Facultad de Informática
Universidad Complutense de Madrid

Examen -- Control 1º -- Ampliación de Bases de datos ------- Abril, 2021

APELLIDOS: **NOMBRE:**

1. *(3,5 puntos)* Dada la tabla **Apartamentos (Piso, Letra, Tamaño, Orientación, DNIPropietario, NombrePropietario)**, donde Piso es: 1º, 2º, 3º, etc., Letra: A, B, C, etc., Orientación: N, S, E, O; y el resto de atributos los valores obvios. El propietario puede tener varios apartamentos, cada apartamento tiene un propietario.
Se pregunta:

a) ¿Dado un conjunto de DFs de una tabla : Qué pasos se dan para saber si sobra una DF determinada?
SOLUCION:
3.1 Hago el cierre del lado izq (sin usar esa DF)
3.2 Aplicamos esta condición:
- Si en ese cierre está el atributo del lado dcho de la DF: esa DF sobra,
- quitarla y seguir haciendo 3.1 y 3.2 usando las DFs sin la que hemos quitado
- Si en ese cierre no está el atributo del lado dcho de la DF: esa DF se conserva
- seguir haciendo 3.1 y 3.2 usando las mismas DFs

b) ¿ Qué relación de significado hay entre los atributos de una Dependencia Funcional (DF) así X → Y?
SOLUCION: todas las filas con el mismo valor en X, tienen un valor en Y que será el mismo en todas esas filas
Por el significado de los conceptos.

c) ¿Qué relación de significado hay entre los atributos de la DF así: **Orientación → Tamaño**
SOLUCION: Que cualquier apartamento orientado en la misma dirección tiene el mismo tamaño

d) ¿Qué significa que haya una DF Parcial del la Clave Primaria (PK)?.
SOLUCION: que un atributo no dependa de la PK completa, sino solo de algún atributo de élla

e) Para que en **Apartamentos** haya una DF parcial: 1.- cúal debe ser la PK. 2).- Indica en qué atributos se da la DF Parcial y qué relaciones de significado debe haber entre esos atributos?
SOLUCION: PK = **(Piso, Letra)** una DF parcial es **Letra → Tamaño** Cuando todos los departamentos de la misma letra sean del mismo tamaño, pero tengan diferentes tamaños según la letra. Podría haber otra es **Letra → Orientación.**

f) ¿Qué significa que haya una DF Transitiva de la PK?
SOLUCION:
Que hay unos atributos Z que no dependen de la PK sino que dependen de otros atributos Y que sí dependen de la PK: PK → Y y Y→ Z

g) Para que en **Apartamentos** haya una DF transitiva de la PK: 1.- cuál debe ser la PK. 2).- Indica en qué atributos se puede dar la DF transitiva y qué relaciones de significado debe haber entre esos atributos?
SOLUCION: PK = **(Piso, Letra)** y la DF: **DNIPropietario → NombrePropietario** donde el Nombre no depende de la PK sino del DNI: el mismo DNI siempre tiene el mismo Nombre

h) ¿En qué Forma Normal está la tabla incluyendo lo asumido en f), es decir que tiene una DF transitiva (pero no hay DF parcial?
SOLUCION: en 2º FN

i) ¿Cómo se solucionan la DFs no deseadas? Aplícalo a la situación del apartado anterior ?
SOLUCION:
Descomponiendo en dos tablas :
Apartamentos (Piso, Letra, Tamaño, Orientación, DNIPropietario) y
Propietarios(DNIPropietario, NombrePropietario)

2. *(1,5 puntos)* Tenemos dos tablas normalizadas : **`FlotaCoches (Matrícula, Marca, Modelo, Color)`** y **`Revisiones (Matrícula, codTemaRevisión, puntuación)`**. Todos los coches tienen el mismo conjunto de TemasRevisión, unos cuarenta. Hay una consulta muy frecuente: Obtener todos los datos de un coche concreto y la puntuación para cada uno de sus TemasRevisión.

a) Indica qué técnica de desnormalización nos conviene aplicar para agilizar esa consulta y explica *cómo aplicarla*.
SOLUCION: Creamos una sola tabla que sustituye a las dos anteriores,
`FlotaCoches(Matrícula,Marca,Modelo,Color,codTemaRevisión1,. . ,codTemaRevisiónN)`
Donde **`codTemaRevisión-i`** es la puntuación para el código del Tema i.

b) ¿En qué puede perjudicar las actualizaciones?
SOLUCION: puede necesitar más memoria para acceder a la nueva tabla respecto a la **`Revisiones`**

3. *(5 puntos)* **Usando PLSQL y PLSQL Dinámico**

a) Escribe un procedimento **`confirmar_reservas (dni de un cliente)`**. Disponemos de la tabla **`reserva_vuelo(dni, cod_vuelo, fecha_res, num_plazas, confirmación)`** cuya PK es (dni, cod_vuelo). También tenemos una tabla para cada cliente con sus reservas más recientes: **`reservas_XXXXXXX (cod_vuelo, fecha_res, num_plazas)`** donde XXXXXX es el DNI es el cada cliente. El procedimiento debe hacer estos pasos:

1.- compruebo que la tabla reservas_XXXXXXX existe, si no existe da un mensaje y termina.
2.- Borrar reservas caducadas (de hace más de 100 días) de reservas_XXXXXXX
3.- Recorrer la tabla reserva_vuelo y para cada reserva de ese cliente:
  3.1.- Actualizar el atributo confirmación en tabla reserva_vuelo con el literal 'confirmado'
  3.2.- Insertar la reserva en la tabla reservas_XXXXXXX de las reservas de ese cliente

```
create o update procedure confirmar_reservas (DniCliente reserva_vuelo.dni_reserva%TYPE)

NombreTabla VARCHAR (50);

si_esta INT;

BEGIN
    -- 1.-  comprobar que la  tabla reservas_XXXXXXX existe

    NombreTabla  := RTRIM( UPPER('reservas_' || DniCliente));

  select count(table_name) into si_esta
    from tabs where
    table_name =  NombreTabla;

  IF (si_esta = 0) THEN
              DBMS_output.put_line('no existe Tabla');
  ELSE

   -- 2.- Borrar reservas caducadas (de hace más de 100 días) de   reservas_XXXXXXX

      Execute inmediate   'BEGIN Delete from reservas_' || DniCliente || '
                           where fecha_reserva + 100 < sysdate;  END;';

   -- 3.- Recorrer la tabla reserva_vuelo y para cada reserva de ese cliente

    for unaRes in    (select * from reserva_vuelo where DNI= DniCliente )
    LOOP

      -- 3.1.- actualizar  confirmacion_reserva en reserva_vuelo con 'confirmado'

        UPDATE reserva_vuelo
            set confirmacion = 'confirmado'
        where  unaRes.dni = dni and unaRes.cod_vuelo = cod_vuelo;

    -- 3.2.- insertar la reserva en la tabla reservas_XXXXXXX  de las reservas de ese
       cliente

       Execute inmediate   'BEGIN
         insert into reservas_' || DniCliente || ' VALUES
                           (unaRes.cod_vuelo,unaRes.fechas_res,unaRes.num_plazas);
         END;';


   -- 4.- Si hay más de 100 reservas en Reservas_XXXXXXX:
       insertamos una reserva en Reservas_XXXXXXX con su DNI

    END LOOP;

END IF;
END;
```

b) Escribe un trigger **`tr_confir`** que , después de cada vez que se crea una fila en la tabla **`reserva_vuelo`**, se activa. El trigger ejecuta el procedimento **`confirmar_reservas (dni)`** con el DNI adecuado. Si hay cualquier excepción, la captura dando un mensaje de aviso.

```
create or replace TRIGGER tr_confir
  AFTER INSERT ON reserva_vuelo
  FOR EACH ROW
DECLARE
  Mensaje VARCHAR(40);
BEGIN
  confirmar_reservas (:new.dni);

EXCEPTION
  WHEN OTHERS THEN
    DBMS_output.put_line('algo ha fallado');
END;
```