

# Examen de Fundamentos de Bases de Datos

Julio 2015

## Indicaciones Generales:

- La duración del examen es de **tres (3)** horas.
- En la prueba **NO** se permite consultar material alguno.
- Empezar cada ejercicio en una hoja nueva.
- Escribir con lápiz y de un solo lado de las hojas.
- Numerar todas las hojas. Incluir en cada hoja la cédula y el nombre. En la primer hoja, incluir la cantidad de hojas que se entregan.

## Ejercicio 1. (30 puntos)

Se desea implementar el servicio **YouBook** que permite realizar compra y venta de libros en cualquier parte del mundo y recibirlos por correo.

- (a) De los libros, se conoce su ISBN, su título, su lista de autores, y el año de la edición.

Los usuarios se registran usando un email como identificación, y deben agregar su nombre, su dirección postal indicando país, provincia (o estado o departamento), ciudad, calle y número dentro de la ciudad, además de un conjunto de nros de teléfonos.

En el momento del registro, los usuarios indican un conjunto de libros de interés (al menos uno) pudiendo agregar más en cualquier momento. Cada vez que registran interés en un libro, se registra la fecha. Los usuarios pueden registrar su interés en un libro, para realizar una operación (compra, venta, etc) o simplemente para estar enterados de lo que sucede con ese libro.

- (b) En el sistema, se pueden realizar diferentes operaciones con los libros:

- Ofrecer a la venta un determinado libro.
- Solicitar un determinado libro.
- Reservar un libro ofrecido por otro usuario.
- Efectivizar una compra.

Es de particular interés para la empresa, hacer el seguimiento de las operaciones por lo que se registra la fecha y hora en que se realiza cada operación. Esta fecha y hora en combinación con otros datos (aún no discutidos) servirá para diferenciar las operaciones unas de otras.

- (c) Tanto las solicitudes como las ofertas, son realizadas por un usuario sobre un libro de su interés. Tanto las solicitudes como las ofertas están identificadas por el libro, el usuario y la fecha y hora en que se realizó. Sin embargo, para las ofertas el usuario debe registrar cuantas copias pone disponibles para vender, el precio al que aspira y los medios de pago que acepta.

- (d) Una reserva, es en algún sentido, la promesa que hace un usuario que solicitó un libro, de comprar una determinada oferta de ese libro.

Es así que cada reserva, está identificada por una solicitud y una oferta junto con la fecha y hora en que se realiza e indica la cantidad de copias que se reservan.

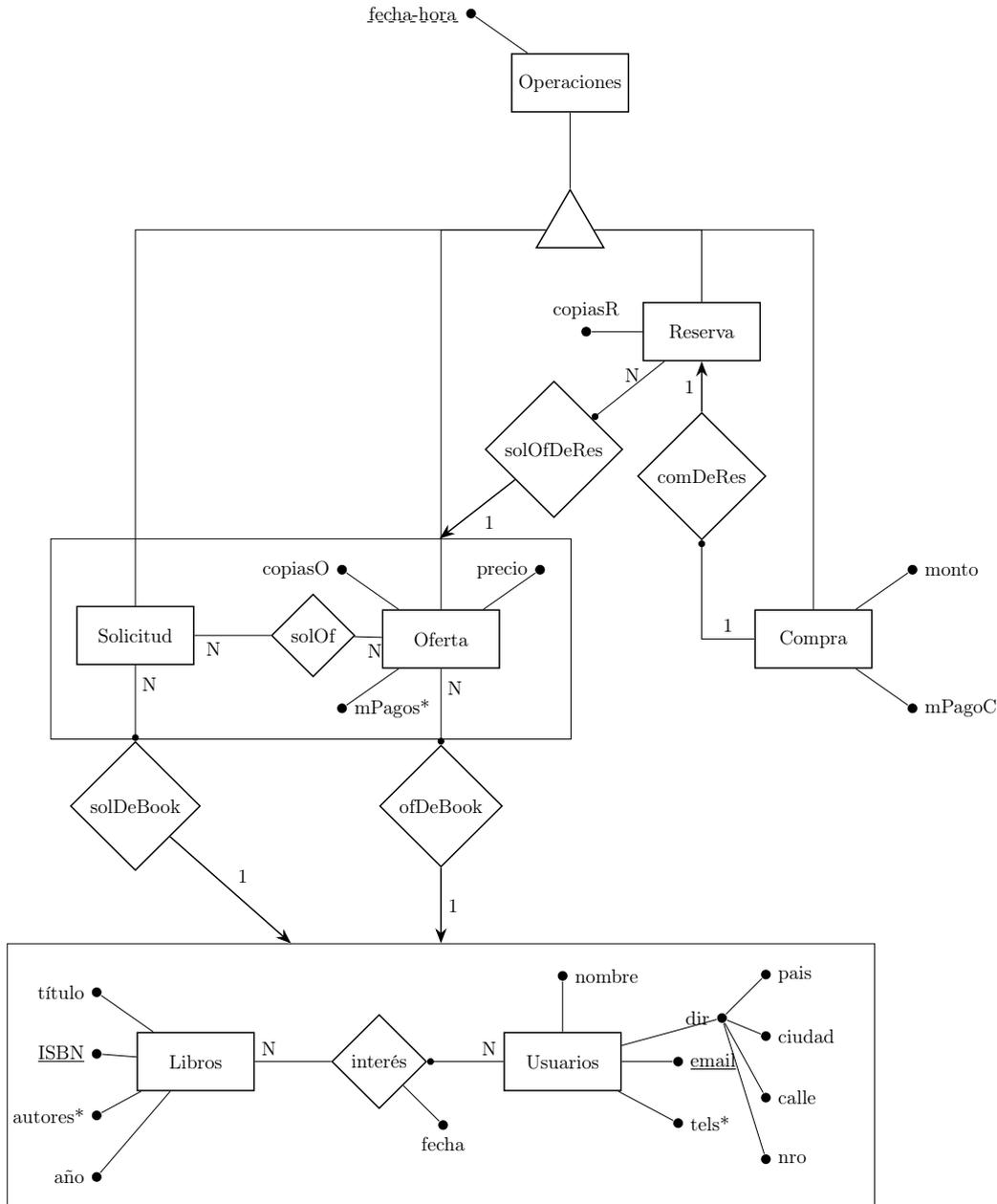
Esta cantidad siempre tiene que ser menor o igual que la ofertada.

La solicitud y oferta asociadas a una reserva deben ser sobre el mismo libro y distinto usuario.

- (e) Cada compra ejecuta una reserva, es decir, que concreta la compra que promete la reserva. Una compra está identificada por una reserva y la fecha y hora en que se realizó. Además se registra, el monto total de la compra y el medio de pago a utilizar. Este medio de pago debe ser uno de los admitidos por el oferente. A su vez, una reserva puede tener una única compra asociada.

**Se Pide:** Mer completo incluyendo restricciones no estructurales. La división en partes es solo una guía para la mejor resolución del ejercicio.

**Solución.**



## RNE's:

- El medio de pago de una compra debe ser uno de los aceptados por la oferta de la reserva correspondiente.

$$\forall c \in compra.$$

$$\forall r \in Reserva.$$

$$\forall s \in Solicitud.$$

$$\forall o \in Oferta.$$

$$(\langle c, r \rangle \in comDeRes \wedge \langle r, \langle s, o \rangle \rangle \in solOfDeRes \rightarrow c.mPagoC \in o.mPagos)$$

- La cantidad de copias de una reserva tiene que ser menor o igual a la cantidad de la oferta correspondiente.

$$\forall r \in Reserva. \forall s \in Solicitud. \forall o \in Oferta. (\langle r, \langle s, o \rangle \rangle \in solOfDeRes \rightarrow r.copiasR \leq o.copiasO)$$

- Una oferta asociada a una solicitud por la relación solOf, tiene que tener el mismo libro y distinto usuario.

$$\forall o \in Oferta.$$

$$\forall s \in Solicitud.$$

$$(\langle o, s \rangle \in solOf \rightarrow$$

$$\exists l_1 \in Libros. \exists u_1 \in Usuarios.$$

$$(\langle s, \langle l_1, u_1 \rangle \rangle \in solDeBook \wedge \exists l_2 \in Libros. \exists u_2 \in Usuarios.$$

$$\langle o, \langle l_2, u_2 \rangle \rangle \in ofDeBook \wedge l_1 = l_2 \wedge u_1 \neq u_2))$$

- Las categorías de operación son disjuntas dos a dos y la categorización de Operaciones es completa.

$$Solicitud \cap Oferta = \emptyset$$

$$Solicitud \cap Reserva = \emptyset$$

$$Solicitud \cap Compra = \emptyset$$

$$Oferta \cap Reserva = \emptyset$$

$$Oferta \cap Compra = \emptyset$$

$$Reserva \cap Compra = \emptyset$$

$$Solicitud \cup Oferta \cup Reserva \cup Compra = Operaciones$$

**Ejercicio 2.** (25 puntos)

Un supermercado mantiene parte de su información en el siguiente esquema relacional:

**Personal** (ci, nombre, fechaIngreso) En esta tabla se mantiene la información del personal del supermercado, cédula de identidad, nombre y su fecha de ingreso a la empresa.

**Seccion** (codSec, nombre, metros, ciEncargado) En esta tabla se mantiene la información de las diferentes secciones del supermercado. Para cada una de ellas se conoce su código, nombre, la superficie que ocupa (en metros cuadrados) y la cédula del encargado de la misma.

**Gondola** (codGondola, ciResp, valor) En esta tabla se mantiene la información de las góndolas del supermercado. Para cada una de ellas se conoce su código, cédula del empleado responsable

y un número que indica el valor de la misma. Por ejemplo la góndola ubicada a la entrada del supermercado tiene un valor alto ya que necesariamente todos los clientes pasan por ella.

**Articulos** (codArt, nombre, categoría, codGondola) En esta tabla se mantiene la información de los artículos que se encuentran a la venta en el supermercado. Para cada uno de ellos se conoce su código, nombre, categoría a la que pertenece (ej lácteo) y el código de la góndola donde se encuentra ubicado.

**AsigPersonal** (ci,codSec) En esta tabla se encuentran las diferentes asignaciones actuales del personal a las secciones del supermercado. Cada vez que hay un cambio de sección, se actualizan las tuplas correspondientes en esta tabla.

En estos esquemas se cumplen las siguientes dependencias de inclusión:  $\Pi_{ciEncargado}(Seccion) \subseteq \Pi_{ci}(Personal)$   
 $\Pi_{ciResp}(Gondola) \subseteq \Pi_{ci}(Personal)$   
 $\Pi_{codGondola}(Articulos) \subseteq \Pi_{codGondola}(Gondola)$   
 $\Pi_{ci}(AsigPersonal) \subseteq \Pi_{ci}(Personal)$   
 $\Pi_{codSec}(AsigPersonal) \subseteq \Pi_{codSec}(Seccion)$

**Se Pide:** Resolver las siguientes consultas en los lenguajes indicados.

(a) **En Álgebra Relacional:**

- I. Obtener los nombres de los empleados que ingresaron al supermercado después del 1/1/15 y han estado asignados a todas las secciones del mismo

**Solución.**

Esta es la solución:

$$\Pi_{nombre}(\sigma_{fechaIng > '1/1/15'}((\Pi_{ci,codSec}(AsigPersonal) \% \Pi_{codSec}(Secciones)) * Personas))$$

(b) **En Cálculo Relacional:**

- I. Obtener el nombre y superficie de las secciones donde solo han sido asignados (no importa el momento) empleados que son responsables de góndolas que tienen un valor menor a 3.

**Solución.**

$$\{\langle s.nombre, s.metros \rangle / Secciones(s) \wedge \\ \forall a.(AsigPersonal(a) \wedge a.codSec = s.codSec \\ \rightarrow \exists g.(Gondola(g) \wedge g.valor < 3 \wedge a.CI = g.ciResp))\}$$

(c) **En SQL (sin usar vistas ni subconsultas en el FROM):**

- I. Para cada categoría de artículos, listar la categoría y la cantidad de artículos que se encuentran en góndolas donde su responsable ingreso al supermercado antes del 1/1/10.

**Solución.**

```
Select categoria, count(codArt)
From Articulos
Where codGondola in (Select codGondola
From Gondola, Personal
where ciResp = ci and fechaIng < 1/1/10
)
group by categoria
```

- II. Obtener los nombres de las secciones que tienen a la mayor cantidad de empleados.

**Solución.**

```

Select codSec
From AsigPersonal A
Where (Select count(ci)
      From AsigPersonal A1
      Where A.codSec = A1.codSec ) >=ALL ( Select count(A2.ci)
      From AsigPersonal A2
      group by A2.codSec
    )

```

**Ejercicio 3.** (25 puntos)

- (a) Considere el esquema  $R(A, B, C, D, E)$  y las siguientes dependencias funcionales:

$$F = \{AB \rightarrow CE, D \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

1. Aplique el algoritmo para encontrar  $F_1$ . cubrimiento minimal de  $F$ . Explique la aplicación de cada paso que realiza.

**Solución.**

**1. Separar las dependencias de forma que todas tengan un sólo atributo a la derecha**

Dado que la única dependencia que tiene más de un atributo del lado derecho es la primera, es la única sobre la que hay que trabajar.

El resultado es:

$$F_1 = \{AB \rightarrow C, AB \rightarrow E, D \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

**2. Eliminar atributos redundantes del lado izquierdo** Para eso debemos trabajar con las dependencias que tienen más de un atributo del lado izquierdo. El trabajo a realizar es verificar si un atributo es redundante, testeado la clausura del lado izquierdo sin ese atributo, contra todas las dependencias actuales. Si se encuentra el lado derecho de la dependencia en esa clausura, entonces se puede eliminar el atributo que se descartó.

De esta forma, los únicos conjuntos de atributos que se deben considerar son:  $AB$ , y  $DC$ :

$$A^+ = \{A\} \text{ Por lo que B no se puede eliminar de } AB$$

$$B^+ = \{B\} \text{ Por lo que A no se puede eliminar de } AB$$

$$C^+ = \{C, D, E\} \text{ Por lo que D si se puede eliminar de } DC$$

De esta forma, el conjunto de dependencias queda:

$$F_2 = \{AB \rightarrow C, AB \rightarrow E, D \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

**3. Eliminar dependencias redundantes.** Para decidir qué dependencia es redundante, se deberían tomar todas las dependencias y verificar si se cumplen en el conjunto de dependencias, sin considerarla a ella misma. Sin embargo, este proceso se puede acelerar bastante dado que sólo pueden ser redundantes dependencias que alcancen al mismo atributo. De esta forma, las únicas dependencias que debemos considerar son  $AB \rightarrow E$  con  $D \rightarrow E$  y  $AB \rightarrow C$  con  $E \rightarrow C$ :

$$AB_{F_2 - \{AB \rightarrow E\}}^+ = \{A, B, C, E, D\}$$

De esta forma, el resultado parcial que nos queda es:

$$F_3 = \{AB \rightarrow C, D \rightarrow E, C \rightarrow D, E \rightarrow C\}$$

Ahora necesitamos controlar las otras dos dependencias, por lo que:

$$AB_{F_3 - \{AB \rightarrow C\}}^+ = \{A, B\} \qquad E_{F_3 - \{E \rightarrow C\}}^+ = \{E\}$$

Por lo que no se pueden eliminar ninguna de las dos dependencias y  $F_3$  es el minimal buscado.

- II. Deduzca 3 dependencias no triviales a partir de cubrimiento que encontré en la parte anterior usando al menos 3 reglas diferentes. Indique las reglas que utiliza.

**Solución.**

$$\begin{aligned} \{AB \rightarrow C, E \rightarrow C\} &\models ABE \rightarrow C \text{ Unión} \\ \{AB \rightarrow C, C \rightarrow E\} &\models AB \rightarrow E \text{ Transitiva} \\ \{AB \rightarrow C\} &\models ABD \rightarrow CD \text{ Aumentación} \end{aligned}$$

- III. Construya una descomposición en 3NF con join sin pérdida y sin pérdida de dependencias aplicando el algoritmo visto en el curso. Explique la aplicación de cada paso.

**Solución.**

**Construir una tabla por cada lado izquierdo, con esos atributos y los que aparecen del lado**

El resultado es:

$$\begin{aligned} R_1(ABC) \\ R_2(ED) \\ R_3(CD) \\ R_4(EC) \end{aligned}$$

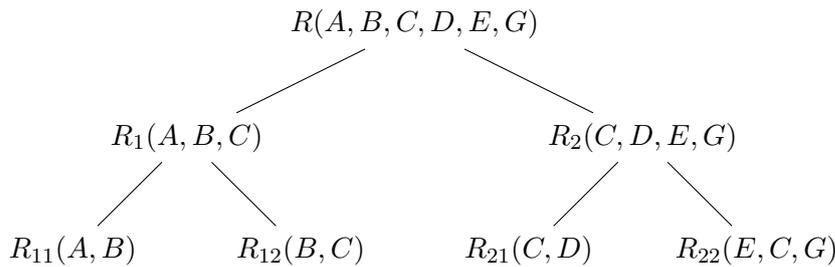
**Controlar si existe una clave de  $R$  incluida en alguna de las tablas.** Para ver esto, se puede testear si alguna de las tablas construidas es verificar si alguna de las tablas es una superclave de  $R$ :

$$ABC^+ = \{A, B, C, D, E\}$$

Por lo que no es necesario agregar ninguna tabla.

**Eliminar tablas que tengan todos los atributos en otra tabla** Es el caso no hay.

- (b) Considere el esquema  $R(A, B, C, D, E, G)$  y la siguiente secuencia de descomposiciones lograda por aplicación del algoritmo de BCNF.



- I. Escriba un conjunto con 3 dependencias funcionales que puede generar ese árbol de descomposición. Justifique su respuesta.

**Solución.**

Los algoritmos de BCNF y 4NF aplican el teorema de Join Sin Pérdida para dividir las tablas que no cumplen con la forma normal. De esta forma, los “hijos” de una tabla se construyen tomando los atributos de una dependencia y la tabla original sin los atributos del lado derecho de la dependencia. Esto hace que para dividir entre  $R_1(A, B, C)$  y  $R_2(C, D, E, G)$  se deba usar, por ejemplo la dependencia  $R_1 \cap R_2 \rightarrow R_1 - R_2$  o sea,  $C \rightarrow AB$  y entre  $R_{11}(A, B)$  y  $R_{12}(B, C)$ ,  $B \rightarrow A$ . Con el mismo criterio, entre  $R_{21}$  y  $R_{22}$ , la división se puede producir por  $C \rightarrow D$ .

El resultado final es:

$$\{C \rightarrow AB, B \rightarrow A, C \rightarrow D\}$$

- II. Escriba una dependencia multivaluada embebida con respecto a  $R$ , tal que el árbol de descomposición de la figura se debe extender para tenerla en cuenta al llevar  $R$  a 4NF. Asuma que las dependencias que eligió en la parte anterior son las únicas funcionales que se cumplen sobre  $R$ . Justifique su respuesta.

**Solución.**

La única tabla sobre la que se puede inducir una división es sobre  $R_{22}$ . Por lo que una multivaluada que lo puede dividir es la:  $G \twoheadrightarrow E$ .

Cuando se ve esta como una dependencia embebida sobre  $R$  es la siguiente:  $G \twoheadrightarrow E|C$

**Ejercicio 4.** (20 puntos)

Considere las siguientes transacciones:

$$T_1 : r_1(x), w_1(x), w_1(y), c_1$$

$$T_2 : r_2(z), w_2(z), c_2$$

- (a) Justifique por qué todas las historias de  $T_1$  y  $T_2$  son serializables.

**Solución.**

Las dos transacciones trabajan sobre items de datos diferentes, lo que hace que no haya conflictos, por lo que cualquier historia será serializable.

- (b) Escriba una transacción más  $T_3$  y una historia  $H_1$  involucrando a  $T_1$  y  $T_3$  tal que la  $H_1$  es **no serializable**. Justifique su respuesta.

**Solución.**

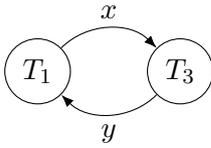
Sean  $T_3$  y  $H_1$  las siguientes:

$$T_3 : r_3(y), w_3(x), c_3$$

$$H_1 : r_1(x), r_3(y), w_1(x), w_3(x), w_1(y), c_3, c_1$$

En la historia se pueden detectar dos conflictos. El primero es sobre el item  $x$  y va de  $T_1$  a  $T_3$  y el otro es sobre el item  $y$  en el sentido contrario.

Por este motivo, el grafo de seriabilidad de  $H_1$  es el siguiente:



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

- (c) Escriba una transacción  $T_4$  idéntica a  $T_3$  pero que aborta en vez de confirmar. Escriba una historia  $H_2$  que involucre a  $T_1$  y  $T_4$  que genera abortos en cascada y sin embargo, es serializable. Justifique su respuesta.

**Solución.**

Sean  $T_4$  y  $H_2$  las siguientes:

$$T_4 : r_4(y), w_4(x), a_4$$

$$H_2 : r_4(y), w_4(x), r_1(x), w_1(x), w_1(y), a_3, \dots$$

La historia es serializable, dado que todos los conflictos van de  $T_4$  a  $T_1$ , sin embargo,  $T_1$  está obligada a abortar porque leyó de  $T_4$  que aborta.

