

FUNDAMENTOS DE BASES DE DATOS

Examen Diciembre 2011 - 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 (25 puntos).

Se desea modelar la realidad correspondiente a una cadena de hoteles.

De los hoteles se sabe que tanto la ciudad como el país al cual pertenece cada hotel son necesarios para identificarlo. Además, se conoce el nombre del hotel (que es único dentro de cada ciudad), sus teléfonos y su dirección de página web. Cada hotel tiene un conjunto de habitaciones con un número que las identifica dentro del mismo, pero este número no es único dentro de la cadena hotelera. Existen dos tipos de habitaciones. Las habitaciones dobles, que tienen al menos una cama matrimonial y las habitaciones simples que sólo tienen camas de una plaza. De las habitaciones dobles se sabe que algunas tienen una cama de una plaza extra, en este caso, interesa saber cuáles. De las simples, interesa registrar cuantas camas de una plaza hay en cada habitación.

La cadena hotelera cuenta con un grupo de clientes, y algunos de estos envían quejas de los servicios a los hoteles. Dichas quejas se registran como un texto, con la fecha correspondiente. Un cliente puede enviar más de una queja en una misma fecha.

De los clientes se registra su ci, su pasaporte (cualquiera de éstos los identifica), su nombre, todos sus teléfonos y un e-mail. Los clientes realizan reservas de las habitaciones y todas las fechas de reserva quedan registradas para saber que habitaciones son las más pedidas en las distintas temporadas.

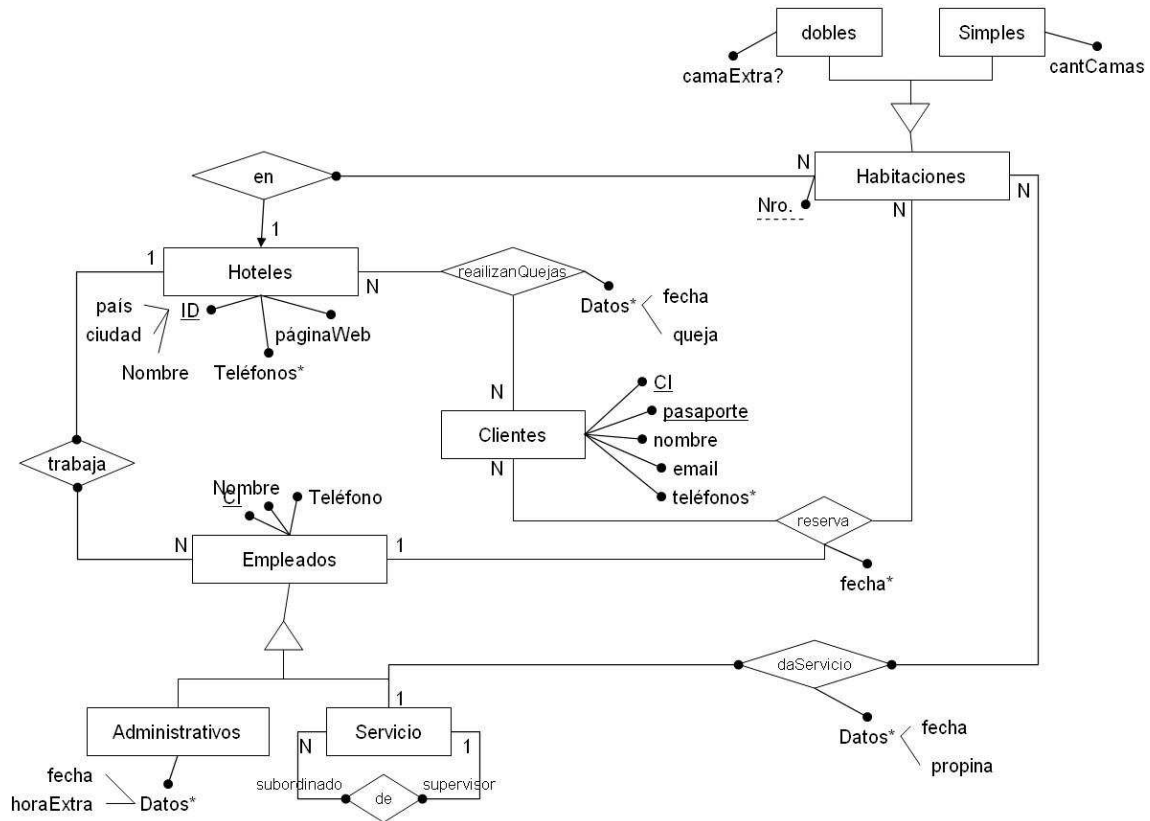
En los hoteles trabajan empleados y cada empleado trabaja en un único hotel de la cadena. De los empleados se conoce su ci, su nombre y un teléfono.

Por un lado están los empleados administrativos de los cuales se registra, para cada fecha, las horas extras que realizó. Además, interesa saber qué empleado administrativo registró cada reserva de habitación por parte de un cliente.

Por otro lado están los empleados de servicio, que se encargan de la limpieza de las habitaciones. Cada empleado de servicio se encarga de un conjunto de habitaciones y cada habitación es atendida por un único empleado. Por una cuestión de control, interesa registrar para cada fecha la propina recibida por el empleado en cada habitación. Dentro de los empleados de servicio, algunos son supervisores de otros. Todo empleado de servicio tiene un único supervisor y todo supervisor tiene al menos un empleado a su cargo.

Se pide: Modelo Entidad Relación completo.

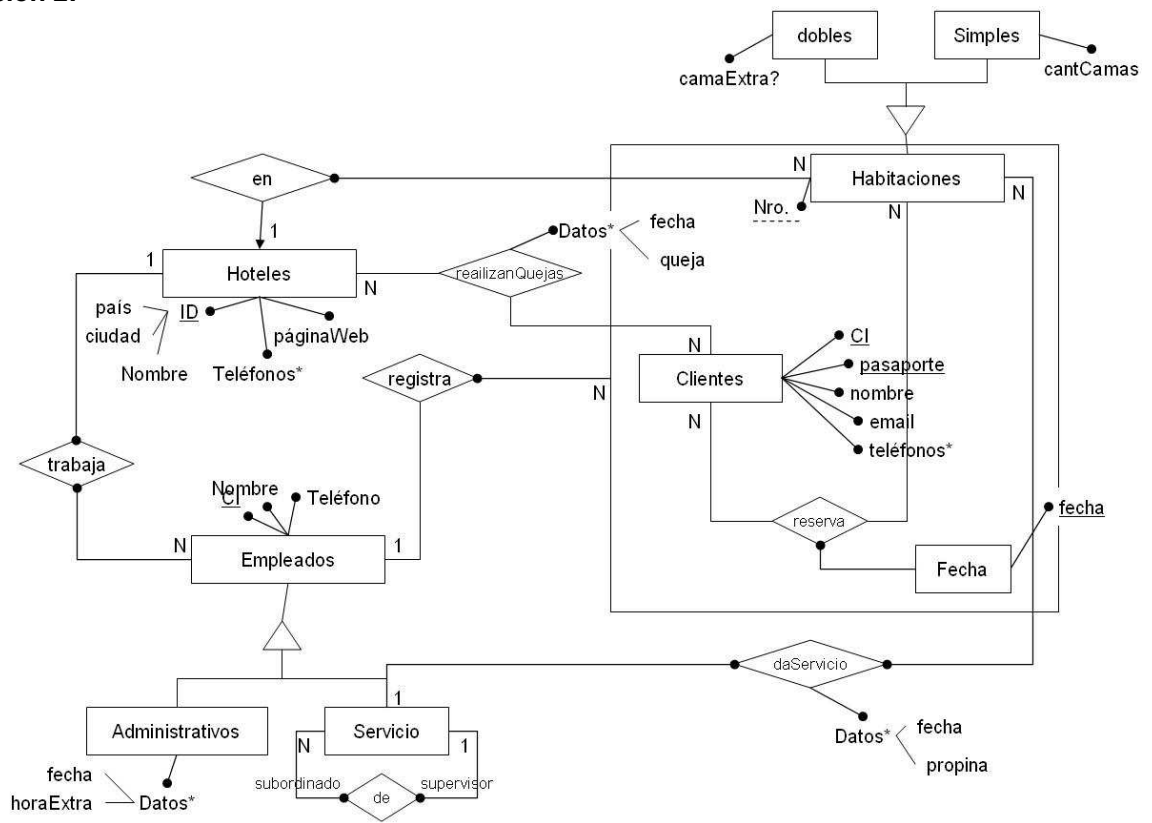
Solución 1:



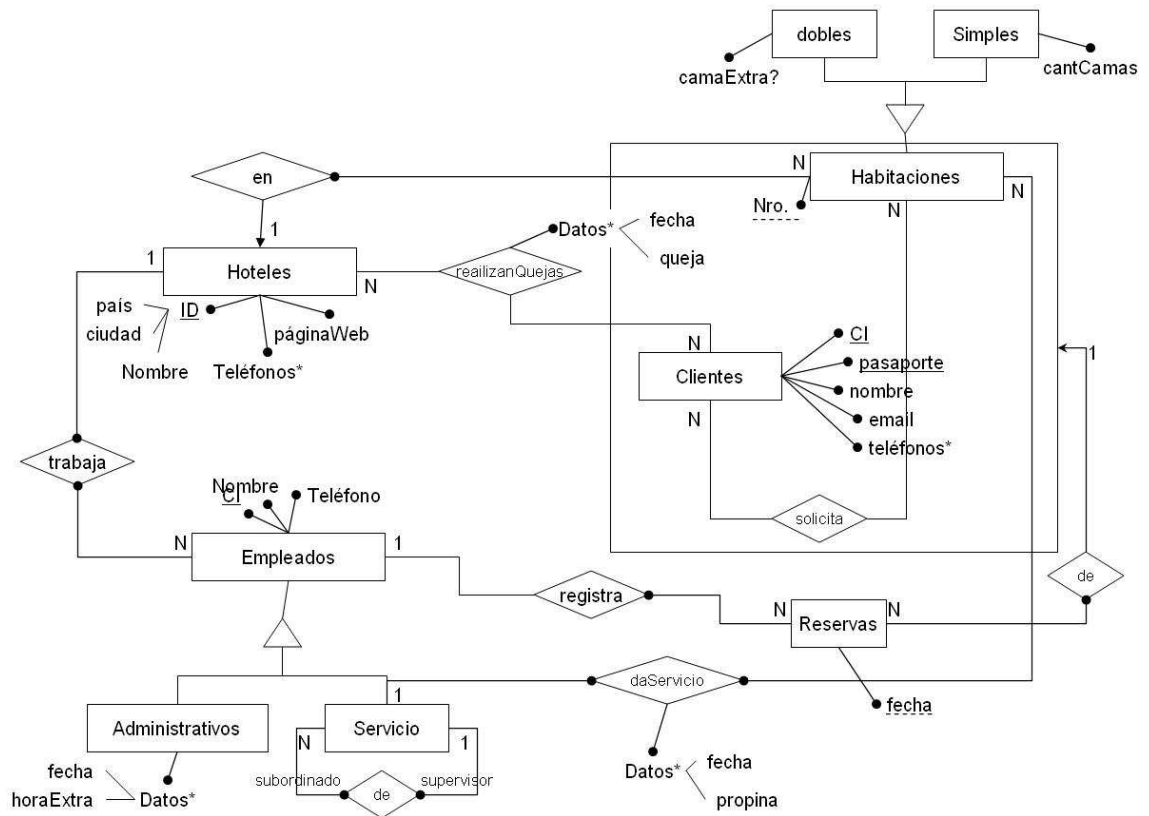
RNE:

- $\text{empleados} = \text{administrativos} \cup \text{servicio}$
- $\text{administrativos} \cap \text{servicio} = \emptyset$
- $\text{habitaciones} = \text{dobles} \cup \text{simples}$
- $\text{dobles} \cap \text{simples} = \emptyset$
- Un empleado registra reservas de las habitaciones del hotel donde trabaja:
 $(\forall e \in \text{Empleados}) (\forall h \in \text{Habitaciones}) (\forall c \in \text{Clientes}) (\text{reserva}(e,h,c) \rightarrow (\exists ho \in \text{Hoteles}) (\text{en}(ho,h) \wedge \text{trabaja}(ho,e)))$
- Un empleado da servicios en el hotel donde trabaja:
 $(\forall e \in \text{Empleados}) (\forall h \in \text{Habitaciones}) (\text{daServicio}(e,h) \rightarrow (\exists ho \in \text{Hoteles}) (\text{en}(ho,h) \wedge \text{trabaja}(ho,e)))$
- La relación de supervisión es irreflexiva y simétrica

Solución 2:



Solución 3:



Ejercicio 2 (15 puntos).

Dada la siguiente descripción de una realidad:

En una academia de inglés se quiere modelar en una base de datos la siguiente información. De los profesores interesa la cédula de identidad, el nombre, el teléfono, el correo electrónico y su nivel de formación. Los cursos tienen un nombre que los identifica, un nivel, y el público objetivo (niños, jóvenes o adultos). Los salones tienen un número que los identifica, el piso donde se encuentran y la capacidad. Por último existen los grupos. Cada grupo se identifica por el nombre de curso y el nombre de grupo (A, B, C, etc.), y se guarda la cantidad de estudiantes que tiene. Además, un grupo tiene asociados un conjunto de profesores y un conjunto de salones. El salón y el profesor en cada grupo van variando a lo largo del año, pudiendo tocarle a cualquier profesor en cualquier salón.

- Deducir las dependencias funcionales que se cumplen.
- Obtener una descomposición en 3NF con JSP aplicando el algoritmo visto en el curso, partiendo del esquema universal.
- Decir si la descomposición obtenida en la parte b) es con Preservación de Dependencias, justificando.
- Decir si se cumple alguna dependencia multivaluada en el esquema obtenido en la parte b), que no sea dependencia funcional ni multivaluada trivial.
- Llevar el esquema de la parte b) a 4NF.

SOLUCION:

a)

ci-prof → nom-prof, tel, mail, form

nom-cur → nivel, publico

salon → piso, cap

nom-cur, grupo → cant-est

b)

U (ci-prof, nom-prof, tel, mail, form, nom-cur, nivel, publico, nro, piso, cap, nom-cur, grupo, cant-est)

$F_{\text{minimal}} = \{$
 ci-prof → nom-prof
 ci-prof → tel
 ci-prof → mail
 ci-prof → form
 nom-cur → nivel
 nom-cur → publico
 salon → piso
 salon → cap
 nom-cur, grupo → cant-est }
 $\}$

F_{minimal} es minimal porque cumple las 3 propiedades:

- Las dfs tienen un solo atributo a la derecha.
- Ninguna df tiene atributos redundantes a la izquierda, ya que:

(nom-cur)+ = { nom-cur, nivel, publico }, NO SE CUMPLE: nom-cur → cant-est

(grupo)+ = { grupo }, NO SE CUMPLE: grupo → cant-est

- No hay dfs redundantes, ya que todos los atributos de los lados derechos de las dfs no

aparecen en el lado derecho de ninguna otra df, es decir que no los determina nadie más. Por lo tanto no puedo sacar ninguna df y que se sigan determinando a los mismos atributos.

Aplicando el algoritmo de 3NF visto en el curso, se obtienen las siguientes relaciones.

Aplicando el paso de generar una relación por cada grupo de dfs que tengan el mismo conjunto de atributos a la izquierda:

Profesores (ci-prof, nom-prof, tel, mail, form)

Cursos (nom-cur, nivel, publico)

Salones (salon, piso, cap)

Grupos (nom-cur, grupo, cant-est)

Aplicando el paso del algoritmo para asegurar que sea con JSP, agregamos una relación con una clave de la relación U:

La única clave de U es: (nom-cur, grupo, ci-prof, salon) porque:

ci-prof y salon tienen que estar en toda clave, porque no están a la derecha de ninguna df.

(nom-cur, ci-prof, salon)+ no contiene a todos los atributos

(grupo, ci-prof, salon)+ no contiene a todos los atributos

(nom-cur, grupo, ci-prof, salon)+ sí contiene a todos los atributos

El resto de los atributos no pertenecen a ninguna clave por no estar a la izquierda de ninguna df.

GruProfSal (nom-cur, grupo, ci-prof, salon)

Descomposición = {Profesores, Cursos, Salones, Grupos, GruProfSal }

c)

La descomposición obtenida en la parte b) es con preservación de dependencias porque el algoritmo aplicado me lo asegura.

d)

Por lo dicho en la letra: "Además, un grupo tiene asociados un conjunto de profesores y un conjunto de salones. El salón y el profesor en cada grupo van variando a lo largo del año, pudiendo tocarle a cualquier profesor en cualquier salón.", se cumple la siguiente dependencia multivaluada:

nom-cur, grupo ->> ci-prof

e)

Esquema de la parte b):

Profesores (ci-prof, nom-prof, tel, mail, form)

{ ci-prof → nom-prof, tel, mail, form }

Cursos (nom-cur, nivel, publico)

{ nom-cur → nivel, publico }

Salones (salon, piso, cap)

{ salon → piso, cap }

Grupos (nom-cur, grupo, cant-est)

{ nom-cur, grupo → cant-est }

GruProfSal (nom-cur, grupo, ci-prof, salon)

{ nom-cur, grupo ->> ci-prof }

Profesores, Cursos, salones y Grupos se encuentran en 4NF.

GruProfSal se encuentra en BCNF y no en 4NF porque tiene una dependencia multivaluada cuyo lado izquierdo no es superclave.

Aplicando el algoritmo de 4NF nos quedan las siguientes relaciones:

GruProf (nom-cur, grupo, ci-prof)
GruSal (nom-cur, grupo, salon)

La nueva descomposición queda:

Descomposición2 = {Profesores, Cursos, Salones, Grupos, GruProf, GruSal }

Ejercicio 3 (15 puntos)

Parte 1

Dado el esquema de relación R(A,B,C,D,E,G,H) y el conjunto de dependencias funcionales

$$F = \{ \begin{array}{l} AB \rightarrow GH, \\ E \rightarrow CB, \\ H \rightarrow AC, \\ G \rightarrow E, \\ D \rightarrow AH, \\ C \rightarrow AB, \\ B \rightarrow DG \end{array} \}$$

Y los siguientes subesquemas de R:

$$R1(A,B,E,C)$$

$$R2(D,B,E,G)$$

$$R3(H,E,D)$$

a) Hallar la proyección de F en cada uno de los subesquemas.

$$A^+ = \{A\}$$

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

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

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

$$\Pi_{R1}(F) = \{B \rightarrow ACE, E \rightarrow CAB, C \rightarrow ABE\}$$

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

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

$$\Pi_{R2}(F) = \{D \rightarrow BEG, B \rightarrow DEG, E \rightarrow DBG, G \rightarrow DBE\}$$

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

$$\Pi_{R3}(F) = \{H \rightarrow ED, E \rightarrow HD, D \rightarrow HE\}$$

b) Determinar si la descomposición de R en (R1,R2,R3) preserva las dependencias funcionales. Justificar la respuesta.

Se preserva la dependencias si $\Pi_{R1}(F) \cup \Pi_{R2}(F) \cup \Pi_{R3}(F)$ es equivalente a F.

Por construcción es necesario verificar si $F \subseteq (\Pi_{R1}(F) \cup \Pi_{R2}(F) \cup \Pi_{R3}(F))^+$

$$\Pi_{R1}(F) \cup \Pi_{R1}(F) \cup \Pi_{R1}(F) = J = \{ \begin{array}{l} B \rightarrow ACEDG, \\ E \rightarrow CABDGH \\ C \rightarrow ABE, \\ D \rightarrow BEGH, \\ G \rightarrow DBE, \\ H \rightarrow ED \end{array} \}$$

Las dependencias de F que no se proyectan directamente en los esquemas R1,R2 R3 son:
 $AB \rightarrow GH, H \rightarrow AC, D \rightarrow A$

$(AB)^+_J = \{A,B,C,D,E,G,H\}$, GH pertenecen a $(AB)^+_J$ por lo tanto $AB \rightarrow GH \in J^+$

$H_J^+ = \{H, E, D, A, B, C, G\}$, AC pertenecen a H_J^+ por lo tanto $H \rightarrow AC \in J^+$
 $D_J^+ = \{D, B, E, G, H, A, C\}$, $A \in D_J^+$ por lo tanto $D \rightarrow A \in J^+$

Por lo tanto todas las dependencias de F pertenecen a J^+ entonces la descomposición preserva las dependencias.

Parte 2

Dado el esquema relación R (A,B,C,D,E,G,H)

Se pide dar conjuntos de dependencias funcionales no vacíos y sin dependencias triviales de forma que se cumplan las siguientes situaciones. Justificar en todos los casos

c) Dar F_{3NF} tal que la máxima forma normal en que se encuentra R según F_{3NF} es 3NF.

$F_{3NF} = \{A \rightarrow B, B \rightarrow A\}$

$A^+ = \{A, B\} \neq R$, A no es superclave, entonces $A \rightarrow B$ viola la condición de BCNF.

(CDEGH) pertenecen a todas las claves por no pertenecer a los lados derechos de las dependencias funcionales.

$(CDEGH)^+ = \{C, D, E, G, H\} \neq R$ por lo tanto (CDEGH) no es superclave.

Busco claves con 6 atributos

$(ACDEGH)^+ = \{A, B, C, G, H, D, E\} = R$, por lo tanto ACDEGH es clave

$(BCDEGH)^+ = \{C, B, A, G, H, D, E\} = R$, por lo tanto BCDEGH es clave

Por lo tanto A y B son atributos primos entonces se cumple la condición de 3NF de que los lados derechos de las df sean atributos primos.

d) Dar F_{1NF} tal que la máxima forma normal en que se encuentra R según F_{1NF} es 1NF.

$F_{1NF} = \{C \rightarrow D\}$

C pertenece a todas las claves por no pertenecer a los lados derechos de las dependencias funcionales.

$C^+ = \{C, D\} \neq R$, por lo tanto C no es superclave pero es parte de todas las claves.

D no pertenece a las claves ya que solo pertenece a los lados derechos de las df, por lo tanto D es no primo.

Sea X una clave cualquiera de R según F_{1NF} , $C \in X$

$X \rightarrow D$ es una dependencia parcial de clave a un atributo no primo por lo tanto viola la condición de 2NF. R está en 1NF según F_{1NF} .

Ejercicio 4 (25 puntos).

Una empresa nacional dedicada a la administración de alquileres de propiedades mantiene su información en una base de datos que entre otros tiene los siguientes esquemas relación:

SUCURSAL (idsuc, depto, direccion, nomEncargado, horarioExtendido?)

Contiene la información de todas las sucursales de la empresa. Para cada una de ellas se conoce su identificador, el departamento donde se encuentra, la dirección en el mismo, el nombre del encargado actual y un indicador de si la sucursal atiende con horario extendido (por ej. fin de semana) o no.

PROPIEDAD (idprop, tipo, idpropietario, idsucursallng, fechaIngreso)

Contiene la información sobre las propiedades administradas. De cada una de ellas se conoce un identificador, el tipo de propiedad que es (campo, terreno, casa, etc.), un identificador del propietario y el identificador de la sucursal donde la propiedad ingreso a la empresa junto a la fecha en que ingreso.

INQUILINO (iding, nombre, dir)

Contiene los datos de los inquilinos de toda la empresa. Para cada uno de ellos se conoce un identificador, nombre y dirección del mismo.

ALQUILER (idprop, iding, fechaInicio, fechaFin)

Contiene la información de los alquileres realizados en la empresa. De cada alquiler se conoce: el identificador de la propiedad que se alquila, el identificador del inquilino que realiza el alquiler y las fechas de inicio y fin del mismo. Si el alquiler está vigente la fecha de fin es nula.

PAGOS (idprop, iding, idsuc, fechaPago, monto)

Contiene la información sobre los pagos que recibe la empresa por motivo de pagos de alquileres. Cualquier sucursal puede recibir pagos de alquileres de propiedades ingresadas en cualquier sucursal. De cada pago se conoce: la propiedad alquilada, el inquilino de la misma, la sucursal que recibe el pago, la fecha en que se realiza el pago y el monto del mismo.

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

- $\Pi_{idprop}(\text{ALQUILER}) \subseteq \Pi_{idprop}(\text{PROPIEDAD})$
- $\Pi_{iding}(\text{ALQUILER}) \subseteq \Pi_{iding}(\text{INQUILINO})$
- $\Pi_{idprop, iding}(\text{PAGOS}) \subseteq \Pi_{idprop, iding}(\text{ALQUILER})$
- $\Pi_{idsuc}(\text{PAGOS}) \subseteq \Pi_{idsuc}(\text{SUCURSAL})$

Resolver en álgebra relacional las siguientes consultas:

- a) Obtener el nombre del encargado, departamento en que se encuentra y dirección de las sucursales tales que todas las propiedades ingresadas a la misma son de tipo CAMPO.

Sucursales a las que han ingresado propiedades de tipo CAMPO:

$$A = \prod_{\text{idSuc}} (\sigma_{\text{tipo}=\text{CAMPO}} (\text{PROPIEDAD}))$$

Sucursales a las que han ingresado propiedades de tipo diferente a CAMPO:

$$B = \prod_{\text{idSuc}} (\sigma_{\text{tipo} \neq \text{CAMPO}} (\text{PROPIEDAD}))$$

$$\text{SOL} = \prod_{\text{nomEncargado, depto, direccion}} ((A - B) * \text{SUCURSAL})$$

- b) Obtener para cada propiedad manejada por la empresa que ha sido alquilada: el identificador de la propiedad, el nombre del último inquilino y la fecha de inicio de este último alquiler.

Alquileres tales que existe un alquiler posterior para esa propiedad:

$$A = \prod_{\$1, \$2, \$3} (\text{ALQUILER} \succ\prec \text{ALQUILER})_{\$1=\$5 \wedge \$3 < \$7}$$

Últimos alquileres para cada propiedad:

$$B = (\prod_{\$1, \$2, \$3} (\text{ALQUILER}) - A)$$

$$\text{SOL} = \prod_{\text{idprop, nombre, dir}} (B * \text{INQUILINO})$$

Resolver en cálculo relacional las siguientes consultas:

- c) Obtener la dirección y departamento en que se encuentran las sucursales que tienen horario extendido y que han alquilado todas las propiedades de tipo CASA ingresadas a la sucursal después del 1/7/2011.

$$\{ \text{s.direccion, s.depto} / \text{SUCURSAL}(s) \wedge \text{s.HorarioExtendido?} \wedge (\forall p)(\text{PROPIEDAD}(p) \wedge p.\text{tipo} = \text{CASA} \wedge p.\text{idsucursal} = s.\text{idsuc} \wedge p.\text{fechaIngreso} > 1/7/2011 \rightarrow (\exists a)(\text{ALQUILER}(a) \wedge a.\text{idprop} = p.\text{idprop})) \}$$

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

- d) Para cada sucursal y cada propiedad ingresada en la misma obtener idsuc, idprop y el monto total de pagos realizados por alquileres de esa propiedad (independientemente de la sucursal donde se realizó el pago) considerando únicamente las propiedades que han tenido por lo menos 3 inquilinos distintos.

```
SELECT pro.idSuc, pro.idprop, SUM(pa.monto)
FROM PROPIEDAD pro NATURAL JOIN PAGOS pa
WHERE pro.idprop IN
    (SELECT a.idprop
     FROM ALQUILER a
     GROUP BY a.idprop
     HAVING count(distinct a.idinq) >= 3)
GROUP BY pro.idSuc, pro.idprop
```

- e) Obtener el nombre de los encargados de las sucursales que han recibido la mayor cantidad de pagos.

```
SELECT nomEncargado
FROM SUCURSAL
WHERE idsuc IN
    (SELECT pa.idSuc
     FROM PAGOS pa
     GROUP BY pa.idsuc
     HAVING count(*) >=ALL
        (SELECT count(*)
         FROM PAGOS pa2
         GROUP BY pa2.idSuc
        )
    )
```

Ejercicio 5 (20 puntos).

Considere la siguiente función que implementa una transferencia en un banco.

```
function int transferir(int Origen,int Destino,float Monto)
{
    start_transaction;
    SO=read_saldo(Origen); //lee el saldo de la cuenta y lo carga en SO
    SON=SO-Monto;
    if(SON < 0 )
    then {
        rollback;
        return(-1);
    };
    write_saldo(Origen, SON); // graba el saldo de la cuenta según la variable SON
    SD=read_Saldo(Destino);
    SDN=SD+Monto;
    write_saldo(Destino,SDN);
    commit;
    return(0);
}
```

Suponiendo que al inicio de las ejecuciones la cuenta X tiene 2000 de saldo, la cuenta Y tiene 1500 de saldo y la cuenta Z tiene 500 de saldo.

- a) Construir las transacciones para las siguientes ejecuciones de esta función. No considere Locks. Use los nombre T1 y T2 respectivamente para la primer y segunda invocación de la función.

transferir(X,Y,100)
transferir(Y,Z,200)

- b) Reescriba las transacciones anteriores agregando Locks Binarios (simples, no read-write) de forma que las dos transacciones respeten 2PL Básico pero no estricto. Justifique.
- c) Construya una historia entrelazada de las transacciones anteriores (que se intercalen operaciones de read y/o write de las dos transacciones).
- d) Clasifique la historia que construyó antes de acuerdo a la recuperabilidad. Justifique.
- e) Puede decidir si la historia es serializable sin construir el grafo de seriabilidad? Justifique.

SOLUCION:

a)

T1: $r_1(X), w_1(X), r_1(Y), w_1(Y), c_1$
T2: $r_2(Y), w_2(Y), r_2(Z), c_2$

b)

T1: $l_1(X), r_1(X), w_1(X), l_1(Y), u_1(X), r_1(Y), w_1(Y), u_1(Y), c_1$
T2: $l_2(Y), r_2(Y), w_2(Y), l_2(Z), u_2(Y), r_2(Z), u_2(Z), c_2$

Si se analiza T1, sigue 2PL básico porque el primer Unlock ($u_1(X)$) aparece después del último Lock ($l_1(Y)$). Lo mismo sucede con T2. Sin embargo, para seguir 2PL estricto TODOS Unlock deberían aparecer después de la última operación (r o w) y esto no es así, ya que precisamente $u_1(X)$ en T1 aparece ANTES de los read.

c)

H1: $l_1(X), r_1(X), w_1(X), l_2(Y), r_2(Y), w_2(Y), l_2(Z), u_2(Y), l_1(Y), u_1(X), r_1(Y), w_1(Y), u_1(Y), c_1, r_2(Z), r_2(Z), u_2(Z), c_2$

d) Observar que T1 lee Y de T2 dado que el read que hace T1 sobre el item Y es la siguiente operación sobre ese item luego del write que hace T2. Sin embargo, T1 confirma antes que T2

por lo que no se respeta el orden del flujo de datos para confirmar. Por esto, esta historia, ni siquiera es recuperable.

Otras Observaciones.

Para que la historia fuera sólo recuperable, se hace necesario retrasar la confirmación de T1 hasta después de la de T2. Con la historia anterior, esto es un cambio menor:

H1: $l_1(X), r_1(X), w_1(X), l_2(Y), r_2(Y), w_2(Y), l_2(Z), u_2(Y), l_1(Y), u_1(X), r_1(Y), w_1(Y), u_1(Y), r_2(Z), r_2(Z), u_2(Z), c_2, c_1$

Para que la historia evitara abortos en cascada, T2 debería confirmar antes de que T1 leyera el ítem Y, quedando:

H1: $l_1(X), r_1(X), w_1(X), l_2(Y), r_2(Y), w_2(Y), l_2(Z), u_2(Y), l_1(Y), u_1(X), r_2(Z), r_2(Z), u_2(Z), c_2, r_1(Y), w_1(Y), u_1(Y), c_1$

d) Si, es posible afirmarlo. Tenemos la garantía de las historias son serializables porque lo garantiza el hecho de que las transacciones siguen 2PL.