

# **FUNDAMENTOS DE BASES DE DATOS**

## **SOLUCION Examen Febrero 2016**

**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

### **Ejercicio 1 ( puntos)**

Se desea modelar el sistema de administración de shoppings en Uruguay. Los shoppings cuentan con un conjunto de locales de los que se conoce nombre, identificador de local (único respecto al shopping), metros cuadrados que ocupa, piso en el que se encuentra y teléfonos. De los locales se registran los vendedores, de los que se conoce su cédula, nombre completo y correo electrónico. A su vez cada local está vinculado a una o varias categorías que se identifican por un código y un nombre (por ejemplo: el código "RES" corresponde al tipo "Restaurant"). Por último, de cada shopping se conoce su dirección y nombre que lo identifica.

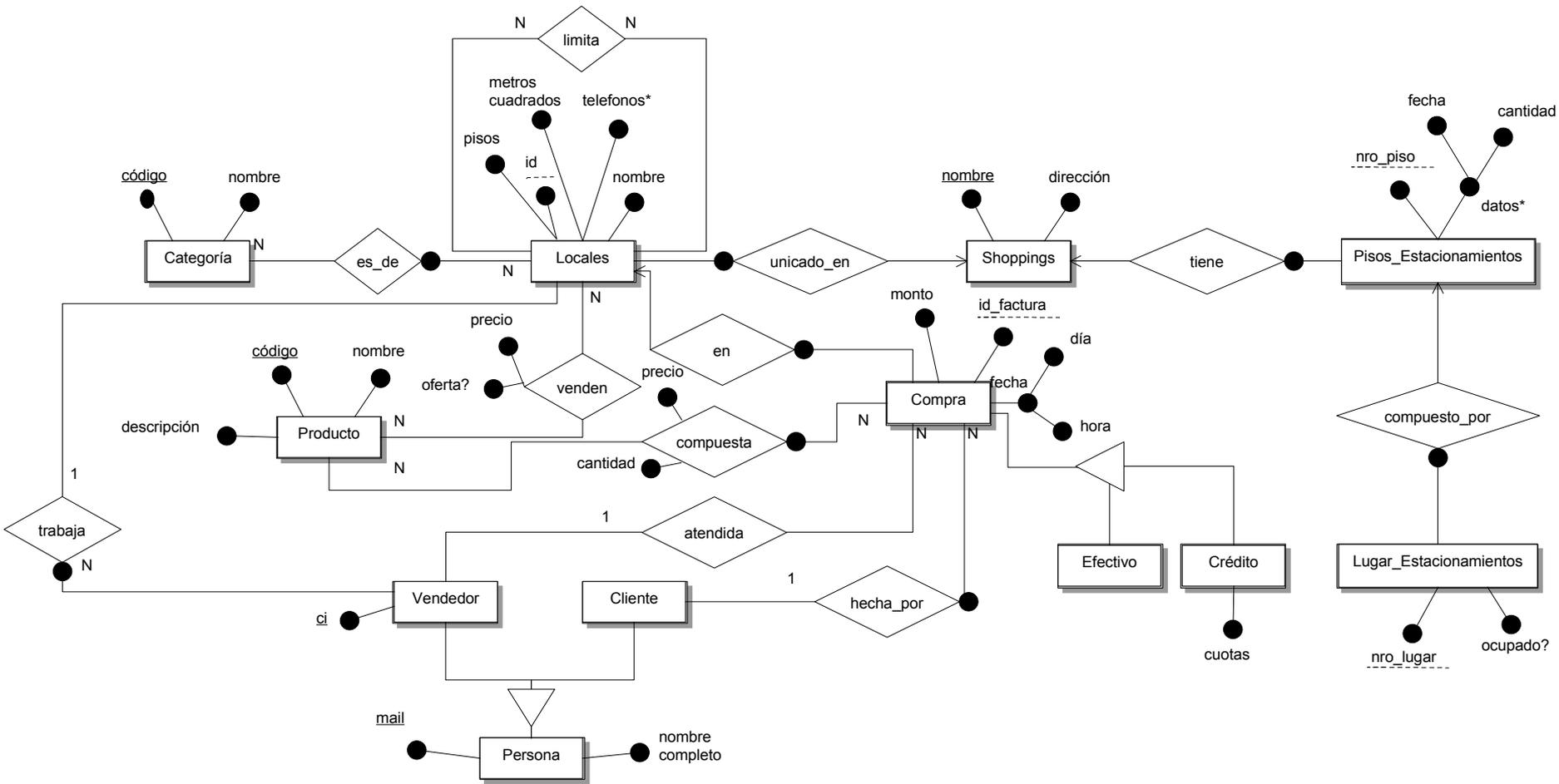
Interesa registrar los productos ofrecidos por los locales. De estos se cuenta con un código universal, nombre y descripción. Varios locales podrán ofrecer el mismo producto pero puede que no coincidan en el precio. También se quiere saber si el producto se encuentra en oferta para cada local.

De los clientes registrados en el sistema se conoce su nombre completo y correo electrónico (único en el sistema), de los cuales se registran las compras que realizan en los locales. Una compra puede ser realizada a través de un vendedor o a través de internet (por ejemplo, a un supermercado). Las compras por internet no tienen un vendedor asociado. De cada compra se conoce los productos comprados, junto a la cantidad y su precio unitario, fecha y hora y el número de factura que es único entre las facturas emitidas por el local. La compra puede ser pagada en efectivo (en tal caso interesa saber el monto total) o puede ser con tarjeta de crédito (en donde también interesa saber la cantidad de cuotas). Se asumirá que un cliente no puede realizar dos compras para una fecha y hora.

Con el fin de saber qué locales limitan entre sí, interesa registrar locales limítrofes. Existen locales que no limitan con ninguno llamado *islas*, locales que limitan solamente con otro local y locales que limitan con dos lugares. Se asume que no se puede limitar con más de 2 locales.

Por último se quiere llevar un registro de los estacionamientos pertenecientes a los shoppings. Los estacionamientos están compuestos por lugares identificados por el piso y el número de lugar. Con el nuevo sistema que automáticamente identifica que un vehículo está estacionado, interesa registrar si el lugar está libre u ocupado en ese momento. Se desea registrar la cantidad de autos que estacionan en cada hora por piso para identificar horarios picos y posibles mejoras en capacidad de estacionamiento.

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



RNE

1-  $(\forall c \in \text{Compra})(\forall c1 \in \text{Compra})(\forall cl \in \text{Cliente})$   
 $(\langle c, cl \rangle \in \text{Hecho\_por} \wedge \langle c1, cl \rangle \in \text{Hecho\_por} \rightarrow \text{fecha}(c) \neq \text{fecha}(c1))$

2- Limita es simétrica e irreflexiva

3-  $(\forall l \in \text{Locales}) |\{l1 \in \text{Locales} \wedge \langle l, l1 \rangle \in \text{Limita}\}| \leq 2$

4- Vendedores  $\cup$  Clientes = Personas

5- Efectivo  $\cup$  Crédito = Compras

6- Efectivo  $\cap$  Crédito =  $\emptyset$

## Ejercicio 2 ( puntos)

Una empresa dedicada a la administración de alquileres de propiedades en todo el país mantiene su información en una base de datos con las siguientes tablas:

### **SUCURSALES** (idSuc, depto, direccion, nomEncargado)

Contiene la información de todas las sucursales de la empresa. Para cada una de ellas se conoce un identificador, el departamento de Uruguay donde se encuentra, la dirección en el mismo y el nombre del encargado actual.

### **PROPIEDADES** (idProp, tipo, depto, direccion, CI, idSuc, fechaIngreso)

Contiene la información sobre las propiedades administradas. De cada una de ellas se conoce un identificador, el tipo de propiedad (apartamento, casa, campo etc.), el departamento de Uruguay donde se encuentra, la dirección en el mismo, la cédula de identidad del propietario y el identificador de la sucursal donde la propiedad ingresó a la empresa junto a la fecha en que ingresó.

### **CLIENTES** (CI, nombre, telefono)

Contiene los datos de los clientes de la empresa, tanto los propietarios como los que alquilan (inquilinos). Para cada uno de ellos se conoce su cédula de identidad, nombre y teléfono de contacto.

### **CONTRATOS** (idProp, CI, fechaInicio, fechaFin)

Contiene la información de los contratos de alquiler realizados en la empresa. De cada contrato de alquiler se conoce: el identificador de la propiedad que se alquila, la cédula de identidad del inquilino, y las fechas de inicio y fin del contrato de alquiler.

### **PAGOS** (idProp, CI, fechaInicio, idSuc, fechaPago, monto)

Contiene la información sobre los pagos que recibe la empresa por motivo de pagos de alquileres. Cualquier sucursal puede recibir pagos de contratos de alquiler de propiedades ingresadas en cualquier sucursal. De cada pago se conoce: la propiedad alquilada, el inquilino de la misma, la fecha de inicio del contrato, la sucursal que recibe el pago, la fecha en que se realiza el pago y el monto del mismo.

En este esquema no existen tablas vacías y se cumplen las siguientes relaciones de inclusión:

- $\prod_{idSuc} (PROPIEDADES) \subseteq \prod_{idSuc} (SUCURSALES)$
- $\prod_{CI} (PROPIEDADES) \subseteq \prod_{CI} (CLIENTES)$
- $\prod_{idProp} (CONTRATOS) \subseteq \prod_{idProp} (PROPIEDADES)$
- $\prod_{CI} (CONTRATOS) \subseteq \prod_{CI} (CLIENTES)$
- $\prod_{idProp, CI, fechaInicio} (PAGOS) \subseteq \prod_{idProp, CI, fechaInicio} (CONTRATOS)$
- $\prod_{idSuc} (PAGOS) \subseteq \prod_{idSuc} (SUCURSALES)$

### **Resolver en álgebra relacional las siguientes consultas:**

- a) Obtener el identificador de la propiedad, el tipo de propiedad y la fecha de ingreso de aquellas propiedades que ingresaron antes de 1/12/2010 y nunca fueron alquiladas.

Propiedades que ingresaron antes del 1/12/2010

$$A = \prod_{idProp} (\sigma_{fechaIng <= 1/12/2010} PROPIEDADES)$$

Propiedades que ingresaron antes del 1/12/2010 y no figuran en contratos

$$B = A - \prod_{idProp} (CONTRATOS)$$

$$SOL = \prod_{idProp, tipo, fechaIng} (PROPIEDADES * B)$$

### **Resolver en cálculo relacional las siguientes consultas:**

- b) Obtener CI, nombre y teléfono de los inquilinos que tienen contratos de alquiler sobre los cuales no se ha realizado ningún pago al 1/12/2015.

{ c.CI, c.nombre, c.telefono / CLIENTES(c)  $\wedge$

( $\exists$  co) (CONTRATOS (co)  $\wedge$  co.CI = c.CI  $\wedge$

$\neg$  ( $\exists$  p) (PAGOS (p)  $\wedge$  p.idProp = co.idProp  $\wedge$  p.iCI = co.CI  $\wedge$  p.fechaInicio = co.fechaInicio  $\wedge$  p.fechaPago  $\leq$  1/12/2015) ) }

- c) Obtener el identificador de sucursal y el nombre del encargado de aquellas sucursales que sólo han recibido pagos de propiedades que ingresaron a la empresa en esa sucursal.

$$\{ s.\text{idSuc}, s.\text{nomEncargado} / \text{SUCURSALES}(s) \wedge$$

$$(\forall p) (\text{PAGOS}(p) \wedge p.\text{idSuc} = s.\text{idSuc} \rightarrow$$

$$(\exists pr) (\text{PROPIEDADES}(pr) \wedge pr.\text{idProp} = p.\text{idProp} \wedge pr.\text{idSuc} = s.\text{idSuc})) \wedge$$

$$(\exists p1) (\text{PAGOS}(p1) \wedge p1.\text{idSuc} = s.\text{idSuc}) \}$$

**Resolver en SQL, sin utilizar vistas ni sub-consultas en el FROM, las siguientes consultas:**

- d) Para cada sucursal y cada propiedad ingresada en la misma obtener idSuc, idProp y el monto total de pagos realizados por alquileres de esa propiedad (independientemente de la sucursal donde se realizó el pago) considerando únicamente las propiedades que han tenido por lo menos 4 inquilinos distintos.

```
SELECT pro.idSuc, pro.idProp, SUM(pa.monto)
FROM PROPIEDADES pro, PAGOS pa
WHERE pro.idProp = pa.idProp AND
      pro.idprop IN
      (SELECT idProp
       FROM CONTRATOS
       GROUP BY idprop
       HAVING count(distinct CI) >= 4
      )
GROUP BY pro.idSuc, pro.idProp
```

- e) Obtener para cada propiedad que alguna vez fue alquilada, el identificador de la propiedad y el nombre y teléfono de su último inquilino.

```
SELECT c.idProp, cli.nombre, cli.telefono
FROM CONTRATOS c NATURAL JOIN CLIENTES cli
WHERE NOT EXISTS (SELECT *
                  FROM CONTRATOS c1
                  WHERE c1.idProp = c.idProp AND c1.fechaInicio > c.fechaInicio)
```

### Ejercicio 3 ( puntos)

Sea el esquema relacional  $R(A,B,C,D,E,G)$  y el conjunto de dependencias funcionales  $F$ :

$$F = \{ A \rightarrow CD, E \rightarrow B, DA \rightarrow GE, BC \rightarrow A \}$$

- a) Determine si las siguientes afirmaciones son verdaderas o falsas. Justifique su respuesta.
- 1) ACEB es superclave de  $R$  según  $F$ .
  - 2)  $G$  es un atributo primo
  - 3) Existe una clave de  $R$ :  $X$  tal que  $ACEB \cap X = \emptyset$
  - 4)  $R$  tiene una única clave según  $F$ .
- b) Obtener una descomposición de  $R$  en 3NF con join sin pérdida y preservación de dependencias aplicando el algoritmo visto en curso.
- c) Sea la siguiente descomposición de  $R$ ,  $\rho = (R_1, R_2)$  donde  $R_1(A,C,D,G)$  y  $R_2(E,C,B,G)$
- 1) Determinar si  $\rho$  tiene join sin pérdida con respecto a  $F$ . Justificar la respuesta.
  - 2) Determine la máxima forma normal en que se encuentra  $R_2$  según  $F$ . Justifique la respuesta.
  - 3) Sea el conjunto de dependencias  $J = F \cup \{ G \twoheadrightarrow B \mid CE \}$ .  
Obtener una descomposición de  $R_2$  en 4NF según  $J$  con join sin pérdida.

#### Solución:

Sea el esquema relacional  $R(A,B,C,D,E,G)$  y el conjunto de dependencias funcionales  $F$ :

$$F = \{ A \rightarrow CD, E \rightarrow B, DA \rightarrow GE, BC \rightarrow A \}$$

- a) Determine si las siguientes afirmaciones son verdaderas o falsas. Justifique su respuesta.
- 1) ACEB es superclave de  $R$  según  $F$ .  
ACEB es superclave sii  $(ACEB)^+ = R$   
 $(ACEB)^+ = \{A,C,E,B,D,G\} = R$   
**VERDADERA**
  - 2)  $G$  es un atributo primo  
 $G$  es un atributo primo sii pertenece a alguna clave de  $R$  según  $F$ .  
 $G$  no pertenece a los lados izquierdos de las dependencias funcionales, por lo tanto no pertenece a ninguna clave.  
**FALSA**
  - 3) Existe una clave de  $R$ :  $X$  tal que  $ACEB \cap X = \emptyset$   
El máximo conjunto  $X$  posible es  $R - \{A,C,E,B\} = \{D,G\}$ .  
Para que  $DG$  sea clave debe ser superclave minimal.  
 $(DG)^+ = \{D,G\} \neq R$ , o sea no es superclave.  
**FALSA**
  - 4)  $R$  tiene una única clave según  $F$ .  
Por a.1) sabemos que ACEB es superclave, por lo tanto contiene una clave.  
 $A^+ = \{A,C,D,G,E,B\} = R$   
 $A$  es superclave y mínima por lo tanto es clave.  
Si existen más claves  $R - \{A\}$  debe ser superclave.  
 $R - \{A\} = \{B,C,D,E,G\}$   
 $(BCDEG)^+ = \{B,C,D,E,G,A\} = R$   
Por lo tanto existen más claves  
**FALSA**
- b) Obtener una descomposición de  $R$  en 3NF con join sin pérdida y preservación de dependencias aplicando el algoritmo visto en curso.

El primer paso a realizar es obtener un cubrimiento minimal de  $F$ .

- b.I) Los lados derechos de las dependencias deben tener un único atributo.  
 $F_1 = \{ A \rightarrow C, A \rightarrow D, E \rightarrow B, DA \rightarrow G, DA \rightarrow E, BC \twoheadrightarrow A \}$
- b.II) Eliminación de atributos redundantes en los lados izquierdos de las dependencias.

Es necesario analizar  $DA \rightarrow G, DA \rightarrow E, BC \rightarrow A$

Por a.4) sabemos que  $A$  es clave de  $R$  según  $F$ , por lo tanto también lo es según  $F_1$  ( $F$  y  $F_1$  son equivalentes).

Por lo tanto  $\{G,E\} \subseteq A^+$  (según  $F_1$ ), o sea que  $D$  es redundante tanto en  $DA \rightarrow G$  como en  $DA \rightarrow E$ .

$B^+ = \{B\}$ ,  $A \notin B^+$ ,  $\Rightarrow C$  no es redundante en  $BC \rightarrow A$

$C^+ = \{C\}$ ,  $A \notin C^+$ ,  $\Rightarrow B$  no es redundante en  $BC \rightarrow A$

$F2 = \{A \rightarrow C, A \rightarrow D, E \rightarrow B, A \rightarrow G, A \rightarrow E, BC \rightarrow A\}$

- b.III) Eliminación de dependencias redundantes.

En los lados derechos de las dependencias no hay atributos repetidos  $\Rightarrow$  No hay dependencias redundantes.

F2 es un cubrimiento minimal de F.

Construyo esquemas según el algoritmo:

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

R2 (E,B)

R3 (B,C,A)

A es clave y pertenece a R1 (también a R3) por lo tanto no es necesario agregar más esquemas para garantizar el JSP.

La descomposición en R1,R2,R3 es una descomposición de R en 3NF con JSP y preservación de dependencias.

- c) Sea la siguiente descomposición de R,  $\rho = (R1,R2)$  donde R1(A,C,D,G) y R2(E,C,B,G)

- 1) Determinar si  $\rho$  tiene join sin pérdida con respecto a F. Justificar la respuesta.

Como la descomposición es en 2 subesquemas es posible aplicar el teorema de JSP que dice que si  $R1 \cap R2 \rightarrow R1 - R2 \in F^+$  o  $R1 \cap R2 \rightarrow R2 - R1 \in F^+$  entonces la descomposición es con JSP.

$R1 \cap R2 = \{C,G\}$

$R1 - R2 = \{A,D\}$

$R2 - R1 = \{E,B\}$

$(CG)^+ = \{C,G\}$

$\{A,D\}$  no está contenido en  $(CG)^+$  entonces no se cumple  $CG \rightarrow AD$  en  $F^+$

$\{E,B\}$  no está contenido en  $(CG)^+$  entonces no se cumple  $CG \rightarrow EB$  en  $F^+$

**La descomposición es con join con pérdida.**

- 2) Determine la máxima forma normal en que se encuentra R2 según F. Justifique la respuesta.

R2(E,C,B,G)

Las dependencias que se proyectan en este esquema son:  $\{E \rightarrow B, BC \rightarrow G\}$

$E \rightarrow B$  se proyecta en forma directa y  $(BC)^+ = \{B,C,A,D,E,G\}$

C pertenece a todas las claves de R2 por no pertenecer al lado derecho de las dependencias que en él se cumplen.

$C^+ = \{C\} \neq R2$ , por lo tanto no es clave.

$(BC)^+ = R2$ , o sea BC es una clave.

$BC \rightarrow G$  cumple con las condiciones de BCNF ya que el lado izquierdo es superclave.

$E^+ = \{E,B\} \neq R2$ , por lo tanto E no es superclave,  $E \rightarrow B$  no cumple las condiciones de BCNF.

B es un atributo primo (pertenece a la clave BC) entonces  $E \rightarrow B$  cumple las condiciones de 3NF.

**R2 está en 3NF.**

- 3) Sea el conjunto de dependencias  $J = F \cup \{G \twoheadrightarrow B \mid CE\}$ .

Obtener una descomposición de R2 en 4NF según J con join sin pérdida.

El conjunto de dependencias de J que se proyectan en R2 son:  $\{E \rightarrow B, BC \rightarrow G, G \twoheadrightarrow B\}$

$G^+ = \{G\} \neq R2$ , por lo tanto G no es superclave y  $G \twoheadrightarrow B$  no cumple las condiciones de 4NF.

Aplicando el algoritmo visto en el curso se descompone R2 en:

R21(G,B)

R22 (E,C,G)

R21 se encuentra en 4NF por ser un esquema con 2 atributos.

En R22 se proyecta la dependencia  $EC \rightarrow G$ , EC es superclave y no hay dependencias multivaluadas no triviales, por lo tanto R22 está en 4NF.

**La descomposición buscada es R21(G,B), R22 (E,C,G).**

## Ejercicio 4 ( puntos)

Dada la realidad que se presenta en el ejercicio de consultas se consideran las siguientes tablas:

**PROPIEDADES** (idProp, tipo, depto, direccion, CI, idSuc, fechaIngreso)

**CLIENTES** (CI, nombre, telefono)

**CONTRATOS** (idProp, CI, fechaInicio, fechaFin)

Se cuenta además con la siguiente información:

- Los contratos de las propiedades se han realizado uniformemente en el tiempo, esto significa que los contratos se distribuyen uniformemente para cada fecha de inicio registrada y hay 50 fechas de inicio distintas.
- El 80% de las propiedades son de tipo "apartamento".
- El tamaño de las tablas es el siguiente:
  - Propiedades: 800 tuplas
- Clientes: 350 tuplas
  - Contratos: 50000 tuplas
- Los índices existentes son los siguientes:

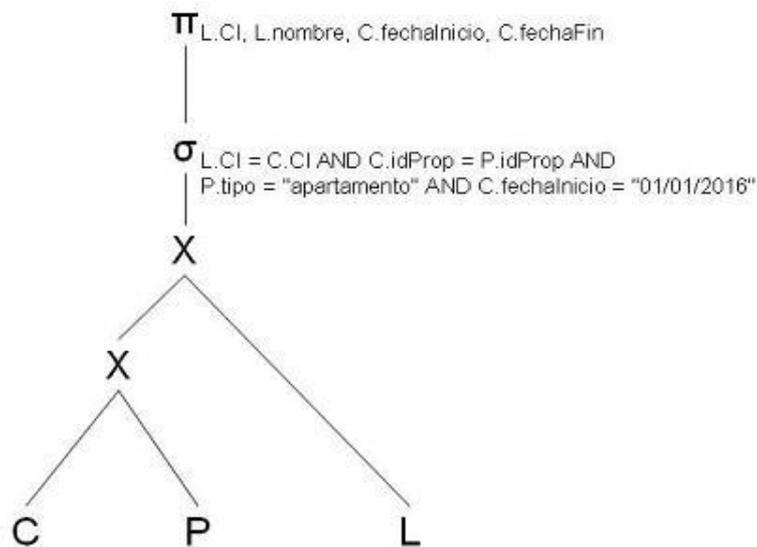
Indice	Tabla: atributo	Tipo
indIdProp	Propiedades: idProp	Primario
indTipo	Propiedades: tipo	Secundario
indCI	Clientes: CI	Primario
indFechaInicio	Contratos: fechaInicio	Secundario

Considere la siguiente consulta sobre el esquema dado:

```
SELECT L.CI, L.nombre, C.fechaInicio, C.fechaFin  
FROM Contratos C, Propiedades P, Clientes L  
WHERE L.CI = C.CI AND C.idProp = P.idProp AND  
P.tipo = "apartamento" AND C.fechaInicio = "01/01/2016";
```

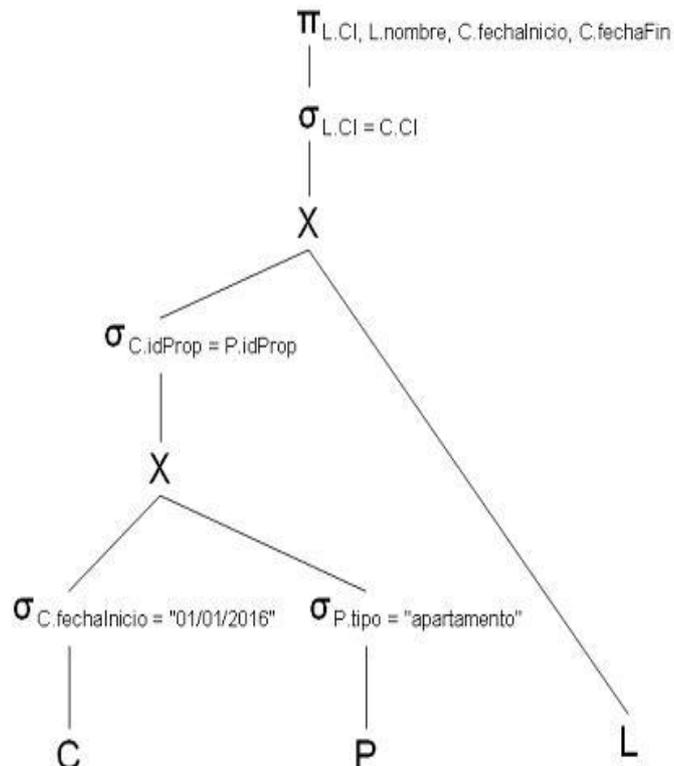
Se pide:

a) Dar el árbol canónico del plan lógico para la consulta dada.



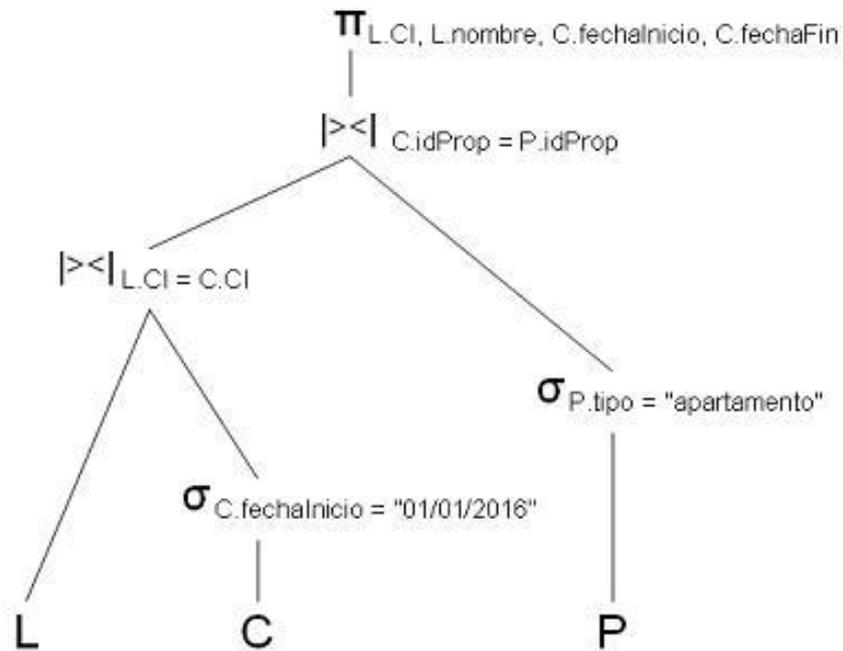
b) Dar un plan lógico para la consulta, optimizado mediante las heurísticas vistas en el curso. Calcular los tamaños que sean necesarios para poder aplicar las heurísticas correctamente. Justifique cada uno de los pasos.

A partir del árbol canónico, se sustituye la conjunción de selecciones por una cascada de selecciones, bajando las mismas lo máximo posible.



A partir del árbol anterior se cambia el orden de las hojas para que los lados izquierdos sean los más chicos. En este caso no sólo se deben tener en cuenta la cantidad de tuplas de cada tabla, sino también la cantidad de tuplas obtenidas luego de aplicar la selección. Además, para cambiar de lugar las tablas de acuerdo a la cantidad de tuplas, también se debe considerar que se puedan realizar los joins y no quede un producto cartesiano. En este caso, la tabla  $L$  es la de menor tamaño (350) y quedará en primer lugar a la izquierda. Luego, si bien después de aplicar la selección sobre la tabla  $C$  se obtienen 1000 tuplas, que es un número mayor que el 80% de las tuplas de  $P$  (640 tuplas), la tabla  $C$  debe ir más a la izquierda que la tabla  $P$ , ya que

de lo contrario quedaría un producto cartesiano entre L y P. Finalmente, se cambia cada conjunción de producto cartesiano y selección por un join.



c) Dar un plan físico para el plan lógico de la parte anterior, usando de manera eficiente los índices existentes. Justifique su respuesta.

En este caso se utilizan los índices para realizar cada selección. Dado que después de realizar cada selección se pierden los índices, no es posible usar los mismos para realizar los joins, por lo tanto, se aplica un join con bucle anidado en cada caso.

