Ejercicios Diseño Físico

Dados los siguientes esquemas: EMP (eid, enombre, direc, sal, edad, deptid)
DEPT (did, dnombre, planta, presupuesto)
Supongamos las siguientes consultas:
1.- eid, enombre y direc de empleados con edad dentro de un rango especificado por el usuario
2.-eid, enombre y direc de empleados que trabajen en el departamento cuyo nombre es especificado por el usuario
3.- eid y direc de empleados cuyo nombre sea el especificado por el usuario
4.- salario medio de los empleados
5.- salario medio de empleados por edad (es decir, para cada edad listar la edad y el salario medio correspondiente)
6.- listar toda la información de departamentos ordenados por planta

a) Escribir las sentencias SQL correspondientes a cada consulta
b) para cada una de las consultas anteriores, considerándolas de forma individual, ¿qué índices y de qué tipo se deberían crear? Justifica tus elecciones
c) si el SGBD soporta planes solo-índice ¿cómo cambiaría tu respuesta?
d) Repetir los casos b) y c) considerando la existencia simultánea de las 6 consultas, siendo todas ellas equivalentes en frecuencia e importancia, y considerando que son más importantes que las actualizaciones.

a) CONSULTAS EN SQL:

```
1.- SELECT eid, enombre, direc      FROM EMP        WHERE edad BETWEEN  ...AND ....;

2.- SELECT eid, enombre, direc      FROM EMP, DEPT
        WHERE emp.deptid = dept.did             AND dept.dnombre = 'X';

3.- SELECT eid, direc               FROM EMP        WHERE enombre = 'X';

4.- SELECT AVG(sal)                 FROM EMP;

5.- SELECT edad, AVG(sal)           FROM EMP        GROUP BY edad;

6.- SELECT *                        FROM DEPT       ORDER BY planta;
```

b)

CON	NOMBRE	ESTRUCTURA	Agrupado/NoAgrup	Denso/Disp
1	<Emp.edad>	B+ (rango)	Agrupado (si ↑↑ duplicados, consulta por rango) No Agrupado (si ↓↓ duplicados, ordenación cara)	Disperso (+ pequeño) Denso (única opción)
2	<Emp.deptid> <Dept.dnombre>	Hash (igualdad) Hash (igualdad)	Agrupado (No clave, ∃ duplicados) No Agrupado (↓↓ duplicados, ordenación cara, No consulta por rango) Agrupado (si ↑↑ duplicados)	Denso (única opción Hash) Denso (única opción) Denso (única opción Hash)
3	<Emp.enombre>	Hash (igualdad)	Agrupado (si ↑↑ duplicados) No Agrupado (si ↓↓ duplicados, ordenación cara)	Denso (única opción Hash) Denso (única opción)
4	No interesa hacer índice porque con la opción básica el SGBD accede siempre a la tabla			
5	<Emp.edad>	Hash (se recorre todo el índice) ó B+ (ordenado)	Agrupado (↑↑ duplicados, en cada edad los empleados se obtienen consecutivamente) Agrupado (↑↑ duplicados, los empleados se obtienen consecutivamente)	Denso (única opción Hash) Disperso (+ pequeño)
6	<Dept.planta>	B+ (ordenado)	Agrupado (ordenado)	Disperso (+ pequeño)

Para la consulta 5 se puede crear un índice hash aunque no se disponga del valor de edad dentro de la consulta. En este caso la función hash no se aplica, sino que simplemente se recorre todo el índice accediendo a los diferentes registros con la misma edad (que, necesariamente, están en el mismo bucket) y calculando la media de sus salarios. Cuando se acabe de recorrer las entradas de un bucket se pasará al siguiente y así hasta recorrer todo el índice.

La diferencia entre crear un índice hash o uno B+ está en el orden de la salida de la consulta. Con el índice hash los resultados se obtendrán comenzando por los valores de las edades cuya función hash es 0 (bucket 0) seguido de las edades cuya función hash es 1 y así sucesivamente.
Por el contrario, si se crea un índice B+ la salida de la consulta mostrará las edades en orden numérico.

c) Si existen **planes solo-índice**, la mejor opción sería sustituir el índice especificado para DEPT en la consulta 2 por un índice compuesto <Dept.dnombre, Dept.did>, para no tener que acceder a dicha tabla. No puede ser Hash porque el valor concreto de Dept.did en cada caso no viene indicado en la consulta, por lo que no se podría aplicar la función Hash.
Bajo esta aproximación, la consulta 4 se vería muy favorecida de la creación de un índice por el campo sal, y para la consulta 5 se recomienda la creación de un índice compuesto <Emp.edad,Emp.sal>. Además, sería mejor opción cambiar los índices dispersos por densos si toda la información necesaria está en el índice.

CONS	NOMBRE	ESTRUCTURA	Agrupado / No Agrupado	Denso / Disperso
1	<Emp.edad>	B+ (rango)	Agrupado (∃ duplicados, consulta por rango)	Disperso (+ pequeño)
2	<Emp.deptid> **<Dept.dnombre, Dept.did>**	Hash (igualdad) **B+ (rango)**	Agrupado (ordenado) **No Agrupado (no se accede a la tabla)**	Denso (única opción Hash) **Denso (única opción para solo índice)**
3	<Emp.enombre>	Hash (igualdad)	Agrupado (si ↑↑ duplicados) No Agrupado (si ↓↓ duplicados, ordenación cara)	Denso (única opción Hash) Denso (única opción)
4	**<Emp.sal>**	**Hash (se recorre todo el índice)**	**NO Agrupado (no se accede a la tabla)**	**Denso (única opción para solo índice)**
5	**<Emp.edad, Emp.sal>**	**B+ (rango)**	**No Agrupado (no se accede a la tabla)**	**Denso (única opción para solo índice)**
6	<Dept.planta>	B+ (ordenado)	Agrupado (ordenado)	Disperso (+ pequeño)

d) Para el caso b) se definirían los mismos índices mencionados. Únicamente, es necesario considerar que, debido a que solo puede existir un índice agrupado por tabla, debería modificarse un índice en Emp.

CONS	NOMBRE	ESTRUCTURA	Agrupado / No Agrupado	Denso / Disperso
1, 5	<Emp.edad>	B+ (rango)	Agrupado (∃ duplicados, consulta por rango)	Disperso (+ pequeño)
2	<Emp.deptid>	Hash (igualdad)	**NO Agrupado (solo puede haber un índice agrupado por tabla)**	Denso (única opción)
2	<Dept.dnombre>	Hash (igualdad)	No Agrupado (solo puede haber un índice agrupado por tabla)	Denso (única opción)
3	<Emp.enombre>	Hash (igualdad)	No Agrupado (↓↓ duplicados)	Denso (única opción)
6	<Dept.planta>	B+ (rango)	Agrupado (ordenado)	Disperso (+ pequeño)

Para el caso de planes solo-índice, el índice compuesto <edad, sal> recomendado en b) para la consulta 5 sigue resultando interesante considerando la existencia de todas las consultas, descartándose los índices simples EMP.edad y EMP.sal.

CONS	NOMBRE	ESTRUCTURA	Agrupado / No Agrupado	Denso / Disperso
2	<Emp.deptid>	Hash (igualdad)	Agrupado (ordenado)	Denso (única opción Hash)
2	<Dept.dnombre, Dept.did>	B+ (rango, para cada *dnombre* se accede a todos los *did*)	NO Agrupado (**no se accede a la tabla**)	Denso (**única opción para sólo índice**)
3	<Emp.enombre>	Hash (igualdad)	No Agrupado (↓↓ duplicados)	Denso (única opción)
1, 4, 5	**<Emp.edad, Emp.sal>**	**B+ (rango)**	**NO Agrupado** (no se accede a la tabla)	**Denso** (única opción para sólo índice)
6	<Dept.planta>	B+ (rango)	Agrupado (ordenado)	Disperso (+ pequeño)

Modificar la elección de índices suponiendo que ahora las consultas son de la forma:

1.- eid y direc de empleados con un nombre de empleado (enombre) especificado por el usuario

2.- salario máximo total para empleados

3.- salario medio de empleados para cada departamento. Es decir, indicar el valor de deptid y el salario medio de los empleados en ese departamento.

4.- suma de presupuestos de los departamentos que ocupan cada planta. Es decir, para cada planta indicar la planta y su presupuesto total.

CONSULTAS EN SQL:

```
1.- SELECT eid, direc              FROM EMP          WHERE enombre = 'X';
2.- SELECT MAX(sal)                FROM EMP;
3.- SELECT deptid, AVG(sal)        FROM EMP          GROUP BY deptid;
4.- SELECT planta, SUM(presupuesto) FROM DEPT        GROUP BY planta;
```

b) En la consulta 2 se opta por un índice B+ para poder acceder a la última hoja del árbol (o a la primera, si el índice se establece en orden decreciente de salario), que corresponde al máximo. Esto no sería posible con un índice Hash.

CON	NOMBRE	ESTRUCTURA	Agrupado/NoAgrupado	Denso/Disperso
1	<Emp.enombre>	Hash (igualdad)	Agrupado (si ↑↑ duplicados) No Agrupado (si ↓↓ duplicados, ordenación cara)	Denso (única opción Hash) Denso (única opción)
2	<Emp.sal>	B+ (ordenado)	No Agrupado (1 tupla)	Denso (única opción)
3	<Emp.deptid>	Hash (se recorre todo el índice) ó B+ (ordenado)	Agrupado (↑↑ duplicados, en cada *deptid* los empleados se obtienen consecutivamente) Agrupado (↑↑ duplicados, los empleados se obtienen consecutivamente)	Denso (única opción Hash) Disperso (+ pequeño)
4	<Dept.planta>	Hash (se recorre todo el índice) o B+ (ordenado)	Agrupado (↑↑ duplicados, los departamentos se obtienen consecutivamente por *planta*)	Denso (única opción Hash) Disperso (+ pequeño)

c) Si existen planes solo-índice, la mejor opción sería sustituir el índice especificado para la consulta 3 por un índice compuesto <deptid,sal>y la consulta 4 por <planta, presupuesto>. Además, habría que cambiar todos los índices dispersos por densos para que de este modo todos los valores del índice estuviesen como entradas de datos en el índice. Por último, en estos casos (consultas 3 y 4) tampoco tiene sentido tener la tabla de datos agrupada ya que no se va a acceder a ella para la consulta.

CON	NOMBRE	ESTRUCTURA	Agrupado/NoAgrupado	Denso/Disp
1	<Emp.enombre>	Hash (igualdad)	No Agrupado (se supone ↓↓ tuplas)	Denso (única opción)
2	<Emp.sal>	B+ (ordenado)	No Agrupado (solo índice)	Denso (única opción, solo índice)
3	<Emp.deptid, Emp.sal>	B+ (ordenado, rango)	No Agrupado (solo índice)	Denso (única opción para solo índice)
4	<Dept.planta, Dept.presupuesto>	B+ (ordenado, rango)	No Agrupado (solo índice)	Denso (única opción para solo índice)

d) Quedaría todo igual

Dados los siguientes esquemas:

PROYECTO (pno, pnombre, pdept, pmgr, tema, presupuesto)

MANAGER (mid, mnombre, mdept, salario, edad, sexo)

- Cada proyecto es responsabilidad de un departamento (pdept)
- Cada *manager* pertenece a un departamento (mdept)
- El *manager* de un proyecto (pmgr) no tiene por qué pertenecer al departamento responsable (pdept) del mismo.

Supongamos que las 5 siguientes consultas son las más habituales, son equivalentes en frecuencia e importancia, y son mucho más frecuentes que las actualizaciones:

1.- Nombre, edad y salario de los *managers* de un determinado sexo (especificado por el usuario) que trabajan en un departamento determinado. Asumir que hay muchos departamentos y que en cada departamento trabajan pocos *managers*.

2.- Códigos de los departamentos donde trabajan *managers* que dirigen proyectos cuyo responsable es dicho departamento

3.- Nombres de los proyectos cuyos *managers* tienen edades comprendidas en un rango especificado por el usuario (p.ej. "menores de 30").

4.- Nombre del proyecto con el presupuesto más bajo

5.- Nombre de los *managers* que trabajan en el departamento responsable de un número de proyecto especificado por el usuario

a) Escribir las sentencias SQL correspondientes a cada consulta

1.- SELECT mnombre, edad, salario FROM MANAGER WHERE sexo = ' ' AND mdept = ' ';

2.- SELECT DISTINCT mdept FROM PROYECTO, MANAGER
WHERE pmgr = mid AND pdept = mdept;

3.- SELECT pnombre FROM PROYECTO, MANAGER
WHERE pmgr = mid AND (edad BETWEEN 'X' AND 'Y');

4.- SELECT pnombre FROM PROYECTO
WHERE presupuesto = (SELECT MIN(presupuesto) FROM PROYECTO);

5.- SELECT mnombre FROM MANAGER, PROYECTO
WHERE pdept = mdept AND pno = 'X';

b) SIN APROXIMACIÓN SOLO ÍNDICE: Si el SGBD accede SIEMPRE a la información del fichero de datos, aunque la información del índice fuese suficiente para resolver la consulta, ¿qué índices y de qué tipo se deberían crear considerando la totalidad de las consultas? Justifica tus elecciones

CONS	NOMBRE	ESTRUCTURA	Agrupado / No Agrupado	Denso/Disperso
1, 2, 5	<Manager.mdept>	Hash (igualdad)	No Agrupado (solo puede haber un índice agrupado por tabla)	Denso (única opción)
2, 3	<Proyecto.pmgr, Proyecto.pdept>	B+ (rango)	No Agrupado (↓↓ valores)	Denso(única opción)
3	<Manager.edad>	B+ (rango)	Agrupado (ordenado)	Disperso (+ pequeño)
4	<Proyecto. presupuesto>	B+ (ordenado)	No Agrupado (1 valor, o muy pocos)	Denso (única opción)
5	<Proyecto.pno>	Hash (igualdad)	No Agrupado (1 valor, ya que es clave)	Denso (única opción)

JUSTIFICACIÓN:

- Para la consulta 1 se crea un índice **Hash no agrupado por *mdept* en Manager**. Omitimos *sexo* de la clave por no ser muy selectivo (si se supone que hay managers mujeres y hombres en la misma proporción); sin embargo, su inclusión no habría sido demasiado costosa porque es muy improbable que se modifique su valor.
  Este índice también beneficia, como se verá a continuación, a las consultas 2 y 5.

- Para la consulta 2 podemos utilizar un índice **B+ por los campos <*pmgr, pdept*> en Proyecto**. De este modo, el gestor puede seguir los siguientes pasos:
  1) acceder a la primera hoja del árbol y obtener el primer valor *pdept*.
  2) aplicar a *pdept* la función hash asociada al índice de *mdept* de Manager creado en el punto anterior. De este modo, se puede localizar el bucket donde se encuentra el valor *mdept* asociado a este *pdept.*
  3) recuperar de disco el(los) registro(s) correspondiente(s) de la tabla Manager que se corresponden al valor *mdept* y traerlos a memoria
  4) comparar el valor del atributo *mid* de cada registro recuperado de Manager con el valor *pmgr* existente en el índice B+.
  5) repetir este proceso hasta que se haya recorrido todo el árbol.

  Se crea del tipo B+ (y no Hash) para poder localizar las entradas para cada *pmgr*. Esto es utilizado para la consulta 3.
  Otra opción sería crear un índice compuesto para Manager con los atributos <mdept, mid> de tipo B+. De este modo se podría resolver de modo similar a la opción sólo-índice (ver explicación más adelante), aunque con 1 acceso a la tabla de Manager en caso de coincidencia de valores.

- Para la consulta 3 se crea un **índice agrupado por *edad*** para filtrar los managers con edad en el rango especificado, y traer los registros correspondientes a memoria. Luego, para cada *mid*, se localizan los proyectos que dirige mediante el índice en Proyecto creado anteriormente.

- Para la consulta 4 puede crearse un **índice B+ por *presupuesto* en Proyecto** y moverse por los nodos hoja del árbol hasta encontrar el presupuesto más bajo.

- Para la consulta 5 es necesario crear un **índice por *pno*** para Proyecto. A través de este índice se puede obtener el departamento *pdept*, y utilizar posteriormente el índice *mdept* en Manager para obtener los managers de dicho departamento.

c) APROXIMACIÓN SOLO ÍNDICE: Si el SGBD incluyese la posibilidad de NO acceder a la información del fichero de datos si la información del índice fuese ya suficiente para resolver la consulta, ¿cómo cambiaría tu respuesta?

CONS	NOMBRE	ESTRUCTURA	Agrupado / No Agrupado	Denso / Disperso
1, 2, 5	**<Manager.mdept, Manager.mid>**	**B+ (rango)**	**No Agrupado (no se accede a la tabla)**	**Denso (única opción)**
2, 3	<Proyecto.pmgr, Proyecto.pdept>	B+ (rango)	No Agrupado (no se accede a la tabla)	Denso (única opción)
3	**<Manager.edad, Manager.mid>**	**B+ (rango)**	**No Agrupado (no se accede a la tabla)**	**Denso (única opción)**
4	**<Proyecto.presupuesto, Proyecto.pnombre>**	**B+ (rango)**	**No Agrupado (no se accede a la tabla)**	**Denso (única opción)**
5	**<Proyecto.pno, Proyecto.pdept>**	**Hash (igualdad)**	**No Agrupado (no se accede a la tabla)**	**Denso (única opción)**

JUSTIFICACIÓN:

- Para la consulta 2, al crear un índice por <*mdept*, *mid*> (y al existir el de Proyecto por <*pmgr, pdept*>), se puede resolver sin tener que acceder a ninguna de las dos tablas.

- Con los índices creados para la consulta 3 podemos localizar directamente las entradas en el índice de Manager de los *managers* cuya edad esté en el rango especificado (sin necesidad de acceder a Manager), y utilizar el índice de *Proyecto.pmgr* para conocer los proyectos que dirige cada manager.
  Sólo es necesario acceder al registro de la tabla Proyecto para recuperar el nombre.

- Para la consulta 5 se podría crear un índice compuesto <*pno*, *pdept*> en Proyecto. Sin embargo, ya que para cada proyecto hay un único departamento responsable no mejorará considerablemente el rendimiento (evita únicamente una operación I/O por proyecto).