AMPLIACION de BASES DE DATOS

(Profesor : Héctor Gómez Gauchía)

Práctica 1 - SEMANA 1: Corregir y Mejorar el diseño de BDs con Dependencias Funcionales (DFs), Dependencias Multivaloradas (DMs) y Formas Normales (FNs)

Resultado:

- respuestas de cada apartado (todos en un solo fichero word),
- los ficheros .sql de lo que desarrolles en sql y plsql,
- Haz lista de dudas concretas sin resolver sobre tus respuestas para consultar con el profesor

Modo de entrega*:* No se Entrega

- Si terminas alguno de los apartados indicado en pizarra, avisa al profesor para puntuar la parte de participación en clase
- Los conceptos de esta práctica se evalúan en un examen de control (tal y como se indica en la Ficha Docente), cuya fecha se avisará a tiempo.

NOTA: En casa, puedes instalarte una Máquina Virtual con Oracle: Oracle Express, ver instrucciones en Instalar-MaqVirtual-en-Linux.pdf y otras instrucciones para instrucciones-instal-OracleDB11gR2Express-WIN.pdf

APARTADO 0.- Preparar tu cuenta de Oracle:

- Hacer los pasos indicados en `instrucciones-para-ABD.pdf`

APARTADO 1.- Reconoce DFs, FNs y consigue un BD en 3FN

Dadas estas dos relaciones que forman una BD:

```
reserva(dni_cliente_reserva, cod_vuelo_cliente_reserva, nomb_cliente_reserva,
   apell_cliente_reserva, telefono_cliente_reserva,cuenta_cliente_reserva,
   fecha_cliente_reserva, clase_cliente_reserva, nume_plaza_cliente_reserva,
   confirmacion_cliente_reserva, cod_ciudad_cliente_reserva,
   nombre_ciudad_cliente_reserva)

vuelo( cod_vuelo, cod_avion, plazas_vuelo_avion, origen_cod_aeropuerto_vuelo,
   destino_cod_aeropuerto_vuelo, cod_pais_aeropuerto_origen,
   cod_ciudad_aeropuerto_origen, nombre_ciudad_aeropuerto_origen,
   direcciónPostal_aeropuerto_vuelo_origen, modelo_avion)
```

Puede que necesites poner otros atributos en alguna tabla teniendo en cuenta que:
El nombre de una ciudad puede estar repetido en dos paises.
En el vuelo tenemos, entre otros atributos: el código del avión y la aerolínea.

Se pide hacer lo siguiente. Este apartado tiene varias fases:

Fase 1: En parejas de alumnos (escribe un documento con resultado) → *se da un tiempo determinado (se avisa)*

- Encuentra la lista de DFs que hay en las dos relaciones.
- Define las tablas necesarias para que toda la BD esté en 3FN (incluyendo las PKs)
- Define dos tablas válidas con más de dos atributos que estén en FN1, pero no estén en la siguiente FN2.
- Haz lo mismo : dos tablas que estén en FN2 paro no en FN3

Fase 2: Compara y discute los resultados con otra pareja → *se da un tiempo determinado (se avisa)*

- Encuentra DFs distintas entre las dos parejas
- Llegar a un acuerdo y escribe porqué es correcta o porqué falla.

Fase 3: Negocia cuál de las dos parejas ha acertado más y comunica al profesor vuestras conclusiones

SOLU: Estado Final de la BD en 3ªFN:

--- pais(cod_pais,nomb_pais) Pk: cod_pais
--- ciudad(cod_ciudad, cod_pais, nom_ciudad) PK: (cod_ciudad, cod_pais)
--- aeropuerto(cod_aeropuerto, cod_pais, cod_ciudad, nomb_aeropuerto, dire_aeropuerto) , PK : cod_aeropuerto
--- avion (cod_avion, modelo_avion, nume_plazas) PK : cod_avion
--- vuelo (cod_vuelo, origen, destino, aerolinea,cod_avion, plazas_vuelo, precio) PK: cod_vuelo
--- cliente (dni, nomb_cliente_reserva,apell_cliente_reserva, telefono_cliente_reserva, cuenta_cliente_reserva, cod_pais, cod_ciudad) PK: dni
--- reserva_vuelo(dni_reserva, cod_vuelo_reserva, fecha_reserva, clase_reserva, nume_plaza_reserva, confirmacion_reserva) PK: (dni, cod_vuelo)

APARTADO 2.- (conceptos básicos) Reconocer la clave primaria (CP), clave ajena (CA), claves candidatas (CC), dependencia funcional (DF) y dependencia multivalorada (DM), datos erroneos

Utilizando la BDejemplo.sql tal y como está definida, sin CP ni CA, (decide tu cuáles deben ser las CP en los apartados que lo necesites) realiza los siguientes apartados:

2.0- *Crea la BDejemplo en Oracle, ejecutando el script* ***BDejemplo.sql***
Revisa los comentarios dentro del script para entender el significado de los atributos.

2.1- *Describe lo indicado en cada apartado*:
2.1.x Qué resultado da la siguiente consulta. Indica si es correcto? Qué norma no cumple? Cómo sería correcta?
Consulta: quienes son y donde viven los Clientes activos (tabla *Cliente*) que además son morosos

```
Select Cliente.DNI, Cliente.Direccion
from Cliente, Moroso
where Cliente.Direccion = Moroso.Direccion;
```

SOLU: devuelve el DNI '0..1' que no está en Moroso: hay unión con pérdida información, por hacer la unión de dos tablas por un atributo que no es CP ni CC en ninguna de las dos tabla: Datos erróneos.

2.1.a En la tabla *Cliente*, cuándo podemos considerar CC el atributo Dirección? Describe la respuesta usando la relación que hay entre los valores de los clientes y los valores de las direcciones donde viven.
SOLU: Cuando el resto de atributos dependan de la dirección, sucede esto Cuando cada cliente viva en una Dirección distinta siempre. Cada fila tiene Dirección distinta.

2.1.b Si, en la tabla *Puesto*, ahora suponemos que cada título tiene el mismo sueldo para todos los clientes que tengan ese título. Indica la DF que provoca redundancia. Cómo se si es problemática?. Cómo la corriges aplicando la teoría? En qué FN queda? Indica los axiomas que cumple para estar en esa FN.
SOLU: Método de Razonamiento: tengo la DF Titulo→ Sueldo
- Pregunta original ¿Es necesario corregirla? →
→ Solo la corrijo si es DF mala:
→ ¿Cómo se si es buena o mala? → si ha quedado en 3ªFN o superior es buena
→ ¿cómo se si está en 3ª FN? → ¿Cumple los axiomas de 3ªFN?
→ Sí : entonces está en 3ªFN y puedo dejarla porque la DF es buena
→ NO: la DF es mala, debo corregirla ¿cómo? → Normalizando → ¿cómo? → Explicación completa:

Es necesaria corregirla porque hay una DF transitiva DNI → Titulo → Sueldo luego se queda en 2ª FN y debemos tener como mínimo 3ªFN para una calidad razonable
Se puede corregir normalizando, descomponiendo en dos tablas T1(DNI, Titulo) y T2(Titulo,sueldo) asumiendo que el cliente tiene solo un Título.

En desnormalización se puede considerar no descomponer por ser solo un atributo, y controlar la redundancia con un trigger que se activa cada vez que alguien cambia el sueldo a un puesto, ej.: secretarios.

2.1.c Si, en la tabla *Puesto*, ahora suponemos que un cliente tiene más de un Título, cual debe ser la CP? Corrige lo necesario en la BD dentro de Oracle y añade filas con datos inventados para que cumpla lo dicho.
SOLU: para tener identificación única de una fila necesito que la CP sean DNI y Titulo juntos.Sólo así todos los atributos dependen funcionalmente de la clave

2.2- *Qué relaciones se tienen que dar (en cuanto al significado de los atributos) en cada apartado, para que exista la DF indicada:*
2.2.X EJEMPLO: Si se cumple en *Empresa*: Cotización → Capital,
Solución: cuando las cotizaciones de las empresas fijan el capital que tienen. Es decir, que para dos Empresas con la misma Cotización tendrán el mismo Capital.

2.2.a - Tabla *Cliente*: NombreC → DNI . Además de decir las relaciones: comprueba en las filas creadas si se cumplen esas relaciones y Corrige las que no las cumplan, inventando datos.
SOLU: no puede haber dos clientes con el mismo nombre. Pero sí los hay: las filas del cliente 1,4, y 5 tienen el mismo nombre. Invento nombres únicos para cada cliente.
(Si la implemento puedo provocar esa DF poniendo el atributo NombreC como *unique*)

2.2.c - Tabla *Compras*: Tienda → NumF , donde NumF es el número de factura. Además de decir qué significa la DF, indica si es razonable esta DF y porqué?

SOLU:Si cada tienda tuviera un solo NumF : no es razonable que eso suceda

2.2.d - Tabla *Compras*: NumF → Tienda Además de decir las relaciones, indica si es razonable esta DF y porqué.

SOLU: un NumF corresponde a una tienda: solo si no hubiera, para dos tiendas, el mismo NumF, esto es, que la numeración de facturas fuera única para todas las tiendas. Esto no es razonable.

2.3- *Descubrir la Dependencia y haz un procedimiento para mantener la consistencia de datos*:
(se asume que los datos actuales de la BD son correctos)

2.3.a Qué dependencia y de qué clase hay en la tabla *Invierte*, en la siguiente situación:
- Cada Cliente invierte en varias Empresas, pero está obligado a invertir en los mismos Tipos en cada empresa que invierta; y todas las inversiones del mismo Tipo siempre deben tener la misma cantidad.

Esta situación provoca que:
- cada vez que un Cliente invierte en una empresa nueva, tiene que invertir en todos los tipos en los que había invertido antes.
- También sucede que cada vez que el Cliente quiera invertir en otro Tipo diferente a los que ya tenía, le obligan a invertir en todas las empresas (en las que ya tiene inversiones) con ese Tipo y la misma cantidad que tiene en otras inversiones ese mismo Tipo.

SOLU: Es una DM : DNI -->> NombreE , DNI -->> Tipo, cantidad
Cuando un cliente decide invertir en un Tipo nuevo, hay que añadir tantas filas como Empresas en las que tiene inversiones. Y cuando se borra un Tipo hay que recorrer toda la tabla borrando las inversiones de ese Tipo. Si hay que borrar una Empresa, hay que borrar las inversiones de cada Tipo.