

INTELIGENCIA DE NEGOCIOS

INDICES

ÍNDICES

HEAP.

- Es una tabla sin un índice clustered.
- Los datos no son almacenados en un orden específico
- SQL Server utiliza IAM-Index Allocation Map para mantener los heap.
- Páginas IAM:
 - Contiene información acerca de donde almacena SQL Server los niveles del heap. La tabla de sistema sys.partitions almacena un apuntador a la primer página IAM asociada



ÍNDICES

HEAP.

- Habilita la navegación a través del heap hasta encontrar espacio vacío al momento de realizar una inserción.
- Asocia las páginas de datos con las tablas.

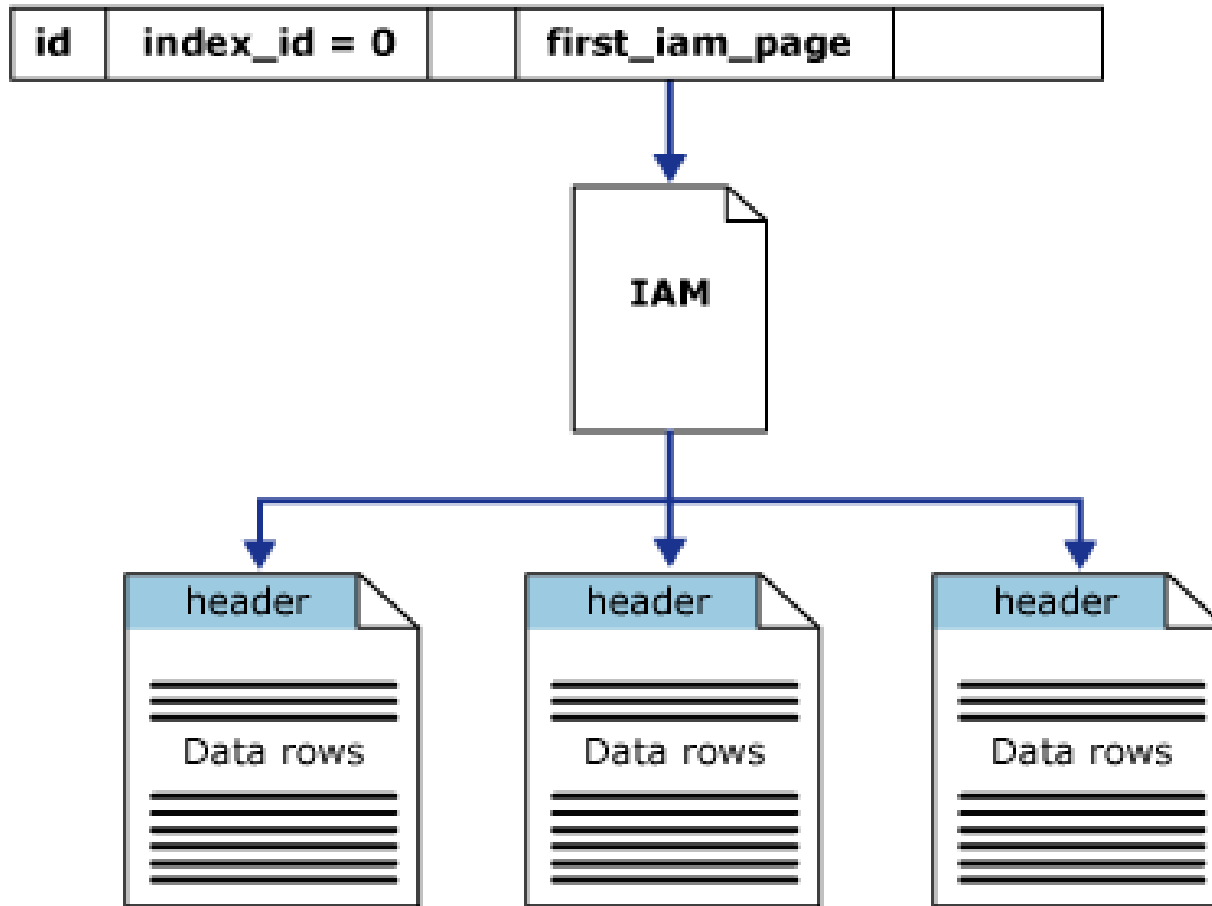
Cuando utilizar un HEAP:

- Contienen datos volátiles, donde los registros son añadidos, eliminados y actualizados frecuentemente. (La sobrecarga del mantenimiento del índice puede ser más costoso que los beneficios)
- Contiene duplicidad de información
- Existe mucha escritura y poca lectura.

EJEMPLO: LOG de Auditoría.



HEAP



HEAP - EJERCICIO

1. Crear Base de Datos: BD_HEAP
2. Crear tabla: Tabla_Heap

Tabla_HEAP	
Nombre	Tipo
id_heap	int
dato_heap	char(3000)

3. Insertar Seis registros / 3 páginas.
4. Borrar un registro
5. Insertar un registro y validar que ocupa el lugar (página) del registro borrado en el punto 4

INDICES

¿Cómo accesa SQL Server los Datos?

1. Escaneando todas las páginas de datos (Escaneo de tabla)
 - Comienza en el inicio de la tabla.
 - Escanea de página a página a través de todos los renglones de la tabla
 - Extrae los registros que corresponden al criterio de la consulta
2. Usando índices.
 - Recorre la estructura de árbol de índices hasta encontrar los renglones que corresponden al criterio de la consulta
 - Extrae solo los renglones necesarios.



ÍNDICES

SQL detecta si existe un índice. Entonces el optimizador de consultas se responsabiliza de la generación del plan de ejecución para la consulta, determinando si deberá realizarse por un escaneo o usando un acceso más eficiente a través de un índice.



ÍNDICES

- Clustered
- Non-Clustered.

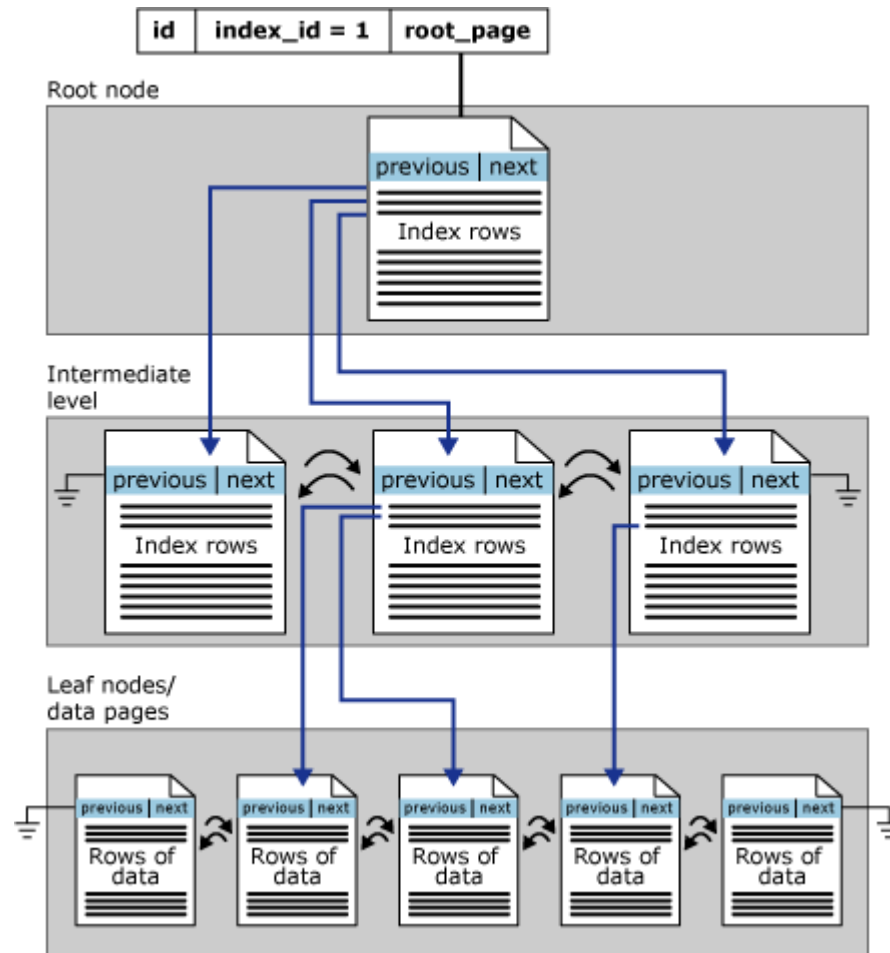
Índices Clustered.

Ideal para buscar datos que sus valores están juntos (escaneo de tabla)

- Cada tabla puede tener un solo índice Clustered.
- Existen nodo raíz, nodo intermedio (página de índices), nodo de datos.
- Las páginas en cada nivel tienen una doble liga. (eficientar búsquedas)



INDICES CLUSTERED



ÍNDICES

¿Cuándo utilizar un Índice Clustered?

- Se desea obtener información entre un rango de valores usando el operador BETWEEN, <, <=, >, >=
- Regresar datos ordenados utilizando clausulas ORDER BY o GROUP BY
- Regresar datos combinando el uso de clausulas JOIN y llaves foráneas.
- Regresar gran cantidad de datos en el resultado.



ÍNDICES

¿Cuándo NO utilizar un Índice Clustered?

- Los datos en las columnas índice cambiarán frecuentemente
- Las llaves índices están compuestas por muchas columnas de gran tamaño.



ÍNDICES

- Índices **Clustered**.

Ejemplo: Páginas Blancas del directorio telefónico.

index clustered por apellido(s) y Nombre.

Utilidad: Buscar un número telefónico de forma muy eficiente.



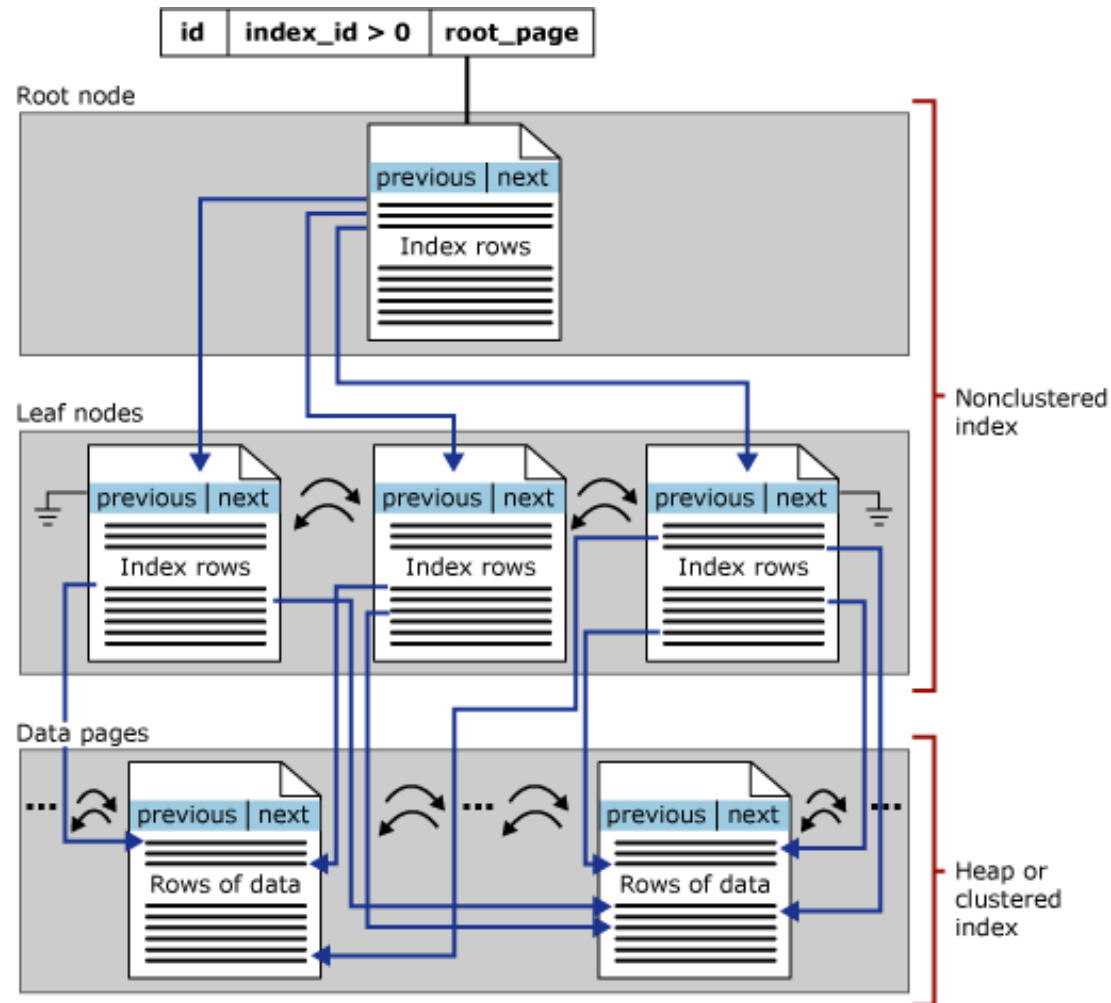
ÍNDICES.

Non-Clustered.

- Índices que mantienen un sub-conjunto de las columnas de la tabla en orden.
- Estos índices no modifican el orden de las filas de la tabla, mantienen una lista ordenada de referencias a filas de la tabla original.
- Se pueden crear hasta 249 índices
Non_Clustered.



NON-CLUSTERED



ÍNDICES.

Cuando utilizar un índice Non-Clustered:

- Cuando el usuario requiere múltiples formas de encontrar los datos. Ejemplo se desea buscar un cliente por su ID y por su apellido paterno + nombre.
 - Crear índice Clustered del ID
 - Crear índice Non-Clustered del Apellido Paterno + Nombre.
- Provee rendimiento en consultas que utilizan JOIN o GROUP BY.
- La tabla contiene un volumen muy grande de información pero pocas actualizaciones a sus datos
- Tienes conocimiento de que las consultas no regresaran gran cantidad de información



ÍNDICES.

Cuando utilizar un índice Non-Clustered:

- Requieres columnas indexadas que contienen diferentes valores como por ejemplo la combinación del primer apellido con el nombre.
- Requieres columnas índices que son frecuentemente envueltas en condiciones de búsqueda de un query – tales como la clausula WHERE .

RECOMENDACIÓN: Crear primero índice CLUSTERED y luego los NON-CLUSTERED



ÍNDICES

Índices UNIQUE:

- Es un índice que asegura que todos los datos indexados en una columna no contienen duplicados. (permite un NULL)

EJEMPLO:

```
CREATE UNIQUE NONCLUSTERED INDEX  
[Ak_Employee_LogINIO] ON  
[HumanResources].[Employee] ([LoginID] Asc)
```



ÍNDICES

Índices Compuestos.

- Es aquel formado dos o más columnas de la tabla
- Puede incluir hasta 16 columnas y 900 bytes en la llave.
- La clausula WHERE de una consulta debe referenciar a la primer columna del índice compuesto.
- Las columnas del índice compuesto deben pertenecer todas a la misma tabla.



ÍNDICES

Cuando crear un índice Compuesto.

- Dos o más columnas son mejores para una búsqueda como una llave.
- Las consultas referencian solo las columnas en el índice.



ÍNDICES

○ Incluidos:

- ❑ un índice no clúster se puede ampliar incluyendo columnas sin clave además de las columnas de clave de índice.
- ❑ Los índices que incluyen columnas sin clave resultan especialmente útiles cuando abarcan la consulta. Esto significa que los índices contienen todas las columnas a las que se hace referencia en la consulta
- ❑ Se utilizan en un índice no clúster para evitar sobrepasar las limitaciones actuales de tamaño del índice de un máximo de 16 columnas de clave y un tamaño máximo de las claves de índice de 900 bytes

```
CREATE INDEX IX_3 ON [dbo].[Datos1]  
(Numero) INCLUDE (Descripcion,ID)
```



ÍNDICES

○ Incluidos (Ejemplo):

```
USE AdventureWorks14;
```

```
GO
```

```
SELECT AddressLine1, AddressLine2, City, StateProvinceID,  
PostalCode FROM Person.Address
```

```
WHERE PostalCode BETWEEN N'98000' and N'99999';
```

```
-- CREACION DEL INDICE
```

```
CREATE INDEX IX_Address_PostalCode ON Person.Address  
(PostalCode) INCLUDE (AddressLine1, AddressLine2, City,  
StateProvinceID);
```



ÍNDICES

- **Filtrados (SQL Server 2008):**

- Índices que aplican solo a un grupo de registros.

```
create index IX4 on mtro_ventas (cliente,importe)  
where importe < 1000
```



ÍNDICES

○ Filtrados (Ejemplo):

```
USE AdventureWorks2008R2;
```

```
GO
```

```
SELECT ProductAssemblyID, ComponentID, StartDate  
FROM Production.BillOfMaterials WHERE EndDate IS NOT  
NULL AND ComponentID = 5 AND StartDate > '01/01/2008' ;
```

```
GO
```

```
CREATE NONCLUSTERED INDEX
```

```
FIBillOfMaterialsWithEndDate ON
```

```
Production.BillOfMaterials (ComponentID, StartDate)
```

```
WHERE EndDate IS NOT NULL ;
```

```
GO
```



ÍNDICES

FILLFACTOR

Permite el manejo de un porcentaje (0-100) de espacio libre en las páginas de los índices. Este porcentaje determina cuantos niveles de páginas deben llenarse.

Un factor de 65 porciento, deja 35 porciento de espacio libre para nuevos registros.

- Usar FILLFACTOR bajo en aplicaciones OLTP
- Usar FILLFACTOR alto en aplicaciones OLAP



ÍNDICES

Comandos para obtener información de índices:

SP_HELP

SP_HELPINDEX

sys.indexes

sys.index_columns

Sys.stats

sys.stats_columns



EJEMPLO: INDICES

- Ejemplo 1: Creación de índices
 - Empleados (**SQL Server Management Studio**)
- Ejemplo 2:
 - Creación Índices Clustered y Non-Clustered



PRÁCTICA

- Realizar la práctica de creación de Índices:

Tarea_Indices.PDF

