

<b>EVALUACIÓN</b>	Obligatorio	<b>GRUPOS</b>		<b>FECHA</b>	Mayo 2019
<b>MATERIA</b>	Bases de Datos 2				
<b>CARRERA</b>	Analista en Tecnologías de Información / Analista Programador				
<b>CONDICIONES</b>	<p>- <b>Entrega:</b></p> <p>- <b>Puntos:</b> <u>Máximo:</u> 25. <u>Mínimo:</u> 0.</p> <p><b><u>IMPORTANTE</u></b></p> <p>- Los grupos deben estar conformados por hasta un máximo de dos personas.</p> <p>- Inscribirse (sacar la "<u>boleto de entrega</u>").</p>				

Sobre el modelo del Obligatorio 1

Sucursal(IdSucursal,DscSucursal,DirSucursal,MailSucursal,TelSucursal)

TipoCuenta(IdTipo,DscTipo)

Moneda(IdMoneda,DscMoneda,SimboloMoneda)

Cliente(IdCliente,NombreCliente,TipoCliente,TelCliente,MailCliente)

Cuenta(IdCuenta,IdTipo,IdMoneda,IdSucursal,IdCliente)

Movimiento(IdMovim,FchMovim,TipoMovim,IdCuenta,ImporteMovim)

Transferencia(IdTransfer,FchTransfer,IdMovim,TipoTransfer,CtaDestino,BancoDestino,StatusTransfer)

---

**Se pide**

1. Crear procedimientos o funciones según corresponda para:

- a. Crear un procedimiento almacenado '*SaldosDeCuentaCliente*' que reciba como parámetros un número de cuenta y un rango de fechas y retorne por parámetros de salida el saldo anterior a la fecha inicial del rango y el saldo actual a la fecha final del rango.
- b. Agregar a la tabla Cuentas una campo *SaldoCuenta* y crear un procedimiento almacenado '*generarSaldos*', que permita cargar el saldo de la cuenta de acuerdo a los movimientos que tiene.
- c. Implementar una función '*verSaldoCuenta*', que reciba como parámetros una cuenta y una fecha y retorne el saldo de dicha cuenta a la fecha indicada, no puede utilizar el campo *SaldoCuenta*.
- d. Crear una función '*maximoSaldoCliente*' que dado un cliente y una moneda retorne el máximo saldo que en el corriente año tuvo dicho cliente para dicha moneda.
- e. Crear un procedimiento o función '*sobregiroClienteUSD*', según corresponda, que reciba un *idCliente* y retorne *nombreCliente* e importe de sobregiro en dólares de dicho cliente al día de hoy.

(10 puntos)

2. Crear disparadores necesarios para realizar las siguientes actividades (considerar acciones múltiples para la implementación de los triggers):

- a. Cuando se ingrese un movimiento de tipo Transferencia, realizar un disparador que registre dicha transferencia en la tabla correspondiente, si la cuenta destino de la transferencia es una cuenta de otro banco (Externa), que registre como número de cuenta 99999, como nombre de banco 'Banco Externo' y como estado 'Auditoria'.
- b. Crear un disparador que al modificarse el importe de un movimiento deje un registro en una tabla de auditoría, esta tabla debe tener la siguiente estructura:

*Auditoria(idAudit,fchAudit,idMovim,idCliente,NombreCliente,ImporteAnterior,ImporteActual)*

El campo *idAudit* debe ser autoincremental y *fchAudit* también debe registrar la hora

- c. Mediante el uso de un disparador, no permitir ingresar un Movimiento de *Salida* de una cuenta que no tenga saldo.
- d. Mediante un disparador, no permitir crear una nueva cuenta si el cliente ya tiene una cuenta en la misma moneda y en la misma sucursal.
- e. Implementar un disparador que controle el borrado de una sucursal, para permitir el mismo, dicho disparador debe "mover" antes todas las cuentas a la sucursal más antigua del banco (obtener la sucursal más antigua de acuerdo a los movimientos).

(10 puntos)

3. Crear las siguientes Vistas:

- a. crear una vista que muestre la cantidad de depósitos que tiene el banco para cada uno de los meses del año actual. La vista debe retornar la siguiente estructura:

Mes	Cant_Depo
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

- b. crear una vista que muestre para cada cliente la cantidad de movimientos que ha realizado discriminado por tipo de movimiento y la fecha del último movimiento realizado en cada uno de esos tipos de movimientos. En el resultado deben aparecer todos los clientes.

(5 puntos)

**Se debe entregar:**

- Script: *Procedimientos.sql* con los procedimientos almacenados
- Script: *Funciones.sql* con las funciones solicitadas
- Script: *Disparadores.sql* con el código de cada disparador implementado
- Script: *Vista.sql* con la vista pedida

**Consideraciones generales:**

- Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.
- La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.

3. Durante la última semana los docentes no contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.