



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

EJERCICIO 1 (2,5 puntos)

a) Citar y explicar *brevemente* las cuatro propiedades fundamentales de las transacciones.

ATOMICIDAD: Dada una transacción o todas las operaciones se realizan o ninguna. Obliga a completar la transacción. Se encarga de cumplirla el **método de recuperación. (ROLLBACK)**

CONSISTENCIA: Una vez ejecutada la transacción se debe conservar la consistencia.

Estado consistente de la Base de Datos: Cuando los datos que hay cumplen las restricciones.
Se encargan de cumplirla los **CHECKS, TRIGGERS, etc** definidos sobre la Base de Datos por sus programadores.

AISLAMIENTO entre transacciones: Garantiza que una transacción en curso no permite acceso a sus transacciones antes de que termine. Lo evita el **control de concurrencia.**

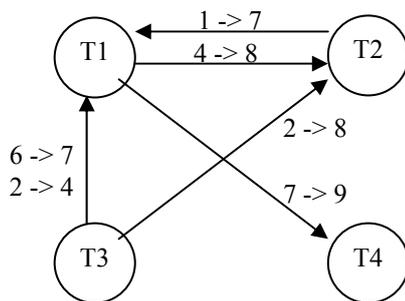
DURABILIDAD: Permanencia de los datos después de haber terminado una transacción con éxito. Se encarga de ello el **método de recuperación.**

b) ¿Cuándo es secuenciable una planificación?

Si existe otro P' secuencial, que es equivalente a P. Esto es que todo par de operaciones en conflicto se ejecutan en el mismo orden en ambos P y P'.

c) Dada la siguiente planificación, se pide contestar estos apartados:

1) Asumiendo que la lectura (L) exige un bloqueo y la escritura (E) un desbloqueo, aplicar el test de secuencialidad al plan para determinar si es secuenciable.



No es secuenciable porque tiene un ciclo entre T1 y T2.

2) ¿La planificación sería secuenciable si solo tuviera las operaciones que afectan a Y? ¿Por qué?

- Sí, porque el ciclo desaparece.

	T1	T2	T3	T4
1		L(X)		
2			E(Y)	



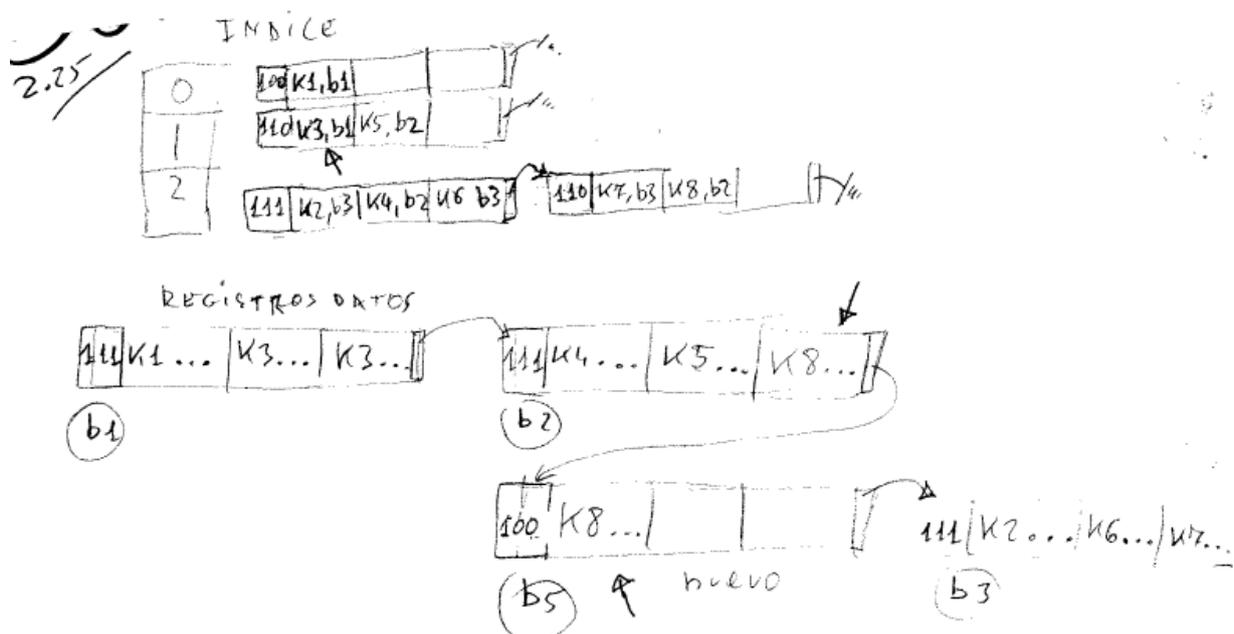
3			L(X)	
4	L(Y)			
5				
6			L(X)	
7	E(X)			
8		E(Y)		
9				L(X)

EJERCICIO 2 (2,5 puntos)

Se pide hacer un esquema (con todos los detalles) del archivo de datos y del índice para un archivo de registros de longitud fija con un índice secundario denso, teniendo en cuenta que:

- El índice denso tiene una organización de correspondencia directa (hash) con $N = 3$.
- Factor de bloque, tanto de índices como de datos, igual a 3.
- El archivo primario y desbordamiento es el mismo fichero con asignación de bloques enlazada e indicación del espacio libre con mapas de bits en cada bloque, tanto en archivo de índice como de datos.
- Los registros con igual valor de la clave aparecen consecutivos en el fichero de datos.
- En la tabla se indica la función hash y el bloque donde se encuentra el registro de datos (con repeticiones) con esa clave:

Claves	$h(K_i)$	Bloque
k1	0	b1
k2	2	b3
k3	1	b1
k3	1	b1
k4	2	b2
k5	1	b2
k6	2	b3
k7	2	b3
k8	2	b2
k8	2	¿?





EJERCICIO 3 (2,5 puntos)

Dado el siguiente esquema relacional:

```
Pilotos (NumDNI, CodCiudad, Dirección, Edad, NumTarjeta)
Ciudades (CodCiudad, Nombre, País, Tamaño)
Tarjetas (NumTarjeta, Modelo, Compañía, Consumo)
```

Se supone que las tablas están ya creadas con sus datos.

a) Se pide crear un procedimiento con el siguiente tratamiento:

- Definir el cursor adecuado para los siguientes pasos.
- Abrir un cursor con nombre Cursor3 que contenga el NumDNI y NumTarjeta para todos los pilotos de España.
- Usando el Cursor3 se desea cambiar el modelo de tarjeta a "alumno".
- Debe incluir una excepción para tratar errores inesperados.

---- no está probado

```
CREATE OR REPLACE PROCEDURE P AS
TNumt Pilotos.NumTarjeta%TYPE;
TDNI Pilotos.NumDNI%TYPE;
```

```
Cursor cursor3 is
  Select NumDNI, NumTarjeta
  From Pilotos, Ciudades
  Where Pilotos.CodCiudad = Ciudades.CodCiudad and Ciudades.Pais = 'España';
```

```
Begin
  Open cursor3
  Loop
    fetch cursor3 into TDNI,TNumt;
    exit when cursor3%NOTFOUND;
    update Tarjetas
      set modelo = 'alumno'
      where NumTarjeta = Tnumt;
  endloop;
  close cursor3;
```

```
EXCEPTION
  WHEN OTHERS THEN
    Tcoderror:= SQLCODE;
    Ttexterror:= SUBSTR(SQLERRM,1, 100);
    DBMS_output.put_line('--> ERROR desconocido ');
END P;
```

b) Escribir las instrucciones que van al comienzo de un disparador para que se active cada vez que se modifique una fila de tarjeta y cuando se cambia el atributo Modelo al valor 'alumno'.

```
create or replace TRIGGER XXXX
after update of modelo on tarjeta
for each row
when new.modelo = 'alumno'
```



EJERCICIO 4 (2,5 puntos)

Dado el siguiente esquema relacional:

```
DP (titulo, año)
RE (DNIActor, titulo, sueldo)
CR (DNICritico, titulo)
```

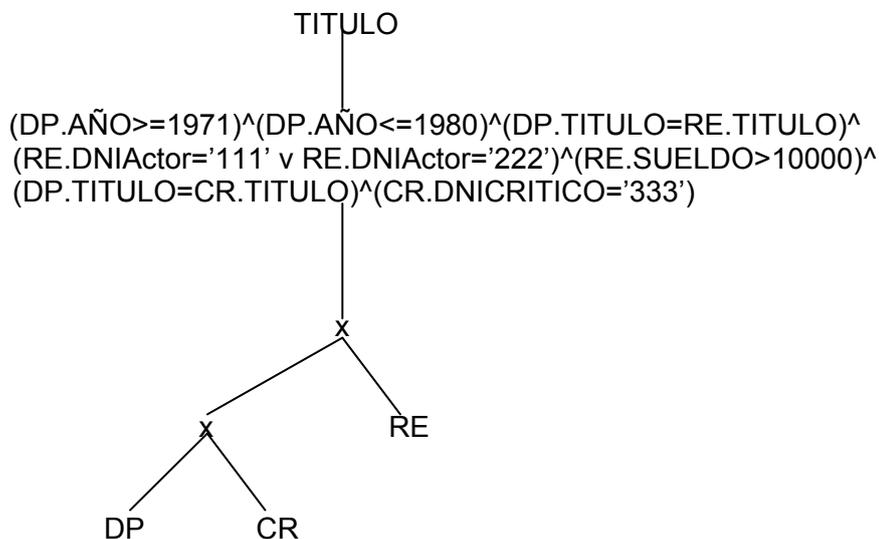
y la siguiente consulta:

```
SELECT titulo
FROM DP,RE,CR
WHERE DP.año>=1971 AND DP.año<=1980 AND DP.titulo=RE.titulo AND
(RE.DNIActor='111' OR RE.DNIActor='222') AND RE.sueldo>10000 AND
DP.titulo=CR.TITULO AND CR.DNICritico='333';
```

Donde el número de filas seleccionadas al ejecutar las condiciones para cada tabla son: el de DP menor que el de CR y éste menor que el de RE.

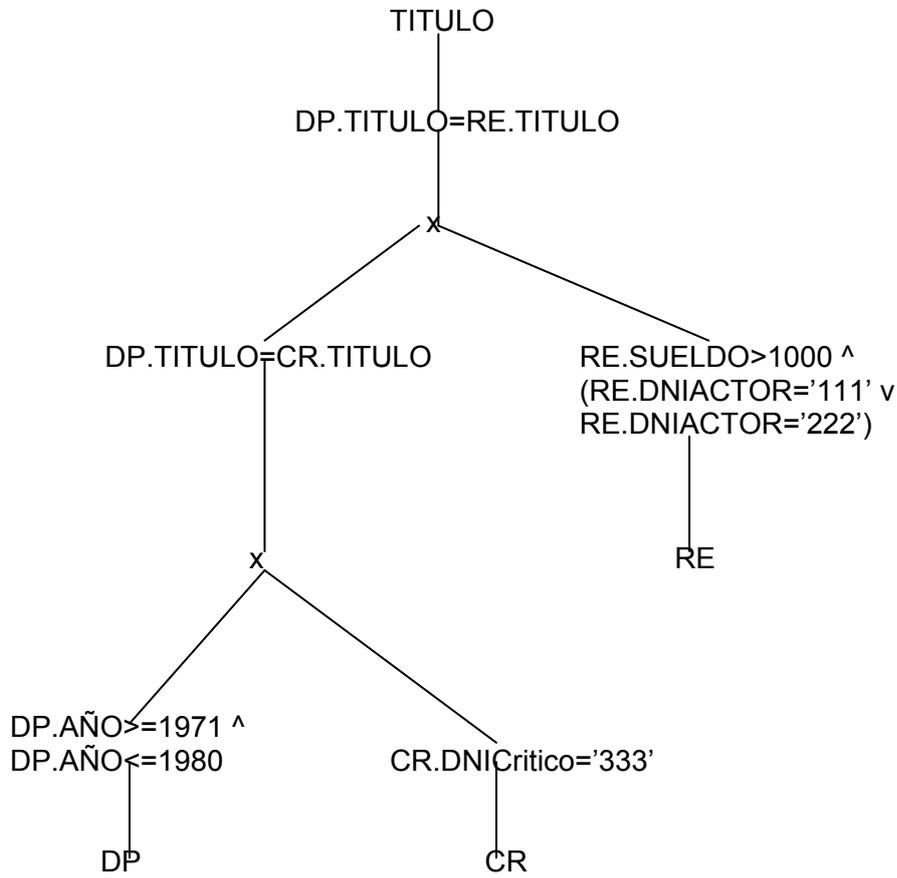
Se pide optimizar de forma heurística la consulta. Para ello aplicar el algoritmo basado en reglas de equivalencia para obtener el árbol optimizado.

- 1) Construyo el árbol. Sitúo a la izquierda las ramas más largas. Me baso en RE>CR>DP como dice el enunciado.



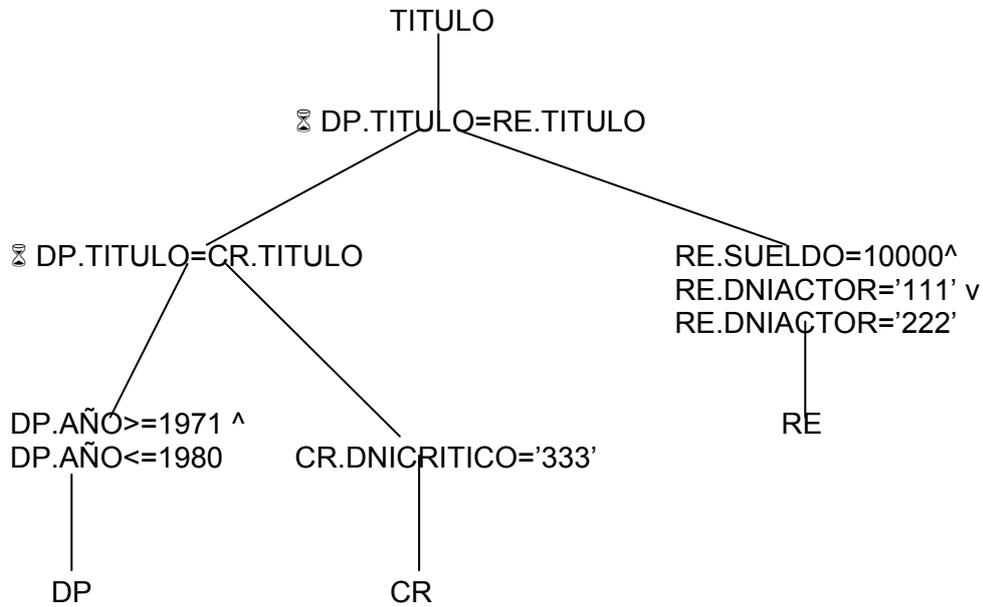


2) y 3) Bajo las selecciones y ejecuto las mas selectivas primero.





4) Cambio selecciones y productos cartesianos por productos naturales.



5) Bajo las proyecciones en el árbol.

