SESIÓN 1: PLANIFICACIÓN DEL ALMACENAMIENTO E INDEXACIÓN

Ejercicio 1 (Importante: Tamaño de archivos, índice secuencial, índice multinivel, índice árbol B+, índice árbol B)

Se dispone de un disco con bloques de B = 512 bytes. Un puntero a un bloque tiene P = 6 bytes de longitud y un puntero a un registro tiene P_r = 7 bytes de longitud. Un fichero tiene 30.000 registros de longitud fija de una tabla EMPLEADO. Cada registro contiene los siguientes campos:

- Nombre : 30 bytes
- NSS: 9 bytes
- Código departamento: 9 bytes
- Dirección: 40 bytes
- Teléfono: 9 bytes
- Fecha de nacimiento: 8 bytes
- Sexo: 1 byte
- Código puesto: 4 bytes
- Salario: 4 bytes
- Se utiliza 1 byte adicional como marca de eliminación de registro.

Se pide:

1. Calcular el número de bloques del archivo.
2. Suponer que el fichero está ordenado según el campo clave NSS y que se desea construir un índice secuencial primario sobre NSS. Calcular:
   (a) Factor de bloques del índice (Número de registros/bloque).
   (b) Número de entradas y de bloques del índice.
   (c) Si se convierte en un índice multinivel, el número de niveles hasta obtener un árbol.
   (d) Número total de bloques del índice multinivel
   (e) Número de accesos al bloque si se utiliza el índice primario o el índice multinivel.
3. Suponer que el fichero no está ordenado según el campo clave NSS y se desea construir un índice secuencial secundario sobre dicho campo. Repetir la sección b) y comparar los resultados.
4. Suponer que el fichero no está ordenado según el campo clave NSS y que se desea construir una estructura de acceso de árbol B^+ sobre NSS. Calcular:
   (a) Los órdenes n de los nodos intermedio y nodos hoja del árbol B^+.
   (b) Número de bloques en el nivel de hoja requeridos si los bloques están ocupados aproximadamente al 69% de su capacidad.
   (c) Número de niveles requeridos si los nodos internos están ocupados también al 69%.
   (d) Número total de bloques que ocupa el árbol.
   (e) Número de accesos a bloque para buscar y recuperar un registro del fichero.
5. Repetir la sección anterior para el caso de un árbol B.

Ejercicio 2

Construir un árbol B^+ con el siguiente conjunto de valores de la clave:

(2, 3, 5, 7, 11, 17, 19, 23, 29, 31)

Asumir que el árbol está inicialmente vacío y que se añaden los valores en orden ascendente. Construir árboles para el caso de que el número de punteros que cabe en un nodo es:

(a) cuatro
(b) seis
(c) ocho

Ejercicio 3

Para cada árbol del ejercicio anterior, mostrar el aspecto del árbol después de cada una de las siguientes operaciones:

(a) Insertar 9
(b) Insertar 10
(c) Insertar 8
(d) Borrar 23
(e) Borrar 19

Ejercicio 4

Repetir el ejercicio 2 para un árbol B.

Ejercicio 5

Suponer que se utiliza una asociación dinámica en un archivo que contiene registros con los siguientes valores de la clave de búsqueda:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31

Mostrar la estructura asociativa dinámica para este archivo si la función de asociación es $h(x) = x \bmod 8$ y los cajones pueden contener hasta tres registros.

Ejercicio 6

Mostrar como cambia la estructura del ejercicio 5 como resultado de realizar los siguientes pasos:

(a) Borrar 12
(b) Borrar 31
(c) Insertar 1
(d) Insertar 15

Ejercicio 7 (Importante: Índice Mapa de bits)

Considerar la relación cuenta que se muestra a continuación:

C-217	Barcelona	750
C-101	Daimiel	500

C-110	Daimiel	600
C-215	Madrid	700
C-102	Pamplona	400
C-201	Pamplona	900
C-218	Pamplona	700
C-222	Reus	700
C-305	Ronda	350

1. Construir un índice de mapa de bits sobre los atributos nombre-sucursal y saldo, dividiendo saldo en cuatro rangos: menores que 250, entre 250 y menor que 500, entre 500 y menor que 750, y 750 o mayor.
2. Considerar una consulta que solicite todas las cuentas de Daimiel con un saldo entre 500 ó más. Describir los pasos para responder a la consulta y mostrar los mapas de bits finales e intermedios construidos para responder la consulta.
3. Si el bloque ocupa 512 bytes, ¿cuántos bloques ocupa el índice?
4. Para la consulta de apartado2, ¿cuál es el coste de procesar la consulta?

Ejercicio 8 (Importante: Índice árbol B+, índice asociativo secundario campo no clave, índice secuencial sobre campo no clave)

Considerar un archivo de datos que mantiene información sobre estudiantes. Los registros de este archivo tienen los siguientes campos:

Campo	**Longitud (bytes)**	**Observaciones**
Carnet	20	
Nombre	40	
CodCarrera	2	16 carreras diferentes Distribución Uniforme
Edad	16	
IndiceAcademico	32	Valores: 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0 Distribución Uniforme

El archivo de datos tiene 100.000 registros y se desea construir tres índices sobre este archivo:

- El primer índice está definido sobre el atributo Carnet y se implementará usando un árbol B^+.
- El segundo índice se define sobre el atributo CodCarrera y será implementado utilizando un hash estático con la función **CodCarrera mod 8**.
- El tercer índice estará definido sobre el atributo IndiceAcadémico y será un índice secuencial secundario.

Se disponen de bloques de 512 bytes y los punteros a bloques ocupan 6 bytes mientras que los punteros a registros 7 bytes. Además para el archivo de datos cada

bloque se llena al 65% y para los índices, los nodos y los cajones se ocupan al 69%. Se pide:

(a) ¿Cuánto espacio se requiere para almacenar el archivo y sus índices?
(b) Si se desea insertar un registro nuevo de un estudiante. ¿Cuál es el coste de realizar dicha operación?
(c) Si se desea buscar un estudiante por su carnet, ¿Cuánto cuesta esta operación?
(d) ¿Cuál es el coste de listar a todos los estudiantes que cursan una carrera dada?. Asumir el peor caso.
(e) ¿Cuál es el coste de listar a todos los estudiantes que tienen un índice académico mayor que 3.0?. Asumir el peor caso.
(f) ¿Cómo se podrían mejorar los procesos de lectura anteriores?

Ejercicio 9 (Importante: Organización asociativa de fichero de datos, índice secuencial primario, índice árbol B+ primario)

Se dispone de 1 fichero de 1 millón de registros, donde cada registro ocupa 200 bytes de longitud de los cuales 10 bytes corresponden al campo clave. Un bloque de disco ocupa 1000 bytes de longitud y un puntero a bloque/registro es de 5 bytes. Se pide:

(a) Usando una organización asociativa del fichero con 1000 cajones, calcular el tamaño del cajón en bloques asumiendo que todos los bloques contienen el mismo número medio de registros y el tamaño total en bloques del fichero. ¿ Cuál es el número medio de accesos necesarios para localizar un registro?
(b) Usando un índice secuencial ordenado primario denso de 1 nivel para el atributo clave sobre el fichero y asumiendo que todos los bloques están tan llenos como sea posible, ¿cuántos bloques se necesitan para el índice?. Si se emplea una búsqueda binaria sobre el índice, ¿cuantos accesos son requeridos en media para encontrar un registro?
(c) Si se usa ahora un árbol B^+ sobre el archivo ordenado y asumiendo que todos los bloques están tan llenos como sea posible, ¿Cuántos bloques se necesitan para el índice?. ¿Cuál es la altura del árbol y la principal característica de éste árbol?

Ejercicio 10

Rellene la siguiente tabla donde se muestran las *potenciales* combinaciones de campos (clave o no) con la existencia de índices. Para cada celda de la tabla indique:

1. ¿Hasta cuántos campos pueden existir? ¿Qué restricciones?
2. ¿Cómo se realizarán las búsquedas en los registros?
3. *[Solo para índices]* Clasifique el **tipo**: Denso/Disperso y Clasifique la **estructura**: Indexado/Arbol B+/Hash/...

En el caso de que una celda **no tenga sentido** marque **N/A** ó **X**.

(Por ejemplo, suponga que tiene un archivo de registros con 10 atributos, de los cuales 5 son campos clave y 5 no lo son)

Campo / *Índice*	Clave Primaria	Clave Secundaria	No clave
No existe índice	① ②	① ②	① ②

Existe índice	① ② ③	① ② ③	① ② ③

Ejercicio 11

Independientemente de lo que haya respondido en ejercicio anterior, se tiene una tabla de una base de datos con A atributos y N registros. Cada uno de los campos tiene una longitud de L_A bytes, totalizando una longitud de registro de tamaño fijo de L_R bytes. El tamaño del bloque de disco es de B bytes. Se sabe que existen algunos campos que son clave, y que se querría realizar consultas por ciertos campos (p.e. la mitad), sean del tipo que sean.

En esta situación, un cierto "Consultor de Bases de Datos" recomienda que uno de los campos clave sea clave primaria, y los campos clave restantes sean claves secundarias. Además el consultor, argumentando razones de gran **eficiencia de espacio en disco** y **eficiencia en búsquedas**, recomienda **únicamente** crear índices en las claves secundarias, y no crear índices para el resto de campos. Suponga que los índices creados son secuenciales indexados.

Haga una critica (constructiva) de estos argumentos. Para ello:

(a) *Eficiencia de espacio*: Calcule el tamaño del fichero de registros. Calcule el tamaño del fichero de índices (si existe) para cada tipo de campo: Clave primaria, clave secundaria y no clave. Compárelos: Exprese el tamaño en función del tamaño del fichero de registros. Si lo necesita, suponga el tamaño de los punteros mucho menor que el tamaño del campo.
(b) *Eficiencia en búsquedas*: Calcule el coste de efectuar una búsqueda sin y con el índice (si lo hay) para cada tipo de campo. Compárelos.
(c) ¿Qué recomendación final haría? ¿Qué modificaría, eliminaría o matizaría?

Ejercicio 12 (Importante: Organización de archivo de datos mediante árbol B+)

Un sistema emplea una organización de la información de imágenes mediante árboles B^+ y páginas de datos de 4 Kbytes. La información asociada a cada imagen que se quiere organizar ocupa 128 bytes, de los cuales 12 bytes de los anteriores son para la clave de búsqueda. En el servidor se indexa realmente esta información; y la imagen misma se almacena aparte en páginas especiales de 4 MBytes por lo que la información que se organiza ya incluye un enlace a su correspondiente imagen. Sabiendo que en este árbol cada puntero ocupa 4 bytes y que la ocupación media de cada bloque del árbol es del 80%, determinar:

(a) Determinar:

- Factor de bloque a aplicar a las **páginas de datos** a organizar, considerando que la información de control de cada página de datos ocupa 196 bytes.
- Orden del árbol B^+, sabiendo que cada **página de índices** guarda una información de control de 12 bytes
- Niveles necesarios del árbol para organizar 100.000 imágenes.
- Número máximo de nodos de cada nivel.
- Cantidad de datos que como máximo organiza cada nivel.

- Número máximo de páginas de datos.
- Número máximo de registros que se podrán almacenar.
- Espacio en Kbytes que ocupa cada nivel como máximo.

(b) El sistema destina como máximo 824 Kbytes para albergar el árbol en memoria principal. Con una ocupación media del 80%, ¿Cuántos accesos serán necesarios para localizar y leer una determinada imagen y su respectiva información?

(c) Y si se quisiera obtener todas las imágenes y su respectiva información de forma ordenada según la clave ¿Cuántos accesos a disco se realizarían?

Ejercicio 13

Considere un árbol B+ que indexa 300 registros.

(a) Si este árbol B+ fuera de orden 9 (es decir, cada nodo tiene como mucho 9 claves), *justifique* cual sería la altura (profundidad) mínima y máxima del árbol (un árbol con un nodo –el raíz- tiene una profundidad de 1).

(b) Si este árbol B+ tuviera una altura (profundidad) de 2, *justifique* cual sería el orden mínimo y máximo.

Ejercicio 14

Considere un índice en retícula (*grid*) sobre un atributo A. Se ha hecho una partición del atributo en 5 rangos, por ejemplo precios en Euros, en los rangos de [0,100), [100,200), [200,300), [300,400), [400,500). Se han indexados 15.000 registros, y cada bloque del índice en retícula puede almacenar claves y punteros hasta 3.300 registros (los registros en sí, estarán almacenados en algún otro lugar). Si se tuvieran más registros de un rango, entonces se utilizarían bloques de desbordamiento (*overflow*) de la misma capacidad. Suponga que no existen duplicados del atributo A en los datos.

(a) Suponga que los valores de A están uniformemente distribuidos, esto es, un registro cualquiera puede estar en cualquiera de los cinco rangos con igual probabilidad, y una clave de búsqueda puede estar en cualquier rango con igual probabilidad. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(b) Ahora suponga que existe un cierto sesgo en los datos. En particular, la probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el rango j es j/15 (tenga en cuenta que la suma de estas probabilidades desde j=1 hasta 5 vale 1). ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(c) Ahora considere un caso de sesgo extremo. La probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el rango 1 es de 1 y, por lo tanto, la probabilidad es 0 para los otros rangos. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro, dado un valor de A que no está en el índice? *Justifíquelo*.

(d) Si se usa un árbol B+ para la misma aplicación, ¿Cuál es el número de operaciones de I/O esperado (nuevamente para un valor que no está en el índice)? Suponga que los nodos de un árbol B+ contienen 3.300 punteros. *Justifique* brevemente porqué.

(e) *Justifique* qué índice, el de retícula o un árbol B+, es mejor para consultas de rangos.

(f) *Justifique* qué índice, el de retícula o un asociativo (*hash*), es mejor para consultas de rangos.

Ejercicio 15

Se tiene un disco que posee las siguientes características conocidas:

- Cada pista está formada por 100 sectores
- Cada sector contiene un espacio no útil del 50%, y los datos útiles ocupan 1.024 bytes
- Un bloque está formado por 2 sectores
- El tiempo de transferencia de 1 bloque es de 0.12 mseg, y el tiempo (medio) de búsqueda es de 10 mseg

1. Calcular la velocidad de rotación del disco
2. Calcular el tiempo (medio) que se tardaría en leer un bloque cualquiera0
3. Calcular la tasa de transferencia de pico y la tasa de transferencia sostenida
4. Si en estas condiciones se modifica la geometría del disco y ahora un bloque estuviera formado por 5 sectores ¿Cuál sería el nuevo tiempo de transferencia de 1 bloque?

Ejercicio 16 (Importante: índice asociativo)

Considere un índice asociativo sobre un atributo A, y tome como función de hash `h = ⌊ A / 100 ⌋`. Los valores que puede tomar A (precios en €) están en el rango [0, 499.99]. Se han indexados 16.000 registros, y cada bloque de cada cajón puede almacenar claves y punteros para 4.000 registros. Si se tuvieran más registros de un rango, entonces se utilizarían bloques de desbordamiento (*overflow*) de la misma capacidad. Suponga que no existen duplicados del atributo A en los datos.

1. Calcule el número N de cajones distintos necesarios. *Justifíquelo*.
2. Suponga que los valores de A están *uniformemente distribuidos*, esto es, un registro cualquiera puede estar en cualquiera de los N cajones con igual probabilidad, y una clave de búsqueda puede estar en cualquier rango con igual probabilidad. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.
3. Ahora suponga que existe un *cierto sesgo* en los datos. En particular, la probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el cajón i es **2*i / N*(N+1)** (tenga en cuenta que la suma de estas probabilidades, desde i=1 hasta N, vale 1). ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.
4. Ahora considere un caso de *sesgo extremo*. La probabilidad de que un valor de A (tanto en el registro como en la búsqueda) esté en el cajón 1 ó 2 es de 1/2 y, por lo tanto, la probabilidad es 0 para los otros rangos. ¿Cuál es el número de operaciones de I/O esperado para buscar un registro? *Justifíquelo*.

5. Si se usa un árbol B+ para la misma aplicación, ¿Cuál es el número de operaciones de I/O esperado? Suponga que los nodos de un árbol B+ contienen también 4.000 punteros y claves. *Justifique* porqué.

6. *Justifique* qué índice, el asociativo o un árbol B+, es mejor para consultas de rangos en dos situaciones: la actual y una general.

Ejercicio 17 (Cuestión teórica sobre índices y búsquedas)

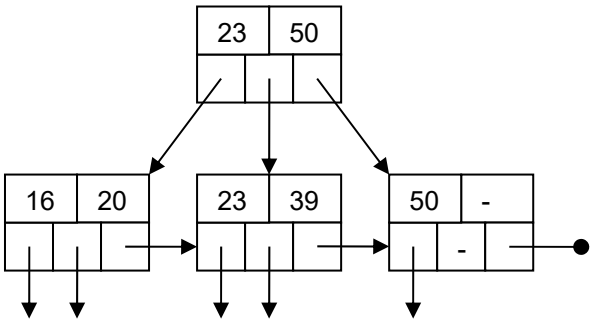
Considere una relación R(A,B,C,D,E) que contiene 5.000.000 registros, donde cada página o bloque de la relación almacena 10 registros. R está organizado como un fichero ordenado y además posee índices secundarios. Suponga que A es una clave de R, con valores en el rango 0 a 4.999.999, y que R está almacenada según el orden de A. Se le van a proponer más adelante cuatro expresiones distintas de álgebra relacional. Para cada una de las expresiones, ***justifique*** cual de las siguientes tres situaciones es más probable que sea la mejor (más económica en operaciones de I/O):

- Acceder directamente al fichero de datos ordenado por A
- Usar como índice un árbol-B+ sobre el atributo A
- Usar como índice uno asociativo (*hash*) sobre el atributo A

1. $\sigma_{A < 50{,}000} (R)$
2. $\sigma_{A = 50{,}000} (R)$
3. $\sigma_{A > 50{,}000 \wedge A < 50{,}010} (R)$
4. $\sigma_{A \neq 50{,}000} (R)$

Ejercicio 18

Considere el siguiente árbol B+ de orden 2 (2 claves y 3 punteros).



1. Muestre el árbol B+ completo (esto es, **con todas** las claves y **con todos** los punteros) que resultaría tras insertar la clave 19.
2. A partir del resultado anterior, muestre ahora el árbol B+ completo (esto es, **con todas** las claves y **con todos** los punteros) que resultaría tras borrar la clave 23.

Ejercicio 19 (Importante: índice rejilla multiclave)

Se dispone de la tabla cuenta con los siguientes campos:

Cuenta (numero_cuenta, nombre_sucursal, saldo)

Se desea crear un índice multiclave (rejilla) de dos dimensiones sobre los campos **numero_cuenta** y **saldo**. La tabla cuenta con 20.000 registros y además se sabe todos los saldos son diferentes. Se supone una distribución uniforme de valores para cada uno de los campos y se sabe que hay 5 rangos en **cada** escala lineal. Si el bloque es de 1K y cada campo de cualquier tipo ocupa 20 bytes, determinar:

a) Dibujar el esquema completo índice más archivo de datos.
b) Determinar el tamaño en bloques del archivo de datos.
c) Determinar el número de bloques del índice.
d) Cuál es el número de accesos necesarios para cargar un registro de datos en memoria