

# MySQL -DB Bancaria

## Martín Heredia

### 1 - Sobre la base de datos:

Se encuentra compuesta por un total de 14 tablas (de las cuales dos de ellas son tablas de auditoria). Las mismas están dispuestas de manera tal que pueda individualizarse las responsabilidades de cada tabla, fragmentando la información para un mayor dinamismo a la hora de realizar modificaciones y actualizaciones sobre la base de datos.

Así mismo para entender la relación que poseen entre las tablas se adjunta el archivo .mwb que posee el diagrama entidad relación de la presente base de datos.

Para poder realizar la correcta creación de la db, se debe ejecutar el script bancaria-herediaV2.sql

### 2 - Sobre los scripts:

Una vez creada la base de datos con sus correspondientes tablas debemos empezar a nutrirla y darle información y funcionalidad, lo cual lo haremos con los scripts correspondientes.

#### A - Script de inserción de datos:

Es el primero que debe ejecutarse ya que le brindara los datos necesarios a la base de datos para que dejen de ser tablas vacías, y comiencen a demostrar una cierta coherencia en cuanto a la información que contienen las mismas y para ir entendiendo la lógica que poseen las mismas.

#### B - Script de creación de vistas:

Las vistas son tablas virtuales que nos sirven para facilitar ciertas consultas que podrían ser ejecutadas diariamente, esta base de datos posee 6 views.

### **B-1) Clientes\_cuentas\_radicación:**

Esta vista nos permite ver los datos del cliente, con sus saldos y donde tiene radicada la cuenta con la dirección de la sucursal. Utilizando inner joins, toma información de las tablas clientes, clasificación\_cuenta, moneda, sucursal, provincias, localidades, y las ordena por el id\_cliente de manera ascendente.

### **B-2) datos\_empleados:**

Esta vista nos permite ver los datos de los empleados, y en que sucursal se desempeñan. Esta toma la información de las tablas empleados, puesto\_laboral, sucursal, provincias, localidades, y las ordena por el id\_empleado.

### **B-3) detalle\_operaciones:**

Esta vista permite visualizar las transacciones ejecutadas, ordenadas por cliente.

### **B-4) operaciones\_aprobadas:**

Esta vista permite visualizar las transacciones con el estado "aprobado".

### **B-5) operaciones\_en\_proceso:**

Esta vista permite visualizar las transacciones con el estado "en proceso".

### **B-6) operaciones\_rechazada:**

Esta vista permite visualizar las transacciones con el estado "rechazada".

Estas 3 views toman datos de la tabla transacciones, utilizando datos de la tabla operaciones y estado, donde dependiendo el caso filtrará por la columna estado\_operación.

## **C - Script de creación de funciones:**

Las funciones son utilizadas para realizar cálculos simples o complejos, de resultados que necesitamos a la brevedad, la misma pueden recibir parámetros (o no), dependiendo que se busque realizar con ellas.

### **C-1) fn\_calcular\_antigüedad:**

La función calcular antigüedad, recibe como parámetro el ID del empleado, tomando como dato a calcular la diferencia entre su fecha de ingreso y el día en el cual se está realizando la consulta indicando la antigüedad en años.

### **C-2) fn\_calcular\_clientes\_por\_sucursal:**

**La función cantidad de clientes por sucursal, recibe como parámetro el ID de la sucursal, con dicho dato y a través de la función COUNT, realiza el conteo de clientes en la sucursal deseada.**

## **D - Script de creación de stored procedures:**

**Los procedimientos almacenados nos permiten modificar nuestras tablas o mostrarlas de diferentes maneras, y poder limitar incluso la información que devuelven, logrando un mayor dinamismo.**

### **D-1) sp\_ordenar\_tabla:**

**Este procedimiento nos permite ordenar una tabla según se necesite, se utiliza como primer parámetro el nombre de tabla, luego como segundo parámetro el nombre del campo por el cual deseamos ordenar, y por ultimo pasaremos asc o desc, según que orden deseemos.**

### **D-2) sp\_agregar\_eliminar\_cliente:**

**Este procedimiento nos permite agregar o eliminar un cliente, en caso de que como parámetro reciba la letra 'a', agregara el cliente con los datos que requiere el procedimiento para rellenar la tabla cliente.**

**En caso de que queramos eliminar un cliente, utilizaremos la letra 'd', y luego el ID del cliente, y con eso procederemos a la eliminar del registro.**

## **E - Script de creación de triggers:**

**Los triggers o 'gatillos', son elementos que se activan antes o después de determinada acción, estos capturan información que se este insertando, actualizando o eliminando, para llevar un control u auditoria de la base de datos.**

**En esta base de datos se crearon un total de 6 los cuales 3 se ejecutan sobre la tabla cuentas, y 3 sobre la tabla clientes, controlando las funciones de insert, delete y update. Esto es en pos de registrar movimientos que podrian editar los detalles de saldos o de datos personales de los clientes.**