

FUNDAMENTOS DE BASES DE DATOS

Examen Diciembre 2018

SOLUCION

La duración del examen es de 3 horas y media.

Presentar la resolución del examen:

- Con las hojas numeradas y escritas de un solo lado.
- Con la cantidad de hojas entregadas en la primer hoja.
- Con cédula de identidad y nombre en cada hoja.
- Escrita a lápiz y en forma prolija.
- Comenzando cada ejercicio en una nueva hoja

Ejercicio 1 (27 puntos).

Una inmobiliaria desea modelar la administración de los distintos espacios comunes de un grupo de edificios y gastos que se generan en los mismos.

La inmobiliaria considera a distintas personas, los administradores de los edificios y los usuarios de los mismos. Los usuarios pueden ser inquilinos y/o propietarios. De todas las personas se conoce CI, nombre y todos sus teléfonos. De los administradores también se conoce su e-mail que es único.

De los edificios se conoce un código que los identifica, un nombre y una dirección (con calle y número). Cada edificio tiene un conjunto de apartamentos (aptos) y un conjunto de barbacoas. De cada apto se conoce el número que lo identifica dentro del edificio, los metros cuadrados y la cantidad de ambientes que tiene. Mientras que de cada barbacoa se conoce, también el número que la identifica dentro del edificio y su capacidad.

La inmobiliaria registra información sobre todos los aptos que tiene un propietario y se sabe que cada apto puede tener más de un propietario. Cuando un apto se alquila se registra el inquilino, la fecha de inicio y el valor del alquiler. Un inquilino puede alquilar varios aptos, pero los aptos son alquilados por un solo inquilino. Por otro lado, se registran todas las fechas en las que un usuario (ya sea inquilino o propietario) utilizó una de las barbacoas. Por supuesto, un usuario solo puede usar las barbacoas que están en el mismo edificio que el apto que alquila y/o es propietario.

En cada edificio se realizan asambleas donde se abordan distintos temas. Para cada asamblea se registra la fecha en que se realiza, que la identifica, y los temas que se tratan. A las asambleas solo asisten los propietarios de los aptos. Para cada propietario que asiste a una asamblea, interesa registrar el apto al cual representa, ya que podría representar a uno o varios de sus aptos. Cada propietario puede asistir a todas las asambleas que le interesen y, además, podría ir más de un propietario (de un mismo apto), a cada asamblea. Podría pasar que no asista nadie a alguna asamblea.

Finalmente, a la inmobiliaria le interesa saber cuáles son los administradores que se encargan de cada edificio, ya que podrían ser varios los que se encargan de un edificio.

Se pide: Modelo Entidad-Relación completo del problema.

Como la asamblea tiene un identificador propio, la primer solución que surge es la que se muestra en la figura 1, con las *Asambleas* como una entidad fuerte. Sin embargo, otra solución posible es considerar a las *Asambleas* como una entidad débil de los edificios. Si bien las asambleas tienen un identificador propio, se sabe que las asambleas siempre se realizan en un edificio, por lo que no existen asambleas que no se asocien a ningún edificio. De esto último surge la solución 2 que se muestra en la figura 2.

Las RNE son válidas para ambas soluciones y las mismas se muestran en la figura 3.

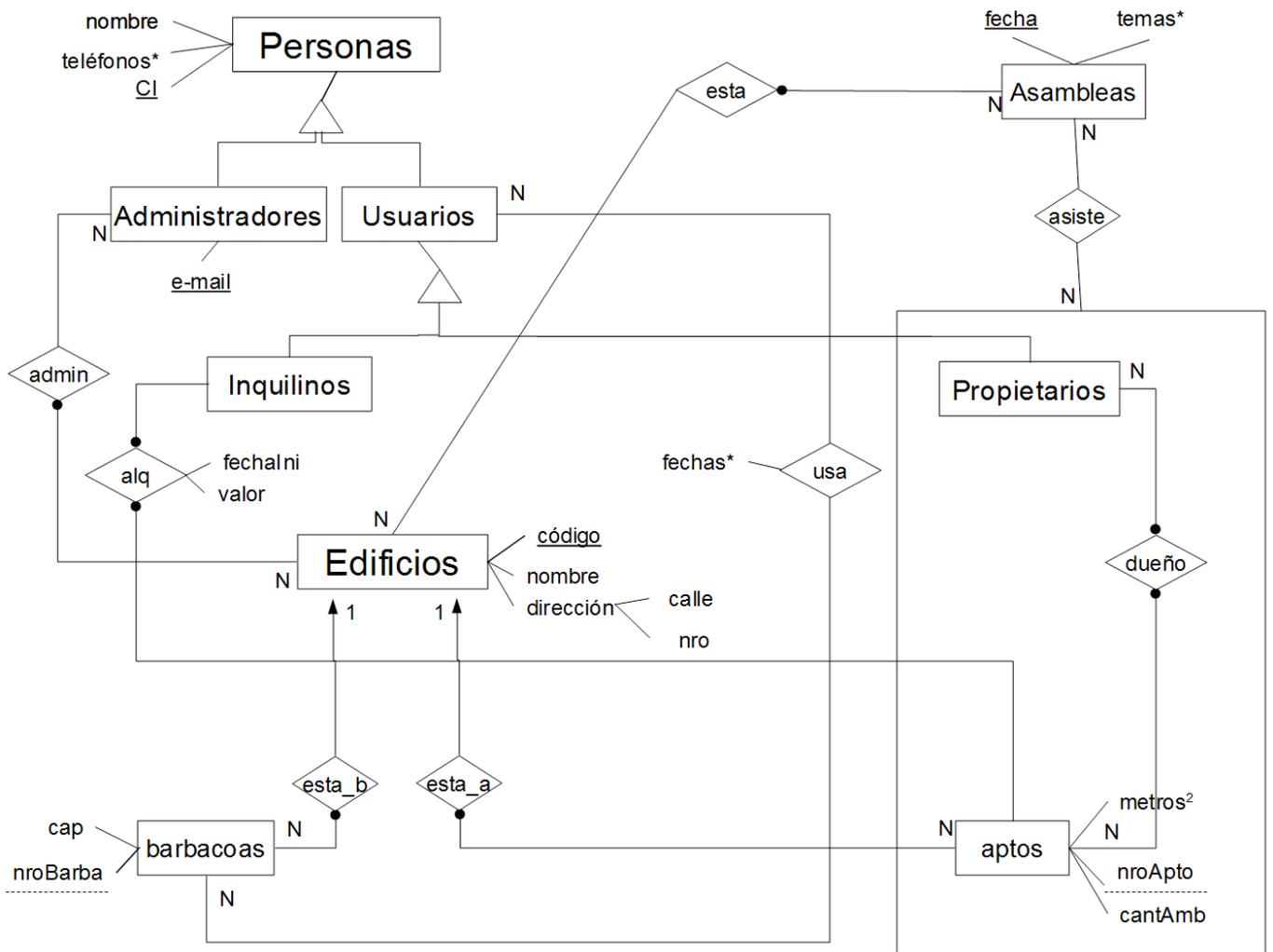


Figura 1. MER con Asambleas como entidad fuerte.

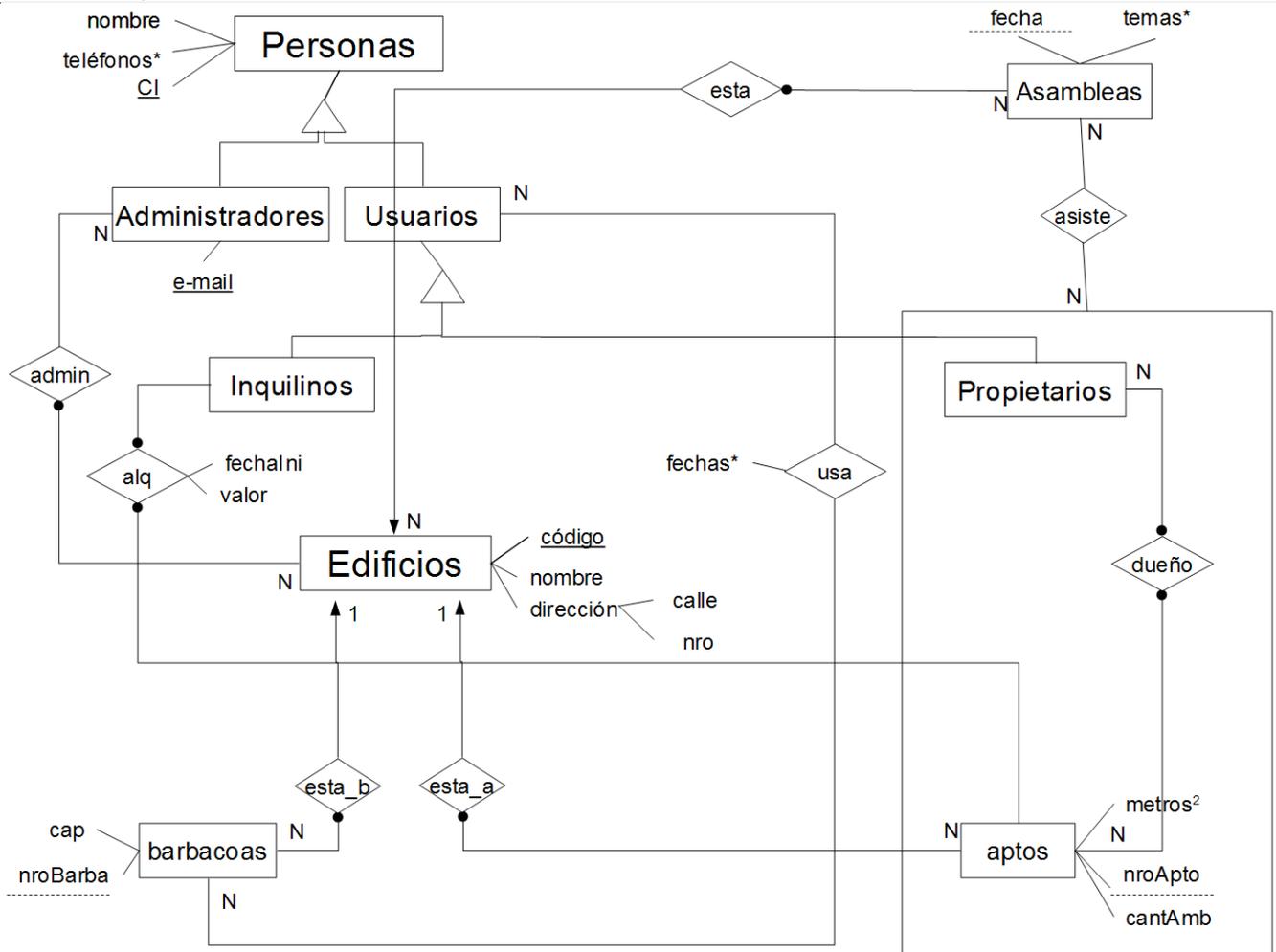


Figura 2. MER con Asambleas como entidad débil.

RNE:

1. $Personas = Administradores \cup usuarios$
2. $Usuarios = Inquilinos \cup Propietarios$

3. un usuario solo puede usar las barbacoas que están en el mismo edificio que el apto que alquila y/o es propietario:

$$(\forall b \in Barbacoas)(\forall u \in Usuarios)(\langle u, b \rangle \in usa \rightarrow (\exists e \in Edificios)(\exists a \in Aptos)(\langle b, e \rangle \in esta - b \wedge \langle a, e \rangle \in esta - a \wedge (\langle u, a \rangle \in dueño \vee \langle u, a \rangle \in alq)))$$

4. un propietario asiste a una asamblea de un edificio al cual pertenece uno de sus aptos:

$$(\forall p \in Propietarios)(\forall a \in Aptos)(\forall s \in Asambleas)(\langle a, p \rangle \in dueño \wedge \langle \langle a, p \rangle, s \rangle \in asiste \rightarrow (\exists e \in Edificios)(\langle s, e \rangle \in esta \wedge \langle a, e \rangle \in esta - a))$$

Figura 3. RNE del MER.

Ejercicio 2 (28 puntos).

Considere la siguiente relación con el siguiente conjunto de dependencias funcionales:

$R(A, B, C, D, E, G, H)$ $F = \{ BD \rightarrow A, C \rightarrow H, DH \rightarrow E, GC \rightarrow D, DB \rightarrow G \}$

1) Indicar cuáles de las siguientes dependencias están en F^+ y cuáles no. Justifique.

- a. $AG \rightarrow B$
- b. $CBG \rightarrow E$
- c. $BC \rightarrow A$

$(AG)^+ = \{ A G \} \Rightarrow \mathbf{AG \rightarrow B \notin F^+}$

$(CBG)^+ = \{ C B G H D A E \} \Rightarrow \mathbf{CBG \rightarrow E \in F^+}$

$(BC)^+ = \{ B C H \} \Rightarrow \mathbf{BC \rightarrow A \notin F^+}$

2) Hallar todas las claves de R. Justificar.

Todos los atributos de R están en alguna df.

B y C pertenecen a todas las claves porque no se encuentran a la derecha en ninguna df.

A y E no pertenecen a ninguna clave porque no se encuentran a la izquierda en ninguna df.

Sabemos que (BC) no es clave y que (BCG) es clave por las clausuras calculadas en la parte anterior.

$(BCH)^+ = \{ B C H \}$

$(BCD)^+ = \{ B C D A H E G \} \Rightarrow (BCD)$ es clave

No hay más claves porque si intentamos con 4 atributos, habría que agregar algún atributo a (BCH), pero los únicos posibles son D y G y en esos casos obtendría superclaves.

Las únicas claves son: (BCG) y (BCD)

3) Decir si F es un conjunto minimal, justificando su respuesta. Si no lo es, hallar un cubrimiento minimal de F. Si lo es, hallar un conjunto K equivalente a F, que no sea minimal.

F es un conjunto minimal, ya que cumple:

- Todas sus dfs tienen un sólo atributo a la derecha
- Ninguna df tiene atributos redundantes a la izquierda:

En las dfs: $BD \rightarrow A, DH \rightarrow E, DB \rightarrow G$ si sacamos un atributo a la izquierda, no se puede aplicar ninguna df al atributo que queda solo.

En $GC \rightarrow D$: $G^+ = \{ \}$ y $C^+ = \{ C H \}$, por lo que ni G, ni C es redundante.

- No hay dfs redundantes:

Todos los lados derechos de las dfs son distintos, por lo tanto ninguno puede determinarse si no está esa df.

$$K = \{ BDG \rightarrow A, C \rightarrow H, DH \rightarrow E, GC \rightarrow D, DB \rightarrow G \}$$

K es equivalente a F porque:

- todas las dfs de F pertenecen a K^+ , ya que $BD \rightarrow A$ es la única diferente y se cumple que $(BD)^{+K} = \{ B D G A \}$
 - todas las dfs de K pertenecen a F^+ , ya que $(BDG)^{+F} = \{ B D G A \}$
- y K no es minimal, ya que $(BD)^{+K} = \{ B D G A \}$, por lo tanto G es redundante en $BDG \rightarrow A$

4) Llevar R a 3NF con join sin pérdida y preservación de dependencias, aplicando el algoritmo visto en el curso.

- Primer paso: Utilizamos F, ya que es minimal.

- Segundo paso: Generamos los esquemas relación juntando los atributos de las dfs que tienen el mismo lado izquierdo.

R1 (B, D, A, G)

R2 (C, H)

R3 (D, H, E)

R4 (G, C, D)

- Tercer paso: Si ningún esquema contiene a una clave, generamos uno nuevo con una clave.

R5 (B, C, G)

Descomposición obtenida: $\rho = \{R1, R2, R3, R4, R5\}$

5) Dada la siguiente descomposición de R: $\rho = \{ R1(A,B,D,G), R2(A,C,D,H,E) \}$

1. Decir si ρ tiene join sin pérdida con respecto a F, justificando.

$$R1 \cap R2 = (A D)$$

$$R1 - R2 = (B G)$$

$$R2 - R1 = (C H E)$$

Se cumple $AD \rightarrow BG$? NO

Se cumple $AD \rightarrow CHE$? NO => **La descomposición NO tiene JSP**

2. Para cada R_i decir en qué forma normal se encuentra, justificando.

R1:

$$F_{R1} = \{ BD \rightarrow A, BD \rightarrow G \}$$

BD es única clave

R1 está en **BCNF**

R2:

$$F_{R2} = \{ C \rightarrow H, DH \rightarrow E \}$$

A, C y D pertenecen a todas las claves (por no estar a la derecha en ninguna df), E no pertenece a ninguna (por no estar a la izquierda en ninguna df).

$(ACD)^+ = \{ A, C, D, H, E \}$, ACD es la única clave

$C \rightarrow H$ viola 2NF, ya que H no es primo y depende en forma parcial de una clave

R2 está en **1NF**

3. Decir si es posible que se cumpla la dependencia multivaluada $A \twoheadrightarrow BD$ en R1, justificando.

Si se cumple $A \twoheadrightarrow BD$, entonces puede existir la siguiente instancia:

A	B	D	G
a1	b1	d1	g1
a1	b2	d2	g2
a1	b1	d1	g2
a1	b2	d2	g1

Sabemos que $F_{R1} = \{ BD \rightarrow A, BD \rightarrow G \}$, y podemos observar que esta instancia viola la df $BD \rightarrow G$. Por lo tanto, no es una instancia válida, entonces **$A \twoheadrightarrow BD$ no puede cumplirse.**

Ejercicio 3 (25 puntos)

Considere una asociación de pequeños almacenes que les permite hacer compras conjuntas. La asociación tiene un depósito en donde acumula compras que hace a los distribuidores y luego los socios retiran del depósito una determinada cantidad de cada producto. Cada almacén vende los productos que retira a determinado precio unitario, el cual queda a criterio del almacenero.

A continuación se describen las tablas de esta realidad.

Socios (idSocio, nombre, dir, tel)

Cada tupla representa un socio.

Distribuidores(idDist, nomDist, dirDist, telDist)

Cada tupla contiene la información de un distribuidor determinado.

Productos(idProd, categoria, desc, marca, unidad, presentacion,)

Cada tupla contiene la información de un producto. La categoría indica si el producto es un comestible, una bebida, un producto de tocador o de limpieza, bazar, etc. La descripción (desc) y la marca son textos. La unidad indica la unidad de medida que se considera para ese producto, y la presentación es el número de unidades. De esta forma, una paquete de harina podría tener unidad "kg" y presentación 1 y otro podría tener unidad "g" y presentación 500. Estos dos casos representan productos distintos.

Compra(idProd, idDist, fechaCompra, precio, cantidad)

Cada tupla representa una compra que realiza la asociación a un distribuidor. Se registra el producto, el distribuidor, la fecha, el precio unitario al que se compró y la cantidad de ítems que se compraron.

Retiros(idProd, idSocio, fechaRet, cantRet, precioVU)

Cada tupla representa un retiro de determinado producto por un socio. Se registra la cantidad de ítems, la fecha y el precio de venta unitario al cual el socio lo venderá.

Se Pide:

En álgebra relacional:

a) Devolver los nombres de los socios que a lo largo de toda su historia, retiraron algún producto proveniente de cada uno de los distribuidores a los que la asociación les compró alguna vez.

SOL:

$$A = \Pi_{idSoc, idDist}(Compra * Retiros)$$

$$B = A \% (\Pi_{idDist}(A))$$

$$Res = \Pi_{nombre}(B * Socios)$$

En cálculo relacional:

b) Devolver los identificadores, nombres y direcciones de los distribuidores de algún productos de limpieza que fue retirado por un sólo socio.

SOL:

$$\{ \langle t.idDist, t.nomDist, t.dirDist \rangle /$$

$$Dist(t) \wedge \exists p (Prods(p) \wedge p.categoría = 'limpieza' \wedge$$

$$\exists c (Compra(c) \wedge t.idDist = c.idDist \wedge c.idProd = p.idProd) \wedge$$

$$\exists r (Retiro(r) \wedge r.idProd = p.idProd \wedge$$

$$\neg \exists r_1 (Retiro(r_1) \wedge r.idSocio \neq r_1.idSocio \wedge$$

$$r_1.idProd = p.idProd)$$

$$)$$

$$)$$

$$)$$

$$\}$$

En SQL (sin utilizar sub-consultas en el FROM):

c) Devolver la identificación y descripción de los productos con stock total en 0 o negativo.

```

Select idProd, desc
from productos P
where ( select sum(cantidad)
from compras C
where C.idProd=p.IdProd
<=
( select sum(cantRet)
from retiros R
where R.idProd=p.IdProp)

```

Otra solución posible:

```

Select idProd, desc
from productos P
where p.idProd in (
select C..idProd
from compras C, retiros R
where C.idProd=R.IdProd)
group by C.idProd
having(sum(C.cantidad) -sum(R.cantidad) <= 0)

```

d) Devolver para cada producto, la fecha del primer retiro, la cantidad de retiros realizados y el precio promedio de venta considerando todos los retiros de ese producto, para los productos que tienen más de 10 retiros.

```

select idProd, min(fechaRet), count(*), avg(precioVU)
from Retiros
group by idProd
having count(*) > 10

```

Ejercicio 4 (20 puntos)

Considere parte del esquema de la bases de datos de la administración de liceos del Uruguay, en la cual se registra a los funcionarios asignados para dar mantenimiento a los bancos de los salones de cada liceo:

Liceos (nroLiceo, departamento, cantSalones)

Para cada liceo, esta tabla contiene el número identificador de cada liceo, el departamento en el cual se encuentra el mismo y la cantidad de salones que contiene.

Salones (nroLiceo, nroSalon, piso, cantBancos)

La tabla contiene el número de liceo al cual pertenece cada salón y el piso en el que se encuentra, además la cantidad de bancos que hay en el mismo.

Funcionarios (ciFun, nroLiceo, nroSalon, fechaAsignacion)

Esta tabla contiene el número de cada funcionario, el número del liceo y el salón al cual fue asignado. Además, las fechas en la que fue asignado a cada salón.

Considere la siguiente consulta SQL sobre esta base de datos:

```
SELECT F.ciFun, L.nroLiceo, S.nroSalon, F.fechaAsignacion
FROM Funcionarios F, Salones S, Liceos L
WHERE L.nroLiceo = S.nroLiceo AND S.nroLiceo = F.nroLiceo AND
      S.nroSalon = F.nroSalon AND L.departamento = 'Montevideo' AND
      S.cantBancos > 30;
```

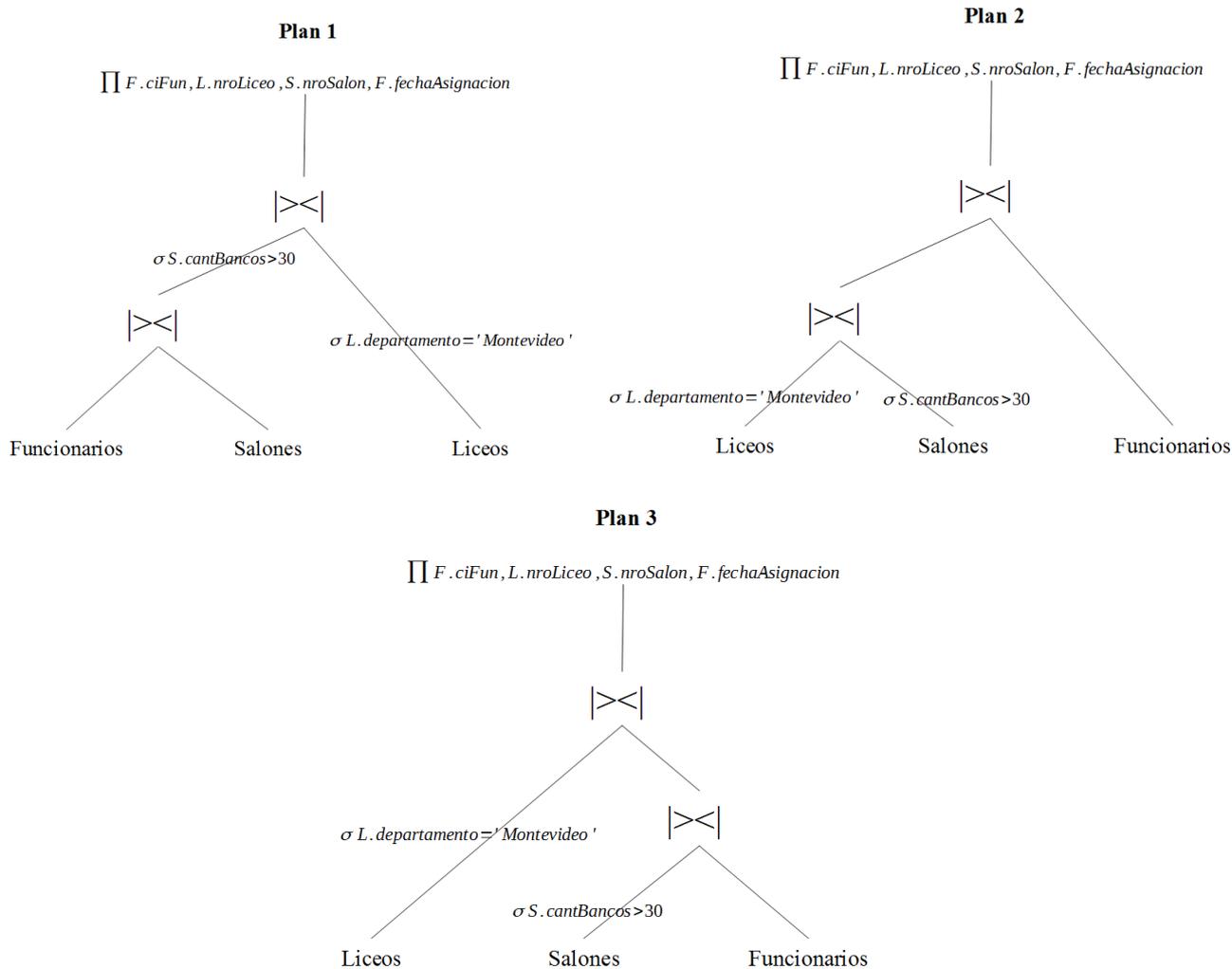
En cada tabla existe un índice primario por la clave primaria correspondiente. Pero además, existen los siguientes índices:

Tabla	Tipo	Atributos	Nombre
Liceos	Índice secundario	Departamento	indDepartamento
Funcionarios	Índice secundario	NroLiceo, nroSalon	indLiceoSalon

Se sabe que todos los atributos tienen distribución uniforme y que el 15% de las tuplas de la tabla *Salones* verifican $cantBancos > 30$. Recuerde que $V(a,T)$ es la cantidad de valores distintos que tiene un atributo 'a' en una tabla 'T'. Además, de cada tabla se conoce la siguiente información:

Tabla	Cantidad de Tuplas	Atributos
Liceos	57	$V(\text{departamento}, L) = 19$
Salones	1000	$V(\text{piso}, S) = 10$ $V(\text{cantBancos}, S) = 25$
Funcionarios	4500	

1. Dados los 3 posibles planes lógicos para la consulta anterior, indique, **justificando su respuesta**, el mejor plan según las heurísticas. Además, describa brevemente por qué los otros dos planes no fueron seleccionados.



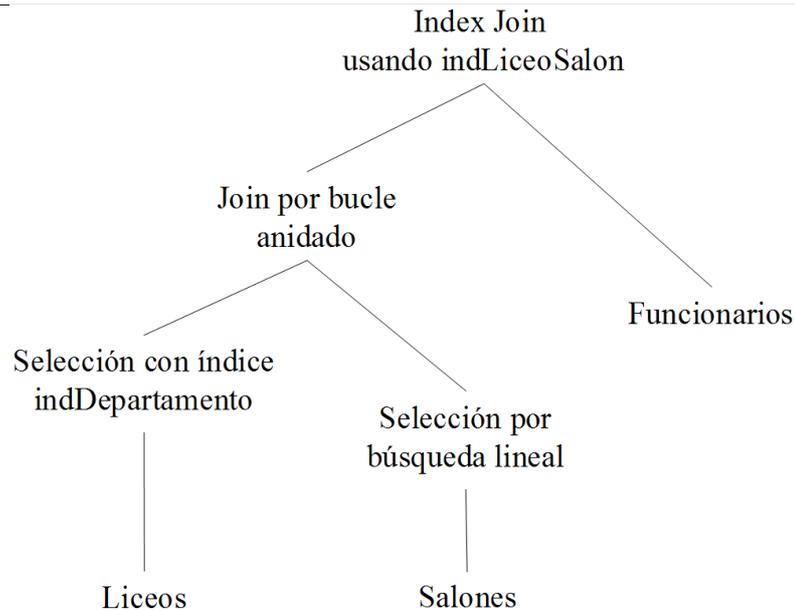
SOLUCION:

El mejor es el Plan 2, porque el árbol está recostado a la izquierda, realiza las selecciones lo antes posible y además tiene las ramas más restrictivas colocadas a la izquierda del árbol.

Por otro lado, el Plan 1 y el Plan 3 no pueden ser resultado de la aplicación de las heurísticas, ya que el Plan 1 no baja las selecciones lo más posible en el árbol, y el Plan 3 no se encuentra recostado a la izquierda.

2. Para el plan lógico seleccionado en la parte a) dar 1 plan físico que utilice los índices cuando sea posible.

SOLUCION:



3. ¿Qué información necesitaría para calcular el costo de almacenamiento del resultado de la consulta en disco?

SOLUCION:

Le llamaremos JOIN1 al join entre Liceos y Salones y JOIN2 al join entre el resultado del join anterior y Funcionarios.

El costo de almacenamiento del resultado de la consulta es igual a la cantidad de bloques que ocupa el resultado final. Para calcular esto, es necesario conocer:

- el tamaño (cantidad de tuplas) resultante de JOIN1
- el tamaño resultante de JOIN2
- la cantidad de tuplas del resultado de JOIN2 que entran en un bloque (factor de bloqueo de JOIN2)

Para calcular a) necesitaríamos conocer la selectividad de JOIN1, o saber cuántas tuplas de la tabla Salones con cantidad de bancos >30 corresponden a liceos de Montevideo.

Para calcular b) necesitaríamos conocer la selectividad de JOIN2, o saber cuántos funcionarios están asignados a salones de más de 30 bancos de liceos de Montevideo.

Para calcular c), si no se tiene el dato del factor de bloqueo, éste se puede calcular conociendo el largo del bloque y el largo del registro resultante de JOIN2, en cantidad de bytes.

Aclaración: Para a) y b) se asume que el atributo nroLiceo de la tabla Salones es clave foránea hacia la tabla Liceos, donde éste es clave primaria, y los atributos (nroLiceo, nroSalon) de la tabla Funcionarios es clave foránea hacia la tabla Salones, donde el mismo par es clave primaria.