

## **FUNDAMENTOS DE BASES DE DATOS**

### **Examen Diciembre 2021**

**La duración del examen es de 3 horas.**

**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
- Justifique sus respuestas

### **Ejercicio 1 (25 puntos).**

Una veterinaria desea construir una base de datos con la información referida a las mascotas que atiende, sus dueños y los veterinarios que trabajan en la misma.

De las mascotas interesa saber un número de socio (que las identifica), su nombre, su tipo (por ejemplo perro, gato, roedor, etc), el color de su pelaje y su fecha de nacimiento.

Las personas relacionadas con la veterinaria pueden ser dueños o veterinarios. Se sabe que no hay otro tipo de persona y además que un veterinario puede ser dueño de alguna mascota. De las personas interesa saber su cédula, su nombre, sus teléfonos, su dirección compuesta por calle y número, y un email de contacto que es único. De los veterinarios interesa saber la fecha en la que obtuvieron el título y la universidad que se los otorgó. Además es de interés saber qué especialidades tienen.

Cada dueño tiene una o más mascotas a su cargo y todas las mascotas tienen por lo menos un dueño.

Es de interés registrar los tratamientos existentes. Cada tratamiento tiene un código que lo identifica, un nombre y un costo asociado.

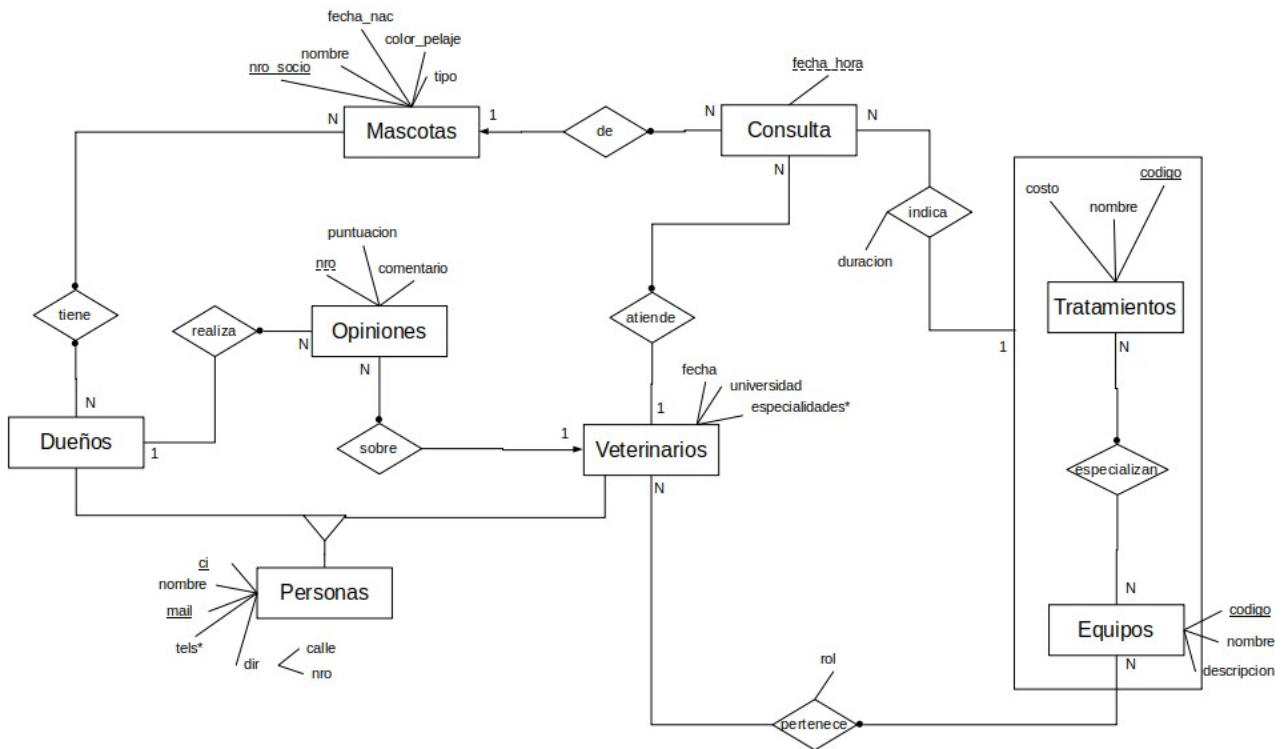
Los veterinarios trabajan en equipos especializados en realizar tratamientos, todos los tratamientos cuentan con al menos un equipo que lo realiza. Cada equipo tiene un código que lo identifica, un nombre y una descripción. Además se conocen los veterinarios que lo conforman. Es necesario registrar el rol que cumple cada veterinario en un equipo. Cada veterinario puede formar parte de múltiples equipos y todos los equipos están conformados por al menos un veterinario.

La veterinaria sólo funciona con agenda previa, y cada mascota tiene como máximo una consulta en cierto día y hora. Cuando una mascota va a una consulta agendada es atendida por un único veterinario, que no tiene por qué ser el mismo todas las veces. Interesa saber todas las veces que una mascota fue atendida. Por ejemplo la mascota "Luna" puede ser atendida por el Dr. Pérez el 01/03/2021 a las 14:00 y el 04/03/21 a las 15:30; y luego por la Dra. González el 01/03/21 a las 18:00.

Algunas de las veces que un veterinario atiende a una mascota le indica un tratamiento y el equipo que deberá realizarlo. Además se indica la duración del tratamiento. No es necesario que el veterinario que atendió a la mascota forme parte del equipo que realizará el tratamiento.

Los dueños pueden opinar sobre los veterinarios. Cada opinión de un veterinario tiene un número relativo al veterinario, que la identifica, una puntuación y un comentario. Cada dueño puede hacer múltiples opiniones de cualquiera de los veterinarios.

**Se pide:** Modelo Entidad-Relación completo del problema.



**RNE**

$$Dueños \cup Veterinarios = Personas$$

Un veterinario no puede realizar una opinión sobre sí mismo

$$(\forall d \in Dueños)(\forall o \in Opiniones)((d, o) \in realiza \rightarrow de(o) \neq d)$$

## Ejercicio 2 (30 puntos).

### Parte A)

En una agencia de viajes se registra la información de venta de pasajes a clientes, la cuál se describe a continuación en una versión simplificada de la realidad.

De los clientes se conoce su cédula de identidad (ci-cliente), nombre (nom-cliente), dirección (direccion), teléfono (telefono) y los sistemas de acumulación de millas (millas), que pueden ser varios. Además, se registran los destinos que vende la agencia, cuyos datos son la ciudad (que identifica al destino), país y continente, y por otro lado, las aerolíneas con las que trabaja, de las cuales se registra un código (cod-aero), un nombre (nom-aero), un contacto (contacto) y el país de origen (pais-aero).

La agencia quiere registrar las ventas de pasajes a los clientes, guardando además del cliente y el destino, la fecha de salida (fec-salida), la fecha de llegada (fec-llegada), la forma de pago (crédito o contado) y la aerolínea con la que viaja.

Un cliente en una fec-salida tiene asociada una única compra de pasaje.

Se utilizan los siguientes nombres de atributos:

ci-cliente	ciudad	contacto
nom-cliente	pais	pais-aero
direccion	continente	fec-salida
telefono	cod-aero	fec-llegada
millas	nom-aero	pago

### Se pide:

a) Deducir las dependencias funcionales que deberían cumplirse para esta realidad.

b) Dada la siguiente descomposición:

**R1** (ci-cliente, nom-cliente, direccion, telefono, millas, ciudad, fec-salida, fec-llegada, pago, cod-aero)

**R2** (ciudad, pais, continente)

**R3** (cod-aero, nom-aero, contacto, pais-aero)

Decir en qué forma normal se encuentra, justificando.

c) Si es posible, dar una multivaluada embebida que se cumpla en R1, justificando.

d) Llevar el esquema dado en la parte **b)** a 3NF, de manera que tenga JSP y preservación de dependencias. Justificar.

e) Identifica algún problema de redundancia en el esquema obtenido en la parte anterior? Si es así, explicarlo.

### Parte B)

Ahora considere el siguiente esquema relacional, junto con el conjunto de dependencias funcionales que se cumplen en él:

R (A, B, C, D, E, G)                      F = { AB → E, GE → B, G → D, A → G }

a) Hallar todas las claves, justificando.

b) Hallar una descomposición que se encuentre en BCNF y cumpla la propiedad de JSP. Justificar.

c) Verificar si la descomposición anterior preserva las dependencias funcionales. Justificar.

## SOLUCIÓN

### Parte A)

#### a)

ci-cliente → nom-cliente, direccion, telefono

ciudad → pais, continente

cod-aero → nom-aero, contacto, pais-aero

ci-cliente, fec-salida → fec-llegada, ciudad, pago, cod-aero

#### b)

Debemos proyectar las dependencias funcionales y encontrar las claves para cada relación.

**R1**(ci-cliente, nom-cliente, direccion, telefono, millas, ciudad, fec-salida, fec-llegada, pago, cod-aero)

$\Pi_{R1} = \{$  ci-cliente → nom-cliente, direccion, telefono  
ci-cliente, fec-salida → ciudad, fec-llegada, pago, cod-aero  $\}$

- ci-cliente y fec-salida aparecen solo a la izquierda, así que pertenecen a todas las claves.  
- millas no aparece ni a la izquierda ni a la derecha, por lo que también pertenece a todas las claves.

- nom-cliente, direccion, telefono, ciudad, fec-llegada, pago, cod-aero aparecen solo a la derecha, por lo que no pertenecen a ninguna clave.

$($ ci-cliente, fec-salida, millas $)^+ = \{$ ci-cliente, fec-salida, millas, nom\_cliente, direccion, telefono, fec-llegada, ciudad, pago, cod-aero $\}$

(ci-cliente, fec-salida, millas) es la única clave, ya que cualquier otra debería incluirla a ella.

La dependencia ci-cliente → nom-cliente viola 2NF porque ci-cliente es parte de una clave y nom-cliente es un atributo no primo.

**R1 está entonces en 1NF**, y como esta es la mínima forma normal posible, podemos afirmar que el esquema está en 1NF sin necesidad de hallar las formas normales de R2 y R3.

#### c)

ci-cliente →> millas | ciudad

Se cumple porque tanto las millas del cliente como las ciudades a las que viajó tienen muchos valores y son independientes entre sí.

Otra posible sería ci-cliente →> millas | fec-salida

#### d)

En R2 se proyecta solo la dependencia *ciudad* → *pais, continente*, por lo que ciudad es clave y entonces la única dependencia que hay no viola BCNF. R2 está entonces en BCNF.

En R3 se proyecta solo la dependencia  $cod-aero \rightarrow nom-aero, contacto, pais-aero$ , por lo que  $cod-aero$  es clave y entonces la única dependencia que hay no viola BCNF. R3 está entonces en BCNF.

Ya vimos que R1 está en 1NF. Usaremos el algoritmo de descomposición a 3NF que garantiza JSP y preservación de dependencias.

### Paso 1 – Hallar un cubrimiento minimal G

Teníamos las siguientes dependencias:

$ci-cliente \rightarrow nom-cliente, direccion, telefono$   
 $ci-cliente, fec-salida \rightarrow ciudad, fec-llegada, pago, cod-aero$

Dejando solo un atributo en el lado derecho de cada dependencia tenemos:

$ci-cliente \rightarrow nom-cliente$   
 $ci-cliente \rightarrow direccion$   
 $ci-cliente \rightarrow telefono$   
 $ci-cliente, fec-salida \rightarrow ciudad$   
 $ci-cliente, fec-salida \rightarrow fec-llegada$   
 $ci-cliente, fec-salida \rightarrow pago$   
 $ci-cliente, fec-salida \rightarrow cod-aero$

En las dependencias con dos atributos a la izquierda, ninguno de los atributos es redundante. Esto se comprueba haciendo la clausura de  $ci-cliente$  por un lado y verificando que no se llega a los atributos de la derecha, y luego lo mismo con la clausura de  $fec-salida$ .

Por otro lado, ninguna dependencia puede sacarse del conjunto dado que todas tienen un atributo diferente a la derecha.

### Paso 2 – Crear los nuevos esquemas

Creamos un nuevo esquema para cada parte izquierda de las dependencias funcionales

R11 ( $ci-cliente$ ,  $nom-cliente, direccion, telefono$ )

R12 ( $ci-cliente, fec-salida$ ,  $ciudad, fec-llegada, pago, cod-aero$ )

### Paso 3 – Crear un esquema con una clave

Habíamos hallado anteriormente que la clave de R1 es ( $ci-cliente, fec-salida, millas$ ). Como ninguno de los esquemas contiene a la clave, se agrega otro con ella:

R13 ( $ci-cliente, fec-salida, millas$ )

Como ninguno de los tres esquemas está contenido en otro, no es necesario eliminar ninguno.

La descomposición  $\{R11, R12, R13, R2, R3\}$  está en 3NF, tiene JSP y preserva dependencias.

Al esquema relacional obtenido podemos asignarle nombres mnemotécnicos que dan una mejor idea de la información que quedó en cada tabla:

Clientes (ci-cliente, nom-cliente, direccion, telefono)

Pasajes (ci-cliente, fec-salida, ciudad, fec-llegada, pago, cod-aero)

Clientes-Viajes-Millas (ci-cliente, fec-salida, millas)

Ciudades (ciudad, pais, continente)

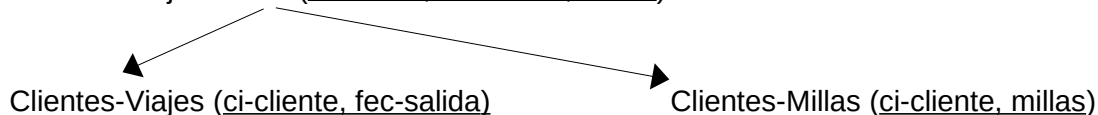
Aerolineas (cod-aero, nom-aero, contacto, pais-aero)

**d)**

Se genera redundancia en la tabla Clientes-Viajes-Millas, ya que se cumple la dependencia multivaluada  $ci-cliente \twoheadrightarrow millas$  (y su complemento  $ci-cliente \twoheadrightarrow fec-salida$ ). Entonces en esa tabla deberían existir las tuplas que combinan todos los valores de  $fec-salida$  con los de  $millas$ , generándose redundancia.

Para eliminar esta redundancia se debería llevar este esquema a 4NF, de la siguiente forma:

Clientes-Viajes-Millas (ci-cliente, fec-salida, millas)



Finalmente, la tabla Clientes-Viajes se debería eliminar por estar incluida en Pasajes, y el esquema final estaría en 4NF y sería el siguiente:

Clientes (ci-cliente, nom-cliente, direccion, telefono)

Pasajes (ci-cliente, fec-salida, ciudad, fec-llegada, pago, cod-aero)

Ciudades (ciudad, pais, continente)

Aerolineas (cod-aero, nom-aero, contacto, pais-aero)

Clientes-Millas (ci-cliente, millas)

**Parte B)**

R (A, B, C, D, E, G)

F = { AB → E, GE → B, G → D, A → G }

**a)**

Atributos que aparecen solo a la izquierda: {A}

Atributos que aparecen solo a la derecha: {D}

Atributos que no aparecen en ninguna dependencia funcional: {C}

Otros atributos: {B,G,E}

Entonces A y C pertenecen a todas las claves, D a ninguna y B,G,E pueden o no pertenecer a alguna clave.

(CA)<sup>+</sup> = {C, A, G, D}

Como CA no es clave, probamos agregando alguno de los tres atributos que aparecen de ambos lados:

(CAB)<sup>+</sup> = {C, A, B, G, D, E} es clave

(CAE)<sup>+</sup> = {C, A, G, E, B, D} es clave

Verifico si hay más claves: R – {D, B, E} = {A C G} contiene alguna clave?

(ACG)<sup>+</sup> = {A, C, G, D}

Las únicas claves son entonces (CAB) y (CAE).

**b)** Usaremos el algoritmo de descomposición en BCNF con JSP.

La dependencia G → D viola BCNF porque G no es superclave.

Descomponemos entonces a R:

R1(A,B,C,E,G)

Π<sub>R1</sub> = {AB → E, GE → B, A → G}

Claves: AB, AE

R2(G,D)

Π<sub>R2</sub> = {G → D}

Clave: G

R2 está en BCNF porque G es clave y la única dependencia que se proyecta tiene a G del lado izquierdo.

En R1 la dependencia GE → B viola BCNF porque GE no es superclave.

Descomponemos a R1:

R11(A,C,E,G)

Π<sub>R11</sub> = {AB → E, A → G}

Clave: AB

R12(G, E, B)

$\Pi_{R12} = \{GE \rightarrow B\}$

Clave: GE

R12 está en BCNF porque GE es clave y la única dependencia que se proyecta tiene a GE del lado izquierdo.

En R11 la dependencia  $A \rightarrow G$  viola BCNF porque A no es superclave.

Descomponemos a R11:

R111(A,C,E)

$\Pi_{R12} = \{\}$

Clave{A,C,E}

R112(A,G)

$\Pi_{R12} = \{A \rightarrow G\}$

Clave: A

R111 está en BCNF porque no se proyecta ninguna dependencia.

R112 está en BCNF porque A es superclave y la única dependencia que se proyecta tiene a A del lado izquierdo.

La siguiente descomposición está en BCNF y tiene JSP:

R2(G, D), R12(G, E, B), R111(A, C, E), R112(A, G)

c) Las dependencias proyectadas en la descomposición son las siguientes:

$GE \rightarrow B$

$A \rightarrow G$

$G \rightarrow D$

Falta  $AB \rightarrow E$ , y esta dependencia no se puede deducir de las 3 que tenemos porque E no aparece del lado derecho de ninguna. La descomposición entonces no preserva dependencias.



### Ejercicio 3 (25 puntos)

Una reconocida asociación de ingenieros nuclea a todos los profesionales a nivel mundial, con el objetivo de generar estándares. A continuación se presentan las tablas que permiten modelar los aspectos más significativos de esta realidad.

#### **CAPITULOS (cod-capitulo, nombre)**

Contiene los datos de los capítulos en los que se organiza la asociación. Se conoce el código que identifica al capítulo y su nombre (por ejemplo, "Caribe").

#### **CAPITULOS\_PAISES (cod-capitulo, cod-pais)**

Contiene los datos de los países que componen cada uno de los capítulos. Se conoce el código del capítulo y el código del país. Un país pertenece a un solo capítulo.

#### **SOCIOS (pasaporte, nombre, cod-pais, fecha-ingreso)**

Contiene los datos de los ingenieros que son miembros activos de la asociación. Se conoce su pasaporte, su nombre, el código del país de residencia y la fecha de alta a la asociación. Cada socio pertenece al capítulo al que pertenece su país de residencia.

#### **ESTANDARES(cod-estandar, nombre, tipo, fecha-inicio)**

Contiene los datos de los diferentes estándares en los que la asociación se encuentra trabajando. De cada estándar se tiene un código, que lo identifica, nombre, tipo y fecha en la que se iniciaron las actividades asociadas a la confección del estándar.

#### **REUNIONES(cod-reunion, cod-estandar, fecha)**

Contiene los datos de las reuniones que se realizan para trabajar sobre los estándares. Se registra el código de identifica a la reunión, el código del estándar que motiva la reunión.

#### **ASISTENCIA\_REUNIONES(cod-reunion, pasaporte)**

Contiene los datos de la asistencia de los socios a las reuniones. En este esquema no existen tablas vacías y se cumplen las siguientes relaciones de inclusión:

$$\begin{aligned} \Pi_{\text{cod-pais}}(\text{SOCIOS}) &\subseteq \Pi_{\text{cod-pais}}(\text{CAPITULOS\_PAISES}) \\ \Pi_{\text{cod-capitulo}}(\text{CAPITULOS\_PAISES}) &\subseteq \Pi_{\text{cod-capitulo}}(\text{CAPITULOS}) \\ \Pi_{\text{cod-estandar}}(\text{REUNIONES}) &\subseteq \Pi_{\text{cod-estandar}}(\text{ESTANDARES}) \\ \Pi_{\text{pasaporte}}(\text{ASISTENCIA\_REUNIONES}) &\subseteq \Pi_{\text{pasaporte}}(\text{SOCIOS}) \\ \Pi_{\text{cod-reunion}}(\text{ASISTENCIA\_REUNIONES}) &\subseteq \Pi_{\text{cod-reunion}}(\text{REUNIONES}) \\ \Pi_{\text{pasaporte-convocante}}(\text{REUNIONES}) &\subseteq \Pi_{\text{pasaporte}}(\text{SOCIOS}) \\ \Pi_{\text{pasaporte-secretario}}(\text{REUNIONES}) &\subseteq \Pi_{\text{pasaporte}}(\text{SOCIOS}) \\ \Pi_{\text{cod-estandar}}(\text{REUNIONES}) &\subseteq \Pi_{\text{cod-estandar}}(\text{ESTANDARES}) \end{aligned}$$

Resolver en álgebra relacional las siguientes consultas:

1) Nombre de los capítulos en los que al menos un socio ha participado en todas las reuniones de los estándares del tipo "Telecomunicaciones"

Resolver en cálculo relacional las siguientes consultas:

2) Nombre de los socios que se incorporaron a la asociación el mismo día que asistieron a una reunión y a partir de ese momento asistieron a todas las reuniones del mismo estándar que la reunión inicial.

Resolver en SQL, sin utilizar vistas ni sub-consultas en el FROM, la siguientes consulta:

3) Pasaporte y nombre de capítulo de los socios que han asistido a alguna reunión

4) Código y nombre de los capítulos que trabajan en mayor cantidad de estándares. Se considera que un capítulo trabaja en un estándar si algún socio perteneciente a dicho capítulo asiste a alguna reunión de ese estándar.

## SOLUCIÓN

1. Nombre de los capítulos de los que al menos un socio ha participado en todas las reuniones de los estándares del tipo "Telecomunicaciones"

Todas las reuniones de los estándares del tipo Telecomunicaciones

$A = \text{REUNIONES} * (\sigma_{\text{tipo} = \text{Telecomunicaciones}} \text{ESTANDARES})$

Los socios que han participado de todas las reuniones de A que tuvieron asistencia

$B = \text{SOCIOS} \% \prod_{\text{pasaporte}} (A * \text{ASISTENCIAS\_REUNIONES})$

Los capítulos a los cuales pertenecen los socios en B

$C = (B \bowtie_{\text{codPaisResidencia} = \text{codPais}} \text{CAPITULOS\_PAISES})$

Los nombres de los capítulos de C, siendo esta la solución de la consulta

$\text{SOL} = \prod_{\text{nombre}} (\prod_{\text{codcapitulo}} (C) * \text{CAPITULOS})$

2. Nombre de los socios que se incorporaron a la asociación el mismo día que asistieron a una reunión y a partir de ese momento asistieron a todas las reuniones del mismo estándar que la reunión inicial.

```
{ s.nombre / SOCIOS(s) ^
    (∃ r)(REUNIONES(r) ^ (∃ a)(ASISTENCIA_REUNIONES(a) ^ a.ci=s.ci ^
a.codreunion = c.codreunion ^ r.fecha = s.fechaIncorporacion ^
    (∀ r1)(REUNIONES(r1) ^ r1.codEstandar = c.codEstandar ^ r1.fecha >
r.fecha
        → (∃ a1)(ASISTENCIA_REUNIONES(a1) ^ a1.codreunion =
c1.codreunion ^ a1.pasaporte = s.pasaporte
        )
    )
    )
}
```

3. Pasaporte y nombre de capítulo de los socios que han asistido a alguna reunión

```
SELECT s1.pasaporte, c1.nombre
FROM
  Capitulo c1 NATURAL JOIN Capitulo_Paises p1
  INNER JOIN Socio s1 ON s1.codpaisResidencia = c1.codPais
  INNER JOIN Asistencia_Reunion A1 ON a1.pasaporte = s1.pasaporte
  INNER JOIN Reunion r1 ON r1.codReunion = a1.codReunion
```

4. Código y nombre de los capítulos que trabajan en mayor cantidad de estándares. Se considera que un capítulo trabaja en un estándar si algún socio perteneciente a dicho capítulo asiste a alguna reunión de ese estándar.

Tomamos como punto de partida la solución de la consulta 3, en la cual podemos obtener los socios que participaron de alguna reunión. La lógica de la consulta radica en contar la cantidad de asistencias a reuniones agrupados por estándar.

```
SELECT DISTINCT c1.codCapitulo, c1.nombre
FROM
  Capitulo c1 NATURAL JOIN Capitulo_Paises p1
  INNER JOIN Socio s1 ON s1.codpaisResidencia = c1.codPais
  INNER JOIN Asistencia_Reunion A1 ON a1.pasaporte = s1.pasaporte
  INNER JOIN Reunion r1 ON r1.codReunion = a1.codReunion
GROUP BY (c1.codCapitulo, c1.nombre, r1.codEstandar)
HAVING COUNT (*) =>
  (SELECT COUNT(*)
   FROM
     Asistencia_Reunion a2 NATURAL JOIN Reunion r2
     INNER JOIN Socio s2 ON s2.pasaporte = a2.pasaporte
     INNER JOIN Capitulo_Paises p2 ON p2.codPais = s2.codPaisResidencia
   WHERE
     p2.codCapitulo <> c1.codCapitulo
  GROUP BY (p2.codCapitulo, p2.nombre, r2.codEstandar))
```

### Ejercicio 4 (20 puntos)

Dado el siguiente esquema relacional:

**CLIENTES** (ci-cliente, nom-cliente, direccion, telefono)

**PASAJES** (ci-cliente, ciudad, fec-salida, fec-llegada, pago, cod-aero)

**AEROLINEAS** (cod-aero, nom-aero, contacto, pais-aero)

y la siguiente consulta:

```
SELECT C.nom-cliente, A.nom-aero
```

```
FROM CLIENTES C, PASAJES P, AEROLINEAS A
```

```
WHERE C.ci-cliente = P.ci-cliente AND P.cod-aero = A.cod-aero AND P.ciudad = 'Paris'
```

Sabiendo que todos los atributos tienen distribución uniforme (recuerde que  $V(A,T)$  es la cantidad de valores distintos que tiene un atributo A en una tabla T), y contando con la siguiente información:

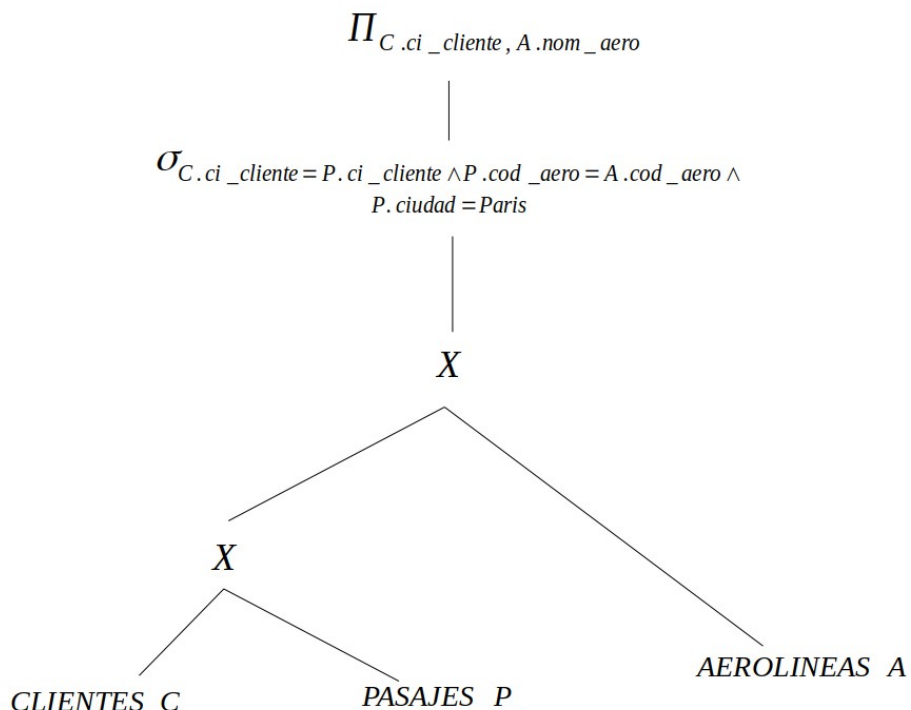
Relación R	$n_R$	Atributos	Índices
CLIENTES	800		Índice primario sobre ci-cliente
PASAJES	4500	$V(\text{ciudad}, P) = 10$	Índice primario sobre (ci-cliente, ciudad, fec-salida) Índice secundario sobre cod-aero
AEROLINEAS	300		Índice primario sobre cod-aero

**Se pide:**

1. Dar el árbol canónico para la consulta.
2. Aplicar las heurísticas para optimización llegando al plan lógico optimizado. Explique cada uno de los pasos ejecutados.
3. Dar un plan físico para el plan lógico obtenido en la **parte 2**, utilizando índices en los casos en que sea posible.

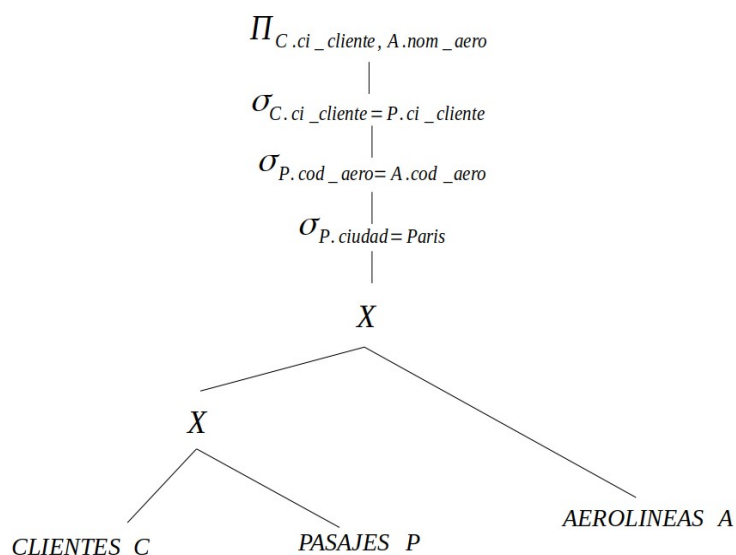
**SOLUCIÓN**

1. El árbol canónico para la consulta es el siguiente

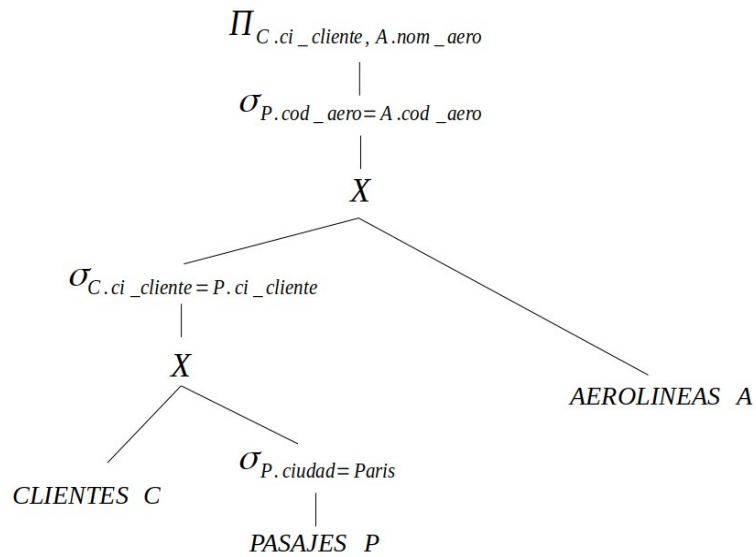


2. Aplicar las heurísticas para optimización llegando al plan lógico optimizado. Explique cada uno de los pasos ejecutados.

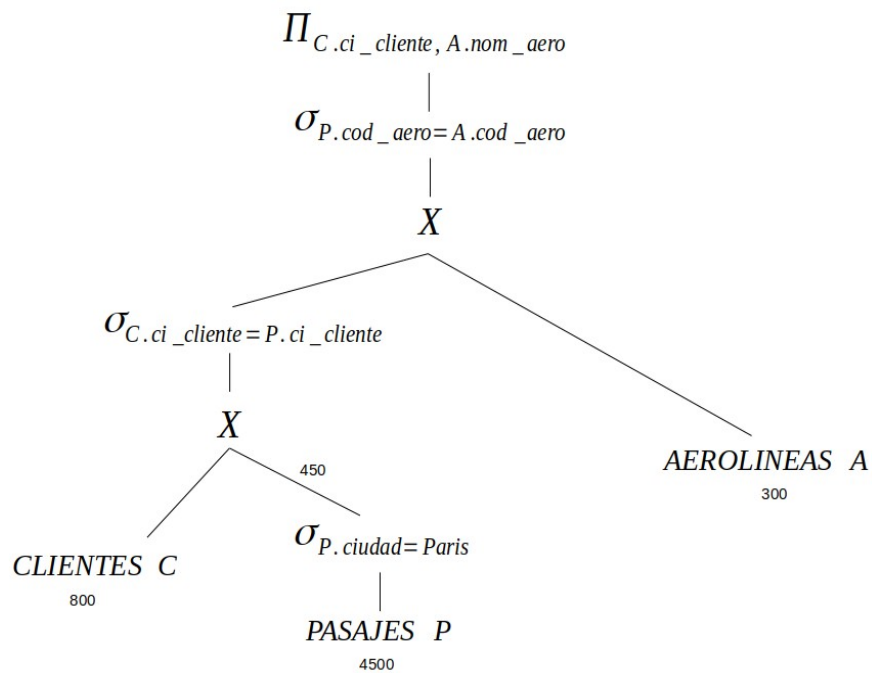
Paso 1: partiendo del árbol canónico, se separa la selección en una cascada de selecciones

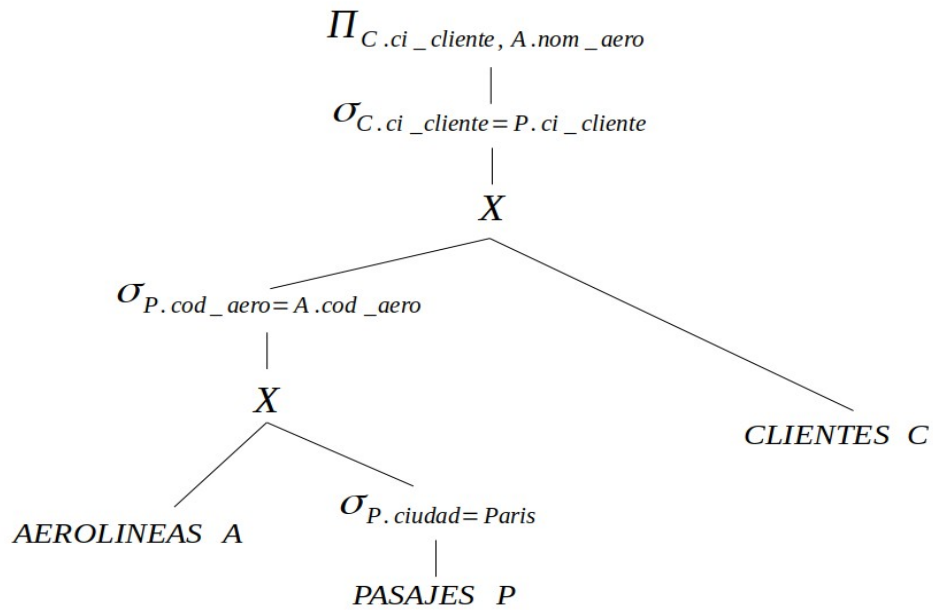


Paso 2: bajar las selecciones lo más posible en el árbol

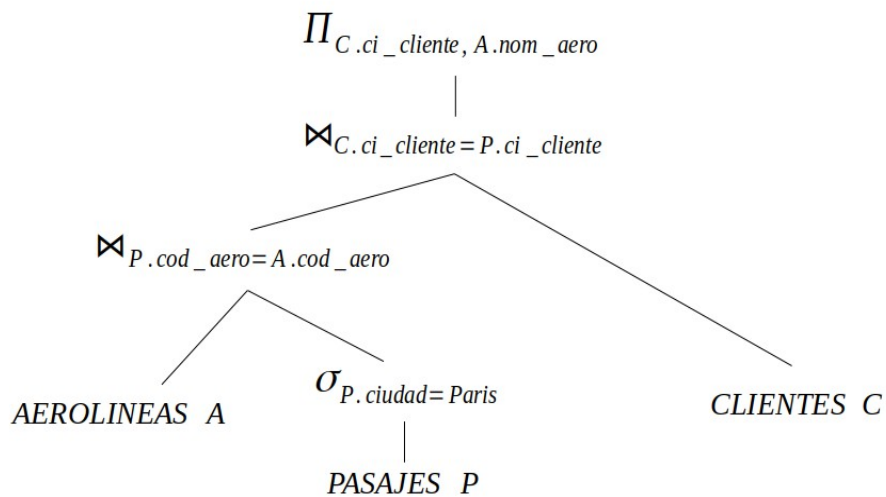


Paso 3: reordenar las hojas de forma de que la que tenga menor cantidad de tuplas quede a la izquierda sin generar productos cartesianos que no se puedan convertir en join.

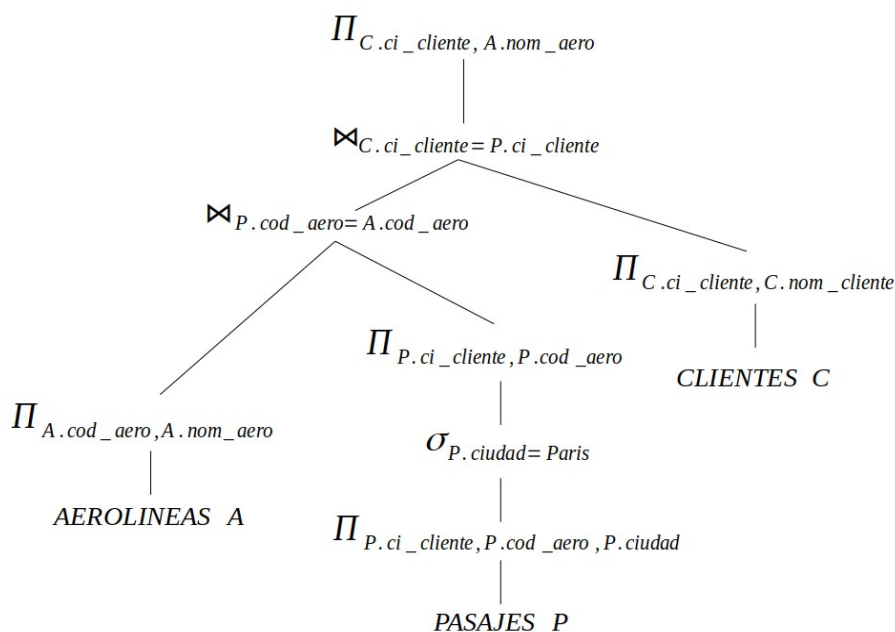




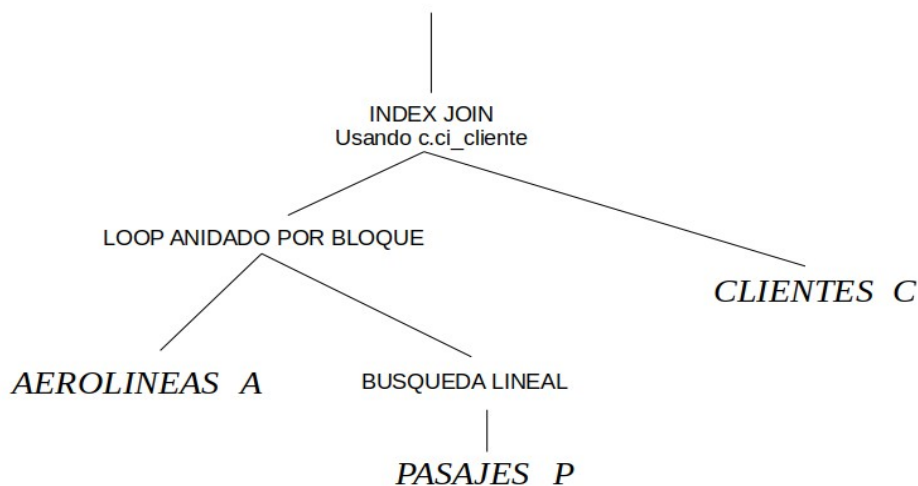
Paso 4: cambiar las selecciones y productos cartesianos por joins



Paso 5: agregar las proyecciones que sean necesarias para manejar la mínima cantidad de datos



3. Dar un plan físico para el plan lógico obtenido en la parte 2, utilizando índices en los casos en que sea posible.



La búsqueda lineal no se puede hacer mediante el índice primario porque ciudad no es el primer atributo.

El loop anidado por bloque no se puede realizar mediante un index join porque se pierden los índices de la tabla PASAJES luego de realizar la búsqueda.