

# Segundo Parcial de Fundamentos de Base de Datos

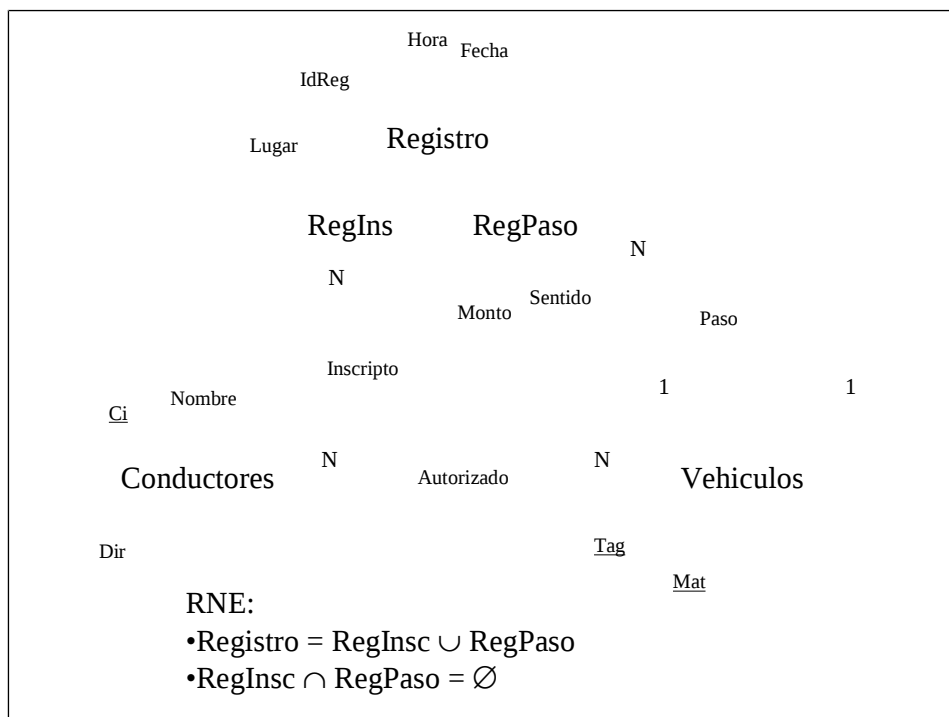
Noviembre 2011

Presentar la resolución del parcial:

- Duración: 3 horas
- 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.**

## Ejercicio 1 (17 puntos)

Una empresa que administra carreteras y peajes ha implementado un sistema llamado Telepague. En este sistema, a cada vehículo que se suscribe se le coloca un chip (llamado Tag) que es leído por antenas colocadas en cada senda del peaje y si tiene saldo, levanta la barrera correspondiente. Cada vehículo tiene un único Tag pero puede tener varios conductores autorizados. Cuando un usuario se registra en un peaje determinado, obtiene un descuento en las pasadas sólo en ese peaje. Esto hace que un usuario deba presentar el vehículo y registrarse en los diferentes peajes en los que desee descuento. Cada vez que se pasa por un peaje se registra el sentido de circulación y el saldo que queda en la cuenta. El siguiente MER representa una parte de la base de datos de ese sistema:



Se asume que este mer representa la realidad correctamente.

1. Considere la siguiente tabla que tiene todos los atributos que aparecen en el mer:  
**R(CI,Nombre,Dir,Tag,Mat,Fecha,Hora,Lugar,Monto,Sentido)**
  - a. Indique cuáles son las dependencias funcionales se cumplen sobre R correspondiente a ese Mer. Justifique por qué agrega cada dependencia.
  - b. Indique en qué forma normal se encuentra esa tabla. Justifique.
2. Alguien construyó las siguientes tablas en base a esa realidad.
 

**Telepague(CI,Lugar,Fecha,Hora,Tag)**  
**Conductores(CI,Nombre,Dir)**  
**Pasadas(Mat,Tag,Lugar,Fecha,Hora,Sentido,Monto)**

  - a. En qué forma normal está este esquema relacional con respecto al conjunto de dependencias funcionales de la parte 1.a.? Justifique.
  - b. Se cumple sobre la tabla Telepague la dependencia  $Tag \twoheadrightarrow CI$ ? Justifique.
3. Construya un esquema relacional a partir del Mer. Sobre cada tabla indique todas las claves, las dependencias funcionales y de inclusión. Indique en qué forma normal se encuentra este esquema relacional.

**SOLUCION:**

1.

a.

De cada atributo determinante, surge una dependencia funcional, de esta forma surgen:

$CI \rightarrow \text{Nombre,Dir}$

$Tag \rightarrow \text{Mat}$

$Mat \rightarrow \text{Tag}$

Por otra parte, la entidad Registro es en realidad la unión de dos entidades débiles. El punto importante es que en la determinación de la entidad débil debe participar uno de los determinantes de la entidad fuerte, por lo que las dependencias funcionales que surgen de esas entidades débiles son las siguientes:

$Fecha,Hora,Mat \rightarrow \text{Lugar,Monto,Sentido}$

Observar que a pesar que las entidades débiles son relaciones N:1 no generan dependencias funcionales.

b.

La tabla universal contiene es la siguiente:

**R(CI,Nombre,Dir,Tag,Mat,Fecha,Hora,Lugar,Monto,Sentido)**

Y las dependencias son las siguientes (las propuestas en la parte a):

$F = \{CI \rightarrow \text{Nombre,Dir} ; Tag \rightarrow \text{Mat} ; Mat \rightarrow \text{Tag} ; Fecha,Hora,Mat \rightarrow \text{Lugar,Monto,Sentido} \}$

Para determinar la forma normal es necesario determinar qué atributos son primos y cuáles no lo son. Para hacer esto, podemos comenzar a calcular las claves.

Los atributos que no aparecen a la derecha son: CI, Fecha,Hora. Estos atributos están en todas las claves. Ahora controlamos si constituyen por sí mismos una clave:  $\{CI,Fecha,Hora\}^+ = \{CI,Fecha,Hora,Nombre,Dir\}$

Con esta clausura nos damos cuenta que esos tres atributos no constituyen por sí solos una clave.

Los atributos que aparecen sólo a la derecha con Nombre,Dir,Lugar,Monto y Sentido.

Con la información que se tiene hasta el momento, estamos seguros que CI es primo pero no es clave y además que Nombre y Dir no son primos. Por esto que la dependencia  $CI \rightarrow \text{Nombre,Dir}$  induce una dependencia parcial de una clave sobre atributos no primos violando 2NF.

Conclusión: el esquema está en 1NF.

2.

a.

Para determinar en qué forma normal está una descomposición, el primer punto es proyectar las dependencias sobre el esa descomposición.

La descomposición en cuestión tiene las siguientes tablas:

**Telepague(CI,Lugar,Fecha,Hora,Tag)**

**Conductores(CI,Nombre,Dir)**

**Pasadas(Mat,Tag,Fecha,Hora,Sentido,Monto)**

Cálculo de las proyecciones sobre Telepague

Si observamos los atributos de Telepague, se puede ver que no hay dependencias que se proyecten directamente.

Sin embargo, si miramos las dependencias, nos encontramos que  $\text{Tag} \rightarrow \text{Mat}$  por lo que, aplicando pseudotransitividad con la última dependencia de F, obtenemos:

$\text{Fecha,Hora,Tag} \rightarrow \text{Lugar,Monto,Sentido}$

Por otro lado, el atributo CI sólo participa en en determinar Nombre y Dir que no determinan a ningún otro atributo y CI tampoco es determinado por ningún otro atributo por lo que no debiera participar en la proyección sobre esta tabla.

Dado que Mat y Tag son equivalentes de acuerdo a las dependencias y además no son determinados por ningún atributo, entonces no debiera haber más dependencias (no triviales) que la mencionada sobre esta tabla.

El resultado de este proceso es entonces:

$\Pi_{\text{Telepague}}(F) = \{\text{Fecha,Hora,Tag} \rightarrow \text{Lugar}\}$

La clave de Telepague es Fecha,Hora,CI por lo que la dep. Anterior induce una dependencia parcial de clave., dejando la tabla y todo el esquema relacional en 1NF.

Cálculo de las proyecciones sobre Conductores.

$\Pi_{\text{Conductores}}(F) = \{CI \rightarrow \text{Nombre,Dir}\}$  Dado que los atributos de la tabla no están involucrados en más dependencias en F, no pueden estar involucrados en más dependencias no triviales en F+. Por lo que no hay más dependencias que considerar en la proyección.

Dado que la dependencia tiene una clave del lado izquierdo, el esquema está en BCNF.

Cálculo de las proyecciones sobre Pasadas.

$\Pi_{\text{Pasadas}}(F) = \{\text{Tag} \rightarrow \text{Mat}; \text{Mat} \rightarrow \text{Tag}; \text{Fecha,Hora,Mat} \rightarrow \text{Monto,Sentido}\}$

La única dependencia que falta considerar es  $\text{Fecha,Hora,Mat} \rightarrow \text{Lugar}$  y como Lugar no determina ningún otro atributo, todas las dependencias de F+ que corresponde quedan cubiertas con las incluídas en el resultado.

Las claves de la tabla son Fecha,Hora,Mat y Fecha,Hora,Tag lo que hace que Mat y Tag sean atributos primos. Esto hace que no haya dependencias

que violen 3NF pero sí esas dos dependencias violan BCNF por lo que el esquema está en 3NF.

Conclusión: El esquema total está en 1NF.

**b.**

La tabla **Telepague(CI,Lugar,Fecha,Hora,Tag)** tiene atributos que surgen de diferentes conjuntos de entidades del mer, y através de diferentes relaciones. Hay que observar que la relación entre los conductores (CI) y los vehículos (Tag) es N:N e independiente de la relación entre cada Registro (Lugar,Fecha,Hora) y los vehiculos. (Esto es un primer indicio de que debería cumplirse alguna mvd.)

Esta tabla tiene incluídas de alguna forma esas dos relaciones por lo que, para cada Tag, deberían parecer todas las CI que tiene asociadas y para cada CI también tienen que estar todos los tags asociados. Por otra parte, para cada Tag deben aparecer SOLO los valores Lugar,Fecha,Hora en donde se hizo el registro. De esta forma, si hay dos tuplas como las siguientes

$c_1, l_1, f_1, h_1, t_1$

$c_1, l_2, f_2, h_2, t_2$

No tienen por qué haber otras tuplas que crucen los valores de tag con Lugar,Fecha,Hora dado que no todos los tags tienen por qué haber sido registrados en todos los peajes.

Por este motivo no se cumple la dependencia:  $Ci \twoheadrightarrow Tag$

Por otra parte, si se consideran las siguientes dos tuplas

$c_1, l_1, f_1, h_1, t_1$

$c_2, l_2, f_2, h_2, t_1$

Como para cada tag deben aparecer todos los conductores, entonces deben existir dos tuplas más con los siguientes valores.

$c_2, l_1, f_1, h_1, t_1$

$c_1, l_2, f_2, h_2, t_1$

Por este motivo, sí se cumple la dependencia  $Tag \twoheadrightarrow Ci$ .

**3.**

El punto más difícil de resolver es la categorización del Registro. Si se observa, el registro es en realidad la unión de dos conjuntos de entidades débiles que tienen el mismo determinante parcial. Esto hace que esa unión sea disjunta y total (Ver RNE). Por este motivo, parece más razonable, dentro de todas las opciones posibles, representar esas entidades y luego definir la supercategoría como una vista sobre las otras.

Así que la categorización nos queda con las siguientes tablas y restricciones:

$RegIns(\underline{Fecha, Hora, Tag}, Lugar)$

$Fecha, Hora, Tag \rightarrow Lugar$

$RegPaso(\underline{Fecha, Hora, Tag}, Lugar, Monto, Sentido)$

$Fecha, Hora, Tag \rightarrow Lugar, Monto, Sentido$

$Registro = RegIns \cup \Pi_{Fecha, Hora, Tag, Lugar}(RegPaso)$

El resto de las tablas no plantean mayores problemas a menos de las totalidades que se resuelven con las dependencias de inclusión:

$Conductores(Ci, dir, Nombre)$

$Ci \rightarrow dir, Nombre$

$Vehiculos(Tag, Mat)$

$Tag \rightarrow Mat$

$Mat \rightarrow Tag$

$Autorizados(\underline{Ci}, Mat)$

$\Pi_{Ci}(Autorizados) \subseteq \Pi_{Ci}(Conductores)$

$\Pi_{Ci}(Conductores) \subseteq \Pi_{Ci}(Autorizados)$

$\Pi_{Mat}(Autorizados) \subseteq \Pi_{Mat}(Vehiculos)$

$\Pi_{Mat}(Vehiculos) \subseteq \Pi_{Mat}(Autorizados)$

## Ejercicio 2 (15 puntos)

Dado el esquema relación R (A, B, C, D, E, G, H), y el conjunto de dependencias funcionales

$$F = \{ AB \rightarrow G, B \rightarrow GH, E \rightarrow CDH, BD \rightarrow E, G \rightarrow H \}$$

- Hallar todas las claves de R según F.
- Hallar un cubrimiento minimal de F, mostrando los pasos seguidos.
- Llevar R a 3NF con JSP y preservación de dependencias, justificando cada una de estas propiedades.
- Decir cuáles de los sub-esquemas obtenidos se encuentran en BCNF y cuáles no, justificando.
- Dar una descomposición de R distinta de la obtenida en la parte c), que no se encuentre en BCNF, justificando.
- Considere que se cumple la dependencia multivaluada embebida  $D \twoheadrightarrow E \mid B$ . Llevar la descomposición obtenida en la parte c) a 4NF, utilizando el algoritmo dado en el curso.

### Solución:

Dado el esquema relación R (A, B, C, D, E, G, H), y el conjunto de dependencias funcionales

$$F = \{ \begin{array}{l} AB \rightarrow G, \\ B \rightarrow GH, \\ E \rightarrow CDH, \\ BD \rightarrow E, \\ G \rightarrow H \end{array} \}$$

- Hallar todas las claves de R según F.

A,B no pertenecen a la derecha de las df  $\Rightarrow$  pertenecen a todas las claves.

$$(AB)^+ = \{A,B,G,H\} \neq R \Rightarrow (AB) \text{ no es clave.}$$

Se busca posibles claves con 3 atributos.

C,H solo pertenecen a los lados derechos de las df  $\Rightarrow$  no pertenecen a las claves.

Posibles claves con 3 atributos: (ABD), (ABE) y (ABG)

$$(ABD)^+ = \{ A,B,D,G,H,E,C \} = R \Rightarrow \mathbf{(ABD) \text{ es clave.}}$$

$$(ABE)^+ = \{ A,B,E,C,D,H,G \} = R \Rightarrow \mathbf{(ABE) \text{ es clave.}}$$

$$(ABG)^+ = \{ A,B,G,H \} \neq R \Rightarrow (ABG) \text{ NO es clave.}$$

Si hay más clave tiene que ser un subconjunto de atributos de R que cumpla las siguientes condiciones:

- Contener A y B
- No contener C
- No contener H
- No contener D

5) No contener E

No es posible obtener un subconjunto que cumpla estas condiciones  $\Rightarrow$  No hay más claves!

**Claves de R: ABD y ABE**

h) Hallar un cubrimiento minimal de F, mostrando los pasos seguidos.

$$F = \left\{ \begin{array}{l} AB \rightarrow G, \\ B \rightarrow GH, \\ E \rightarrow CDH, \\ BD \rightarrow E, \\ G \rightarrow H \end{array} \right\}$$

1) Obtener un conjunto de dependencias con un único atributo en el lado derecho: (aplicando descomposición).

$$F_1 = \left\{ \begin{array}{ll} AB \rightarrow G, & (1) \\ B \rightarrow G, & (2) \\ B \rightarrow H, & (3) \\ E \rightarrow C, & (4) \\ E \rightarrow D, & (5) \\ E \rightarrow H, & (6) \\ BD \rightarrow E, & (7) \\ G \rightarrow H & (8) \end{array} \right\}$$

2) Eliminación de atributos redundantes en el lado izquierdo de las df.

Se determina si en las df (1) y (7) hay atributos redundantes.

$(A)^+_{F_1} = \{ A \}$

$G \notin (A)^+_{F_1} \Rightarrow B$  no es redundante en la df (1)

$(B)^+_{F_1} = \{ B, G, H \}$

$G \in (B)^+_{F_1} \Rightarrow A$  es redundante en la df (1) – se elimina.

$E \notin (B)^+_{F_1} \Rightarrow D$  no es redundante en la df (7)

$(D)^+_{F_1} = \{ D \}$

$E \notin (D)^+_{F_1} \Rightarrow B$  no es redundante en la df (7)

$$F_2 = \left\{ \begin{array}{ll} B \rightarrow G, & (2) \\ B \rightarrow H, & (3) \\ E \rightarrow C, & (4) \\ E \rightarrow D, & (5) \\ E \rightarrow H, & (6) \\ BD \rightarrow E, & (7) \\ G \rightarrow H & (8) \end{array} \right\}$$

}

### 3) Eliminación de df redundantes

Las dependencias (2),(4),(5) y (7) no son redundantes por ser las únicas con sus respectivos lados derechos. (propiedad ej8 práctico 5)

a. Sea  $F_3 = F_2 - \{B \rightarrow H\}$   
 $F_3 \subseteq F_2 \Rightarrow (F_3)^+ \subseteq (F_2)^+$

Es necesario determinar si  $(F_2)^+ \subseteq (F_3)^+$

Por las definiciones de  $F_2$  y  $F_3$  es necesario determinar si  $B \rightarrow H \in (F_3)^+$

$$(B)^+_{F_3} = \{ B, G, H \}$$

$$H \in (B)^+_{F_3} \\ \Rightarrow B \rightarrow H \in (F_3)^+$$

$$\Rightarrow (F_2)^+ \subseteq (F_3)^+$$

$\Rightarrow F_3$  es equivalente a  $F_2$

Por lo tanto se continúa trabajando con  $F_3$ :

$$F_3 = \left\{ \begin{array}{ll} B \rightarrow G, & (2) \\ E \rightarrow C, & (4) \\ E \rightarrow D, & (5) \\ E \rightarrow H, & (6) \\ BD \rightarrow E, & (7) \\ G \rightarrow H & (8) \end{array} \right\}$$

En  $F_3$  las dependencias a verificar si son redundantes son las (6) y (8)

Para esto se procede como lo realizado en el caso anterior:

b. Sea  $F_4 = F_3 - \{E \rightarrow H\}$

$$(E)^+_{F_4} = \{E, C, D, \}$$

$$H \notin (E)^+_{F_4} \\ \Rightarrow E \rightarrow H \notin (F_4)^+$$

$\Rightarrow F_4$  NO es equivalente a  $F_3$

c. Sea  $F_5 = F_3 - \{G \rightarrow H\}$

$$(G)^+_{F_5} = \{G\}$$

$$H \notin (G)^+_{F_5} \\ \Rightarrow G \rightarrow H \notin (F_5)^+$$

$\Rightarrow F_5$  NO es equivalente a  $F_3$

No hay dependencias redundantes en  $F_3$ .

**Cubrimiento Minimal de F:**

$$F_m = \left\{ \begin{array}{l} B \rightarrow G, \\ E \rightarrow C, \\ E \rightarrow D, \\ E \rightarrow H, \\ BD \rightarrow E, \\ G \rightarrow H \end{array} \right\}$$

- i) Llevar R a 3NF con JSP y preservación de dependencias, justificando cada una de estas propiedades.

En esta parte se aplica el algoritmo visto en el curso que garantiza que el resultado del mismo es una descomposición del esquema original en 3NF con JSP y preservación de df por lo tanto de esta forma se garantizan las propiedades pedidas.

A partir del conjunto de df minimal:

$$F_m = \left\{ \begin{array}{l} B \rightarrow G, \\ E \rightarrow C, \\ E \rightarrow D, \\ E \rightarrow H, \\ BD \rightarrow E, \\ G \rightarrow H \end{array} \right\}$$

Se obtienen los siguientes esquemas:

R1(B,G)  
R2(E,C,D,H)  
R3(B,D,E)  
R4(G,H)

Como no hay una clave de R según F (ABD,ABE) incluidas en alguno de los esquemas anteriores se agrega un esquema que contenga una de las claves para garantizar el JSP :

R5(A,B,E)

Resultado:

$$\rho = \{ R1(B,G), R2(E,C,D,H), R3(B,D,E), R4(G,H), R5(A,B,E) \}$$

- j) Decir cuáles de los sub-esquemas obtenidos se encuentran en BCNF y cuáles no, justificando.

a. R1(B,G)

Es un esquema con 2 atributos  $\Rightarrow$  (por propiedad ej11 práctico 6)

**R1 está en BCNF.**



b. R2(E,C,D,H)

$$\Pi_{R2}(F) = \{ E \rightarrow C,D,H \}$$

$\Rightarrow E$  es clave de R2 según sus df  $\Rightarrow E$  es superclave  $\Rightarrow E \rightarrow CDH$  cumple con la condición de BCNF (todos los lados izquierdos de las df son superclave).  $\Rightarrow$

**R2 está en BCNF.**

c. R3(B,D,E)

$$\Pi_{R3}(F) = J = \{ E \rightarrow D, BD \rightarrow E \}$$

$E^+_J = \{E,D\}$  donde B no pertenece por lo tanto E no es clave  $\Rightarrow E$  no es superclave  $\Rightarrow E \rightarrow D$  no cumple la condición de BCNF.

**R3 no está en BCNF**

d. R4(G,H)

Es un esquema con 2 atributos  $\Rightarrow$  (por propiedad ej11 práctico 6)

**R4 está en BCNF.**

e. R5(A,B,E)

Es un esquema que solo contiene una clave de R según F  $\Rightarrow$  (por condición de minimal de una clave) no se proyectan df no triviales en este esquema  $\Rightarrow$

**R5 está en BCNF.**

k) Dar una descomposición de R distinta de la obtenida en la parte c), que no se encuentre en BCNF, justificando.

Una posible descomposición que cumpla estas condiciones es cualquier descomposición de R que incluya el esquema R3.

Por ejemplo:

$$\rho = \{ R1(A,B,C,G,H), R3(B,D,E) \}$$

l) Considere que se cumple la dependencia multivaluada embebida  $D \twoheadrightarrow E \mid B$ . Llevar la descomposición obtenida en la parte c) a 4NF, utilizando el algoritmo dado en el curso.

$$\rho = \{ R1(B,G), R2(E,C,D,H), R3(B,D,E), R4(G,H), R5(A,B,E) \}$$

La dependencia  $D \twoheadrightarrow E$  se cumple en el esquema R3.

En el resto de los esquemas no se cumplen dependencias multivaluadas no triviales y según lo verificado en la parte d) todos ellos están en BCNF por lo tanto están en 4NF.

$$\begin{aligned} &R3(B,D,E) \\ &BD \twoheadrightarrow E, \\ &E \rightarrow D \\ &D \twoheadrightarrow E \end{aligned}$$

D no es superclave en R3 por lo tanto la dependencia  $D \twoheadrightarrow E$  viola la condición de 4NF (todos los lados izquierdos de las dependencias multivaluadas deben ser superclaves). Se aplica el algoritmo visto en el curso que garantiza la obtención de una descomposición en 4NF con JSP:

R31(D,E)

R32(B,D)

Estos esquemas solo tienen dos atributos por lo tanto no se cumplen dependencias multivaluadas no triviales  $\Rightarrow$  estos esquemas están en 4NF.

Resultado:

$\rho = \{ R1(B,G), R2(E,C,D,H), R31(D,E), R32(B,D), R4(G,H), R5(A,B,E) \}$

### Ejercicio 3 (13 puntos)

La siguiente realidad corresponde a una peluquería femenina en la cual se registra la atención de sus clientas desde el momento en que abrió por primera vez. Las tablas relevantes de la base son 3:

**Cientas** (ciCli, nombre, nacionalidad, teléfono) : Contiene la información de las clientas de la peluquería.

**Peluqueras** (ciPel, nombre, teléfono, cargo): Contiene la información de todas las peluqueras que trabajaron en algún momento en la peluquería.

**Atencion** (ciPel, ciCli, fechaAtención): Contiene la información de cada vez que una peluquera atendió a una clienta.

Sea la siguiente consulta SQL:

```
SELECT      A.fechaAtencion
FROM        Clientas C, Peluqueras P, Atencion A
WHERE       A.ciPel = P.ciPel and A.ciCli = C.ciCli and
           C.nacionalidad = "uruguaya" and P.cargo = "peinadora";
```

Y se conoce la siguiente Información:

Relación R	Tamaño tupla (bytes)	bf <sub>R</sub>	n <sub>R</sub>	Atributos	Índices
Peluqueras P	200	100	1000	V(nombre, P) = 25 V(cargo, P) = 25 (distribución uniforme)	Índice Primario de un nivel sobre ciPel Índice secundario árbol b+ de 3 niveles sobre telefono, cargo.
Clientas C	80	250	2000	V(nacionalidad, C) = 25 (distribución uniforme)	Índice Primario de un nivel sobre ciCli Índice secundario árbol b+ de 3 niveles sobre nacionalidad, nombre.
Atención A	20	1000	50000	El 60% de las tuplas corresponden a la atención por parte de peluqueras de cargo 'peinadora', y el 30% de dichas atenciones fueron hechas a clientas uruguayas.	Índice Primario de un nivel sobre ciPel, ciCli, fechaAtención.

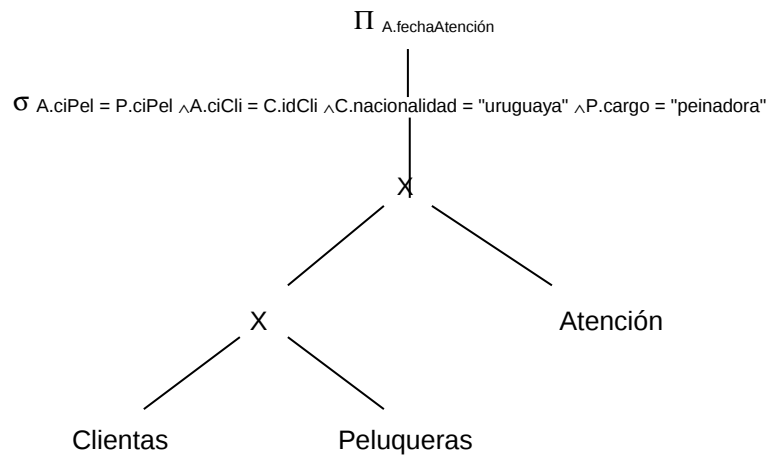
- Se dispone de 5 buffers de memoria.

**Se pide:**

- 1) Construir el árbol canónico de la consulta SQL dada.
- 2) Construir un plan lógico optimizado utilizando las heurísticas y teniendo en cuenta los tamaños si es necesario.
- 3) Calcular los tamaños de los resultados de cada operación del plan lógico construido.
- 4) Dar un plan físico correspondiente al plan lógico dado en la parte 2.

## SOLUCIÓN:

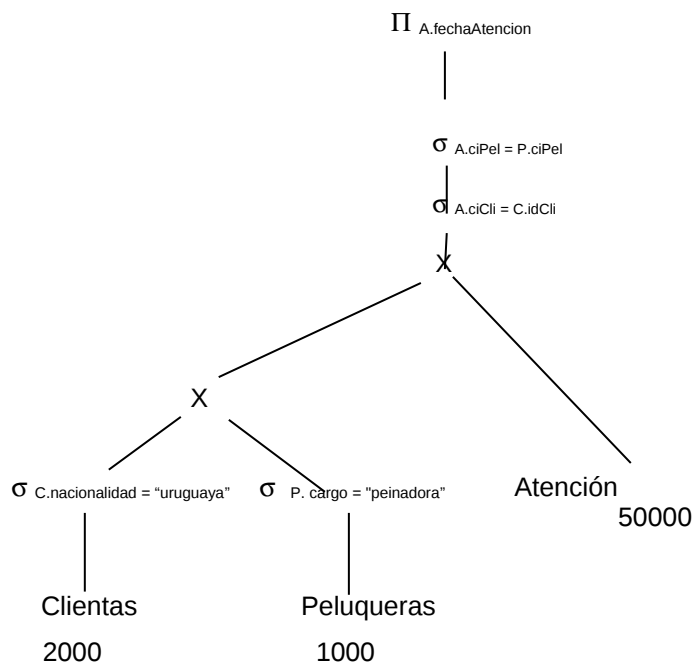
### 1) Construir el árbol canónico de la consulta SQL dada.



Se debe tener en cuenta que en el árbol canónico se respeta el orden de las tablas según se presentan en la consulta SQL. Además, el árbol siempre se recuesta sobre el lado izquierdo.

### 2) Construir el plan lógico utilizando las heurísticas y teniendo en cuenta los tamaños si es necesario.

En primer lugar, a partir del árbol canónico, se realiza una cascada de selecciones, bajando las selecciones lo más abajo posible:



Posteriormente, se deben reordenar las hojas según los tamaños de las mismas. Para esto se sabe que la tabla "Atención" tiene 50000 tuplas, la tabla "Clientas" tiene 2000 tuplas y la tabla "Peluqueras" tiene 1000 tuplas. Pero en el caso de las últimas 2 tablas se realiza una selección sobre las mismas, y se sabe que tanto el atributo "nombre" como para el atributo "cargo" tienen una distribución uniforme igual a 25.

Considerando los tamaños intermedios correspondientes a dichas selecciones se obtiene:

$$|P| * 1/V(\text{cargo}, P) = 40$$

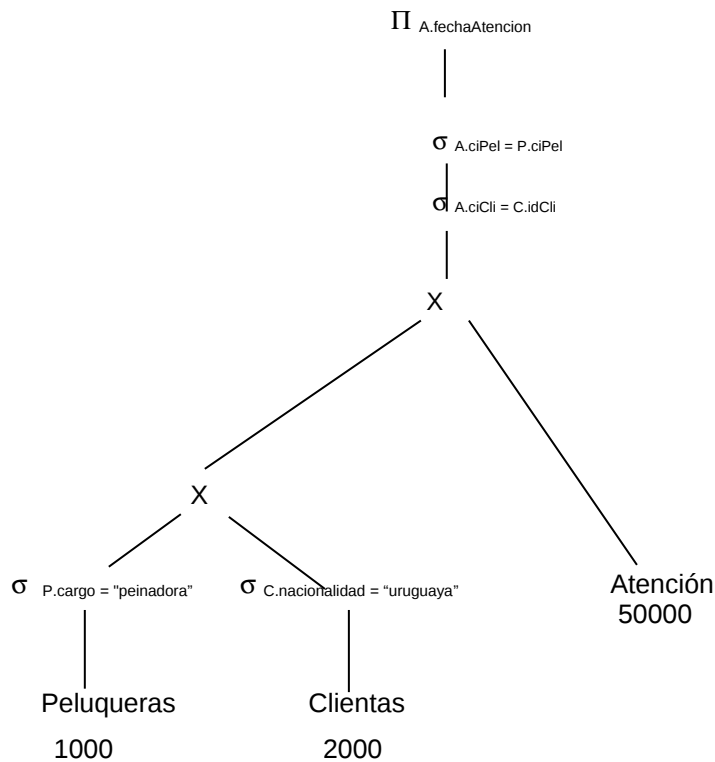
$$|C| * 1/V(\text{nacionalidad}, C) = 80$$

Finalmente, se deben comparar todas las hojas:

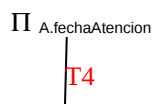
- la tabla "Atención", con 50000 tuplas (no tiene selección)
- la tabla "Clientas", con una selección que devuelve 80 tuplas
- la tabla "Peluqueras", con una selección que devuelve 40 tuplas

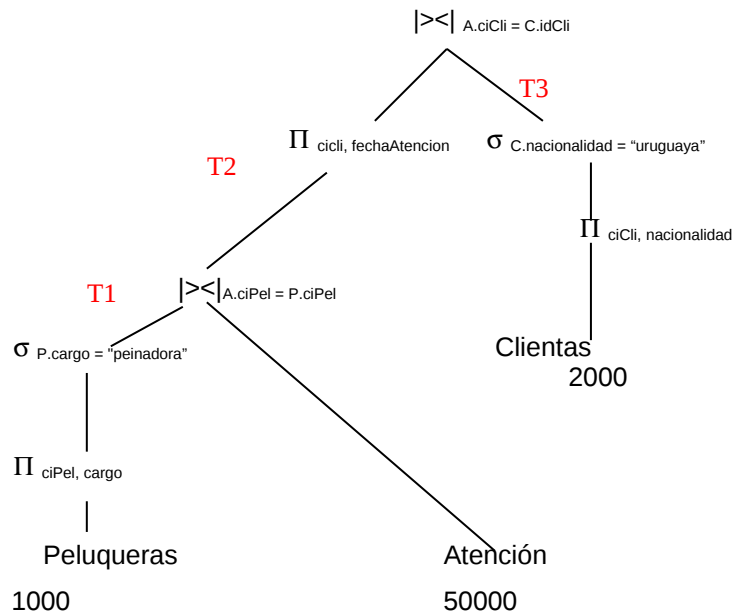
Por lo tanto, la hoja correspondiente a la tabla "Peluqueras" es la que debe ir más a la izquierda, ya que la selección  $\sigma_{P.\text{cargo} = \text{"peinadora"}}$  es la más restrictiva, es decir, genera la hoja con menor cantidad de tuplas.

Luego se deben dar vueltas las tablas, según lo considerando anteriormente, obteniendo el siguiente plan lógico:



La heurística sobre el cambio del orden de las hojas también dice que se deben evitar los productos cartesianos. Vemos que Peluqueras y Clientas quedarían como un producto cartesiano, ya que no se podría generar un join entre ellas con la heurística correspondiente. Por lo tanto, se debe realizar un nuevo cambio en el orden de las hojas. Como lo indican las heurísticas, en el siguiente paso se deben cambiar la secuencia de selecciones y productos cartesianos por joins.





Como se observa, en este caso también se debe tener en cuenta que el árbol correspondiente al plan lógico siempre se recuesta sobre el lado izquierdo. Por último, para obtener el plan lógico final, se bajan las proyecciones lo más abajo posible.

### 3) Calcular los tamaños de los resultados de cada operación del plan lógico construido.

$$T1 = \sigma_{P.cargo = "peinadora"} P$$

$$|T1| = |P| * 1/V(cargo, P) = 1000/25 = 40$$

$$T2 = T1 \underset{A.ciPel = P.ciPel}{\underset{A}{\text{join}}} A$$

Dado que el 60% de las tuplas en A corresponden a la atención dada por peluqueras con cargo "peinadora", éstas serán las tuplas resultados del join, porque ciPel es clave en A, por lo tanto, hay una tupla por cada ciPel de la tabla A.

$$|T2| = |A| * 0,6 = 50000 * 0,6 = 30000$$

$$T3 = \sigma_{C.nacionalidad = "uruguayaya"} C$$

$$|T3| = |C| * 1/V(nacionalidad, C) = 2000/25 = 80$$

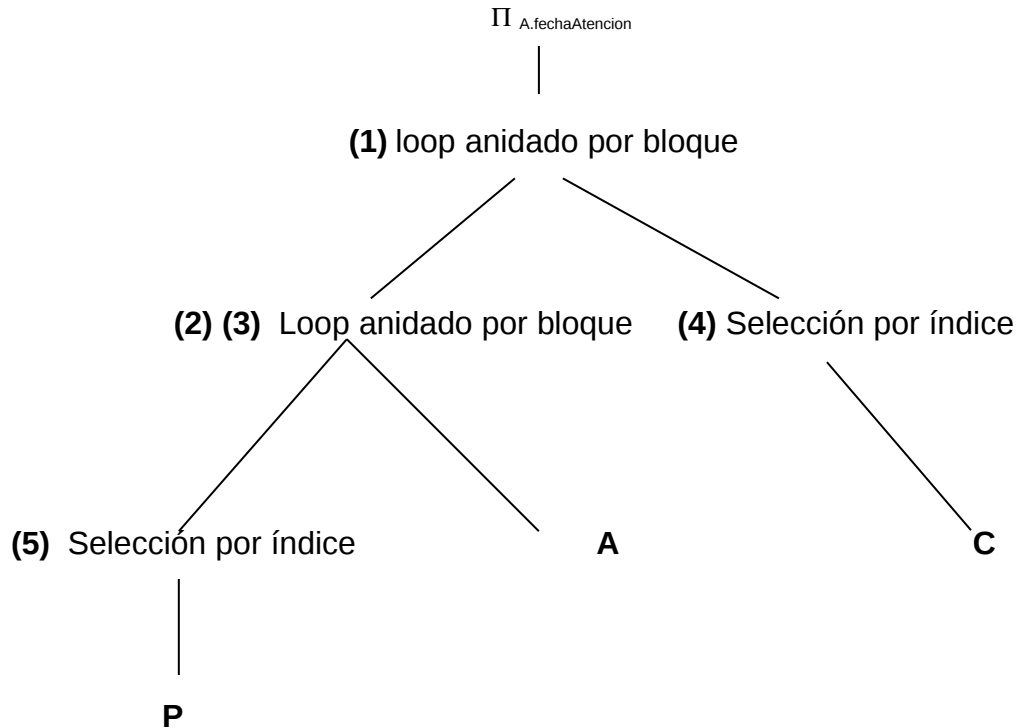
$$T4 = T3 \underset{A.ciCli = C.ciCli}{\underset{A}{\text{join}}} T2$$

El 30% de las clientas uruguayas fueron atendidas por "Adela Ruíz", por lo tanto, éstas serán el resultado del join.

$$|T4| = |T2| * 0,3 = 30000 * 0,3 = 9000$$

### 4) Dar un plan físico correspondiente al plan lógico dado en la parte 2.

En este caso, se debe tener en cuenta que siempre que se propone un plan físico, se deben considerar todos los datos dados en la letra para poder llegar al plan físico óptimo.



**(1)** No se puede utilizar index join porque con la selección se pierden todos los índices.

**(2)** Podría usarse un index join por CiPel, porque A tienen un índice primario sobre (CiPel, CiCli, FechaAtencion) por lo que los datos también están ordenados por CiPel.

**(3)** El loop anidado por bloque del join entre la selección de P y A se puede cambiar por un SORT-MERGE. La base de esto es que los datos de P están ordenados por CiPel por la existencia de un índice primario sobre ese atributo y los de A tienen un índice primario sobre (CiPel, CiCli, FechaAtencion) por lo que los datos también están ordenados por CiPel.

**(4)** Hay un índice secundario por el atributo “nacionalidad”

**(5)** Hay un índice secundario por el atributo “cargo”



## **Ejercicio 4 (15 puntos)**

Dadas las siguientes transacciones:

**T1: r1(X) w1(X) r1(Y) w1(X) w1(Y) c1**

**T2: r2(Y) w2(Y) r2(X) c2**

### **Parte 1)**

- a) Dar una historia de T1 y T2 entrelazada, que sea recuperable, justificando.

**r1(X) w1(X) r1(Y) r2(Y) w2(Y) r2(X) w1(X) w1(Y) c1 c2**

**Esta historia es recuperable porque T2 lee de T1, T1 no lee de T2, y T1 se confirma (hace commit) antes que T2.**

**T2 lee de T1 el elemento X (operaciones w1(X), r2(X) ).**

- b) ¿Qué ventaja tiene el hecho de que una historia sea recuperable?

**La ventaja es que si una transacción se cancela (hace abort) en vez de confirmarse, todas las transacciones que leyeron de ella pueden cancelarse también (abortos en cascada). Esto es porque ninguna transacción se habrá confirmado antes que otra de la cual leyó.**

- c) Dar una historia de T1 y T2 entrelazada, que Evite Abortos en Cascada, justificando.

**r2(Y) w2(Y) r1(X) r2(X) c2 w1(X) r1(Y) w1(X) w1(Y) c1**

**Esta historia Evita Abortos en Cascada porque T1 lee de T2 luego que T2 ya hizo commit, y no hay ninguna otra lectura de una transacción hacia otra.**

**T1 lee de T2 el elemento Y (operaciones w2(Y), r1(Y) ).**

**Parte 2)** Ahora considere a T1 y T2 con los siguientes bloqueos y desbloqueos:

**T1: r1(X) r1(X) w1(X) w1(X) r1(Y) r1(Y) w1(X) w1(Y) u1(X) w1(Y) u1(Y) c1**

**T2: r12(Y) r2(Y) w12(Y) w2(Y) r12(X) u2(Y) r2(X) u2(X) c2**

- a) Para cada una de las siguientes historias decir si puede suceder en el sistema o no, justificando.

**H1: r1(X) r1(X) w1(X) r12(Y) r2(Y) w12(Y) w2(Y) r12(X) u2(Y) r2(X) u2(X) c2 w1(X) r1(Y) r1(Y) w1(X) w1(Y) u1(X) w1(Y) u1(Y) c1**

**H1 no puede suceder en el sistema. La operación r12(X) no puede suceder en ese lugar, ya que T1 tiene bloqueado a X para escritura. T1 ejecutó w1(X), que es un bloqueo exclusivo, y no lo liberó antes de la operación r12(X).**

**H2: r12(Y) r2(Y) w12(Y) w2(Y) r12(X) r1(X) r1(X) u2(Y) r2(X) u2(X) c2 w1(X) w1(X) r1(Y) r1(Y) w1(X) w1(Y) u1(X) w1(Y) u1(Y) c1**

**H2 puede suceder en el sistema, ya que los bloqueos y desbloqueos son respetados en la historia.**

- b) Decir para cada una de las transacciones T1 y T2, si siguen el protocolo 2PL básico.

**T1 sigue el protocolo 2PL básico, ya que cumple con las 2 fases requeridas: primero aumenta sus bloqueos y luego los libera.**

**T2 sigue el protocolo 2PL básico, ya que cumple con la misma condición explicada para T1.**

- c) Para cada una de las historias posibles de la parte a) decir si es serializable, justificando.

**H2, que es la única historia posible de la parte a), es serializable, ya que por la parte c) sabemos que las transacciones que participan en ella siguen 2PL, y esto nos asegura seriabilidad para la historia.**