

## FUNDAMENTOS DE BASES DE DATOS

### Examen Diciembre 2013

La duración del examen es de 3 horas.

Presentar la resolución del examen:

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

### Ejercicio 1 (25 puntos).

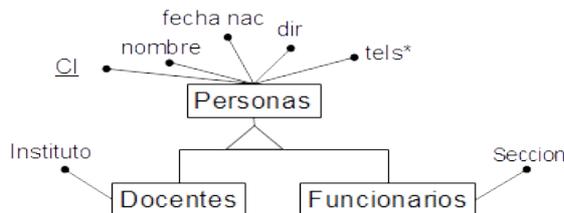
En este ejercicio el enunciado de la letra se encuentra presentado en forma incremental en diferentes partes.

**Para cada parte se pide:** Construir un modelo entidad relación incluyendo restricciones no estructurales que modelen la realidad planteada en cada parte.

Para esto, debe copiar lo que considere necesario para la comprensión del diagrama de cada parte o incluso, copiar con cambios algunas partes de la etapa anterior. Por ejemplo, si pone una relación, debe incluir las entidades participantes, aunque puede omitir los atributos de la(s) entidades que están resueltas en partes anteriores si no corresponde realizarle ningún cambio.

a) La sección Personal de facultad debe administrar los legajos de los docentes y funcionarios. De cada persona se conoce su cédula, su nombre completo, su fecha de nacimiento, dirección y un conjunto de teléfonos. Además se sabe si es docente o funcionario. Si es docente, se conoce el instituto en el que trabaja. Si es funcionario, se conoce la sección en donde trabaja.

**SOLUCIÓN:**

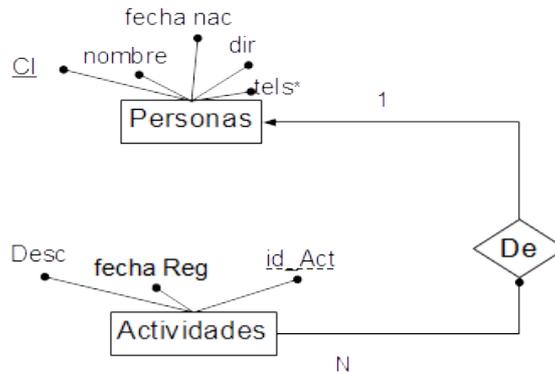


RNE's:

$$Docentes \cup Funcionarios = Personas$$

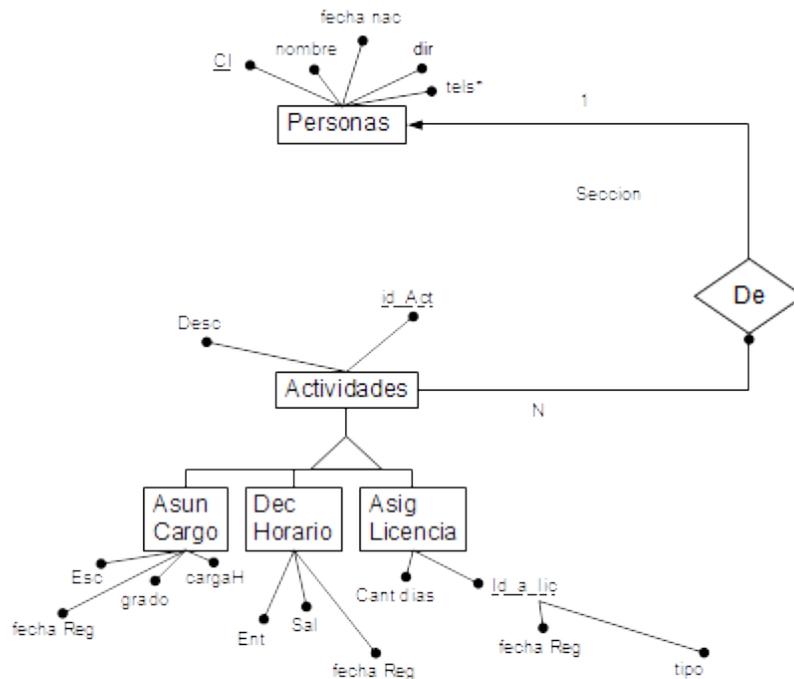
b) De cada persona además, se debe llevar el legajo. Esto es el conjunto de actividades que realiza el docente o funcionario en la facultad y le interesan a la sección personal. Cada una de estas actividades, tiene una identificación correlativa dentro de las actividades de la persona, una fecha de registro y una descripción.

**SOLUCIÓN:**



c) Las actividades pueden ser de diferentes tipos.  
 De las actividades de tipo Asunción de Cargo se conoce el escalafón, grado y carga horaria  
 De las actividades de Declaración de Horario, se conoce la hora de inicio y fin. Se asume que se hará el mismo horario todos los días de lunes a viernes y que todos los horarios son corridos.  
 De las actividades de Asignación de Licencia, se conoce una cantidad de días y un tipo. Notar que una persona puede tener varias licencias asignadas en la misma fecha de registro pero deben ser con distinto tipo. Usualmente se usan ciertos tipos de licencia (Reglamentaria, por elecciones, por antigüedad) pero pueden aparecer nuevos tipos en cualquier momento.

**SOLUCIÓN:**

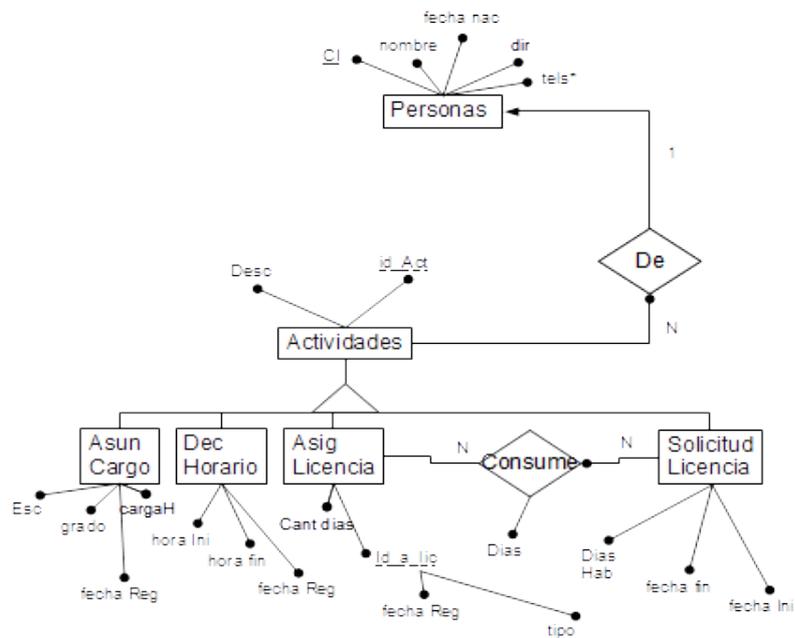


RNEs:

- $Asun\ Cargo \cap Dec\ horario = \emptyset$
- $Dec\ Horario \cap Asig\ Licencia = \emptyset$
- $Asig\ Licencia \cap Asun\ Cargo = \emptyset$
- $Asun\ Cargo \cap Solicitud\ Licencia = \emptyset$
- $Dec\ Horario \cap Solicitud\ Licencia = \emptyset$
- $Asig\ Licencia \cap Solicitud\ Licencia = \emptyset$

d) Otro tipo de actividad posible, es la Solicitud de Licencia. Para esta clase de actividades, se conoce una fecha de inicio, una fecha de fin y la cantidad de días hábiles que hay entre las dos fechas. Además se indica cuantos días se toman de determinada asignación de licencia. Por ejemplo un docente puede pedir licencia desde el 7/1/2014 al 17/1/2014 lo que totaliza 9 días hábiles, de los cuales puede consumir 4 días de la licencia registrada con fecha 4/12/2012 de tipo antigüedad y 5 de días de la licencia registrada el 4/12/2013 de tipo reglamentaria. No puede tomar más de una vez los mismos días. Esto significa que el total de los días que son consumidos de una asignación de un tipo no debiera mayor que los días asignados de ese tipo.

**SOLUCIÓN:**



RNEs:

- Una solicitud de licencia de una persona no puede consumir licencia asignada a otra persona distinta.  
 $\forall sl \in Solicitud\ Licencia.$   
 $\forall al \in Asig\ Licencia.$
- $((sl, al) \in Consume \rightarrow \exists p \in Personas. ((sl, p) \in De \wedge (al, p) \in De))$
- La suma de todas las cantidades de días consumidos de una asignación de licencia no puede ser mayor que la cantidad de días asignados.  
 $\forall al \in Asig\ Licencia.$
- $\sum_{sl \in \{x | \langle al, x \rangle \in Consume\}} Dias(\langle al, sl \rangle) \leq Cantdias(al)$

e) Por otro lado, el cargo de una persona puede transformarse en otro que puede tener diferente grado o carga horaria, pero siempre dentro del mismo escalafón. Es necesario saber qué cargos se transformaron en qué otros.



De cada persona se conoce su cédula de identidad que lo identifica, su nombre y el año de ingreso al sistema.

**COCHES**(matricula, nro, tipo)

En esta tabla se mantiene la información sobre los diferentes coches que realizan los viajes correspondientes a los distintos recorridos/sentidos.

Para cada uno de ellos se conoce su matrícula que lo identifica, el número de coche y el tipo (Ej. Micro).

**VIAJES**(#recorrido,sentido,horaSalida, matricula)

En esta tabla se mantiene la información relativa a los coches utilizados en cada una de las salidas de los recorridos/sentido.

**VIAJES\_PERSONAL**(#recorrido,sentido,horaSalida, ci, función)

En esta tabla se mantiene la información relativa al personal involucrado en cada una de las salidas de los recorridos/sentido indicando la función que cumple (Ej. Chofer).

En esta base de datos se cumplen las siguientes restricciones de inclusión:

$$\Pi_{\#parada}(\text{REC-PARADAS}) \subseteq \Pi_{\#parada}(\text{PARADAS})$$

$$\Pi_{\#recorrido,\text{sentido}}(\text{PARADAS}) \subseteq \Pi_{\#recorrido,\text{sentido}}(\text{RECORRIDO})$$

$$\Pi_{\#recorrido,\text{sentido}}(\text{VIAJES}) \subseteq \Pi_{\#recorrido,\text{sentido}}(\text{RECORRIDO})$$

$$\Pi_{\#recorrido,\text{sentido}}(\text{VIAJES_PERSONAL}) \subseteq \Pi_{\#recorrido,\text{sentido}}(\text{RECORRIDO})$$

$$\Pi_{\text{matricula}}(\text{VIAJES}) \subseteq \Pi_{\text{matricula}}(\text{COCHES})$$

$$\Pi_{\text{ci}}(\text{VIAJES_PERSONAL}) \subseteq \Pi_{\text{ci}}(\text{PERSONAL})$$

$$\Pi_{\#recorrido,\text{sentido},\text{horaSalida}}(\text{VIAJES}) \subseteq \Pi_{\#recorrido,\text{sentido},\text{horaSalida}}(\text{VIAJES_PERSONAL})$$

$$\Pi_{\#recorrido,\text{sentido},\text{horaSalida}}(\text{VIAJES_PERSONAL}) \subseteq \Pi_{\#recorrido,\text{sentido},\text{horaSalida}}(\text{VIAJES})$$

Aclaración: NO hay tablas vacías.

**Resolver las siguientes consultas en Álgebra Relacional:**

(1) Nombre del personal que realizó todos sus viajes en recorridos con origen en 'Ciudad Vieja', cumpliendo la función de chofer.

**SOLUCIÓN A:**

$$A = \Pi_{\text{ci}} \left( \sigma_{\text{origen} = \text{'Ciudad Vieja'} \wedge \text{función} = \text{chófer}} (\text{VIAJES_PERSONAL} * \text{RECORRIDO}) \right)$$

$$B = \Pi_{\text{ci}} \left( \sigma_{\text{NOT}(\text{origen} = \text{'Ciudad Vieja'} \wedge \text{función} = \text{chófer})} (\text{VIAJES_PERSONAL} * \text{RECORRIDO}) \right)$$

$$\text{SOL} = \Pi_{\text{nombre}} ((A - B) * \text{PERSONAL})$$

**SOLUCIÓN B:**

$$A = \Pi_{\text{ci}} \left( \sigma_{\text{origen} = \text{'Ciudad Vieja'} \wedge \text{función} = \text{chófer}} (\text{VIAJES_PERSONAL} * \text{RECORRIDO}) \right)$$

$$B = \Pi_{\text{ci}} \left( \sigma_{\text{origen} = \text{'Ciudad Vieja'} \wedge \text{función} \neq \text{chófer}} (\text{VIAJES_PERSONAL} * \text{RECORRIDO}) \right)$$

$$\text{SOL} = \Pi_{\text{nombre}} ((A - B) * \text{PERSONAL})$$

(2) Número de los recorridos (#recorrido) tales que todo el personal ingresado después del año 2010 realizaron algún viaje en esos recorridos. viaje en esos recorridos.

**SOLUCIÓN:**

$$A = \Pi_{\text{ci}} \left( \sigma_{\text{añoIngreso} > 2010} (\text{PERSONAL}) \right)$$

$$\text{SOL} = \Pi_{\#recorrido, \text{ci}} (\text{VIAJES_PERSONAL}) \% A$$

**Resolver las siguientes consultas en Cálculo Relacional de tuplas:**

(3) Matrícula y tipo de los coches que siempre realizan viajes en el mismo número de recorrido.

**SOLUCIÓN:**

$$\{ c.\text{matricula}, c.\text{tipo} / \text{COCHES}(c) \wedge (\exists v) (\text{VIAJES}(v) \wedge v.\text{matricula} = c.\text{matricula} \wedge \neg (\exists v_2) (\text{VIAJES}(v_2) \wedge v_2.\text{matricula} = v.\text{matricula} \wedge v_2.\#\text{recorrido} \neq v.\#\text{recorrido})) \}$$

(4) Paradas (#parada, nombre) tales que todos los recorridos que comparten esa parada tienen como destino "Punta Carretas"

**SOLUCIÓN:**

$$\{ p.\#\text{parada}, p.\text{nombre} / \text{PARADAS}(p) \wedge (\forall p_2) (\text{REC\_PARADAS}(p_2) \wedge p_2.\#\text{parada} = p.\#\text{parada} \rightarrow (\exists r) (\text{RECORRIDO}(r) \wedge r.\#\text{recorrido} = p_2.\#\text{recorrido} \wedge r.\text{destino} = \text{'Punta Carretas'})) \}$$

**Resolver la siguiente consultas en SQL:**

(5) Matrícula de los coches que hicieron algún viaje en alguno de los recorridos, considerando el sentido, con mayor cantidad de paradas.

**SOLUCIÓN:**

```
SELECT matricula
FROM Viajes
WHERE #recorrido IN (SELECT #recorrido
                     FROM Rec_Paradas
                     GROUP BY #recorrido,sentido
                     HAVING COUNT(*) >= ALL (SELECT COUNT(*)
                                             FROM Rec_Paradas
                                             GROUP BY #recorrido, sentido))
```

**Ejercicio 3 (25 puntos).**

- i. Dado un esquema de relación R, un conjunto de dependencias funcionales F y una descomposición ρ, defina qué significa que:
  - a) ρ preserva las dependencias funcionales.
  - b) ρ tiene join sin pérdida.

**SOLUCIÓN:**

- a) una descomposición ρ = {R<sub>1</sub>, R<sub>2</sub>, ..., R<sub>m</sub>} de R preserva las dependencias respecto a F si se

cumple que  $(\pi_{R_1}(F) \cup \pi_{R_2}(F) \cup \dots \cup \pi_{R_m}(F))^+ = F^+$ .

b) una descomposición  $\rho = \{R_1, R_2, \dots, R_m\}$  de  $R$  tiene join sin pérdida con respecto a  $F$ , si dada cualquier instancia  $r$  de  $R$  se cumple que el join de sus proyecciones sobre cada uno de los esquemas es exactamente igual a  $r$ . Esto es,  $(\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_m}(r)) = r$ .

ii. Dado un esquema relacional  $R$  se tienen dos descomposiciones  $D_1$  y  $D_2$  tales que:

a)  $D_1$  preserva las dependencias funcionales y no tiene JSP.

b)  $D_2$  tiene JSP y no preserva dependencias.

¿Cuál de las dos descomposiciones elegiría para representar la realidad? Justifique por qué elige una y por qué no elige la otra.

### SOLUCIÓN:

Una descomposición que no tenga JSP no sirve para representar la realidad, ya que no es posible determinar cuáles son las tuplas erróneas al reconstruir el join. Por otro lado, preservar las dependencias es una propiedad deseable, ya que permite garantizar la equivalencia de las descomposiciones que se construyen a partir de la descomposición original de la que se parte. Perder una dependencia funcional implica programar lo necesario para forzar a que se cumpla o bien asumir que no tiene por qué cumplirse.

Es por estos motivos que las descomposiciones deben ser con JSP y, preferentemente, con preservación de dependencias. En consecuencia, elegiría la descomposición  $D_2$ .

iii. Se tiene una descomposición  $\rho = (R_1, R_2)$  de un esquema relacional  $R$ , tal que alguna clave de  $R$  está en  $R_1$  y  $R_2$ . ¿Es posible afirmar que  $\rho$  tiene join sin pérdida? Justifique.

### SOLUCIÓN:

Si, es posible.

Sea  $X$  la clave de  $R$  que está en  $R_1$  y  $R_2$ . Dado que  $X$  está en  $R_1$  y en  $R_2$ , se tiene que  $X \subseteq R_1 \cap R_2$ . Es decir,  $R_1 \cap R_2$  es una superclave de  $R$  y, en consecuencia, se cumple que:

$$R_1 \cap R_2 \rightarrow R$$

Luego, por regla reflexiva de dependencia funcional se tiene que  $R_1 \cap R_2$  determina un subconjunto de atributos de  $R$ :

$$R_1 \cap R_2 \rightarrow R_1 \quad (\text{y} \quad R_1 \cap R_2 \rightarrow R_2)$$

Nuevamente, por regla reflexiva de dependencia funcional se tiene que  $R_1 \cap R_2$  determina un subconjunto de atributos de  $R_1$  y de  $R_2$ :

$$R_1 \cap R_2 \rightarrow R_1 - R_2 \quad (\text{y} \quad R_1 \cap R_2 \rightarrow R_2 - R_1)$$

Por tanto, la dependencia anterior está en  $F^+$ . Aplicando el teorema visto en el curso es posible afirmar que la descomposición  $\rho = (R_1, R_2)$  tiene JSP.

#

iv. Sea  $R(A,B,C,D,E,G)$  un esquema relación,  $F = \{A \rightarrow B, BA \rightarrow C, C \rightarrow DA, D \rightarrow E\}$  el conjunto de dependencias sobre  $R$ .

a) Determine qué forma normal está  $R$ . Justifique

### SOLUCIÓN:

Observando las dependencias, se puede ver que  $E$  no está en ninguna clave y que  $G$  está en todas. Además, por transitividad,  $C \rightarrow E$ .

Por otro lado,  $GC^+ = \{C, G, D, A, B, E\}$  por lo que  $CG$  es clave.

Por todo lo anterior,  $C \rightarrow E$  induce una dependencia parcial de una clave en un atributo no primo, por lo que R está en 1NF, dado que se viola 2NF.

b) Determine en qué forma normal está R con respecto a  $F' = F \cup \{E \rightarrow BG\}$ . Justifique

### SOLUCIÓN:

El nuevo conjunto de dependencias es  $F' = \{A \rightarrow B, BA \rightarrow C, C \rightarrow DA, D \rightarrow E, E \rightarrow BG\}$ .

Se calculan las claves. Para eso, hay que considerar que:

- i. G no se encuentra nunca a la izquierda por lo que no pertenece a ninguna clave.
- ii. El resto de los atributos aparecen tanto a la izquierda como a la derecha, por lo que pueden aparecer o no en un clave

$(A)_{F'}^+ = \{A, B, C, D, E, G\}$       A es clave de R

$(B)_{F'}^+ = \{B\}$

$(C)_{F'}^+ = \{C, D, A, B, E, G\}$       C es clave de R

$(D)_{F'}^+ = \{D, E, B, G\}$

$(E)_{F'}^+ = \{E, B, G\}$

Para saber si las únicas claves de R son A y C, debemos ver si entre el resto de los atributos que pueden estar en una clave queda una superclave, o sea, tenemos que ver la clausura de EBD:

$(EBD)_{F'}^+ = \{D, E, B, G\}$

Por esto, A y C son las únicas claves.

Luego, la dependencia  $D \rightarrow E$  viola 3NF ya que D no es superclave de R ni E es un atributo primo. Dado que ningún atributo no primo depende parcialmente de una clave, se tiene que el esquema de relación R está en 2NF.

### Ejercicio 4 (20 puntos).

Dadas las siguientes transacciones:

T1: w1(Z) r1(Y) w1(Y) c1

T2: r2(Y) w2(Y) r2(X) w2(Z) c2

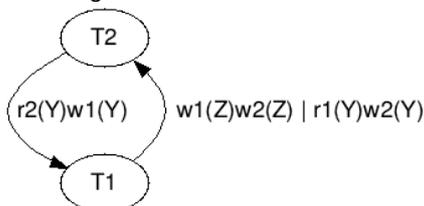
#### Parte A

Para cada una de las siguientes afirmaciones, indique si son verdaderas o falsas. **Justifique.**

- a. La siguiente historia es estricta: w1(Z) r2(Y) r1(Y) w1(Y) c1 w2(Y) r2(X) w2(Z) c2
- b. La siguiente historia evita abortos en cascada: w1(Z) r2(Y) r1(Y) w1(Y) c1 w2(Y) r2(X) w2(Z) c2
- c. La siguiente historia es serializable: w1(Z) r2(Y) r1(Y) w1(Y) c1 w2(Y) r2(X) w2(Z) c2
- d. Toda historia que evita abortos en cascada es serializable.

### SOLUCIÓN

- a. **Verdadero.** Para determinar si la historia presentada es estricta, debemos ver las operaciones de escritura, y si para esa misma variable, hay otra que la lee o escribe, la primer transacción tiene que haber confirmado. Las operaciones de escritura son  $w1(Z)$ ,  $w1(Y)$ ,  $w2(Y)$  y  $w2(Z)$ . Para el caso de  $w1(Z)$ , la operación  $w2(Z)$  hace que T2 escriba en T1. Cuando  $w2(Z)$  se ejecuta, T1 ya confirmó, por lo que en este caso no viola la definición. Para el caso de  $w1(Y)$  sucede exactamente lo mismo. Para los demás casos, no hay operaciones que lean de esas variables luego de ejecutar esas operaciones. Por lo tanto, la historia es estricta.
- b. **Verdadero.** En este caso, ninguna transacción lee de otra (ver definición de  $T_i$  lee de  $T_j$ ), por lo que se cumple la definición de historia que evita abortos en cascada.
- c. **Falso.** A continuación se presenta el grafo de seriabilidad:



Dado que el grafo tiene un ciclo, la historia no es serializable.

- d. **Falso.** La historia que presenta la parte b. evita abortos en cascada y es no serializable (ver demostraciones de partes b. y c.).

### Parte B

- a. Agregue bloqueos de escritura y de lectura a T1 y T2 de forma tal que las transacciones sigan 2PL estricto. Justifique su respuesta.

### SOLUCION:

Una solución posible es la siguiente:

T1:  $w1(Z), w1(Z), w1(Y), r1(Y), w1(Y), u1(Y), u1(Z), c1$   
T2:  $w2(Y), r2(Y), w2(Y), r2(X), r2(X), w2(Z), w2(Z), u2(Y), u2(Z), u2(X), c2$

Otra alternativa es la siguiente:

T1:  $w1(Z), w1(Z), w1(Y), r1(Y), w1(Y), c1, u1(Y), u1(Z)$   
T2:  $w2(Y), r2(Y), w2(Y), r2(X), r2(X), w2(Z), w2(Z), c2, u2(Y), u2(Z), u2(X)$

La diferencia importante es que en la primera alternativa se asume que poniendo los desbloqueos inmediatamente antes de los commits, es lo mismo que lo que se propone en la segunda (ponerlos inmediatamente después).

- b. ¿Es posible escribir una historia entrelazada de T1 y T2 de forma tal que las transacciones sigan 2PL estricto? Justifique su respuesta.

### SOLUCION:

El problema de estas transacciones es que las dos comienzan con los items en conflicto. Esto hace que las posibilidades de concurrencia sean bastante limitadas.

Si en la parte a se tomo la primer alternativa una posibilidad es la siguiente:

H1:  $w1(Z), w1(Z), w1(Y), r1(Y), w1(Y), u1(Y), w2(Y), r2(Y), w2(Y), r2(X), r2(X), u1(Z), w2(Z), w2(Z), u2(Y), u2(Z), c1$

En caso de la segunda alternativa eso no es posible dado que como los items en conflicto son los que aparecen primero en cada transacción, cada transacción debe esperar a que la otra desbloquee lo que obliga a que las únicas historias entre las dos transacciones sean seriales (una comienza, después que la otra termina).