

FUNDAMENTOS DE BASES DE DATOS

Examen Febrero 2020

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 (26 puntos).

Se desea modelar la realidad correspondiente al sistema de publicación de trabajos científicos. Los trabajos científicos son artículos que tienen un código identificador, un título, un resumen y un conjunto de palabras clave. Además, cada artículo tiene un conjunto de referencias a otros artículos.

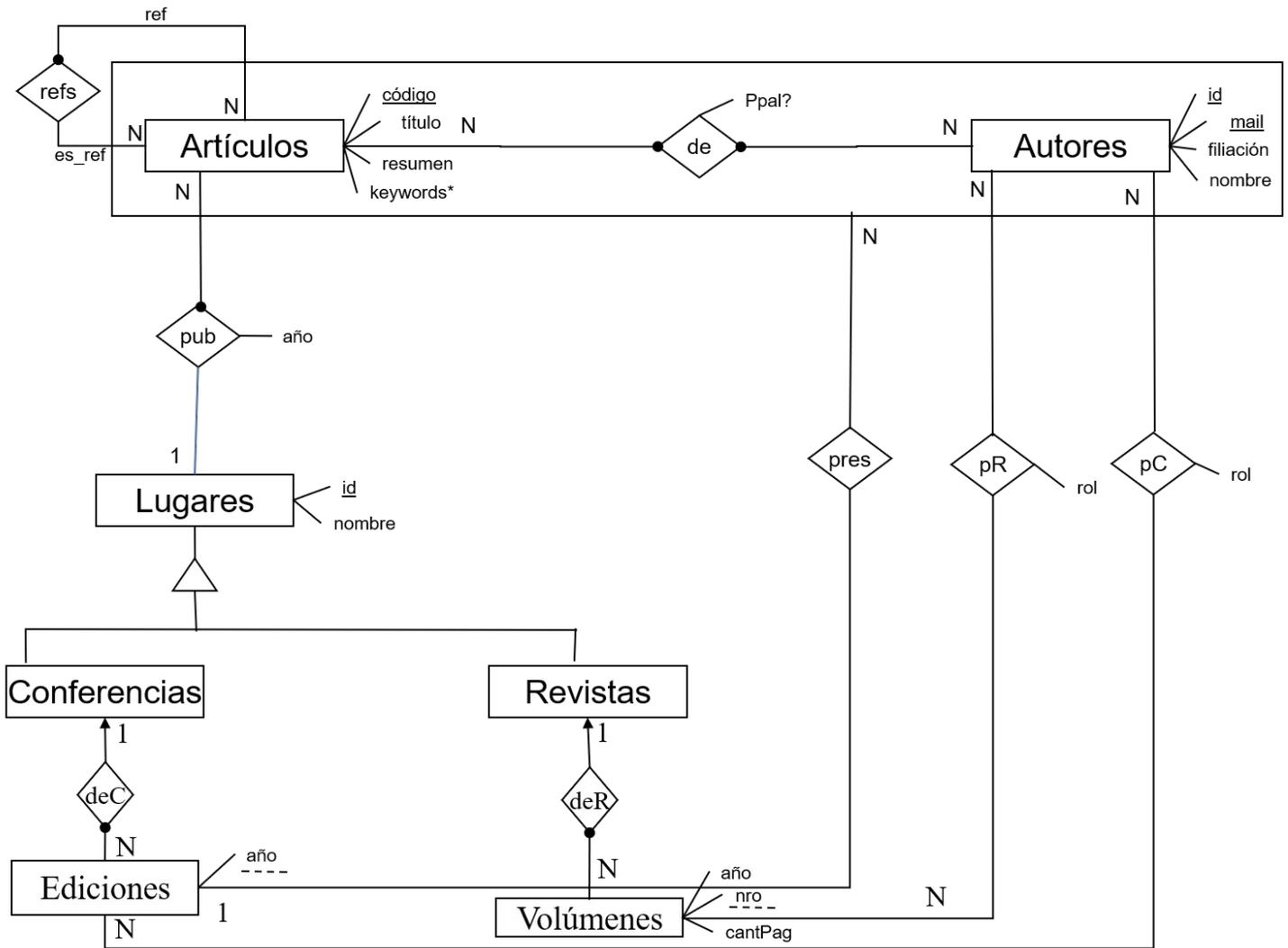
Los artículos son escritos por uno o más autores, por lo que interesa saber cuál es el autor principal. De cada autor se conoce su identificador, su nombre, su filiación (lugar de trabajo) y su e-mail que también lo identifica.

Los artículos se publican en diferentes lugares que pueden ser conferencias o revistas. Todo artículo es publicado en un único lugar y se registra su año de publicación. De cada lugar se conoce un código identificador y su nombre. Cada revista tiene, para cada año, varios volúmenes. De cada volumen se conoce además del año, un número que no puede repetirse dentro de la revista y la cantidad de páginas. Cada conferencia tiene varias ediciones y a cada edición la identifica el año de realización de la misma.

Algunos artículos son presentados por uno o varios de sus autores en la única edición de la conferencia en la que fue publicado dicho artículo. Para cada presentación interesa registrar el artículo, los autores que lo presentan y la edición de la conferencia.

Finalmente, los autores pueden tener tareas administrativas (revisión, coordinación, etc.) en los distintas ediciones de una conferencia y/o volúmenes de una revista. En ambos casos interesa registrar el rol que cumplió.

Se pide: Modelo Entidad-Relación completo del problema.



1. $Conferencias \cap Revistas = \emptyset$
2. $Conferencias \cup Revistas = Lugares$
3. $(\forall a \in Artículos) \langle a, a \rangle \notin refs$
4. $(\forall a \in Artículos) (\forall p \in Autores) (\forall e \in Ediciones) (\langle \langle a, p \rangle, e \rangle \in pres \rightarrow (\exists c \in Conferencias) (\langle e, c \rangle \in deC \wedge \langle a, c \rangle \in pub))$

Ejercicio 2 (28 puntos).

Considere una óptica donde se desea registrar la información de los clientes, de los empleados, de los productos que se venden, de las ventas y de las consultas realizadas a cada cliente.

De cada cliente se conoce su cédula, teléfono, dirección, y un conjunto de emails. De cada empleado se conoce su cédula, nombre y teléfono. De cada producto se guarda un código (que lo identifica), un tipo, y un nombre. De cada venta que se realiza a un cliente se registra el cliente, el producto, la fecha y la cantidad vendida. De cada consulta realizada se guarda el cliente, el empleado que lo atendió, la graduación del ojo derecho, la graduación del ojo izquierdo, y la fecha de la consulta.

Las cédulas se utilizan como identificadores de las personas.

Se asume que en una misma fecha, el mismo cliente no realizará más de una consulta. Además, se asume que en una misma fecha el cliente no compra más de una vez el mismo producto, por lo tanto, a una fecha, un producto y un cliente les corresponde una sola cantidad vendida.

Se utilizan los siguientes nombres de atributos:

ci-cliente	nom-emp	fecha-venta
tel-cliente	tel-emp	cant-vend
direccion	cod-prod	fecha-cons
email	tipo-prod	grad-od
ci-emp	nom-prod	grad-oi

Se pide:

- Deducir las dependencias funcionales que deberían cumplirse para esta realidad.
- Partiendo de la relación universal, dar una descomposición en 3NF, aplicando el algoritmo visto en clase.
- Decir si la descomposición obtenida preserva dependencias funcionales y tiene JSP.
- Hallar una dependencia multivaluada embebida que se cumpla en alguna de las relaciones obtenidas en la parte b). Justificar.
- Ahora, suponiendo que se llegó a la siguiente descomposición de la relación universal:

Clientes (ci-cliente, tel-cliente, direccion, email)

Empleados (ci-emp, nom-emp, tel-emp)

Productos (cod-prod, tipo-prod)

Consultas (ci-cliente, fecha-cons, ci-emp, grad-od, grad-oi)

Ventas (ci-cliente, fecha-venta, cod-prod, cant-vend, nom-prod)

decir en qué forma normal se encuentra la descomposición, justificando.

SOLUCION:

a)

ci-cliente → tel-cliente, direccion

ci-emp → nom-emp, tel-emp

cod-prod → tipo-prod, nom-prod

ci-cliente, fecha-cons → ci-emp, grad-od, grad-oi

ci-cliente, fecha-venta, cod-prod → cant-vend

b)

RelUniversal (ci-cliente, tel-cliente, direccion, email, ci-emp, nom-emp, tel-emp, cod-prod, tipo-prod, nom-prod, fecha-cons, fecha-venta, cant-vend, grad-od, grad-oi)

Conjunto de dfs minimal:

F= {
 ci-cliente → tel-cliente
 ci-cliente → direccion
 ci-emp → nom-emp
 ci-emp → tel-emp
 cod-prod → tipo-prod
 cod-prod → nom-prod
 ci-cliente, fecha-cons → ci-emp
 ci-cliente, fecha-cons → grad-od
 ci-cliente, fecha-cons → grad-oi
 ci-cliente, fecha-venta, cod-prod → cant-vend }

F es minimal porque cumple las 3 propiedades:

(1) Todas las dfs tienen un solo atributo a la derecha.

(2) Ninguna df tiene atributos redundantes a la izquierda. Esto lo verificamos en las últimas 4 dfs, donde si sacamos cualquiera de los atributos de una df no se puede obtener el de la derecha de la df.

(3) No hay dfs redundantes. Esto se verifica muy fácilmente, viendo que todos los atributos de la derecha de las dfs aparecen una sola vez, por lo tanto si elimino una df no voy a poder llegar a determinar el atributo de la derecha de esa df.

Unica clave: ci-cliente, cod-prod, fecha_cons, fecha-venta, email

Es clave porque a partir de ese conjunto de atributos determino a todos los demás atributos de RelUniversal y si quito a cualquiera de ellos, ya no determino a todos. Es la única clave porque cada uno de esos atributos tienen que estar en todas las claves.

Aplicamos siguientes pasos del algoritmo de 3NF con JSP y preservación de dfs:

Construir relaciones a partir de las dfs:

Cientes (ci-cliente, tel-cliente, direccion)

Empleados (ci-emp, nom-emp, tel-emp)

Productos (cod-prod, tipo-prod, nom-prod)

Consultas (ci-cliente, fecha-cons, ci-emp, grad-od, grad-oi)

Ventas (ci-cliente, fecha-venta, cod-prod, cant-vend)

Agregar una relación con una clave:

Ventas-Cons (ci-cliente, cod-prod, fecha_cons, fecha-venta, email)

c)

La descomposición preserva las dependencias funcionales y tiene JSP, porque esto lo asegura el algoritmo aplicado.

d)

Algunas dependencias multivaluadas embebidas que se cumplen en la relación Ventas-Cons son las siguientes:

ci-cliente -->> fecha_cons | fecha-venta

ci-cliente -->> cod-prod | email

ci-cliente -->> cod-prod | fecha-cons

e)

La forma normal de la descomposición planteada es la forma normal menos restrictiva de las de los sub-esquemas de la descomposición.

Forma normal de **Cientes (ci-cliente, tel-cliente, direccion, email)**:

Proyectamos las dfs halladas en la parte (a) sobre la relación Cientes.

$F_{\text{Cientes}} = \{ \text{ci-cliente} \rightarrow \text{tel-cliente, direccion} \}$

Clave: (ci-cliente, email)

La df ci-cliente \rightarrow tel-cliente, direccion viola 2NF, ya que es una dependencia parcial de una clave.

Por lo tanto, la relación Cientes está en 1NF.

No es necesario hallar las formas normales de los otros sub-esquemas de la descomposición, ya que 1NF es la forma normal menos restrictiva posible. **Por lo tanto, la descomposición se encuentra en 1NF.**

Otra manera posible de llegar a la solución:

Forma normal de **Ventas (ci-cliente, fecha-venta, cod-prod, cant-vend, nom-prod)**:

$F_{Ventas} = \{ \text{ci-cliente, fecha-venta, cod-prod} \rightarrow \text{cant-vend, cod-prod} \rightarrow \text{nom-prod} \}$

Clave: (ci-cliente, fecha-venta, cod-prod)

La df $\text{cod-prod} \rightarrow \text{nom-prod}$ viola 2NF, ya que es una dependencia parcial de una clave.

Por lo tanto, la relación Ventas está en 1NF.

Ejercicio 3 (28 puntos)

La intendencia de Montevideo cuenta con información de las distintas carreras que se llevan a cabo en la ciudad y sus participantes, en una base de datos con el siguiente esquema.

CARRERAS(*nomCarrera, año, organizador*)

Contiene información de las distintas carreras que se realizan.

PARTICIPANTES(*CI, nomParticipante, edad, género*)

Contiene información básica de todas las personas que alguna vez participaron de una carrera.

CATEGORIAS(*idCategoria, nomCategoria, distancia*)

Cuenta con información sobre las distintas categorías existentes. Por ejemplo, "mujeres de entre 18 y 29 años" que corren una distancia de 5km.

INSCRIPCIONES(*nomCarrera, año, CI, idCategoria*)

Para cada carrera y persona que se inscribe a la misma se registra la categoría en la que participó.

$$\Pi_{nomCarrera, año}(INSCRIPCIONES) \subseteq \Pi_{nomCarrera, año}(CARRERAS)$$

$$\Pi_{CI}(INSCRIPCIONES) \subseteq \Pi_{CI}(PARTICIPANTES)$$

$$\Pi_{idCategoria}(INSCRIPCIONES) \subseteq \Pi_{idCategoria}(CATEGORIAS)$$

Se pide:

a. Resolver la siguiente consultas en Álgebra Relacional:

- I. Cédula de las personas que se inscribieron a todas las carreras de 2018 en una categoría de distancia 5km.

Sol

$$Car_{2018} = \Pi_{nomCarrera, año}(\sigma_{año=2018}(CARRERAS))$$

$$Insc_{5k} = \Pi_{CI, nomCarrera, año}(\sigma_{distancia=5k}(CATEGORIAS) * INSCRIPCIONES)$$

$$Sol = (Insc_{5k} \div Car_{2018})$$

b. Resolver la siguiente consulta en Cálculo Relacional:

- I. Nombre de carrera, año y id categoría tal que todos los inscriptos a la carrera lo hicieron en esa categoría.

Sol

$$\{ \langle C.nomCarrera, C.año, Cat.idCategoria \rangle / \\ Carrera(C) \wedge Categoria(Cat) \wedge \\ (\forall I)(Inscriptos(I) \wedge \\ I.nomCarrera = C.nomCarrera \wedge I.año = C.año \rightarrow \\ I.idCat = Cat.idCat) \}$$

c. Resolver las siguientes consultas en SQL:

I. Nombre de las carreras (y año) con inscripciones en la mayor cantidad de categorías distintas.

Sol

```
SELECT nomCarrera, año
FROM INSCRIPCIONES
GROUP BY nomCarrera, año
HAVING COUNT(DISTINCT(idCategoria)) ≥ ALL
            (SELECT COUNT(DISTINCT(idCategoria))
             FROM INSCRIPCIONES
             GROUP BY nomCarrera, año
            )
```

II. Devolver para cada cédula, la cantidad de km totales que corrió esa persona.

Sol

```
SELECT CI, SUM(distancia)
FROM INSCRIPCIONES NATURAL JOIN CATEGORIAS
GROUP BY CI
```

Ejercicio 4 (18 puntos)

La siguiente realidad corresponde a una clínica veterinaria especialista en perros. Las tablas relevantes de la base son 3:

Perros(*matricula*, *nombre*, *raza*, *telefonoDueño*)

Contiene la información de los pacientes de la clínica.

Veterinarios(*ciVet*, *nombre*, *telefono*, *especialidad*)

Contiene la información de todos los veterinarios que trabajaron en algún momento en la clínica.

Atencion(*ciVet*, *matricula*, *fechaAtención*, *costo*)

Contiene información de cada atención a un animal.

Sea la siguiente consulta SQL:

```
SELECT A.fechaAtención
FROM Perros P, Veterinarios V, Atencion A
WHERE A.ciVet=V.ciVet AND A.matricula=P.matricula AND
      P.raza='caniche' AND A.costo>10.000
```

Se conoce la siguiente información:

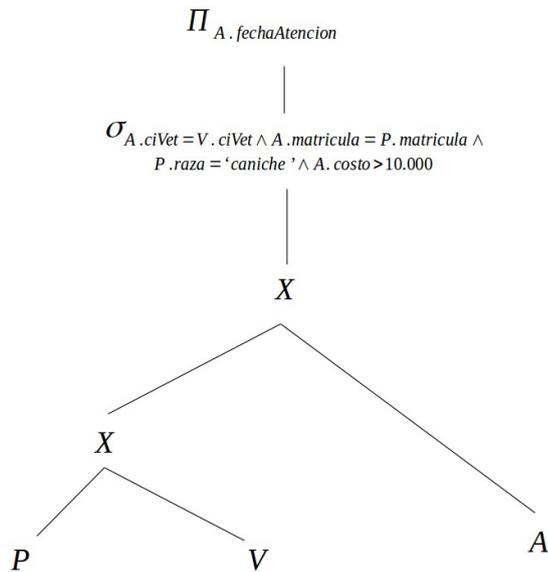
Relación R	n _R	Atributos	Índices
Veterinarios D	100	V(nombre, D) = 25 V(especialidad, D) = 5 (distribución uniforme)	- Índice Prim. de 1 nivel sobre ciVet
Perros P	2000	V(raza, P) = 25 (distribución uniforme)	- Índice Prim. de un nivel sobre matricula - Índice Sec. árbol b+ de 3 niveles sobre raza
Atención A	5000	El 1% de las tuplas corresponden a atenciones mayores a \$10.000.	Índice Prim. de 1 nivel sobre ciVet, matricula, fechaAtención.

Se pide:

- Dar el árbol canónico
- Dar el plan lógico, aplicando heurísticas, justificando cada paso.
- Dar 1 plan físico posible.

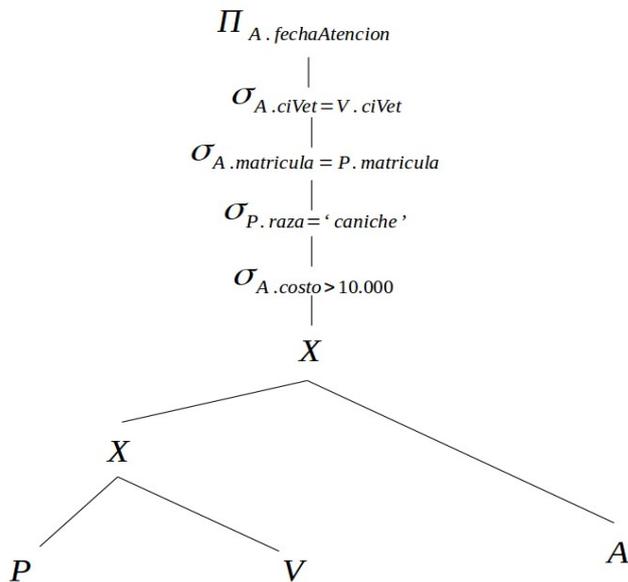
Solución

a. Dar el árbol canónico

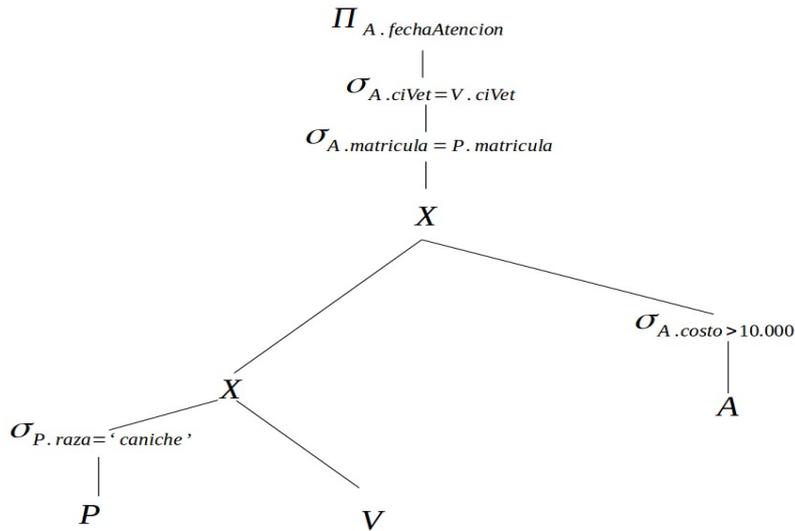


b. Dar el plan lógico, aplicando heurísticas, justificando cada paso.

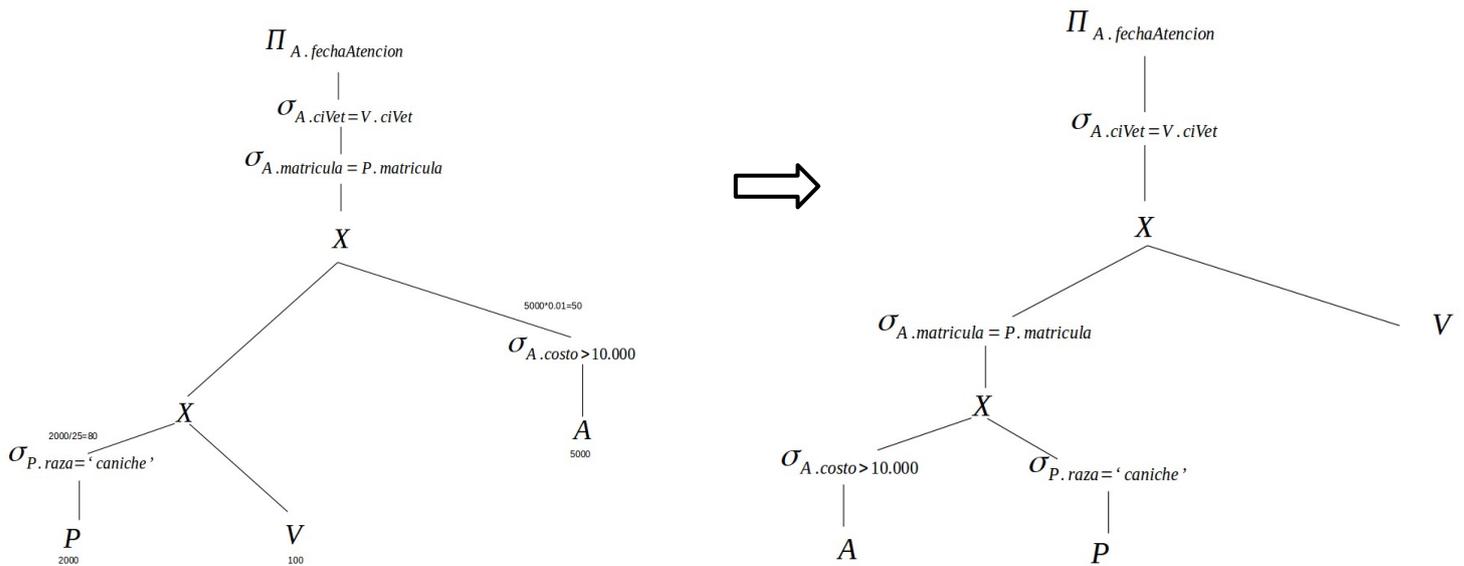
Paso 1: separar las selecciones en cascada de selecciones



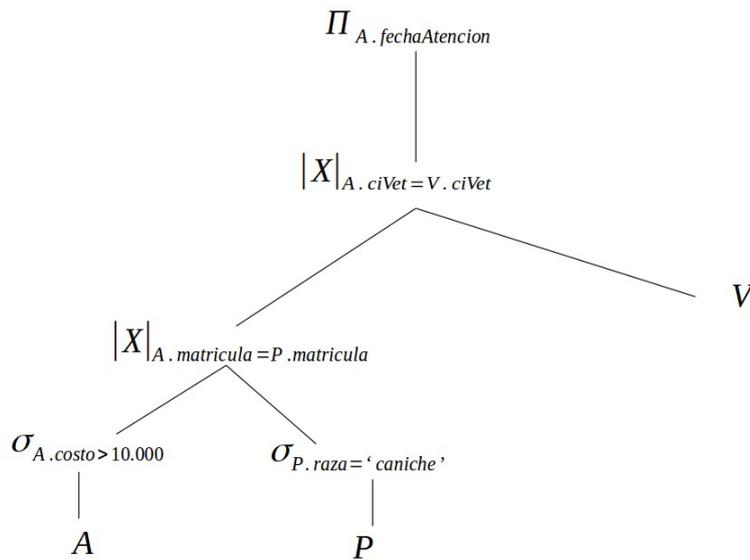
Paso 2: bajar las selecciones los más posible



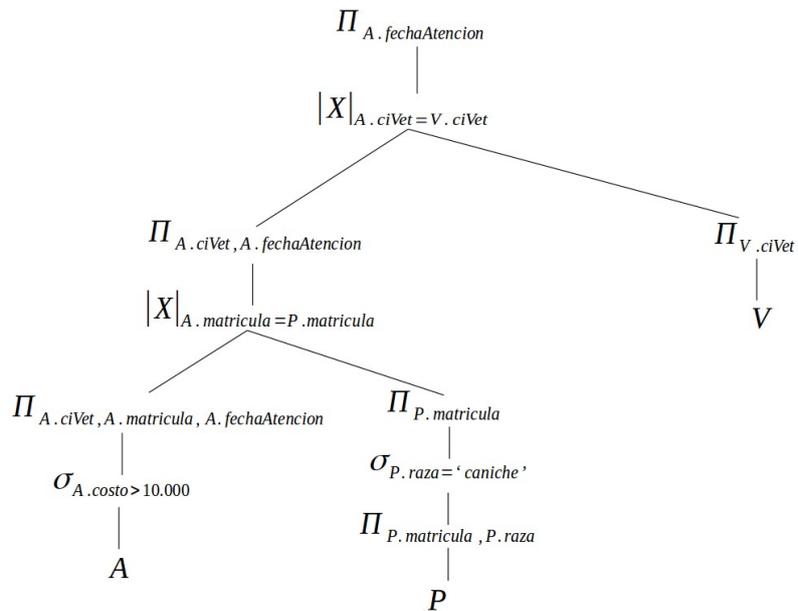
Paso 3: mover a la izquierda las hojas con menos tuplas, sin generar productos cartesianos innecesarios



Paso 4: cambiar productos cartesianos y selecciones por joins



Paso 5: agregar todas las proyecciones necesarias para manejar la menor cantidad de datos posibles



c. Dar 1 plan físico posible.

