

Segundo Parcial de Fundamentos de Base de Datos

Noviembre 2007

Soluciones

Ejercicio 1 (10 pts)

Dada la instancia válida r del esquema relacional R , contestar Verdadero (V) o Falso (F) en cada una de las partes justificando sus respuestas.

Notación: en los valores de la instancia, subíndices distintos significan valores distintos

R (nombre, ci, email, tel, dir)

r :

nombre	ci	email	tel	dir
n_1	c_1	e_1	t_1	d_1
n_1	c_1	e_2	t_2	d_2
n_1	c_1	e_1	t_2	d_3
n_1	c_1	e_2	t_1	d_4

- La instancia: $\prod_{\text{nombre,ci,email,tel}}(r)$, satisface la dependencia : nombre, ci \twoheadrightarrow email | tel
- La instancia r satisface la dependencia: nombre, ci \twoheadrightarrow dir
- La instancia r satisface la dependencia: tel, dir \rightarrow ci, nombre
- dir no es una clave en el esquema R
- ci es una clave en el esquema R

Solución:

a) Verdadero

Esa instancia satisface la definición de la multivaluada, ya que para todas 2 tuplas t_1, t_2 , tales que:

- $t_1[\text{nombre,ci}] = t_2[\text{nombre,ci}]$

existen otras 2 tuplas t_3, t_4 , que cumplen:

- $t_3[\text{nombre,ci}] = t_4[\text{nombre,ci}] = t_1[\text{nombre,ci}] = t_2[\text{nombre,ci}]$
- $t_1[\text{email}] = t_3[\text{email}], t_2[\text{tel}] = t_3[\text{tel}], t_2[\text{email}] = t_4[\text{email}], t_1[\text{tel}] = t_4[\text{tel}]$

b) Falso

Ya que en r existen las tuplas t_1, t_2 , tales que:

- $t_1[\text{nombre, ci}] = t_2[\text{nombre,ci}]$

ya no existen otras 2 tuplas t_3 y t_4 , tales que:

- $t_1[\text{nombre, ci}] = t_2[\text{nombre,ci}] = t_3[\text{nombre,ci}] = t_4[\text{nombre,ci}]$
- $t_1[\text{dir}] = t_3[\text{dir}],$
 $t_2[\text{email,tel}] = t_3[\text{email,tel}]$
 $t_2[\text{dir}] = t_4[\text{dir}]$
 $t_1[\text{email,tel}] = t_4[\text{email,tel}]$

Otra forma de justificar :

Ya que en r existen las tuplas t_1, t_2 , tales que:

- $t_1[\text{nombre, ci}] = t_2[\text{nombre,ci}]$

pero no se puede cumplir la definición porque:

- $t_1[\text{dir}] \neq t_2[\text{dir}] \neq t_3[\text{dir}] \neq t_4[\text{dir}]$

c) Verdadero

Ya que para todas las tuplas t_i y t_j de r , $t_i[\text{tel, dir}] \neq t_j[\text{tel, dir}]$

- d) Falso
Ya que dir es clave en la instancia r, no puedo decir que no lo sea en el esquema R con los datos que tengo.
- e) Falso
ci no es clave en la instancia r ya que existen dos tuplas t1 y t2 tales que:
- t1[ci] = t2[ci] y sin embargo
 - t1[email, tel, dir, nombre] <> t2[email, tel, dir, nombre]
- y si no es clave en la instancia no puede ser clave en el esquema R.

Ejercicio 2 (15 pts)

En una agencia de publicidad trabajan creativos que diseñan piezas publicitarias para diversos clientes. Cada pieza, luego es realizada por una compañía productora de audiovisuales. La agencia desea guardar información acerca de los clientes, las productoras y las piezas de publicidad realizadas.

De los clientes, se mantiene un identificador del cliente, su nombre, dirección y teléfono. De las productoras se mantiene el nombre de la productora y nombre del dueño, dirección, teléfono y tipo de producción que realiza. Pueden existir más de una productora con el mismo nombre, pero no más de una con el mismo nombre y el mismo dueño. Una pieza de publicidad tiene un nombre que la identifica y se conoce, el creativo que la ideó, el cliente para el que se hizo, la productora que la realizó, qué premios ganó, y la fecha de realización. Cada cliente tiene un único creativo asignado, que es el que siempre trabaja para él (un creativo puede estar asignado a varios clientes). Un creativo no puede estar en la realización de piezas de dos clientes distintos el mismo día.

- a) Decir que dependencias funcionales se cumplen entre sus datos.
- b) Suponiendo que construimos un esquema relación compuesto por todos los datos del cliente más los datos que identifican a la productora, decir en que forma normal se encuentra ese esquema, justificando.

Nota: Utilizar los siguientes nombres de atributos: id_cli, nom_cli, dir_cli, tel_cli, nom_prod, nom_dueño, dir_prod, tel_prod, tipo, nom_pieza, creativo, premio, fecha_realizacion.

Solución:

a)

id_cli → nom_cli, dir_cli, tel_cli
 nom_prod, nom_dueño → dir_prod, tel_prod, tipo
 nom_pieza → creativo, id_cli, nom_prod, nom_dueño, fecha_realizacion
 id_cli → creativo
 creativo, fecha_realizacion → id_cli

b)

R (id_cli, nom_cli, dir_cli, tel_cli, nom_prod, nom_dueño)

F = { id_cli → nom_cli, dir_cli, tel_cli }

Clave: id_cli, nom_prod, nom_dueño

Forma normal: 1NF, porque la df viola 2NF, por ser una dependencia parcial de una clave.

Ejercicio 4 (10 pts)

Considere las siguientes transacciones T_1 y T_2 :

T_1 : $r_1(X)$ $w_1(X)$ $r_1(Y)$ $w_1(Z)$ c_1

T_2 : $r_2(Y)$ $w_2(Y)$ $r_2(Z)$ $w_2(Z)$ c_2

- Dar una historia de T_1 y T_2 , serializable pero no recuperable.
- Dar una historia de T_1 y T_2 , que evite abortos en cascada.
- Dar una historia de T_1 y T_2 , mostrando bloqueos (de lectura y escritura) y desbloqueos, donde las transacciones sigan 2PL básico.
- ¿Qué cambiaría en los bloqueos y desbloqueos de T_1 y T_2 para asegurar que todas las historias formadas por estas transacciones sean estrictas?

Solución:

a) $r_1(X)$ $w_1(X)$ $r_2(Y)$ $w_2(Y)$ $r_1(Y)$ $r_2(Z)$ $w_2(Z)$ $w_1(Z)$ c_1 c_2

b) $r_1(X)$ $w_1(X)$ $r_2(Y)$ $w_2(Y)$ $r_2(Z)$ $w_2(Z)$ c_2 $r_1(Y)$ $w_1(Z)$ c_1

c) $rl_1(X)$ $r_1(X)$ $wl_1(X)$ $w_1(X)$ $rl_2(Y)$ $r_2(Y)$ $wl_2(Y)$ $w_2(Y)$ $wl_2(Z)$ $u_2(Y)$ $rl_1(Y)$ $r_1(Y)$ $r_2(Z)$ $w_2(Z)$ $u_2(Z)$
 $wl_1(Z)$ $w_1(Z)$ $u_1(X)$ $u_1(Y)$ $u_1(Z)$ c_1 c_2

d) En la parte c, eran:

T_1 : $rl_1(X)$ $r_1(X)$ $wl_1(X)$ $w_1(X)$ $rl_1(Y)$ $r_1(Y)$ $wl_1(Z)$ $w_1(Z)$ $u_1(X)$ $u_1(Y)$ $u_1(Z)$

T_2 : $rl_2(Y)$ $r_2(Y)$ $wl_2(Y)$ $w_2(Y)$ $wl_2(Z)$ $u_2(Y)$ $r_2(Z)$ $w_2(Z)$ $u_2(Z)$

Para asegurar historias estrictas deben seguir 2PL estricto:

T_1 : $rl_1(X)$ $r_1(X)$ $wl_1(X)$ $w_1(X)$ $rl_1(Y)$ $r_1(Y)$ $wl_1(Z)$ $w_1(Z)$ c_1 $u_1(X)$ $u_1(Y)$ $u_1(Z)$

T_2 : $rl_2(Y)$ $r_2(Y)$ $wl_2(Y)$ $w_2(Y)$ $wl_2(Z)$ $r_2(Z)$ $w_2(Z)$ c_2 $u_2(Y)$ $u_2(Z)$

Ejercicio 5 (10 pts)

Las siguientes tablas son utilizadas por un sistema que provee una plataforma para el Gobierno Electrónico:

- **Organismo (id_org, nombre)** Contiene todos los organismos que usan el sistema, ya sea usando o proveyendo servicios.
- **Servicio (id_s, org_p)** Contiene una identificación (id_s) para cada servicio provisto y cuál es el organismo que lo provee (org_p).
- **Uso (id_s, org_u, fh_ini, fh_fin)** contiene una tupla para cada uso del servicio indicado por id_s que hizo el organismo org_u, registrando la fecha y hora de inicio y fin de ese uso.

El manejador sobre el que está implementado sigue las estrategias de optimización presentadas en el curso, sólo que nunca es capaz de cambiar el orden de las tablas que vienen desde la consulta SQL. Además el manejador es capaz de explotar prefijos sobre los índices es decir, que si el índice está creado sobre 7 atributos, entonces el manejador es capaz de usar el índice por el primer atributo, o por el primero y el segundo, etc. Sin embargo no es capaz de utilizar el índice como si fuera sólo por un atributo intermedio. Para explotar los índices de esta forma, el manejador puede construir árboles con condiciones conjuntivas en joins o selecciones.

El manejador además, crea un índice primario para cada clave primaria con los atributos en el orden en que están declarados en la tabla.

Considere que normalmente, la tabla Uso será mucho más grande que el producto cartesiano de Organismo y Servicios.

Considerando las siguientes consultas:

Consulta 1)

```
Select O.nombre
from Uso U, Organismo O, Servicio S
where S.id_s='5454' and S.id_s=U.id_s and
      U.org_u=O.id_org;
```

Consulta 2)

```
Select O.nombre
from Organismo O, Uso U, Servicio S
where S.id_s='5454' and S.id_s=U.id_s and
      U.org_u=O.id_org;
```

Consulta 3)

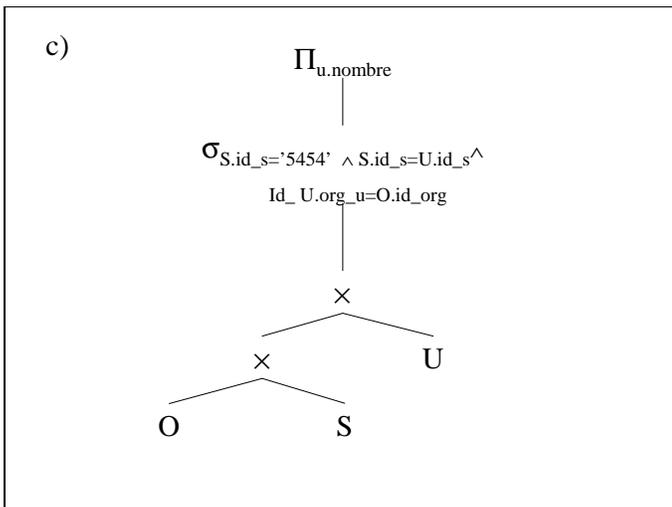
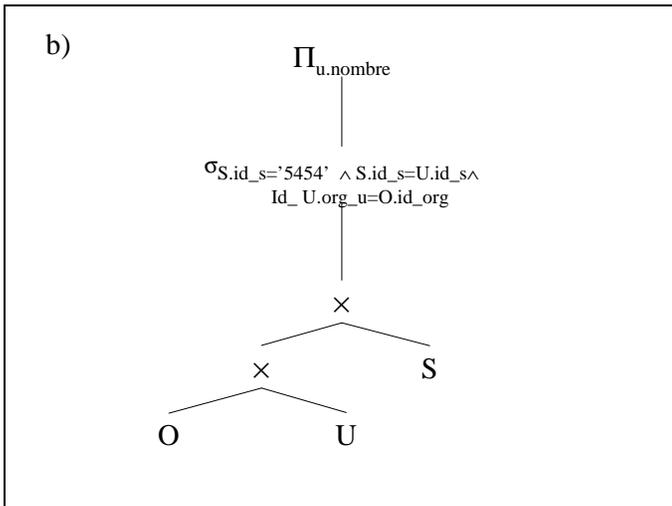
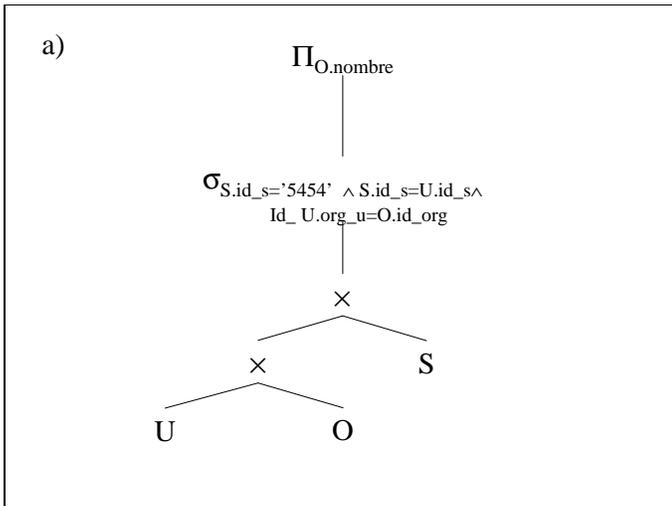
```
Select O.nombre
from Organismo O, Servicio S, Uso U
where S.id_s='5454' and S.id_s=U.id_s and
      U.org_u=O.id_org;
```

Se pide:

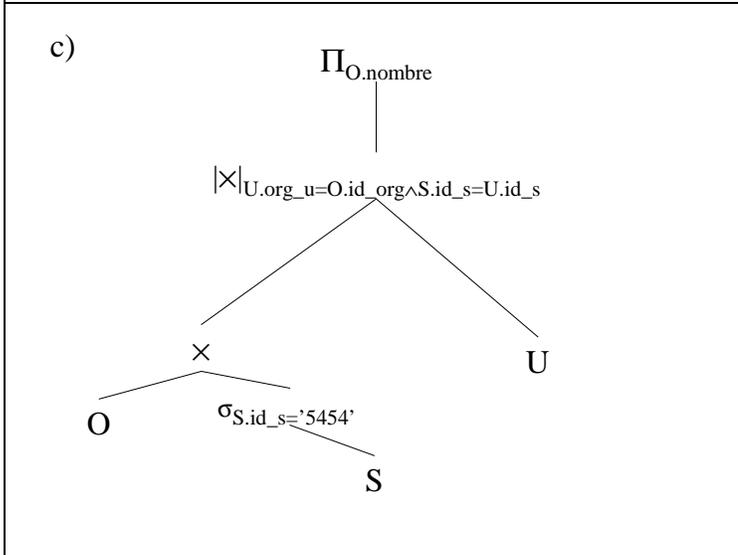
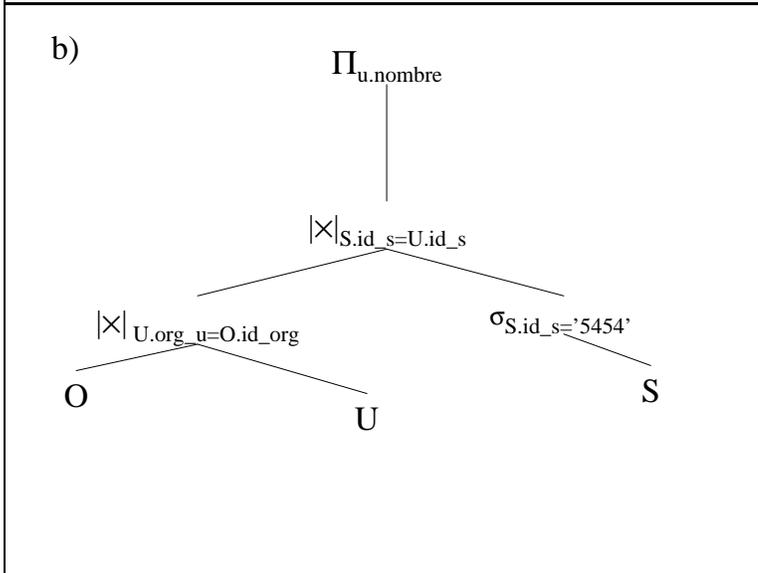
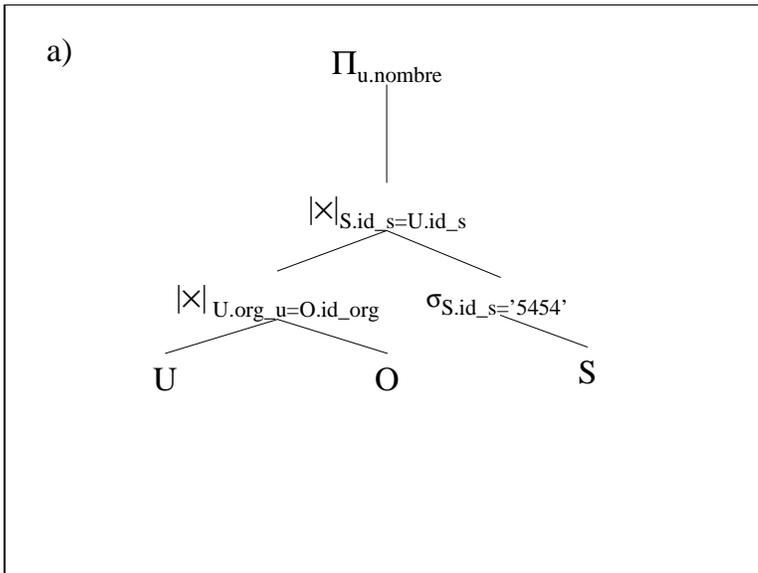
- a) Dar los árboles canónicos de las consultas.
- b) Dar los árboles optimizados según las heurísticas utilizadas por este manejador.
- c) Indicar cuál de las siguientes consultas debería ser más eficiente o si todas serán igual de eficientes. Justifique su respuesta.

Solución:

Los árboles canónicos en los tres casos son similares sólo que cambia el orden de las tablas:



Sin embargo al hacer la distribución de selecciones y sustitución por joins los resultados son claramente diferentes:



Dado que la tabla Uso es mucho más grande que las otras, es razonable pensar que explotando el índice de uso se debe obtener mejor performance. Esto sólo lo permite C porque permite precisamente la explotación por prefijo, ya que en ese punto se conoce el valor de los dos primeros atributos del índice de la tabla.