellidoMayus)
GO
```

Una Solución al Problema

```
SELECT Apellidos
FROM   Miembros
WHERE ApellidosMayus LIKE 'MAN%'
```

Otros Motivos de Consultas Lentas

- ✓ Problemas de red
- ✓ Memoria inadecuada en el equipo servidor o falta de memoria disponible para el servidor
- ✓ Falta arreglo de discos (RAID)
- ✓ Falta de estadísticas útiles
- ✓ Falta de índices útiles
- ✓ Falta de particiones útiles

Problemas de Red

- ❑ El problema de rendimiento de las consultas puede estar relacionado con un componente distinto a las mismas consultas
- ❑ Por ejemplo, el problema se puede deber al rendimiento lento de la red
- ❑ También se debe monitorear:
  - La actividad de los discos
  - La actividad de la CPU
  - El uso de la memoria

Estadísticas Utiles

- **Las estadísticas para la optimización de consulta son objetos que contienen información estadística acerca de la distribución de valores en una o más columnas de una tabla**
- **El optimizador de consultas utiliza estas estadísticas para estimar la cardinalidad, o número de filas en el resultado de la consulta**
- **Estas estimaciones de cardinalidad habilitan al optimizador de consultas para crear un plan de consulta de alta calidad**

Estadísticas Utiles

Para actualizar la información sobre la distribución de las claves en una tabla especifica se utiliza UPDATE STATISTICS

```
UPDATE STATISTICS SalesOrderDetail;
GO
UPDATE STATISTICS SalesOrderDetail
AK_SalesOrderDetail_rowguid;
GO
```

Índices Utiles

- Si se utiliza un gran número de índices en una tabla, el rendimiento de las instrucciones INSERT, UPDATE, DELETE y MERGE se verá afectado, ya que todos los índices deben ajustarse adecuadamente a medida que cambian los datos de la tabla
- Evite crear demasiados índices en tablas que se actualizan con mucha frecuencia y mantenga los índices razonables, es decir, defínalos con el menor número de columnas posible

Índices Utiles

- Utilice un número mayor de índices para mejorar el rendimiento de consultas en tablas con pocas necesidades de actualización, pero con grandes volúmenes de datos.
- Un gran número de índices contribuye a mejorar el rendimiento de las consultas que no modifican datos, ya que el optimizador de consultas dispone de más índices entre los que elegir para determinar el método de acceso más rápido

Índices Utiles

- La indexacion de tablas pequeñas puede no ser una solución óptima, porque se tarda más tiempo en realizar la búsqueda de los datos a través del índice que en realizar un simple recorrido de la tabla
- De este modo, es posible que los índices de tablas pequeñas no se utilicen nunca; sin embargo, sigue siendo necesario su mantenimiento a medida que cambian los datos de la tabla

Plan de Ejecución

- **Un plan de ejecución es el conjunto de pasos que tiene que realizar el DBMS para ejecutar una consulta**
- **Un plan de ejecución de una consulta SQL es una definición de:**
  - **La secuencia en la que se tiene acceso a las tablas de origen**
  - **La forma como utilizara los índices de las tablas**

Optimización

- El proceso de selección de un plan de ejecución entre varios planes posibles se conoce como optimización
- El optimizador de consultas es uno de los componentes más importantes de un sistema de base de datos SQL

Identificación de consultas mal elaboradas

- Las herramientas que se pueden utilizar son:
  - *SQL Server Management Studio*
  - *El analizador de SQL Server*
  - El asistente para la optimización del motor de base de datos
  - Las vistas de administración dinámicas

Identificación de consultas mal elaboradas

```
USE Ciclismo
GO
SELECT DISTINCT  *
FROM  Empleados
WHERE SUBSTRING(Apellido,1,1) = 'P'
```

Identificación de consultas mal elaboradas

- ✓ Ingresar al SQL Server Profiler
- ✓ Identificar un trace
- ✓ Seleccionar:
  - ExistingConnection
  - TSQL:SQL:BatchCompleted
- ✓ Escribir un query
- ✓ Revisar la traza

Identificación de consultas mal elaboradas

Realizar la revisión de la traza para:

```
USE Ciclismo
GO
SELECT  *
FROM  Empleados
WHERE
  Apellido  LIKE  'P%'
```

Identificación de consultas mal elaboradas

¿ Analizar a que pueden deberse las diferencias de rendimiento entre ambas consultas ?

Análisis de Planes de Ejecución

- Si se quiere determinar porque tarda tanto en ejecutarse y que esta causando el problema de una consulta se tiene que examinar su plan de ejecución
- Se puede utilizar las siguientes herramientas:
  - Las opciones de la instrucción SET
  - Las clases de eventos del analizador
  - El plan de ejecución grafico de SQL Server Management Studio

Plan de Ejecución Actual y Estimado

- Se puede ver los siguientes planes:
  - El plan de ejecución de un plan para una consulta que fue ejecutada (plan actual)
  - El plan de ejecución de cómo se ejecutara una consulta (el plan estimado)

Plan de Ejecución Actual y Estimado

```
USE AdventureWorks
GO
SELECT *
FROM  HumanResources.Employee
WHERE  SUBSTRING(NationalIDNumber,1,1) = '8'
UNION
SELECT  *
FROM  HumanResources.Employee
WHERE Title like 'P%'
```

Plan de Ejecución Actual y Estimado

```
USE AdventureWorks
GO
SELECT  *
FROM  HumanResources.Employee
WHERE  NationalIDNumber LIKE '8%' OR
          Title LIKE 'P%'
```