

Ejercicio 1: Dadas las siguientes tablas:

Tabla	Bloques	Registros	Índice
cliente(<u>cld</u> , cNombre, calle, ciudad)	10.000	100.000	cld ¹ , ciudad ²
depósito(<u>cld</u> , <u>dfecha</u> , <u>dhora</u> , dMonto)	50.000	1.000.000	(dfecha, dhora, cld) ¹ , cld ²

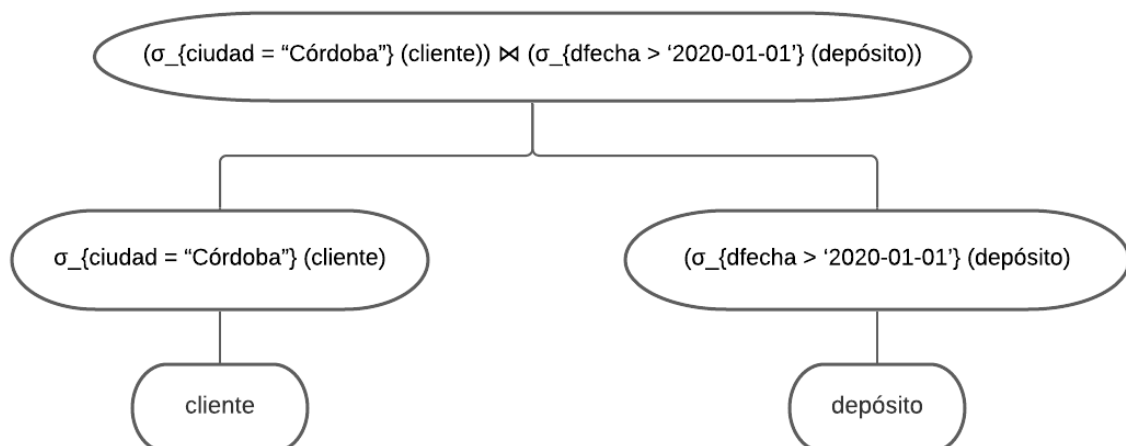
Asuma que es fin de año, que los depósitos están equi-distribuidos por año y por cliente, y que comenzaron hace 10 años. Asuma además que hay 100 ciudades con igual cantidad de clientes. Además, en la memoria entran 12 bloques. Para simplificar, puede asumir que los bloques están llenos y los índices son densos.

Indique el costo total de procesamiento en transferencias de bloques de las siguientes consultas, aclarando los algoritmos que utiliza y por qué, y los costos intermedios. Si falta información, puede preguntar, asumirla, o utilizar una variable.

- $(\sigma_{\{ciudad = \text{"Córdoba"}\}}(cliente)) \bowtie (\sigma_{\{dfecha > '2020-01-01'\}}(depósito))$.
- $(\cup(\Pi_{\{cid, dfecha\}}(\sigma_{\{dfecha > '2020-01-01'\}}(depósito)))) \times [{"OK"}]$

• Fase 1: decidir el plan de ejecución

- Armar el árbol binario de ejecución
- Calcular el factor de selectividad para selecciones y reuniones (selectivas y naturales).
- Decidir operadores físicos.
 - Solo se usan índices si la tabla de la BD lo amerita.



$r = \sigma_{\{ciudad = \text{"Córdoba"}\}} (\text{cliente})$

$fs(A = c, r) = 1/V(A, r)$ (UNIFORMIDAD)

$fs(\text{cliente.ciudad} = \text{"Córdoba"}, \text{cliente}) = 1/100$

$s = \sigma_{\{dfecha > \text{'2020-01-01'}\}} (\text{depósito})$

$fs(A \geq c, r) = (\max(A, r) - c) / (\max(A, r) - \min(A, r))$

$fs(dfecha > \text{'2020-01-01'}, \text{depósito}) = (3650 - 3285) / (3650 - 1 + 1) = 1/10$

$fs(r.A = s.B, r, s) = 1 / \max(V(A, r), V(B, s))$ (UNIFORMIDAD)

$fs(\text{cliente} = \text{depósito}, r, s) = 1/100.000$

Elegimos 100.000 porque todos los cid en la tabla de cliente es Primary key, entonces el cid nunca va a ser más grande en depósito que en cliente.

Algoritmo para índice primario usando árbol B+ con igualdad en clave candidata:

- Recorrer la altura del árbol más una E/S para recoger el registro.
- Cada una de estas operaciones requiere un acceso a bloque y una transferencia de bloque.
- Costo = $(h_i + 1) * (t_T + t_S)$,
- donde h_i denota la altura del índice.

Altura del árbol B+ = $\lceil \log_{n/2}(K) \rceil$ donde K cantidad de valores de la clave de búsqueda en la tabla. Y n cantidad de entradas por nodo.

K = 100. Supongamos que una ciudad ocupa 50 B, un puntero 6B. Un bloque tiene 4 KiB. Esto da 73 entradas por bloque. Así n = 73.

Altura del árbol B+ = $\lceil \log_{n/2}(K) \rceil = \lceil \log_{73/2}(100) \rceil = \lceil \log_{37}(100) \rceil = \lceil \log(100)/\log(37) \rceil = \lceil 2/1,56 \rceil = \lceil 1,28 \rceil = 2$.

K = 100000

suponer

$h_i = \lceil \log_{n/2} k \rceil = \lceil \log_6 100000 \rceil = \lceil 6.4254860446923 \rceil = 7$

Para la selección: algoritmo de búsqueda lineal

Cantidad de bloques que ocupa cada tabla

$B_{\text{cliente}} = 10000$

$B_{\text{depósito}} = 50000$

$r' = \sigma_{\text{cliente.ciudad} = \text{"Córdoba"}} (\text{cliente})$

$r'' = \sigma_{\text{dfecha} > \text{'2020-01-01'}} (\text{depósito})$

$s = r' \bowtie r''$

$|r'| = |\text{fs}(\text{cliente.ciudad} = \text{"Córdoba"}, \text{cliente})| = 1/100 * 100.000 = 1.000$

$|r''| = |\text{fs}(\text{dfecha} > \text{'2020-01-01'}, \text{depósito})| = (10 - 1)/(10-1+1) = 1 / 10 * 1.000.000 = 100.000$

$|s| = |r'| * |r''| * \text{fs}(\text{cliente} = \text{depósito}, r', r'') = 1000 * 100.000 * 1/1.000.000 = 100$

Ejercicio 4: Realice las mismas tareas que las indicadas en el ejercicio 2 para las siguientes consultas:

a) $\Pi_{\text{legajo, profe.nombre}} (\sigma_{\text{materia.nombre} = \text{"Intro a los Algoritmos"}} (\text{profe legajo } \bowtie \text{ a cargo materia}))$

b) $\Pi_{\text{legajo, profe.nombre}} (\text{profe legajo } \bowtie \text{ a cargo } \sigma_{\text{materia.nombre} = \text{"Intro a los Algoritmos"}} (\text{materia}))$

Anote en los árboles generados el costo (en cantidad de bloques leídos y escritos) de cada operación.

La tabla materia tiene 100 registros, ocupando 20 bloques, mientras que profe tiene 2.000 registros, ocupando 500 bloques. Considere los diferentes escenarios según la existencia de índice (ninguno, primario, o secundario) en las columnas relevantes de las tablas.

Recuerde que la tabla temporal que escribe un operador no tiene índices.

$\Pi_{\text{legajo, profe.nombre}}$

Ejercicio 2: Sea la BD con las siguientes tablas:

persona(DNI, nombre, edad)

bibliotecario(DNI, antigüedad)

trabajaEn(DNI, nombreBib, rol)

Sea la siguiente consulta:

$\Pi_{\text{nombre, DNI, nombreBib}} (\sigma_{\text{antigüedad} > 5 \text{ and rol} = \text{'bibliotecario'}} (\text{Persona } \bowtie \text{bibliotecario } \bowtie \text{trabajaEn}))$

Asumir que el optimizador sigue el siguiente orden:

1. Optimización por costo usando programación dinámica para las reuniones naturales asumiendo la tabla de abajo.
2. Transformar la consulta obtenida en el paso anterior aplicando optimización heurística de filminas 46 y 47.

persona	bibliotecario	trabajaEn
1000 tuplas	400 tuplas	1500 tuplas
$V(\text{persona}, \text{edad}) = 60$	$V(\text{bibliotecario}, \text{antigüedad}) = 40$	$V(\text{trabajaEn}, \text{rol}) = 5$
$V(\text{persona}, \text{nombre}) = 900$		$V(\text{trabajaEn}, \text{DNI}) = 750$
		$V(\text{trabajaEn}, \text{nombreBib}) = 200$

No se pide elegir algoritmos más adecuados para operadores de la consulta obtenida.

1)

	Persona	bibliotecario	trabajaEn
--	---------	---------------	-----------

size	1000	400	1500
cost	0	0	0
best plan	persona	bibliotecario	trabajaEn

$| \text{persona} \bowtie \text{bibliotecario} | = | \text{persona} | * | \text{bibliotecario} | * 1/\max(V(\text{persona}, \text{DNI}), V(\text{bibliotecario}, \text{DNI})) = 1000 * 400 * 1/1000 = 400$

$| \text{bibliotecario} \bowtie \text{trabajaEn} | = | \text{bibliotecario} | * | \text{trabajaEn} | * 1/\max(V(\text{bibliotecario}, \text{DNI}), V(\text{trabajaEn}, \text{DNI})) = 400 * 1500 * 1/750 = 800$

$| \text{trabajaEn} \bowtie \text{persona} | = | \text{trabajaEn} | * | \text{persona} | * 1/\max(V(\text{trabajaEn}, \text{DNI}), V(\text{persona}, \text{DNI})) = 1500 * 1000 * 1/1000 = 1500$

	persona, bibliotecario	bibliotecario, trabajaEn	trabajaEn, persona
size	400	800	1500
cost	0	0	0
best plan	persona \bowtie bibliotecario	bibliotecario \bowtie trabajaEn	trabajaEn \bowtie persona

	{ persona \bowtie bibliotecario , trabajaEn }	{ bibliotecario \bowtie trabajaEn , persona }	{ trabajaEn \bowtie persona , bibliotecario }
cost	400 + 1500 = 1900	800 + 1000 = 1800	1500 + 400 = 1900

2)

$\Pi_{\text{nombre}, \text{DNI}, \text{nombreBib}} (\sigma_{\text{antigüedad} > 5 \text{ and } \text{rol} = \text{'bibliotecario'}} ((\text{bibliotecario} \bowtie \text{trabajaEn}) \bowtie \text{Persona})$

$\Pi_{\text{nombre}, \text{DNI}, \text{nombreBib}} (((\sigma_{\text{antigüedad} > 5} \text{ bibliotecario} \bowtie \sigma_{\text{rol} = \text{'bibliotecario'}} \text{ trabajaEn}) \bowtie \text{Persona})$

$((\Pi_{\text{DNI}} (\sigma_{\text{antigüedad} > 5} \text{ bibliotecario}) \bowtie \Pi_{\text{DNI}, \text{nombreBib}} (\sigma_{\text{rol} = \text{'bibliotecario'}} \text{ trabajaEn})) \bowtie \Pi_{\text{DNI}, \text{nombre}} \text{ Persona})$

Ejercicio 3: Sea $R = (A, B, C, D, E, G)$

Con $F = \{AB \rightarrow C; AC \rightarrow B; AD \rightarrow E; B \rightarrow D; BC \rightarrow A; E \rightarrow G\}$

Se pide:

1. Escribir deducción de $AC \rightarrow G$ a partir de F
2. Dar una dependencia que no está en F^+ . Justifique su respuesta.
3. Listar claves candidatas de F . Probar para una de ellas que es clave candidata.
4. Sea el esquema relacional $R = (A, B, C, D, E)$ con el conjunto de dependencias funcionales:

$$F = \{AB \rightarrow C, AB \rightarrow E, A \rightarrow B, C \rightarrow D\}.$$

Calcule el cubrimiento canónico F_C . En su respuesta tiene que estar justificado que efectivamente es un cubrimiento canónico de F .

Aplicando estas reglas *repetidamente*, se puede hallar todo F^+ , dado F . Este conjunto de reglas se denomina **axiomas de Armstrong** en honor de la persona que las propuso por primera vez.

- **Regla de la reflexividad.** Si α es un conjunto de atributos y $\beta \subseteq \alpha$, entonces se cumple que $\alpha \rightarrow \beta$.
- **Regla de la aumentatividad.** Si se cumple que $\alpha \rightarrow \beta$ y γ es un conjunto de atributos, entonces se cumple que $\gamma\alpha \rightarrow \gamma\beta$.
- **Regla de la transitividad.** Si se cumple que $\alpha \rightarrow \beta$ y también se cumple que $\beta \rightarrow \gamma$, entonces se cumple que $\alpha \rightarrow \gamma$.
- **Regla de la unión.** Si se cumple que $\alpha \rightarrow \beta$ y que $\alpha \rightarrow \gamma$, entonces se cumple que $\alpha \rightarrow \beta\gamma$.
- **Regla de la descomposición.** Si se cumple que $\alpha \rightarrow \beta\gamma$, entonces se cumple que $\alpha \rightarrow \beta$ y que $\alpha \rightarrow \gamma$.
- **Regla de la pseudotransitividad.** Si se cumple que $\alpha \rightarrow \beta$ y que $\gamma\beta \rightarrow \delta$, entonces se cumple que $\alpha\gamma \rightarrow \delta$.

Con $F = \{$

$AB \rightarrow C;$

$AC \rightarrow B;$

$AD \rightarrow E;$

$B \rightarrow D;$

$BC \rightarrow A;$

$E \rightarrow G\}$

Se pide:

1. Escribir deducción de $AC \rightarrow G$ a partir de F

1. $AC \rightarrow B, B \rightarrow D$

{transitividad}

$AC \rightarrow D$

{aumentatividad}

$AAC \rightarrow AD$

{trivialmente}

$AC \rightarrow AD$

2. $AD \rightarrow E, E \rightarrow G$

{transitividad}

$AD \rightarrow G$

Por 1, por 2 y {transitividad} $AC \rightarrow G$

La conmutatividad legal y válida por duran:

$AC \rightarrow B \Leftrightarrow CA \rightarrow B$

2. Justificar usando la regla del cierre

- Se puede comprobar si se cumple la dependencia funcional $\alpha \rightarrow \beta$ (o, en otras palabras, si se halla en F^+), comprobando si $\beta \subseteq \alpha^+$. Es decir, se calcula α^+ empleando el cierre de los atributos y luego se comprueba si contiene a β . Esta prueba resul-

Por ejemplo, $G \rightarrow A$ no está en F , ya que $G^+ = \{G\}$, luego A no está contenido en G^+ y por lo tanto no pertenece a F^+ .

3.

- Comprobar si α es una superclave, se calcula α^+ y se comprueba si α^+ contiene todos los atributos de R .

Una clave candidata es un conjunto de atributos que si le saco un atributo deja de ser superclave.

Probar encontrar una superclave al azar, si no es superclave, le agrego el atributo que falta para que lo sea. Luego, ya sé que esa es una clave candidata. (Solución peruana de Mati)

Dato: Las superclaves suelen ser pares (nunca un elemento solo)

Ver si AB es superclave. $R = (A,B,C,D,E,G)$

$AB \rightarrow C$ {aumentatividad} $AB \rightarrow AC$ {descomposición} $AB \rightarrow A$ ✓, $AB \rightarrow C$

$AB \rightarrow C$ {aumentatividad} $AB \rightarrow CB$ {descomposición} $AB \rightarrow C$, $AB \rightarrow B$ ✓

Por definición, $AB \rightarrow C$ ✓

$AB \rightarrow B$, $B \rightarrow D$ {transitividad} $AB \rightarrow D$ ✓

$AB \rightarrow A$ {aumentatividad} $ABD \rightarrow AD$, $AD \rightarrow E$ {transitividad} $ABD \rightarrow E$ ✓

$ABD \rightarrow E$, $E \rightarrow G$ {transitividad} $ABD \rightarrow G$ ✓

$AB^+ = (A,B,C,D,E) = R$

Luego, como AB es superclave y si se le quitara un atributo deja de ser superclave, entonces AB es clave candidata.

AC y BC también son claves candidatas.

4.

```

Fc = F
repeat
    Utilizar la regla de unión para sustituir las dependencias de Fc de la forma
         $\alpha_1 \rightarrow \beta_1$  y  $\alpha_1 \rightarrow \beta_2$  con  $\alpha_1 \rightarrow \beta_1 \beta_2$ .
    Hallar una dependencia funcional  $\alpha \rightarrow \beta$  de Fc con un atributo raro en  $\alpha$  o en  $\beta$ .
        /* Nota: la comprobación de los atributos raros se lleva a cabo empleando Fc, no F */
    Si se halla algún atributo raro, hay que eliminarlo de  $\alpha \rightarrow \beta$ .
until Fc ya no cambie.

```

FIGURA 7.8. Cálculo del recubrimiento canónico.

Se dice que un atributo de una dependencia funcional es **raro** si se puede eliminar sin modificar el cierre del conjunto de dependencias funcionales. La definición formal de los **atributos raros** es la siguiente. Considérese un conjunto F de dependencias funcionales y la dependencia funcional $\alpha \rightarrow \beta$ de F .

- El atributo A es raro en α si $A \in \alpha$ y F implica lógicamente a $(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$.
- El atributo A es raro en β si $A \in \beta$ y el conjunto de dependencias funcionales $(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\}$ implica lógicamente a F .

$F_c = F = \{AB \rightarrow C, AB \rightarrow E, A \rightarrow B, C \rightarrow D\}$

$F_c^+ = (A, B, C, D, E)$

it 1:

$AB \rightarrow C, AB \rightarrow E \{union\} AB \rightarrow CE$

$F_c = \{AB \rightarrow CE, A \rightarrow B, C \rightarrow D\}$

B es raro en AB ya que $A^+ = (A, B, C, D, E)$ y $CE \subseteq A^+$

1. $A \rightarrow A$ trivialmente

2. $A \rightarrow B$

3. $A \rightarrow AB$ {por union 1 y 2}

4. $A \rightarrow E$ { $A \rightarrow AB$ y $AB \rightarrow CE$, transit $A \rightarrow CE$, {descomp} $A \rightarrow C$ y $A \rightarrow E$ }

5. $A \rightarrow C$, {transitividad $C \rightarrow D$ } $A \rightarrow D$

$F_c = \{A \rightarrow CE, A \rightarrow B, C \rightarrow D\}$

$A \rightarrow CE, A \rightarrow B \{union\} A \rightarrow CEB$

$F_c = \{A \rightarrow CEB, C \rightarrow D\}$ es el recubrimiento canónico.

- Para probar si un atributo $A \in \beta$ es raro en β

3. Computar α^+ usando solo las DF en

$$G = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - A)\},$$

4. Chequear que $A \in \alpha^+G$; si es cierto, entonces A es raro.

C raro de $A \rightarrow CEB$?

Quiero ver que $F_c - \{A \rightarrow CEB\} \cup \{A \rightarrow EB\}$ implica F_c

La union es $G = \{A \rightarrow EB, C \rightarrow D\}$

$A \rightarrow A$

$A \rightarrow A$ y $A \rightarrow EB$ entonces tengo $A \rightarrow E$ y $A \rightarrow B$ (descomposición)

$A \rightarrow E$ {aumentatividad} $AC \rightarrow EC$ {descomposición} $AC \rightarrow E, AC \rightarrow C$, {transitividad} $AC \rightarrow D$ ¿ES LEGAL ESTO?

$F_c = \{A \rightarrow EB, C \rightarrow D\}$ es el recubrimiento canónico.

Ejercicio 4: Sea el esquema relacional $R = (A, B, C, D)$ con el conjunto de dependencias funcionales: $F = \{A \rightarrow C; AD \rightarrow B; AB \rightarrow D\}$.

- Encontrar una descomposición de R en FNBC utilizando el algoritmo de descomposición. Justificar que cada dependencia funcional utilizada sea testigo y que la descomposición encontrada quedó en FNBC.
- Considere la descomposición $R_1 = (A, C), R_2 = (A, B, D)$. ¿Está en 3FN? Justificar.