

FUNDAMENTOS DE BASES DE DATOS
Examen Julio 2019
SOLUCION

La duración del examen es de 3 horas y media.
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 (26 puntos).

La corte electoral desea analizar información relacionada con la organización de las elecciones internas.

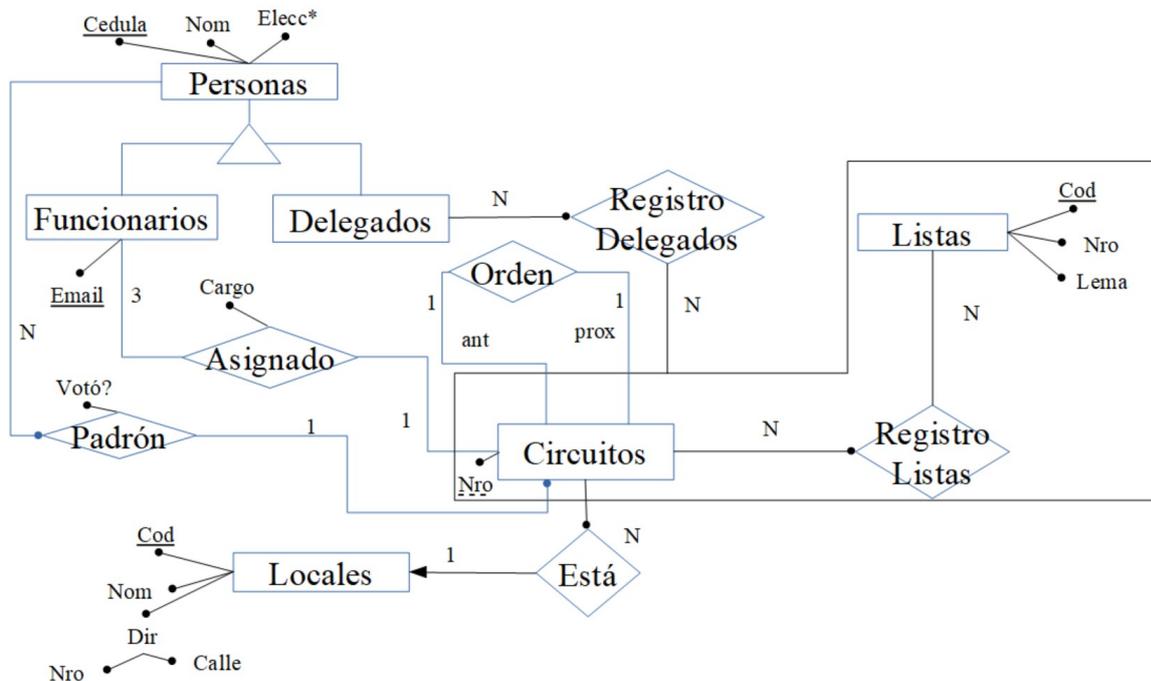
De las personas que participan en la elección, se conoce su cédula de identidad, su nombre y las fechas de las elecciones en las que votó. Las personas pueden ser funcionarios, delegados o votantes. Se debe tener en cuenta que todas las personas son votantes. De los funcionarios se conoce su e-mail y el cargo que tienen en el circuito (presidente, secretario o vocal).

Cada local de votación tiene un código que los identifica, un nombre y la dirección (calle y número). En cada local hay un conjunto de circuitos, identificados con un número dentro de ese local. Cada circuito tiene asignados exactamente 3 funcionarios: un presidente, un secretario y un vocal. Un funcionario sólo puede estar asignado a un circuito.

Las listas tienen un código identificador, un número y un lema. Las listas son registradas en cada circuito. Los delegados se registran en algunos circuitos para una o más listas que deben ser del mismo lema. Una lista puede estar registrada en un circuito sin que ningún delegado se haya registrado para ella, sin embargo todos los delegados tienen que estar registrados por alguna lista.

Los circuitos, dentro de un mismo local, son recorridos por quien los fiscaliza en un orden. Interesa saber qué circuito está inmediatamente antes que otro según dicho orden. En cada circuito se conoce el padrón, es decir, la lista de votantes habilitados en ese circuito. Se debe llevar el registro de qué personas votaron en cada circuito.

Se pide: Modelo Entidad-Relación completo del problema.



Una persona no puede ser delegado y funcionario

$$\text{Funcionarios} \cap \text{Delegados} = \emptyset$$

Todo circuito debe tener exactamente 3 funcionarios: un presidente, un secretario y un vocal.

$$\forall c \in \text{Circuitos} \forall d \in \text{Funcionarios} (\text{cargo}(\langle c, d \rangle) \in \{ 'presidente', 'secretario', 'vocal' \})$$

$$\forall c \in \text{Circuitos} (\# \{ d | \langle c, d \rangle \} = 3)$$

$$\forall d_1 \in \text{Funcionarios} \forall d_2 \in \text{Funcionario} (d_1 \neq d_2 \wedge \langle c, d_1 \rangle \wedge \langle c, d_2 \rangle \rightarrow \text{cargo}(\langle c, d_1 \rangle) \neq \text{cargo}(\langle c, d_2 \rangle))$$

Equivalente a las tres anteriores:

$$\forall c \in \text{Circuitos} \exists d_1 \in \text{Funcionarios} \exists d_2 \in \text{Funcionarios} \exists d_3 \in \text{Funcionarios} (\langle c, d_1 \rangle \wedge \langle c, d_2 \rangle \wedge \langle c, d_3 \rangle \wedge \text{cargo}(\langle c, d_1 \rangle) = 'presidente' \wedge \text{cargo}(\langle c, d_2 \rangle) = 'secretario' \wedge \text{cargo}(\langle c, d_3 \rangle) = 'vocal')$$

Todas las listas registradas por un delegado son del mismo lema

$$\forall d \in \text{Delegados} \forall l_1 \in \text{Listas} \forall l_2 \in \text{Listas} ((\exists c \text{ Circuito}(c) \wedge \langle d, \langle l_1, c \rangle \rangle \in \text{RegistroDelegados}) \wedge (\exists c \text{ Circuito}(c) \wedge \langle d, \langle l_2, c \rangle \rangle \in \text{RegistroDelegados}) \rightarrow \text{lema}(l_1) = \text{lema}(l_2))$$

Ejercicio 2 (26 puntos).

Considere una tabla $R(A,B,C,D,E)$. En esta tabla se deberían cumplir las siguientes dependencias funcionales, pero no hay garantías de que la instancia actual de la base las cumplan:

$$AB \rightarrow C$$
$$B \rightarrow E$$

Sí hay garantías de que ningún atributo de ninguna tupla tiene valores nulos.

- Escriba una consulta SQL que devuelva los valores de B que violan la segunda dependencia. Justifique.
- Escriba una consulta SQL que devuelva los valores de los atributos de alguna de las claves tal que esa clave se viola para esos valores. Justifique.
- Escriba una descomposición de la tabla R con dos tablas de 3 atributos cada una, tal que no tenga join sin pérdida. Justifique.
- Escriba una consulta SQL que devuelva las tuplas espurias para la descomposición mencionada en c).
- Determine en qué forma normal se encuentra R. Justifique.
- Encuentre una descomposición de R en BCNF siguiendo el algoritmo visto en el curso.

Solución:

a) Se deben encontrar los valores de B que tienen asociados distintos valores de E. Obviamente, esto debe ser en diferentes tuplas, por lo que se va a necesitar hacer referencia a dos tuplas de R simultáneamente. La consulta debe ser equivalente a:

```
select B
from R R1
where exists (
    select 1
    from R R2
    where R1.B = R2.B and R1.E != R2.E )
```

b) Primero hay que determinar las claves de R.

Los atributos $\{A,B,D\}$ nunca están a la derecha de las dependencias, por lo que deberán estar en todas las claves. Dado que $ABD \rightarrow \{ABCDE\}$, se cumple que ABD es la única clave.

Se deben recuperar tuplas que tengan el mismo valor en ABD y distinto valor en alguno de los atributos restantes.

La consulta entonces, es la siguiente:

```
Select A,B,D
from R R1
where exists (
    select 1
    from R R2
    where R1.A=R2.A and R1.B=R2.B and R1.D=R2.D and (R1.C != R2.C or R1.E!=R2.E)
)
```

c) Considere el siguiente esquema relacional con dos tablas de tres atributos cada una: $R_1(A,B,C)$, $R_2(D,B,E)$.

En esta descomposición de R, se cumple que $R_1 \cap R_2 = \{B\}$ pero B no determina ni BE ni AC por lo que la descomposición presenta join con pérdida (no tiene join sin pérdida).

d) Una consulta posible es la siguiente:

```
Select *
from R1, R2
where R1.B=R2.B and not exists (
    select 1
    from R
    where R.A=R1.A and R.B=R2.B and R.D=R1.D and R.C=R1.C and R.E=R1.E
)
```

e) Dado que la única clave es ABD, cualquiera de las dos dependencias existentes van de una parte de la clave en atributos no primos por lo que son dependencias parciales. Por esto, la tabla está en 1NF con respecto a las dependencias dadas.

f) La primer dependencia viola BCNF por lo que dividimos R en R₁(A,B,C),R₂(A,B,D,E)

Las proyecciones del conjunto de dependencias son las siguientes:

$$\Pi_{R_1}(F) = \{ AB \rightarrow C \}$$
$$\Pi_{R_2}(F) = \{ B \rightarrow E \}$$

R₁ está en BCNF porque tiene una clave del lado izquierdo pero R₂ tiene una dependencia por la que se viola BCNF. Por eso, se divide R₂ en: R₂₁(B,E),R₂₂(ABD).

La descomposición final, en BCNF, es: R₁(A,B,C), R₂₁(B,E), R₂₂(ABD).

Ejercicio 3 (26 puntos)

Una empresa dedicada a la realización de mudanzas maneja su información en una base de datos con el siguiente esquema relacional:

MUDANZAS (codMud, ciCli, direcciónOrigen, direcciónDestino, nroCajas, nroMuebles, fechaInicio)

Para cada una de las mudanzas que realiza se conoce un código que la identifica, la cédula del cliente que contrata el servicio, las direcciones de origen y de destino, la cantidad de cajas y muebles a mover, y la fecha de inicio de la mudanza. Las mudanzas donde coinciden la dirección de origen y de destino se denominan mudanzas internas.

EMPLEADOS (ci, nombre, fechaIngreso, especialidad)

De cada empleado de la empresa se conoce su cédula que lo identifica, su nombre, la fecha de ingreso y su especialidad.

CAMIONES (mat, marca, año, ciChofer)

La empresa no posee vehículos propios. En caso de necesitarlos los contrata por fuera. Cada vehículo contratado tiene un chofer propio que no es empleado de la empresa. En esta tabla se mantiene información de los camiones con que trabaja la empresa. Para cada uno de ellos se conoce su matrícula, marca y año y la cédula del chofer.

ASIGNACIONES (ci, codMud)

En esta tabla se mantiene la asignación de empleados a las mudanzas.

CONTRATACIONES (codMud, mat)

En esta tabla se mantiene la asignación de camiones contratados a las mudanzas.

En estos esquemas se cumplen las siguientes dependencias de inclusión:

$$\begin{aligned} \Pi_{ci}(ASIGNACION) &\subseteq \Pi_{ci}(EMPLEADOS) \\ \Pi_{codMud}(ASIGNACION) &\subseteq \Pi_{codMud}(MUDANZAS) \\ \Pi_{codMud}(CONTRATACIONES) &\subseteq \Pi_{codMud}(MUDANZAS) \\ \Pi_{mat}(CONTRATACIONES) &\subseteq \Pi_{mat}(CAMIONES) \end{aligned}$$

a) Resolver las siguientes consultas en Álgebra Relacional:

1. Cédula y nombre de los empleados que fueron asignados a las mudanzas en que se trasladó el mayor número de cajas.

A contiene los números de cajas:

$$A = \Pi_{nroCajas}(MUDANZAS)$$

B contiene el máximo número de cajas:

$$B = (A \bowtie_{nroCajas >= nroMenor} \rho_{nroCajas \rightarrow nroMenor}(A)) \% A$$

SOL contiene la Cédula y nombre de los empleados que fueron asignados a mudanzas donde se trasladó la cantidad de cajas contenida en B:

$$SOL = \Pi_{ci,nombre}(EMPLEADOS * ASIGNACIONES * (B * MUDANZAS))$$

2. Cédulas de identidad de los choferes que fueron contratados en todas las mudanzas donde se asignaron empleados de especialidad ARMADOR.

A contiene las mudanzas donde se asignaron ARMADORES:

$$A = \Pi_{codMud}(ASIGNACIONES * \sigma_{especialidad = \text{“ARMADOR”}}(EMPLEADOS))$$

B contiene parejas (CHOFER, MUDANZA), tal que el chofer fue contratado para esa mudanza:

$$B = \Pi_{ciChofer, codMud}(CONTRATACIONES * CAMIONES)$$

SOL es el cociente entre B y A:

$$SOL = B \% A$$

b) Resolver las siguientes consultas en Cálculo Relacional:

1. Nombre de los empleados que sólo fueron asignados a mudanzas internas.

$$\begin{aligned} &\{e.nombre/EMPLEADOS(e) \wedge (\exists a)(ASIGNACIONES(a) \wedge \\ &(\exists l)(MUDANZAS(l) \wedge a.codMud = l.codMud \wedge a.ci = e.ci \wedge l.direccionOrigen = l.direccionDestino)) \wedge \\ &\neg(\exists a1)(ASIGNACIONES(a1) \wedge (\exists l1)(MUDANZAS(l1) \wedge a1.ci = e.ci \wedge a1.codMud = l1.codMud \wedge \\ &l1.direccionOrigen \neq l1.direccionDestino))\} \end{aligned}$$

2. Cédula de los choferes que manejan un único camión, y éste fue contratado para trabajar en al menos dos mudanzas en 2018. NOTA: asuma que tiene una función AÑO, que dada una fecha devuelve el año.

$$\begin{aligned} &\{c.ciChofer/CAMIONES(c) \wedge \neg(\exists c1)(CAMIONES(c1) \wedge c1.ciChofer = c.ciChofer \wedge c1.mat \neq c.mat) \wedge \\ &(\exists m1)(MUDANZAS(m1) \wedge AÑO(m1.fechaInicio) = 2018) \wedge \\ &(\exists co1)(CONTRATACIONES(co1) \wedge co1.codMud = m1.codMud \wedge co1.mat = c.mat) \wedge \\ &(\exists m2)(MUDANZAS(m2) \wedge AÑO(m2.fechaInicio) = 2018 \wedge co2.codMud \neq co1.codMud) \wedge \\ &(\exists co2)(CONTRATACIONES(co2) \wedge (co2.codMud = m2.codMud \wedge co2.mat = c.mat))\} \end{aligned}$$

c) Resolver la siguiente consulta en SQL (sin usar vistas ni subconsultas en el FROM):

1. Fecha de inicio y dirección de origen de las mudanzas para las cuales se contrató un único camión pero se asignaron más de 10 empleados.

```
SELECT L.fechaInicio, L.direccionOrigen
FROM MUDANZAS L
WHERE L.codMud IN
    (SELECT C.codMud
     FROM CONTRATACIONES C
     GROUP BY C.codMud
     HAVING count(*) >= 10
     INTERSECT
     SELECT A.codMud
     FROM ASIGNACIONES A
     GROUP BY A.codMud
     HAVING count(*) = 1
```

)

Ejercicio 4 (22 puntos).

Dada las siguientes transacciones:

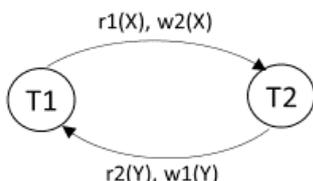
T1: r1(X), w1(X), w1(Y), r1(Y), w1(X), c1

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

a) Indicar si las siguientes historias son serializables, recuperables, seriales y si evitan abortos en cascada. En cada caso justifique su respuesta. En caso de ser serializable, muestre una historia serial equivalente.

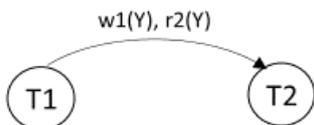
- i) H1: r1(X), w1(X), r2(Y), w1(Y), w2(Y), r1(Y), r2(X), w2(X), c2, w1(X), c1
- ii) H2: r1(X), w1(X), w1(Y), r1(Y), r2(Y), w2(Y), w1(X), c1, r2(X), w2(X), c2
- iii) H3: r1(X), w1(X), r2(Y), w2(Y), w1(Y), r1(Y), w1(X), c1, r2(X), w2(X), c2

i)



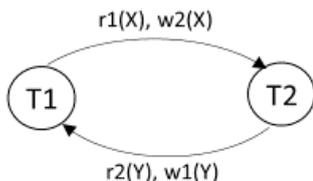
- El grafo de seriabilidad tiene ciclos, por tanto H1 no es serializable.
- No es recuperable porque T2 lee de T1 (lee X, dato escrito por T1 previamente) y T2 confirma (hace commit) antes que T1.
- No es serial porque tiene operaciones entrelazadas.
- No evita abortos en cascada porque T2 lee de transacciones no confirmadas.

ii)



- El grafo de seriabilidad no tiene ciclos, por tanto H2 es serializable.
Su historia serial equivalente es:
H2': r1(X), w1(X), w1(Y), r1(Y), w1(X), c1, r2(Y), w2(Y), r2(X), w2(X), c2
- Es recuperable porque T2 lee de T1 (lee Y, dato escrito por T1 previamente) y T1 confirma antes que T2. Por otro lado, T1 no lee de T2.
- No es serial porque tiene operaciones entrelazadas.
- No evita abortos en cascada porque T2 lee de transacciones no confirmadas.

iii)



- El grafo de seriabilidad tiene ciclos, por tanto H3 no es serializable.
- Es recuperable porque T2 lee de T1 (lee X, dato escrito por T1 previamente) y T1 confirma antes que T2. Por otro lado, T1 no lee de T2, ya que cuando T1 lee Y, éste fue escrito por T1 luego de haber sido escrito por T2.
- No es serial porque tiene operaciones entrelazadas.
- Evita abortos en cascada porque T2 lee de T1 cuando T1 ya confirmó.

b) Dar una historia de T1 y T2 con operaciones entrelazadas, recuperable y donde las transacciones sigan el protocolo 2PL básico. ¿Es Serializable? Justifique.

T1: r1(X), r1(X), w1(X), w1(X), w1(Y), w1(Y), r1(Y), u1(Y), w1(X), u1(X), c1

T2: r12(Y), r2(Y), w12(Y), w2(Y), r12(X), r2(X), w12(X), w2(X), u2(Y), u2(X), c2

H5: r1(X), r1(X), w1(X), w1(X), w1(Y), w1(Y), r1(Y), u1(Y), r12(Y), r2(Y), w12(Y), w2(Y), w1(X), u1(X), c1, r12(X), r2(X), w12(X), w2(X), u2(Y), u2(X), c2

Es recuperable porque T2 lee de T1, T1 no lee de T2, y T1 confirma primero.

Es serializable porque toda historia cuyas transacciones siguen el protocolo 2PL es serializable.

c) Decir qué tipo de protocolo 2PL deberían seguir T1 y T2 para asegurarse de que no van a entrar en deadlock.

T1 y T2 deberían seguir el protocolo 2PL-Conservador, donde todos los locks se hacen antes del comienzo de la transacción.