

Tema I.- Diseño Físico

Ficheros

Índices

1

Fundamentos de sistemas de bases de datos. Elmasri, R; Navathe, S. Addison-Wesley

[cap.13 en 5ª edic.]

Índice

- Registros / Ficheros
- Operaciones sobre ficheros
- Ficheros de registros desordenados (heap)
- Ficheros de registros ordenados
- **Técnicas de dispersión (fichero hash)**
 - **Dispersión interna (memoria)**
 - **Dispersión externa (disco)**

Técnicas de dispersión (hash)

- Proporcionan un acceso muy rápido a los registros bajo una condición de igualdad sobre el **campo de dispersión** (campo hash).
 - Si el campo es clave se denomina clave de dispersión (**clave hash**)
- Un registro con campo hash de valor K se almacena en la posición i , donde $i=h(K)$, y h es la **función hash**.
- Tipos de dispersión:
 - Interna (en memoria)
 - Externa (en disco)

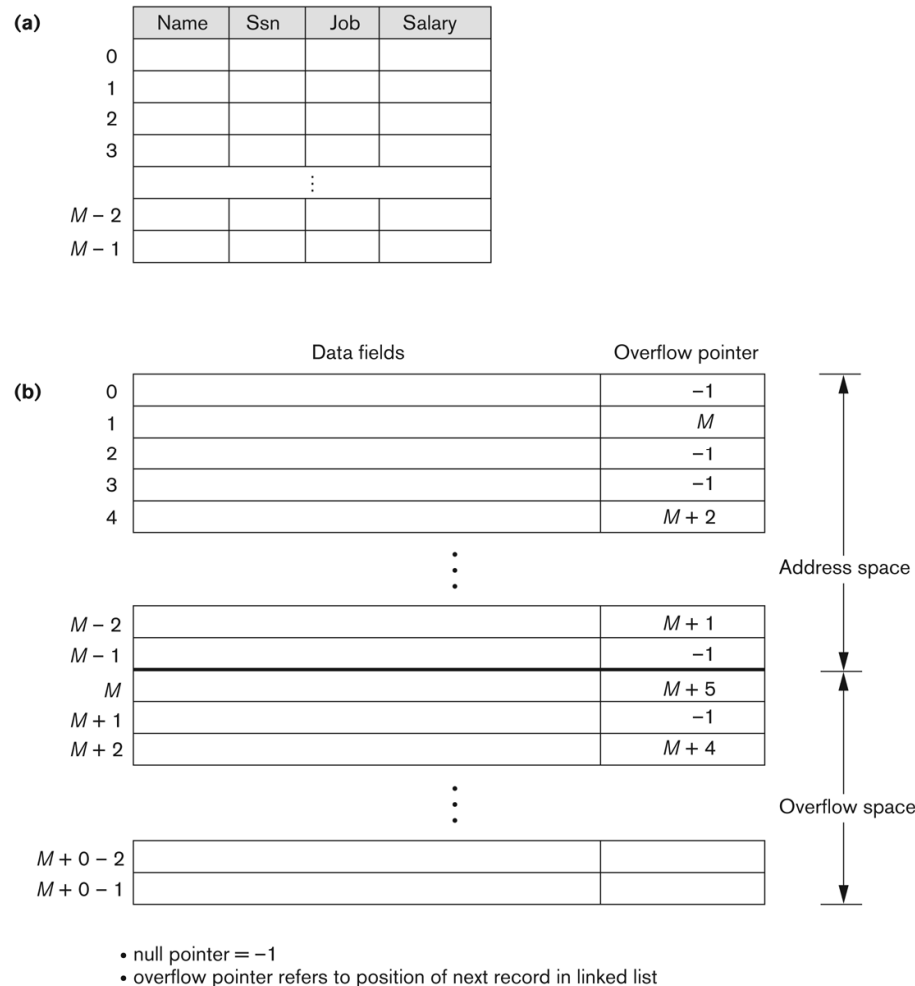
Dispersión interna (memoria)

- La tabla hash es un **array de registros**
 - Cada elemento del array es un registro de datos
- Resolución de colisiones:
 1. **Direccionamiento abierto**
 - Se recorren las siguientes posiciones hasta encontrar una vacía y se coloca.
 2. **Encadenamiento**
 - El array se extiende con posiciones de desbordamiento.
 - Se añade un puntero a cada ubicación de registro.
 - El registro se coloca en una ubicación de desbordamiento libre.
 - Se mantiene una lista enlazada de registros de desbordamiento por cada dirección hash.
 3. **Dispersión múltiple**
 - Se aplica una segunda función (llamada **rehash**).
 - Si se produce otra colisión, se utiliza direccionamiento abierto u otra tercera función, y después direccionamiento abierto, si fuese necesario.

Dispersión interna(memoria)

Figure 13.8

Internal hashing data structures. (a) Array of M positions for use in internal hashing.
(b) Collision resolution by chaining records.



a) Array de M posiciones para hashing interno

b) Resolución de colisiones por encadenamiento

Ejercicio dispersión interna

Un fichero con CODCLI como clave de direccionamiento calculado, que contiene registros con los siguientes valores de CODCLI.

Cargar estos registros en un array de tamaño 8 en el orden dado, empleando la función de direccionamiento calculado $h(K) = K \bmod 8$.

a) Resolver las colisiones utilizando la técnica de **encadenamiento**

Registro	K	h(K)
record1	7107	3
record2	844	4
record3	1540	4
record4	4800	0
record5	826	2
record6	7110	6
record7	1821	5
record8	2376	0
record9	2002	2
record10	4981	5
record11	962	2
record12	2084	4
record13	2306	2

ESPACIO DE DIRECCIONES

Posición	Registros	puntero
0	record 4	9
1		-1
2	record 5	10
3	record 1	-1
4	record 2	8
5	record 7	11
6	record 6	-1
7		-1

ESPACIO DE DESBORDAMIENTO

Posición	Registros	puntero
8	record 3	13
9	record 8	-1
10	record 9	12
11	record 10	-1
12	record 11	14
13	record 12	-1
14	record 13	-1

b) Resolver las colisiones por **direccionamiento abierto**

ESPACIO DE DIRECCIONES

Posición	Registros
0	record4
1	record8
2	record5
3	record1
4	record2
5	record3
6	record6
7	record7

A partir del registro 9 (inclusive) no se incluyen más registros

Ejercicio dispersión interna

3 – Supongamos un fichero CLIENTES con *Numcli* como clave de direccionamiento calculado, que contiene registros con los siguientes valores de *Numcli*.

Cargar estos registros en un array de tamaño 8 en el orden dado, empleando la función de direccionamiento calculado $h(K) = K \bmod 8$.

c) Resolver las colisiones por **dispersión múltiple** aplicando una segunda función de dispersión $h_2(K) = K \bmod 2$

Registro	K	h(K)
record1	7107	3
record2	844	4
record3	1540	4
record4	4800	0
record5	826	2
record6	7110	6
record7	1821	5
record8	2376	0
record9	2002	2
record10	4981	5
record11	962	2
record12	2084	4
record13	2306	2

ESPACIO DE DIRECCIONES

Posición	Registros
0	record3 (*)
1	record4 (**)
2	record5
3	record1
4	record2
5	record7
6	record6
7	record8 (***)

A partir del registro 9 (inclusive) no se incluyen más registros porque el fichero está lleno y no hay zona de desbordamiento

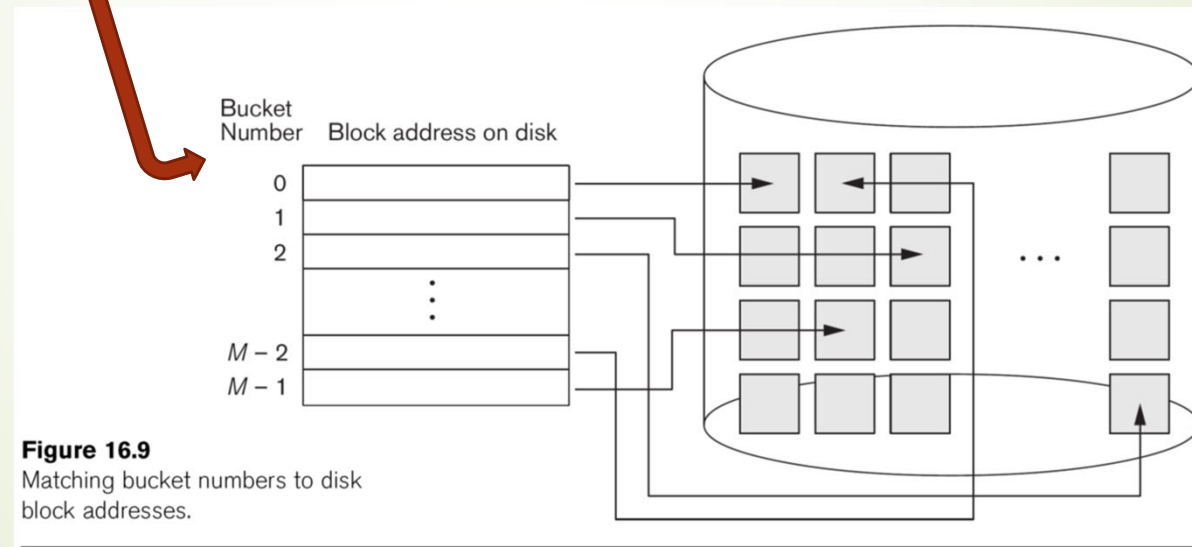
(*) $h(1540) = 4 \Rightarrow$ colisión $\Rightarrow h_2(1540) = 0 \Rightarrow$ se coloca en la posición 0

(**) $h(4800) = 0 \Rightarrow$ colisión $\Rightarrow h_2(4800) = 0 \Rightarrow$ colisión $\Rightarrow$ se coloca en la siguiente posición libre (la posición 1)

(***) $h(2376) = 0 \Rightarrow$ colisión $\Rightarrow h_2(2376) = 0 \Rightarrow$ colisión $\Rightarrow$ se coloca en la siguiente posición libre (la posición 7)

Dispersión externa (en disco)

- El espacio de direcciones se compone de cubos (buckets). En Oracle son páginas
 - Un **cubo** puede ser 1 bloque de disco o un grupo de bloques contiguos
 - Cada cubo almacena varios registros
- La función hash mapea el valor del campo hash a un nº de cubo
- La cabecera del fichero incluye una tabla que convierte el nº de cubo en la dirección absoluta de bloque de disco



Dispersión externa (en disco)

- Las **colisiones** se pueden resolver utilizando una variación del encadenamiento:
 - Cada cubo tiene un puntero a una lista enlazada de registros de desbordamiento
 - Son punteros de registro <dirección de bloque, posición del registro dentro del bloque>

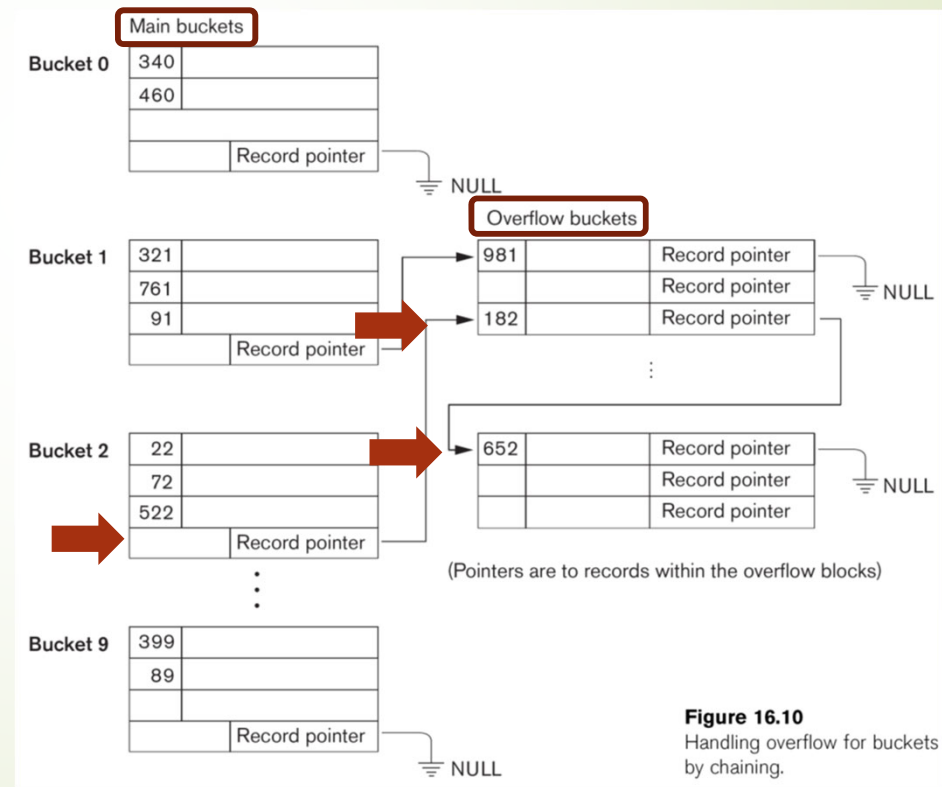


Figure 16.10
Handling overflow for buckets
by chaining.

Ejercicio: Un fichero con CODCLI como clave de direccionamiento calculado contiene registros con los valores K de CODCLI que se muestran en la siguiente tabla. El fichero utiliza 8 cubos, numerados del 0 a 7. Cada cubo es 1 bloque del disco y puede contener 2 registros. Cargar estos registros en el fichero en el orden dado, empleando la función de direccionamiento calculado $h(K) = K \bmod 8$. Para resolver las colisiones se utiliza la técnica de encadenamiento (dispersión estática) para ficheros de disco

Registro	K	$h(K) = K \bmod 8$
record1	6987	3
record2	9244	4
record3	3600	0
record4	1364	4
record5	826	2
record6	7110	6
record7	1461	5
record8	2376	0
record9	1522	2
record10	4541	5
record11	874	2
record12	1588	4
record13	4800	0

Cubo	Registros	Puntero registro
0	record3 (3600) record8 (2376)	
1		
2	record5 (826) record9 (1522)	NULL
3	record1 (6987)	
4	record2 (9244) record4 (1364)	NULL
5	record7 (1461) record10 (4541)	NULL
6	record6 (7110)	NULL
7		NULL

Bloques desbordamiento
record11(874) record12(1588)
record13(4800)

The diagram illustrates a pointer overflow. A thick black arrow originates from the 'Puntero registro' column of Cubo 0 and points to the first row of the 'Bloques desbordamiento' table. Another arrow points from the second row of Cubo 0 to the second row of the overflow blocks. Subsequent arrows show the sequence of pointers within the overflow blocks, eventually leading to NULL values.

Dispersión externa (en disco)

- ▶ Para reducir los registros de overflow, el fichero hash habitualmente se mantiene entre un 70-80% lleno
- ▶ La función hash debe redistribuir los registros uniformemente entre los cubos
 - ▶ En caso contrario, el tiempo de búsqueda se incrementa mucho debido a la cantidad de registros de overflow existente
- ▶ **Desventajas de la dispersión externa estática:**
 - ▶ Un n° fijo de cubos es problemático si el n° de registros del fichero crece o disminuye
 - ▶ La búsqueda de un registro por un campo distinto al campo hash es tan costosa como en caso de un fichero desordenado
 - ▶ El acceso ordenado por el campo hash es ineficiente (hay que ordenar los registros)