**Tercer Trabajo de Bases de Datos 2 (25%)**

**MongoDB, Oracle y Java**

En este trabajo se usará una base de datos Oracle para registrar las ventas diarias de supermercados. Se usará una base de datos en MongoDB para recopilar estadísticas sobre las ventas. **En clase se explica la conexión de Oracle con Java**. Usted debe consultar la conexión de MongoDB con Java.

La base de datos Oracle es así:

CREATE TABLE sucursal(

codigo NUMBER(8) PRIMARY KEY,

nombre VARCHAR2(20) UNIQUE NOT NULL

);

CREATE TABLE vendedor(

codigo NUMBER(8) PRIMARY KEY,

nombre VARCHAR2(20) NOT NULL

);

CREATE TABLE cliente(

codigo NUMBER(8) PRIMARY KEY,

nombre VARCHAR2(20) NOT NULL,

genero CHAR(1) CHECK (genero IN ('m ', 'f '))

);

--Note que el género puede ser nulo, por ejemplo, el cliente no quiso indicarlo…

CREATE TABLE producto(

codigo NUMBER(8) PRIMARY KEY,

nombre VARCHAR2(20) NOT NULL,

tipo VARCHAR2(20) NOT NULL,

marca VARCHAR2(20) NOT NULL

);

CREATE TABLE venta(

codventa NUMBER(15) PRIMARY KEY,

sucursal NUMBER(8) NOT NULL REFERENCES sucursal,

vendedor NUMBER(8) NOT NULL REFERENCES vendedor,

cliente NUMBER(8) NOT NULL REFERENCES cliente,

producto NUMBER(8) NOT NULL REFERENCES producto,

nro\_unidades NUMBER(10) NOT NULL CHECK(nro\_unidades > 0)

);

Estas tablas se llenan con datos por medio de la consola (SQL\*Plus) (**NO** se llenarán por medio del programa Java que se describe a continuación).

Cree un programa en **Java** que haga lo siguiente. El programa tiene una interfaz gráfica sencilla que tiene tres botones:

**(20%)** El **primer botón** se llama “Consultar Datos”; este botón tiene al lado una lista de selección donde el usuario elige la tabla que desea consultar (sucursal, vendedor, cliente, producto o venta). Si el usuario selecciona, por ejemplo, la tabla sucursal, entonces al pulsar el botón, el programa Java se conecta a la base de datos **Oracle** y se le deben mostrar al usuario (en la interfaz de Java) todos los datos de dicha tabla.

**(50%)** El **segundo botón** se llama “Calcular Estadísticas”. Al pulsar este botón, el programa Java se conecta a la base de datos Oracle y también a la base de datos de MongoDB. El programa Java genera entonces en la base de datos de MongoDB, **a partir de los datos de la base de datos Oracle**, las siguientes colecciones:

Colección Marcas: en esta colección cada documento JSON tiene la siguiente estructura:

* nombreMarca: nombre de la marca.
* misVentas: es un arreglo de documentos JSON, donde cada documento JSON tiene: el nombre de la sucursal, el tipo de producto y el total de unidades vendidas de ese tipo de producto en la sucursal para la marca correspondiente.
* grantotal: total de unidades vendidas por la marca considerando todas las sucursales.

Veamos un ejemplo concreto de **dos documentos** de la colección Marcas:

(por simplicidad, en estos ejemplos no se muestran las comillas de las cadenas de caracteres…).

{ nombreMarca: Colanta,

misVentas: [ {nomsucursal: Exito Robledo, TipoProd: Carnicos, TotalUni: 560},

{nomsucursal: Exito Envigado, TipoProd: Carnicos, TotalUni: 800},

{nomsucursal: Exito Robledo, TipoProd: Lacteos, TotalUni: 300},

{nomsucursal: Jumbo 65, TipoProd: Carnicos, TotalUni: 490}

],

grantotal: 2150

}

--Esto quiere decir, por ejemplo, que en el **Exito de Robledo** se vendieron un total de **560**

--unidades de productos **Carnicos** pertenecientes a la marca **Colanta**.

{ nombreMarca: Alpina,

misVentas: [ {nomsucursal: Exito San Antonio, TipoProd: Carnicos, TotalUni: 700},

{nomsucursal: Exito Envigado, TipoProd: Carnicos, TotalUni: 290},

{nomsucursal: Exito Envigado, TipoProd: Lacteos, TotalUni: 8000}

],

grantotal: 8990

}

Colección Generos: en esta colección cada documento JSON tiene la siguiente estructura:

* género: género (m, f o null)
* misVentas: es un arreglo de documentos JSON, donde cada documento JSON tiene: el nombre de la sucursal, el tipo de producto y el total de unidades vendidas de ese tipo de producto en esa sucursal para el género correspondiente.
* grantotal: total de unidades vendidas por el género considerando todas las sucursales.

Veamos un ejemplo concreto de **un documento** de la colección Generos para el caso del género m (masculino):

{ genero: m,

misVentas: [ {nomsucursal: Exito Robledo, TipoProd: Carnicos, TotalUni: 1000},

{nomsucursal: Exito Envigado, TipoProd: Carnicos, TotalUni: 900},

{nomsucursal: Jumbo 65, TipoProd: Lacteos, TotalUni: 490}

],

grantotal: 2390

}

Note que la colección Géneros tendrá como máximo tres documentos, uno por cada género (m, f y null).

Colección Vendedores: en esta colección cada documento JSON tiene la siguiente estructura:

* Codvendedor: código del vendedor.
* Nomvend: nombre del vendedor.
* misVentas: es un arreglo de documentos JSON, donde cada documento JSON tiene: el nombre de la sucursal y el total de unidades vendidas en esa sucursal por el vendedor correspondiente.
* grantotal: total de unidades vendidas por el vendedor considerando todas las sucursales.

Veamos un ejemplo concreto de **dos documentos** de la colección Vendedores:

{ codvendedor: 348,

nomvend: Ellen Barazzo,

misVentas: [ {nomsucursal: Exito Robledo, TotalUni: 765},

{nomsucursal: Exito Envigado, TotalUni: 894},

{nomsucursal: Jumbo 65, TotalUni: 5}

],

grantotal: 1664

}

{ codvendedor: 666,

nomvend: Chucky Damian,

misVentas: [ {nomsucursal: Exito San Antonio, TotalUni: 580},

{nomsucursal: Exito Envigado, TotalUni: 90},

{nomsucursal: Jumbo 65, TotalUni: 996}

],

grantotal: 1666

}

**(30%)** El **tercer botón** se llama “Consultar Estadísticas”; este botón tiene al lado una lista de selección donde el usuario elige la colección que desea consultar (Marcas, Géneros o Vendedores). Si el usuario selecciona, por ejemplo, la colección Marcas, entonces al pulsar el botón, el programa Java se conecta a la base de datos MongoDB y se le deben mostrar al usuario (en la interfaz de Java) todos los datos de dicha colección y al final de la impresión se debe mostrar la **suma** de todos los grantotales de la colección, o sea el super gran total de todos los grantotales.

**Notas adicionales:**

• Para enviar por email a *fjmoreno@unal.edu.co* el sábado 12 de Noviembre hasta la 1 pm. Solo se califican trabajos enviados a ese correo.

• No se reciben trabajos en hora posterior. No se reciben versiones “mejoradas”. No se califican

trabajos enviados “por accidente” a otros correos.

**• No se califica la belleza de las interfaces, solo se califica la funcionalidad.**

• Grupos máximo de tres personas.

• Los trabajos deben ser independientes entre los grupos. Trabajos copiados así sea en una SOLA parte se califican con 0 (cero) en su totalidad para todos los integrantes.

• El monitor les puede ayudar con los aspectos técnicos, pero su función no es hacerle el trabajo ni está autorizado para cambiar las condiciones del trabajo.

• Si trabaja con otros sistemas de bases de datos o lenguajes, así su trabajo funcione y sea

“espectacular”, el trabajo se califica con cero.

• Si encuentra errores en el enunciado por favor avisar lo más pronto posible para hacer las

correcciones respectivas.

• Por simplicidad en las pruebas, los datos de las marcas y de los tipos de productos se insertarán sin tildes, la primera letra en mayúscula y el resto en minúsculas.

• **Favor enviar en un solo mensaje todo el trabajo. Incluya en el mensaje los nombres de los**

**integrantes.**