

FUNDAMENTOS DE BASES DE DATOS
Examen Julio 2016
Solución

Ejercicio 1 (25 puntos).

Interesa registrar la información de todas las franquicias de una red de cobranzas. Para esto, se desea modelar la base de datos que contendrá toda la información al respecto.

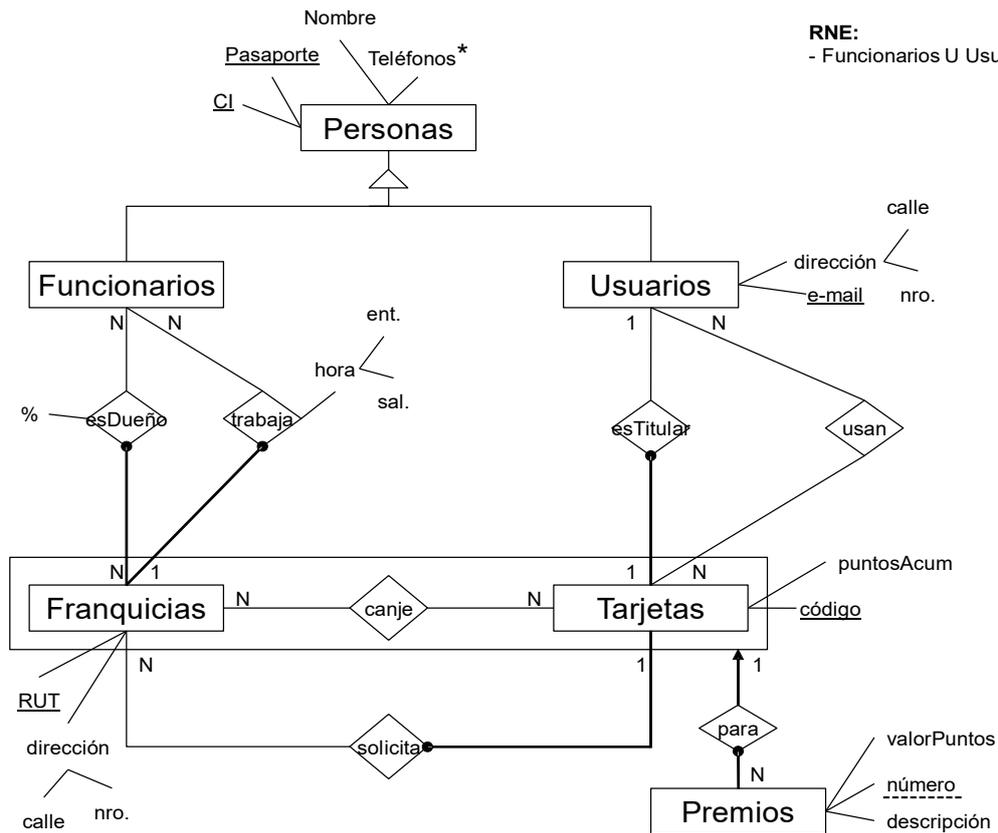
De las franquicias se conoce el RUT que la identifica y su dirección (calle y nro.). Las franquicias pueden ser compradas por personas uruguayas o extranjeras, por esta razón, tanto el pasaporte como la CI identifican a las personas. Además, interesan su nombre y todos sus teléfonos.

Dentro de las personas están los usuarios de la red de pagos y los funcionarios. De los primeros se registra su e-mail, que es único, y su dirección (calle y nro.). Los dueños de cada franquicia pueden ser una sociedad de funcionarios, y de cada uno interesa saber el porcentaje que le corresponde dentro de la sociedad. A su vez, un funcionario puede pertenecer a varias sociedades dentro de cada franquicia. Por otro lado, un funcionario trabaja en una única franquicia e interesa registrar la hora de entrada y de salida en la misma. Por supuesto, los funcionarios pueden ser usuarios de la red de cobranzas.

Cada vez que un usuario hace uso de la red de cobranza puede acumular puntos a través de una tarjeta. De dicha tarjeta interesa un código que la identifica, el total de puntos acumulados y el usuario titular de la misma. Aunque la tarjeta tiene un único titular, la misma puede ser usada por varios usuarios, basta con presentar el e-mail del titular.

Los usuarios pueden obtener la tarjeta solicitándola en una de las franquicias, pero el canje de los puntos por premios puede ser realizado en cualquiera de las franquicias. Cada premio tiene un número que lo identifica y que depende de la tarjeta y de la franquicia en la cual fue canjeado. Además, tiene una descripción (cartera de cuero marrón, termo de acero inoxidable, libro de cuento, etc.) y su valor en puntos.

Se pide Modelo Entidad Relación completo.



Ejercicio 2 (25 puntos).

El departamento de gestión de recursos humanos docentes de un instituto de enseñanza terciaria mantiene su información en una base de datos con las siguientes tablas:

DOCENTE (Cl, nomDoc, grado, depto)

Contiene la información de todos los docentes del instituto. De cada uno de ellos se registra su cédula de identidad, su nombre, su grado y el departamento en el que trabaja.

ASIGNATURA (codAsig, nomAsig, tipo, credits)

Contiene la información sobre las asignaturas que se dictan en el instituto. De cada una de ellas se conoce un código identificador, el nombre, el tipo (obligatoria o electiva) y la cantidad de créditos que otorga.

ASIGNACION (año, semestre, codAsig, Cl)

Contiene la información sobre las asignaciones de docentes a asignaturas, las cuales se realizan para cada semestre de cada año.

DICTADO (año, semestre, codAsig)

Contiene la información de los dictados de cada asignatura en cada semestre y año.

En este esquema no existen tablas vacías y se cumplen las siguientes relaciones de inclusión:

- $DICTADO \subseteq \prod_{año, semestre, codAsig} (ASIGNACION)$
- $\prod_{codAsig} (ASIGNACION) \subseteq \prod_{codAsig} (ASIGNATURA)$
- $\prod_{Cl} (ASIGNACION) \subseteq \prod_{Cl} (DOCENTE)$

Resolver en Algebra Relacional las siguientes consultas:

- a. Devolver nombre de los docentes que solo tienen asignaciones a asignaturas de tipo "electiva".

$$A = \Pi_{CI} (ASIGNACION * (\sigma_{\text{tipo=electiva}} ASIGNATURA))$$

docentes asignados a asignaturas de tipo electiva

$$B = \Pi_{CI} (ASIGNACION * (\sigma_{\text{tipo} \neq \text{electiva}} ASIGNATURA))$$

docentes asignados a asignaturas de tipo distinto a electiva

$$SOL = \Pi_{\text{nombre}} ((A-B) * DOCENTE)$$

- b. Devolver nombre, asignatura, año y semestre de dictado tal que para ese dictado están asignados todos los docentes grado 1 del departamento "Programación".

$$A = \Pi_{CI} (\sigma_{\text{grado=1 and depto=PROGRAMACION}} (DOCENTE))$$

$$SOL = \Pi_{\text{nombre,año,semestre}} ((ASIGNACION \% A) * ASIGNATURA)$$

Resolver en Cálculo Relacional las siguientes consultas:

- c. Devolver las triplas año, nombre de asignatura, nombre de docente, tal que en ese año el docente está asignado a esta asignatura por primera vez.

$$\{ a.\text{año}, a.\text{nombre}, d.\text{nombre} / ASIGNACION(a) \wedge ASIGNATURA(as) \wedge DOCENTE(d) \wedge \\ a.\text{codAsig} = as.\text{codAsig} \wedge a.\text{ci} = d.\text{ci} \\ \neg (\exists a') (ASIGNACION(a') \wedge a'.\text{ci} = a.\text{ci} \wedge a'.\text{codAsig} = a.\text{codAsig} \wedge a'.\text{año} < a.\text{año}) \\ \}$$

- d. Obtener nombre y cantidad de créditos de las asignaturas tales que todos los docentes con grado mayor que 2 han sido asignados a ellas por lo menos una vez.

$$\{ a.\text{nombre}, a.\text{creditos} / ASIGNATURA(a) \wedge \\ (\forall d) (DOCENTE(d) \wedge d.\text{grado} > 2 \\ \rightarrow (\exists as)(ASIGNACION(as) \wedge as.\text{codAsig} = a.\text{codAsig} \wedge as.\text{Ci} = d.\text{as})) \\ \}$$

Resolver en SQL, sin utilizar vistas ni sub-consultas en el FROM, las siguientes consultas:

- e. Obtener nombre y tipo de las asignaturas tales que en todos sus dictados tienen más de 3 docentes asignados.

```
SELECT A.nombre,A.tipo
FROM ASIGNATURAS A
WHERE NOT EXISTS
  (SELECT *
   FROM DICTADO D
   WHERE D.codAsig = A.codAsig
   and (D.codAsig,D.año,D.semestre) NOT IN
     (SELECT codAsig, año, semestre
      FROM ASIGNACION
      GROUP BY codAsig,año,semestre
      HAVING count(*) > 3
     )
  )
```

Ejercicio 3 (25 puntos).

Dada la siguiente realidad:

En el sector de tránsito de una intendencia se lleva el registro de las infracciones cometidas por automovilistas. Para esto se almacenan los siguientes datos.

Sobre el automóvil (cuando hablamos de automóvil nos referimos a automóvil que tuvo infracción) se registra la matrícula (que lo identifica), la marca, el modelo y el color. Además se registra la fecha de la infracción. Sobre el conductor, se registra su CI, su nombre, su dirección y su teléfono. La infracción es detectada por uno o más inspectores. Del inspector se registra su CI, su nombre, su rango y su teléfono. La infracción se comete en una o más calles (porque puede ser un cruce). De cada calle se registra el nombre (que la identifica), y la cantidad de cuadras de longitud que tiene. Se asume que en una fecha un conductor puede tener una sola infracción, y por otro lado, en una fecha un mismo automóvil puede tener una sola infracción. Esto significa que para una fecha y un conductor hay un solo automóvil asociado, y por otro lado, dados una fecha y un automóvil hay un solo conductor asociado.

Utilizando los siguientes nombres para los atributos:

mat	nom-c	tel-i
marca	dir-c	nom-calle
modelo	tel-c	long
color	ci-i	
fecha	nom-i	
ci-c	rango	

Se pide:

- a) Deducir todas las dependencias funcionales que se cumplen.

mat \rightarrow marca, modelo, color

ci-c \rightarrow nom-c, dir-c, tel-c

ci-i \rightarrow nom-i, rango, tel-i

nom-calle \rightarrow long

fecha, ci-c \rightarrow mat

fecha, mat \rightarrow ci-c

- b) Para la relación universal R (tabla que contiene todos los atributos), y el conjunto de dependencias funcionales hallado en la parte a), hallar todas las claves. Justificar.

R (mat, marca, modelo, color, ci-c, nom-c, dir-c, tel-c, ci-i, nom-i, rango, tel-i, nom-calle, long, fecha)

Atributos que no se encuentran a la derecha de ninguna df: { ci-i, nom-calle, fecha }, deben estar en todas las claves.

Atributos que no se encuentran a la izquierda de ninguna df: { marca, modelo, color, nom-c, dir-c, tel-c, nom-i, rango, tel-i, long }, no pueden estar en ninguna clave.

$(ci-i, nom-calle, fecha)^+ = \{ ci-i, nom-calle, fecha, nom-i, rango, tel-i, long \}$

Probamos agregando atributos que puedan estar en alguna clave:

$(ci-i, nom-calle, fecha, ci-c)^+ = R \Rightarrow$ es clave

$(ci-i, nom-calle, fecha, mat)^+ = R \Rightarrow$ es clave

Son las únicas claves porque el resto de los atributos no pueden estar en ninguna clave.

- c) Dar una descomposición de R en 3NF sin pérdida de dependencias y con JSP. Aplicar el algoritmo dado en el curso (mostrando los pasos).

Dados R y

$F = \{ \text{mat} \rightarrow \text{marca, modelo, color}$
 $\text{ci-c} \rightarrow \text{nom-c, dir-c, tel-c}$
 $\text{ci-i} \rightarrow \text{nom-i, rango, tel-i}$
 $\text{nom-calle} \rightarrow \text{long}$
 $\text{fecha, ci-c} \rightarrow \text{mat}$
 $\text{fecha, mat} \rightarrow \text{ci-c} \}$

Paso 1) Hallamos F_{minimal} :

$F_1 = \{ \text{mat} \rightarrow \text{marca}$
 $\text{mat} \rightarrow \text{modelo}$
 $\text{mat} \rightarrow \text{color}$
 $\text{ci-c} \rightarrow \text{nom-c}$
 $\text{ci-c} \rightarrow \text{dir-c}$
 $\text{ci-c} \rightarrow \text{tel-c}$
 $\text{ci-i} \rightarrow \text{nom-i}$
 $\text{ci-i} \rightarrow \text{rango}$
 $\text{ci-i} \rightarrow \text{tel-i}$
 $\text{nom-calle} \rightarrow \text{long}$
 $\text{fecha, ci-c} \rightarrow \text{mat}$
 $\text{fecha, mat} \rightarrow \text{ci-c} \}$

Buscamos dfs. con atributos redundantes a la izquierda:

- $\text{fecha, ci-c} \rightarrow \text{mat}$

$(\text{fecha})^+ = \{ \text{fecha} \} \Rightarrow \text{ci-c}$ no es redundante porque no obtuve mat

$(\text{ci-c})^+ = \{ \text{ci-c, nom-c, dir-c, tel-c} \} \Rightarrow \text{fecha}$ no es redundante porque no obtuve mat

- $\text{fecha, mat} \rightarrow \text{ci-c}$

$(\text{fecha})^+ = \{ \text{fecha} \} \Rightarrow \text{mat}$ no es redundante porque no obtuve ci-c

$(\text{mat})^+ = \{ \text{marca, modelo, color} \} \Rightarrow \text{fecha}$ no es redundante porque no obtuve ci-c

Buscamos dfs. redundantes:

No existen dfs. redundantes, ya que los atributos de la derecha de cada una de ellas no se encuentran a la derecha de ninguna otra df, por lo tanto no se podrían obtener si no estuviera esa df.

Por lo tanto: $F_{\text{minimal}} = F_1$

Paso 2) Generamos una relación para cada grupo de dfs con el mismo lado izquierdo

R1 (mat, marca, modelo, color)

R2 (ci-c, nom-c, dir-c, tel-c)

R3 (ci-i, nom-i, rango, tel-i)

R4 (nom-calle, long)

R5 (fecha, ci-c, mat)

Paso 3) Generamos una relación, cuyos atributos sean los de una de las claves, para asegurar la propiedad de JSP.

R6 (ci-i, nom-calle, fecha, mat)

Descomposición $\rho = \{ R1, R2, R3, R4, R5, R6 \}$

- d) Decir si la descomposición anterior está en BCNF, justificando. Si no está en BCNF, llevarla a dicha forma normal, aplicando el algoritmo dado en el curso (mostrando los pasos).

Para que la descomposición r esté en BCNF, cada relación debe estar en BCNF:

R1 (mat, marca, modelo, color)

$F_{R1} = \{ \text{mat} \rightarrow \text{marca, modelo, color} \}$

=> clave única: mat

R1 está en BCNF, ya que todas sus dfs cumplen que su lado izquierdo es superclave.

R2 (ci-c, nom-c, dir-c, tel-c)

$F_{R2} = \{ \text{ci-c} \rightarrow \text{nom-c, dir-c, tel-c} \}$

=> clave única: ci-c

R2 está en BCNF, ya que todas sus dfs cumplen que su lado izquierdo es superclave.

R3 (ci-i, nom-i, rango, tel-i)

$F_{R3} = \{ \text{ci-i} \rightarrow \text{nom-i, rango, tel-i} \}$

=> clave única: ci-i

R3 está en BCNF, ya que todas sus dfs cumplen que su lado izquierdo es superclave.

R4 (nom-calle, long)

$F_{R4} = \{ \text{nom-calle} \rightarrow \text{long} \}$

=> clave única: (nom-calle)

R4 está en BCNF, ya que todas sus dfs cumplen que su lado izquierdo es superclave.

R5 (fecha, ci-c, mat)

$F_5 = \{ \text{fecha, ci-c} \rightarrow \text{mat, fecha, mat} \rightarrow \text{ci-c} \}$

=> claves: (fecha, ci-c), (fecha, mat)

R5 está en BCNF, ya que todas sus dfs cumplen que su lado izquierdo es superclave.

R6 (ci-i, nom-calle, fecha, mat)

$F_6 = \{ \}$

=> clave única: (ci-i, nom-calle, fecha, mat)

R6 está en BCNF, ya que no tiene dfs por lo tanto ninguna viola la definición de BCNF.

Finalmente, **p está en BCNF.**

- e) Deducir todas las dependencias multivaluadas (no triviales) que se cumplen en cada tabla de la última descomposición obtenida. Para cada multivaluada que proponga, mostrar qué parte de la letra lo induce a pensar que ésta se cumple.

Existe una dependencia multivaluada solamente en R6 (ci-i, nom-calle, fecha, mat), y es la siguiente:

fecha, mat -» nom-calle

Por complemento, también se cumple:

fecha, mat -» ci-i

por "La infracción es detectada por uno o más inspectores" y "La infracción se comete en una o más calles", y considerando que las calles y los inspectores son independientes entre sí.

- f) Llevar la última descomposición obtenida a 4NF, aplicando el algoritmo dado en el curso (mostrando los pasos).

R1, R2, R3, R4 y R5, ya se encuentran en 4NF, porque no tienen dependencias multivaluadas no triviales.

Debemos llevar R6 a 4NF, aplicando el algoritmo visto en el curso:

R6 (ci-i, nom-calle, fecha, mat)

$D_{R6} = \{ \text{fecha, mat -} \rightarrow \text{nom-calle} \}$

clave: (ci-i, nom-calle, fecha, mat)

fecha, mat -» nom-calle viola 4NF porque el lado izquierdo no es superclave.

Tomamos la mvd fecha, mat -» nom-calle, que viola 4NF, y generamos:

R61 (fecha, mat, nom-calle), $D_{R61} = \{ \}$, ya que solo tengo una mvd trivial => está en **4NF**

R62 (fecha, mat, ci-i), $D_{R62} = \{ \}$, ya que solo tengo una mvd trivial => está en **4NF**

Descomposición final, en 4NF:

{R1, R2, R3, R4, R5, R61, R621, R622}

Ejercicio 4 (25 puntos)

La siguiente es una parte de la base de datos de una cadena de supermercados:

PRODUCTOS (codProd, nomProd, descripción, precio)
 Contiene los datos sobre los productos manejados por los supermercados.

PROVEEDORES (nroRut, nomProv, nomContacto, dirProv, telProv)
 Contiene los datos de las empresas que son proveedores de los supermercados y la persona que es el contacto referente de cada empresa proveedora.

COMPRAS (nroRut, codProd, fechaPedido, fechaEntrega, cantCompra)
 Contiene los datos de las compras de productos realizadas a los proveedores, la fecha de pedido, la fecha de entrega y la cantidad comprada.

Considere la siguiente consulta:

```
SELECT P.nomProd, P.precio, R.nomProv
FROM Productos P, Proveedores R, Compras C
WHERE P.codProd = C.codProd AND R.nroRut = C.nroRut AND
      C.fechaPedido = '12/7/2016' AND P.precio > 300;
```

y la siguiente información sobre la base de datos;

	PRODUCTOS	PROVEEDORES	COMPRAS
Cantidad Tuplas	300000	520	1200000
Índices primarios	IndP – por codProd	IndR – por nroRut	IndC – por (codProd, nroRut)
Índices secundarios	IndPP – por precio		IndFP - por fechaPedido
Cantidad de tuplas por bloque	50	30	50
Información adicional	Existen 40 rangos de precios, considerando cada 100 pesos (hasta 100, hasta 200, etc.), donde hay distribución uniforme de los precios.		Por día se hacen 150 pedidos

Se pide:

- Construir el plan lógico que construiría un manejador, comenzando por el árbol canónico y luego aplicando las heurísticas de optimización. Calcular los tamaños (cantidad de tuplas) intermedios que sean necesarios para la aplicación de las heurísticas. Mostrar cada paso aplicado en el proceso.
- Estimar todos los tamaños (cantidad de tuplas) obtenidos luego de cada operación aplicada en el árbol. Justifique.
- Dar un plan físico correspondiente al plan lógico construido en la parte a), que utilice los índices disponibles siempre que sea posible.