

5 faltas maximo	Avance de proyecto Antes de cada examen	Directo	Las reuniones de proyecto se realizan los días viernes
30% Proyecto		1er examen	26 Febrero Rev 21 Feb
50% Examen		2do examen	18 Abril Rev 13 Abril
20% Tareas y Trabajos		3er examen	28 - 30 Mayo Rev 23 Mayo
10%		Faltas	1 Junio

Proyecto

Requerimientos del avance de
Proyecto

→ Diccionario de Datos

	All	Mod	Elim	Consulta	
• Entidades	✓	✓	✓	✓	
• Atributos	✓	✓	✓	✓	

Proyecto en visual
no serializar

Alumno (Entidad)

- Clv Uniu
- nombre
- Fecha nace
- Clv Correm

(Atributos) → Metadatos

Bibliografía

Fundamentos de Base de Datos

Henry F. North

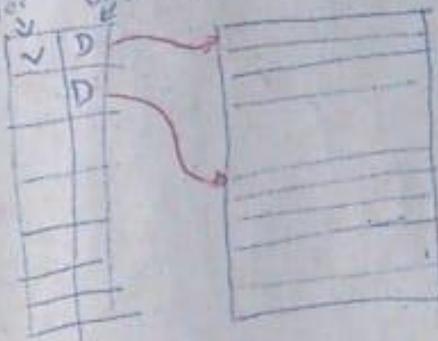
Abraham Silverchatz

Stu Ed.

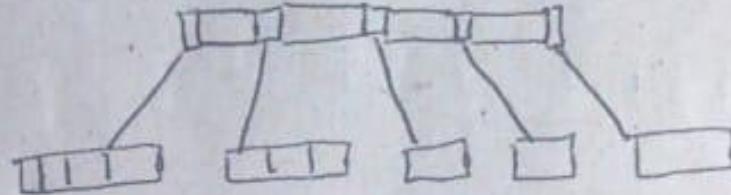
-Mc Grawhill

Indice Distintos tipos de índice

Júler Dirección

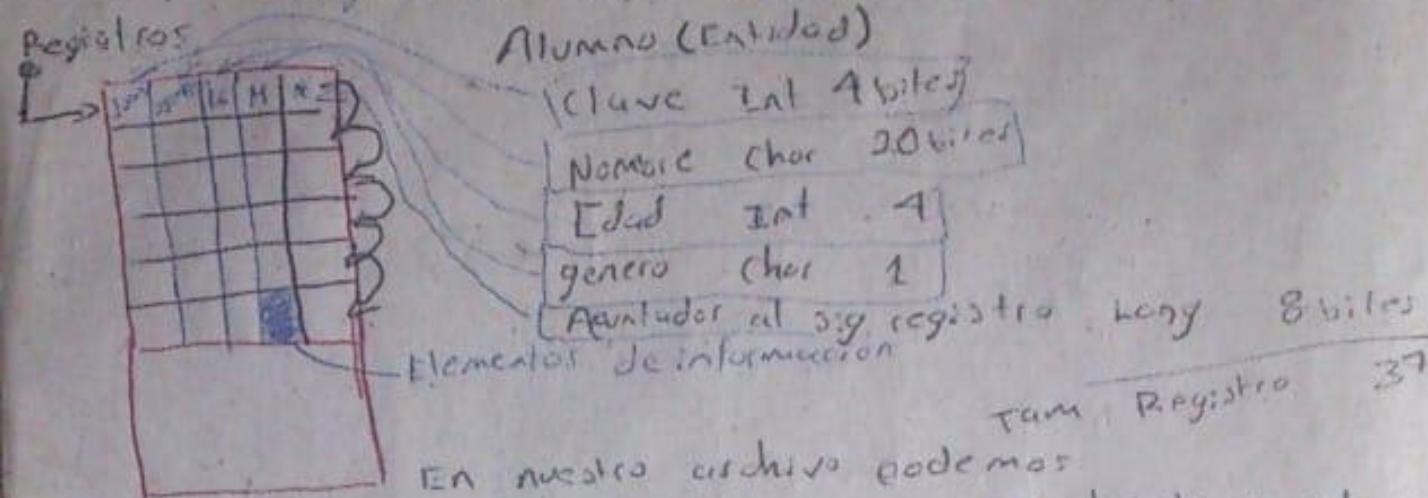


Articol BT

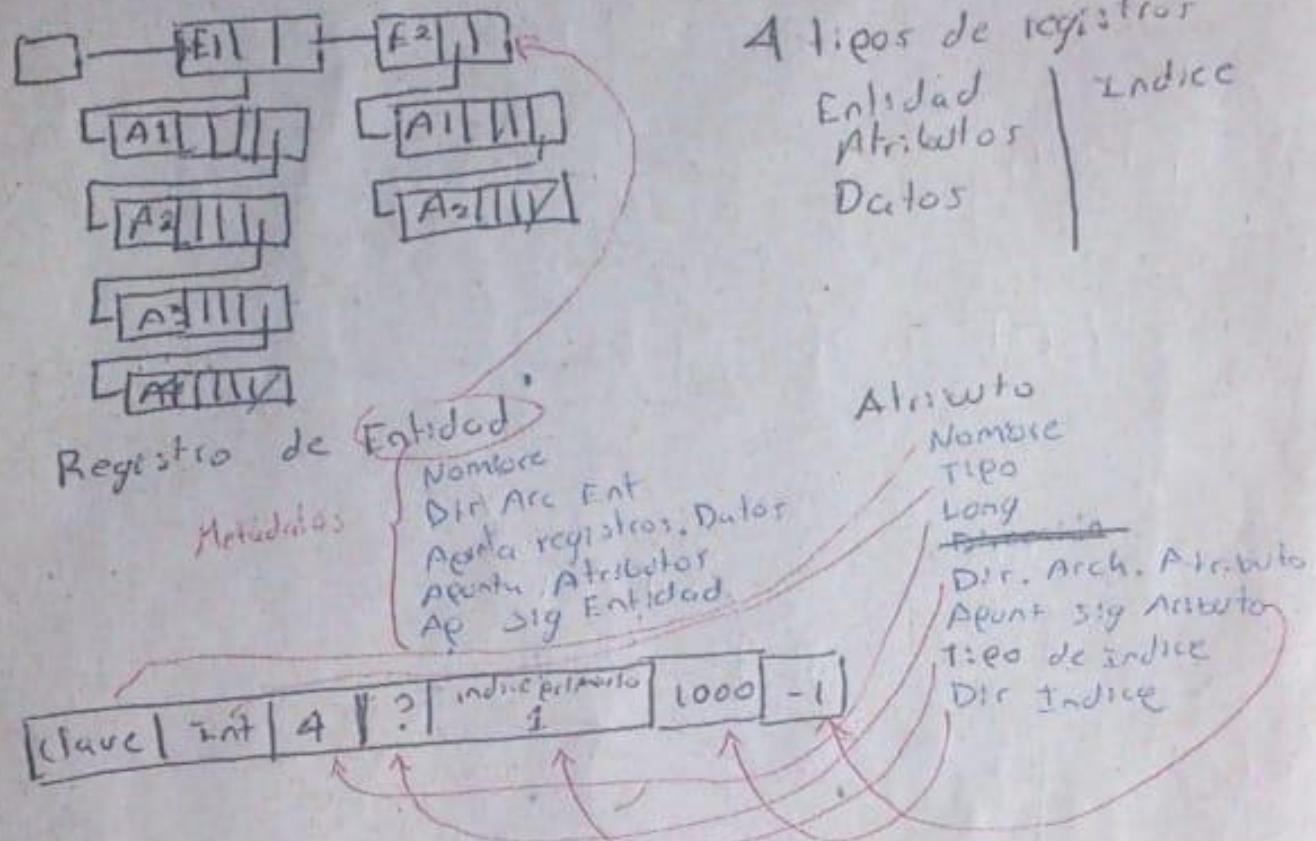


Descargo: temario de la materia y pegarlo en la libreta
Archivos: Es una colección de registros lógicamente relacionados

Registro: Es una estructura de campos o elementos de información lógicamente relacionados.



EN nuestro archivo podemos:
Almacenar Mas de 1 entidad y Mas de 1 atributo en cada entidad



Clasificación de Archivos

- Trabajo con los que trabajan algunos programas
- Texto Archivos del usuario
- Reporte se genera un reporte a partir de este
- Configuración del sistema operativo
- Transacción Se guardan todas las modificaciones que se realizan en una base de datos.
- Programa archivos que contienen código

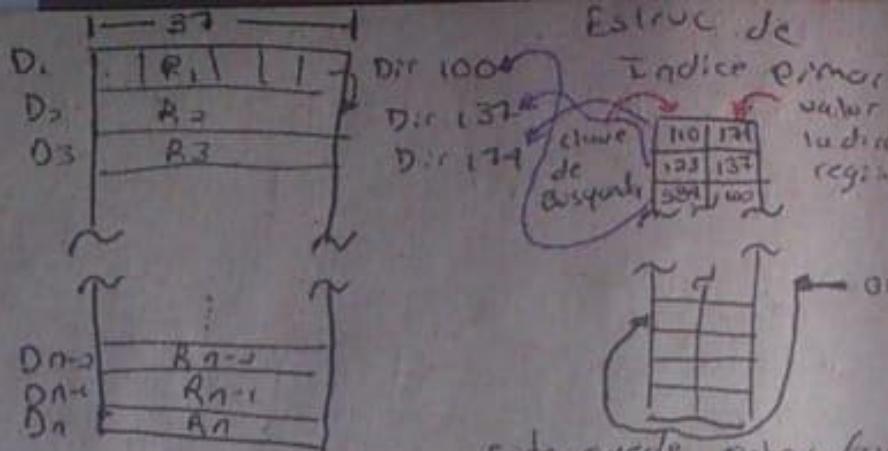
Operaciones de archivos Tarea para el módulo Miércoles 31

- Crear
- Abrir
- Desplazar
- Cerrar
- Leer
- Escribir

- Trabajo operaciones de archivos en C++ colocar un ejemplo, solicitar un nombre de archivo, abrirla, después determinar:
- Crear un nuevo archivo agregando un _copy.txt
 - Volcado de datos (copiar el contenido del primer archivo_copy.txt en otro)
 - Cerrar los archivos

Cloves

- Búsqueda de ~~o~~ ^oPK
 - primaria → no se repite
 - secundaria o externa



R. ~~class~~ SBA

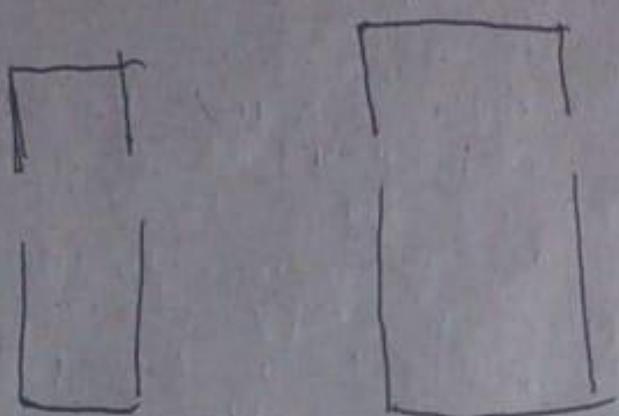
Ra 23

B3 110

Indice erratum

Oidenado y

sus valores deben estar ordenados



secundario

- No se registran en la estructura
 - Listado de apuntadores a otros registros
 - Ordenados puede
no ser

B 584 Juan 18 M

B2 123 June 16 M

R3 110 Maria LS F

s registros

Puede haber mas de un criterio de búsqueda

Juan	
Maria	

→ 100
134

→ 121

- Clasificación de archivos
 - Trabajo con los que tratan algunos programas
 - Texto Archivos del usuario
 - Puede generarse un reporte a partir de este
 - Configuración del sistema operativo
 - Transacción se guardan todas las modificaciones que se realizan en una base de datos
 - Programa archivos que contienen código
- Operaciones de archivos *Tutoría para el módulo Miércoles 3.1*
- Crear
 - Abrir
 - Desplazar
 - Cerrar
 - Leer
 - Escribir
- Tutoría operaciones de archivos en C++ colocar un ejemplo, solicitar un nombre de archivo, abrirlo, después obtener un nuevo archivo agregando un _copy.ext
- Volcado de datos (copiar el contenido del primer archivo_copy.ext en otro)
 - Cerrar los archivos

Los registros que se crean pueden estar de su vez almacenados siguiendo un orden, semejante a como los libros están ordenados en una biblioteca por algún atributo como el número decimal Dewey

- Un archivo puede tener varios índices según diferentes claves de búsqueda. Si el archivo que contiene los registros está ordenado secuencialmente, si el archivo que contiene los registros está ordenado secuencialmente,

Los índices con agrupación también se llaman índices primarios.

Los índices cuyas claves de búsqueda especifican un orden diferente del orden secuencial del archivo se llaman índices sin agrupación o secundarios

Estos archivos con un índice con agrupación según la clave de búsqueda sellan los archivos secundarios indexados

Índice Densol

Barcelona	C- 217	Barcelona	750	↓
Daimiel	101	Daimiel	500	↓
Madrid	110		600	↓
Pamplona	215	Madrid	700	↓
Reus	102	Pamplona	400	↓
Ronda	201		900	↓
	218		700	↓
	222	Reus	700	↓
	305	Ronda	350	↓

buscada en el archivo.

En un índice denso con agrupación el registro índice contiene el valor de la clave con un apuntador al primer registro con ese valor de la clave de búsqueda.

El resto de registros con el mismo valor de la clave de búsqueda se almacenarán consecutivamente después del primer registro, dado que, ya que,

una implementación de índices densos pueden almacenar una lista de apuntadores a todos los registros con el mismo valor de la clave de búsqueda; Esto no es necesario para los índices con agrupación.

Índice Denso - Inserción

Primeramente se realiza una búsqueda usando el valor de la clave de búsqueda del registro a insertar.

Si el valor de la clave de búsqueda no aparece el índice el sistema inserta en este un registro índice con el valor de la clave de búsqueda en la posición adecuada.

En caso contrario se ejecutan las sig acciones:

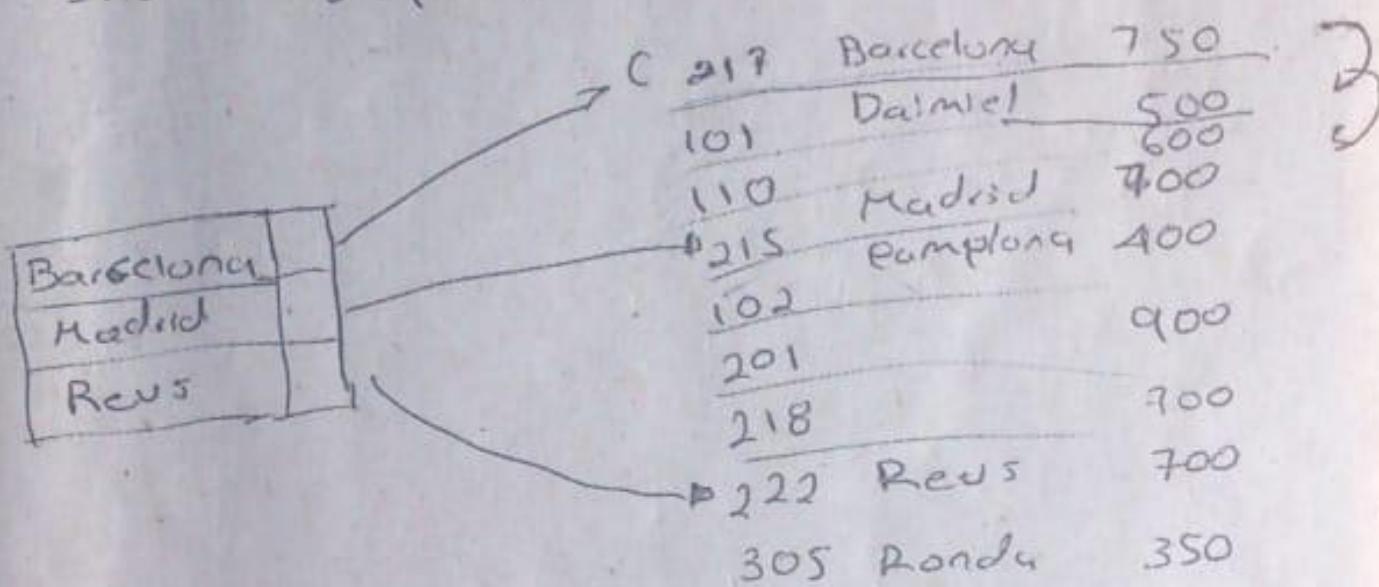
Si el registro índice almacenó apuntadores a todos los registros con el mismo valor de la clave de búsqueda el sistema añade un apuntador al nuevo registro en el índice.

En caso contrario, el registro índice almacena un apuntador solo hacia el primer registro con el valor de la clave de búsqueda. El sistema sitúa el registro insertado después de los otros con los mismos valores de la clave de búsqueda.

Índice Denso Eliminación

- Para borrar un registro, primero se busca el índice abierto.
 - Si el registro borrado era el único registro con ese valor de la clave de búsqueda, el sistema borra el registro índice correspondiente del índice.
- En caso contrario se ejecutan las sig acciones.
 - Si el registro índice almacena apuntadores a todos los registros con el mismo valor de la clave de búsqueda, el sistema borra del registro índice el apuntador al registro borrado.
- En caso contrario, el registro índice almacena un apuntador solo al primer registro con el valor de la clave de búsqueda, el sistema actualiza el registro índice para apuntar al sig registro.

Índice Disperso



Solo se guardan los primeros
algunos valores.

Al igual que en los índices densos, cada registro índice contiene un valor de la clave de búsqueda y un apuntador al primer registro con ese valor de la clave.

Para localizar un registro se trae la entrada del índice

se empieza por el registro apuntador por ese entrada del índice y se continua con los apuntadores del archivo hasta encontrar el registro deseado.

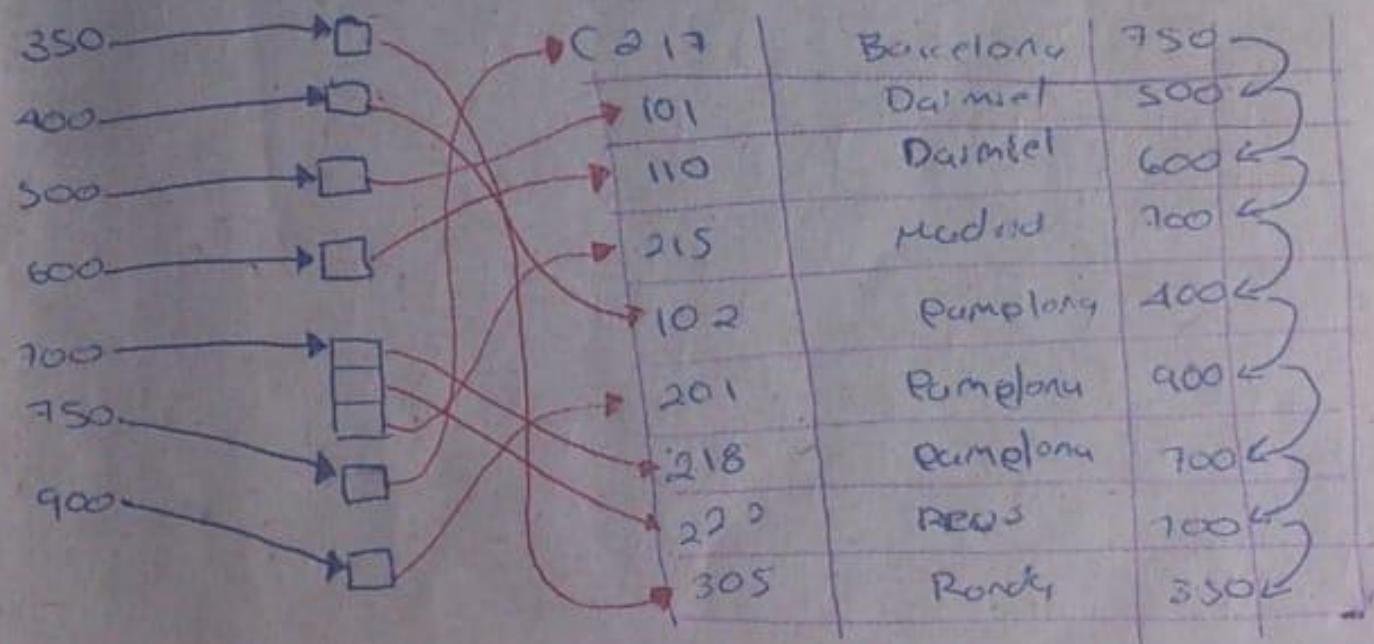
Índice Ímpreso Inserción

Se asume que el índice almacena una entrada por cada bloque, si el sistema crea un bloque nuevo, inserta el primer valor de la clave de búsqueda (en el orden de la clave de búsqueda) que aparece en el nuevo bloque del índice. Por otra parte, si el nuevo registro tiene el menor valor de la clave de búsqueda en su bloque, el sistema actualiza la entrada del índice que apunta al bloque; si no, el sistema no realiza ningún cambio sobre el índice.

Índice disperso - Eliminación

- Si el índice no contiene un índice con el valor de la clave de búsqueda del registro borrado, no hay que hacer nada.
- En caso contrario se emplean las siguientes acciones:
 - Si el registro de borrado era el único registro con la clave de búsqueda el sistema reemplaza el registro índice correspondiente con un registro índice para el sig valor de la clave de búsqueda (en el orden de la clave de búsqueda). Si el sig valor de la clave de búsqueda ya tiene una entrada en el índice, se borra en lugar de reemplazarla.
 - En caso contrario, si el registro índice para el valor de la clave de búsqueda apunta al registro a borrar, el sistema actualiza el registro índice para que apunte al siguiente registro con el mismo valor de la clave de búsqueda.

Índice secundario (Clave Foranea o extrana)



Los índices secundarios deben ser únicos y apuntar a un índice en el índice por cada valor de la clave de búsqueda, y un apuntador a cada registro del archivo.

Los apuntadores en estos índices secundarios no apuntan directamente al archivo en vez de eso, cada apuntador apunta a un cuadro que contiene apuntadores al archivo.

Los índices secundarios mejoran el rendimiento de las consultas, sin embargo, implican un tiempo adicional importante al modificar la base de datos. El diseñador de la base de datos debe ~~decidir~~ saber que índices secundarios son deseables, según una estimación sobre las ~~frecuentes~~ frecuencias de las consultas y de las modificaciones.

índice secundario se programará

Ejercicio

Se desea diseñar una base de datos que almacene información sobre los préstamos de las películas de un video club.

En la actualidad la gestión de la información se lleva de la siguiente manera:

Cuando se realiza un préstamo se llena un formulario en el que se anotan el socio que se lleva la película, fecha y el número del código que se llevará que es único (De cada película hay varios copias en enlos distintos). Este formulario se almacena en un archivero de películas prestadas. Cuando el socio devuelve la película, el formulario se traslada al archivero de películas devueltas. El video club tiene, además un archivero con fichas de las películas ordenadas por título; cada ficha tiene además el género de la película (comedia, terror, ...) el nombre del director de la película y los nombres de los protagonistas; también se tiene un archivero con las fichas de los socios, ordenadas por el código que el video club les da cuando les hace su carnet o credencial de afiliación. Cada ficha tiene el nombre del socio, su dirección y teléfono, los nombres de sus directores de películas favoritos, los nombres de sus actores favoritos y los géneros favoritos

Identificar Entidades

1 Prestamo

2 Pelicula

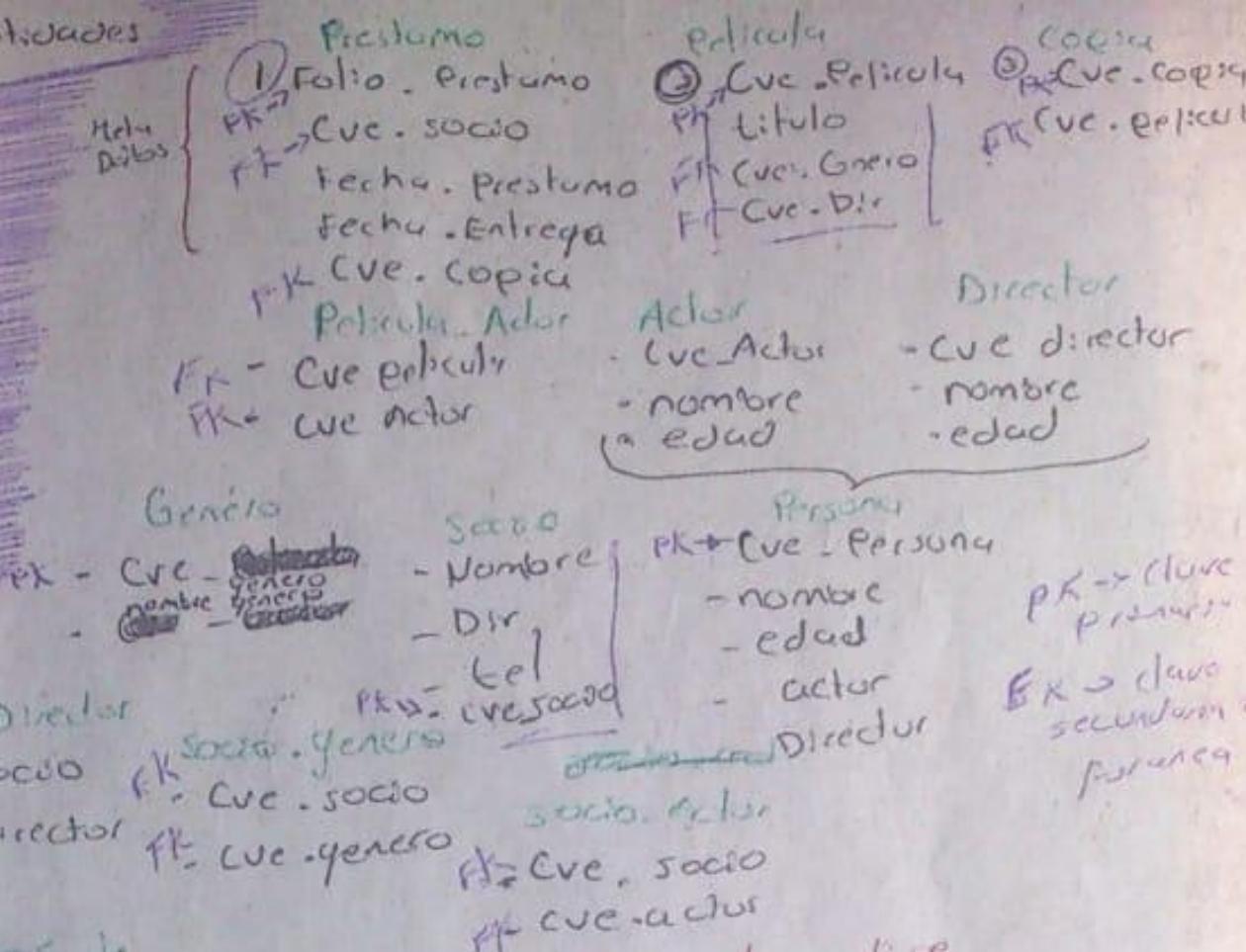
3 Copia

4 Genero

5 Director

6 Actor

7 Socio



Relaciones de

Socio
Cve_Socio
Nombre
Direccion
Telefono

Algo tamaño

	Int	1
Cve_Socio	Char	20
Nombre	Char	30
Direccion	Char	10
Telefono		

Tipo de indice

0 No indice

1 Cve_Busq

2 PK

3 FK

Cve_Socio
Nombre
Direccion
Telefono

Tipo indice Dir Ind sig atributo

-1 -1 -1

-1 -1 -1

-1 -1 -1

-1 -1 -1

Director → null

Diseño lógico de una base de datos

El diseño lógico de una base de datos es el proceso de trasladar, o ubicar, el diseño conceptual en un diseño lógico que se ajuste al modelo de BD elegido (relacional, orientado a objetos, de objetos, jerárquico)

A un especialista que desarrolla el diseño lógico de una base de datos

Diccionario de Datos.

- Donde se almacenan las descripciones de los datos y sea accesible por los usuarios.

Este catálogo es lo que se denomina Diccionario de datos y contiene información que describe los datos de la base de datos (metadatos).

Normalmente, un diccionario de datos describe entre otras cosas:

Nombre, tipo y tamaño de datos. (Entidades y Atributos).

Relaciones entre los datos.

Restricciones de integridad sobre datos.

Usuarios autorizados a acceder a los objetos de BD

Elementos de un diseño conceptual

Entidad

Atributo

Relación

Regla conceptual o de Negocios

Datos de Intersección

Entidad: Una entidad es una persona, lugar o cosa, suceso o concepto sobre el que se recopilan datos. En otras palabras, las entidades son los objetos reales que nos interesan lo suficiente como para capturar y guardar sus datos en una BD. Una entidad se representa como un rectángulo en el diagrama. Cualquier cosa se designe con un nombre puede ser una entidad.

Atributo:

- Es un hecho unitario que caracteriza o describe de alguna manera a una entidad.
- Se dice que son un hecho aislado porque deben ser indivisibles, lo que significa que no pueden dividirse en unidades más pequeñas que tengan algún significado. Por lo tanto, un atributo es la unidad de datos con nombre más pequeño que aparece en un sistema de base de datos.

Relaciones:

- Son asociaciones entre las entidades. Como las bases de datos se concentran en guardar datos relacionados, las relaciones se vuelven el elemento que mantiene unida la base de datos.

Regla empresarial o de negocios:

- Una regla de negocios es una política, procedimientos o norma adoptada por una organización. Las reglas de negocios son muy importantes en el diseño de una base de datos por que determina los controles que deben aplicarse a los datos.

Las reglas de negocios se implementan en las bases de datos como restricciones, que son las reglas formularias definidas que limitan, de alguna manera, los valores.

Nombre: Se define como el identificador propio del objeto que se quiere representar.

Dirección de la entidad: Dirección del archivo en donde se almacena la entidad

Dirección de los atributos: Dirección de archivo en donde se almacena la entidad

Dirección de los datos:

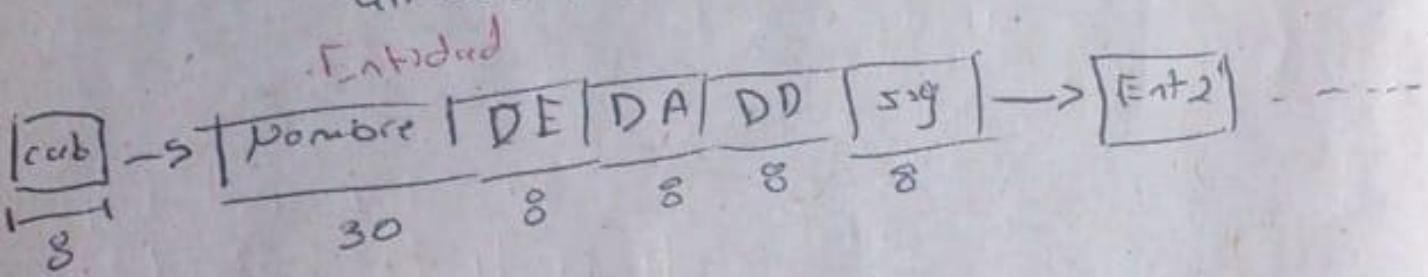
Dirección de la sig entidad

Métadatos de un atributo

Nombre ID del atributo

Tipo de dato: Para el estudio de la materia se usará el tipo entero (E) y carácter (C).

Longitud del tipo de dato: longitud en bytes del tipo de dato
Dirección del atributo: Dirección de archivo en donde se almacena el atributo.

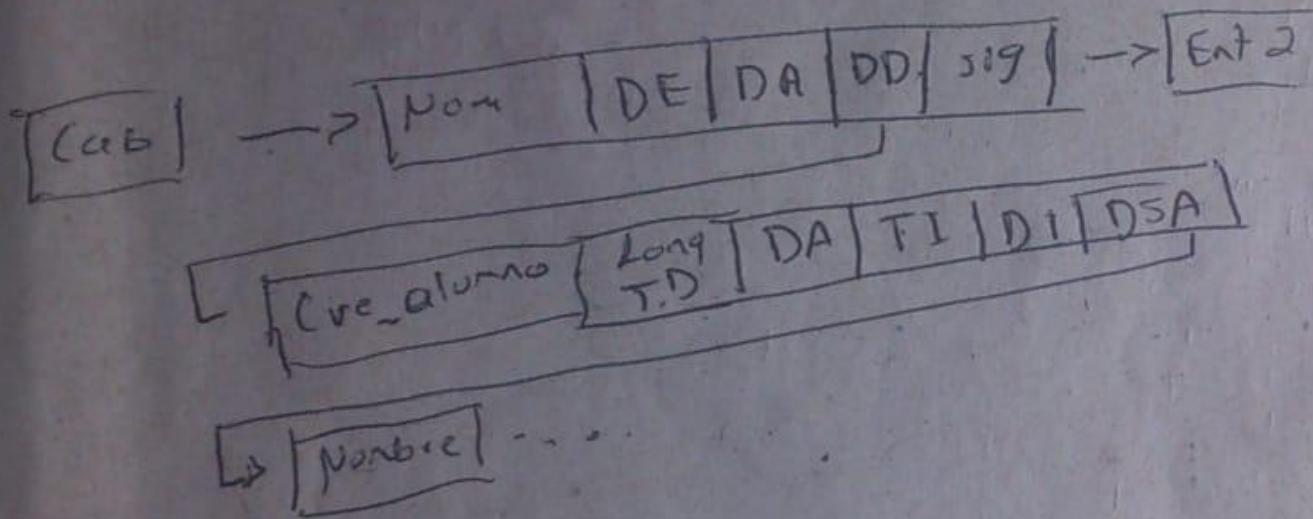


Atributos: Identifica el tipo de estructura de datos que se usa para representar el índice.

- 0 sin tipo de índice
- 1 clase de búsqueda
- 2 índice primario
- 3 índice sec

Dirección del índice: Dirección de archivo en donde se comienza a almacenar la estructura de datos utilizada para representar ~~el directorio~~
~~siguiente~~ el índice

Dir del sig atributo: dirección del sig atributo que define una misma entidad.



pidir nombre del archivo
generar el archivo bin bacio
cabecera = -1

Pelicula

PK Cve_pelicula

Título

FK Cve_Genero

FK Cve_Director

Entidad → ~~Actor_Accesorios~~

nombre = 30

dir_ent = 8

dir_atr = 8

dir_datos = 8

dir_sig_ent = 8

Atributo ~~Actor~~

nombre = 30

tipo = 1

longitud = 4

dir_atrib = 8

tipo indice = 4

dir_index = 8

dir_sig_atrib = 8

} 62 bytes

} 63 bytes

No debe repetirse el nombre de la entidad en un mismo archivo.

Las entidades deben ser insertadas de manera ordenada

No debe repetirse el nombre de un atributo dentro de la misma entidad. Los atributos no se ordenan se accomodan como van llegando

Entidad Pelicula

① Pelicula | -1 | -1 | -1 | -1 |

Siempre se inserta al final del archivo

② Insertar Pelicula | 70 | -1 | -1 | 8 |

↓ cubreca
-1
8 bytes

8 | Pelicula | 8 | -1 | -1 | -1 |

80 | Pelicula | 8 | -1 | -1 | -1 |

copia | 70 | -1 | -1 | 8 |

Alta Atributos

(-10) 0

70
132

0	Pelicula	80	-1	-1
10	Copia	10	-1	8
132	Copiar	132	21	-1

Entidad Pelicula
Copia
Pelicula

Atributos

Nombre T.Ind Longitud

Tipo Indice

1	E
2	C
3	
4	

[Alta] [Modifica] [Elimina]

Nombre	Tipo Indice	Longitud	Tipo Ind	DA	DI	DSA
ex Relacion	1	2	132	132	-1	-1

(Titulo) C 80 195 0 -1 -1

[ex. Pelicula, Tit 19 132 21 -1 195]

La secretaría de una Universidad necesita que se de
las sig. entidades@ materiales (clave, nombre, créditos),
horarios de materias (folio del horario, clave materia, grupo de
este, salón, clave del profesor)

④ Estudiantes (clave, nombre, carrera)

⑤ Profesores (Clave, nombre, Clave Dto, título)

Además la inspección de los estudiantes en asignaturas
y las notas obtenidas de los estudiantes en cada
asignatura en la que están inscritos.

⑥ Materia

Clave	E	A
nombre	C	30
créditos	L	A

⑦ Horarios

Folio	E	A
Clave Mat	E	A
profesor	C	30
grupo	E	A
semestre	C	10
salón	C	10

1

⑧ Estudiantes

Clve	
nombre	
carrera	

⑨ profesor

Clve	E	A
nombre	C	30
Clve Dto	E	A
título	C	20

3

⑩ Horario-Alumno

Clve Estudiante	
Título	

Cve

Organización de archivos secuenciales

Los archivos secuenciales están diseñados para el procesamiento eficiente de los registros de acuerdo a un orden fijo o alguna clave de búsqueda. La clave de búsqueda es cualquier atributo o conjunto de atributos que tiene por que ser la clave primaria. Para permitir la recuperación rápida de los registros según el orden de la clave de búsqueda, estos se vinculan mediante apuntadores. El apuntador de cada registro señala el siguiente registro según el orden indicado por la clave de búsqueda. Además para minimizar el número de accesos a los bloques en el procesamiento de los archivos secuenciales, los registros se guardan en el orden indicado por la clave de búsqueda.

La organización secuencial de archivos permite que los registros se lean de forma ordenada, lo que puede resultar útil para la visualización, así como para ciertos algoritmos de procesamiento de consultas.

sin embargo resulta difícil mantener el orden físico secuencial a medida que se insertan y borran registros, dado que resulta costoso desplazar muchos registros como consecuencia de una sola operación de inserción o borrado. Se puede gestionar el borrado mediante cadena de apuntadores. Para la inserción se aplican las sig reglas

① Localizar el registro que precede al registro que se va a insertar según el orden de la clave de búsqueda.

② Si existe algún registro vacío (es decir esperando que haya quedado libre después de una operación de borrado) dentro del mismo bloque que este registro, el registro nuevo se insertará ahí, en caso contrario, el nuevo registro se insertará en un bloque de desbordamiento. En cualquier caso, hay que ajustar los apuntadores para vincular los registros según el orden de la clave de búsqueda.

dr	C-215	Beccar II.	750	↑	Cuentas
	C-101	Centro	500	↓	
	C-110	Centro	600	↓	no cuenta sucursal saldo
	C-305	Collado Mediano	100	↓	
	C-107	Galapagar	400	↓	
	C-222	Hortaliza car	900	↑	
	C-102	Navacerrada	100	↓	
	C-201	Navacerrada	700	↓	
	C-218	Navacerrada	350	-1	

Nuevo reg → C-888 Leyanes 1800

Alumno

Cve_Unique	E	4	0
Nombre	C	40	1
Cve_Carrera	E	4	0
fecha_nace	C	12	0
genero	C	12	0

Datos

Cve_Unique	Nombre	Cve_Carrera	Fecha_nace	genero
413	Juan	74	16/01/1993	Masculino
721	Jorge	75	01/03/1994	Masculino
983	Susana	63	13/10/1991	Femenino
293	Monica	68	18/02/1993	Femenino
330	Antonio	72	01/10/1990	Masculino
108	Gabriela	63	28/08/1992	Femenina
536	Agustin	74	15/01/1993	Masculino
295	Ruben	81	01/07/1991	Masculino
621	Jose	74	10/08/1992	Masculino

3	Alumno	8	70	-1	-1
70	Cve-Nombre	E	A	(0 -1)	133
3	nombre	C	40	(133)	1 -1 196
9	Cve-Apellido	E	A	196	0 -1 259
29	Fecha-Nace	C	12	259	0 -1 320
322	genero	C	12	320	0 -1 -1
585		385	413	Juan	74 13/01/1993 Masculino -1

Interface para agregar un registro

Cve-Nombre E A

nombre C 40

Cve-Apellido E A

Fecha-Nace C 12

genero C 12

cuando se quiere insertar un registro se tiene que validar la casilla que sea diferente de -1 por que sino, quiere decir que no hay atributos.

cuando se da insertar se verifican el signo de los datos.

473	473	721	Jorge	75	01/Mar/1994	Masculino	385

8	Alumno	8	-1	-1	-1	70
70	Profesor	70	-1	-1	-1	
18	Materia	18	-1	-1	-1	

Alumno
Profesor
Materia

AAA545 238A

- Manual de Usuario
- Documentación del proyecto
- Que contienen el directorio
 - Clases
 - Metodos
 - Funciones

Requerimientos para avance proyecto

- Archivo
 - Nuevo
 - Abrir
- Entidades
 - Crear
 - Modificar
 - Eliminar
 - Consultar
- Attributos
 - Crear
 - Modificar
 - Eliminar
 - Consultar

Entidad: Materia

Cve_materia E A Ø

materia C 30 1

creditos E 4 Ø

creditos

Cve Materia	Semestre	Int	II	creditos
1002	seminario	Int	II	1
0000	Geometria y trigonometria			3
1101	Humanidades	A		8
0041	Algebra	A		5
1120	Geometria Descritiva			3
2310	Estructuras de archivos			

1137	Materia	197	1234	1982	-1	
	Cve Materia	E	4	1234	0	-1

Cve_Usuario E A P
 Nombre C 40 1
 Cve_Carrera E A P
 Fecha_Nacimiento C 12 P
 Genero C 12 P

B) Alumnos 81709131117

70	Cve_Usuario	E	A	10	0	-1	133
123	Nombre	C	40	133	1	-1	P
126	Cve_Carrera	E	A	196	0	-1	254
251	Fecha_Nacimiento	C	12	254	0	-1	322
322	Genero	C	12	322	0	-1	1

485	385	413	NaN	71	18/01/1993	Masculino	51
473	473	721	Jorge	75	01/03/1944	Masculino	589
561	361	1983	Susana	63	13/10/1991	Femenino	-1
649	649	243	Mónica	68	18/02/1993	Femenino	561
737	737	350	Antonio	77	10/10/1990	Masculino	875
825	825	100	Gabriel	63	28/05/1992	Femenino	433
913	913	536	Aylin	71	15/09/1993	Masculino	737
1001	1001	195	Naomi	81	01/07/1998	Masculino	501
1059	1059	1621	José	74	10/06/1993	Masculino	357
1177	Matricula 1177		1531	1428	-1		
1239	Cve_Mater	C	1	39	0	-1	1302
1302	Materia	C	30	39	1	-1	1365
1365	Creditos	E	4	365	0	-1	-1

1428	1428	1002	Seminario Integrador	1	-1
1485	1485	0006	Geometria y Trigonometria	0	1536
1536	1536	1101	Humanidades A	3	1844
1540	1540	0011	Algebra A	8	1590
1644	1644	1120	Geometria Descriptiva	5	1425
1698	1698	2310	Estructuras de Archivos	8	1485

1752

```

if (indEntidad == -1)
{
  Entidad = -1;
  NuevoRegistro();
  NuevoReg();
  cout << "nuevo" << endl;
}
else
{
  NuevoReg() == 519;
  cout << "Nuevo" << endl;
}
  
```

}

Se requiere diseñar una base de datos que contenga información referente a zoológicos existentes en el mundo, así como las especies animales que estos albergan. De cada zoológico se conoce el nombre, ciudad, y país donde se encuentra, tamaño (en metros cuadrados) y presupuesto anual. De cada especie animal se almacena el nombre, familia a la que pertenece, y si se encuentra en peligro de extinción.

Además, se debe guardar información sobre cada animal que los zoológicos poseen como num.º, especie, sexo, año de nacimiento, país de origen y continente.

Zoológico

- Cve_Zoo E 4 P
- nombre C 30 I
- Ciudad C 30 P
- País C 20 P
- tamaño E 1 A
- Presupuesto E 4 P

Especie

- Cve_Espacio E 1 P
- Nombre C 30 I
- Familia C 30 P
- Extinción E 1 P

Animal

- Cve_Animal E 4 A
- Cve_Zoo E 4 P
- Cve_Especie E 4 P
- Sexo C 10 P
- nacimiento E 4 P
- País C 20 P
- Continente C 20 P

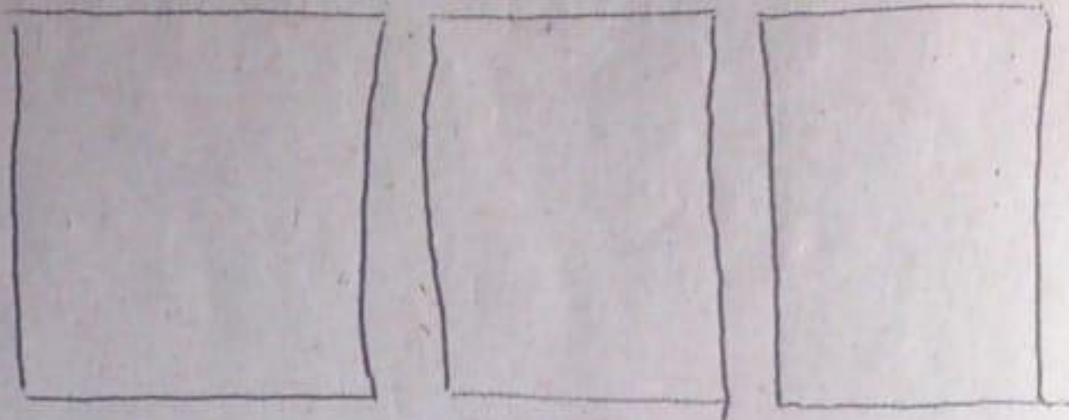
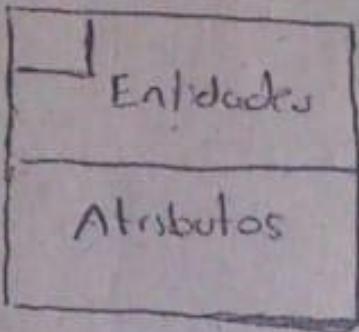
	1	-1	-1	-1	
130) Zoológico	8	1	-1	-1	
Especie	70	-1	-1	8	
animal	132	82	1265	20	
Cve_Zoo	E	4	199	0	-1 257
Extinción	C	30	257	1	-1 320
	C	30	320	0	-1 383
	C	20	383	0	-1 446
	E	4	446	0	-1 509
	E	4	509	0	-1 572
	E	4	572	0	-1 635
	C	30	635	1	-1 698
	C	30	698	0	-1 761

Una vez insertado algún registro ya no se puede cambiar o modificar Entidades y registros.

Alumno

Profesor

Materia



- Registros
↳ Clave Busqueda

Organización secuencial indexada

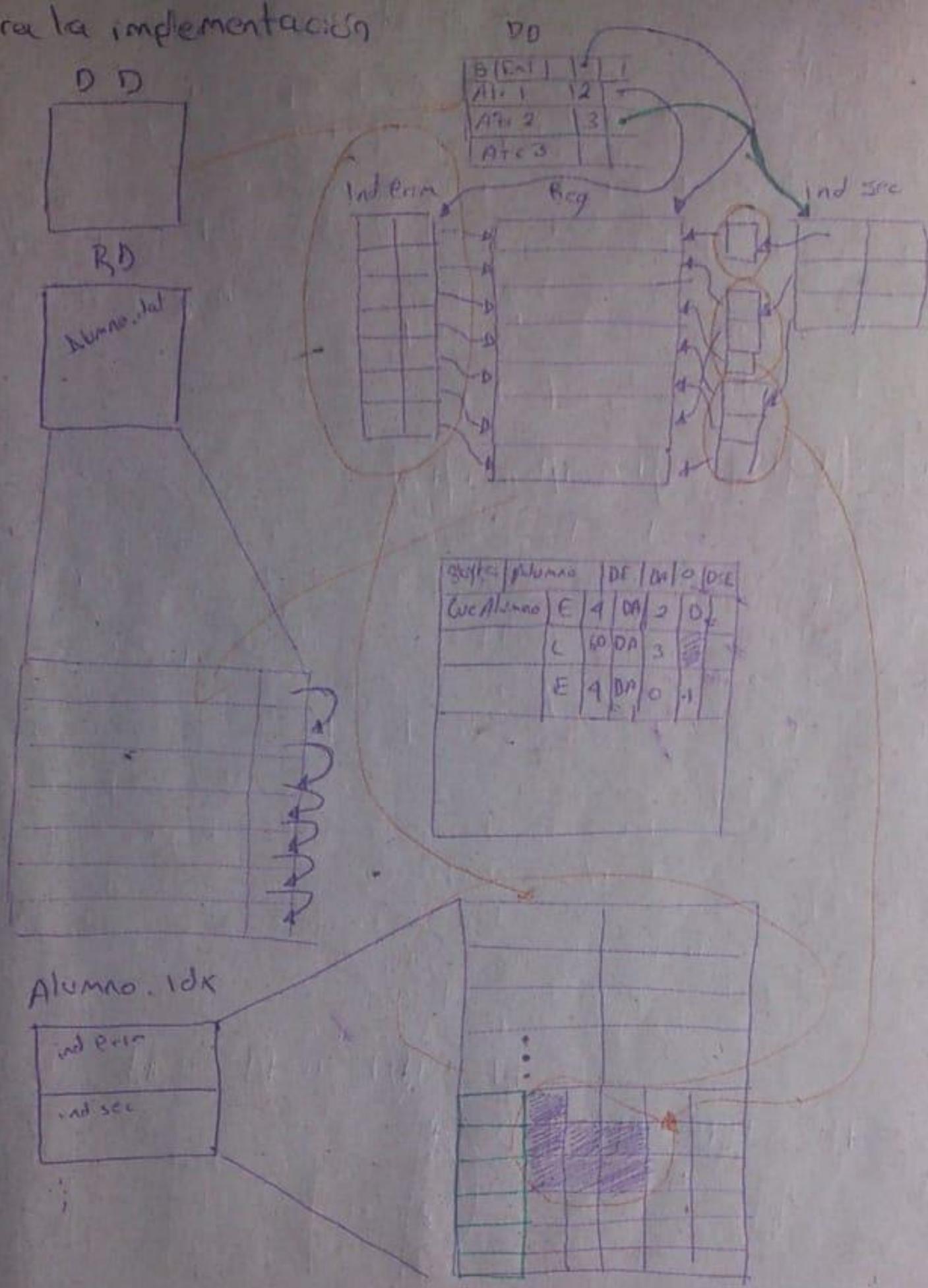
Se utiliza este tipo de organización de archivos cuando existe la necesidad tanto de acceder a los registros secuencialmente, como en una varía serie de búsqueda (tares primaria o secundaria), como de accederlos individualmente. Un archivo secuencial indexado proporciona la combinación de tipos de acceso que manejan un archivo secuencial y un archivo relativo o de acceso directo.

Estructura lógica de un archivo secuencial indexado

En este tipo de organizaciones de archivos se dispone de una tabla en la que aparecen ordenados los valores de la clave de búsqueda y asociado a cada uno de ellos la dirección del registro correspondiente.

C-217	Barcelona	150	↑
C-101	Damniel	500	↑
C-110	Jaime	600	↑
C-215	Manuel	700	↑
C-102	Pamelona	900	↑
C-201	Pamelona	900	↑
C-216	Ramplona	700	↑
C-112	Ramplona	700	↑

Parcial implementacion



Archivos de Índice de Árbol B+

La estructura de índice de Árbol B+ es la más extendida de las estructuras de índices que mantienen su eficiencia a pesar de la inserción y borrado de datos.

Un índice de Árbol B+ toma la forma de un árbol equilibrado donde los caminos de la raíz a cada hoja del grupo son de la misma longitud.

Cada nodo que no sea hoja tiene entre $n/2$ y n hijos donde n es fijo para cada árbol.

Estructura de un árbol B+

P ₁	K ₁	P ₂	K ₂	...	P _{n-1}	K _{n-1}	P _n
----------------	----------------	----------------	----------------	-----	------------------	------------------	----------------

Nodo de un árbol B+

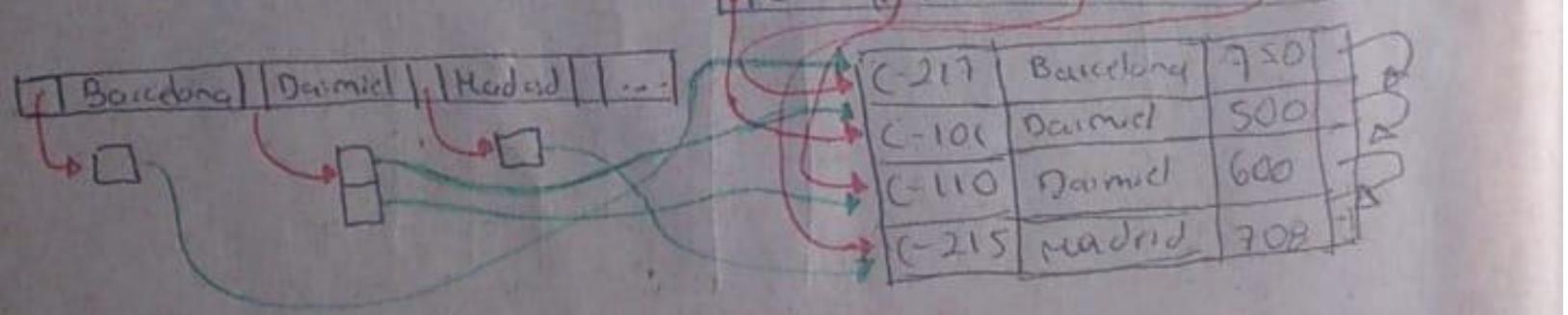
En la figura se muestra un nodo típico de un árbol B+.

Puede contener hasta $n-1$ claves de búsqueda (K_1, K_2, \dots, K_{n-1}) y n apuntadores (P_1, P_2, \dots, P_n). Los valores de las claves de búsqueda se mantienen ordenadas; así, si $i < j$, entonces $K_i < K_j$. Consideremos primero la estructura de nodos hoja. Para $j = 1, 2, \dots, n-1$, el apuntador P_j apunta, bien a un registro del archivo con valor de la clave de búsqueda K_j o bien a una lista de apuntadores, cada uno de los cuales apunta a un registro del archivo con valor de la clave de búsqueda K_j .

C-101	C-110	C-215	C217	...
-------	-------	-------	------	-----

Barcelona	Barcelona	Madrid	...
-----------	-----------	--------	-----

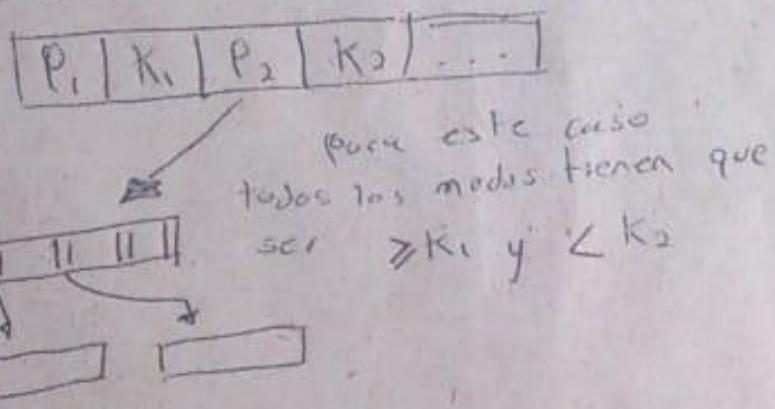
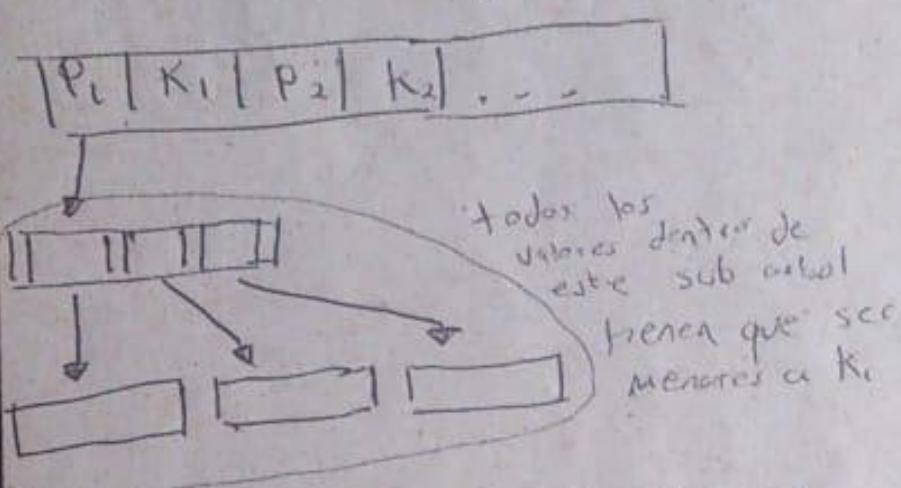
C-217	Barcelona	750
C-101	Barcelona	500
C-110	Barcelona	600
C-215	Madrid	700



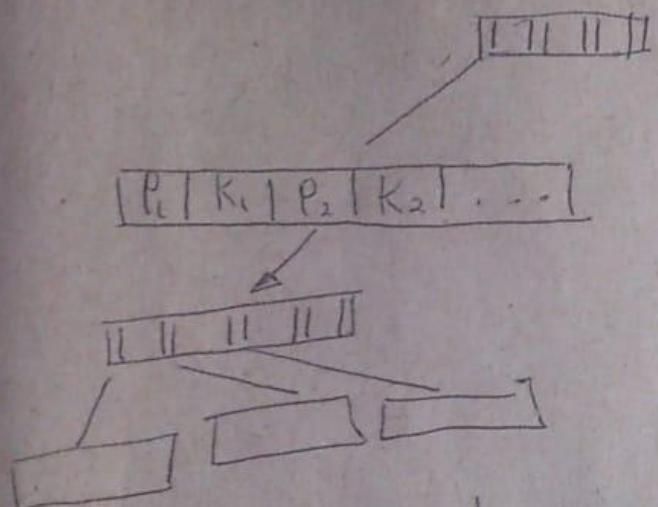
Los nodos internos del Árbol B+ forman un índice multivariado (diseñado) sobre los nodos hoja. La estructura de los nodos internos es la misma que la de los nodos hoja, excepto que todos los apuntadores son apuntadores a nodos del árbol. Un nodo interno podría guardar hasta n apuntadores y debe guardar al menos $\frac{n}{2}$ apuntadores. El número de apuntadores en el nodo se llama grado de salida del nodo.

Considera un nodo que contiene m apuntadores para $i = 2, 3, \dots, m-1$. El apuntador p_i apunta al subárbol que contiene los valores de las claves de búsqueda menores a k_i y mayor o igual que k_{i-1} .

El apuntador m apunta a la parte del subárbol que contiene los valores de la clave mayores o iguales que k_{m-1} y el apuntador p_1 apunta a la parte del subárbol que contiene los valores de las claves menores que k_1 .



El nodo interno siempre depende de otros

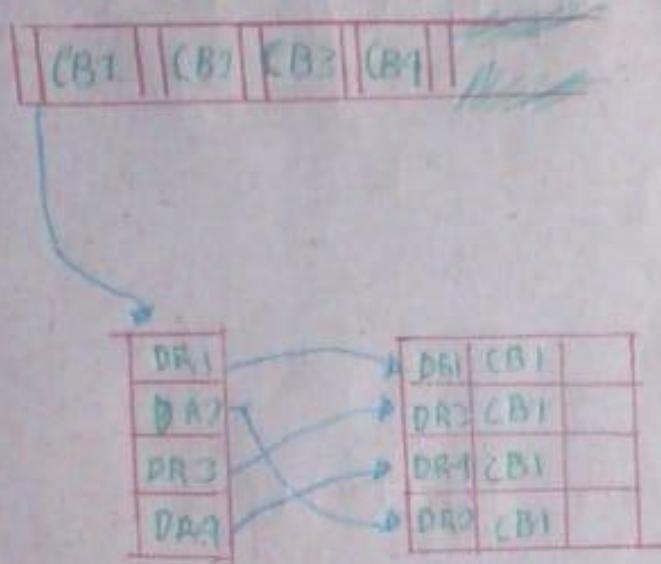


A diferencia de otros nodos internos el nodo raíz puede tener menos de $n/2$; sin embargo, debe tener al menos 2 apuntaduras, salvo que el árbol tenga un solo nodo. El nodo hoja y el nodo raíz son los únicos que pueden tener menos de la mitad de sus valores ocupados.

Añadido y Borrado - B+Tree

El borrado y la inserción son operaciones más complejas que las búsquedas, ya que puede ser necesario dividir un nodo como resultado de una inserción, o fusionar 2 nodos si uno se volviera demasiado pequeño. Cuando se divide o fusiona un par de nodos, se debe asegurar el equilibrio del árbol.

Inserción: Se busca hoja donde tendrá que ir el valor de la clave de búsqueda. Si el valor de la clave de búsqueda ya aparece en el nodo hoja, se inserta un nuevo registro en el orden y, si es necesario un apuntador en la lista, en caso de ser nodo secundario.



Borrado: Se busca el registro a borrar y se elimina del orden. Si no existe una lista asociada con el valor de la clave de búsqueda o si la lista se queda vacía como resultado del borrado, se borra el valor de la clave de búsqueda del nodo hoja.

988	910
688	899
140	715
974	711
172	674
489	708
858	915
580	611
195	779
646	889

Tipo de hoja

Dirección de
la hoja

1	8	8	4	8	4	8	9	8	4	8	= 652 y
		A ₁	K ₁	A ₂	K ₂	A ₃	K ₃	A ₄	K ₄	A _p sig	

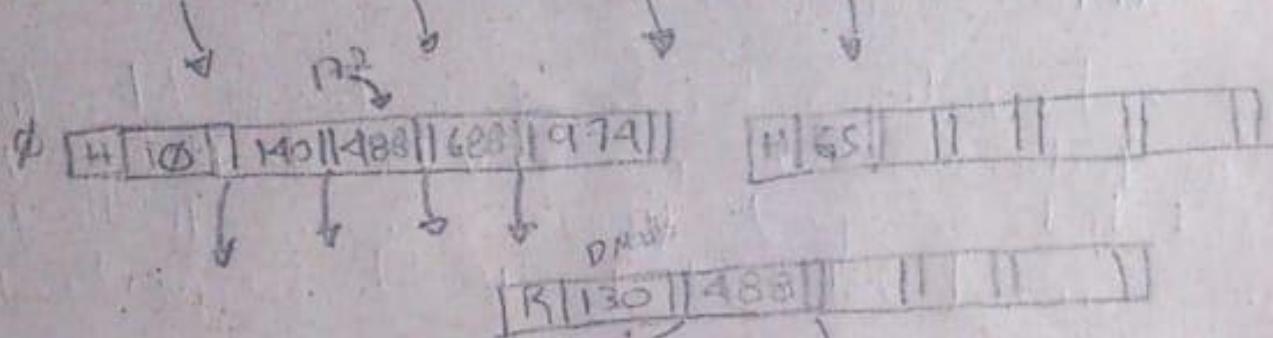
El contador que va en el atributo del índice debe apuntar al primer nodo del arbol

Tipo de índice = 4

Índice Primario

Φ	[H]	P _R	1488	DR ₂	688					
---	-----	----------------	------	-----------------	-----	--	--	--	--	--

Φ	[H]	DR ₃	140	BR ₁	1488	DR ₂	688	DR ₄	977	
---	-----	-----------------	-----	-----------------	------	-----------------	-----	-----------------	-----	--



[H]	Φ	140	142	1491
-----	---	-----	-----	------

[H]	65	488	688	858	971
-----	----	-----	-----	-----	-----

130 [R|DN|180|688|859|]

H | 0 | 140 | 172 | 184 |

H | 1 | 65 | 988 | 646 |

H | 2 | 85 | 688 | 715 | 145 |

H | 250 | 183 | 910 | 944 |

② 910

Φ 325 65 195 260
R | 130 | 408 | 188 | 1688 | 188 |

Ki Ki1

H | 0 | 140 | 172 | 491 | 42A |

H | 1 | 65 | 988 | 646 |

H | 2 | 85 | 688 | 715 | 259 | 944 |

Dirección

847

Realizamos la búsqueda

Si ya existe, then no se inserta

Ki1 >= DAT

Primer se guarda el Registro en su poder obtener
la dirección del registro y después se guarda
el índice para guardar el par TRReg | CuBsq

H | 325 | 1408 | 491 | 42A |

11304081188161

7 455.1 858 110

100

(vendo se pone una

[H]585||411||646||624||

1426018581 8141

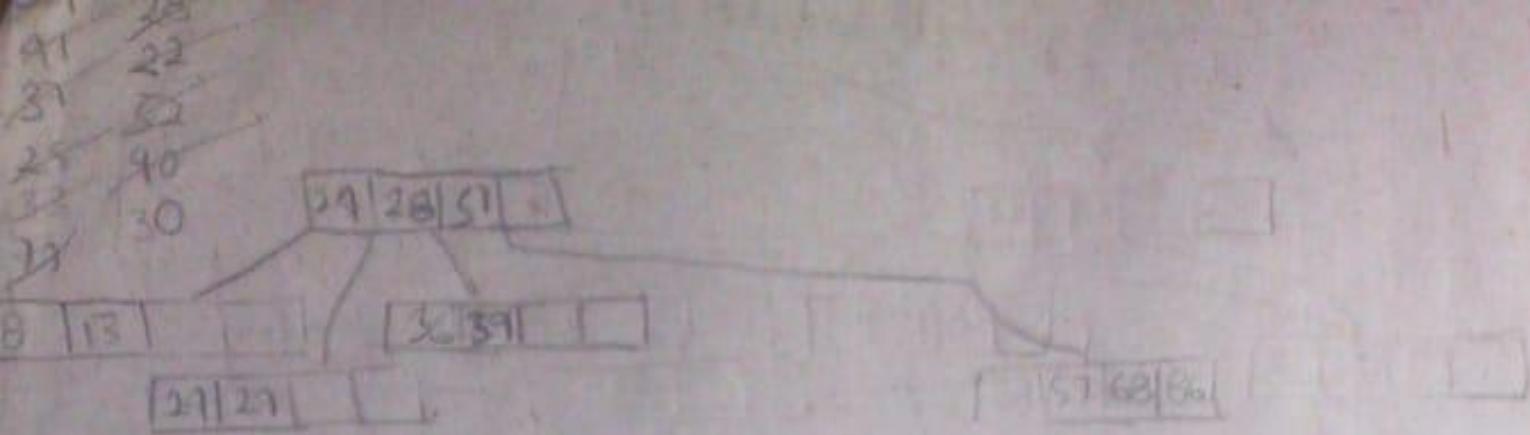
11/2/2014

[H] 340 | 910 | 911

ROT } primary 2 / red original
Y } orange. some also yellow
UHRS 2 / red red

Torre Lender

† Grec.



Organización de archivos multillave

Muchas aplicaciones que existen, exigen que los registros de datos tengan mas de una clave de búsqueda. Los registros de clave multilevel, aparecen, en algunos sistemas de operaciones, pero se les encuentra, mas comúnmente en los sistemas de información.

La funcionalidad de los archivos que se acceden utilizando varias claves, se vuelve mas importante cuando el numero de atributos es grande.

Existen dos métodos para proporcionar trayectorias de acceso adicional hacia los archivos de los registros de datos, los métodos son

- Organización multillave
- La Inversión

Organización multillave

Los archivos multillave mantienen un índice para cada clave secundaria. Existe una entrada en el índice de clave secundaria para cada valor que la clave secundaria tiene en este momento en el archivo de datos. La organización de archivos multilista difiere de la inversión en que mientras que la entrada en el índice de inversión, para un valor de clave tiene un apuntador a cada registro de datos con ese valor de clave, tiene solo un apuntador al primer registro de datos con ese valor de clave. Este registro de datos contiene un apuntador al siguiente registro de datos con ese valor de clave y así sucesivamente. Los encadenamientos en multillave

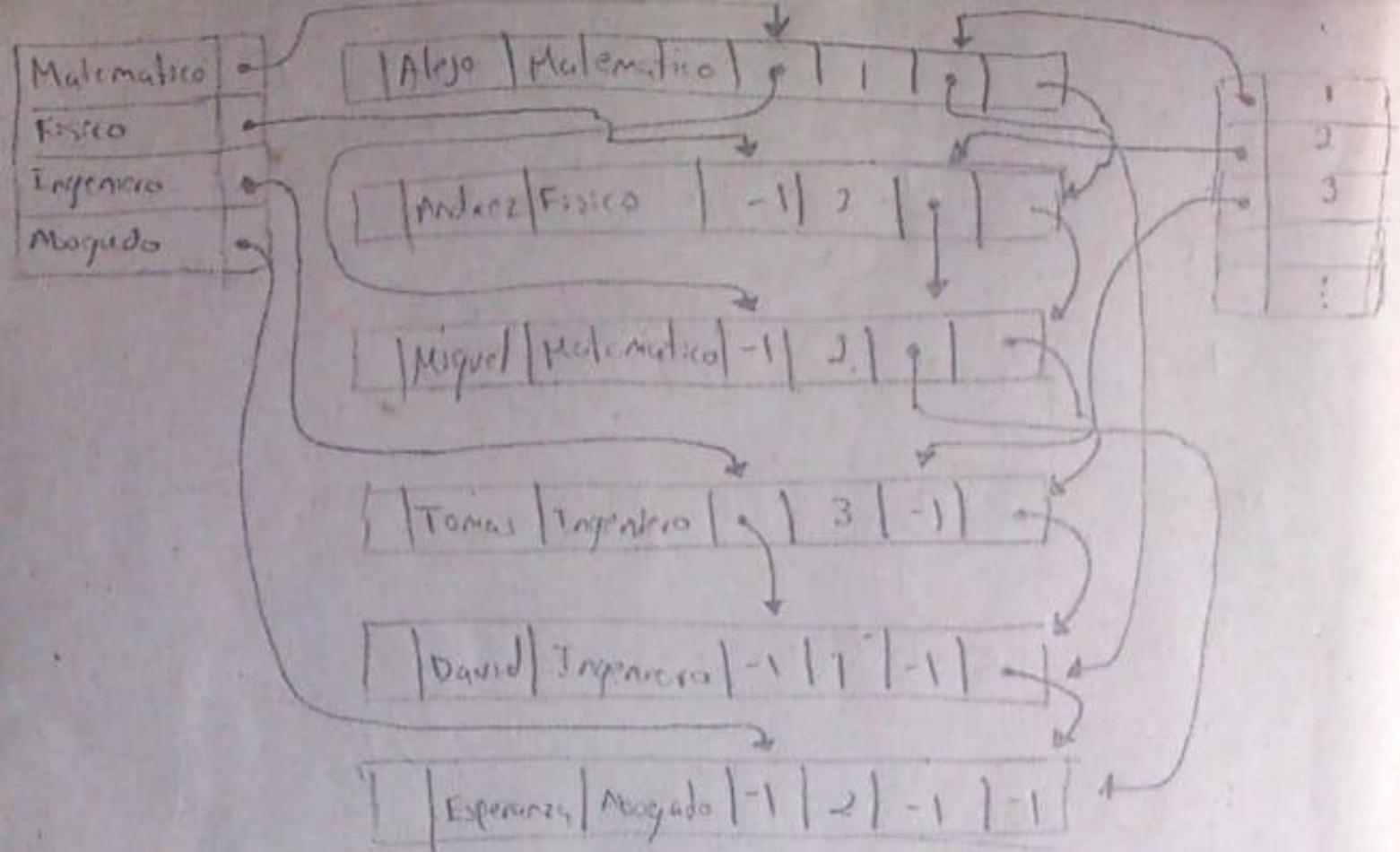
esta forma de organizados de un archivo permite accesar la información de manera ordenada a través de campos llaves. los multilluces permiten llegar a un registro por diferentes caminos, al igual que lo determina el atributo sobre el cual se hace la búsqueda.

Supongase que se tiene un archivo en el cual cada registro almacenará la sig información:

Nombre	Profesión	Categoría
--------	-----------	-----------

En un archivo se van a almacenar los sig registros:

Nombre	Profesión	Categoría
Alejo	Matemático	1
Andrés	Físico	2
Miguel	Matemático	2
François	Ingeniero	3
Diego	Ingeniero	1
Espenzer	Abogado	2



2 semanas y 2 días 16 Mayo

En el caso de la figura, la información puede accederse por medio de su profesión y por medio de su categoría que son los atributos que permiten realizar búsqueda directa en el archivo, como puede observarse. Se tiene una lista por profesión y una por categoría.

La organización de archivos multilista como puede apreciarse, se implementan unas listas ligadas entre los campos clave que se hayan definido y encadenando a todos los registros con un cierto valor.

Todos los registros están ligados entre si por medio de apuntadores que hacen virtualmente varias listas ligadas en un solo archivo. Los registros están almacenados de forma aleatoria y dispersos en una unidad de almacenamiento ya que el sistema de archivos almacena registros en cualquier espacio libre, su secuencia lógica se mantiene por medio de apuntadores.

Mantenimiento a un archivo organizado en multilistas.
 La inserción: esta acción consiste en adicionar un nuevo registro modificando los índices y los apuntadores que sean necesarios cuando se va a insertar un registro en una lista ligada. Puede insertarse en cualquier espacio libre del archivo, sin embargo, los apuntadores de los nodos (registros) afectados, deben ajustarse para que reflejen la secuencia lógica deseada.

- ⑥ Modificación de datos: Esta acción consiste en buscar un registro específico y cambiar el contenido de uno o varios campos del registro. En el caso de la modificación se hace el recorrido por los índices, hasta llegar al registro que deseas modificar. Una vez que se encuentra el registro se cambia el contenido de todos los campos.
- ⑦ Eliminación de registros: Esta acción consiste en eliminar un registro en específico, eliminandolo tanto del archivo de datos como del archivo de los índices.

Cve Unca	Nombre	
413	Juan	79
721	Jorge	75
983	Susana	63
293	Monica	74
332	Juan	72
108	Gabriela	75
536	Jorge	79
295	Patricia	63
133	Roxorio	63

.dat	
0	413
60	721
120	983
180	293

.idx	

CV_número(4) Nombre(20) Genero(1) Carrera(20)

* 10 Manuel H

* 22 Lucas M Informática

* 29 Jorge M Recursos

17 Lois M Geología

* 25 María M Construcción

21 Lucía M Recursos

15 Juan M Acuñación

* 31 Jorge M Construcción

* 13 David M Informática

* 51 Katherin M Geología

* 20 Carlos M Recursos

* 29 Marcos M Informática

* 48 Juan M Informática

* 19 David M Informática

* 60 Carlos M Construcción

① Redescribir un Arbol BT de orden 2 (ACB)
con el atributo CV_ALUMNO

② Insertar un índice hash estípico utilizando la
función de hash: $f(h) = \text{código ASCII del primer}$
caracter (CV_ALUMNO) \rightarrow (modo), donde cada celda tendrá
3 ccs de búsqueda (≈ 4)

③ Insertar los registros de datos tomados en cuenta que
los atributos nombre y carrera son índices multivistas.
Generar los índices correspondientes

④ Realizar la eliminación del Árbol BT (Ejercicio D) de la CB *

La otra forma de estimación posible asigna todos los valores de la clase de búsqueda al mismo rango, una función con el mismo efecto que la que asigna todos los rangos en el mismo rango. Durante una búsqueda se pide que estimen todos los rangos dentro del rango de búsqueda, una función de estimación ideal distribuye los errores uniformemente entre todos los rangos de modo que cada uno de ellos tiene el mismo número de errores. Se puede elegir una función que asigne los errores de la clase de búsqueda a los rangos de manera que la distribución tenga las mismas condiciones.

Distribución uniforme: La función de estimación asigna a cada rango el mismo número de valores de la clase de búsqueda dentro del conjunto de todos los valores posibles de la misma.

Funciones de estimación

- Distribución aleatoria: En el caso promedio, cada rango tiene asignado, con el mismo número valores, independientemente de la distribución real de los valores de la clase de búsqueda.

Asociación Estática (Hash Estática)

La organización de archivos basada en hashing permite evitar el acceso a la estructura del índice. La asociación también proporciona una forma de controlar índices. En esta descripción de la asociación, se usan el término cajón para indicar una unidad de almacenamiento que puede guardar uno o más registros. Formalmente sea A el conjunto de todos los valores de clave de búsqueda y sea B el conjunto de todas las direcciones de cajón.

Una función de asociación h es una función de A a B . Para insertar un registro con clave de búsqueda con el valor de k_1 de la clave de búsqueda, se calcula $h(k_1)$, lo que proporciona la dirección del cajón para el registro. De momento se supone que hay espacio para almacenar el registro en el cajón.

Para realizar la búsqueda con el valor k_1 de la clave de búsqueda, basta con calcular $h(k_1)$ y luego buscar el cajón con esa dirección, suponiendo que dos claves de búsqueda, k_1 y k_2 , tienen el mismo valor de asociación, es decir, $h(k_1) = h(k_2)$ el cajón contendrá registrar con el valor de clave de búsqueda k_1 y k_2 .

El borrado es igual de sencillo si el valor de la clave de búsqueda que se quiere borrar es k_1 , se calcula $h(k_1)$, después se busca el cajón con esa dirección y se elimina el registro.

0	→ DLL	0	c-217	Balcerow	750	10	
1	→ DLL	1	c-101	Daniel	500	80	
2	→ DLL	2	c-110	Daniel	600	120	
3	→ DLL	3	c-215	Kudry	700	160	
4	→ DLL	4	c-102	Ramplow	400	200	
5	→ DLL	5	c-218	Ramplow	900	140	
6	→ DLL	6	c-222	Rewz	700	320	
7	→ DLL	7	c-305	Rondz	320	-1	
8	→ DLL	8					
9	→ DLL	9					
10	→ DLL	10					
11	→ DLL	11					
12	→ DLL	12					
13	→ DLL	13					

rajan

array[2] rk =
Af 2:9(cj) = -1

rajan

rajan

array[2]

rk =

Af 2:9(cj) = -1

168

168

168

168

168