

FUNDAMENTOS DE BASES DE DATOS

Examen Diciembre 2009 - SOLUCION

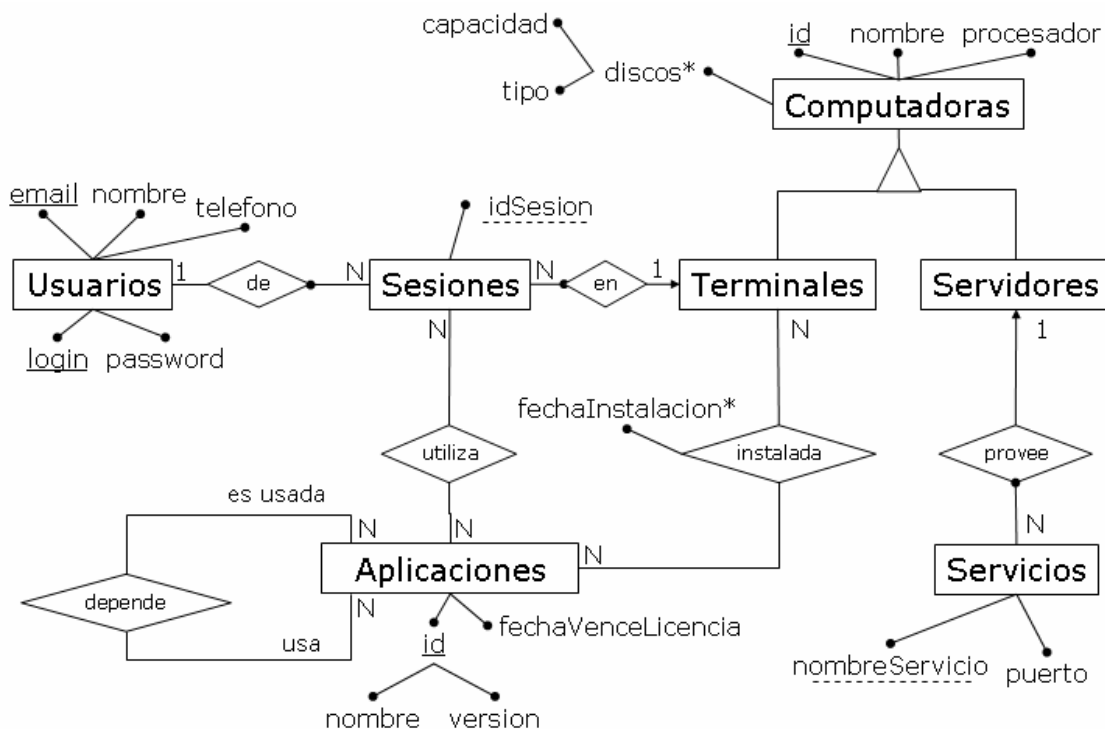
La duración del examen es de 3 horas y ½.

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 (20 puntos)

- a) Dado el siguiente MER deducir el esquema relacional que le parezca más adecuado. Se deben indicar: (1) tablas, (2) clave primaria de cada tabla, (3) dependencias que se cumplen en cada tabla y (4) dependencias de inclusión.



RNE

$\text{Terminales} \cap \text{Servidores} = \emptyset$

$\text{Terminales} \cup \text{Servidores} = \text{Computadoras}$

$\forall t \in \text{terminales}, \forall s \in \text{sesiones}, \forall a \in \text{aplicaciones} (\langle s, t \rangle \in \text{En} \wedge \langle s, a \rangle \in \text{Utiliza} \rightarrow \langle a, t \rangle \in \text{Instalada})$

Si $\langle a1, a2 \rangle \in \text{Depende} \rightarrow a1.id \subset a2.id$

SOLUCION

Entidades

Usuarios (login, nombre, telefono, email, password)

login -> nombre, telefono, email, password

email -> login

Aplicaciones (nombreAplicacion, version, fechaVenceLicencia)

nombreAplicacion, version -> fechaVenceLicencia

Computadoras (idComputadora, nombre, procesador)

idComputadora -> nombre, procesador

Terminales (idTerminal)

$\Pi_{idTerminal}(\text{Terminales}) \subset \Pi_{idComputadora}(\text{Computadoras})$

Servidores (idServidor)

$\Pi_{idServidor}(\text{Servidores}) \subset \Pi_{idComputadora}(\text{Computadoras})$

Atributos multivalorados de Entidades

Discos (idComputadora, tamaño, tipo)

$\Pi_{idComputadora}(\text{Discos}) \subset \Pi_{idComputadora}(\text{Computadoras})$

Entidades débiles

Servicios (nombreServicio, idServidor, puerto)

nombreServicio, idServidor -> puerto

$\Pi_{idServidor}(\text{Servicios}) \subset \Pi_{idServidor}(\text{Servidor})$

Sesiones (idSesion, idTerminal, login)

idSesion, idTerminal -> login

$\Pi_{idTerminal}(\text{Sesiones}) \subset \Pi_{idTerminal}(\text{Terminales})$

$\Pi_{login}(\text{Sesiones}) \subset \Pi_{login}(\text{Usuarios})$

Relaciones

La relación *de* entre usuarios y sesiones se modela agregando el atributo *login* a la tabla *sesiones*.

Depende (nombreApUsa, versionUsa, nombreApUsada, versionUsada)

$\Pi_{nombreApUsa, versionUsa}(\text{Depende}) \subset \Pi_{nombreAplicacion, version}(\text{Aplicaciones})$

$\Pi_{nombreApUsada, versionUsada}(\text{Depende}) \subset \Pi_{nombreAplicacion, version}(\text{Aplicaciones})$

Instalada (nombreAplicacion, version, idTerminal, fechaInstalacion)

$\Pi_{nombreAplicacion, version}(\text{Instalada}) \subset \Pi_{nombreAplicacion, version}(\text{Aplicaciones})$

$\Pi_{idTerminal}(\text{Terminales}) \subset \Pi_{idComputadora}(\text{Computadoras})$

Utiliza (idSesion, idTerminal, nomApp, verApp)

$\Pi_{idSesion, idTerminal}(\text{Utiliza}) \subset \Pi_{idSesion, idTerminal}(\text{Sesion})$

$\Pi_{nomAplicacion, verApp}(\text{Utiliza}) \subset \Pi_{nombreAplicacion, version}(\text{Aplicaciones})$

b) Decir en qué forma normal se encuentra el esquema obtenido en la parte a.

SOLUCION

(1) Usuarios:

Login es clave (determina a todos) y email también.

Las únicas dks que se proyectan tienen claves a la izquierda -> BCNF.

No hay dmvs no triviales -> 4NF

- (2) Aplicaciones
Misma justificación que para (1) con (nombreAplicacion, versión) única clave -> 4NF
- (3) Computadoras
Misma justificación que para (1) con (idComputadora) única clave -> 4NF
- (4) Terminales
No se proyectan dfs ni dmvs -> 4NF
- (5) Servidores
Misma justificación que para (4) -> 4NF
- (6) Discos
Misma justificación que para (4) -> 4NF
- (7) Servicios
Misma justificación que para (1) con (nombreServicio, idServidor) única clave-> 4NF
- (8) Sesiones
Misma justificación que para (1) con (idSesion, idTerminal) única clave-> 4NF
- (9) Depende
Misma justificación que para (4) -> 4NF
- (10) Instalada
Misma justificación que para (4) -> 4NF
- (11) Utiliza
Misma justificación que para (4) -> 4NF

De todo esto se deduce que el esquema obtenido en la parte a está en 4NF

- c) Ahora suponga que se cumple la siguiente propiedad:
 - a. El nro. de puerto determina el nombre del servicio.Decir en qué forma normal se encuentra el esquema obtenido en la parte a.

SOLUCION

En la tabla Servicios tengo:

Servicios (nombreServicio, idServidor, puerto)
nombreServicio, idServidor -> puerto
puerto -> nombreServicio

Claves: (nombreServicio, idServidor), (puerto, idServidor)

La df puerto -> nombreServicio no tiene una superclave del lado izquierdo, por lo tanto viola BCNF. El atributo *nombreServicio* depende parcialmente de una clave pero es primo, por lo tanto esta df no viola 2NF -> el esquema está en 3NF

- d) Considerando las condiciones impuestas en la parte c, llevar el esquema relacional a BCNF. Decir si se pierden dependencias funcionales.

SOLUCION

F = { login -> nombre, telefono, email, password

```

email ->login
nombreAplicacion, version -> fechaVenceLicencia
idComputadora -> nombre, procesador
nombreServicio, idServidor ->puerto
idSesion, idTerminal -> login
puerto ->nombreServicio
}

```

Aplicando el algoritmo visto en curso, como puerto->nombreServicio viola BCNF, descompongo la tabla servicios en:

servicios1(idServidor, puerto)

$\Pi_{servicios1}(F) = \{ \}$

Como no se proyecta ninguna df está en BCNF y como además no hay dmvs no triviales está en 4NF

servicios2(puerto, nombreServicio)

$\Pi_{servicios2}(F) = \{ \text{puerto} \rightarrow \text{nombreServicio} \}$

La única clave en este esquema es puerto. La única df que se proyecta tiene una clave en el lado izquierdo entonces está en BCNF. Como además no hay dmvs no triviales está en 4NF.

```

Ahora F' = { login -> nombre, telefono, email, password
email ->login
nombreAplicacion, version -> fechaVenceLicencia
idComputadora -> nombre, procesador
idSesion, idTerminal -> login
puerto ->nombreServicio
}

```

Quiero saber si se conserva la df nombreServicio, idServidor ->puerto, calculo

$(\text{nombreServicio, idServidor})^+_{F'} = \{ \text{nombreServicio, idServidor} \}$ como el atributo puerto no está en la clausura puedo decir que esta df se pierde, entonces la descomposición no preserva dependencias.

Ejercicio 2 (10 puntos).

Dado el esquema relación $R(A,B,C,D,E)$ y el conjunto de dependencias F sobre R :

$F = \{ A \rightarrow BC, CD \rightarrow E, B \rightarrow A, D \rightarrow C, AD \rightarrow E \}$

Se pide:

- 1) Dar un conjunto de dependencias G que cumple las siguientes condiciones (todas simultáneamente) :
 - a) G es no vacío.
 - b) G es minimal.
 - c) $G^+ \subseteq F^+$.
 - d) G y F no son equivalentes.

Justificar su respuesta.

Solución:

Sea $G = \{A \rightarrow B\}$

- a) G es no vacío.
- b) G es minimal: todas las dependencias son tales que el lado derecho tiene un único atributo, no hay atributos redundantes en los lados izquierdos de las df y no hay dependencias redundantes.

- c) $G^+ \subseteq F^+$ si todas las dependencias de G pertenecen a F^+ .
 $A \rightarrow B \in F^+$ (por aplicación de descomposición a $A \rightarrow BC$).
 Por lo tanto se cumple $G^+ \subseteq F^+$
- d) Teniendo en cuenta lo probado en c), para que G y F no sean equivalentes es necesario que existe una dependencia de F que no se deduzca de las dependencias de G.
 Sea $B \rightarrow A$ pertenece a F y no se deduce de las dependencias de G ya que $B^+_G = \{B\}$ que no contiene a A.

2) Dar un conjunto de dependencias J que cumple las siguientes condiciones (todas simultáneamente):

- J es minimal.
- $F^+ \subseteq J^+$.
- J y F no son equivalentes.

Justificar su respuesta.

Solución:

Construiremos un cubrimiento minimal de F

$$F = \{A \rightarrow BC, CD \rightarrow E, B \rightarrow A, D \rightarrow C, AD \rightarrow E\}$$

Paso 1: Dependencias con un único atributo del lado derecho:

$$F_1 = \left\{ \begin{array}{l} A \rightarrow B, \\ A \rightarrow C, \\ CD \rightarrow E, \\ B \rightarrow A, \\ D \rightarrow C, \\ AD \rightarrow E \end{array} \right\}$$

Paso 2: Eliminación de atributos redundantes en los lados izquierdos:

$$C^+_{F_1} = \{C\}$$

$$D^+_{F_1} = \{D, C, E\} \Rightarrow C \text{ es redundante en } CD \rightarrow E \text{ y } A \text{ es redundante en } AD \rightarrow E$$

$$F_2 = \left\{ \begin{array}{ll} A \rightarrow B, & (1) \\ A \rightarrow C, & (2) \\ D \rightarrow E, & (3) \\ B \rightarrow A, & (4) \\ D \rightarrow C & (5) \end{array} \right\}$$

Paso 3: Eliminación de dependencias redundantes.

Las dependencias 1, 3 y 4 no son redundantes por ser las únicas que tienen su respectivo lado derecho.

Verifico si 2 es redundante:

$$A^+_{F_2 - \{A \rightarrow C\}} = \{A, B\} \quad \text{Por lo tanto la df 2 no es redundante.}$$

Verifico si 5 es redundante:

$$D^+_{F_2 - \{D \rightarrow C\}} = \{D, E\} \quad \text{Por lo tanto la df 5 no es redundante.}$$

Por lo tanto F_2 es minimal y equivalente a F.

$$\text{Sea } J = F_2 \cup \{B \rightarrow D\}$$

$$F \text{ eq } F_2 \Rightarrow F^+ = F_2^+ \quad (i)$$

$$F_2 \subset J \Rightarrow F_2^+ \subseteq J^+ \quad (ii)$$

De (i) y (ii) se puede afirmar que $F^+ \subseteq J^+$

J^+ no está incluido en F^+ si existe una df de J que no pertenece a F^+ .

Sea $B \rightarrow D \in J$, $B \rightarrow D \notin F^+$ ya que $B^+_F = \{B, A, C\}$ y $D \notin B^+_F$

$\Rightarrow J$ y F no son equivalentes.

Sea

$$J = \left\{ \begin{array}{l} A \rightarrow B, \quad (1) \\ A \rightarrow C, \quad (2) \\ D \rightarrow E, \quad (3) \\ B \rightarrow A, \quad (4) \\ D \rightarrow C \quad (5) \\ B \rightarrow D \quad (6) \end{array} \right\}$$

J cumple las siguientes condiciones:

Todas las dependencias tienen un único atributo en el lado derecho.

Todas las dependencias tienen un único atributo en el lado izquierdo por lo tanto no tienen atributos redundantes.

Verificamos si existen dependencias redundantes:

Las posibles son la 2 y la 5 ya que son las únicas que tienen el mismo atributo en el lado derecho.

Verifico si 2 es redundante:

$A^+_{J - \{A \rightarrow C\}} = \{A, B, D, C\}$ Por lo tanto la df 2 es redundante.

$$J_1 = \left\{ \begin{array}{l} A \rightarrow B, \quad (1) \\ D \rightarrow E, \quad (2) \\ B \rightarrow A, \quad (3) \\ D \rightarrow C \quad (4) \\ B \rightarrow D \quad (5) \end{array} \right\}$$

J_1 es equivalente a J y por las propiedades antes mencionadas **es minimal**.

J_1 cumple todas las condiciones pedidas.

Ejercicio 3 (25 puntos).

La administración de un SPA mantiene toda la información en una base de datos con el siguiente esquema:

PROFESIONALES (IdProf, NomProf, Telefono, FechaUltAct)

Esta tabla contiene información acerca de los profesionales que trabajan en el SPA. Registra el identificador del profesional, su nombre, su teléfono y la fecha en la que realizó el último curso de actualización.

TRATAMIENTOS (IdTrat, NomTrat, CantDias)

Esta tabla contiene información acerca de los tratamientos que se realizan en el SPA. Registra el identificador del tratamiento, el nombre del mismo y la cantidad de días que son necesarios para realizar dicho tratamiento. El nombre de los tratamientos también los identifica.

REALIZAN_TRAT (IdProf, IdTrat)

Esta tabla contiene información acerca de quién realiza los diferentes tratamientos.

CLIENTES (IdCliente, NomCliente, Telefono)

Esta tabla contiene información acerca de los clientes del SPA. Registra el identificador del cliente, su nombre y su teléfono.

CLIENTES_TRAT (IdCliente, FechaInicio, IdTrat, IdProf)

En esta tabla se registran los contratos de tratamientos por parte de los clientes. De cada contrato se conoce además la fecha de inicio y el profesional que elige para que se lo realice. Además, un mismo tratamiento puede ser contratado por un cliente más de una vez.

SESIONES_TRAT (IdCliente, Fecha, Hora, IdTrat)

Esta tabla contiene información de las sesiones de tratamientos que se realizan los clientes del SPA. Un cliente nunca se realiza más de una sesión en un mismo día. Un tratamiento que tiene una duración de x días, requiere de x sesiones para ser realizado.

En este esquema no existen tablas vacías y se cumplen las siguientes dependencias de inclusión.

- $\prod_{IdProf}(REALIZAN_TRAT) \subseteq \prod_{IdProf}(PROFESIONALES)$
- $\prod_{IdTrat}(REALIZAN_TRAT) \subseteq \prod_{IdTrat}(TRATAMIENTOS)$
- $\prod_{IdCliente}(CLIENTES_TRAT) \subseteq \prod_{IdCliente}(TRATAMIENTOS)$
- $\prod_{IdProf, IdTrat}(CLIENTES_TRAT) \subseteq REALIZAN_TRAT$
- $\prod_{IdCliente, IdTrat}(SESIONES_TRAT) \subseteq \prod_{IdCliente, IdTrat}(CLIENTES_TRAT)$
- $\prod_{IdCliente, FechaInicio, IdTrat}(CLIENTES_TRAT) \subseteq \prod_{IdCliente, Fecha, IdTrat}(SESIONES_TRAT)$

Resolver las siguientes consultas en Álgebra Relacional:

- Devolver el nombre y el teléfono de los profesionales que realizan únicamente tratamientos de un día de duración.

Solución:

$$A = \prod_{IdProf} (REALIZAN_TRAT * \sigma_{CantDias=1}(TRATAMIENTOS))$$

$$B = \prod_{IdProf} (REALIZAN_TRAT * \sigma_{CantDias \neq 1}(TRATAMIENTOS))$$

$$SOL = \prod_{NomProf, Telefono} (PROFESIONALES * (A - B))$$

- 2) Devolver el nombre de todos los tratamientos que requieren de la máxima cantidad de días para ser realizados y que además hay alguna sesión del mismo que se realiza el día '15/12/2009'.

Solución:

$$A = \prod_{\$1, \$4} (TRATAMIENTOS \underset{\$3 \geq \$6}{><} TRATAMIENTOS)$$

$$B = A \% \prod_{\$1} (TRATAMIENTOS)$$

$$SOL = \prod_{NomTrat} ((B * (\sigma_{Fecha = '15/12/09'}(SESIONES_TRAT)))) * TRATAMIENTOS)$$

Resolver las siguientes consultas en Cálculo Relacional:

- 3) Obtener el nombre del profesional tal que todos los clientes que contratan tratamientos de nombre "COLAGENO" eligen ser atendidos por él.

Solución:

$$SOL = \{ p.NomProf / PROFESIONAL(p) \wedge \\ (\exists t)(TRATAMIENTOS(t) \wedge t.NomTrat = "COLAGENO" \wedge \\ (\forall c)(CLIENTES_TRAT(c) \wedge t.IdTrat = c.IdTrat \rightarrow c.IdProf = p.IdProf) \\) \\ \}$$

- 4) Obtener los nombres de los clientes que no han contratado el mismo tratamiento con 2 profesionales distintos.

Solución:

$$\{c.NomCliente / CLIENTES(c) \wedge \\ \neg (\\ (\exists p1)(CLIENTES_TRAT(p1) \wedge p1.IdCliente = c.IdCliente \wedge \\ (\exists p2) (CLIENTES_TRAT(p2) \wedge p2.IdCliente = c.IdCliente \wedge \\ p1.IdTrat = p2.IdTrat \wedge p1.IdProf \neq p2.IdProf \\) \\) \\) \\ \}$$

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

- 5) Obtener el nombre de los profesionales que realizan más de 3 tratamientos cuya duración es mayor a 10 días.

Solución:

```
SELECT P.NomProf
FROM PROFESIONALES P
WHERE P.IdProf IN
  (SELECT R.IdProf
   FROM TRATAMIENTOS T, REALIZAN_TRAT R
   WHERE T.CantDias > 10 AND
         T.IdTrat = R.IdTrat
   GROUP BY R.IdProf)
```



```
HAVING count(*) > 3  
)
```

- 6) Obtener los nombres de los clientes que contrataron todos los tratamientos que tienen duración de más un día. Si no existe ningún tratamiento que cumpla esta condición no se debe devolver ningún cliente.

Solución:

```
SELECT C.NomCliente  
FROM CLIENTES C  
WHERE NOT EXISTS  
      (SELECT *  
       FROM TRATAMIENTOS T1  
       WHERE T1.CantDias > 1  
       AND NOT EXISTS  
            (SELECT *  
             FROM CLIENTES_TRAT CT  
             WHERE CT.IdTrat = T.IdTrat  
             AND CT.IdCliente = C.IdCliente  
            )  
      )  
AND EXISTS  
      (SELECT *  
       FROM TRATAMIENTOS T2  
       WHERE T2.CantDias > 1  
      )
```

Ejercicio 4 (25 puntos).

- 1) Para cada una de las siguientes historias, indique la cantidad y cuáles son las parejas de operaciones en conflicto que existen.
- i) $r_1(x), w_2(y), w_2(x), r_2(y), c_2, r_1(y), w_1(y), c_1$.
 - ii) $r_1(x), r_2(y), w_1(x), w_2(y), c_2, c_1$.

Solución:

Por definición dos operaciones están en conflicto si cumplen a la vez las siguientes condiciones:

- Pertenecen a distintas transacciones.
- Acceden al mismo ítem.
- Una es un write.

- a. 4 parejas: $(r_1(x), w_2(x)); (w_2(y), r_1(y)); (w_2(y), w_1(y)); (r_2(y), w_1(y))$
- b. Ninguna

- 2) Considere las siguientes historias:

- i) $r_1(x), w_2(y), w_1(x), c_1, r_2(y), c_2$.
- ii) $w_1(x), r_2(x), w_2(x), c_1, w_2(y), r_1(y), c_2$.
- iii) $r_1(x), w_2(z), w_1(y), c_1, w_2(y), r_2(x), c_2$.

- a. Indique cuáles de las siguientes historias no pueden ser generadas por un DBMS. Justifique su respuesta.
- b. Para las historias que SI PUEDEN ser generadas, indique su clase de recuperabilidad (si es recuperable, evita abortos en cascada o es estricta). Justifique su respuesta.

Solución:

- a. La historia ii tiene una operación de la transacción 1 ($r_1(y)$) después de que terminó confirmando. Eso no es posible porque la transacción terminó antes.
- b.
 - i. La historia es estricta, dado que las transacciones no trabajan sobre ítems en común. Por este motivo, no hay operaciones en conflicto y tampoco ocurre que una transacción lea de la otra.
 - ii. La historia también es estricta porque si bien la transacción 1 escribe el ítem y, confirma antes de que la transacción 2 haga algo con él.

3) Considere las siguientes transacciones (s_i indica el comienzo de la transacción i):

$T_1: l_1(x), l_1(y), l_1(z), s_1, w_1(x), r_1(y), w_1(z), u_1(x), u_1(y), u_1(z), c_1.$

$T_2: s_2, l_2(x), r_2(x), u_2(x), l_2(z), w_2(z), u_2(z), c_2.$

$T_3: s_3, l_3(z), r_3(z), u_3(z), l_3(x), w_3(x), u_3(x), c_3.$

$T_4: s_4, l_4(x), l_4(y), l_4(z), w_4(x), r_4(y), w_4(z), c_4, u_4(x), u_4(y), u_4(z).$

- a) Indique cuáles siguen 2PL y cuáles no. Justifique sus respuestas.
- b) Para las que siguen 2PL indique que tipo de 2PL siguen y por qué.

Solución:

a) La idea de 2PL es que en cada transacción haya dos fases diferenciadas: primero se ganan locks, luego se liberan.

Esta idea hace que una transacción no pueda primero tomar un lock(X), luego liberarlo o degradarlo, y luego tomar un lock(Y). Si hace esto entonces no sigue 2PL. De esta forma, entonces T1 y T4 siguen 2 PL pero T2 y T3 no lo siguen.

b) T1 sigue 2PL conservador, porque **bloquea todo lo que necesita antes de que comience** la transacción.

T4 siguió 2PL estricto, porque **hace algún bloqueo después de su comienzo y libera todos sus bloqueos después de que termina** la transacción. (Si hiciera todos los bloqueos antes de comenzar y los liberara después de su fin sería 2PL riguroso)

Ejercicio 5 (25 puntos).

Se desea modelar la realidad correspondiente a una academia de choferes.

En la academia trabajan instructores, de los cuales se conoce la cédula, el nombre, la categoría de su licencia, y sus teléfonos.

Se cuenta con vehículos que son utilizados para capacitar a los alumnos. Todos los instructores tienen asignado un único vehículo y un vehículo puede estar asignado a un instructor o no (por ejemplo, cuando se encuentra en reparación). Del vehículo se conoce su tipo, que puede ser camión o auto y su matrícula que lo identifica. A los vehículos se le realizan controles periódicamente, cada control de un vehículo es identificado por la fecha en que fue realizado y se registra el resultado del mismo.

Se registra información de los alumnos inscriptos en la academia. De ellos se conoce su cédula, su nombre, teléfonos y dirección. Los alumnos pueden ser de dos tipos, amateur (el que saca la libreta por primera vez) o profesional (ya tiene la licencia de conducir). Del último se conoce la categoría de su libreta de conducción actual.

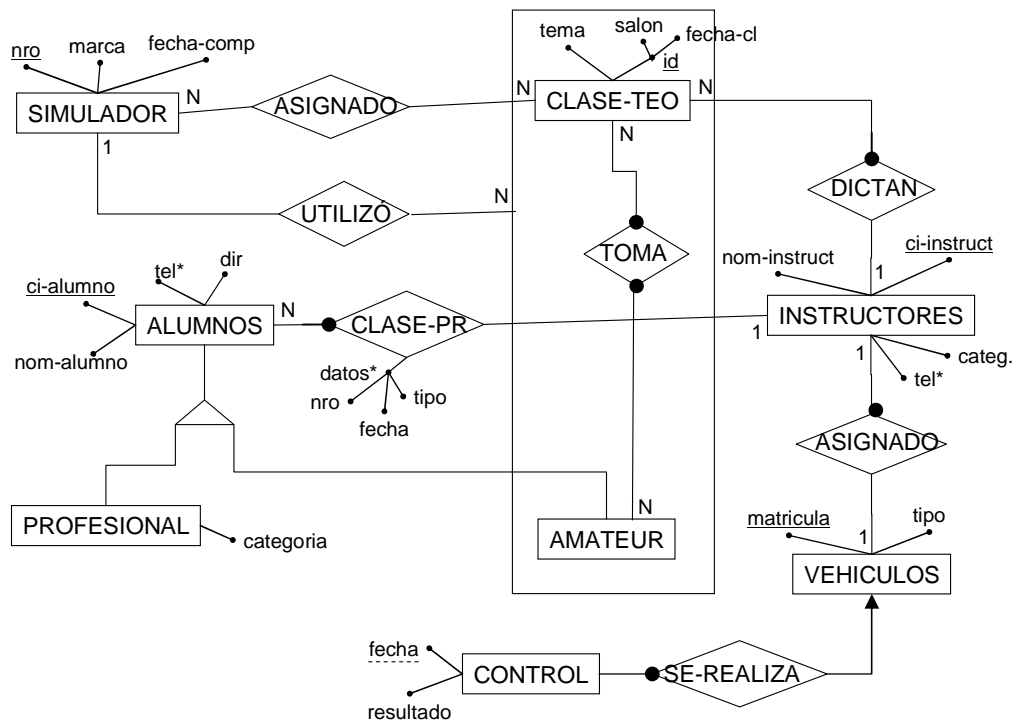
Cada alumno toma clases prácticas con un único instructor (y siempre el mismo). De cada clase práctica interesa registrar el número de clase, la fecha y el tipo de clase práctica, ya que puede ser en ruta o en ciudad. Un alumno puede tomar tantas clases prácticas como se consideren necesarias.

Las clases teóricas son dictadas por los instructores (cada clase tiene un único instructor), no todos dictan clases y un instructor puede dictar varias clases. Interesa registrar de cada clase teórica la fecha y el salón en que es dictada, los cuales la distinguen del resto de las clases, y el tema que es presentado. Todos alumnos que son amateurs y solamente ellos asisten a por lo menos una clase teórica. Los mismos pueden asistir a clases teóricas de cualquier instructor.

Existe un conjunto de simuladores que son utilizados en las clases teóricas. Estos se identifican por un número y se registra también su marca y fecha de compra. Una clase teórica puede tener asignados varios simuladores y un simulador se usa en muchas clases teóricas. Se registra qué alumnos y en qué clases utilizaron cada simulador. Un alumno puede utilizar un solo simulador en una misma clase.

Se pide: Modelo Entidad Relación completo.

Solución:



RNE:

- $ALUMNOS = PROFESIONAL \cup AMATEUR$
- $PROFESIONAL \cap AMATEUR = \emptyset$
- Un alumno AMATEUR con una CLASE-TEO no puede estar en UTILIZO con un SIMULADOR que no está ASIGNADO a esa CLASE-TEO.