



**Examen de *Ficheros y bases de datos*
Convocatoria de junio
II PARCIAL**

- 1) (3 puntos) Considérese un fichero secuencial con un índice primario denso organizado sobre una clave sin repeticiones. Se dispone de asignación enlazada, mapas de bits de existencia para los bloques, un tamaño de bloque de 1.024 bytes y direcciones de bloque de 5 bytes. El registro de datos tiene tres campos: A (campo clave de 4 caracteres), B (150 caracteres) y C (550 caracteres) con codificación ASCII (1 byte por carácter) y tamaño fijo (sin terminador de cadena). Se pide:
- a) (1 punto) Calcular el factor de bloqueo del fichero de índices y del fichero de datos.

Solución:

Fichero de índice:

La estructura del índice: un campo para la clave (A, de 5 bytes), otro para la dirección de bloque correspondiente a la clave, mapa de bits de existencia y dirección de bloque por la asignación enlazada. Por tanto, el factor de bloqueo N se calcula como:

$$N \times (4+5) \times 8 + N + 5 \times 8 \leq 1024 \times 8$$

En esta inecuación se usa el bit como unidad y por ello todos los sumandos se multiplican por 8, salvo el mapa de bits, que ya va expresado en esta unidad.

$$N \leq 111,67$$

Por lo tanto, escogemos el entero $N = 111$.

Fichero de datos:

La estructura del fichero de datos es: campos de datos (A, de 4 bytes, B de 150 y C de 550), mapa de bits de existencia y dirección de bloque por la asignación enlazada. Por tanto, el factor de bloqueo N se calcula como:

$$N \times (4+150+550) \times 8 + N + 5 \times 8 \leq 1024 \times 8$$

$$N \leq 1,45$$

Por lo tanto, escogemos el entero $N = 1$.

- b) (1 punto) Calcular el mínimo y el máximo espacio desperdiciado en bytes en el bloque del fichero de datos.

Solución:

El espacio mínimo desperdiciado corresponde a la situación en la que hay ocupación completa en el bloque. Por lo tanto, se calcula como el tamaño total del bloque menos el espacio ocupado por el registro que puede contener y la información fija (no es necesario el mapa de bits y dirección de enlace).

$$1024 \times 8 - (8 \times (4+150+550) \times 1 + 5 \times 8) = 2519 \text{ bits} = 314 \text{ bytes y } 7 \text{ bits}$$

El espacio máximo desperdiciado corresponde al mismo caso del espacio mínimo porque solo se puede almacenar un registro en cada bloque.

- c) (1 punto) Calcúlese el tiempo de acceso al registro de datos realizando una búsqueda mediante el valor de la clave que ocupa la posición 1.000 en el fichero de índice. Asíumase que se dispone de un disco de 7.200 rpm con un tiempo de búsqueda de 5 ms y 256 sectores por pista.

Solución:

Suponiendo una distribución aleatoria de los bloques en disco, el tiempo medio de lectura o de escritura de un bloque es:

Tiempo medio operación E/S = Tiempo de búsqueda + Tiempo de latencia + Tiempo de transmisión

Tiempo de búsqueda = 5 ms

$$\text{Tiempo de latencia} = 1/2 * \text{Inversa de la frecuencia} = \frac{1}{2} \frac{1}{7.200 \frac{1}{\text{min}} \frac{1 \text{ min}}{60.000 \text{ ms}}} = 4,16 \text{ ms}$$

$$\text{Tiempo de transmisión} = \frac{1}{256} * \text{Tiempo de rotación (el doble del tiempo de latencia)} = 0,032 \text{ ms}$$

$$\text{Tiempo medio operación de lectura de un bloque} = 5 + 4,16 + 0,032 = 9,19 \text{ ms}$$

El número de accesos necesarios es:

- Fichero de índice: $\lceil 1.000 \rceil / 111 = 10$ bloques hasta llegar al registro 1.000.

- Fichero de datos: 1 bloque para leer el registro de datos.

$$\text{El tiempo total de acceso es: } 9,19 \text{ ms} * 11 = 101,09 \text{ ms}$$

2) (2 puntos) Árbol B+.

a) (0,3 puntos) Indíquense las restricciones que debe cumplir un árbol B+ con grado de salida $n = 5$ (hasta 5 hijos por nodo).

Solución:

Cada nodo interno (no hoja) tiene entre $\lceil n/2 \rceil$ y n hijos: 3-5.

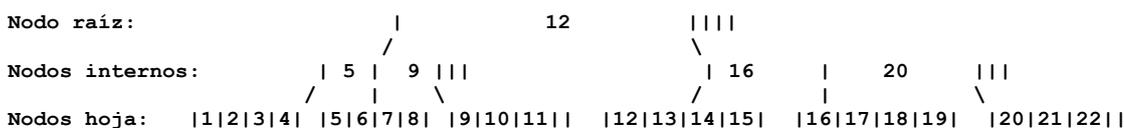
El nodo raíz tiene entre 1 y n hijos: 1-5.

Los nodos hoja contienen entre $\lceil (n-1)/2 \rceil$ y $n-1$ valores: 2-4.

b) (0,7 punto) Constrúyase un árbol B+ con este grado de salida que indexa las primeras 22 letras del alfabeto.

Solución:

Uso números en lugar de letras: El mínimo número de nodos hoja $22/4 = 5,5$. Son necesarios 6 y sobrará algún hueco. Como cada nodo puede direccionar hasta 5 hijos, son necesarios 2 niveles. Una posible solución:



c) (1 punto) Determinése el número de accesos (sólo relativos al fichero de índice) necesarios para eliminar todos los valores menos uno del nodo hoja más a la derecha. ¿Cuál es el coste si es necesario el doble de tiempo para las escrituras con respecto a las lecturas y el tiempo de acceso es de 11 ms?



```

        COUNT(DISTINCT númAsignatura) AS TOTAL_ASIG
    FROM Matriculas
    GROUP BY númAlumno;
    númAlumnos INTEGER;
BEGIN

    númAlumnos := 0;
    FOR alumno IN listaMatriculas LOOP
        DBMS_OUTPUT.PUT_LINE('Alumno: ' || alumno.ID
                             || 'Créditos: ' || alumno.TOTAL_CREDS
                             || 'Asignaturas: ' || alumno.TOTAL_ASIG);
        númAlumnos := númAlumnos + 1;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('Matriculados: ' || númAlumnos);
    DBMS_OUTPUT.PUT_LINE('-----');
END;
```

- c) (1,5 puntos) Implementar un disparador que se encargue de gestionar la dependencia funcional descrita, de modo que no se permita insertar nuevas tuplas en la tabla Matriculas que vayan en contra de la dependencia funcional.

Ayuda (definición de dependencia funcional): $X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in r$ tales que $t_1[X] = t_2[X]$ entonces $t_1[Y] = t_2[Y]$. Es decir, que los componentes Y de cada tupla de r están determinados unívocamente por los valores de X.

Solución:

```

create or replace TRIGGER control_deps_funcs
BEFORE INSERT
ON Matriculas
FOR EACH ROW
DECLARE
    nrows NUMBER;
BEGIN

    -- Comprobación de {númAsignatura, año} -> {delegado}
    SELECT COUNT(*) INTO nrows
    FROM Matriculas M
    WHERE
        :NEW.númAsignatura = M.númAsignatura
        AND :NEW.año = M.año
        AND :NEW.delegado <> M.delegado;

    -- nrows debe ser 0
    IF nrows > 0 THEN
        RAISE_APPLICATION_ERROR(-20101, 'La nueva fila viola la DF');
    END IF;

END;
```

- d) (1,5 puntos) Escribese una transacción como un procedimiento almacenado PL/SQL que modifique los alumnos matriculados de la asignatura 619 a la asignatura 900221. A continuación se calculará la suma del número de créditos de todos los alumnos matriculados en 900221. Si este número es superior a 100, se descartarán los cambios con un retroceso de la transacción y en caso contrario se comprometerán. No se debe olvidar incluir el nivel de aislamiento y el modo de autocompromiso que se considere necesario.

Solución:

```

set transaction isolation level read committed;
set autocommit off;

create or replace procedure matriculación as
    v_créditos NUMBER(6,2);
```



```
begin
  update Matriculas set númAsignatura = 900221 where código_númAsignatura =
  619;
  select sum(créditos) into v_créditos from Matriculas where númAsignatura =
  900221;
  if (v_créditos > 100) then
    rollback;
  else
    commit;
  end if;
end;
```