

BASES DE DATOS

SOLUCION EXAMEN – Febrero 2000

Parte A: DISEÑO RELACIONAL (25 ptos)

Ejercicio 1 (14 ptos)

Dada la siguiente tabla R(ABCDEG) y el siguiente conjunto de dependencias funcionales:

$\{BC \rightarrow A, A \rightarrow EG, A \rightarrow D, EG \rightarrow D, D \rightarrow A\}$

Se pide:

- a) Calcule todas las claves. Fundamente su respuesta.

Solución:

Los atributos BC no son determinados por ningún otro atributo, por lo que deben participar en todas las claves.

$(BC)^+ = \{B, C, A, E, G, D\}$ por lo que BC es clave. Como estos atributos deben participar en todas las claves, entonces es la única clave.

- b) Encuentre un cubrimiento minimal para el conjunto de dependencias.

Solución:

El conjunto de dependencias original es

$\{BC \rightarrow A, A \rightarrow EG, A \rightarrow D, EG \rightarrow D, D \rightarrow A\}$

i)

La única dependencia que tiene varios atributos en el lado derecho es la segunda, por lo que al dividirla, el conjunto que obtenemos es:

$F = \{BC \rightarrow A, A \rightarrow E, A \rightarrow G, A \rightarrow D, EG \rightarrow D, D \rightarrow A\}$

ii)

Luego, eliminamos los atributos redundantes a la izquierda. Las únicas dependencias interesantes son la primera y la penúltima ya que son las únicas que tienen varios atributos a la izquierda. En la primera, ninguno de los dos atributos es determinado por ningún otro, de manera que ninguno de los dos es redundante. Para estudiar la otra calculamos las clausuras de los atributos:

$E^+ = \{E\}$

$G^+ = \{G\}$

Como ninguno de los atributos pertenece a la clausura del otro, podemos asegurar que ninguno de los dos atributos es redundante con respecto al otro.

iii)

Para la eliminación de las dependencias redundantes, consideramos la clausura de los atributos de la izquierda de la dependencia con respecto al conjunto de dependencias sin la dependencia que estudiamos. Si obtenemos los atributos de la derecha, entonces la dependencia es redundante.

$(BC)^+_{F-\{BC \rightarrow A\}} = \{B, C\} \Rightarrow$ La dependencia $BC \rightarrow A$ no es redundante.

$(A)^+_{F-\{A \rightarrow E\}} = \{A, G, D\} \Rightarrow$ La dependencia $A \rightarrow E$ no es redundante.

$(A)^+_{F-\{A \rightarrow G\}} = \{A, E, D\} \Rightarrow$ La dependencia $A \rightarrow G$ no es redundante.

$(A)^+_{F-\{A \rightarrow D\}} = \{A, E, G, D\} \Rightarrow$ La dependencia $A \rightarrow D$ **SI** es redundante. Se puede eliminar.

De aquí en adelante seguimos trabajando con

$$F' = \{BC \rightarrow A, A \rightarrow E, A \rightarrow G, EG \rightarrow D, D \rightarrow A\}$$

$(EG)^+_{F'-\{EG \rightarrow D\}} = \{E, G\} \Rightarrow$ La dependencia $EG \rightarrow D$ no es redundante.

$(D)^+_{F'-\{D \rightarrow A\}} = \{D\} \Rightarrow$ La dependencia $D \rightarrow A$ no es redundante.

Estos tres pasos nos aseguran que F' es un cubrimiento minimal de F . De aquí en adelante, seguimos trabajando con F' .

- c) Aplique el algoritmo para calcular una descomposición en BCNF, comenzando por la dependencia $D \rightarrow A$. En cada paso, estudie la pérdida de dependencias exponiendo las proyecciones correspondientes del conjunto de dependencias.

Solución:

Considerando F' como conjunto de dependencias y el esquema original $R(ABCDEG)$, la única dependencia que no viola BCNF es la primera, ninguna de las demás es dependencia de clave.

Paso 1.

Comenzando con la dependencia $D \rightarrow A$, el algoritmo nos produce la siguiente partición:

$$R_1(BCDEG) \quad R_2(DA)$$

Donde

$$\Pi_{R_1}(F') = \{EG \rightarrow D, BC \rightarrow E, BC \rightarrow G, D \rightarrow EG\} = F''$$

$$\Pi_{R_2}(F') = \{D \rightarrow A, A \rightarrow D\}$$

Hay que recordar que cuando se realizan las proyecciones de un conjunto de dependencias F sobre un esquema, se incluyen las dependencias que tiene todos los atributos en el esquema, pero dependencias de F^+ y no de F solamente. Por eso, es que si bien la dependencia $BC \rightarrow A$ parece perderse, se deben incluir todas las transitivas posibles que se forman con esa dependencia y cualquier otra del conjunto. De esa forma se incluyen $BC \rightarrow E$ y $BC \rightarrow G$. En este caso particular, ocurre que cuando calculamos la clausura de BC con respecto a la unión de las proyecciones, nos encontramos con el siguiente conjunto:

$$\{E, G, D, A\}$$

por lo que la dependencia $BC \rightarrow A$ NO SE PERDIÓ.

Paso 2.

En R_1 la única clave sigue siendo BC por un razonamiento similar al utilizado en la parte a.

Por esta razón, ese esquema no está en BCNF. La única dependencia que viola BCNF es: $EG \rightarrow D$. Los nuevos esquemas que forman son:

$$R_{11}(BCEG) \quad R_{12}(EGD)$$

Si calculamos las proyecciones de las dependencias de F'' sobre estos esquemas obtenemos

$$\Pi_{R_{11}}(F'') = \{BC \rightarrow E, BC \rightarrow G\}$$

$$\Pi_{R_{12}}(F'') = \{EG \rightarrow D\}$$

Con lo cual es claro que no se perdió ninguna dependencia.

La descomposición final en BCNF es:

$R_{11}(BCEG)$ $R_{12}(EGD)$ $R_2(DA)$ y sin pérdida de dependencias. El algoritmo, asegura además el join sin pérdida.

Ejercicio 2

(12 ptos)

Una inmobiliaria desea construir una base de datos con las casas que tiene para alquilar, los clientes y los propietarios. De cada casa se conoce un código que la identifica, su dirección, la cédula de su propietario, la cantidad de dormitorios, la cantidad de baños y el alquiler mensual. De los clientes se conoce la cédula de identidad, su nombre, su última dirección conocida, su último teléfono conocido (uno solo) y la cédula del propietario que le sale de garantía (uno solo). De los propietarios se conoce su cédula, su dirección y su teléfono (único). Para cada casa alquilada se establece un contrato que tiene un número que lo identifica y registra la cédula del cliente, el código de la propiedad y la fecha de inicio y de fin del contrato. En una fecha de inicio de un contrato, no se puede alquilar una misma casa a más de un cliente.

Se pide:

- a) Determinar las dependencias funcionales y multivaluadas que se cumplen.

Solución:

De cada casa se conoce un código que la identifica (CA), su dirección (CD), la cédula de su propietario (PC), la cantidad de dormitorios (CND), la cantidad de baños (CNB) y el alquiler mensual (CAL).

De este párrafo se entiende que el código de la casa determina a todos los demás atributos de la casa:

CA → CD, PC, CND, CNB, CAL.

De los clientes se conoce la cédula de identidad (CLI), su nombre (CLN), su última dirección conocida (CLD), su último teléfono conocido (CLT) y la cédula del propietario (PC) que le sale de garantía.

La cédula de identidad determina todos los datos de la persona. Por ello,

CLI → CLN, CLD, CLT, PC

De los propietarios se conoce su cédula (PC), su dirección (PD) y su teléfono (PT).

Nuevamente, la cédula del propietario determina su dirección y su teléfono., por lo que:

PC → PD, PT

Para cada casa alquilada se establece un contrato que tiene un número que lo identifica (CC) y registra la cédula del cliente (CLI), el código de la casa (CA) y la fecha de inicio (CFI) y de fin (CFF) del contrato. En una fecha de inicio de un contrato, no se puede alquilar una misma casa a más de un cliente. Además se desea guardar esta información todo el tiempo posible.

Del párrafo anterior se deduce que CC determina el resto de los atributos, ya que identifica al contrato.

CC → CLI, CA, CFI, CFF.

Además, una misma casa pudo estar bajo contrato varias veces en diferentes fechas iniciales. También se indica que en la misma fecha nunca hay que haber dos contratos para la misma casa, por lo que:

CA, CFI → CC

No hay dependencias multivaluadas. El conjunto de dependencias que se cumplen es:

{ CA → CD, PC, CND, CNB, CAL ;
CLI → CLN, CLD, CLT, PC;
PC → PD, PT;
CC → CLI, CA, CFI, CFF;
CA, CFI → CC }

- b) Determinar una descomposición en 3nf con join sin pérdida y sin pérdida de dependencias.

Solución:

Paso 1. Calculamos un cubrimiento minimal para el conjunto de dependencias.

- a. Dejamos dependencias que sólo tienen un atributo en el lado derecho.

$F = \{$
CA \rightarrow CD;
CA \rightarrow PC;
CA \rightarrow CND;
CA \rightarrow CNB;
CA \rightarrow CAL;
CLI \rightarrow CLN;
CLI \rightarrow CLD;
CLI \rightarrow CLT;
CLI \rightarrow PC;
PC \rightarrow PD;
PC \rightarrow PT;
CC \rightarrow CLI;
CC \rightarrow CA;
CC \rightarrow CFI;
CC \rightarrow CFF;
CA, CFI \rightarrow CC $\}$

- b. Eliminamos atributos redundantes del la izquierdo.

La única que puede tener atributos redundantes es la última dependencia. Por eso sólo controlamos los atributos que están a la izquierda de esta dependencia.

$(CA)^+ = \{CA, CD, PC, CND, CNB, CAL, PD, PT\}$

$(CFI)^+ = \{CFI\}$

Por lo que no hay atributos redundantes.

- c. Eliminamos dependencias redundantes.

Para ello habría que calcular, para cada dependencia, la clausura de los atributos que están a la izquierda con respecto al conjunto de las dependencias restantes. Si en esa clausura aparecen los atributos de la derecha, entonces se puede eliminar la dependencia.

Los razonamientos que siguen a continuación es una forma abreviada de hacer lo mismo. Pero hay que hacerlos con mucho cuidado.

Consideremos primero todas las dependencias que tienen CA a la izquierda

El único de los atributos que aparece a la derecha de esas dependencias que aparece a la izquierda de alguna otra es PC. Por lo tanto es la única fuente posible de transitividades que tengan a CA a la izquierda. Ahora bien,

$(PC)^+ = \{PC, PD, PT\}$ y no están explícitamente en el conjunto de dependencias las dependencias CA \rightarrow PD y CA \rightarrow PT. Además, ni PT ni PD determinan ningún otro atributo, por lo que no hay más atributos que sean determinados por CA. Conclusión: No hay dependencias redundantes que comiencen con CA.

Consideremos ahora todas las dependencias que tienen CLI a la izquierda. Por un razonamiento análogo al anterior, vemos que el único atributo de estas dependencias que determinan otros atributos es PC. Siguiendo un razonamiento análogo al anterior podemos ver que ninguna de estas dependencias es redundante.

Siguiendo el mismo razonamiento con las dependencias que comienzan con PC, observamos que no hay dependencias redundantes entre ellas.

Lo mismo sucede con CC y con CA,CFI.

Conclusión: no hay dependencias redundantes. F es un cubrimiento minimal.

Paso 2. Construimos una tabla para cada dependencia, agrupando aquellas que tienen los mismos atributos del lado izquierdo.

R1(CA, CD, PC, CND, CNB, CAL)
R2(CLI, CLN, CLD, CLT, PC)
R3(PC, PD, PT)
R4(CC, CLI, CA, CFI, CFF)
R5(CA, CFI, CC)

Como tenemos que R5 está incluida en R4 la podemos eliminar. Con lo que obtenemos:

R1(CA, CD, PC, CND, CNB, CAL)
R2(CLI, CLN, CLD, CLT, PC)
R3(PC, PD, PT)
R4(CC, CLI, CA, CFI, CFF)

Para garantizar el join sin pérdida, debemos encontrar una clave que esté en alguna de las tablas..

Los siguientes atributos, nunca aparecen del lado izquierdo a una dependencia, por lo que sabemos que nunca van a pertenecer a una clave: { CD, CND, CNB, CAL, CLN, CLD, CLT, CFF }. Con esto, sólo queda por considerar el los atributos del siguiente conjunto: S={ CA, CLI, PC, CC, CFI }

Seguramente, este conjunto es una superclave.

Consideremos la clausura de CC:

$(CC)^+ = \{CC, CLI, CA, CFI, CFF, CD, PC, CND, CNB, CAL, CLN, CLD, CLT, PD, PT\}$ por lo que CC es una clave y está incluida en R4. Esto garantiza que el join es sin pérdida.

Parte B: CONSULTAS (25 pts)

Ejercicio 3

Una importante empresa de venta de pinturas con varias sucursales decide formar una sección que se encargue de realizar las combinaciones de colores requeridas por los clientes. A su vez mantienen una base de datos con los pedidos para poder obtener estadísticas acerca de las mezclas mas frecuentes. Dicha base de datos posee las siguientes tablas:

- Colores (idC, descripcion, tipo, brillo). Representa la información de cada uno de los colores. IdC es el identificador del color. Descripción es el nombre asignado al color. Tipo indica si el color es puro (provisto por los fabricantes, sin necesidad de realizar mezclas) o mezcla (ese color no se vende, hay que realizar la mezcla). Brillo indica si el color es brillante o mate. Descripción identifica también al color.
- Mezclas (idC, idCPuro, proporcion). Representa la información de los colores que se obtienen como combinación de colores puros. IdC es el identificador del color de la mezcla. IdCPuro y proporción indican la proporción del color puro para realizar la mezcla.
- Productos (IdP, descripcion, marca, tamaño). Representa la información de los productos (latas de pintura). IdP es un identificador. Descripción es el nombre del producto. Marca es la marca del producto. Tamaño es la cantidad de litros de la lata. Se consideran productos distintos los que tienen diferentes tamaños, pero no los que tienen diferentes colores, es decir, que IdP no indica el color que tiene la pintura.
- ColorProd (idP, idC). Representa la información de los colores puros en que viene cada producto. IdP es el identificador del producto, e idC es el identificador del color. Sólo se venden latas de colores puros.
- Solicitudes (idSol, fecha, sucursal, idC, tamaño). Representa las solicitudes de mezclas realizadas por las diferentes sucursales. Sólo se solicitan mezclas, no colores puros. IdSol es un identificador. Fecha es la fecha en que se realizó la solicitud, y sucursal la sucursal de la empresa desde la que se hizo la misma. IdC es el identificador de color y tamaño es la cantidad de litros de la mezcla pedida.

Se pide:

Resolver cada consulta en el lenguaje indicado (álgebra relacional, cálculo relacional, sql).

- a) **En cálculo:** Dar las marcas, que entre sus productos, tienen todos los colores necesarios para realizar todas las mezclas.

Solución:

$$\{ t.marca / productos(t) \wedge$$
$$(\forall m) (mezclas(m) \rightarrow$$
$$(\exists p) (productos(p) \wedge (\exists cp) (colorprod(cp) \wedge p.idP = cp.idP \wedge$$
$$p.marca = t.marca \wedge cp.idC = m.idCBase$$
$$))$$
$$)$$
$$\}$$

- b) **En álgebra:** Dar el color puro utilizado en más de 3 mezclas.

Solución:

$$M2(\text{color puro}, \text{mezcla1}, \text{mezcla2}) = \Pi_{\%2, \%1, \%4}(\text{Mezclas} \ast_{\%2 = \%5 \wedge \%1 < \%4} \text{Mezclas})$$
$$M3(\text{color puro}, \text{mezcla1}, \text{mezcla2}, \text{mezcla3}) = \Pi_{\%1, \%2, \%3, \%4}(\text{M2} \ast_{\%2 < \%4 \wedge \%3 < \%4 \wedge \%1 = \%5} \text{Mezclas})$$
$$\text{Sol} = \Pi_{\%1}(\text{M3} \ast_{\%2 < \%5 \wedge \%3 < \%5 \wedge \%4 < \%5 \wedge \%1 = \%6} \text{Mezclas})$$

- c) **En SQL:** Dar la lista de mezclas mates solicitados más de 5 veces, y la cantidad total de litros solicitados de cada una.

Solución:

```
SELECT C.idC, sum (S.tamanio)
FROM solicitudes S, colores C
WHERE S.idC = C.idC
AND C.brillo = "mate"
GROUP BY C.idC
HAVING count(*) > 5
```

- d) **En álgebra:** Dar la lista de colores que pueden prepararse mezclando exclusivamente algunos o todos los colores del producto P13.

Solución:

$$A = \Pi_{\%2}(\sigma_{\%1 = "P13"}(\text{colorprod}))$$
$$B = \Pi_{\%1}(\text{mezclas} \mid \> \mid_{\%2 = \%4} A)$$
$$C = \Pi_{\%1}(\text{colores}) - A$$
$$D = \Pi_{\%1}(\text{mezclas} \mid \> \mid_{\%2 = \%4} C)$$
$$\text{Sol} = B - D$$

Parte C: M.E.R. (25 pts)

Ejercicio 4

Una empresa de entretenimientos y vacaciones para niños en edad escolar y preescolar desea automatizar el manejo de la información de sus clientes y las asociaciones con las que trabaja. La información que se desea mantener tiene las siguientes características:

Existen varias asociaciones juveniles, las cuales tienen sus propias colonias de vacaciones. Cada asociación tiene varias colonias, pero cada colonia pertenece a una única asociación. De cada asociación se conoce su nombre, que la identifica, la dirección y un teléfono de referencia. De las colonias se conoce su código y ubicación; el código puede repetirse para las distintas asociaciones.

En las colonias trabajan varios líderes de grupos, de los cuales se conoce su C.I., nombre y teléfono. Cada líder puede trabajar para varias colonias. Todos los líderes deben tener una certificación que los acredita como tales, interesa la fecha, el grado y la asociación que emitió el certificado. En caso de tener más de un certificado interesa sólo el más reciente.

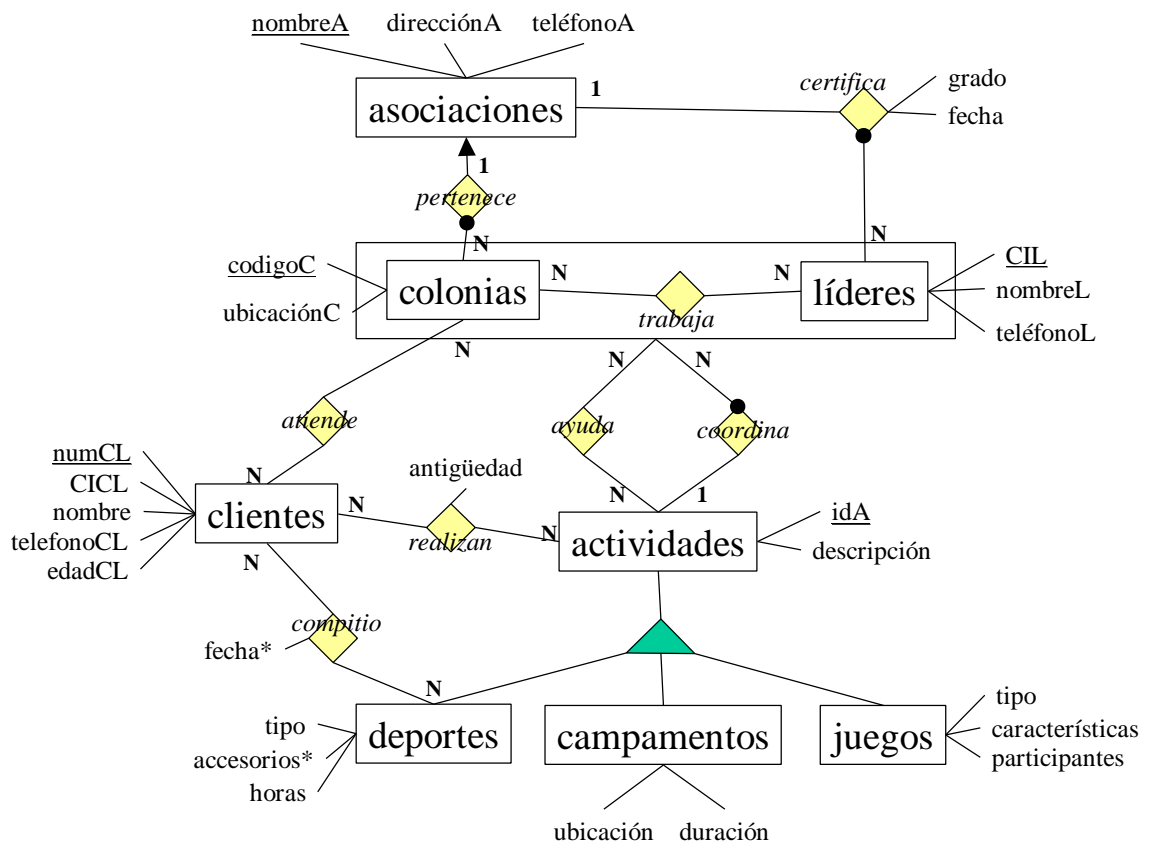
Cada líder en una colonia coordina exactamente una actividad, pero puede ayudar en otras. Las actividades a su vez son desarrolladas (coordinación y ayuda) por varios líderes de colonias. De las actividades se conoce su identificador y una breve descripción de la misma. Estas pueden ser de los siguientes tipos: campamentos, deportes y juegos. De los campamentos interesa la ubicación y la duración en días, de los deportes interesa el tipo, los accesorios necesarios y la cantidad de horas semanales de entrenamiento, de los juegos interesa el tipo de juego, una descripción de las características y la cantidad de participantes.

Cada colonia atiende a un conjunto de clientes, algunos de ellos asisten a más de una colonia. Nos interesa el número de cliente que lo identifica, nombre, C.I., teléfono y edad. Los clientes realizan diversas actividades, interesando la antigüedad con que las realizan. En el caso de los deportes, interesan también las fechas en las que el cliente participó en competencias. Los clientes sólo realizan actividades de las disponibles en su colonia.

Se pide:

Modelo Entidad-Relación completo.

Solución:



R.N.E.: Si un cliente *realiza* una actividad, ésta debe ser *coordinada* por alguno de los líderes de alguna de las colonias que *atienden* al cliente.

R.N.E.: Si un cliente *compitio* en un deporte, debe *realizar* ese deporte (actividad).

Parte D: CONCURRENCIA Y RECUPERACION (25 ptos)

Ejercicio 5 (6 ptos)

Sean las transacciones:

T1: r1(x), w1(y), c1
T2: r2(y), w2(x), c2

Se pide:

Para las siguientes historias, decir si son: Serializables, Recuperables, evitan Abortos en Cascada, son Estrictas.

a) r1(x), r2(y), w2(x), w1(y), c1, c2

Solución:

- Ninguna transacción lee de otra, o graba sobre valores grabados de otra. Por lo que H1 es estricta..
- En el SG(H1) hay arcos de: T1→T2, T2→T1. Hay un ciclo, no es Serializable.

b) r1(x), w1(y), r2(y), w2(x), c2, c1

Solución:

- T2 lee de T1, pero c2 precede a c1. Por lo tanto H2 no es recuperable.
- En el SG(H2) hay arcos de: T1→T2. No hay ciclo, es Serializable.

c) r1(x), r2(y), w1(y), w2(x), c2, c1

Solución:

- Ninguna transacción lee de otra, o graba sobre valores grabados de otra. Por lo que H es estricta..
- En el SG(H3) hay arcos de: T1→T2, T2→T1. Hay un ciclo, no es Serializable.

Ejercicio 6 (12 ptos)

Sean las transacciones:

T1: w1(x), r1(y), c1
T2: r2(x), w2(z), c2

Se pide:

a) Dar una historia de T1 y T2 que sea serializable, la operación r2(x) se ejecute después de w1(x), y evite abortos en cascada.

Solución:

H: w1(x), r1(y), c1, r2(x), w2(z), c2

Para que evita abortos en cascada r2(x) debe hacerse luego de c1, por lo que la historia queda serial.

b) Dar una historia de T1 y T2 que sea serializable y NO recuperable, y en la cual T1 y T2 sigan 2PL básico.

Solución:

H: lw1(x), lr1(y), w1(x), u1(x), lr2(x), r2(x), u2(x), lw2(z), w2(z), u2(z), c2, r1(y), u1(y), c1

H no es recuperable porque T2 lee de T1 y comitea antes.

Es serializable ya que hay un solo arco en el SG(H), el arco de T1 a T2.

Sigue 2PL básico.

- c) Dar una historia de T1 y T2 que sea serializable, recuperable, pero que no evite abortos en cascada, y en la cual T1 y T2 sigan 2PL básico.

Solución

H: lw1(x), lr1(y), **w1(x)**, u1(x), lr2(x), **r2(x)**, u2(x), lw2(z), **w2(z)**, u2(z), **r1(y)**, u1(y), **c1, c2**

H es recuperable porque T2 lee de T1 y comitea despues, pero no evita abortos en cascada porque el r2(x) no se realiza sobre valores comiteados..

Es serializable ya que hay un solo arco en el SG(H), el arco de T1 a T2.

Sigue 2PL básico.

Ejercicio 7 (6 ptos)

Se pide:

En los diferentes casos de fallas, decir en qué pasos consistiría el proceso de recuperación.

- a) Se trabaja en modo de Actualización Inmediata, y ocurre una caída del sistema junto con falla física.

Solución:

- Instalar backup de BD.
- Aplicar proceso de UNDO.
- Aplicar proceso de REDO.

- b) Se trabaja en modo de Actualización Diferida y ocurre una caída del sistema.

Solución:

- Aplicar proceso de REDO.