Segundo Parcial de Fundamentos de Bases de Datos
SOLUCION

Noviembre 2017

Duración: 3 horas y media

Presentar la resolución del parcial:

- Con las hojas numeradas y escritas de un solo lado. Comience cada ejercicio en una hoja nueva.
- 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.**

Ejercicio 1 (22 puntos)

Una empresa financiera emite tarjetas de crédito o de débito y ofrece promociones de descuentos en diferentes locales. Cada tarjeta tiene un número que la identifica (idT), una marca (mT), un tipo (tT- por ejemplo crédito o débito) y pertenece a un único cliente (cT).
A cada local que ofrece promociones se le asigna un identificador (idL). De cada uno de estos locales se conoce la dirección (uL) y los rubros en los que trabaja (rL- por ejemplo: vestimenta femenina, accesorios, cine, teatro, restaurante, etc.). Cada local puede trabajar en más de un rubro.
De las promociones se conoce una identificación (idP), las marcas y tipos de tarjeta que pueden usarse (por ejemplo, válida para VISA débito y American Express crédito), los locales que ofrecen esta promoción, los rubros válidos en cada local para esa promoción y el porcentaje de descuento (descP) que tiene la promoción en cada local y rubro. Una promoción puede aplicar a varias marcas y tipos de tarjeta, así como a varios rubros y/o locales, con descuentos que pueden ser diferentes para cada tarjeta, tipo de tarjeta, local y rubro.
Además se lleva control de los usos de cada tarjeta en cada promoción, en donde se incluye además de la promoción, el local y el rubro, el monto total (montoU) consumido en ese uso, la tarjeta usada, y la fecha (fechaU - incluye la hora) de la utilización.

Parte a) Escriba las dependencias funcionales que existen en la realidad.

Solución Parte a)

Las dependencias funcionales que se desprenden de la realidad son las siguientes:
Acerca de las tarjetas: (df1) idT→ mT,tT,cT
Acerca de los locales: (df2) idL→ uL
Acerca de las promociones: (df3) idP,mT,tT,rL, idL →descP

Acerca de los usos de las promociones:

Si se asume que en cierto momento (fechaU) y local, un cliente puede utilizar una tarjeta y promoción sobre diferentes rubros, y para cada rubro tengo un monto diferente, la dep que surge es:

(df4) idP,idL,rL,fechaU,idT→ montoU

También podría asumirse que cada uso corresponde a un único rubro, en cuyo caso la dependencia es:
(df4') idP,idL,fechaU,idT→ montoU,rL

Por último, dado que una persona no puede usar su tarjeta en dos locales al mismo tiempo:

(df5) fechaU,idT→idL

Parte b)
Suponga que la versión actual del sistema, utiliza las tablas siguientes para representar las promociones y su uso:

```
Uso(idP, idT, fechaU, idL, rL, montoU)
Promo(idP, mT, tT, idL, rL, descP)
```

1. Indique qué dependencias funcionales se cumplen en cada una de las tablas `Uso` y `Promo`.
2. Indique qué dependencias multivaluadas no funcionales se cumplen en cada una de estas tablas. Justifique tanto si encuentra alguna como si no encuentra ninguna.
3. Determine todas las claves de las tablas `Uso` y `Promo`. Justifique.
4. Indique la máxima forma normal en que están las tablas `Uso` y `Promo`.

Solución Parte b)

1) Proyección de dfs sobre las tablas Uso y Promo

Sean F y F' las dos opciones manejadas en la sol de la parte a)
F = {idT → mT,tT,cT; idL → uL; idP,mT,tT,rL,idL → descP; idP,idL,rL,fechaU,idT → montoU;
fechaU,idT → idL }
F' = {idT → mT,tT,cT; idL → uL; idP,mT,tT,rL,idL → descP; idP,idL,fechaU,idT → montoU,rL;
fechaU,idT → idL }

Considerando F
Π_F(Promo) = { idP,mT,tT,rL,idL → descP }
Π_F(Uso) = {idP,idL,rL,fechaU,idT → montoU; fechaU,idT → idL }

Considerando F'
$\Pi_{F'}$(Promo) = Π_F(Promo)
$\Pi_{F'}$(Uso) = {idP,idL,fechaU,idT → montoU,rL; fechaU,idT → idL }

2) Dependencias Multivaluadas

Las dependencias multivaluadas que interesa reflejar son las que surgen de la realidad pero no de otra forma. Es por esto que no se consideran dependencias multivaluadas triviales, ni dependencias multivaluadas que se infieran de las funcionales ya sea por replicación o complemento.

Una posible candidata a multivaluada a considerar es idL ->>rL.

Si se considera en la tabla Promo, para el mismo local, deberían combinarse todos los rubros con el resto de los datos asociados a ese local. Eso hace que debieran combinarse todos loa rubros con todos los descuentos (entre otros datos). La única forma en que esto sería posible es que todos los descuentos en cada local fuesen el mismo, independientemente del rubro, pero se sabe que eso no sucede. Por lo tanto, la dependencia no se cumple en Promo.

Un razonamiento similar se puede aplicar en Uso, en donde para el mismo local, debería combinarse el rubro con las fechas y montos (entre otros). Esto obligaría a que cada local atendiera clientes en una única fecha.

3) Claves:

Consideremos las deps proyectadas en la parte b1).

Considerando F
Claves de Promo:
La única clave de Promo es (idP,mT,tT,rL,idL) dado que en la clausura de ese conjunto de atributos (que son el lado izq de la única df) están todos los atributos de la tabla.

Claves de Uso:
Los atribs (idP,rL,fechaU,idT) están en toda clave porque no aparecen nunca a la derecha

$(idP,rL,fechaU,idT)^+$ según $\Pi_F(Uso)$ = {idP,rL,fechaU,idT,idL,montoU} = Uso, por lo que (idP,rL,fechaU,idT) es la única clave en la tabla Uso

Considerando F'
Claves de Promo:
La única clave de Promo es (idP,mT,tT,rL,idL) por las mismas razones que si consideramos el conj F.

Los atribs (idP,fechaU,idT) están en toda clave porque no aparecen nunca a la derecha
$(idP,fechaU,idT)^+$ según $\Pi_{F'}(Uso)$ = { idP,fechaU,idT, idL,montoU,rL} = Uso, por lo que (idP,fechaU,idT) es la única clave en la tabla Uso

4) Formas normales:
Promo está en 4NF, porque no hay DMVs no triviales y la única DF que se proyecta tiene una superclave del lado izquierdo.

Uso está en 1NF tanto si considero a F como a F', ya que fechaU,idT→idL genera una dependencia parcial de clave que viola 2NF.

Parte c)

Luego de varios años de trabajar de esta forma, la empresa decide funcionar de una forma diferente para simplificar algunos procesos. Una de las nuevas reglas impone que en cada promoción se usará el mismo descuento en un mismo local, para cualquier rubro.
Para resolver esto, el equipo de desarrollo decide agregar nuevas tablas `Uso2` y `Promo2` que tienen exactamente la misma estructura que las tablas `Uso` y `Promo`. Éstas últimas no se borran, pero se deja de insertar registros a partir de que entran en vigencia estos cambios.

1. Indique qué dependencias funcionales se cumplen en cada una de las tablas `Uso2` y `Promo2`.
2. Indique qué dependencias multivaluadas se cumplen en cada una de estas tablas. Justifique tanto si encuentra alguna como si no encuentra ninguna.
3. Determine todas las claves de las tablas `Uso2` y `Promo2`. Justifique.
4. Indique la máxima forma normal en que están las tablas `Uso2` y `Promo2`.
5. Si es posible, encuentre una descomposición para las tablas `Uso2` y `Promo2` en una forma normal más avanzada que la indicada en el punto anterior, aplicando los algoritmos vistos en clase.

1) Proyección de dfs

Como el rubro no es necesario para determinar el descuento la única df que se modifica es la df3. Los nuevos conjuntos de dfs son

F = {idT→mT,tT,cT; idL→uL; idP,mT,tT,idL→descP; idP,idL,rL,fechaU,idT→ montoU; fechaU,idT→idL }
F' = {idT→mT,tT,cT idL→uL; idP,mT,tT,idL→descP; idP,idL,fechaU,idT→ montoU,rL;fechaU,idT→idL }

Considerando F
$\Pi_F(Promo2)$ = { idP,mT,tT,idL→descP }
$\Pi_F(Uso2)$ = {idP,idL,rL,fechaU,idT→ montoU; fechaU,idT→idL }

Considerando F'
$\Pi_{F'}(Promo2)$ = $\Pi_F(Promo2)$
$\Pi_{F'}(Uso2)$ = {idP,idL,fechaU,idT→ montoU,rL; fechaU,idT→idL }

2) Dependencias Multivaluadas

La tabla Promo2 es la forma en que los diseñadores tratan de reflejar las nuevas reglas de negocio (la realidad) en la base de datos. La nueva realidad implica que la relación entre los locales y los rubros y los locales y los descuentos son totalmente independientes entre sí, es decir, el descuento en un local es independiente de los rubros en los que trabaja ese local. Sin embargo, la nueva realidad no cambia nada con

respecto a la dependencia (de hecho es una funcional) del descuento desde la promoción, el tipo de tarjeta, etc.Es por eso que se cumple la siguiente dependencia multivaluada:

idP,mT,tT,idL->> rL

Pero esta dependencia, es la complementaria de la funcional, por lo que no hay porqué considerarla.

3) Claves:

Consideremos las deps proyectadas en la parte c1).

Claves de Promo2:
Los atributos (idP,mT,tT,rL,idL) están en toda clave porque no aparecen nunca a la derecha, y como en la clausura de ese conjunto de atributos están todos los atributos de la tabla es la única clave (la misma que en Promo). Esto se cumple tanto en el caso de considerar F como F'

Claves de Uso2:
Como las proyecciones de F y F' sobre Uso2 son las mismas que sobre Uso, las claves son las mismas que las encontradas para la tabla Uso: (idP,rL,fechaU,idT) si considero a F y (idP,fechaU,idT) si considero a F'.

4) Formas normales:

Promo2 no está en BCNF, dado que idP,mT,tT,idL → descP no tiene una superclave del lado izquierdo (falta rL). Esto hace que la dependencia idP,mT,tT,idL,rL → descP (la clave determina descP) es parcial y sobre un atributo no primo descP, por lo que viola 2NF y por lo tanto, está en 1NF.

Uso2 está en 1NF tanto si considero a F como a F', ya que fechaU,idT → idL genera una dependencia parcial de clave que viola 2NF.

5) Descomposición de las tablas Uso2 y Promo2 en una forma normal más avanzada.

Para Promo2:
Lo que resulta más práctico es aplicar el algoritmo para llevar a 4NF y BCNF.
Como idP,mT,tT,idL → descP viola 4NF, descompongo Promo2 en:
- $Promo2_1$(idP, mT, tT, idL, descP)
- $Promo2_2$(idP, mT, tT, idL, rL)

Proyecto dfs y dmvs en cada subesquema:

$\Pi_F(Promo2_1)$ = { idP,mT,tT,idL → descP }
$\Pi_F(Promo2_2)$ = {idP,mT,tT,idL->>rL}

Ahora ambos subesquemas están en 4NF ($Promo2_1$ está en BCNF y no hay dmvs no triviales y $Promo2_2$ sólo tiene una dmv trivial y no hay dfs).

Para Uso2:
Lo que resulta más práctico es aplicar el algoritmo para llevar a BCNF.

Considerando F
Π_F(Uso2) = {idP,idL,rL,fechaU,idT → montoU; fechaU,idT → idL } única clave (idP,rL,fechaU,idT)

Como fechaU,idT → idL viola BCNF descompongo Uso2 en:
- $Uso2_1$(idP, rL, fechaU, idT, montoU)
- $Uso2_2$(fechaU,idT,idL)

Proyecto dfs en cada subesquema:

Como $(idP, rL, fechaU, idT)^+_{\text{según F}}$ = { idP, rL, fechaU, idT, idL, montoU}

$\Pi_F(Uso2_1)$ = { idP, rL, fechaU, idT → montoU}
$\Pi_F(Uso2_2)$ = { fechaU,idT→idL }

Ambos subesquemas quedan en BCNF.

Considerando F'
$\Pi_{F'}(Uso2)$ = {idP,idL,fechaU,idT→ montoU,rL fechaU,idT→idL } única clave (idP,fechaU,idT)

Como fechaU,idT→idL viola BCNF descompongo Uso2 en:
$Uso2_1$(idP, rL, fechaU, idT, montoU)
$Uso2_2$(fechaU,idT,idL)

Proyecto dfs en cada subesquema:

Como $(idP, fechaU, idT)^+_{\text{según } F'}$ = { idP, fechaU, idT, idL, montoU,rL}
$\Pi_{F'}(Uso2_1)$ = { idP, fechaU, idT → montoU, rL}
$\Pi_F(Uso2_2)$ = { fechaU,idT→idL }

Ambos subesquemas quedan en BCNF.

Ejercicio 2 (22 puntos)

Una cadena de hoteles cuenta con una base de datos donde se registran los hoteles, sus clientes y las estadías. El esquema relacional es el siguiente:

Personas (<u>ci</u>, nombre, email)
Contiene, para cada persona que se ha hospedado en alguno de los hoteles de la cadena, su documento, su nombre y email.

Hoteles (<u>nombreHotel</u>, ciudad, direccion, telefono)
Contiene información de los hoteles de la cadena. De cada hotel se conoce su nombre, la ciudad donde se encuentra, su dirección y teléfono.

Estadias (<u>nombreHotel, ci, fechaIni</u>, fechaFin, puntaje)
Contiene información de las estadías en los hoteles de la cadena, el nombre del hotel, el documento del cliente, las fechas de inicio y de fin y el puntaje otorgado por el cliente según el servicio brindado.

Además se cuenta con la siguiente información:

	Cantidad de Tuplas	Atributos	Índices
Personas	5000		Índice Primario sobre ci
Hoteles	200		Índice Primario sobre nombreHotel Indice Sec árbol B + por ciudad
Estadías	10000	Los puntajes van del 1 al 10. Se distribuyen uniformemente en la tabla	Índice Primario sobre nombreHotel, ci, fechaIni

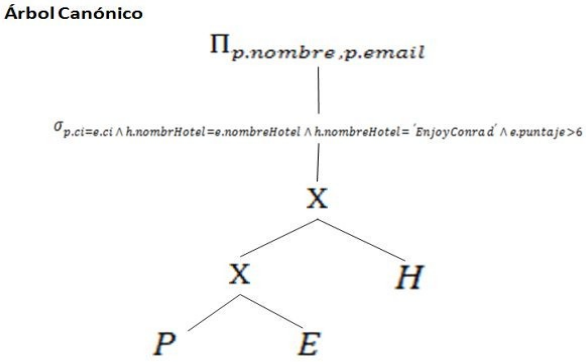
Parte a)

Considere la siguiente consulta SQL sobre esta base de datos:

```
SELECT   p.nombre, p.email
FROM     Personas p, Estadias e, Hoteles h
WHERE    p.ci = e.ci and h.nombreHotel = e.nombreHotel and
         h.nombreHotel = 'Enjoy Conrad' and e.puntaje > 7
```

1. Dar el árbol canónico correspondiente a la consulta SQL indicada.

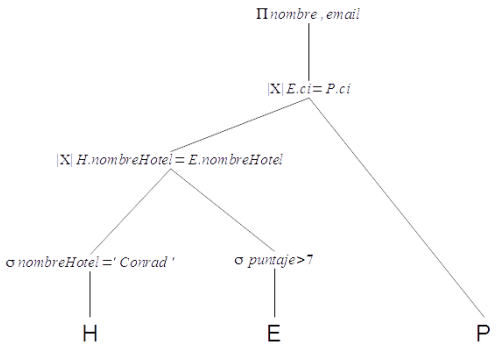
Construcción del árbol canónico, se proyectan todos los atributos que aparecen en el “select” de la consulta, y se seleccionan sólo aquellas tuplas que cumplen la condición “where”. Además, el orden de las tablas debe ser el que aparece en la consulta SQL.



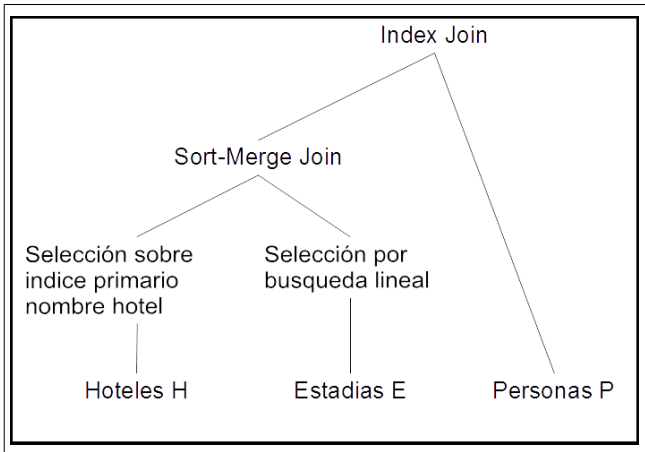
2. Partiendo del árbol canónico del paso anterior, aplicar las heurísticas vistas en el curso para obtener un plan lógico optimizado. Explicar cada paso.

Aplicar las heurísticas que le permitan llegar al plan lógico optimizado consiste en:

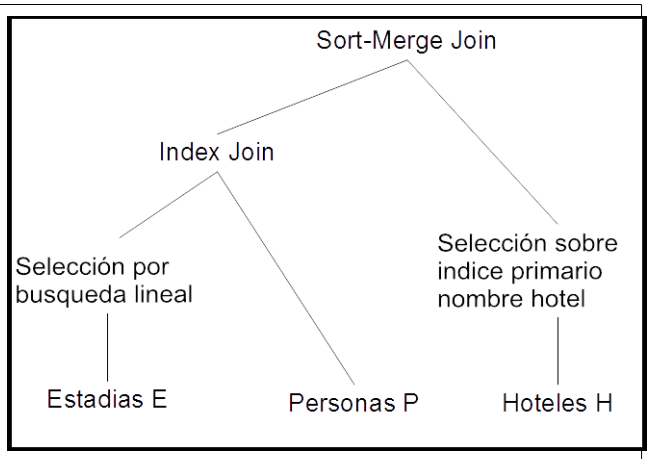
1) Se cambian las selecciones conjuntivas por una cascada de selecciones simples, moviendo dichas selecciones lo más abajo posible en el árbol.
2) Se colocan a la izquierda de los productos las hojas que generan menos tuplas, asegurando que el orden de las hojas no cause operaciones de producto cartesiano.
3) Luego, se deben cambiar las secuencias de selecciones y productos cartesianos por joins.
4) Se mueven las proyecciones lo más abajo posible en el árbol, agregando las proyecciones que sean necesarias (Opcional).
   El plan lógico que se obtiene a partir del árbol canónico, y aplicando todas las heurísticas, es el que se presenta a continuación:



3. Dados los siguientes planes físicos indique, para cada uno de ellos, si es una posible implementación para el plan lógico obtenido en el paso anterior. Justifique.



Opción 1



Opción 2

La opción 1 tiene la forma adecuada, y es posible utilizar los algoritmos especificados para cada operación, mientras que la opción 2 no concuerda con el plan lógico obtenido (observar por ejemplo el orden de las tablas)

Parte b)

1. En el contexto de la ejecución del plan físico de una consulta ¿Cuál es la diferencia entre las estrategias de implementación *Pipelined* y *No Pipelined* ? Indique, según los criterios vistos en el curso, al menos un operador que siga cada una de estas estrategias.

Pipelined: hay operadores que se ejecutan simultáneamente y pueden pasarse los resultados a medida que se generan. No necesita grabar los resultado intermedios. Ejemplo : Proyección
No Pipelined: los operadores se ejecutan secuencialmente y es necesario grabar resultados intermedios. Ejemplo: Selección y Join. La diferencia radica en la necesidad de grabar los resultados intermedios.

2. Explique cómo funciona el algoritmo y cual es su costo de lectura en el mejor caso.

Considere dos tablas A y B y un conjunto de atributos X en A y Y en B. Se desea realizar el join de estas dos tablas aplicando la condición X=Y. Como precondición para aplicar el algoritmo *Sort-Merge Join* es necesario que cada tabla esté ordenada por los atributos que participan de la condición. En caso de no tener los registros de las tablas ordenados, deberían ordenarse y agregar los costos de ordenación.
Luego, el algoritmo consiste en recorrer ambas tablas en paralelo siguiendo el orden, y combinado los registros que tienen los mismos valores para A y B. Los registros de cada tabla se exploran una sola vez cada uno para comprobar su correspondencia con los de la otra tabla.
El costo de lectura, medido en bloques, corresponde a la cantidad de bloques de A+cantidad de bloques de B (bA + bB)

Ejercicio 3 (16 puntos)

Parte a)

1. ¿Cuál es el objetivo de tener mecanismos de control de concurrencia en sistemas de bases de datos?

El objetivo es lograr que varias transacciones se ejecuten simultáneamente asegurando las propiedades ACID:

- Atomicidad (Atomicity): desde el punto de vista del resultado, o se ejecuta totalmente, o no se ejecuta.
- Consistencia (Consistency preservation): siempre lleva a la base de un estado consistente a otro estado consistente.
- Aislamiento (Isolation): una transacción no debe interferir con otra.
- Durabilidad (Durability): los cambios de una transacción confirmada deben ser permanentes en la base.

2. ¿Qué ventaja/s tiene contar con historias recuperables?

La ventaja de contar con historias recuperables es que permiten que la base de datos pase de un estado consistente a otro estado consistente.

Parte b)

Considere las siguientes transacciones:

T1: r1(X), w1(X), w1(Y), c1
T2: r2(Z), w2(Z), w2(V), c2
T3: r3(X), w3(X), w3(V), c3

1. Indique si la siguientes historias H1 y H2 son recuperables y serializables. Justifique su repuesta.

H1: r1(X), r2(Z), w1(X), w2(Z), w1(Y), w2(V), c2, c1
H2: r1(X), w1(X), r3(X), w3(X), w1(Y), w3(V), c3, c1

- T1 y T2 operan sobre ítems distintos, cada transacción lee ítems que no son grabados por la otra transacción. Por lo tanto, H1 es recuperable.
- El grafo de seriabilidad de H1 es el que se muestra a continuación, como se observa, el mismo no presenta arista, por tanto el grafo no tiene ciclos y H1 es serializable.

- T3, que hace el commit primero, lee el ítem X que es grabado por T1, que hace el commit al final. Por lo tanto, H2 no es recuperable.
- El grafo de seriabilidad de H2 es el que se muestra a continuación, como se observa, el mismo presenta una sola arista, por tanto el grafo no tiene ciclos y H2 es serializable.

2. Reescriba T1 y T2 de forma que incluyan locks de lectura/escritura, siguiendo el protocolo 2PL riguroso.

El 2PL riguroso exige que no se libere ningún lock (read o write), hasta después de terminar la transacción.

$\mathbf{T_1}$: $rl_1(X),r_1(X),wl_1(X),w_1(X),wl_1(Y),w_1(Y),c_1,u_1(X),u_1(Y)$

$\mathbf{T_2}$: $rl_2(Z),r_2(Z),wl_2(Z),w_2(Z),wl_2(V),w_2(V),c_2,u_2(Z),u_2(V)$