

## FUNDAMENTOS DE BASES DE DATOS

### Examen Febrero 2014

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 Ud. deberá comprender la letra y construir un modelo entidad relación incluyendo restricciones no estructurales.

La letra está separada en varias partes y para resolver cada parte Ud. deberá hacer una parte del diagrama general. Para esto, puede copiar lo que considere necesario para la comprensión del diagrama o incluso, copiar con cambios algunas partes de la etapa anterior. Por ejemplo, si pone una relación, debe poner las entidades participantes, aunque puede omitir los atributos de la entidad que está resuelta en otra parte.

En la última parte, deberá construir el modelo completo.

Una asociación de productores y comerciantes de productos agrícolas desea llevar adelante un proceso de trazabilidad de la producción. La trazabilidad consiste en que cada vendedor, pueda obtener la información de cada producto que vende.

- a) De cada empresa se conoce su RUT (Registro Único Tributario), nombre y dirección. A su vez, las empresa pueden ser productoras o vendedoras. Cada empresa productora, opera uno o más establecimientos. De cada establecimiento se conoce su nombre que sirve como identificador y su dirección. Hay que tener en cuenta que muchas veces, diferentes empresas operan el mismo establecimiento.

**Se pide:** construir el MER para esta parte de la realidad, incluyendo RNE's si son necesarias.

- b) De cada producto agrícola (ej: lechuga mantecosa, durazno pavía) se conoce un identificador, un descriptor y un conjunto de nutrientes que aporta, conociéndose además del nombre, la cantidad que aporta y el tipo (carbohidrato, grasa saturada, etc.) del nutriente. Además los productos pueden ser para consumo humano, en cuyo caso se conoce la fecha de autorización para su consumo por parte de la FDA, de consumo animal en cuyo caso se conoce el conjunto de especies que lo pueden consumir, o para industria (fabricación de conservas, aceites, combustibles, etc.). Los productos de consumo animal no pueden ser consumidos por humanos, pero cualquiera puede ser utilizado para industria. Pueden eventualmente, existir productos que aún no se sabe que destino tienen. Por otra parte, tenga en cuenta que normalmente hay productos industriales (Ej: Salsa de tomates) que son para consumo humano y otros que son para consumo animal (Ej: Raciones).

Por otro lado, los productos agrícolas pueden ser verduras, en cuyo caso se sabe si son de hoja o no; granos, de los que se sabe si son oleaginosos o no; o frutas. Cualquier producto existente debe ser catalogado de una y sólo una de estas formas.

**Se pide:** construir el MER para esta parte de la realidad, incluyendo RNE's si son necesarias.

- c) En cada establecimiento, existen predios que son responsabilidad de sólo una de las empresas que opera en el establecimiento. Cada predio tiene un número que lo identifica dentro del establecimiento y la empresa .

**Se pide:** construir el MER para esta parte de la realidad, incluyendo RNE's si son necesarias.

- d) En cada predio, están ubicadas un conjunto de plantas de las que se conoce la ubicación dentro del predio y su tipo, o sea, de qué producto agrícola se está hablando. Cada planta es de un único tipo (ej: la planta en la ubicación 32 del predio XXXXXX es de tipo lechuga mantecosa). Además se

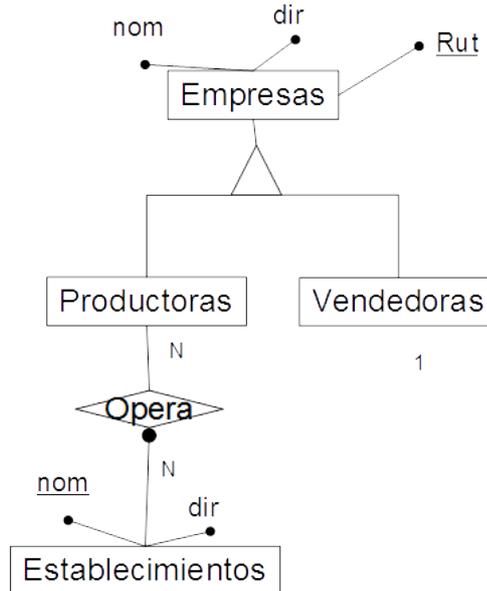
conoce la fecha de vencimiento, esto es la fecha máxima en la cual puede ser utilizada para el destino que sea. Cada planta, puede ser vendida por una empresa vendedora. Para cada planta y empresa vendedora, se conoce la fecha en que entró esa planta a la empresa.

**Se pide:** construir el MER para esta parte de la realidad, incluyendo RNE's si son necesarias.

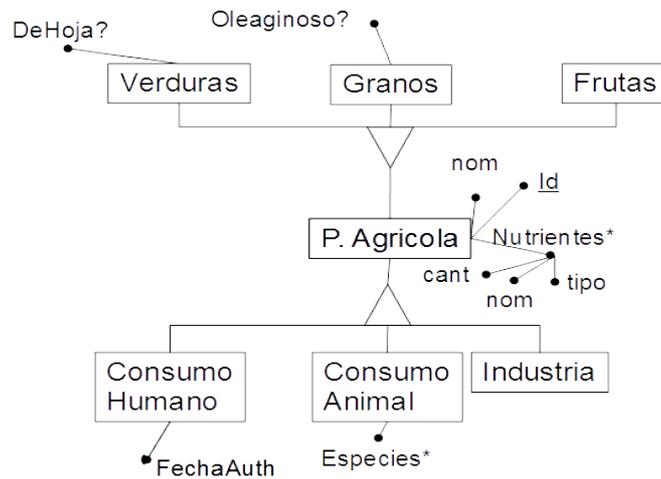
- e) Compagine el MER completo de la realidad anterior. Para cada entidad debe colocar al menos los atributos identificadores. Agregue las restricciones de integridad pertinentes que no hayan sido incluidas en partes anteriores.

**SOLUCION:**

a)

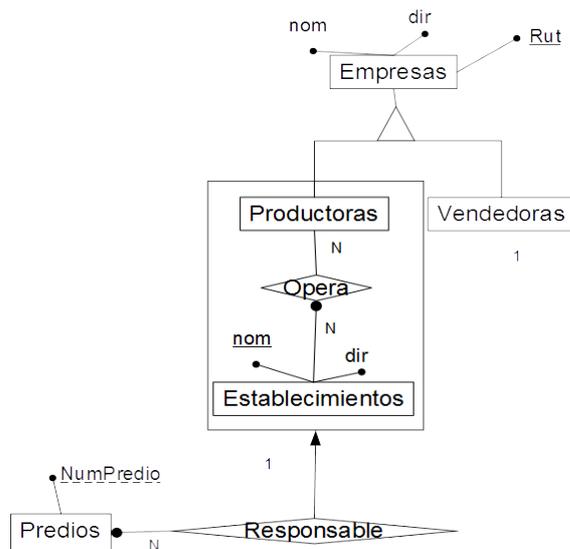


b)

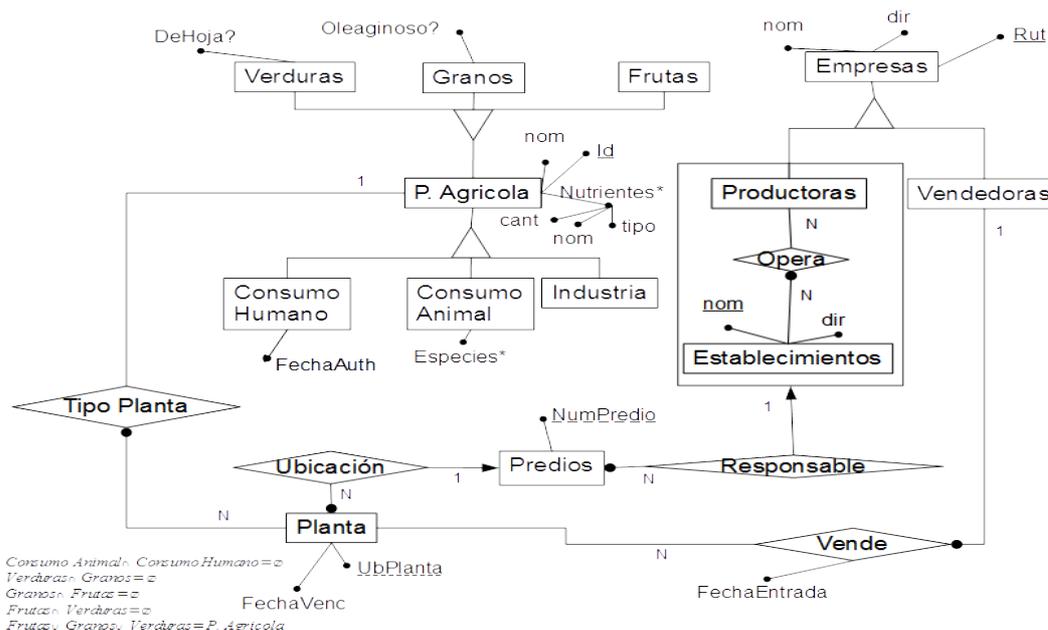


$Consumo\ Animal \cap Consumo\ Humano = \emptyset$   
 $Verduras \cap Granos = \emptyset$   
 $Granos \cap Frutas = \emptyset$   
 $Frutas \cap Verduras = \emptyset$   
 $Frutas \cup Granos \cup Verduras = P.\ Agricola$

c)



d)



e)

En la parte anterior ya quedó completo el diagrama.

## Ejercicio 2 (25 puntos).

Se desea organizar un "BOOTCAMP". Estos eventos consisten en varios días de trabajo intensivo sobre uno o más temas. El evento está organizado en talleres que duran un día cada uno, en los que se trata un tema determinado y se realizan en un día determinado (no se repiten).

De cada tema se conoce un identificador, un nombre y su descripción.

Cada taller está organizado por un "gurú" del que se conoce el nombre completo y la empresa para la que trabaja, además de su curriculum. Cada una de estas personas puede dar más de un taller. Tenga en cuenta que pueden haber diferentes oradores que hablen de los mismos temas, aunque no lo pueden hacer

en los mismos días. Por otro lado, pueden hacerse varios talleres en el mismo día, siempre que sean de temas distintos. Además hay un único orador por taller y cada taller es de todo el día. Observe que un gurú no puede dar más de un taller en un mismo día.

No hay dos personas con el mismo nombre en la misma empresa.

De los asistentes se conoce su documento, su nombre y su email. No se admite que un asistente haga dos talleres el mismo día.

1. Quien está diseñando esta base de datos, comenzó a construir la siguiente relación universal:

R(*fechaT*, *idTema*, *nomTema*, *descTema*, *nomGuru*, *empGuru*, *currGuru*, *docAsis*, *nomAsis*, *mailAsis*)

- a) Determine las dependencias funcionales que se cumplen en esta tabla.
- b) Indique en qué forma normal está esta tabla. Justifique su respuesta.

2. En un proceso posterior, el diseñador dejó anotadas las siguientes tablas:

**Taller(*fechaT*, *idTema*, *nomTema*, *descTema*, *nomGuru*, *empGuru*, *currGuru*)**  
**Asistencia(*docAsis*, *nomAsis*, *mailAsis*, /\* acá van los datos del taller \*/)**

- a) Sustituya el comentario en la tabla **Asistencia** por el conjunto mínimo de atributos necesarios de la tabla taller para que la descomposición sea con join sin pérdida. Justifique su respuesta.
- b) Indique en qué forma normal está la descomposición.

3. Considere la siguiente tabla:

Q(*fechaT*, *idTema*, *nomGuru*, *nomAsis*)

Indique al menos una dependencia multivaluada no funcional ni trivial que se cumpla sobre esta tabla. Justifique su respuesta.

## **SOLUCION.**

1. a)

R(*fechaT*, *idTema*, *nomTema*, *descTema*, *nomGuru*, *empGuru*, *currGuru*, *docAsis*, *nomAsis*, *mailAsis*)

Un tema está identificado por su *idTema*, por lo que:

*idTema* --> *nomTema*, *descTema*

Un gurú, esta identificado por su nombre y la empresa en la que trabaja. Por esto:

*nomGuru*, *empGuru* --> *currGuru*

Un asistente está identificado por su documento por lo que:

*docAsis* --> *nomAsis*, *mailAsis*

Un taller está identificado por la *fechaT* y el Tema, además tiene un único orador (gurú).

*FechaT*, *idTema* --> *nomGuru*, *empGuru*

Por otra parte, cada taller dura un día entero por lo que un gurú en un día, no puede dar más de un taller, o sea, no puede hablar de más de un tema.

*FechaT*, *nomGuru*, *empGuru* --> *idTema*

Como un asistente no puede hacer dos talleres el mismo día entonces no puede escuchar de más de un tema en cada día, por lo que se cumple:

*FechaT*, *docAsis* --> *idTema*

De esta forma, las dependencias son:

*idTema* --> *nomTema*, *descTema*

*nomGuru*, *empGuru* --> *currGuru*

*docAsis* --> *nomAsis*, *mailAsis*

*FechaT*, *idTema* --> *nomGuru*, *empGuru*

*FechaT*, *docAsis* --> *idTema*

*FechaT*, *nomGuru*, *empGuru* --> *idTema*

1.b) Indique en qué forma normal está esta tabla. Justifique su respuesta.

La clausura de  $idTema$  es:  $docAsis^+ = \{nomAsis, mailAsis\}$ . Por esto  $docAsis$  no es clave.

Por otro lado, si consideramos  $fechaT, docAsis$  nos encontramos con que:

$(fechaT, docAsis)^+ = \{ fechaT, docAsis, idTema, nomTema, descTema, nomGuru, empGuru, currGuru, nomAsis, mailAsis\}$

y

$fechaT^+ = \{fechaT\}$ .

Por todo lo anterior, vemos que  $fechaT, docAsis$  es una clave de  $R$  pero  $docAsis$  no lo es. Además  $nomAsis$  claramente no participa de ninguna clave dado que sólo aparece a la derecha de las dependencias.

Por todo esto,  $docAsis \rightarrow nomAsis$  induce una dependencia parcial de un atributo no primo desde una clave, por lo que viola 2NF.

Conclusión:

$R$  está en 1NF.

2. a) Sustituya el comentario en la tabla asistentes por el conjunto mínimo de atributos necesarios de la tabla taller para que la descomposición sea con join sin pérdida. Justifique su respuesta.

El primer punto es ver qué dependencias se cumplen sobre la tabla

Taller( $fechaT, idTema, nomTema, descTema, nomGuru, empGuru, CurrGuru$ )

por lo que hay que proyectar:

$\Pi_{taller}(F) = \{idTema \rightarrow nomTema, descTema; nomGuru, empGuru \rightarrow currGuru; FechaT, idTema \rightarrow nomGuru, empGuru; FechaT, nomGuru, empGuru \rightarrow idTema\}$

De esta forma,  $fechaT, idTema$  es clave en esta tabla. (verificar la clausura).

Si la tabla Asistencia se construye de la siguiente forma:

Asistencia( $docAsis, nomAsis, mailAsis, fechaT, idTema$ )

Se obtiene que  $Asistencia \cap Taller = \{fechaT, idTema\}$

y  $Taller - Asistencia = \{nomTema, descTema, nomGuru, empGuru, CurrGuru\}$

por lo que se cumple una de las dependencias indicadas por el teorema de join sin pérdida.

Esto nos garantiza que esta descomposición tiene JSP.

2.b) Indique en qué forma normal está la descomposición.

La descomposición sigue estando en 1NF por la misma dependencia que antes ( $docAsis \rightarrow nomAsis, mailAsis$ ) sobre Asistencia.

3.

Si se proyectan las dependencias funcionales originales sobre  $Q$  nos encontramos con

$fechaT, idTema \rightarrow nomGuru$ . Por la regla de replicación hay una dependencia multivaluada con los mismos atributos de cada lado. Si consideramos la regla de complemento sobre  $Q$ , nos encontramos con la siguiente dependencia multivaluada que sabemos que no es funcional, dado que a cada taller pueden asistir muchas personas:

$fechaT, idTema \twoheadrightarrow nomAsis$

### Ejercicio 3 (25 puntos).

Considere una versión simplificada de un *catálogo* o *diccionario de datos* de un manejador de bases de datos relacionales (RDBMS). El catálogo o diccionario está conformado por un conjunto de tablas y vistas que contienen información acerca de las estructuras almacenadas en las diferentes bases de datos que el RDBMS soporta.

Se tienen las siguientes tablas:

**ESQUEMAS** (*idEsquema*, *nomEsquema*, *fechaCreación*).

Almacena la información de los esquemas creados. Cada esquema agrupa un conjunto de objetos como pueden ser tablas, índices y atributos. Para cada esquema se conoce su identificador, su nombre que lo describe y la fecha en que fue creado.

**TABLAS** (*idTabla*, *idEsquema*, *nomTabla*, *definición*).

Almacena la información de las tablas creadas en un esquema. Para cada una se conoce su identificador, el esquema al que pertenece, su nombre y su definición que es el texto SQL de las sentencias de creación de la misma.

**ATRIBUTOS** (*idAtributo*, *idTabla*, *nomAtributo*, *tipoAtributo*, *clave*)

Almacena la información de los atributos de las tablas de un esquema. Para cada uno se conoce su identificador, la tabla a la que pertenece, su nombre, el tipo de dato admitido y si forma parte o no de la clave primaria de la tabla a la que pertenece (S por SI, N por No).

**INDICES** (*idIndice*, *idTabla*, *nomIndice*, *tipoIndice*)

Contiene la información de los índices creados sobre las tablas de un esquema. Para cada uno se conoce su identificador, la tabla sobre la cual fue creada, su nombre y el tipo del índice (BTREE o HASH).

**ATRIBUTOS\_INDICES** (*idIndice*, *idAtributo*, *posición*, *orden*)

Almacena la información de los atributos que componen los diferentes índices. Para cada atributo de cada índice se conoce la posición que ocupa dentro de la composición del índice. El primer atributo ocupa la posición 1 y así sucesivamente. También se sabe si los registros para dicho atributo en el índice se ordenan de forma ascendente (A) o descendente (D).

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

$$\Pi_{idEsquema}(TABLAS) \subseteq \Pi_{idEsquema}(ESQUEMAS)$$

$$\Pi_{idTabla}(ATRIBUTOS) \subseteq \Pi_{idTabla}(TABLAS)$$

$$\Pi_{idTabla}(INDICES) \subseteq \Pi_{idTabla}(TABLAS)$$

$$\Pi_{idAtributo}(ATRIBUTOS\_INDICES) \subseteq \Pi_{idAtributo}(ATRIBUTOS)$$

$$\Pi_{idIndice}(ATRIBUTOS\_INDICES) \subseteq \Pi_{idIndice}(INDICES)$$

Aclaración: NO hay tablas vacías.

**Resolver la siguiente consulta en Álgebra Relacional:**

(1) Nombre de las tablas, para aquellas tablas que tienen un índice de tipo BTREE que sólo incluye atributos de la clave primaria de la tabla, aunque no necesariamente a todos ellos.

**SOLUCIÓN:**

$$A = \sigma_{tipoIndice=BTREE} INDICES$$

Indices de tipo Btree

$$B = \Pi_{idIndice, idAtributo}(ATRIBUTOS\_INDICES * A)$$

Dado un índice btree todos sus atributos.

$$C = \Pi_{idAtributo}(\sigma_{clave='N'}(ATRIBUTOS))$$

atributos que no pertenecen a una clave.

$$D = \Pi_{idIndice}(B * C)$$

El índice (que es btree) que tiene un atributo de la tabla que no pertenece a la clave primaria.

$$Sol = \Pi_{nomTabla}(((\Pi_{idIndice}(B) - D) * INDICES) * TABLAS)$$

(2) Nombre de la tabla y nombre del esquema al que pertenece, para aquellas tablas que tienen índices, pero no tienen todos los tipos de índice.

**SOLUCIÓN:**

$A = \Pi_{idTabla, tipoIndice} (INDICES)$   
Identificadores de tablas con los tipos de todos sus índices.

$B = A \% \Pi_{tipoIndice} (INDICES)$   
Identificadores de tablas que tienen todos los tipos de índices

$Sol = \Pi_{nomTabla, nomEsquema} (((\Pi_{idTabla} (A) - B) * TABLAS) * ESQUEMAS)$

**Resolver la siguiente consulta en Cálculo Relacional de tuplas:**

(3) Las parejas de nombres de tablas tales que pertenecen a distintos esquemas pero son iguales. Se dice que dos tablas son iguales si sus atributos coinciden en nombre, en tipo y sus claves primarias están compuestas por los mismos atributos (considerando el nombre).

**SOLUCIÓN:**

$$\{t1.nomTabla, t2.nomTabla / TABLAS(t1) \wedge TABLAS(t2) \wedge t1.idEsquema \neq t2.idEsquema \wedge$$

$$(\forall a) (ATRIBUTOS(a) \wedge a.idTabla = t1.idTabla$$

$$\rightarrow$$

$$(\exists a2) (ATRIBUTOS(a2) \wedge a2.idTabla = t2.idTabla \wedge$$

$$a.nomAtributo = a2.nomAtributo \wedge a.tipoAtributo = a2.tipoAtributo \wedge$$

$$a.clave = a2.clave$$

$$) \wedge$$

$$(\forall a) (ATRIBUTOS(a) \wedge a.idTabla = t2.idTabla$$

$$\rightarrow$$

$$(\exists a2) (ATRIBUTOS(a2) \wedge a2.idTabla = t1.idTabla \wedge$$

$$a.nomAtributo = a2.nomAtributo \wedge a.tipoAtributo = a2.tipoAtributo \wedge$$

$$a.clave = a2.clave$$

$$) \wedge$$

$$\}$$

(4) Nombre de las tablas y nombre de los esquemas al que pertenecen tal que la tabla tiene un índice que es de tipo HASH, y al mismo tiempo la tabla tiene todos sus atributos de clave en un índice de tipo BTREE.

**SOLUCIÓN:**

$$\{t.nomTabla, e.nomEsquema / TABLAS(t) \wedge ESQUEMAS(e)$$

$$\wedge t.idEsquema = e.idEsquema \wedge$$

$$(\exists i) (INDICE(i) \wedge i.tipo = 'Hash' \wedge i.idTabla = t.idTabla \wedge$$

$$(\exists i1) (INDICES(i1) \wedge i1.tipo = 'Btree' \wedge i1.idTabla = i.idTabla \wedge$$

$$(\forall a) (ATRIBUTOS(a) \wedge a.clave = 'S' \wedge a.idTabla = i.idTabla \rightarrow$$

$$(\exists ia1) (ATRIBUTOS\_INDICES(ia1) \wedge$$

$$ia1.idIndice = i1.idIndice \wedge$$

$$ia1.idAtributo = a.idAtributo)) \wedge$$

$$\}$$

**Resolver la siguiente consultas en SQL:**

(5) Nombre, tipo de los índices y cantidad de atributos que tienen, de aquellos índices que están definidos sobre más de 5 atributos. (Se asume que los índices son correctos).

**SOLUCIÓN:**

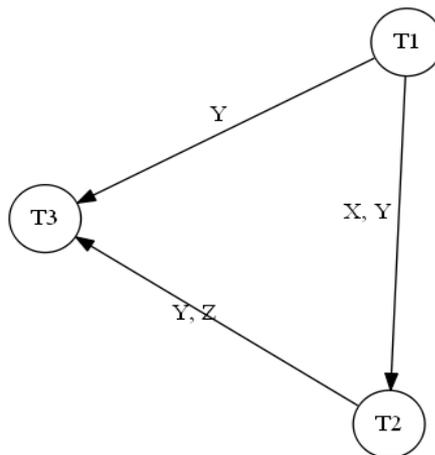
```
SELECT I.nomIndice, I.tipoIndice, count(AI.idAtributo)
FROM INDICES I natural join ATRIBUTOS_INDICES AI
GROUP BY I.idIndice, I.nomIndice, I.tipoIndice
HAVING count(*) > 5
```

(6) Nombre de las tablas y nombre de los esquemas al que pertenecen, tales que todos sus atributos pertenecen a algún índice definido sobre la tabla. (Se asume que los índices son correctos).

**SOLUCIÓN:**

```
SELECT T.nomTabla, E.nomEsquema
FROM (Esquemas E natural join Tablas T)
WHERE NOT EXISTS
  (SELECT *
   FROM Atributos A
   WHERE A.idTabla = T.idTabla
   and NOT EXISTS
     (SELECT *
      FROM Atributos_Indices AI
      WHERE AI.idAtributo = A.idAtributo
     )
  )
```

**Ejercicio 4 (25 puntos)**



Considere el siguiente grafo de seriabilidad:

- a) Escriba una historia no serial que cumpla con este grafo. Justifique su respuesta.
- b) Escriba las transacciones utilizadas en la historia anterior. Justifique su respuesta.
- c) Agregue los bloqueos de lectura y escritura en las transacciones, tal que en ninguna se respete 2PL básico. Justifique su respuesta.
- d) Si es posible, escriba una historia no serializable pero recuperable con las transacciones de la parte c). Justifique su respuesta.

### Solución:

a) Para que la historia sea no serial, debe ser entrelazada. A partir del grafo de seriabilidad, se definen las operaciones en conflicto según las aristas del mismo. Por ejemplo, para el caso de la arista que va de T1 a T3, deben existir dos operaciones en conflicto sobre la variable Y. La historia propuesta (una de todas las posibles) es la siguiente:

Ha: w1(X) w2(Z) r1(Y) w2(X) w2(Y) w3(Y) r3(Z) c1 c2 c3

b) Las transacciones se determinan con las operaciones contenidas en la historia. Como es una historia, las operaciones conservan el orden (definición de historia). Las transacciones son las siguientes:

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

c) Para que no cumpla 2PL básico, es necesario que no se respete la etapa de crecimiento y decrecimiento de los bloqueos.

T1: w1(X) w1(X) u1(X) r1(Y) r1(Y) u1(Y) c1 / Libera el w1(X) y luego bloquea con r1(Y)  
T2: w2(Z) w2(Z) u2(Z) w2(X) w2(X) u2(X) w2(Y) w2(Y) u2(Y) c2 / Libera el w2(Z) y luego bloquea con w2(X)  
T3: w3(Y) w3(Y) u3(Y) r3(Z) r3(Z) u3(Z) c3 / Libera el w3(Y) y luego bloquea con r3(Z)

d) Una historia es recuperable si ninguna transacción confirma hasta que confirmaron todas las transacciones desde las cuales se leyeron ítems.

La siguiente historia es recuperable y utiliza las transacciones de la parte c):

Hd: w1(X) w1(X) u1(X) w2(Z) w2(Z) u2(Z) w2(X) w2(X) u2(X) w2(Y) w2(Y) u2(Y) r1(Y) r1(Y) u1(Y) w3(Y) w3(Y) u3(Y) r3(Z) r3(Z) u3(Z) c2 c1 c3

En la historia se puede encontrar un conflicto de T1 a T2 por X y un conflicto de T2 a T1 por Y, por lo que el grafo de seriabilidad seguramente tiene un ciclo (construirlo todo para verificar) y la historia no es serializable.