**Tercer trabajo de bases de datos 2**

**Punto 1. (50%)**

Se van a tener dos archivos csv: **empleado.csv** y **departamento.csv** como los mostrados en los archivos adjuntos (archivos de ejemplo).

Cada uno de estos archivos va a tener muchas filas…

Use el comando **mongoimport** para cargar los datos de **empleado.csv** y **departamento.csv** en MongoDB.

Luego de cargar los datos haga lo siguiente:

**a) (25%)** Usando el operador **mapReduce** (puede ver una explicación, por ejemplo, en https://docs.mongodb.com/manual/core/map-reduce),programe lo necesario para obtener el *join* entre las colecciones de empleados y departamentos (las cargadas con **mongoimport**) usando este operador (el campo de *join* es **dep**, note que este campo está presente en los dos archivos).

**Nota**: Si lo requiere puede usar varios **mapReduce**, puede también usar colecciones auxiliares durante el proceso si las requiere…

**b) (25%)** Ahora use el comando **$lookup** (puede ver una explicación en <https://www.stackchief.com/tutorials/$lookup%20Examples%20%7C%20MongoDB>) y aplíquelo para obtener el *join* entre las mismas dos colecciones (la de empleados y la de departamentos).

Nota: La salida (estructura de los documentos resultantes) obtenida con el operador **$lookup** puede no ser idéntica a la que usted obtuvo mediante **mapReduce en el punto a)**. Lo importante es que en los resultados, tanto en el punto a) como en el b) se observen los resultados del *join*.

**Punto 2. (50%)**

**a) (15%)** En forma similar a como se conecta Java con Oracle, establezca una conexión de Java con MongoDB. Describa brevemente cómo es el proceso de conexión (que archivos hay que descargar (*driver*), como se establece la conexión y muestre un ejemplo “hola mundo” de cómo leer una colección de documentos e imprimir algunos de sus datos.

**b) (35%)** Se tienen dos tablas en Oracle así:

CREATE TABLE **factura**(

codigofac NUMBER(8) PRIMARY KEY,

codigovendedor NUMBER(8) NOT NULL,

codigocliente NUMBER(8) NOT NULL

);

CREATE TABLE **detalle**(

codigofac NUMBER(8) REFERENCES factura,

codigoproducto NUMBER(8) NOT NULL,

nrounidades NUMBER(8) NOT NULL,

valorunitario NUMBER(8) NOT NULL,

PRIMARY KEY(codigofac,codigoproducto)

);

Estas tablas se van a llenar con datos (*inserts*…) directamente en Oracle (por ejemplo, mediante SQL\*Plus), **NO** a través de Java.

Haga un programa en **Java** que haga lo siguiente:

El programa debe acceder a las dos tablas (factura y detalle) y obtener el total comprado por cada cliente (el total comprado de un cliente es tomar **todas** sus facturas y hacer una sumatoria de **todos** sus detalles, es decir, sumar nrounidades**\***valorunitario).

El programa Java se debe conectar a MongoDB y crear una colección de documentos llamada **totalescliente**, donde **cada documento** tiene esta estructura:

{ codigocliente: …

totalcomprado: …

}

En esta colección se deben insertar los totales calculados, creando un documento para cada cliente con su total.

Hacer lo mismo para los vendedores, es decir, calcular para cada vendedor su total vendido e insertar los resultados en una colección llamada **totalesvendedor**, donde **cada documento** tiene esta estructura:

{ codigovendedor: …

totalvendido: …

}

En resumen el **programa Java** debe tener tres opciones:

1. **Cargar datos**: debe leer los datos de las tablas factura y detalle y cargar los totales correspondientes en las colecciones totalescliente y totalesvendedor en MongoDB.
2. **Mostrar totales clientes**: debe leer y mostrar la colección totalescliente.
3. **Mostrar totales vendedores**: debe leer y mostrar la colección totalesvendedor.

**Fecha de entrega**: Informe impreso (explicando brevemente como se resolvió cada punto) y enviarlo también a *fjmoreno@unal.edu.co*: viernes 9 de agosto hasta las 12 medio día al inicio de clase. No se reciben trabajos luego de esta hora. **Imprima** los códigos del primer punto. No imprima el código Java. Por correo **SÍ** debe enviar TODOS los códigos. **Solo se califica lo enviado por correo hasta esa fecha**.