

Control N°3: Procedimientos Almacenados e Índices

Valeria Beratto Ulloa

Nombre: Benjamín Martínez Jeldres

RUT: 19.993.238-1

La empresa que controla todos los movimientos de la tarjeta MasterPlop y la información de los clientes que la utilizan, requiere una pequeña aplicación que le proporcione información, tanto a los clientes como a sus ejecutivos. La tarjeta tiene un código que es único, una fecha de caducidad, cupo máximo, cupo disponible y cupo utilizado. Así también, cada tarjeta está asociada a un sólo cliente, pero un cliente puede tener más de una tarjeta. De los clientes se tiene el rut, nombre completo, dirección y ciudad. Cada cliente puede realizar dos tipos de movimientos:

- a) **Las compras**, que descuenta el valor de la compra al cupo disponible y lo suman a cupo utilizado.
- b) **Los pagos de la deuda**, que descuenta el valor pagado al cupo utilizado y lo suman al cupo disponible.

Para cada movimiento se registra el tipo de movimiento, un código de operación, valor y fecha.

La empresa realiza distintos tipos de consultas, pero las más relevantes son: por tipo de movimiento, por cliente y los movimientos en un determinado rango de fecha.

Además, mensualmente la empresa emite un **informe clasificando a los clientes** de acuerdo a la cantidad de movimientos realizados en el mes anterior, consideran a quienes realizan más de 10 movimientos mensuales, como cliente frecuente. Quienes realizan menos de 10 movimientos y más de 5, como clientes regulares y los que realizan menos de 5 movimientos, como clientes críticos.

También, se requiere generar el **estado de cuenta**, considerando todos los movimientos del mes anterior, total en compras, total pagado y saldo disponible y saldo utilizado en el mes, donde se debe ingresar el rut como parámetro.

De acuerdo a lo anterior, se requiere que desarrolle en PL/SQL:

1. Los procedimientos necesarios para generar los informes mensuales (60 pts.)
2. Dos índices que considere que sean relevantes para la optimización de consultas, de acuerdo a lo requerido por la empresa, demostrando la disminución en el costo y uso de ellos. (40 ptos.)

Indicaciones Generales:

1. El control se puede realizar en forma individual, cualquier copia será sancionada con nota 1.
2. Utilice el siguiente esquema:

CLIENTE (**rut**, nombre, dirección, ciudad)

TARJETA (**código**, fecha_caducidad, cupo_máximo, cupo_disponible, cupo_utilizado, *rut_cliente*)

TIPO_MOVIMIENTO(**código**, nombre)

MOVIMIENTO(**código_operación**, valor, fecha, *código_movimiento*, *código_tarjeta*)

Donde, los atributos en negrita son claves primarias y los atributos en cursiva son claves foráneas.

3. Para cada respuesta, debe colocar una fotografía o screenshot que contenga del código en PL/SQL y/o resultado obtenido en ORACLE SQL Developer u ORACLE SQL LIVE
4. **Tiempo límite para subir el control a la plataforma en formato .pdf, miércoles 22 de diciembre hasta las 23:59 hrs.**
5. Cualquier inconveniente enviar un correo electrónico a yberatto@ubiobio.cl

No olvidar subir a la plataforma el archivo en formato pdf.

¡ÉXITO!!!

Desarrollo

Tablas:

SQL Worksheet

```
1 create table CLIENTE (  
2   rut varchar2 (30) primary key ,  
3   nombre varchar2 (40),  
4   direccion varchar2 (100),  
5   ciudad varchar2 (40)  
6 );  
7  
8 create table TARJETA (  
9   codigo integer primary key,  
10  fecha_caducidad date,  
11  cupo_maximo integer,  
12  cupo_disponible integer,  
13  cupo_utilizado integer,  
14  rut_cliente varchar2(30),  
15  foreign key (rut_cliente) references cliente  
16 );  
17  
18 create table TIPO_MOVIMIENTO(  
19  codigo integer primary key,  
20  nombre varchar2(30)  
21 );  
22  
23 create table MOVIMIENTO(  
24  codigo_operacion integer primary key,  
25  valor integer,  
26  fecha date,  
27  codigo_movimiento integer,  
28  codigo_tarjeta integer,  
29  foreign key(codigo_movimiento) references tipo_movimiento(codigo),  
30  foreign key(codigo_tarjeta) references tarjeta(codigo)  
31 );  
32
```

Tuplas:

```
-----]  
--tuplas-----]  
  
---CLIENTE---  
insert into cliente values('11111111-1','benjamin martinez','las lomas 111','concepcion');  
insert into cliente values('11111111-2','cristobal jeldres','las palomas 222','talca');  
insert into cliente values('11111111-3','lucho jara','las gaviotas 333','santiago');  
  
---TARJETA---  
insert into tarjeta values(1,'01-nov-2022',100000,50000,0,'11111111-1');  
insert into tarjeta values(2,'02-nov-2022',100000,60000,0,'11111111-2');  
insert into tarjeta values(3,'03-nov-2022',100000,70000,0,'11111111-3');  
  
---TIPO_MOVIMIENTO---  
insert into tipo_movimiento values(10,'compras');  
insert into tipo_movimiento values(11,'pagos deuda');  
  
---MOVIMIENTO---  
insert into movimiento values(100,5000,' 01-nov-2021',10,1);  
insert into movimiento values(101,2000,' 02-nov-2021',10,2);  
insert into movimiento values(102,1000,' 03-nov-2021',10,3);
```

Consultas:

1) Clasificación de clientes

```
--CLASIFICACION DE CLIENTES-----

---CLIENTES FRECUENTES >10 ---
---CLIENTES REGULARES <10 Y >5 ---
---CLIENTES CRITICOS <5 ---

declare
  cli integer;
  cant integer;
begin
  cli:= &ingresa_cliente;
  select count(1) into cant from movimiento where codigo_operacion= cli;

  if cant>10 then
    dbms_output.put_line ('el cliente es frecuente');
  elsif cant<10 and cant>=5 then
    dbms_output.put_line ('el cliente es regular');

  elsif cant<5 then
    dbms_output.put_line ('el cliente es critico');
  end if;

end;

---Estado de cuenta-----]
```

```

declare
  cli integer;
  cant integer;
begin
  cli:= &ingresa_cliente;
  select count(1) into cant from movimiento where codigo_operacion= cli;

  if cant>10 then
    dbms_output.put_line ('el cliente es frecuente');
  elsif cant<10 and cant>=5 then
    dbms_output.put_line ('el cliente es regular');

  elsif cant<5 then
    dbms_output.put_line ('el cliente es critico');
  end if;

end;

el cliente es critico

Procedimiento PL/SQL terminado correctamente.
  
```

Tuve que mandarle mi código a un amigo y me hizo el favor de ejecutarlo, ya que plsqli live no deja usar el comando & .

2) Estado de cuenta

Indices: