Segundo Parcial de Fundamentos de Bases de Datos

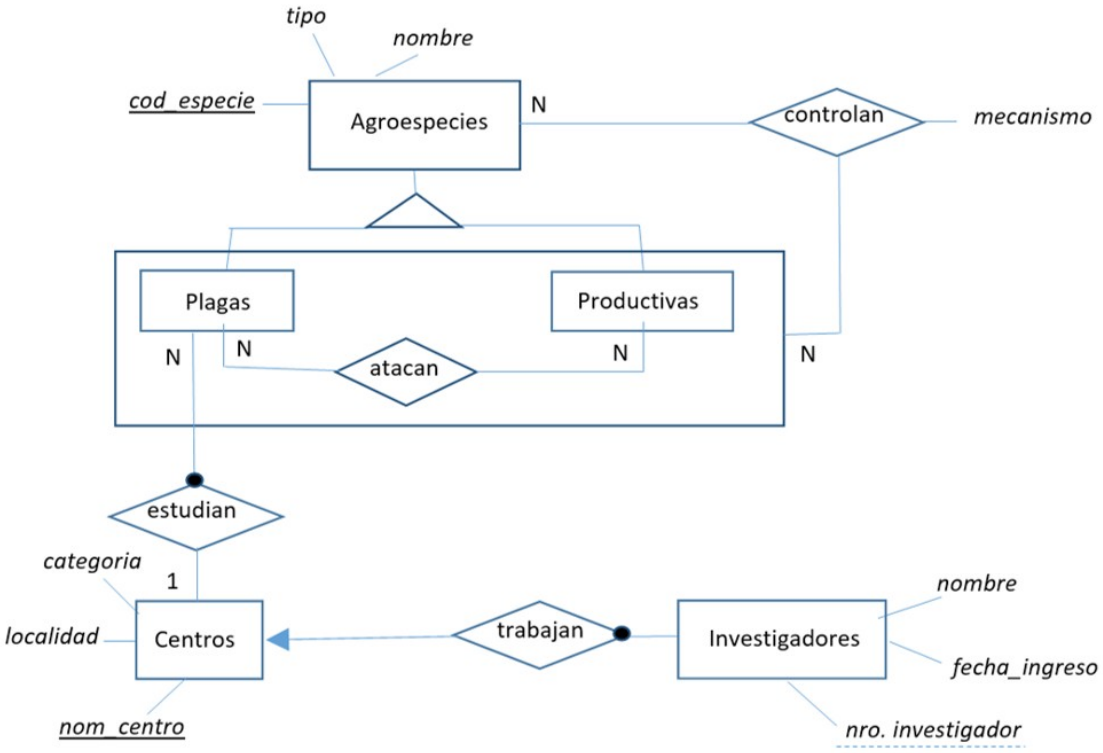
Noviembre 2019

SOLUCION

Ejercicio 1 (10 puntos)

El siguiente Modelo Entidad-Relación es el modelo conceptual de una base de datos de agroespecies.

Las agroespecies son un conjunto de seres vivos que conviven en un ecosistema agrícola, las cuales pueden ser plagas o especies productivas. Cada agroespecie se identifica a través de un código de especie.
Las plagas atacan a una o varias especies productivas, y éstas pueden ser atacadas por una o varias plagas. A su vez, algunas agroespecies (de cualquier tipo) se utilizan para el control biológico de algunas plagas cuando atacan a determinadas especies productivas. Interesa saber el mecanismo por el cual se realiza este control biológico.
Cada plaga es estudiada por un único centro de investigación en control de plagas, que se identifican por su nombre, y cada centro se dedica al estudio de una o más plagas.
En cada centro de control de plagas trabajan un conjunto de investigadores, que se identifican por un número de investigador dentro de cada centro.



RNE: No hay restricciones no estructurales.

Se pide:
Hacer el pasaje de MER a Modelo Relacional, especificando las tablas, sus claves y las dependencias de inclusión.

Solución:

Aclaración:
Durante el parcial se cambió el nombre del atributo ***tipo*** de la entidad ***Agroespecies*** por el nombre ***reino***, para que no generara confusión con algún otro significado de la palabra tipo.

Tablas y claves

Agroespecies (*cod_especie*, *reino*, *nombre*)

Plagas (*cod_especie*, *nom_centro*)

Productivas (*cod_especie*)

Atacan (*cod_especie_plaga*, *cod_especie_productiva*)

Controlan (*cod_especie*, *cod_especie_plaga*, *cod_especie_productiva*, *mecanismo*)

Centros (*nom_centro*, *localidad*, *categoria*)

Investigadores (*nom_centro*, *nro_investigador*, *nombre*, *fecha_ingreso*)

Dependencias de inclusión

$\prod_{cod\_especie}(Plagas) \subseteq \prod_{cod\_especie}(Agroespecies)$

$\prod_{nom\_centro}(Plagas) \subseteq \prod_{nom\_centro}(Centros)$

$\prod_{cod\_especie}(Productivas) \subseteq \prod_{cod\_especie}(Agroespecies)$

$\prod_{cod\_especie\_plaga}(Atacan) \subseteq \prod_{cod\_especie}(Plagas)$

$\prod_{cod\_especie\_productiva}(Atacan) \subseteq \prod_{cod\_especie}(Productivas)$

$\prod_{cod\_especie}(Controlan) \subseteq \prod_{cod\_especie}(Agroespecies)$

$\prod_{cod\_especie\_plaga,\ cod\_especie\_productiva}(Controlan) \subseteq \prod_{cod\_especie\_plaga,\ cod\_especie\_productica}(Atacan)$

$\prod_{nom\_centro}(Investigadores) \subseteq \prod_{nom\_centro}(Centros)$

Ejercicio 2 (20 puntos)

Considere un esquema de relación R(A,B,C,D,E,G) con el siguiente conjunto de dependencias funcionales:

$$F=\{AB\rightarrow DE, DA\rightarrow C, B\rightarrow G, DE\rightarrow B\}$$

y las siguientes dependencias multivaluadas embebidas: $\{C\twoheadrightarrow A \mid D, G\twoheadrightarrow A \mid E\}$

a) Determine todas las claves. Justifique su respuesta.

Primero se calculan tres conjuntos de atributos: los que nunca están a la derecha (ND), los que sólo están a la derecha (SD) y los que están de ambos lados de las dependencias (ID).

$$ND=\{A\}$$
$$SD=\{C,G\}$$
$$ID=\{B,D,E\}$$

Los elementos del conjunto ND tienen que estar en todas las claves y los del conjunto SD nunca están en una clave. Los del conjunto ID pueden estar o no estar en las claves.
Por esto se calcula la clausura del conjunto ND. Si ND es una superclave, entonces es la única.

$$ND^{+}=\{A\}$$

Dado este resultado, entonces hay calcular las clausuras considerando todos los elementos de ND y alguno de ID. Una posibilidad es comenzar considerando combinaciones de atributos que estén (aunque no formen completamente) un lado izquierdo de alguna dependencia. De esta forma es que se calcula lo siguiente:

$$AB^{+}=\{A,B,D,E,C,G\}$$

Este resultado indica que AB es una superclave y por lo tanto (dado que A no lo es y tiene que estar en todas) es una clave.

Ahora se verifica si queda alguna superclave por considerar. Para esto, se computa la clausura de $ND\cup(ID-\{B\})$. Si este conjunto es una superclave, entonces queda alguna clave por encontrar.

$$ADE^{+}=\{A,D,E,C,B,G\}$$

Por lo tanto, se debe verificar si AD o AE son superclaves. En caso que alguno de los dos conjuntos cumplan esa condición, esas serán las claves. En caso contrario, la clave restante será ADE.

$$AE^{+}=\{A,E\}$$
$$AD^{+}=\{A,D,C\}$$

En vista de este resultado, las únicas claves son AB y ADE. Observar que el mecanismo que se siguió garantiza que son todas.

b) Determine en qué forma normal está R.

Dado que B es parte de una clave y G no está en ninguna, la dependencia $B\rightarrow C$ induce una dependencia parcial desde la clave AB sobre G, por lo que se viola 2NF. Esta observación permite concluir que R está en 1NF.

c) Obtener una descomposición en BCNF de R siguiendo el algoritmo visto en el curso. Explique porqué motivo aplica cada paso. (O justifique).

En BCNF, todas las dependencias deben tener una superclave del lado izquierdo. La primer dependencia cumple con la condición pero la segunda no lo hace. Por esto se crea la siguiente descomposición:

$$R_1(A,D,C) \qquad \Pi_{R_1}(F)=\{DA \rightarrow C\} \qquad (BCNF)$$
$$R_2(A,B,D,E,G) \qquad \Pi_{R_2}(F)=\{AB \rightarrow DE, B \rightarrow G, DE \rightarrow B\} \qquad (1NF)$$

Ahora es necesario seguir dividiendo R_2 según la dependencia $B \rightarrow G$:

$$R_{21}(B,G) \qquad \Pi_{R_{21}}(F)=\{B \rightarrow G\} \qquad (BCNF)$$
$$R_{22}(A,B,D,E) \qquad \Pi_{R_{22}}(F)=\{AB \rightarrow DE, DE \rightarrow B\} \qquad (3NF)$$

Observar que R_{22} está en 3NF y no en BCNF dado que B es primo (en esa tabla se conservan las claves). Para obtener en BCNF, hay que separar nuevamente.

$$R_{221}(B,D,E) \qquad \Pi_{R_{221}}(F)=\{DE \rightarrow B\} \qquad (BCNF)$$
$$R_{222}(A,D,E) \qquad \Pi_{R_{222}}(F)=\{\} \qquad (BCNF)$$

El esquema final está formado por las relaciones R_1, R_{21}, R_{221} y R_{222}

d) Determinar si se perdieron dependencias funcionales.

Para determinar si se perdieron dependencias es necesario estudiar la equivalencia entre F y la unión de las proyecciones de las dependencias:

$$F' = \Pi_{R_1} \cup \Pi_{R_{21}} \cup \Pi_{R_{221}} \cup \Pi_{R_{222}} = \{DA \rightarrow C, B \rightarrow G, DE \rightarrow B\}$$

Se ha perdido la dependencia $AB \rightarrow DE$. Para asegurarse de que esa dependencia se perdió, se pude observar que los atributos DE no aparecen a la derecha de ninguna de las dependencias involucradas, por lo que no son alcanzables mediante ninguna regla de inferencia.

e) Considerar las dependencias multivaluadas embebidas y llevar a 4NF con el algoritmo visto en el curso. Justifique.

Para considerar las dependencias multivaluadas embebidas hay que observar si se alcanzaron los esquemas en donde se cumplirían. En este caso, los esquemas serían R'(C,A,D) y R''(G,A,E).

Se observa que el esquema R' coincide con R_1 mientras que ninguno de los esquemas coincide con R''. Por este motivo, sólo hay que considerar la dependencia multivaluada $C \twoheadrightarrow A$ en el esquema R_1 y aplicar el algoritmo para 4NF (que es el mismo que para BCNF):

$$R_{11}(C,A)$$
$$R_{12}(C,D)$$

Dado que no hay dependencias multivaluadas no triviales, ahora las dos relaciones están en 4NF. Las relaciones restantes ya estaban en 4NF dado que todas las dependencias existentes tienen una

superclave del lado izquierdo.

La descomposición final en 4NF es la siguiente:

$R_{11}(C, A)$
$R_{12}(C, D)$
$R_{21}(B, G)$
$R_{221}(B, D, E)$
$R_{222}(A, D, E)$

Ejercicio 3 (15 puntos)

El siguiente es parte del esquema de una base de datos de gestión de un laboratorio de estudios clínicos.

PACIENTES (idPaciente, nomPaciente, institucionMedica)
Representa información de los pacientes. Identificador, nombre e institución médica a la cual pertenece.

ESTUDIOS (idEstudio, nomEstudio, indicaciones)
Representa información respecto a los diferentes estudios que efectúa el laboratorio. Identificador, nombre del estudio e indicaciones.

ESTUDIOS_PACIENTES (idPaciente, idEstudio, fecha, resultado, entregado)
Representa la información sobre los estudios realizados a cada paciente. Identificador del paciente, identificador del estudio, fecha en la cual se realizó el estudio, resultado del mismo y si ya fue entregado al paciente (con valor TRUE si fue entregado y FALSE en caso contrario).

Además, se conoce la siguiente información sobre estos datos:

	Tamaño	Atributos
PACIENTES	12000	InstituciónMedica tiene 200 valores distintos, distribuidos uniformemente.
ESTUDIOS	550	
ESTUDIOS_PACIENTES	450000	Un 98% de los estudios ya fueron entregados al paciente.

Considere la siguiente consulta sobre el esquema dado:

```
SELECT EP.idPaciente, P.nomPaciente, EP.resultado
FROM Estudios_Pacientes EP, Pacientes P, Estudios E
WHERE EP.idPaciente = P.idPaciente AND
      EP.idEstudio = E.idEstudio AND
      P.institucionMedica = 'CASMU' AND
      EP.entregado = TRUE
```

Se pide:

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

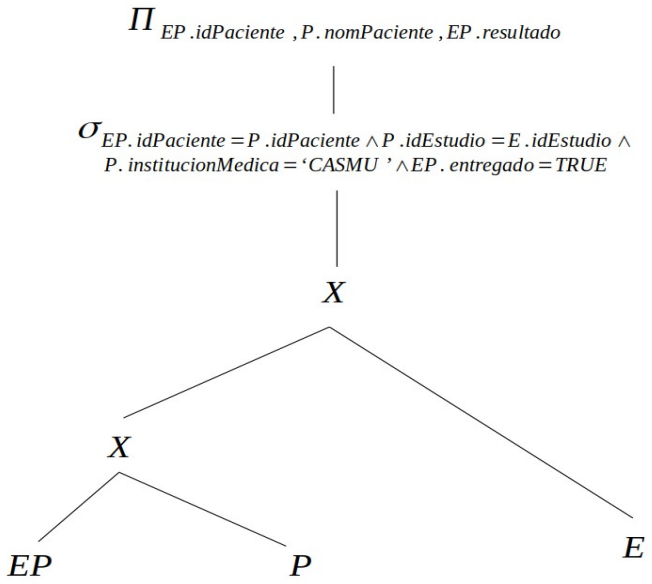
b) Aplicar las heurísticas para optimización llegando al plan lógico optimizado. Explique cada uno de los pasos ejecutados.

c) ¿Qué índices le parece útil tener para este esquema y esta consulta? Para cada uno, decir qué tipo de índice sería y sobre qué atributo/s se definiría.

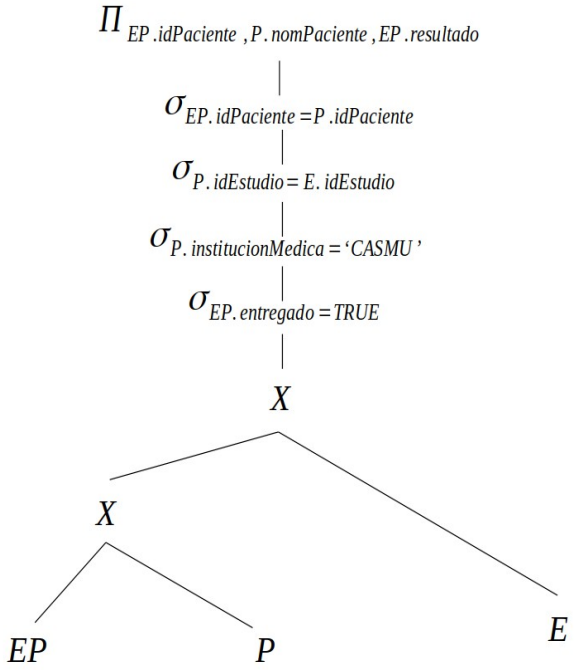
d) Considerando las respuestas de las partes a), b) y c), dar un plan físico que le parezca adecuado.

Solución

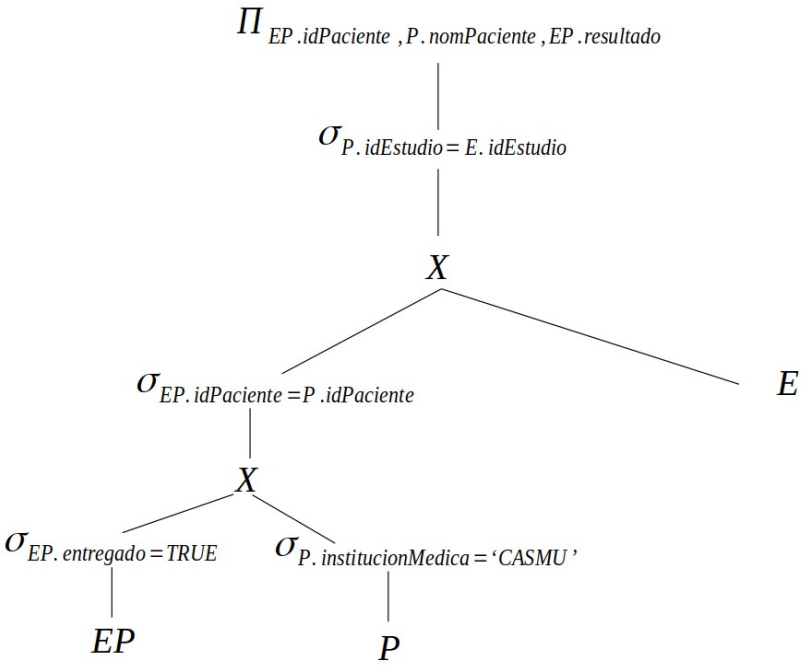
a) Árbol canónico del plan lógico



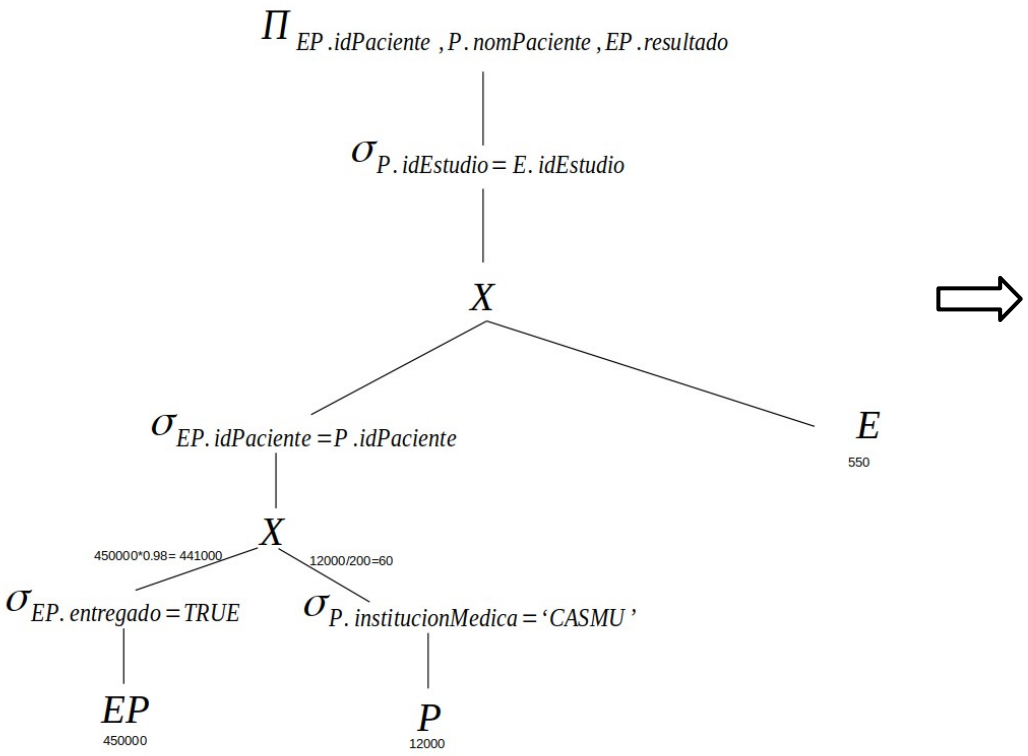
b) Paso 1: separar las selecciones en cascada de selecciones

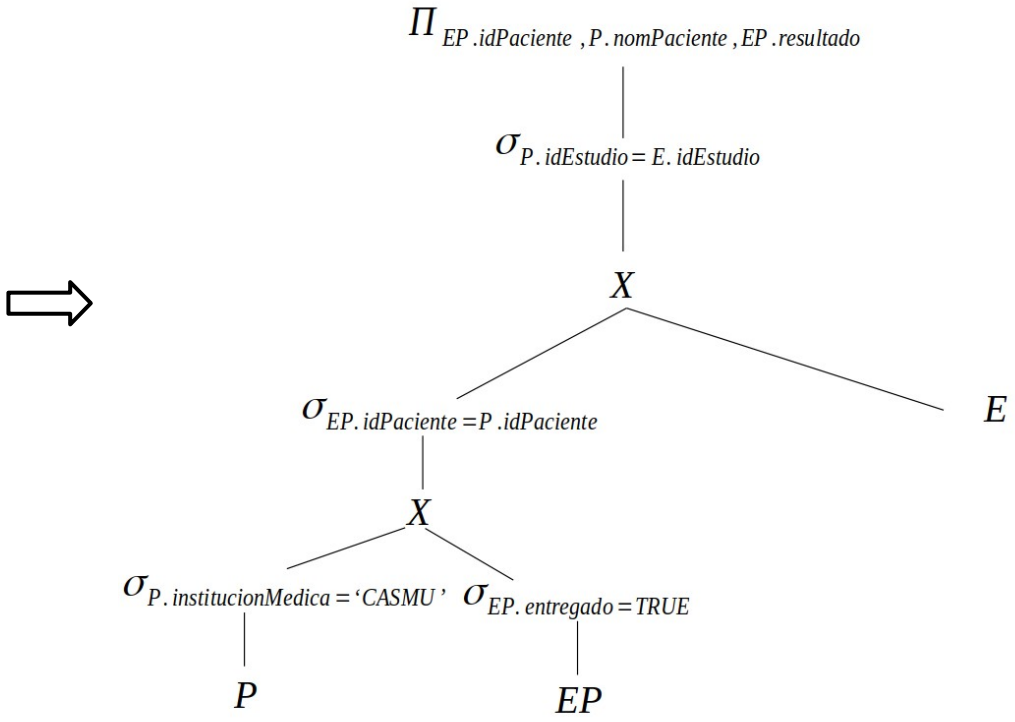


Paso 2: bajar las selecciones los más posible

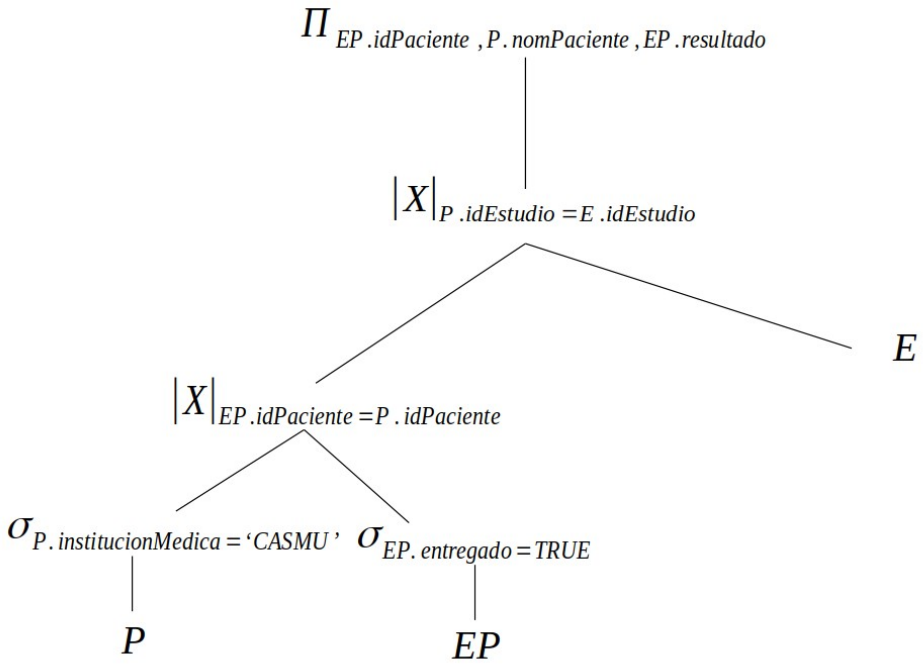


Paso 3: mover a la izquierda las hojas con menos tuplas, sin generar productos cartesianos innecesarios

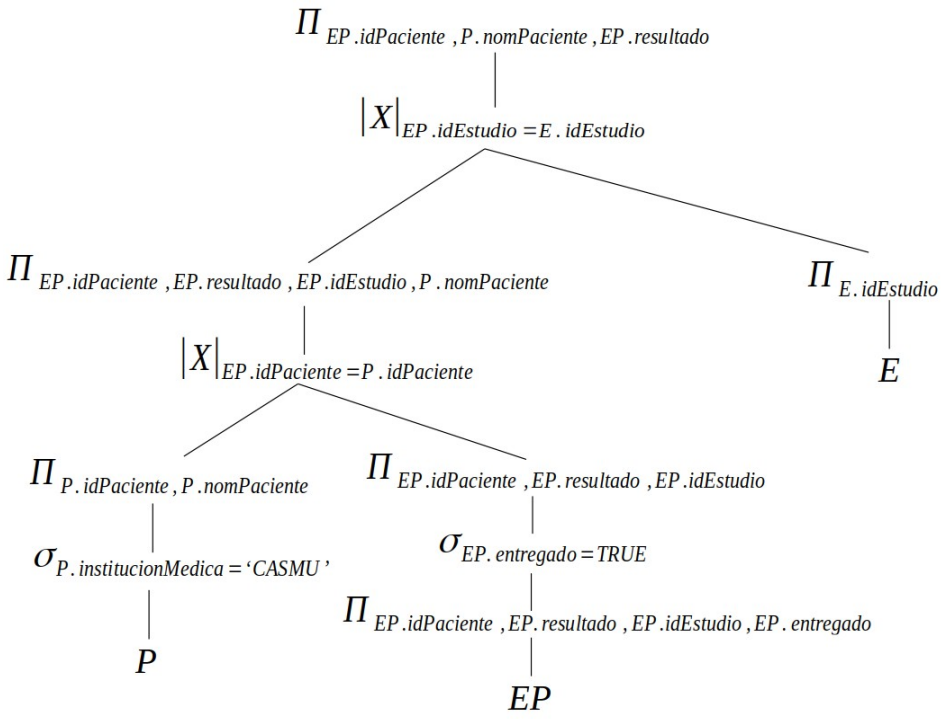




Paso 4: cambiar productos cartesianos y selecciones por joins

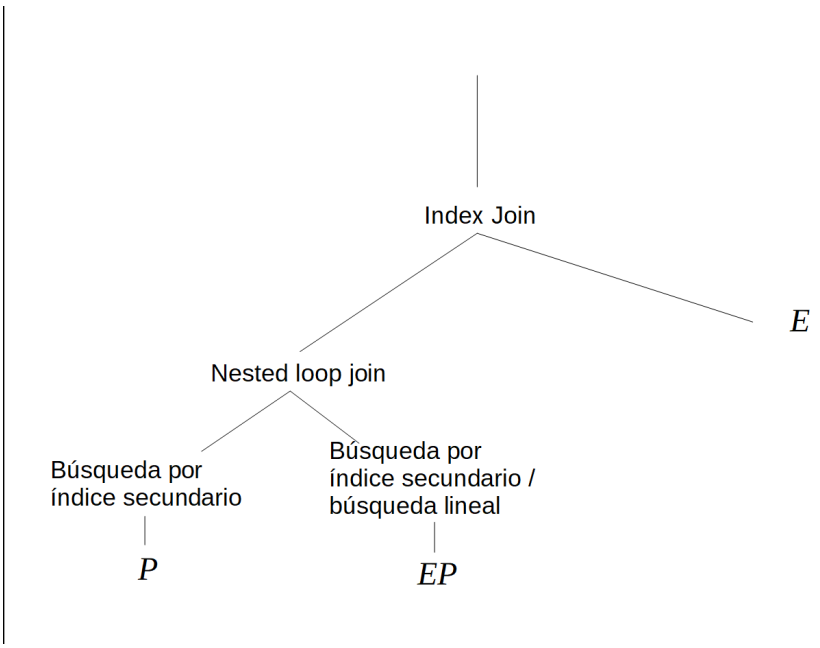


Paso 5: agregar todas las proyecciones necesarias para manejar la menor cantidad de datos posibles



c) Sería útil contar con un índice primario para todas las claves. También para mejorar la performance de las selecciones es de utilidad contar con un índice secundario por el atributo institucionMedica en la tabla Pacientes. Se puede considerar también un índice secundario por el atributo entregado en la tabla Estudios_Pacientes. Observar que como en este caso el 98% de los datos cumplen la condición, no se va a ganar demasiado en cuanto a operaciones de lectura al agregar dicho índice.

d) Un posible plan físico se presenta a continuación.



Ejercicio 4 (15 puntos)

Dadas las siguientes transacciones:

T1: r1(X) w1(X) r1(Y) c1
T2: r2(X) r2(Y) w2(Y) c2

y las siguientes historias:

H1: r1(X) w1(X) r2(X) r2(Y) r1(Y) c1 w2(Y) c2

H2: r1(X) r2(X) r2(Y) w1(X) w2(Y) c2 r1(Y) c1

a) Decir cuál de esas historias sería mejor, considerando si su ejecución es equivalente a una ejecución serial y la robustez de las historias para los casos de fallas y/o abortos. Explicitar las propiedades que verifique siguiendo el siguiente formato y justificando:

	Propiedad1	……..	Propiedadn
H1	si/no	….	si/no
H2	si/no	….	si/no

Solución:

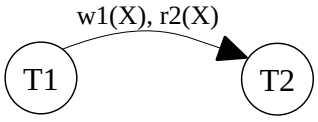
Para saber si su ejecución es equivalente a una ejecución serial, evitándose errores en los resultados, se verificará la propiedad: *serializable*.
Para saber si las historias son robustas en casos de fallas y/o abortos, se verificarán las propiedades: *recuperable*, *evita abortos en cscada* (*EAC*) y *estricta*.

	Serializable	**Recuperable**	**EAC**	**Estricta**
H1	si	si	no	no
H2	si	si	si	si

Justificaciones:

H1:

Serializable:



Es serilizable porque el grafo no tiene ciclos.

Recuperable:

T2 lee de T1 (T1 no lee de T2) y T1 hace commit antes que T2, por lo tanto es recuperable.

La lectura es: w1(X) r2(X)

EAC:

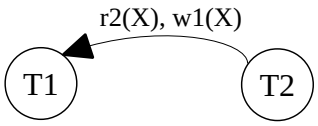
T2 lee de T1, y cuando lee, T1 aún no hizo commit, por lo tanto no cumple EAC.

Estricta:

Si la historia no cumple EAC entonces no es Estricta.

H2:

Serializable:



Es serilizable porque el grafo no tiene ciclos.

Recuperable:

T1 lee de T2 (T2 no lee de T1) y T2 hace commit antes que T1, por lo tanto es recuperable.

La lectura es: w2(Y) r1(Y)

EAC:

T1 lee de T2 cuando T2 ya hizo commit, por lo tanto cumple EAC.

Estricta:

T1 sólo lee un valor escrito por otro, no escribe ningún valor escrito por otro (T2 tampoco). Por lo tanto, la historia cumple la propiedad Estricta, que dice que solo se puede leer o escribir valores que fueron escritos por transacciones que ya hicieron commit.

Nota: Si verificábamos primero la propiedad Estricta, automáticamente hubieramos sabido que cumple EAC y recuperable.

Finalmente, la historia que es mejor es H2, ya que ambas son serializables, pero en H1 pueden ocurrir abortos en cascada y en H2 no.

b) Ahora considere las siguientes transacciones con locks y unlocks de los ítems de datos.

T1: rl1(X) r1(X) wl1(X) w1(X) rl1(Y) u1(X) r1(Y) u1(Y) c1

T2: rl2(X) r2(X) wl2(X) w2(X) rl2(Y) r2(Y) u2(X) u2(Y) c2

T3: rl3(X) r3(X) u3(X) rl3(Y) r3(Y) u3(Y) wl3(Y) w3(Y) u3(Y) c3

b.1) Para cada transacción decir si cumple el protocolo 2PL, justificando.

T1 y T2 cumplen 2PL, ya que primero cumplen la etapa de expansión , adquiriendo más locks y luego la de contracción, liberándolos.

T3 no cumple 2PL, ya que en la porción de la transaccion “rl3(X) r3(X) u3(X) rl3(Y)” toma un lock, lo libera y luego toma nuevamente un lock. Por lo tanto expande, contrae y expande nuevamente.

b.2) Dar un historia H1 que ejecute T1 y T2, y una historia H2 que ejecute T1 y T3. Ambas historias deben ser posibles en el sistema y deben entrelazar operaciones de lectura y/o escritura de las transacciones que ejecutan.

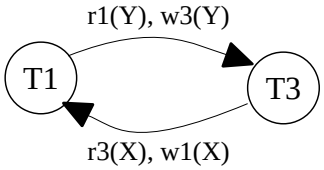
H1: rl1(X) r1(X) wl1(X) w1(X) rl1(Y) u1(X) rl2(X) r2(X) r1(Y) u1(Y) c1 wl2(X) w2(X) rl2(Y) r2(Y) u2(X) u2(Y) c2

H2: rl1(X) r1(X) rl3(X) r3(X) u3(X) wl1(X) w1(X) rl1(Y) u1(X) rl3(Y) r3(Y) r1(Y) u1(Y) u3(Y) wl3(Y) w3(Y) u3(Y) c3 c1

b.3) Decir si las historias H1 y H2 de la parte anterior son serializables, justificando.

H1 es serializable porque las transacciones que la integran cumplen el protocolo 2PL.

Verificamos si H2 es serializable:



H2 no es serializable.