

Universidad ORT Uruguay

Facultad de Ingeniería

Bases de Datos 1

Obligatorio - M3A

Alejandro Durán – 176652
Santiago Teske – 188160

24/06/2021

Ejercicio 1

Obtener la cédula y el nombre de los clientes más antiguos, que tengan cuenta en la sucursal Centro.

Solución en SQL:

```
SELECT cliente.cedula, cliente.nombre
FROM cliente
--unimos cliente con cuenta a través de las cedulas
JOIN cuenta ON cuenta.cedula = cliente.cedula
WHERE cuenta.nombre = 'CENTRO'
AND fechaalta IN (SELECT MIN(fechaalta)
                  FROM cliente
                  JOIN cuenta ON cuenta.cedula = cliente.cedula
                  WHERE cuenta.nombre = 'CENTRO')
```

Para esta consulta se realizo un JOIN entre las tablas CLIENTE y CUENTA ya que el nombre de los clientes pertenece a la tabla CLIENTE mientras que la información relativa a la sucursal de la cuenta y su fecha de apertura pertenecen a la tabla CUENTA. Luego de eso filtramos la tabla obtenida para aquellos valores donde la sucursal fuera CENTRO para finalmente verificar que la fecha de alta se encontrara dentro de los valores mas antiguos.

Solución en álgebra relacional:

Tablas: CUENTA, CLIENTE

fechaDeAltaReciente <- (ρ CUENTA (c1) |X| c1.FechaAlta > c2.FechaAlta ρ CUENTA (c2))

-- Compara los elementos de la tabla CUENTA obteniendo aquellos con fecha de alta más reciente.

fechaDeAltaAntigua <- σ (c.Nombre ="CENTRO" ∧ (c.FechaBaja="NULL") (ρ CUENTA (c) - fechaDeAltaReciente))

--Muestra la tabla CUENTAS con las cuentas más antiguas.

π (Cedula, Nombre) (ρ CLIENTE (cli) |X| cli.Cedula=fdaa.Cedula (ρ fechaDeAltaAntigua (fdaa))

Ejercicio 2

Obtener todos los datos de los clientes que recibieron depósitos en dólares o depósitos en pesos mayores a mil unidades durante el mes anterior, pero que no recibieron depósitos en ambas monedas mayores a dicho monto en ese periodo.

Solución en SQL:

```
SELECT DISTINCT cliente.*
FROM cliente
--movimiento no tiene relación con cliente, tiene relación
con cuenta y cuenta con movimiento
JOIN cuenta ON cuenta.cedula = cliente.cedula
JOIN movimiento ON movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
WHERE movimiento.importe > 1000
AND fecha BETWEEN
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -2)) + 1
        AND
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -1))
AND cliente.cedula NOT IN
(
```

```
SELECT cliente.cedula
FROM cliente
JOIN cuenta ON cuenta.cedula = cliente.cedula
JOIN movimiento ON movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
WHERE cuenta.moneda = 'PESOS'
AND (fecha BETWEEN
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -2)) + 1
        AND
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -1)))
AND movimiento.importe > 1000
```

```
INTERSECT -- UNO LO QUE TIENEN MOVIMIENTO EN PESOS CON LOS
QUE TIENEN EN DOLARES
```

```
SELECT cliente.cedula
FROM cliente
JOIN cuenta ON cuenta.cedula = cliente.cedula
```

```

JOIN movimiento ON movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
WHERE cuenta.moneda = 'DOLARES'
AND movimiento.importe > 1000
AND fecha BETWEEN
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -2)) + 1
        AND
        LAST_DAY (ADD_MONTHS (TRUNC (SYSDATE), -1))
)

```

Para esta consulta realizamos un JOIN entre CLIENTE y CUENTA para mas tarde realizar otro entre CUENTA y MOVIMIENTO, de esa forma logramos acceder a todos los datos necesarios para resolver la consulta. En la primera etapa buscamos obtener la tabla con aquellos clientes con movimientos donde el importe fuera mayor a 1000 y la fecha coincidiera con el mes anterior. En la segunda, se busca obtener la tabla de los clientes que recibieron importes mayores a 1000 pesos durante el mes anterior, luego lo mismo pero en el caso de aquellos que recibieron dólares para posteriormente realizar la intersección de ambas tablas con el fin de llegar a los clientes que realizaron movimientos en pesos y dólares durante el mes anterior. En la tercera etapa finalmente procedimos a calcular la diferencia entre la tabla de clientes que recibieron importes mayores a 1000 unidades en el mes anterior y aquellos que recibieron movimientos en pesos y dólares para llegar así al resultado solicitado.

Casos de prueba:

Es posible acceder a los casos de prueba para esta consulta a través del siguiente [vínculo](#) o se podrá encontrar en la carpeta “Casos de Prueba” bajo el nombre “CPcons2.pdf”.

Solución en álgebra relacional:

Tablas: CLIENTE CUENTA MOVIMIENTO

clientesXcuentaXmovs <- π (cl.Cedula, cl.Nombre, cl.FechaNacimiento) ((p CLIENTE cl) |X| cl.cedula = ct. cedula (p CUENTA ct) ((p MOVIMIENTO mv) |X| mv.nombre = ct.nombre \wedge mv.numero=ct.numero))

--Muestra la información de los clientes combinada con cuenta y movimiento.

clientesMovValidos <- σ (clientesXcuentaXmovs.Fecha >="01/05/2021" \wedge clientesXcuentaXmovs.Fecha<="30/05/2021" \wedge clientesXcuentaXmovs.Importe >1000 (clientesXcuentaXmovs))

-- Muestra la información de los clientes que recibieron depósitos durante el mes anterior.

cliMovP <- σ (clientesMovValidos.Moneda= "Pesos" (clientesMovValidos))

cliMovD <- σ (clientesMovValidos.Moneda= "Dolares" (clientesMovValidos))

--Muestra la informacion de los clientes que recibieron ingresos mayores a 1000 en pesos y dólares respectivamente.

clientesBuscados <- (clientesMovValidos - (cliMovP \cap cliMovD))

--Devuelve la información de los clientes que realizaron movimientos válidos en una sola moneda.

Ejercicio 3

Obtener todos los datos de las cuentas que registraron más de un egreso de fondos mayor a 25.000 durante la primera quincena de marzo de 2021. Considerar solamente cuentas en pesos que no hayan tenido ingresos de fondos en el mismo período.

Solución en SQL:

```
--cuenta 1 tiene todos los datos
SELECT *
FROM cuenta cuenta1
WHERE cuenta1.moneda = 'PESOS'
AND EXISTS
(
--cuenta 2 usa solo las keys
SELECT cuenta2.numero, cuenta2.nombre
FROM cuenta cuenta2
JOIN movimiento ON movimiento.nombre = cuenta2.nombre AND
movimiento.numero = cuenta2.numero

WHERE movimiento.importe < -25000
      AND fecha BETWEEN '1/03/2021' AND '15/03/2021'
      --esto es lo que relaciona cuenta 1 con cuenta 2
      AND cuenta1.nombre = cuenta2.nombre
      AND cuenta1.numero = cuenta2.numero

      GROUP BY cuenta2.nombre, cuenta2.numero

      HAVING COUNT(*) > 1
)
AND NOT EXISTS
(
```

```

--LISTO LAS CUENTAS QUE TIENEN MOV POSITIVOS EN ESE
PERIODO
SELECT cuenta3.numero, cuenta3.nombre
FROM cuenta cuenta3
JOIN movimiento ON movimiento.nombre = cuenta3.nombre AND
movimiento.numero = cuenta3.numero

WHERE movimiento.importe > 0
AND fecha between '1/03/2021' and '15/03/2021'
--esto es lo que relaciona cuenta 1 con cuenta 3
AND cuenta1.nombre = cuenta3.nombre
AND cuenta1.numero = cuenta3.numero
)

--considerar cuentas en pesos que no hayan tenido ingresos en
el mismo periodo

```

En esta consulta dividimos el problema en tres partes, la primera encargada de mostrar el resultado general, es decir, toda la información relativa a las cuentas en pesos, y las otras dos como sub-partes de la anteriormente mencionada. A partir de ese resultado buscamos la existencia de cuentas donde se hubieran registrado más de un egreso mayor a 25000 en la primera quincena de marzo de 2021. Finalmente la última condición devuelve los valores que cumplen con la no existencia de importes positivos en dicho periodo.

Solución en álgebra relacional:

Tablas: CUENTA MOVIMIENTOS

ingresosEnQuincena <- σ (m1.Fecha >= "01/03/2021" \wedge m1.Fecha <= "15/03/2021" \wedge m1.Importe > 0 (p MOVIMIENTOS (m1)))

--Muestra los movimientos que tuvieron ingresos durante la primera quincena de marzo.

sinIngresosEnQuincena <- (p MOVIMIENTOS (m) - ingresosEnQuincena)

--Devuelve la tabla de movimientos que no tuvieron ingresos durante la quincena.

egresosEnQuincena <- σ (m2.Fecha >= "01/03/2021" \wedge m2.Fecha <= "15/03/2021" \wedge m2.Importe < -25000 (p MOVIMIENTOS (m2)))

--Tabla de egresos mayores a 25k durante la primera quincena.

movimientosQueCumplen <- (sinIngresosEnQuincena \cap egresosEnQuincena)

--Devuelve la tabla de movimientos que no tuvieron ingresos durante la quincena pero que sí tuvieron egresos mayores a 25000.

cuentaPesos <- σ (c1.Moneda = "PESOS" (ρ CUENTA (c1)))

--Muestra todas las cuentas en pesos.

cuentasQueCumplen <- (ρ cuentaPesos (cp) |X| (ρ movimientosQueCumplen (mqc)))

--Muestra las cuentas en pesos que no tuvieron ingresos en la quincena pero que sí tuvieron egresos mayores a 25000

--Compara y se queda con aquellos resultados de movimientos que se repitan más de una vez, muestra todos los atributos de CUENTA.

π (Nombre, Numero, Cedula, Moneda, FechaAlta, FechaBaja) (ρ (cuentasQueCumplen) cc1 |X| cc1.Nombre = cc2.Nombre \wedge cc1.Numero = cc2.Numero ρ (cuentasQueCumplen) cc2)

Ejercicio 4

Obtener la cédula, el nombre de los clientes, sus números de cuenta y sucursal, de aquellos clientes que recibieron depósito de todas las sucursales diferentes a la suya durante el año anterior.

Solución en SQL:

```
SELECT cliente.cedula, cliente.nombre, cuenta.numero
,cuenta.nombre
FROM cliente
JOIN cuenta ON cuenta.cedula = cliente.cedula
WHERE
NOT EXIST
--NO EXISTE UNA SUCURSAL QUE NO ESTE EN LOS MOVIMIENTOS
(--SUCURSALES TOTALES - LAS DEL CLIENTE
    SELECT sucursal.nombre
    FROM sucursal
    MINUS
    --sucursales del cliente
    SELECT cuenta1.nombre
    FROM cuenta cuenta1
    WHERE cuenta1.cedula = cuenta.cedula

    MINUS

    --sucursales desde donde recibio depositos
```

```

SELECT movimiento2.nombre
  FROM movimiento
    JOIN cuenta cuenta2 ON cuenta2.numero = movimiento.numero
AND cuenta2.nombre = movimiento.nombre
    JOIN movimiento movimiento2 ON
movimiento.idmovimientoreferencia = movimiento2.idmovimiento
  WHERE movimiento.nombre <> movimiento2.nombre
    AND cuenta2.numero = movimiento.numero AND cuenta2.nombre
= movimiento.nombre
    AND cuenta2.cedula = cuenta.cedula
    AND EXTRACT(YEAR FROM movimiento.fecha) = (SELECT
EXTRACT(YEAR FROM sysdate) - 1 FROM DUAL )
)

```

En este caso tomamos por un lado la tabla CLIENTE unida a CUENTA a través de un JOIN ya que CUENTA nos da la vinculación entre CLIENTE y MOVIMIENTO. Luego, realizamos la diferencia entre toda la lista de sucursales menos aquellas en donde el cliente comparado tiene cuenta, eso da como resultado la lista de sucursales en las que el cliente no tiene cuenta. Posteriormente realizamos la diferencia entre la tabla de sucursales donde el cliente no tiene cuenta y la tabla de sucursales desde donde recibió depósitos, dando como resultado la tabla de sucursales desde las cuales el cliente recibió depósitos y no tiene cuenta. Finalmente unimos esta tabla a la condición de no existencia del principio obteniendo a los clientes que recibieron depósitos de sucursales distintas a la suya.

Solución en cálculo relacional:

TABLAS: CLIENTE, CUENTA, MOVIMIENTO, SUCURSAL

$$\{ \text{cli.Cedula, cli.Nombre, cta.Numero, cta.Nombre} / \text{CLIENTE}(\text{cli}) \wedge (\exists \text{cta}) \text{CUENTA}(\text{cta}) \wedge \text{cli.cedula} = \text{cta.Cedula} \wedge$$

$$((\forall \text{suc}) \text{SUCURSAL}(\text{suc}) \rightarrow (\text{suc.Nombre} = \text{cta.Nombre})) \wedge$$

$$((\exists \text{mov1}) \text{MOVIMIENTO} \text{mov1} \wedge \text{mov1.Fecha} \geq "1/01/2020" \wedge \text{mov1.Fecha} \leq "31/12/2020") \wedge \text{mov1.Numero} = \text{cta.Numero}) \wedge$$

$$(\exists \text{mov2}) \text{MOVIMIENTO} \text{mov2} (\text{mov2.Fecha} \geq "1/01/2020" \wedge \text{mov2.Fecha} \leq "31/12/2020") \wedge \text{mov2.Nombre} = \text{suc.Nombre}) \wedge (\text{mov1.Importe} = \neg (\text{mov2.Importe})) \wedge \text{mov1.IdMovimiento} = \text{mov2.IdMovimientoReferencia}) \wedge \neg (\text{mov1.IdMovimiento} = \text{mov2.IdMovimiento}) \}$$

Ejercicio 5

Obtener datos de los movimientos donde existan para la misma cuenta movimientos por el mismo concepto pero importe opuesto durante el mismo mes y no existan para ese cliente movimientos con el mismo concepto y mismo importe durante ese mismo mes.

Solución en SQL:

```
SELECT DISTINCT movimiento.*
FROM movimiento
JOIN movimiento movimiento2 ON movimiento.numero =
movimiento2.numero AND movimiento.nombre = movimiento2.nombre
WHERE movimiento.concepto = movimiento2.concepto
AND movimiento.importe = (-1)*movimiento2.importe
AND EXTRACT(MONTH FROM movimiento.fecha) = EXTRACT(MONTH FROM
movimiento2.fecha)
AND NOT EXISTS-- movimiento con campos iguales pero id
diferente
(
    SELECT movimiento.*
FROM movimiento movimiento3
WHERE movimiento.numero = movimiento3.numero AND
movimiento.nombre = movimiento3.nombre
AND movimiento.concepto = movimiento3.concepto
AND movimiento.importe = movimiento3.importe
AND EXTRACT(MONTH FROM movimiento.fecha) = EXTRACT(MONTH FROM
movimiento3.fecha)
AND movimiento.idmovimiento <> movimiento3.idmovimiento
)
```

En esta consulta comenzamos por realizar un JOIN entre dos instancias de la tabla MOVIMIENTO, obteniendo aquellos movimientos con mismo concepto, realizados por cuentas coincidentes en un mismo mes, listando a los importes y a sus opuestos. En la segunda instancia obtuvimos aquellos movimientos donde además de cumplirse las anteriores características el importe se repetía (es decir, no era opuesto) pero donde su idMovimiento era diferente. Finalmente, a través de la sentencia NOT EXIST se obtienen aquellos movimientos donde existe otro de importe opuesto como pide la consigna.

Casos de prueba:

Es posible acceder a los casos de prueba para esta consulta a través del siguiente [vínculo](#) o se podrá encontrar en la carpeta “Casos de Prueba” bajo el nombre “[CPcons5.pdf](#)”.

Solución en cálculo relacional:

$\{ \text{mov1} / (\exists \text{mov1}) (\exists \text{mov2}) (\text{MOVIMIENTO}(\text{mov1}) \wedge \text{MOVIMIENTO}(\text{mov2}) \wedge$
 $(\text{mov1.concepto} = \text{mov2.concepto} \wedge \text{mov1.nombre} = \text{mov2.nombre} \wedge \text{mov1.importe} =$
 $(-1) * \text{mov2.importe} \wedge \text{EXTRACT}(\text{MONTH FROM mov1.fecha}) = \text{EXTRACT}(\text{MONTH FROM}$
 $\text{mov2.fecha})) \wedge$
 $\neg ((\exists \text{mov3}) \text{MOVIMIENTO}(\text{mov3}) \wedge \text{mov1.concepto} = \text{mov3.concepto} \wedge$
 $\text{mov1.nombre} = \text{mov3.nombre} \wedge \text{mov1.importe} = \text{mov3.importe} \wedge \text{EXTRACT}(\text{MONTH FROM}$
 $\text{mov1.fecha}) = \text{EXTRACT}(\text{MONTH FROM mov1.fecha})) \}$

--Se utiliza la función SQL "extract month" para referenciar a los movimientos que ocurrieron durante un mismo mes.

Ejercicio 6

Obtener los datos de los clientes y sus números de cuentas. Además, para los clientes que tienen cuentas activas, mostrar en la columna fecha de baja el texto "Activa" y para los clientes que tienen cuenta inactiva mostrar la fecha de baja en esa columna. Mostrar los datos para los clientes que contengan en su nombre "Rodríguez".

```
SELECT cliente.*, cuenta.numero,  
CASE  
WHEN cuenta.fechabaja IS NULL THEN 'ACTIVA'  
ELSE  
TO_CHAR(cuenta.fechabaja)  
END  
  
FROM cliente  
JOIN cuenta ON cuenta.cedula = cliente.cedula  
WHERE UPPER(cliente.nombre) LIKE '%RODRIGUEZ%'
```

Realizamos el JOIN entre las tablas CUENTA y CLIENTE a través de la cedula, donde buscamos los valores de Nombre coincidentes con "RODRIGUEZ", mostrando la palabra "Activa" cuando la fecha de baja es "NULL" y la fecha de baja si la cuenta fue dada de baja.

Ejercicio 7

Obtener para cada cliente la cantidad de cuentas que tiene y la cantidad de movimientos, teniendo en cuenta solamente los clientes que tengan la menor cantidad de cuentas de divisas diferentes. Ordenar el resultado alfabéticamente por nombre y por fecha de alta descendente.

```
SELECT cliente.*, (SELECT count(*)
FROM cuenta
WHERE cuenta.cedula = cliente.cedula) "CANTIDAD CUENTAS"
, (SELECT count(*)
FROM movimiento
JOIN cuenta ON movimiento.nombre =
cuenta.nombre AND
movimiento.numero = cuenta.numero
WHERE cuenta.cedula = cliente.cedula) "CANTIDAD MOVIMIENTOS"

FROM cliente
WHERE (SELECT COUNT (DISTINCT cuenta.moneda)--cantidad de
diferentes divisas del cliente

FROM Cuenta--cantidad de diferentes divisas del cliente
WHERE cuenta.cedula = cliente.cedula)/*cantidad de diferentes
divisas del cliente*/ =
(SELECT COUNT(DISTINCT cuenta.moneda)--la menor
cantidad de divisas diferentes
FROM cuenta --la menor cantidad de divisas diferentes
GROUP BY cuenta.cedula--la menor cantidad de divisas
diferentes
ORDER BY 1--la menor cantidad de divisas diferentes
FETCH FIRST 1 ROWS ONLY)--la menor cantidad de divisas
diferentes

ORDER BY cliente.nombre DESC, cliente.fechanacimiento
DESC
```

Para esta consulta comenzamos buscando la menor cantidad de divisas diferentes de todas las cuentas y luego buscamos que la cantidad de divisas diferentes de cada cuenta fuera igual a la menor de las cantidades hallada previamente. Luego contamos cuantas veces hay un movimiento que cumpla la condición y cuantas cuentas por cada cliente que cumplen la condición anterior.

Ejercicio 8

Obtener las sucursales que tienen la mayor cantidad de clientes millonarios en dólares. Se entiende por millonario a un cliente que tiene un saldo del total de sus cuentas en dólares en esa sucursal, mayor a un millón.

```
SELECT cuental.nombre
FROM cuenta cuental
where exists (SELECT cuenta.nombre,cuenta.cedula
              From movimiento
              join cuenta ON movimiento.numero =
cuenta.numero AND movimiento.nombre = cuenta.nombre
              where cuenta.moneda = 'DOLARES'
              AND cuenta.nombre = cuental.nombre AND
cuental.cedula = cuenta.cedula
              GROUP BY cuenta.nombre, cuenta.cedula
              having SUM(movimiento.importe) > 1000000
)
AND cuental.moneda = 'DOLARES'
GROUP BY cuental.nombre
HAVING COUNT(*) = (SELECT count(*)
                   FROM cuenta cuental
                   where exists (

SELECT cuenta.nombre,cuenta.cedula
FROM movimiento
JOIN cuenta ON movimiento.numero = cuenta.numero AND
movimiento.nombre = cuenta.nombre
WHERE cuenta.moneda = 'DOLARES'
AND cuenta.nombre = cuental.nombre AND cuental.cedula =
cuenta.cedula
GROUP BY cuenta.nombre, cuenta.cedula
HAVING SUM(movimiento.importe) > 1000000
)
                   AND cuental.moneda = 'DOLARES'
                   GROUP BY cuental.nombre
                   ORDER BY 1 DESC
                   FETCH FIRST 1 ROWS ONLY)

ORDER BY 1 DESC
```

En primer lugar agrupamos las cuentas en dólares donde la suma de sus movimientos fuera mayor a un millón. Posteriormente contamos cuantas cuentas cumplieron la condición anterior, obteniendo la mayor cantidad de clientes con más de un millón de dólares en dicha sucursal. Finalmente muestra las cuentas cuya cantidad de clientes millonarios sea igual al resultado máximo de cantidad de clientes millonarios en dólares.

Ejercicio 9

Obtener las sucursales donde se han creado la menor cantidad de cuentas para clientes menores de 25 años pero que dichas cuentas hayan generado la mayor cantidad de movimientos en los últimos 10 años.

```
SELECT sucursal.nombre
From sucursal
JOIN cuenta ON cuenta.nombre = sucursal.nombre
JOIN cliente on cuenta.cedula = cliente.cedula
WHERE cuenta.nombre = sucursal.nombre
AND (( cuenta.fechaalta - cliente.fechanacimiento )/365.25) < 25
AND EXISTS(
    SELECT movimiento.nombre,movimiento.numero, COUNT(*)-- COUNT CUENTA LA
CANTIDAD DE MOVIMIENTOS EN LOS ULTIMOS 10 AÑOS
    From movimiento
    where extract(year from movimiento.fecha) >= (SELECT EXTRACT(YEAR FROM
SYSDATE) -10 FROM DUAL)
    AND cuenta.nombre = movimiento.nombre AND cuenta.numero =
movimiento.numero-- PARA QUE NO ME TRAIGA DE OTRA CUENTA
    GROUP BY movimiento.nombre , movimiento.numero
    HAVING COUNT(*) = (
        SELECT COUNT(*)-- COUNT CUENTA LA CANTIDAD DE MOVIMIENTOS
        From movimiento
        GROUP BY movimiento.nombre , movimiento.numero
        ORDER BY 1 desc
        fetch first 1 rows only -- ME QUEDO CON EL NUMERO MAS GRANDE Y
LUEGO COMPARO CON ESE NUMERO
    )
)
```

Importante: Consideramos que el cliente debería ser menor de 25 al momento de creación de la cuenta.

Primero calculamos la cantidad de movimientos para cada cuenta en los últimos 10 años. Luego usando HAVING COUNT nos quedamos con las cuentas que tienen mayor cantidad de movimientos.

Casos de prueba:

Es posible acceder a los casos de prueba para esta consulta a través del siguiente [vínculo](#) o se podrá encontrar en la carpeta “Casos de Prueba” bajo el nombre “[CPcons9.pdf](#)”.

Ejercicio 10

Obtener para las estaciones de otoño e invierno, la cantidad de movimientos de ingreso y egreso que tuvo cada cliente por sucursal, mostrando además para cada moneda el porcentaje que la suma de los importes de esos movimientos representan sobre el total de importes sumados por ingreso y egreso por sucursal. Mostrar además, cuál fue el cliente que para cada sucursal mantuvo la mayor cantidad de dinero en total de todas sus cuentas bancarias en dólares.

```
SELECT MOVXCLIENTE.Estacion, MOVXCLIENTE.Sucursal ,
MOVXCLIENTE.Cliente , MOVXCLIENTE.TipoMov ,MOVXCLIENTE.cantMov
,MOVXCLIENTE.Moneda,
ROUND ((movxcliente.importe/movtotal.ImporteTotal)*100 , 2)AS
"%Importe/importeTotalSuc", clientesmayorsaldo.cliente AS
ClienteUSD
FROM

--INGRESOS DE OTONO
(
Select 'OTOÑO' AS Estacion,
      movimiento.nombre as Sucursal,
      cliente.nombre as Cliente,
      'Ingreso' as TipoMov,
      count(*) as CantMov,
      cuenta.moneda as Moneda,
      SUM(importe) as Importe

FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
JOIN cliente on cuenta.cedula = cliente.cedula
--OTONO
where ( extract(Month from(movimiento.fecha)) = 4 OR extract(Month
from(movimiento.fecha)) = 5 OR
      ( (extract(Day from(movimiento.Fecha)) >= 20 and
extract(Month from (movimiento.fecha)) = 3) ) OR
      ( (extract(Day from(movimiento.Fecha)) <= 21 and
extract(Month from (movimiento.fecha)) = 6))

AND movimiento.importe >0
GROUP BY movimiento.nombre, cliente.nombre,cuenta.moneda
UNION
--INGRESOS DE INVIERNO
```

```

Select 'INVIERNO' AS Estacion,
      movimiento.nombre as Sucursal,
      cliente.nombre as Cliente,
      'Ingreso' as TipoMov,
      count(*) as CantMov,
      cuenta.moneda as Moneda,
      SUM(importe) as Importe
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
JOIN cliente on cuenta.cedula = cliente.cedula

--invierno
where ( extract(Month from(movimiento.fecha)) = 8 OR extract(Month
from(movimiento.fecha)) = 7 OR
      ( (extract(Day from(movimiento.Fecha)) >= 21 and
extract(Month from (movimiento.fecha)) = 6) ) OR
      ( (extract(Day from(movimiento.Fecha)) <= 22 and
extract(Month from (movimiento.fecha)) = 9)))

AND movimiento.importe >0
GROUP BY movimiento.nombre, cliente.nombre,cuenta.moneda
UNION
--EGRESOS DE OTONO
Select 'OTOÑO' AS Estacion,
      movimiento.nombre as Sucursal,
      cliente.nombre as Cliente,
      'Egreso' as TipoMov,
      count(*) as CantMov,
      cuenta.moneda as Moneda,
      SUM(importe) as Importe
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
JOIN cliente on cuenta.cedula = cliente.cedula
--OTONO
where ( extract(Month from(movimiento.fecha)) = 4 OR extract(Month
from(movimiento.fecha)) = 5 OR
      ( (extract(Day from(movimiento.Fecha)) >= 20 and
extract(Month from (movimiento.fecha)) = 3) ) OR
      ( (extract(Day from(movimiento.Fecha)) <= 21 and
extract(Month from (movimiento.fecha)) = 6)))

AND movimiento.importe < 0
GROUP BY movimiento.nombre, cliente.nombre,cuenta.moneda
UNION
--EGRESOS DE INVIERNO
Select 'INVIERNO' AS Estacion,
      movimiento.nombre as Sucursal,
      cliente.nombre as Cliente,
      'Egreso' as TipoMov,

```

```

        count(*) as CantMov,
        cuenta.moneda as Moneda,
        SUM(importe) as Importe
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
JOIN cliente on cuenta.cedula = cliente.cedula

--invierno
where ( extract(Month from(movimiento.fecha)) = 8 OR extract(Month
from(movimiento.fecha)) = 7 OR
        ( (extract(Day from(movimiento.Fecha)) >= 21 and
extract(Month from (movimiento.fecha)) = 6) ) OR
        ( (extract(Day from(movimiento.Fecha)) <= 22 and
extract(Month from (movimiento.fecha)) = 9)))

AND movimiento.importe < 0
GROUP BY movimiento.nombre, cliente.nombre,cuenta.moneda

) MOVXCLIENTE,

---*****--*****
--INGRESOS DE OTONO
(
Select 'OTOÑO' AS Estacion,
        movimiento.nombre as Sucursal,
        'Ingreso' as TipoMov,
        cuenta.moneda as Moneda,
        SUM(importe) as ImporteTotal
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
--OTONO
where ( extract(Month from(movimiento.fecha)) = 4 OR extract(Month
from(movimiento.fecha)) = 5 OR
        ( (extract(Day from(movimiento.Fecha)) >= 20 and
extract(Month from (movimiento.fecha)) = 3) ) OR
        ( (extract(Day from(movimiento.Fecha)) <= 21 and
extract(Month from (movimiento.fecha)) = 6)))

AND movimiento.importe >0
GROUP BY movimiento.nombre,cuenta.moneda

UNION
--INGRESOS DE INVIERNO
Select 'INVIERNO' AS Estacion,
        movimiento.nombre as Sucursal,
        'Ingreso' as TipoMov,
        cuenta.moneda as Moneda,
        SUM(importe) as ImporteTotal
FROM movimiento

```



```

JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
--invierno
where ( extract(Month from(movimiento.fecha)) = 8 OR extract(Month
from(movimiento.fecha)) = 7 OR
        ( (extract(Day from(movimiento.Fecha)) >= 21 and
extract(Month from (movimiento.fecha)) = 6) ) OR
        ( (extract(Day from(movimiento.Fecha)) <= 22 and
extract(Month from (movimiento.fecha)) = 9)))

AND movimiento.importe >0
GROUP BY movimiento.nombre,cuenta.moneda
UNION
--EGRESOS DE OTONO
Select 'OTOÑO' AS Estacion,
        movimiento.nombre as Sucursal,
        'Egreso' as TipoMov,
        cuenta.moneda as Moneda,
        SUM(importe) as ImporteTotal
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero
--OTONO
where ( extract(Month from(movimiento.fecha)) = 4 OR extract(Month
from(movimiento.fecha)) = 5 OR
        ( (extract(Day from(movimiento.Fecha)) >= 20 and
extract(Month from (movimiento.fecha)) = 3) ) OR
        ( (extract(Day from(movimiento.Fecha)) <= 21 and
extract(Month from (movimiento.fecha)) = 6)))

AND movimiento.importe < 0
GROUP BY movimiento.nombre,cuenta.moneda

UNION
--EGRESOS DE INVIERNO
Select 'INVIERNO' AS Estacion,
        movimiento.nombre as Sucursal,
        'Egreso' as TipoMov,
        cuenta.moneda as Moneda,
        SUM(importe) as ImporteTotal
FROM movimiento
JOIN cuenta on movimiento.nombre = cuenta.nombre AND
movimiento.numero = cuenta.numero

--invierno
where ( extract(Month from(movimiento.fecha)) = 8 OR extract(Month
from(movimiento.fecha)) = 7 OR
        ( (extract(Day from(movimiento.Fecha)) >= 21 and
extract(Month from (movimiento.fecha)) = 6) ) OR
        ( (extract(Day from(movimiento.Fecha)) <= 22 and
extract(Month from (movimiento.fecha)) = 9)))

```

```

AND movimiento.importe < 0
GROUP BY movimiento.nombre,cuenta.moneda
) MOVTOTAL,

-----*****-----
--CLIENTES MAYORES SALDO POR SUCURSAL EN DOLARES
(Select cuenta.nombre AS SUCURSAL, cliente.nombre AS CLIENTE ,
SUM(movimiento.importe) AS SALDO
From Movimiento
Join cuenta on cuenta.nombre = movimiento.nombre AND cuenta.numero
= movimiento.numero
join cliente on cliente.cedula = cuenta.cedula
WHERE cuenta.moneda = 'DOLARES'
group by cuenta.nombre, cliente.nombre
HAVING SUM(movimiento.importe) = (
        Select SUM(movimiento.importe) AS SALDO
        From Movimiento
        Join cuenta c1 on c1.nombre = movimiento.nombre
AND c1.numero = movimiento.numero
        join cliente on cliente.cedula = c1.cedula
        WHERE c1.moneda = 'DOLARES'
        AND cuenta.nombre = c1.nombre
        group by c1.nombre, cliente.nombre
        ORDER BY 1 Desc
        fetch first 1 rows only
        )
) CLIENTESMAYORSALDO

WHERE --engancho la primera con la segunda
      movtotal.estacion = Movxcliente.estacion AND
      movtotal.sucursal = Movxcliente.sucursal AND
      movtotal.tipomov = movxcliente.tipomov AND
      movtotal.moneda = movxcliente.moneda AND
      --engancho la tercera con el resto enganchandola a la
      primera
      Movxcliente.sucursal = clientesmayorsaldo.sucursal

ORDER BY MOVXCLIENTE.Sucursal , MOVXCLIENTE.Cliente ,
MOVXCLIENTE.Estacion, MOVXCLIENTE.TipoMov ,MOVXCLIENTE.Moneda

```

Criterios: El cliente USD se tomó como el o los clientes con mayor saldo al día de hoy en el total de sus cuentas en dólares por sucursal.

El **%IMPORTE/importeTotSuc** es el porcentaje de la suma de los importes (ingresos y egresos) por estación, sucursal y moneda.

Para este ejercicio utilizamos una subconsulta en donde extrajimos cantidad de movimientos , suma de los importes de esos movimientos por cada estación , sucursal, cliente, moneda y tipo de movimiento.

Luego, utilizamos una segunda subconsulta bajo los mismos criterios indicando la cantidad total de movimientos, con la diferencia que esta vez solo se sumaron importes.

Finalmente, utilizamos una tercera subconsulta que devuelve al/los clientes con mayor saldo al día de hoy.

Casos de prueba:

Es posible acceder a los casos de prueba para esta consulta a través del siguiente [vínculo](#) o se podrá encontrar en la carpeta “Casos de Prueba” bajo el nombre “[CPcons10.pdf](#)”.