NOMBRE: Benjamín Farías Valdés

N.ALUMNO: 17642531

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación

IIC3413 — Implementación de Sistemas de Base de Datos — 1' 2020

Control 3

Doy mi palabra que la siguiente solución de la pregunta 1 fue desarrollada y escrita individualmente por mi persona según el código de honor de la Universidad.

Figura 1: Firma

1. Consultas Acíclicas

1.1. Árboles de Join

- **A)** Se utilizará el método de **'cortar orejas'** visto en clase. Primero, obtenemos los conjuntos de variables de cada **hiper-arista**:

$$(a, b, c)|(c, d, e)|(a, f, e)|(a, c, e)$$

Al revisar el primero, podemos ver que su variable b **NO está presente en ningún otro conjunto**. Además, sus otras 2 variables (a y c) están contenidas en el último conjunto (a, c, e). Considerando a este último como la **cara**, se tiene que (a, b, c) cumple con las condiciones para ser una **oreja**, por lo que es eliminado:

$$(c, d, e)|(a, f, e)|(a, c, e)$$

Ocurre algo similar para el conjunto (c, d, e), donde la variable única es la d y las otras 2 también están presentes en (a, c, e). Lo mismo para el conjunto (a, f, e) con su variable única f. Por lo tanto, se eliminan ambos:

$$(a, c, e)$$

Como se logró reducir el hiper-grafo a 1 sola hiper-arista por medio de eliminación de orejas, se tiene que la consulta es **acíclica**. Esto implica que la consulta **tiene un Árbol de Join** (demostrado en clases).

Un posible Árbol de Join es el siguiente:

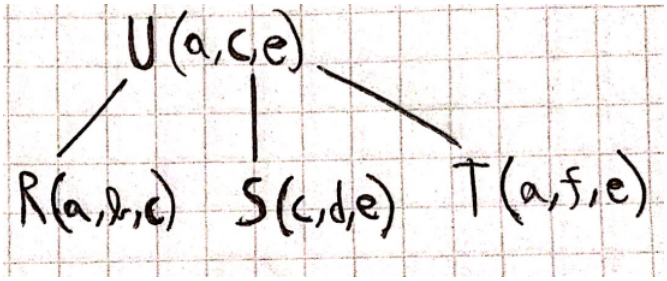

Figura 2: Posible Árbol de Join

- **B)** Se utilizará el método de **'cortar orejas'** visto en clase. Primero, obtenemos los conjuntos de variables de cada **hiper-arista**:

$$(a, b, c)|(b, c, d)|(a, b, e)|(a, b, c, d)$$

Al revisar el primero y el segundo, podemos ver que ambos están contenidos completamente en el último conjunto (a, b, c, d). Considerando a este último como la **cara**, se tiene que (a, b, c) y (b, c, d) cumplen con las condiciones para ser **orejas**, por lo que son eliminados:

$$(a, b, e)|(a, b, c, d)$$

Para el conjunto (a, b, e), la variable e **NO está presente en ningún otro**, mientras que las otras 2 están presentes en (a, b, c, d). Tomando a este último como su **cara**, se tiene que el conjunto (a, b, e) también cumple con ser oreja, por lo que es cortado:

$$(a, b, c, d)$$

Como se logró reducir el hiper-grafo a 1 sola hiper-arista por medio de eliminación de orejas, se tiene que la consulta es **acíclica**. Esto implica que la consulta **tiene un Árbol de Join** (demostrado en clases).

Un posible Árbol de Join es el siguiente:

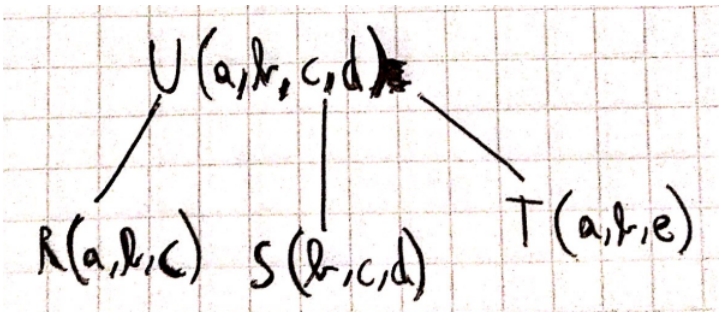

Figura 3: Posible Árbol de Join

1.2. Consulta Conjuntiva Simplificada

Al realizar el **Natural Join** entre Q y R, se forma una nueva **consulta conjuntiva de Joins** Q', que incluye a la nueva relación R. Al analizar los conjuntos de variables que pertenecen a cada hiper-arista, se obtiene lo siguiente:

$$(c : c \in Q) \cup (a_1, ..., a_n)$$

El término de la izquierda corresponde al **conjunto de todos los conjuntos de variables que pertenecen a cada hiper-arista en la consulta Q original**, mientras que el de la derecha es el nuevo conjunto de variables que corresponde a la hiper-arista de la relación R.

Al agregar esta nueva hiper-arista, ocurre la siguiente particularidad:

$$c \subseteq (a_1, ..., a_n), \forall c \in Q$$

Esto ocurre por enunciado, ya que la relación R posee la unión de todas las variables presentes en cada relación de la consulta Q. Gracias a esta propiedad, si consideramos al nuevo conjunto $(a_1, ..., a_n)$ como la **cara**, entonces **TODOS los otros conjuntos de variables corresponderán a orejas**, puesto que todas sus variables están en $(a_1, ..., a_n)$.

Finalmente, al **eliminar todas las orejas**, sólo queda el conjunto asociado a la nueva relación R, por lo tanto se logró reducir el hiper-grafo a 1 sola hiper-arista por medio de eliminación de orejas, y se tiene que la consulta es siempre **acíclica** (demostrado en clases).

NOMBRE: Benjamín Farías Valdés

N.ALUMNO: 17642531

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación

IIC3413 — Implementación de Sistemas de Base de Datos — 1' 2020

Control 3

Doy mi palabra que la siguiente solución de la pregunta 2 fue desarrollada y escrita individualmente por mi persona según el código de honor de la Universidad.

Figura 4: Firma

2. Optimización de Consultas

2.1. Plan Lógico

El primer paso es fijarse en las 3 partes principales de la consulta:

- **SELECT:** En la parte de arriba del árbol se debe realizar una proyección sobre los atributos x, z y u de la relación A. Además, se deben eliminar los duplicados (debido al **DISTINCT**).
- **FROM:** Las tuplas son obtenidas desde la relación A.
- **WHERE:** Se debe aplicar una selección con 2 condiciones, donde la primera es simple y la segunda es una consulta por medio del operador **IN**, que además es **correlacionada**.

Considerando todo esto, el primer bosquejo de la consulta se ve a continuación:

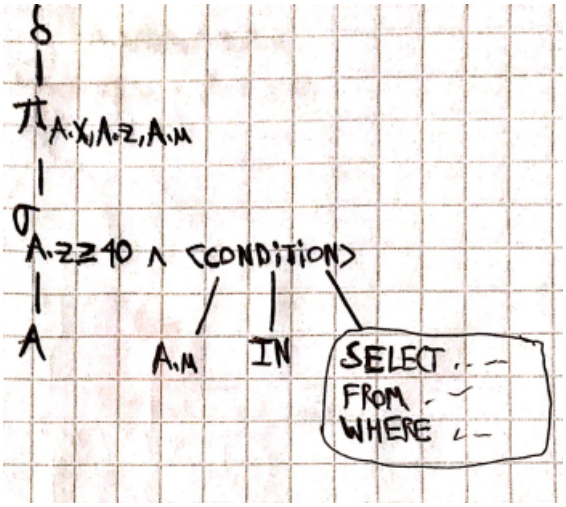

Figura 5: Paso 1 del Plan Lógico

Ahora nos enfocaremos en la parte de la condición compleja. Primero se debe pasar la **sub-consulta SQL a álgebra relacional**. Esta resulta ser la simple proyección de una selección con predicados de igualdad sobre atributos de las relaciones A, B y C. Por lo tanto, se realiza un producto cruz entre B y C, de forma de juntar sus atributos. Esto queda como se ve a continuación:

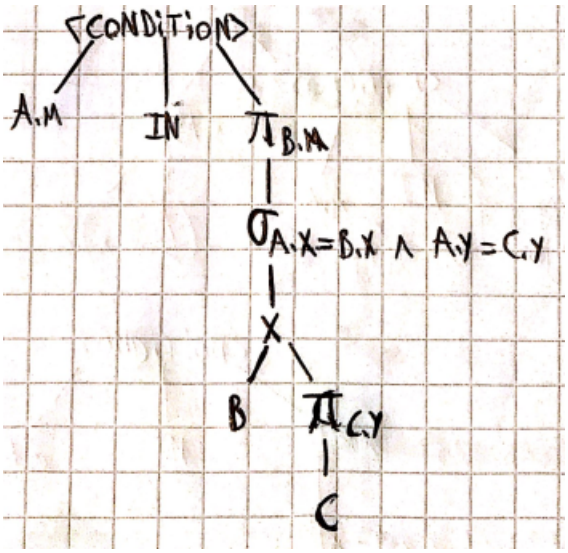

Figura 6: Paso 2 del Plan Lógico

Ahora se debe modificar la condición compleja para que la relación A quede dentro de ella. Esto se realiza mediante 2 pasos:

- **1)** Como la sub-consulta corresponde a un **conjunto de valores del atributo** $B.u$, se deben eliminar sus duplicados.
- **2)** El **IN** se puede formular aplicando un producto cruz entre A y el set de valores $B.u$, para posteriormente filtrar las tuplas que cumplan con $A.u = B.u$ (esto implica que se cumplió la condición de

pertenencia al conjunto). Además, se añade la condición simple $A.z \geq 40$ a la selección, ya que estaba presente en la estructura original.

Tras estos cambios, el plan queda como sigue:

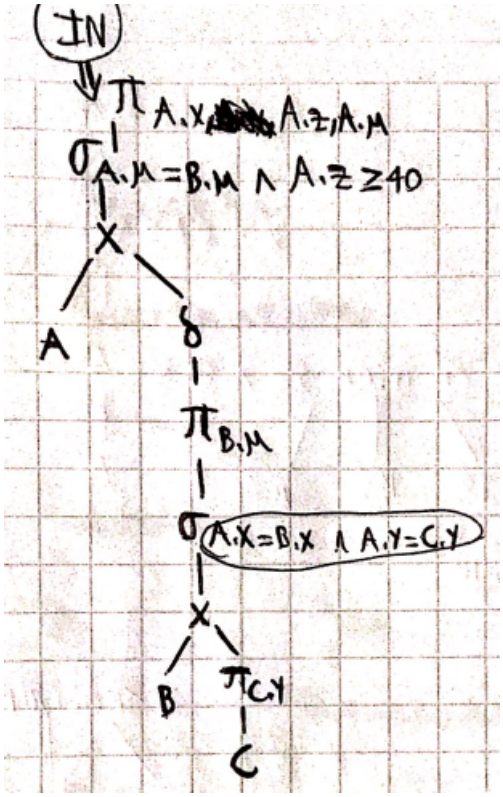

Figura 7: Paso 3 del Plan Lógico

Aún falta resolver el problema de la sub-consulta **correlacionada**, que requiere acceso a las tuplas de A para su condición **WHERE**. Esto se resuelve agregando la relación A al producto cruz entre B y C:

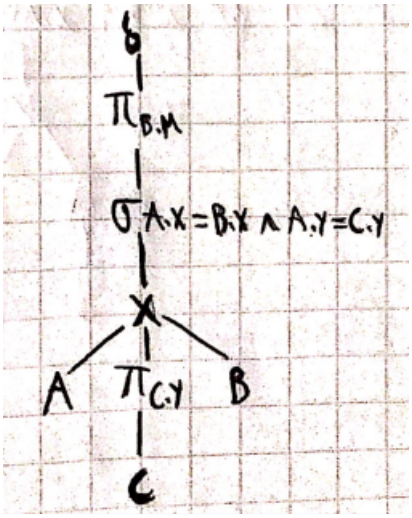

Figura 8: Paso 4 del Plan Lógico

Finalmente, aplicando todos estos pasos, se obtiene el siguiente plan lógico inicial:

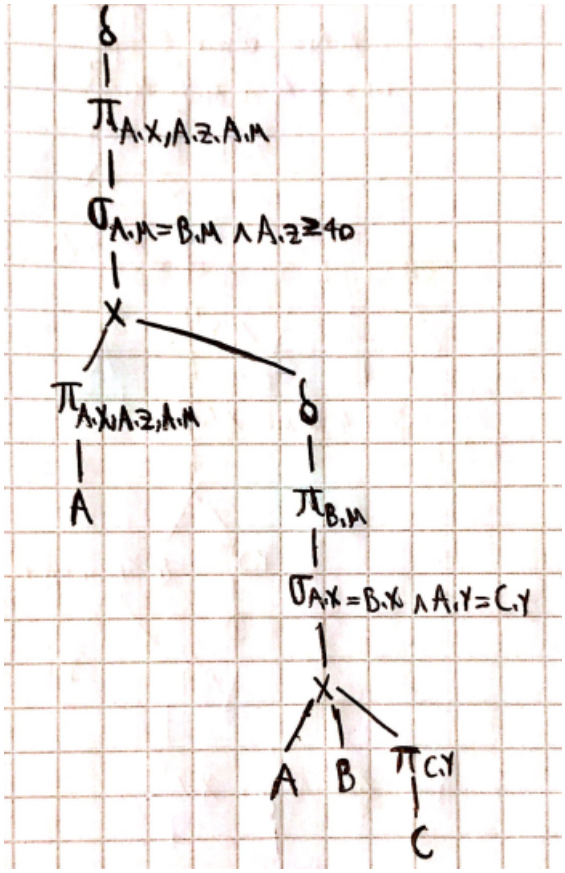

Figura 9: Primer Plan Lógico

2.2. Reescritura de Consultas

Se aplicaron las siguientes reglas de reescritura de consultas:

- **Push de las Selecciones:** Se movió la selección de la condición $A.z \geq 40$ antes de la realización del producto cruz. De esta forma, se logran filtrar muchas tuplas de A para reducir el costo de realizar el producto cruz (posteriormente será un Join). Además, se aprovecha el índice **B+Tree sobre el atributo A.z**, de forma que este filtrado es muy rápido.
- **Convertir los productos en Joins:** Se transformaron ambos productos cruz (más la selección posterior) en Equi-Joins, **reduciendo sustancialmente su costo**.
- **Proyecciones Adicionales:** Se agregó una proyección para los atributos x e y de la relación A antes de entrar al Join con la relación B en la rama derecha. De esta forma se ahorra **memoria** hacia arriba. En el plan lógico original ya se había utilizado **Push de Proyecciones**, por lo que no se aplicó en estas modificaciones.
- **Eliminación de Duplicados:** Se eliminaron los duplicados presentes después de realizar la selección $A.z \geq 40$ con el índice, de forma de **disminuir el costo del Join que le sigue, y cumplir con el DISTINCT de la consulta**. Esto se realizó ya que probablemente aparecerán duplicados al eliminar el atributo $A.y$ en la proyección de las tuplas de A de más abajo. Se decidió colocar este operador **después del filtro mediante el índice de** $A.z$, ya que así se pueden filtrar muchas tuplas con el **B+Tree** primero, reduciendo el costo de la eliminación de duplicados.

Gracias a todas estas reglas de reescritura, se obtuvo un mejor plan lógico, mostrado a continuación:

Figura 10: Plan Lógico Mejorado