

FUNDAMENTOS DE BASES DE DATOS

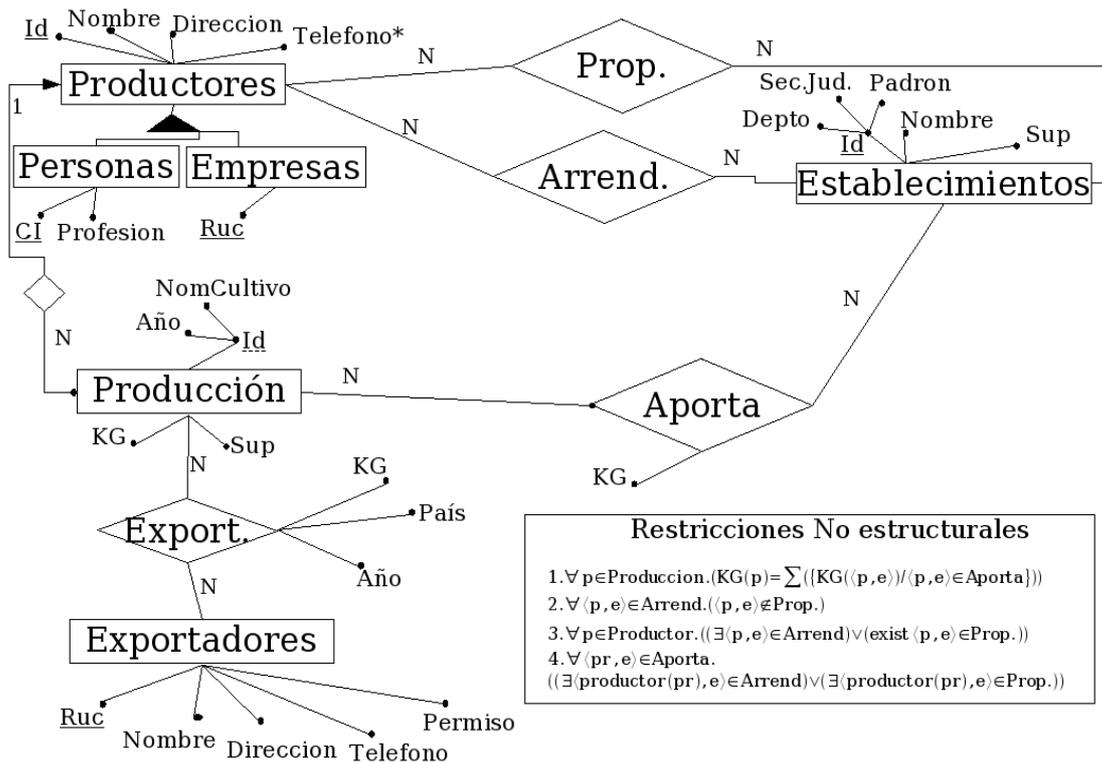
Examen Julio 2004

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.

Parte 1. Modelo Entidad-Relación (25 puntos)

Ejercicio 1. (25 pts)



SE PIDE: Modelo Entidad Relación completo, incluyendo restricciones de integridad.

Parte 2. Diseño Relacional (25 puntos)

Sol ejercicio 1

atributos

PA_NOM
PA_NAC
PA_UNIV
PA_EMAIL
ART_ID
ART_TITULO
ART_AUTOR
ART_IDIOMA
SES_TEMA
SES_FECHA
SES_TURNO
SES_SALON
HORA_PRES

1. Dependencias funcionales .

PA_EMAIL -> PA_NOM, PA_NAC, PA_UNIV
PA_EMAIL -> ART_IDIOMA
ART_ID -> ART_TITULO, ART_IDIOMA
ART_AUTOR, ART_TITULO -> ART_ID
SES_TEMA -> SES_FECHA, SES_TURNO, SES_SALON
SES_FECHA, SES_TURNO, SES_SALON -> SES_TEMA
ART_ID -> SES_TEMA, HORA_PRES
ART_ID -> PA_EMAIL
SES_TEMA, HORA_PRES -> ART_ID
HORA_PRES -> SES_TURNO

2. Dependencias por tabla.

Primeramente, se proyectan las dependencias anteriores en las tablas dadas.

PARTICIPANTES(PA_EMAIL, PA_NOM, PA_NAC, PA_UNIV, ART_IDIOMA)

PA_EMAIL -> PA_NOM, PA_NAC, PA_UNIV
PA_EMAIL -> ART_IDIOMA

SESIONES(SES_TEMA, SES_FECHA, SES_TURNO, SES_SALON)

SES_TEMA -> SES_FECHA, SES_TURNO, SES_SALON
SES_FECHA, SES_TURNO, SES_SALON -> SES_TEMA

ARTICULOS(ART_ID, ART_TITULO, ART_AUTOR, ART_IDIOMA)

ART_ID -> ART_TITULO, ART_IDIOMA
ART_AUTOR, ART_TITULO -> ART_ID

PROGRAMA(SES_TEMA, HORA_PRES, ART_ID, ART_TITULO, PA_EMAIL)

ART_ID -> ART_TITULO, PA_EMAIL
ART_ID -> SES_TEMA, HORA_PRES
SES_TEMA, HORA_PRES -> ART_ID

EXPOSITORES(PA_EMAIL, ART_AUTOR, ART_ID, SES_TEMA)

ART_ID -> PA_EMAIL

ART_ID -> SES_TEMA

Luego se buscan dependencias multivaluadas y no hay (a menos de las triviales o por complemento de las funcionales).

3. Indicar en que forma normal está el esquema propuesto.

Para esto hay que determinar la forma normal en que se encuentra cada tabla.

Participantes: 4NF, dado que no tiene dep. multivaluadas y todas las dependencias son de clave.

Sesiones: 4NF, dado que no tiene dep. multivaluadas y todas las dependencias son de clave.

Articulos: 2NF, dado que las claves de esa tabla son ART_AUTOR, ART_TITULO y ART_AUTOR, ART_ID y por lo tanto ART_ID -> ART_TITULO, ART_IDIOMA viola 3 NF porque tiene una parte de una clave del lado izquierdo y atributos no primos del lado derecho.

Programa: 4NF, dado que no tiene deps. multivaluadas todas las dependencias que tiene son de clave.

Expositores: 2NF, La clave de esa tabla debiera ser ART_AUTOR, ART_ID, por lo que las dependencias que se ven violan 3NF.

Conclusión: El esquema está en 2NF.

4. Indicar si el esquema preserva dependencias y en caso que se pierdan, dar la menos una dependencia que se pierda.

Para ver esto tenemos que ver si la union de las dependencias proyectadas en las tablas cubre a todas las dependencias del conjunto original.

Esta union es la siguiente:

PA_EMAIL -> PA_NOM, PA_NAC, PA_UNIV

PA_EMAIL -> ART_IDIOMA

SES_TEMA -> SES_FECHA, SES_TURNO, SES_SALON

SES_FECHA, SES_TURNO, SES_SALON -> SES_TEMA

ART_ID -> ART_TITULO, ART_IDIOMA

ART_AUTOR, ART_TITULO -> ART_ID

SES_TEMA, HORA_PRES -> ART_ID

ART_ID -> PA_EMAIL

ART_ID -> SES_TEMA, HORA_PRES

Observando las dependencias originales en el orden expuesto anteriormente se puede ver que la primera dependencia que no aparece directamente en este conjunto es

HORA_PRES -> SES_TURNO

Por lo tanto, es necesario ver si esta dependencia se cumple en la union de las proyecciones haciendo la clausura de los atributos de la izquierda:

$(HORA_PRES)^+ = \{HORA_PRES\}$, Con lo que es evidente que esa dependencia se pierde.

Sol ejercicio 3

$F = \{C \rightarrow D, A \rightarrow C, AD \rightarrow GBH, B \rightarrow A, BD \rightarrow H\}$

(1) CLAVES DE R

E no está a la derecha en ninguna dependencia, entonces está en toda clave.

$(E)^+ = \{E\}$ -> NO es clave
 $(EA)^+ = \{E A C D G B H\}$ -> es clave
 $(EB)^+ = \{E B A C D G H\}$ -> es clave
 $(EC)^+ = \{E C D\}$ -> NO es clave
 $(ED)^+ = \{E D\}$ -> NO es clave
 $(EG)^+ = \{E G\}$ -> NO es clave
 $(EH)^+ = \{E H\}$ -> NO es clave

EA y EB son todas las claves de 2 atributos.

Busco claves de 3 atributos, que contengan a E (está en toda clave) y no contengan a las claves ya encontradas.

$(ECD)^+ = \{E C D\}$ -> NO es clave
 $(ECH)^+ = \{E C D H\}$ -> NO es clave
 $(ECG)^+ = \{E C D G\}$ -> NO es clave

$(EDG)^+ = \{E D G\}$ -> NO es clave
 $(EDH)^+ = \{E D H\}$ -> NO es clave
 $(EGH)^+ = \{E G H\}$ -> NO es clave

No hay claves de 3 atributos.

Busco claves de 4 y de 5

$(ECDG)^+ = \{E C D G\}$ -> NO es clave
 $(ECDH)^+ = \{E C D H\}$ -> NO es clave
 $(ECGH)^+ = \{E C D G H\}$ -> NO es clave
 $(ECDGH)^+ = \{E C D G H\}$ -> NO es clave

EA y EB son todas las claves

(2) DESCOMPOSICION EN 3NF CON JSP Y SIN PERDER DEPENDENCIAS

Buscamos un cubrimiento minimal.

1. separo dependencias con mas de un atrib a la dcha.

$C \rightarrow D$
 $A \rightarrow C$
 $AD \rightarrow G$
 $AD \rightarrow B$
 $AD \rightarrow H$
 $B \rightarrow A$
 $BD \rightarrow H$

2. elimino atributos redundantes

Para esto verifico en las dependencias con mas de un atrib a la izq si alguno de los atributos pertenece a la clausura del otro. Si es así, ese atrib es redundante

$A^+ = \{A C D\}$ -> $D \in A^+ \Rightarrow D$ es redundante

$B^+ = \{B A C D\}$ -> $D \in B^+ \Rightarrow D$ es redundante

El conjunto de dependencias luego de eliminar atrib redundantes queda:

$C \rightarrow D$
 $A \rightarrow C$
 $A \rightarrow G$
 $A \rightarrow B$
 $A \rightarrow H$
 $B \rightarrow A$
 $B \rightarrow H$

3. elimino dependencias redundantes

Una dependencia es redundante si su lado derecho pertenece a la clausura del lado izquierdo sobre el conjunto de dependencias, exepctuando la dep. en cuestión.

Los casos a tener en cuenta son $A \rightarrow H$ y $B \rightarrow H$.

(A+) $F-A \rightarrow H = \{ A C G B D H \}$ \rightarrow ES UNA DEPENDENCIA REDUNDANTE
(B+) $F-B \rightarrow H = \{ B A H C G D \}$ \rightarrow ES UNA DEPENDENCIA REDUNDANTE

Se debe eliminar una de ellas, no ambas, dado que al eliminar una la otra deja de ser redundante. Es indistinto cual de las 2 se elimina. Eliminamos $A \rightarrow H$

El cubrimiento minimal obtenido es:

$C \rightarrow D$
 $A \rightarrow C$
 $A \rightarrow G$
 $A \rightarrow B$
 $B \rightarrow A$
 $B \rightarrow H$

Ahora encontramos una descomposición en 3NF. Unimos todas las dependencias con igual lado izquierdo:

$A \rightarrow C$
 $A \rightarrow G$
 $A \rightarrow B$
 $R1(ACGB)$
 $B \rightarrow A$
 $B \rightarrow H$
 $R2(BAH)$
 $C \rightarrow D$
 $R3(CD)$

Para garantizar JSP es necesario de que alguna clave esté incluida en alguna tabla. Ni EA ni EB están incluidos en ninguna de las tablas, por lo que generamos una nueva

$R4(EA)$

(3) DESCOMPOSICION EN 4NF

Como no hay dependencias multivaluadas, se obtiene el resultado correcto aplicando el algoritmo de BCNF.

No hay ninguna dependencia que viole BCNF, por lo que el esquema está en 4NF.

Parte 3. Consultas (25 puntos)

Ejercicio

Una empresa que se dedica a ofrecer servicios para cumpleaños infantiles maneja su información en una base de datos con las siguientes tablas:

LOCALES (codloc, capacidad, dir, exterior?)

ANIMACIONES (codgrupo, tipo, fecha_inicio, cod_responsable)

INFLABLES (codinf, superbase, altura, cantact, fechaincorp)

CLIENTES (codcli, nombre, mes_nac, año_nac, responsable, telefono)

FIESTAS (codfiesta, codcli, fecha)

FIESTA-LOCAL (codfiesta, codloc)

FIESTA-ANIMACION (codfiesta, codgrupo)

FIESTA-INFLABLE (codfiesta, codinf)

SE PIDE:

1. Resolver las siguientes consultas en Álgebra Relacional:

a. **Obtener el código de los inflables que solo han sido contratados para fiestas realizadas en los locales de la empresa.**

$$A = \Pi_{\text{CODFIESTA}}(\text{FIESTAS}) - \Pi_{\text{CODFIESTA}}(\text{FIESTA_LOCAL})$$

$$\text{SOL} = \Pi_{\text{CODINF}}(\text{FIESTA_INFLABLE} * \text{FIESTA_LOCAL}) - \Pi_{\text{CODINF}}(\text{FIESTA_INFLABLE} * A)$$

b. **Obtener los códigos de los clientes que han contratado todos los tipos de animación que ofrece la empresa.**

$$A = \Pi_{\text{CODFIESTA}, \text{TIPO}}(\text{FIESTA_ANIMACION} * \text{ANIMACIONES})$$

$$B = \Pi_{\text{CODCLI}, \text{TIPO}}(A * \text{FIESTAS})$$

$$\text{SOL} = A \% \Pi_{\text{TIPO}}(\text{ANIMACIONES})$$

2. Resolver las siguientes consultas en Cálculo Relacional:

c. **Obtener el nombre de los clientes tales que todas las fiestas que han realizado solo contrataron local.**

$$\{ \text{c.nombre} / \text{CLIENTES}(c) \wedge (\exists f)(\text{FIESTA}(f) \wedge f.\text{codcli} = c.\text{codcli} \wedge (\exists fl)(\text{FIESTA_LOCAL}(fl) \wedge fl.\text{codfiesta} = f.\text{codfiesta} \wedge \neg ((\exists fa)(\text{FIESTA_ANIMACION}(fa) \wedge fa.\text{codfiesta} = f.\text{codfiesta}) \vee (\exists fi)(\text{FIESTA_INFLABLE}(fi) \wedge fi.\text{codfiesta} = f.\text{codfiesta})))) \}$$

d. **Obtener el nombre de los clientes que han contratado locales y siempre contrataron locales que tienen espacio exterior.**

$$\{ \text{n.nombre} / \text{CLIENTES}(n) \wedge (\exists f)(\text{FIESTA}(f) \wedge f.\text{codcli} = n.\text{codcli} \wedge (\exists fl)(\text{FIESTA_LOCAL}(fl) \wedge fl.\text{codfiesta} = f.\text{codfiesta} \wedge$$

```

        )
        )
        )
        )
    }

```

3. Resolver las siguientes consultas en SQL

e. **Obtener el nombre de los clientes y la cantidad de fiestas para las cuales contrataron por lo menos dos tipos de animaciones distintas.**

```

SELECT C.codcli, C.nombre, COUNT(DISTINCT F.codfiesta)
FROM CLIENTES C, FIESTAS F, FIESTA_ANIMACION FA1,
      FIESTA_ANIMACION FA2, ANIMACIONES A1, ANIMACIONES A2
WHERE C.codcli = F.codcli AND F.codfiesta= FA1.codfiesta AND
      F.codfiesta=FA2.codfiesta AND FA1.codgrupo=A1.codgrupo
      AND FA2.codgrupo=A2.codgrupo AND A1.tipo <> A2.tipo

GROUP BY C.codcli

```

f. Para cada inflable obtener la cantidad de fiestas no realizadas en locales de la empresa para las cuales han sido contratados.

```

SELECT FI.codinf, COUNT(*)
FROM FIESTA_INFLABLE FI
WHERE FI.codfiesta NOT IN
      ( SELECT codfiesta
        FROM FIESTA_LOCAL
      )
GROUP BY FI.codinf

```

Parte 4 Optimización y Concurrencia (25 puntos)

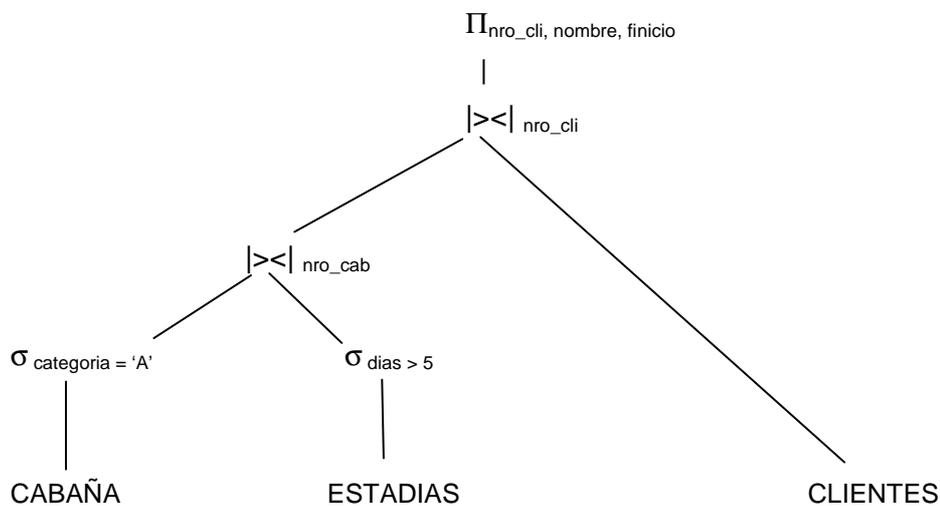
Ejercicio XXXXX.

Dado el siguiente esquema y la siguiente consulta:

CABAÑA (nro_cab, capacidad, zona, categoría)
CLIENTES (nro_cli, nombre, fecontacto, origen)
ESTADIAS (nro_cab, nro_cli, finicio, dias)

```
SELECT C.nro_cli, C.nombre, E.finicio
FROM Clientes C, Estadias E, Cabaña A
WHERE C.nro_cli = E.nro_cli AND E.nro_cab = A.nro_cab
AND A.categoría = 'A' AND E.dias > 5
```

Considerando el siguiente plan lógico para la consulta, con los tamaños intermedios calculados:



Tamaños:

Llamamos:

t1 al resultado de aplicar $\sigma_{\text{categoría} = 'A'}$

t2 al resultado de aplicar $\sigma_{\text{días} > 5}$

t3 al resultado de aplicar $\sigma_{\text{nro_cab}}$

t4 al resultado de aplicar $\sigma_{\text{nro_cli}}$

T(CABAÑA) = 15

T(ESTADIAS) = 1300

T(CLIENTES) = 600

T(t1) = 5

T(t2) = 975

T(t3) = 325

T(t4) = 325

Se pide :

Dar un plan físico adecuado para la consulta y calcular su costo total.

Nota: Considerar costo de escritura en disco para todas las operaciones, excepto la última.

DATOS:

	CLIENTES	ESTADIAS	CABAÑA	CABAÑA >< ESTADIA
Cantidad tuplas por bloque (fbl)	20	30	20	25
Indices primarios	nro-CLI (niveles: 3)			
Indices secundarios (B+)		días (niveles: 2)		

FORMULAS:

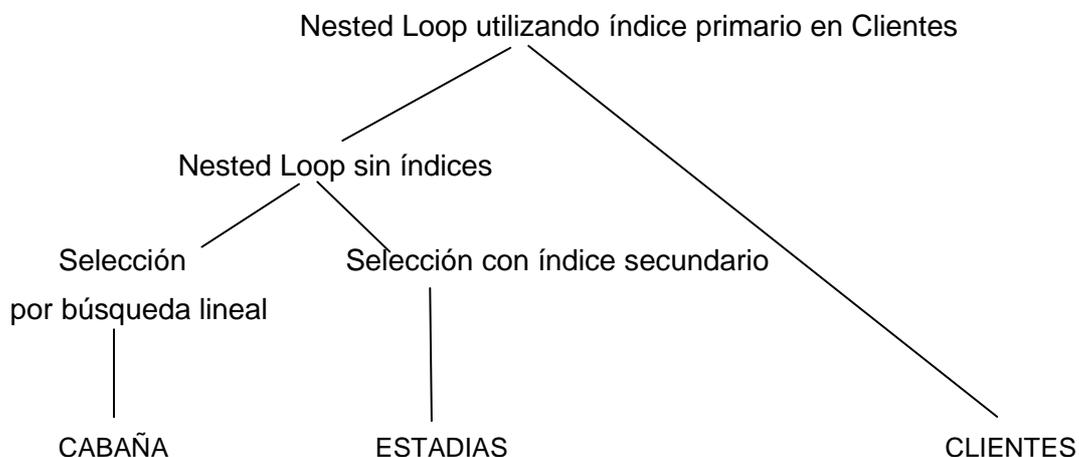
Selección (R)	Búsqueda lineal	b
	Búsqueda binaria	$\log_2 b + \lceil s/fbl_R \rceil - 1$
	Índice primario	$x + 1$
	Índice secundario (B+)	$x + s$
Join (R,S)	Nested Loop (ciclo anidado) sin utilizar índices	$b_R + b_R * b_S$
	Nested Loop (ciclo anidado) utilizando índice secundario para recuperar tuplas que matchean	$b_R + R * (x + s)$ $s = \text{card. de la selección en S por el atributo del join.}$
	Nested Loop (ciclo anidado) utilizando índice primario para recuperar tuplas que matchean	$b_R + R * (x + 1)$
Escribir resultados en disco	Todas la operaciones	n_{res} / fbl_{res}

Notación: b – cantidad de bloques
 fbl – factor de bloqueo
 x – cantidad de niveles del índice
 $|R|$ - cantidad de tuplas de R
 s – cardinalidad de la selección (nro. de tuplas del resultado)
 n_{res} – nro. de tuplas del resultado

Aclaración: Se considera que existe 1 buffer por relación y 1 buffer para el resultado (3 buffers).

SOLUCION:

Plan físico:



Costos:

$$C(\sigma_{\text{categoria} = 'A'}) = b = 1$$

$$C(\sigma_{\text{dias} > 5}) = x + s = 2 + 975 = 977$$

$$C(|><|_{\text{nro_cab}}) = b_R + b_R * b_S = 1 + 1 * 975 / 30 = 1005 / 30 = 34$$

$$C(|><|_{\text{nro_cli}}) = b_R + |R| * (x + 1) = 325 / 25 + 325 * 4 = 1313$$

Costos de grabar en disco:

$$C(t1) = 1$$

$$C(t2) = 975 / 30 = 33$$

$$C(t3) = 325 / 25 = 13$$

$$\text{Costo total de la consulta} = 1 + 977 + 34 + 1313 + 1 + 33 + 13 = 2372$$

Ejercicio XXXXX. (10 pts.)

Dadas las siguientes transacciones T_1, T_2 :

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

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

a) Para cada historia decir si es serializable y si es recuperable, justificando.

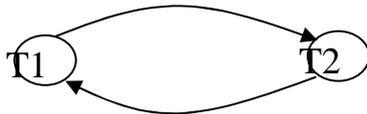
- i) $r_1(X), r_1(Y), r_2(X), w_1(Y), w_1(X), r_2(Y), w_2(X), w_2(Z), c_1, c_2$
- ii) $r_2(X), r_2(Y), w_2(X), r_1(X), r_1(Y), w_1(Y), w_1(X), c_1, w_2(Z), c_2$
- iii) $r_1(X), r_1(Y), r_2(X), r_2(Y), w_1(Y), w_1(X), w_2(X), w_2(Z), c_2, c_1$

b) Indique como haría los bloqueos y desbloqueos en las transacciones para garantizar serializabilidad y a la vez permitir que se entrelacen. Justifique. (Además, debe escribir las transacciones con los cambios propuestos).

SOLUCION

a)

i) Grafo de precedencia, considerando $\langle r_2(X), w_1(X) \rangle$ y $\langle w_1(X), w_2(X) \rangle$:



No es serializable.

T_1 no lee de T_2 . T_2 lee de T_1 . T_1 hace commit antes que T_2 . **Es recuperable.**

ii) Grafo de precedencia, considerando todas las operaciones en conflicto:



Es serializable.

T_1 lee de T_2 . T_2 hace commit antes que T_1 . **No es recuperable.**

iii) Grafo de precedencia, considerando $\langle r_2(Y), w_1(Y) \rangle$ y $\langle w_1(X), w_2(X) \rangle$:



No es serializable.

Ninguna transacción lee de la otra. **Es recuperable.**

b)

$T_1: l_1(X), r_1(X), l_1(Y), r_1(Y), w_1(Y), u_1(Y), w_1(X), u_1(X), c_1$

$T_2: l_2(X), r_2(X), l_2(Y), r_2(Y), w_2(X), l_2(Z), u_2(X), u_2(Y), w_2(Z), u_2(Z), c_2$

Se pueden entrelazar:

$l_2(X), r_2(X), l_2(Y), r_2(Y), w_2(X), l_2(Z), u_2(X), u_2(Y), l_1(X), r_1(X), l_1(Y), r_1(Y), w_1(Y), u_1(Y), w_1(X), u_1(X), c_1, w_2(Z), u_2(Z), c_2$

Se asegura serialidad en todas las historias porque las transacciones siguen el protocolo 2PL, es decir que en cada transacción todos los bloqueos preceden a todos los desbloqueos.