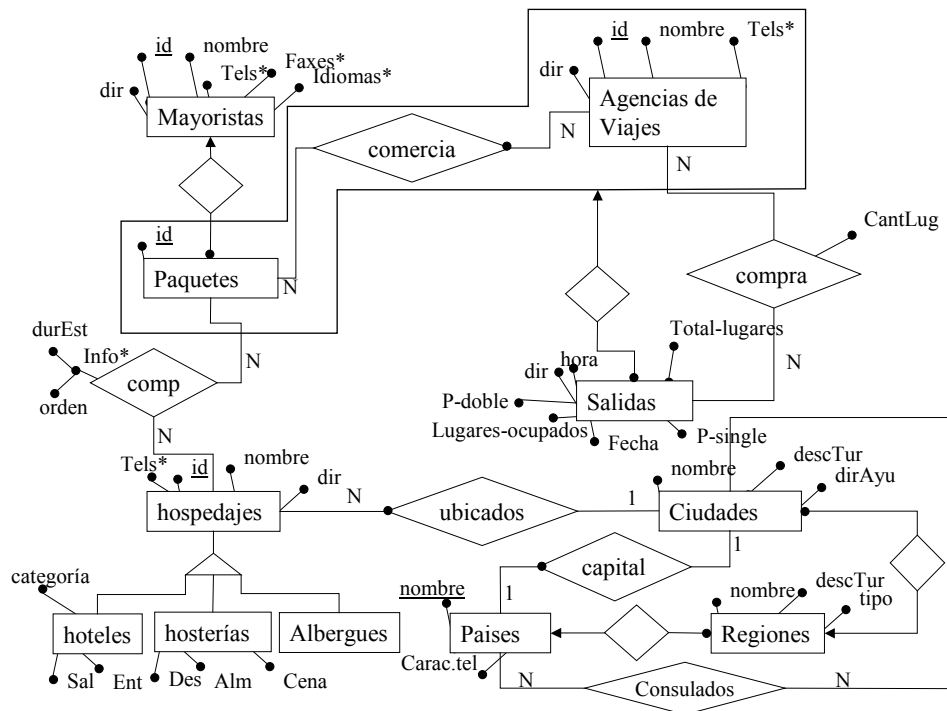


FUNDAMENTOS DE BASES DE DATOS

Examen Febrero 2003 - SOLUCION

Parte 1 – Modelo Entidad-Relación (25 puntos)

Ejercicio 1.



RNE:

1) Si $(C,P) \in \text{Capital}$ entonces C debe ser una ciudad de una región del país P. (Una forma de no usar esta restricción es colocar un atributo booleano en ciudades que indique si es capital o no).

2) Si $s \in \text{Salidas}$ entonces la suma de todas las cantLug de la misma salida s y S.Lugares-Libres debe ser menor o igual que S.Total-Lugares.

En lenguaje "lógico-matemático" podría ser algo así:

$\forall s \in \text{Salidas. } ((\text{lugares-ocupados} + \sum_{cl \in \{<a,s>/<a,s> \in \text{Compra}\}} \text{CantLug}(cl)) \leq \text{Total-Lugares}(s))$

3) Una ciudad no tiene consulado de su propio país.

4) Una agencia no se vende salidas a si misma.

Parte 2 – Diseño Relacional (25 puntos)

Ejercicio 2.

Una emergencia médico móvil administra en forma manual el despacho de ambulancias: Cada vez que suena el teléfono de emergencias, una recepcionista toma la llamada, elige una ambulancia a la que le pasa el llamado, y llena una tarjeta. En esta tarjeta se indica el día, la recepcionista, la ambulancia a la que se le comunicó el llamado y el socio que llamó¹. Además existen otros registros (también manuales) que indican la conformación de la ambulancia en cuanto al equipamiento y personal para cada día.

La emergencia móvil cuenta con un conjunto de ambulancias. De cada ambulancia se posee una ficha con la matrícula que la identifica y su capacidad. En la ficha de la ambulancia, también aparece la lista de los aparatos con que está equipada incluyendo, para cada aparato, un nombre, marca modelo y antigüedad. El nombre del aparato es identificatorio dentro de una ambulancia, pudiendo haber dos aparatos con el mismo nombre en dos ambulancias distintas

Actualmente, existen 3 registros independientes de médicos, recepcionistas y choferes. Es una política de la empresa que ningún funcionario tiene más de una función. Dicho de otra forma, nunca puede aparecer una misma persona en dos de esos registros. De cualquier miembro del personal se conoce su nombre, cédula de identidad que lo identifica, domicilio y teléfono. De los recepcionistas se conoce su horario de trabajo (hora de inicio y hora de fin). De los choferes se conoce su edad y la fecha en la cual sacaron la libreta de conducir. De los médicos se conoce su especialidad y una lista de las mutualistas en las que trabajan.

Todos los años se hacen asignaciones tanto de médicos como de choferes a las ambulancias. Para cada fecha del año y para cada ambulancia se asigna un chofer y un médico. Tanto un chofer como un médico puede estar asignado a lo sumo a una ambulancia en una fecha dada, y por otro lado puede estar asignado a una misma ambulancia en varias fechas distintas.

Se Pide:

- Diseño relacional hasta 3NF, justificando por que esta en esa forma normal. Incluir todas las dependencias funcionales y multivaluadas que se encuentren.
- A partir del diseño resultado de la parte A, obtener un diseño en 4NF con join sin pérdida.

SOLUCION

a)

Atributos del problema:

matricula
capacidad
nom-aparato
marca
modelo
antigüedad
ci-recep
nom-recep
dom-recep
tel-recep
horarioini
horariofin
ci-chofer
nom-chofer
dom-chofer
tel-chofer
edad

¹ Esto último fue agregado en el durante el examen.

fec-lib
ci-medico
nom-medico
dom-medico
tel-medico
especialidad
mutualista
fecha
socio

Dependencias funcionales:

matricula → capacidad
nom-aparato, matricula → marca, modelo, antigüedad
ci-recep → nom-recep, dom-recep, tel-recep, horarioini, horariofin
ci-chofer → nom-chofer, dom-chofer, tel-chofer, edad, fec-lib
ci-medico → nom-medico, dom-medico, tel-medico, especialidad
matricula, fecha → ci-chofer, ci-medico
ci-chofer, fecha → matricula, ci-medico
ci-medico, fecha → matricula, ci-chofer
socio, fecha → ci-recep, matricula

La clave de la relación universal es:

nom-aparato, fecha, mutualista, socio

Obtengo conjunto de dfs minimal:

1er. paso:

matricula → capacidad
nom-aparato, matricula → marca
nom-aparato, matricula → modelo
nom-aparato, matricula → antigüedad
ci-recep → nom-recep
ci-recep → dom-recep
ci-recep → tel-recep
ci-recep → horarioini
ci-recep → horariofin
ci-chofer → nom-chofer
ci-chofer → dom-chofer
ci-chofer → tel-chofer
ci-chofer → edad
ci-chofer → fec-lib
ci-medico → nom-medico
ci-medico → dom-medico
ci-medico → tel-medico
ci-medico → especialidad
matricula, fecha → ci-chofer
matricula, fecha → ci-medico
ci-chofer, fecha → matricula
ci-chofer, fecha → ci-medico
ci-medico, fecha → matricula
ci-medico, fecha → ci-chofer
socio, fecha → ci-recep
socio, fecha → matricula

2º. paso:

Observando la realidad, deduzco que no hay atributos redundantes en ninguna dependencia funcional.

3er. paso:

matricula, fecha → *ci-chofer* es redundante, ya que:

$(matricula, fecha)^+_{F-\{matricula, fecha \rightarrow ci-chofer\}} = \{ci-medico, ci-chofer\}$

entonces $F-\{matricula, fecha \rightarrow ci-chofer\} \equiv F$

De igual forma vemos que: *ci-chofer, fecha* → *matricula* es redundante

Conjunto minimal:

matricula → *capacidad*
nom-aparato, matricula → *marca*
nom-aparato, matricula → *modelo*
nom-aparato, matricula → *antigüedad*
ci-recep → *nom-recep*
ci-recep → *dom-recep*
ci-recep → *tel-recep*
ci-recep → *horarioini*
ci-recep → *horariofin*
ci-chofer → *nom-chofer*
ci-chofer → *dom-chofer*
ci-chofer → *tel-chofer*
ci-chofer → *edad*
ci-chofer → *fec-lib*
ci-medico → *nom-medico*
ci-medico → *dom-medico*
ci-medico → *tel-medico*
ci-medico → *especialidad*
matricula, fecha → *ci-medico*
ci-chofer, fecha → *ci-medico*
ci-medico, fecha → *matricula*
ci-medico, fecha → *ci-chofer*
socio, fecha → *ci-recep*
socio, fecha → *matricula*

Aplico algoritmo de 3NF con join sin pérdida y preservación de dependencias:

- 1) Utilizo conjunto de dfs minimal.
- 2) Una relación para cada conjunto de dfs con igual lado izquierdo.

Ambulancia (matricula , capacidad)
Aparato (nom-aparato, matricula, marca, modelo, antigüedad)
Recepcionista (ci-recep, nom-recep, dom-recep, tel-recep, horarioini, horariofin)
Chofer (ci-chofer, nom-chofer, dom-chofer, tel-chofer, edad, fec-lib)
Médico (ci-medico, nom-medico, dom-medico, tel-medico, especialidad)
Asignación (ci-medico, fecha, matricula, chofer)
Llamado1 (socio, fecha, ci-recep, matricula)

- 2) Construyo una relación que contenga una clave.

Llamado2 (nom-aparato, fecha, mutualista, socio)

Dependencias Multivaluadas:

socio, fecha →→ mutualista
socio, fecha →→ nom-aparato (el complemento)

b)

Aplico algoritmo de 4NF con join sin pérdida:

Todas las relaciones excepto Llamado2 están en 4NF.
Aplico 1er. paso del algoritmo a Llamado2:

Llamado21 (socio, fecha, mutualista)
Llamado22 (socio, fecha, nom-aparato)

Las dos tienen dependencias triviales. Están en 4NF.

Esquema final:

Ambulancia (matricula, capacidad)
Aparato (nom-aparato, matricula, marca, modelo, antigüedad)
Recepcionista (ci-recep, nom-recep, dom-recep, tel-recep, horarioini, horariofin)
Chofer (ci-chofer, nom-chofer, dom-chofer, tel-chofer, edad, fec-lib)
Médico (ci-medico, nom-medico, dom-medico, tel-medico, especialidad)
Asignación (ci-medico, fecha, matricula, chofer)
Llamado1 (socio, fecha, ci-recep, matricula)
Llamado21 (socio, fecha, mutualista)
Llamado22 (socio, fecha, nom-aparato)

Parte 3 – Consultas (25 puntos)

Ejercicio 4.

SE PIDE :

1) Escribir en álgebra relacional las siguientes consultas:

a) Devolver las CI de los televidentes que miran todas las emisiones de "Gran Hermano".

Solucion consulta a

$A = \pi_{\text{NomProg, Dia, HoraComienzo}}(\sigma_{\text{NomProg}="Gran Hermano"}(\text{PROGRAMACION}))$

$B = \pi_{\text{CiTele, NomProg, Dia, HoraComienzo}}(\text{PREFERENCIAS})$

Res = B%A

b) Devolver el nombre de los programas que son emitidos Lunes y Viernes y no son vistos por nadie el día Lunes o no son vistos por nadie el día Viernes

Solución consulta b

$A1 = \pi_{\text{nomProg}}(\sigma_{\text{Dia}="Lunes"}(\text{PROGRAMACION}))$

$A2 = \pi_{\text{nomProg}}(\sigma_{\text{Dia}="Viernes"}(\text{PROGRAMACION}))$

$A = A1 \cap A2$ - Programas emitidos los lunes y los viernes

$B = \pi_{\text{nomProg}}(\sigma_{\text{Dia}="Lunes"}(\text{PREFERENCIAS}))$

$C = \pi_{\text{nomProg}}(\sigma_{\text{Dia}="Viernes"}(\text{PREFERENCIAS}))$

Res = (A-B) \cup (A-C) - No vistos por nadie el lunes o no vistos por nadie el viernes o
- bien ninguno de los dos días

2) Escribir en cálculo relacional las siguientes consultas:

- c) Devolver la CI y el nombre de los empleados que trabajan con un género de programa y que durante su horario de trabajo NO se emiten programas de dicho género.

Solución consulta c

{t.nomEmp, t.ciEmp / Empleados (t) ^
⌋ (∃ p) (Programacion(p) ^ p.horaComienzo >= t.horaEntrada ^ p.horaFin <= t.horaSalida ^
(∃ pr) (Programas (pr) ^ pr.NombreProg = p.NombreProg ^ pr.genero = t.genero))}

Otra:

{t.nomEmp, t.ciEmp / Empleados(t) ^ (∀v) (Programación(v) ^ v.horaComienzo >= t.horaEntrada ^
v.horaFin <= t.horaSalida → (∃w) (Programas(w) ^ v.NombreProg = w.NombreProg ^ w.genero <>
t.genero))}

- d) Devolver los nombres de los televidentes que los jueves sólo miran programas de género "película".

Solución consulta d

{t.nomTele / Televidentes (t) ^ (∃p) (Preferencias (p) ^ t.ciTele = p.ciTele ^ p.Dia = "Jueves" ^ (∃pr)
(Programa(pr) ^ pr.nomProg=p.nomProg ^ pr.Genero ="película" ^ ⌋ (∃ p1) (Preferencias (p1) ^ t.ciTele =
p1.ciTele ^ p1.Dia ="Jueves" ^ (∃pr) (Programa(pr1) ^ pr1.nomProg=p1.nomProg ^ pr1.Genero
<>"película"))))}

{t.nomTele / Televidentes(t) ^ (∀u) (Preferencias(u) ^ t.ciTele = u.ciTele ^ u.dia = 'jueves' → (∃w)
(Programas(w) ^ w.nomProg = u.nomProg ^ w.genero = 'película')) ^
(∃x) (Preferencias(x) ^ x.ciTele = t.ciTele ^ x.dia = 'jueves') }

3) Escribir en SQL las siguientes consultas, teniendo en cuenta que si es necesario se pueden utilizar vistas o tablas temporales.

- e) Devolver cantidad de televidentes, por ocupación, que miran "TeleInforma" los Jueves.

Solución consulta e

```
SELECT T.ocupacion, count (*) as cantidad
FROM televidentes T, preferencias P
WHERE      P.NomProg = "TeleInforma" and
           P.Dia = "jueves" and
           P.CiTele = T.CiTele
GROUP BY T.ocupacion
```

- f) Devolver los nombres de los programas más vistos el día Lunes.

Solución consulta f

```
CREATE VIEW Espectadores AS
SELECT NomProg, Count (*) as cantEspectadores
FROM preferencias
WHERE Dia = "Lunes"
GROUP BY NomProg
```

```
SELECT NomProg
FROM Espectadores
WHERE cantEspectadores =
      (SELECT Max (cantEspectadores)
FROM Espectadores)
```

Parte 4 – Optimización y Concurrencia (25 puntos)

Ejercicio 5.

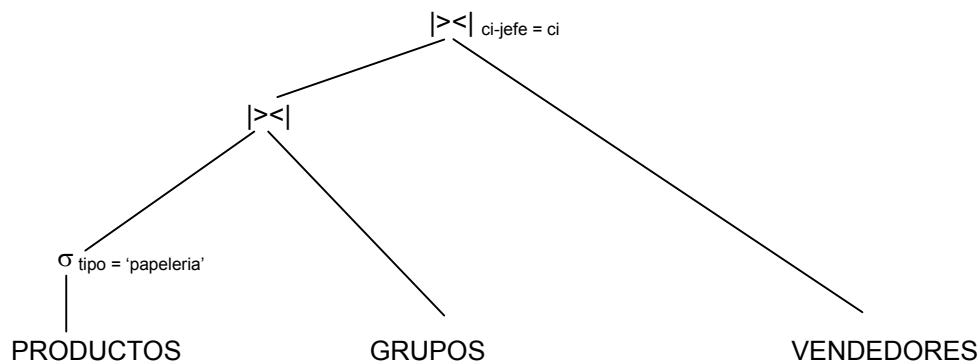
Dado el siguiente esquema relacional:

Productos (tipo, nroprod, nrogrupo)

Grupos (nrogrupo, ci-jefe)

Vendedores (ci, nrogrupo, salario)

y el siguiente plan lógico correspondiente a una consulta:



Y considerando además que:

El resultado de la selección se debe guardar en una tabla temporal Temp1 y el resultado del primer join se debe guardar en una tabla temporal Temp2.

- 1) Dar el plan físico que le parezca mejor para ese plan lógico.
- 2) Calcular los tamaños (cantidad de tuplas) de Temp1 y Temp2.
- 3) Calcular el costo total del plan. (Nota: No olvidar sumar los costos de grabar Temp1 y Temp2)

DATOS:

	PRODUCTOS	GRUPOS	VENDEDORES	TEMP2
Cantidad tuplas	2000	50	10000	
Indices primarios			ci (niveles: 2)	
Indices secundarios (B+)	tipo (niveles: 2)		salario (niveles: 3)	
Cantidad tuplas por bloque	20	10	5	5
Observaciones	- Tipos distintos: 200			

Fórmulas para cálculo de costo y de tamaños:

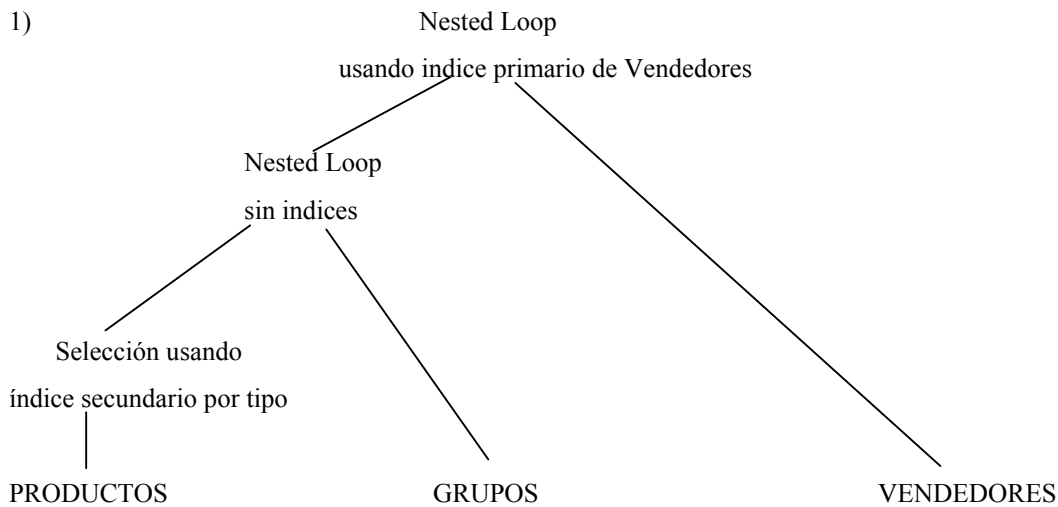
Selección	Búsqueda lineal.	B
	Con índice primario.	$x + 1$
	Con índice secundario.	$x + s$
Join (R,S)	Nested Loop (ciclo anidado) sin utilizar índices	$b_R + b_R * b_S$

	Nested Loop (ciclo anidado) utilizando índice primario para recuperar tuplas que matchean	$b_R + R * (x + 1)$
Selectividad	Selección (atrib. A)	$1 / V(A,R)$
	Join (atrib. A)	$1 / \text{Max}(V(A,R), V(A,S))$

Notación: **b** – cantidad de bloques
x – cantidad de niveles del índice
|R| - cantidad de tuplas de R
s – cardinalidad de la selección
V(A,R) - cantidad de valores distintos del atributo A en R

SOLUCION

1)



2) $Tam(Temp1) = \text{cantidad de tuplas} * \text{selectividad} = |Productos| * sel = 2000 / 200 = 10$
 $Tam(Temp2) = |Temp1| * |Grupos| * js = 10 * 50 / 50 = 10$

$Tam(Temp2)$ también podemos deducirlo pensando de la siguiente forma. $Temp1 \bowtie Grupos$ se hace por el atributo $nrogrupo$, el cual es clave en Grupos. Por lo tanto cada tupla de $Temp1$ va a juntarse con una y solo una tupla de Grupos. Entonces el tamaño resultante va a ser 10.

Nota: sel = selectividad de la selección
 js = selectividad del join

3) Costo $(\sigma_{\text{tipo} = 'papeleria'}) = x + s + \text{costo de grabar } Temp1 = 2 + 10 + b_{Temp1} = 12 + 1 = 13$
 Aclaración: La cantidad de tuplas por bloque (factor de bloqueo) para $Temp1$ es igual a la de Productos.

Costo $(Temp1 \bowtie Grupos) = b_{Temp1} + b_{Temp1} * b_{Grupos} + \text{costo de grabar } Temp2 = 1 + 1 * 5 + b_{Temp2} = 6 + 2 = 8$
 Costo $(Temp2 \bowtie Vendedores) = b_{Temp2} + |Temp2| * (x + 1) = 2 + 10 * 3 = 32$

Costo total = 13 + 8 + 32 = 53

Ejercicio 6.

1. Enumere y explique las 4 propiedades más importantes de las transacciones en los sistemas de bases de datos.

Propiedades de: Atomicidad, Aislamiento, Consistencia y Durabilidad.
 Ver Capítulo 17 Sección 17.3 del Elsmari y Navathe Segunda Edición.

2. Indique qué problemas tienen los siguientes planes.

P1)	T1	T2

	t1 read(a)	
	t2	read(a)
	t3 write(a)	
	t4	write(a)

Problema de la actualización perdida.

P2)

Valores Iniciales: C1 = 40, C2 = 50, C3 = 30.

	T1	T2

	t1 read(C1)	
	t2 suma := suma + C1	
	t3 read(C2)	
	t4 suma := suma + C2	
	t5	read(C3)
	t6	C3 := C3 - 10
	t7	write(C3)
	t8	read(C1)
	t9	C1 := C1 + 10
	t10	write(C1)
	t11 read(C3)	
	t12 suma := suma + C3	

Problema de actualización inconsistente.

Explique el protocolo que considere mejor para evitar estos problemas. **2PL**

Muestre planes nuevos para P1 y P2 usando el protocolo propuesto. Indique si aparecen nuevos problemas. (Posible aparición de bloqueo mortal)

3. Sea el siguiente plan de las transacciones T1, T2 y T3:

	T1	T2	T3

	t1 V1 := Read(a)		

t2	V1 := V1 + 1		
t3		V2:= Read(a)	
t4		V2 := V2 x 2	
t5			V3:= Read(a)
t6			V3 := V1 + 1
t7			Write(a,1)
t8		Write(a,V2)	
t9	Write(a,V1)		

Notación: Write(i,v) significa escribir en el ítem i el valor v

Indique si este plan producirá en cualquier caso (independiente de los valores de a) un valor correcto. **NO**

¿Cómo sabe la respuesta sin ejecutar el plan? **Haciendo el grafo de precedencia y detectando que existen ciclos.**

4. Indique qué consecuencias tienen en la recuperación los siguientes actos:

a) Nunca escribe físicamente los buffers en la base de datos antes del commit

La acción física de deshacer nunca es necesaria y por lo tanto tampoco son necesarios los registros de bitácora para deshacer.

b) Fuerza los buffers hacia la base de datos en el momento del commit.

Nunca es necesario rehacer después de una falla del sistema.