Tema 5: Acceso Multi-Clave

- Introducción: Definición de búsqueda multiclave
- Acceso Invertido
- Organizaciones Invertidas (optimización de indizada)
 - ➤ Listas invertidas
 - > Esquemas de Bits
 - ➤ Comparativa



Tema 5.1: Introducción

El acceso selectivo tiene dos vertientes:

- 1. el resultado del acceso es único (unívoco). Clave de búsqueda única
- 2. el resultado puede afectar a varios registros. " secundaria
- 1. El acceso a través de claves de búsqueda alternativas no unívocas es costoso. Se puede reducir gracias al establecimiento de índices.
- 2. Si una condición de selección involucra varias de estas claves (acceso multiclave) el problema se complica y el coste sube.
- » Las soluciones a esta necesidad pueden partir de organizaciones básicas tradicionales, o de otras desarrolladas expresamente para este tipo de accesos



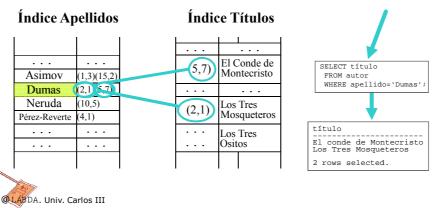
Tema 5.2: Acceso Invertido

- El acceso invertido es un tipo de acceso indizado orientado a optimizar el coste de acceso en procesos muy concretos
- ⇒ El acceso invertido procurará averiguar toda la información accediendo sólo a los índices (sin acceder al archivo de datos)
- → Precisa de punteros que localicen completamente un registro → punteros con parte alta (*bloque*) y parte baja (*posición en el bloque*)



Tema 5.2: Acceso Invertido

- Se ejecutarán primero las condiciones (para obtener conjunto resultado)
- Después se busca en los 'índices objetivo' (incógnitas), ¡pero al revés!: a partir de cada dirección (ptro. relativo) buscamos el valor (ptro. lógico)
- Ejemplo: ¿Cuáles son los títulos libros del autor cuyo apellido es 'Dumas'?



Tema 5.3: Listas invertidas

- El acceso invertido es eficiente si requiere acceder a pocos índices (si accediera a varios no contenidos en M_{int}, costaría más que acceder a los datos)
- Los índices involucrados pueden ser primarios o secundarios
- Para los índices secundarios existe una estructura específica:

Las Listas Invertidas

Una lista invertida para un campo de un fichero consiste en una <u>lista ordenada</u> de todos los <u>posibles valores</u> de ese campo, a los que se asocian referencias a todos los registros que llevan ese valor.

Ejemplo:



Tema 5.3: Listas invertidas

- Son entradas de índice (*'como un índice secundario agrupado por valores'*) (o bien 'como un índice que en lugar de uno tiene varios punteros relativos)
- Se trata de un índice ordenado (por la clave)
- Cada puntero consta de partes *alta* (dir. bloque) y *baja* (pos. en el bloque)
- Esas entradas de índice tienen <u>tamaño variable</u> (de clave y de lista de punteros) (posible implementación: long_clave + clave + long_lista + lista_punteros)

Se dice que un fichero está **invertido** si tiene índices invertidos para todos sus campos (totalmente invertido) o bien solamente para algunos de ellos (parcialmente invertido)



Tema 5.3: Esquemas de bits

Un esquema de bits para un campo es un vector de valores booleanos, de modo que a cada posible valor se le hace corresponder una posición

Ejemplo: idioma (castellano, inglés, francés, alemán, italiano) el esquema de idioma para Juan Valdés: (10000)

- Si el número de valores posibles de un campo fuera pequeño, se podría mantener un índice que en lugar de entradas (valor, puntero) tuviera (puntero, esquema de bits).
- Se pueden considerar esquemas de varios campos concatenados

1811 0001000000,0010,10100,10 departamento categoría idioma sexo



Tema 5.3: Esquemas de bits

• Los esquemas de bits pueden hacer referencia a uno o varios campos (fichero parcialmente invertido) o incluso a todos (totalmente invertido)

Ejemplo:

	departamento categ. idioma sexo
1811	0001000000 0010 10100 10
1008	0001000100 0100 10000 10
1205	0001000000 0010 11100 01

• Sobre estos esquemas, los accesos multiclave son muy eficientes. Ejemplo: empleados del departamento de informática que sepan inglés

 $Q = 0001000000\ 0000\ 01000\ 00$ resultado: serán los punteros cuyo esquema S cumpla: S and Q = Q



Tema 5.3: Comparativa

- El acceso por esquema de bits es muy eficiente
 - entradas de tamaño fijo
 - operaciones de actualización inmediatas (muy eficientes)
 - comprobación de la entrada por aplicación de máscaras (eficiente)
- Pero queda <u>limitado a</u>:
 - claves en las que el dominio (posibles valores) sea reducido (para evitar esquemas de bits demasiado largos)
 - condiciones sencillas (igualdad/desigualdad)
 - ficheros de volumen medio-bajo (no demasiados registros) (¡procesa todas las entradas serialmente!)



Tema 5.3: Comparativa

- El acceso por <u>listas invertidas</u> es complementario: es más eficiente cuanto mayor es el dominio (número de posibles valores)
 - → a más valores más listas
 - → cada lista tendrá menos punteros (listas invertidas más pequeñas).
 - → al ser más pequeñas, son más manejables (por ejemplo, si se implementan en árbol B, mayor será el orden del árbol)
- •Además:
 - buen comportamiento con volúmenes grandes (número elevado de registros)
 - las condiciones que pueden aplicarse son algo más complejas (p.e., rangos)
- Sin embargo, el mantenimiento y manejo son muy costosos:
 - actualizaciones costosas (desplazamiento de entradas, ...)
 - manejo de mucha información (en Mppal y secundaria)
 - algunas condiciones de selección siguen siendo muy costosas.

