

Fragmentación y Asignación

Profesor Heider Sanchez

P1. Estudiante

Dado la siguiente tabla Estudiante con los siguientes datos:

<u>MATRICULA</u>	NOMBRE	DIR	GRUPO	PROMEDIO	EDAD	SEXO
0001	Selene Aguirre	Las palmas 34	A	8.5	17	F
0002	Martin Porres	Blvd. Lopez Mateos	C	9	23	M
0003	Miriam Gutiérrez	Águila 34	A	7	21	F
0004	Benito López	Rueda 23	B	10	19	M
0005	Victor Pérez	Carlos Carrillo 567	B	8	20	M
0006	Nicolás Rosas	20 de nov. 123	C	10	22	M
0007	María Gutiérrez	Av. Américas No. 65	A	6	18	F
0008	José Carrillo	Azueta No. 23	C	8	19	M
0009	Azucena Pérez	Abasolo No. 44	B	9	21	F
0010	Carlos Sosa	Arco Sur No. 426	A	7	17	M
0011	Luis Espino	Hidalgo No. 65	C	10	25	M

- Se sabe que un 60% de las consultas se realizan sobre el campo GRUPO y un 35% sobre el campo EDAD. Usando la técnica de términos mínimos, halle el conjunto de fragmentos finales en base a ambos atributos considerando el vector de partición de EDAD [18, 22].

Predicados: Grupo={A,B,C}, Edad={<18, 18<= and <22, >=22}

60%: select ... From estudiante where grupo = ? and/or...

35%: select ... from estudiante where edad <? and/or...

F1: edad < 18 and grupo=A

F2: edad < 18 and grupo=B

F3: edad < 18 and grupo=C

F4: edad >= 18 and edad < 22 and grupo=A

F5: edad >= 18 and edad < 22 and grupo=B

F6: edad >= 18 and edad < 22 and grupo=C

F7: edad >=22 and grupo=A

F8: edad >=22 and grupo=B

F9: edad >=22 and grupo=C

- Asumiendo que tenemos tres servidores de base de datos disponibles, como quedarían distribuido los fragmentos.

	Fragmentos
Servidor 1	F1, F4, F7 // GRUPO A
Servidor 2	F2, F5, F8 // GRUPO B
Servidor 3	F3, F6, F9 // GRUPO C

3. El área de Control Escolar realiza constantes consultas sobre los campos NOMBRE, DIR y GRUPO, mientras que al área de Estadística le interesan los campos PROMEDIO, EDAD y SEXO.

1. Realice la fragmentación vertical de la tabla.

R1(MATRÍCULA, NOMBRE, DIR, GRUPO)

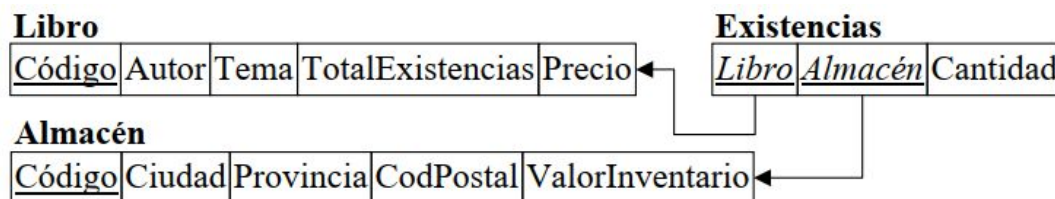
R2(MATRÍCULA, PROMEDIO, EDAD, SEXO)

2. Cree una vista en SQL para unificar los fragmentos. Verifique que cumpla la propiedad de Join sin pérdida.

CREATE VIEW Reconstruccion AS SELECT R1, R2

P2. Libros

Dado el siguiente esquema de BD:



1. Realizar la fragmentación horizontal de Libro sobre el atributo Precio [20,50,100]
 - L1: Precio <=20
 - L2: Precio >20 and Precio <=50
 - L3: Precio > 50 and Precio <=100
 - L4: Precio>100
2. Realizar la fragmentación horizontal de Almacén sobre el atributo CodPostal [3500,70000]

A1: CodPostal <3500
 A2: Codpostal >=3500 and CodPostal<70000
 A3: Codpostal >= 70000

3. Realizar la fragmentación horizontal derivada de Existencias respecto a Almacén.

E1: Existencias SEMIJOIN A1
 E2: Existencias SEMIJOIN A2
 E3: Existencias SEMIJOIN A3

4. Como sería la creación de dicha fragmentación derivada en PostgreSQL.

CREATE TABLE E1 AS SELECT E* FROM EXISTENCIAS E WHERE A1.CODIGO=E.ALMACEN;
 CREATE TABLE E2 AS SELECT E* FROM EXISTENCIAS E WHERE A2.CODIGO=E.ALMACEN;

CREATE TABLE E3 AS SELECT E* FROM EXISTENCIAS E WHERE A3.CODIGO=E.ALMACEN;

5. Si se tiene tres servidores disponibles, asigne a criterio los fragmentos resultantes en cada servidor:

	Servidor 1	Servidor 2	Servidor 3
Libro	L1	L2,L3	L4
Almacén	A1	A2	A3
Existencias	E1	E2	E3

6. En base a la asignación anterior. Qué subconsultas genera la ejecución de la siguiente consulta en cada servidor:

```
select Código, TotalExistencias
from Libro
where Precio>15 and Precio<55
```

SQL- S1:

```
select Código, TotalExistencias
from Libro
where Precio>15 and Precio<=20
```

SQL- S2:

```
select Código, TotalExistencias
from Libro
where Precio>20 and Precio<55
```

SQL- S3:

NO SE EJECUTA NADA

7. Escribir un ejemplo de consulta que ejecutada en el servidor 3 genere una subconsulta en el servidor 2.

```
select Código, TotalExistencias
from Libro L, Almacen A
where Precio>100 and A.Codpostal=15000
```

P3. Fragmentación vertical

Se considera las aplicaciones AP1, AP2, AP3 y AP4. Estas aplicaciones trabajan con la tabla T definido como T(C, C1, C2, C3, C4) donde C es la clave primaria:

- AP1: Select C1 from T where C4 = 100;
- AP2: Select C4 from T;
- AP3: Update T set C3 = 15 where C2=50;
- AP4: Update T set C1=5 where C3=20;

Además, se tiene las frecuencias de acceso de las aplicaciones a los atributos de la siguiente forma:

acc1(AP1)=1 acc2(AP1)=0 acc3(AP1)=2 acc4(AP1)=0

acc1(AP2)=0 acc2(AP2)=4 acc3(AP2)=3 acc4(AP2)=0

acc1(AP3)=0 acc2(AP3)=0 acc3(AP3)=4 acc4(AP3)=0

acc1(AP4)=3 acc2(AP4)=0 acc3(AP4)=0 acc4(AP4)=0

Se pide construir lo siguiente:

1. Matriz de usos.

	C1	C2	C3	C4
Q1	1	0	0	1
Q2	0	0	0	1
Q3	0	1	1	0
Q4	1	0	1	0

2. Matriz de afinidad.

	C1	C2	C3	C4
C1	6	0	3	3
C2	0	4	4	0
C3	3	4	7	0
C4	3	0	0	10

3. Matriz de afinidad agrupada.

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

CONT(C0, C4,C1)= 2 * (bond(0,4) + bond(4,1) - bond(0,1)) = 96

CONT(C1,C4,C3) = 2 * (bond(1,4) + bond(4,3) - bond(1,3)) = 2* (48 + 9 - 39)

CONT(C3,C4,C2) = 2 * (bond(3,4) + bond(4,2) - bond(3,2)) = 2*(9 + 0 - 44)

CONT(C2,C4,C5) = 2 * ((bond(2,4))= 0

4. Fragmentación vertical resultante.

	C4	C1	C3	C2
C4	10	3	0	0
C1	3	6	3	0
C3	0	3	7	4
C2	0	0	4	4

Entregable: en formato PDF.