

FUNDAMENTOS DE BASES DE DATOS

Examen Febrero 2021

Ejercicio 1 (25 puntos).

Una empresa que brinda seguros para automóviles está armando su sistema para lo que necesita construir una base de datos.

La empresa cuenta con personas de las que se conoce su cédula de identidad, su nombre completo, sus teléfonos y su dirección. Las personas pueden ser clientes o funcionarios. Los funcionarios pueden ser o bien telefonistas o bien técnicos y no existe otro tipo de funcionario. En el caso de ser telefonistas se conoce su horario de atención y en el caso de ser técnicos el vencimiento de la libreta de conducir. Para todos los funcionarios se conoce un email empresarial que los identifica dentro de la empresa.

De los automóviles asegurados se conoce su matrícula, la marca, el modelo, el color y la fecha de vencimiento de la próxima cuota. Además un automóvil asegurado en la empresa debe tener uno o más dueños que deben ser clientes de la misma. Solo interesa registrar información de clientes que son dueños de al menos un automóvil.

La empresa brinda únicamente tres tipos de asistencias: de mecánica ligera, de traslado y por siniestro. Cada asistencia se identifica por la fecha, el lugar donde se realiza y el automóvil al que se le brinda. Para cada asistencia se registra además, el teléfono de contacto quien la solicita.

Para las asistencias de tipo mecánica ligera se conoce una descripción (por ejemplo carga de batería o cambio de rueda) Para las asistencias de tipo traslado se conoce el destino del traslado (a dónde se lleva el coche) y el nombre de la persona que lo recibe. Para las asistencias de tipo siniestro se registra si hubo lesionados.

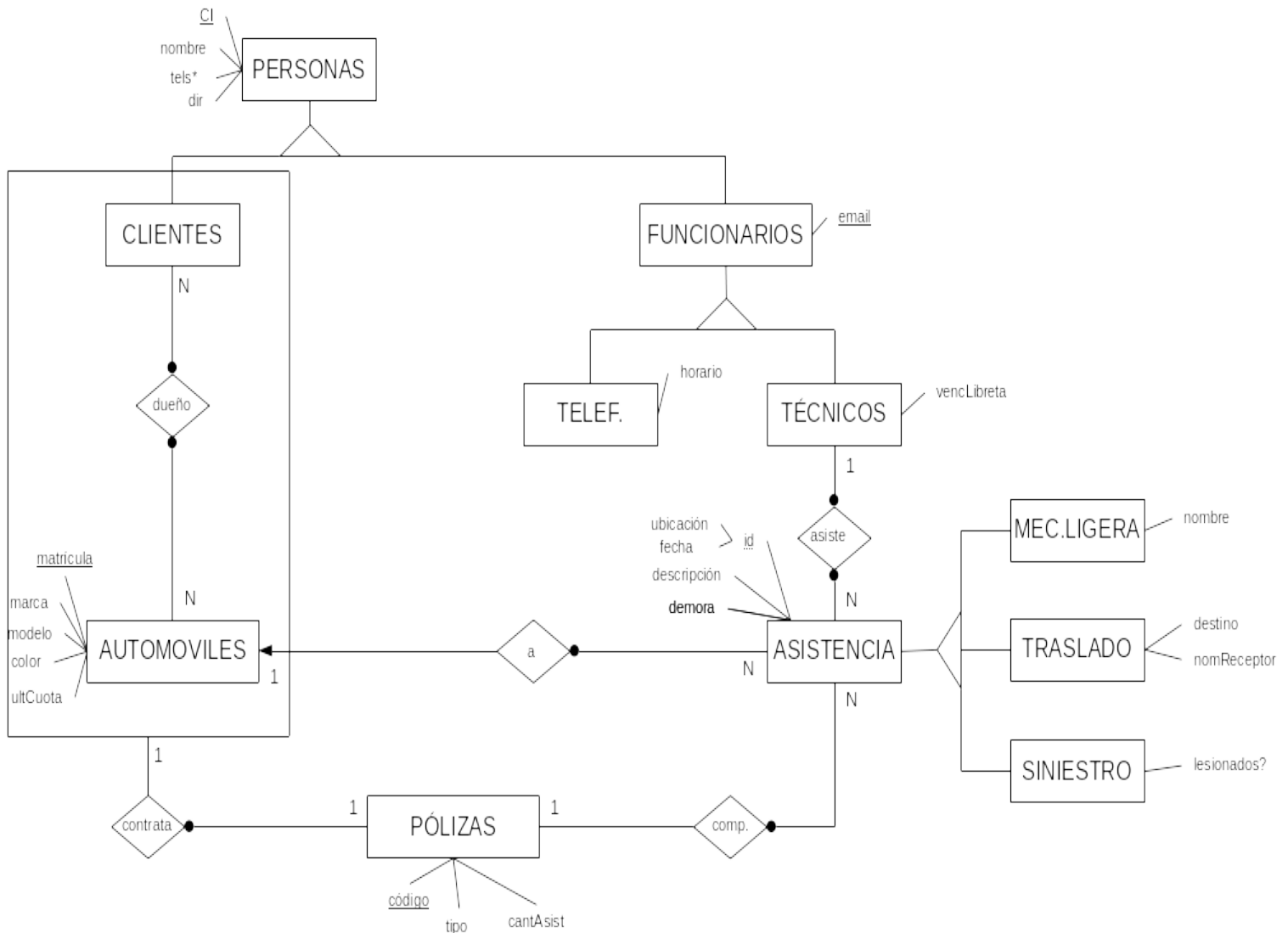
Con respecto a las asistencias, también es de interés conocer el técnico asignado a la misma y la demora estimada para el arribo.

Los clientes contratan pólizas para sus automóviles. Las pólizas se identifican por un código y se conoce el tipo de las mismas (por ejemplo cobertura total, contra terceros, etc) y la cantidad de asistencias que le corresponde. Todos los automóviles deben tener una única póliza contratada y cada póliza le corresponde a un único automóvil. Es de interés saber qué cliente hizo la contratación de la póliza, ya que será el titular de la misma. Para cada póliza se quiere registrar todas las asistencias que fueron utilizadas.

Se Pide:

Modelo Entidad-Relación Completo incluyendo las restricciones no estructurales que correspondan.

SOL:



RNE

$CLIENTES \cup FUNCIONARIOS = PERSONAS$

$TELEFONISTA \cup TECNICO = FUNCIONARIOS$

$TELEFONISTA \cap TECNICO = \emptyset$

$MEC.LIGERA \cup TRASLADO \cup SINIESTRO = ASISTENCIAS$

$MEC.LIGERA \cap TRASLADO = \emptyset$

$MEC.LIGERA \cap SINIESTRO = \emptyset$

$TRASLADO \cap SINIESTRO = \emptyset$

Un automóvil tiene una única póliza contratada

$$(\forall p \in POLIZAS)(\forall a \in AUTOMOVILES)(\forall c_1 \in CLIENTES)(\forall c_2 \in CLIENTES) \\ (\langle \langle c_1, a \rangle, p \rangle \in contrata \wedge \langle \langle c_2, a \rangle, p \rangle \in contrata \rightarrow c_1 = c_2)$$

Las pólizas están compuestas de asistencias que se brindan a automóviles que contrataron dicha póliza

$$(\forall p \in POLIZAS)(\forall as \in ASISTENCIA)(\langle p, as \rangle \in comp \rightarrow (\exists c \in CLIENTES)(\langle \langle c, a(as) \rangle, p \rangle \in contrata))$$

Ejercicio 2 (25 puntos).

Considere un esquema $R(A,B,C,D,E,G)$ y el siguiente conjunto de dependencias funcionales:

$F = \{AB \rightarrow C, C \rightarrow D, DE \rightarrow G, EG \rightarrow C\}$

Considere las descomposiciones:

$\rho_1 = \{R_1(A,B,C), R_2(A,C,D,E,G)\}$

$\rho_2 = \{R_1(A,B,C), R_2(A,B,D,E,G)\}$

1. Para cada una de ellas, indique si tienen Join Sin Pérdida o no.
2. Para cada una de ellas, indique si pierden dependencias o no. En caso de pérdida de dependencias, indique al menos dos dependencias que se pierdan. Justifique todas sus respuestas.
3. Determine todas las claves de R . Justifique sus respuestas.
4. Indique en forma normal están R , ρ_1 y ρ_2 . Justifique sus respuestas.
5. Considere las partes anteriores, y las descomposiciones ρ_1 , ρ_2 y el conjunto de dependencias funcionales originales (No tenga en cuenta la relación R). Cuál de las dos descomposiciones utilizaría para partir a diseñar la base? Justifique su respuesta.

SOL:

1.

Para ver si hay join sin pérdida en una descomposición de dos tablas, estudiamos si se cumple alguna de las dependencias siguientes:

- $R_1 \cap R_2 \rightarrow R_1 - R_2$
- $R_1 \cap R_2 \rightarrow R_2 - R_1$

Para estudiar esto, se calcula la clausura de $R_1 \cap R_2$ que en cada caso es:

ρ_1 : $AC^+ = \{A, C, D\}$ Dado que no está ni $R_1 - R_2$ (B) ni $R_2 - R_1$ (DEG), se puede concluir que **NO TIENE JSP**.

ρ_2 : $AB^+ = \{A, B, C, D\}$ Dado que ya se encuentra a $R_1 - R_2$ (C), se puede concluir que **TIENE JSP**.

2.

Hay varias estrategias para estudiar si se pierden dependencias en una descomposición.

La que parece más simple es buscar si alguna de las dependencias del conjunto original, no se proyecta en la descomposición.

Si encontramos una que no se proyectó, entonces ya se detectó una dependencia que se pierde y lo más seguro allí es hacer la proyecciones del conjunto de dependencias sobre la descomposición. Si, vemos que se proyectan todas, entonces ya sabemos que no se perdió ninguna.

ρ_1 :

$AB \rightarrow C$ se proyecta en R_1 dado que las dos partes de la dependencia están en la tabla.

$C \rightarrow D, DE \rightarrow G, EG \rightarrow C$ se proyectan en R_2 dado que las dos partes de cada dependencia están en la tabla R_2

Conclusión: NO SE PIERDEN DEPENDENCIAS

ρ_2 :

$AB \rightarrow C$ se proyecta en R_1 dado que las dos partes de la dependencia están en la tabla.

$DE \rightarrow G$ se proyecta en R_2 porque las dos partes de la dependencia están en esa tabla.

Sin embargo, $C \rightarrow D$ y $EG \rightarrow C$ se pierden.

Para completar el estudio conviene hacer las proyecciones de las dependencias.

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

$\Pi_{R_2}(F) = \{AB \rightarrow D, DE \rightarrow G, EG \rightarrow D\}$

De esta forma, se detectaron dos dependencias que se pierden "directamente". Sin embargo debería haber alguna más.

Para detectarla, se deduce alguna dependencia más usando las dependencias perdidas y que no se proyecten en ningún esquema.

Por ejemplo, $CE \rightarrow G$ es otra dependencia que se pierde y se deduce por pseudotransitividad de $C \rightarrow D$ y $DE \rightarrow G$.

Conclusión: al menos tres dependencias que se pierden en este caso son $C \rightarrow D$, $EG \rightarrow C$ y $DE \rightarrow G$.

3.

Para determinar todas las claves de R, primero se separan los atributos en tres conjuntos, según su aparición en las dependencias:

Nunca a la derecha: {A,B,E}

Sólo a la derecha: {}

De los dos lados: {CGD}

Ahora se calcula la clausura para los atributos que tienen que estar en todas las claves (Nunca a la derecha):

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

Cómo se alcanzan todos los atributos en la clausura, esta es la única clave de R.

4.

R

En R, hay dependencias parciales de clave sobre atributos no primos.

Ej: $AB \rightarrow C$.

Esto es así porque como se mostró antes, la única clave es ABE por lo que C es un atributo no primo y AB parte de una clave.

Esto hace que R esté en Primera Forma Normal.

ρ_1 :

En R_1 , la única dependencia que se proyecta es $AB \rightarrow C$, por lo que la tabla estaría en 4NF (BCNF sin dependencias MVD)

El resto de las dependencias, se proyectan en R_2

En este caso, conviene calcular primero las claves porque no hay dependencias demasiado obvias que puedan dar una indicación sobre la forma normal como sí había en el caso anterior.

$ND = \{AE\}$

$SD = \{\}$

AmbosLados = {CDG}

$AE^+ = \{A, E\}$ por lo que no es clave.

$AEC^+ = \{A, E, C, D, G\}$ **Clave**

$AED^+ = \{A, E, D, G, C\}$ **Clave**

$AEG^+ = \{A, E, G, C, D\}$ **Clave**

Este resultado hace que todos los atributos sean primos, por lo que **no hay atributos no primos del lado derecho**. Por esta última observación, no hay dependencias ni parciales ni transitivas sobre atributos no primos por lo que está como mínimo en 3NF.

Está en BCNF?, la respuesta es **NO** porque está presente la dependencia $C \rightarrow D$ que no tiene una superclave del lado izquierdo.

Conclusión: La descomposición está en 3NF.

ρ_2 :

La tabla R_1 nuevamente no plantea problemas, está en 4NF como se dijo antes.

Para la tabla R_2 se toma la proyección realizada en la parte 2 que es la siguiente:

$$\Pi_{R_2}(F) = \{AB \rightarrow D, DE \rightarrow G, EG \rightarrow D\}$$

Nuevamente se calculan todas las claves :

$$ND = \{ABE\}$$

$$SD = \{\}$$

$$\text{AmbosLados} = \{DG\}$$

Al hacer la clausura de ABE, se obtiene

$$ABE^+ = \{A, B, E, D, G\}$$

Por este resultado, ABE es la única clave. Además, D y G no serían atributos primos.

La dependencia $AB \rightarrow D$ tiene una parte de la clave del lado izquierdo y un atributo no primo del lado derecho, por lo que sería una dependencia parcial que viola 2NF.

Conclusión: la descomposición está en 1NF.

5.

En la parte 1 se probó que ρ_2 tiene JSP pero ρ_1 no tiene JSP. Esto hace que se deba elegir ρ_2 como punto de partida, dado que esto nos asegura que se tiene la misma información de la realidad. Las dependencias perdidas, se pueden programar, mientras que las tuplas erróneas son imposibles de detectar si no se cuenta con los datos originales.

Ejercicio 3 (25 puntos)

Un sistema de gestión de escuelas de educación primaria de una ciudad maneja una base de datos con el siguiente esquema:

ESCUELAS (nroEsc, nombre, direccion, telefono, barrio, ciDirectora)

Contiene información básica sobre las escuelas de educación primaria de la ciudad.

PERSONAL (CI, nombre, formación, categoría, inspector?)

Contiene información sobre las personas que trabajan en las escuelas. El atributo *formación* contiene el nivel de estudios de la persona, mientras que el atributo *categoría* refiere al número correspondiente a la categoría, el cual se asigna por antigüedad entre otras cosas. El atributo *inspector?* indica si la persona ejerce como inspector, conteniendo los valores "sí" o "no".

TRABAJA (nroEsc, CI, cargo, horas)

Esta tabla contiene la información sobre qué personas trabajan en qué escuelas y cuántas horas diarias lo hacen en cada una. Además, el atributo *cargo* refiere al cargo que desempeñan, por ejemplo, "maestro/a" o "director/a".

INSPECCIONES – ESC (ciInspector, nroEsc, fecha, puntaje)

Contiene información sobre las inspecciones realizadas por los inspectores a las escuelas, la fecha de cada una y el puntaje asignado.

INSPECCIONES – MAES (ciInspector, ciMaestra, fecha, puntaje)

Contiene información sobre las inspecciones realizadas por los inspectores a las maestras, la fecha de cada una y el puntaje asignado.

$$\Pi_{ciDirectora}(ESCUELAS) \subseteq \Pi_{CI}(PERSONAL)$$

$$\Pi_{nroEsc}(TRABAJA) \subseteq \Pi_{nroEsc}(ESCUELAS)$$

$$\Pi_{CI}(TRABAJA) \subseteq \Pi_{CI}(PERSONAL)$$

$$\Pi_{ciInspector}(INSPECCIONES – ESC) \subseteq \Pi_{CI}(PERSONAL)$$

$$\Pi_{nroEsc}(INSPECCIONES – ESC) \subseteq \Pi_{nroEsc}(ESCUELAS)$$

$$\Pi_{ciInspector}(INSPECCIONES – MAES) \subseteq \Pi_{CI}(PERSONAL)$$

$$\Pi_{ciMaestra}(INSPECCIONES – MAES) \subseteq \Pi_{CI}(PERSONAL)$$

a) Resolver en **Álgebra Relacional**:

Devolver la cédula y nombre de las maestras de categoría 3 o superior que nunca fueron inspeccionadas.

$$A = \pi_{CI}(PERSONAL \mid_{><|_{CI=ciMaestra} INSPECCIONES-MAES)$$

$$B = \pi_{CI}(\sigma_{categoría \geq 3}(PERSONAL) * \sigma_{cargo = "maestra"}(TRABAJA))$$

$$Resultado = \pi_{CI, nombre}(PERSONAL * (B - A))$$

b) Resolver en **Cálculo Relacional**:

Devolver las parejas número de escuela y barrio que cumplen que esas escuelas tuvieron un puntaje mayor que 50 en todas las inspecciones recibidas en los últimos 5 años.

$$\{ \langle t.nroEsc, t.barrio \rangle / ESCUELAS(t) \wedge \\ (\forall u) (INSPECCIONES-ESC(u) \wedge u.nroEsc = t.nro.Esc \wedge u.fecha > 1/1/2016 \\ \rightarrow u.puntaje > 50) \wedge \\ (\exists w) (INSPECCIONES-ESC(w) \wedge w.nroEsc = t.nro.Esc \wedge w.fecha > 1/1/2016) \}$$

c) Resolver en **Cálculo Relacional y en SQL**:

Devolver nombre de escuela, nombre de directora, fecha de inspección y puntaje, donde el puntaje obtenido fue el menor puntaje de todas las inspecciones a escuelas que fueron realizadas.

```
{ <t.nombre, u.nombre, i.fecha, i.puntaje> / ESCUELAS(t) ^ PERSONAL(u) ^  
  INSPECCIONES-ESC(i) ^  
  t.ciDirectora = u.CI ^ t.nroEsc = i.nroEsc ^  
  (∀w) ( INSPECCIONES-ESC(w) → w.puntaje >= i.puntaje)  
}
```

```
SELECT E.nombre, P.nombre, I.fecha, I.puntaje  
FROM ESCUELAS E JOIN PERSONAL P ON (ciDirectora = CI) NATURAL JOIN INSPECCIONES_ESC I  
WHERE I.puntaje =  
  (SELECT MIN(puntaje)  
   FROM INSPECCIONES-ESC)
```

Ejercicio 4

Parte 1

a) Falso.

T_1 : $rl_1(x), r_1(x), u_1(x), rl_1(y), r_1(y), u_1(y), c_1$

T_2 : $rl_2(z), r_2(z), u_2(z), c_2$

H: $l_1(x), r_1(x), u_1(x), rl_1(y), r_1(y), u_1(y), c_1, rl_2(z), r_2(z), u_2(z), c_2$

H es serializable (ya que es la historia serial) y T_1 no sigue 2PL ya que contiene la secuencia $rl_1(x) \dots u_1(x) \dots rl_1(y)$.

b) Falso.

T_1 : $wl_1(x), w_1(x), u_1(x), c_1$

T_2 : $rl_2(x), r_2(x), u_2(x), c_2$

H: $wl_1(x), w_1(x), u_1(x), rl_2(x), r_2(x), u_2(x), c_2, c_1$

T_1 y T_2 siguen 2PL básico, pero H no es recuperable ya que T_2 lee de T_1 y además confirma antes que T_1 .

c) Falso.

T_1 : $wl_1(x), w_1(x), u_1(x), c_1$

T_2 : $rl_2(x), r_2(x), u_2(x), c_2$

H: $wl_1(x), w_1(x), u_1(x), rl_2(x), r_2(x), u_2(x), c_2, c_1$

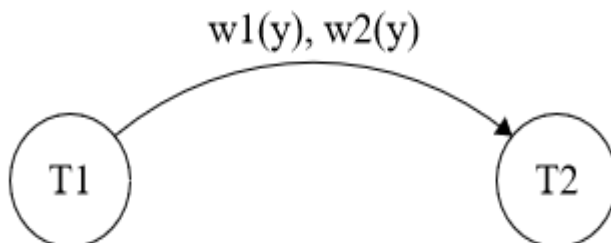
T_1 y T_2 siguen 2PL conservador, pero H no es estricta ya que T_2 lee de T_1 y además confirma antes que T_1 .

d) Falso.

El protocolo 2PL conservador, visto en el curso, asegura la ausencia de deadlocks.

Parte 2

H1

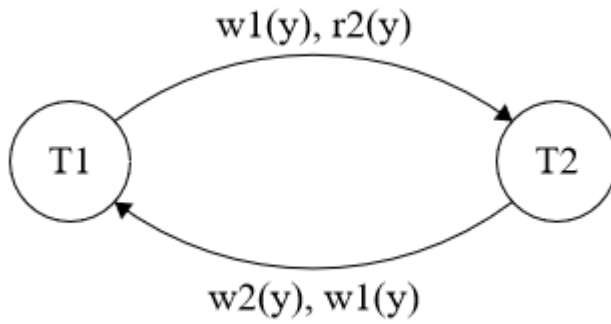


H1 es serializable porque el grafo de seriabilidad no tiene ciclos.

Es recuperable y evita abortos en cascada porque ninguna transacción lee de otra.

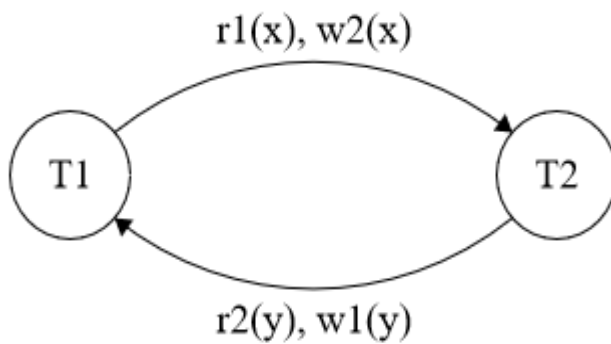
No es estricta porque T_2 escribe sobre T_1 antes de que T_1 haya confirmado.

H2



H2 no es serializable porque el grafo de seriabilidad tiene ciclos.
Es recuperable y evita abortos en cascada porque no se lee de transacciones sin confirmar.
No es estricta porque T_1 escribe sobre T_2 antes de que T_2 haya confirmado.

H3



H3 no es serializable porque el grafo de seriabilidad tiene ciclos.
Es estricta porque no se lee ni escribe de transacciones que no hayan confirmado, por lo tanto también evita abortos en cascada y es recuperable.