



Gestión de Datos Estrategia

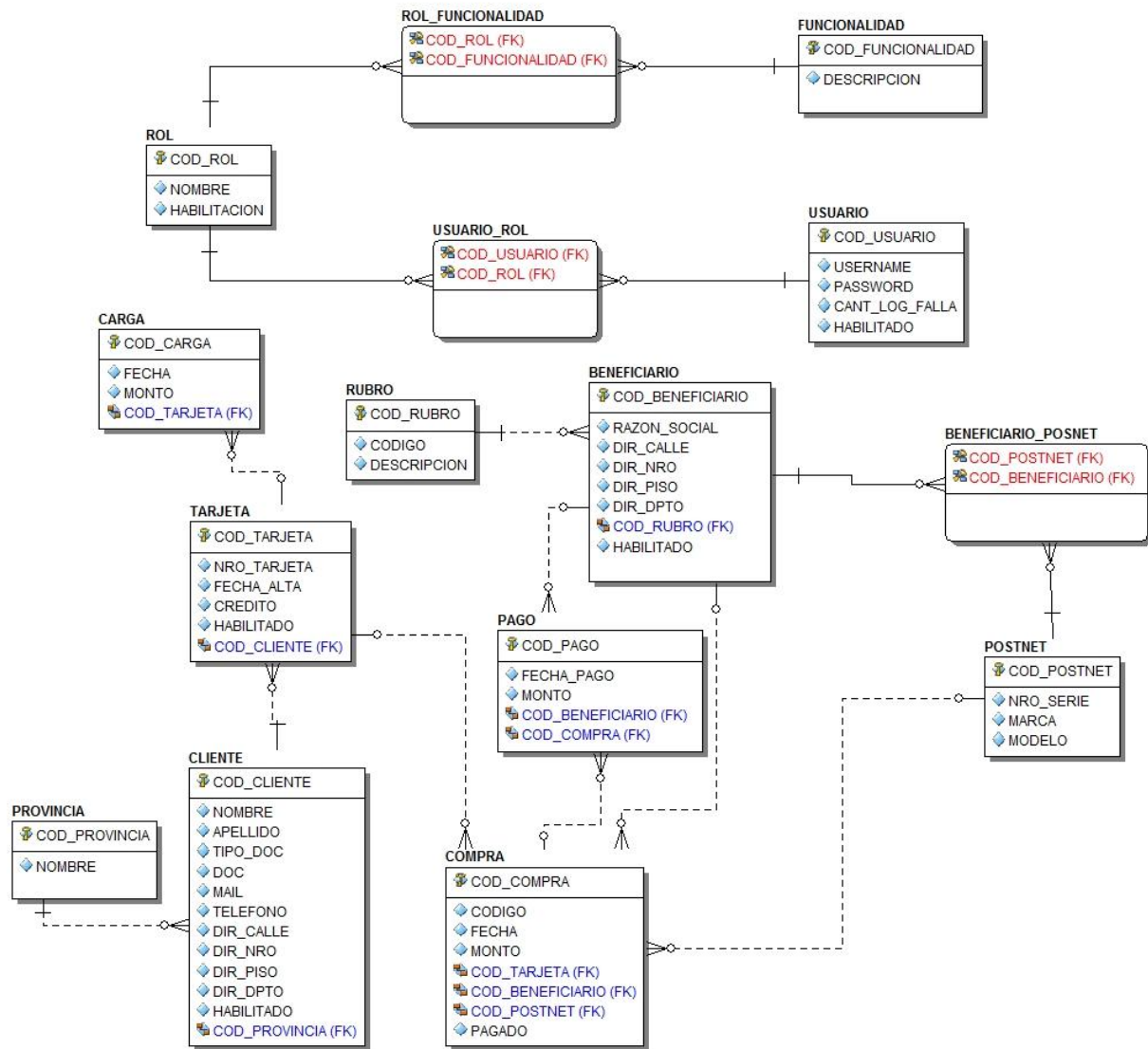
Grupo:
NoTeVaConsultar

Año: 2011

Índice

Tema	Pagina
Modelo de Datos	2
Stored Procedures	2
Functions	3
Cursor	3
Vistas	3
Creación de Tablas	4
Inicialización de Tablas	4
Migración de la tabla Maestra	5
Criterios para campos repetidos	5
Criterios para pagos ya realizados	5
Clientes Premium	6
Inhabilitación de tarjetas	6
Relación cliente-tarjeta	6

Modelo de Datos



Stored Procedures:

NTVC.ACTUALIZARPAGO: Este procedimiento lo que hace es actualizar las compras, verificando de esta manera que la compra haya sido pagada al Beneficiario correspondiente.

NTVC.CREDITOTARJETA: Este procedimiento, realiza la actualizacion del credito en la Tarjeta, obteniendo el valor de todo lo que cargo y disminuyendole el monto de todo lo que gasto.

NTVC.DNIREPETIDOS: Este procedimiento, utiliza un cursor. Se utiliza para poder verificar que no existan DNIs repetidos, de existir se lo multiplica por -1, logrando de esta forma, que el DNI repetido sea negativo. Llegado al caso de querer modificarlo, no dejaría guardarlo como negativo.

Functions:

NTVC.TARJETASINACTIVAS: Esta funcion devuelve las tarjetas inactivas que existen. Desde la aplicación Desktop simplemente se llama a la funcion y esta le devuelve la tabla completa, con los campos requeridos por el enunciado. No tiene parametros de entrada.

NTVC.CLIENTEPREMIUM: Esta funcion devuelve los clientes premium (con todas las especificaciones que estan numeradas en el enunciado). La aplicación Desktop llama a esta funcion y obtiene la tabla correspondiente. Recibe como parametro el año que se quiere consultar los clientes premium.

Cursor:

Se utilizo el cursor CURSORDNI el cual va verificando cada DNI, donde exista un duplicado, teniendo en cuenta que el DNI es igual pero se diferencian en Nombre y Apellido.

Vistas

Las vistas utilizadas, se crearon con el fin de dinamizar y agilizar la consulta de las tarjetas inactivas. Esta consulta demoraba considerablemente dado que se tenía que filtrar los clientes que eran inactivos, y luego de esto, obtener distintos datos correspondientes a el.

Las vistas creadas fueron:

- Cliente_activo: Vista que contiene todos los clientes que cumplen la condición de tener como estado activo.
- Cliente_inactivo: Todos los clientes que no cumplen con la condición anterior (se utiliza Cliente_Activo, de esta forma se agiliza la consulta).
- Monto_total: obtiene el monto acumulado total de los clientes que se encuentran inactivos (se utiliza Cliente_Inactivo, agilizando la consulta).
- Fecha_Ultima_Compra: Obtiene las últimas compras correspondientes a los clientes inactivos (se utiliza Cliente_Inactivo).
- Datos_Ultima_Compra: Obtiene los datos de las ultimas compras que corresponden a los clientes inactivos. (Se utiliza Fecha_Ultima_Compra y Cliente_Inactivo, de esta forma, se lograron filtrar la última consulta considerablemente).

Creación de las tablas

Todas las tablas fueron creadas con una *Primary Key identity(1,1) not null*, es decir, la PK es un valor numérico que comienza en 1 y va incrementándose en una unidad, imposibilitando de esta manera que exista una PK nula.

Esta estrategia fue elegida, dado que los datos que se encontraban en la tabla *Maestra* y que eran claves candidatas, mostraban algunos problemas para utilizarlas como claves primarias. Para citar algunos ejemplos, se encontraban duplicadas, o simplemente no existían.

Se crearon las tablas intermedias Rol_Funcionalidad para poder administrar todas las funcionalidades que puede tener un rol. Además, Usuario_Rol, donde se administra los roles que tiene un usuario. Otra tabla intermedia es Beneficiario_Postnet, el cual organiza los post-nets que pertenecen a los beneficiarios.

Inicialización de tablas

Se creó una tabla *Provincia* la cual se inicializa automáticamente, una vez que se ejecuta el script, con todas las provincias de la Republica Argentina (incluyendo la Ciudad Autónoma de Buenos Aires) y una fila adicional la cual esta especificada para personas extranjeras. De esta manera, solucionamos el inconveniente que se presentaba si un extranjero solicitaba la tarjeta.

Todas las tablas que tienen la posibilidad de inhabilitar alguna de sus filas, el campo *Habilitado* se inicializa en 1, es decir, se inicializa con la fila habilitada.

Se inicializaron todas las funcionalidades descriptas en el enunciado del TP V1.0, con sus respectivas características. A su vez, se crea el rol “Administrador General” como bien indica el enunciado, para poder permitir una prueba del programa. Además, se creó el usuario admin, con las características preestablecidas.

Se crearon post-nets inicializados por defecto, llegado el caso de que se cree un Beneficiario. De esta forma, se puede asignar a un Beneficiario, estos post-nets creados que no tienen asignado un dueño.

Por último, se crearon los roles “Gestor” y “Punto de Venta” con la funcionalidades de todos los ABMs para el primero, y Carga de Crédito y Efectuar Compra para el segundo.

Migración de la tabla Maestra

Al migrar los datos de los clientes de la tabla *Maestra* hacia la tabla *Clientes*, existe la nulidad en el campo *Provincia*, por lo cual se carga por defecto con el atributo “**Migrada**”. De esta forma, dejamos en claro, que este cliente fue introducido al sistema a través de la migración y no contaba con una provincia.

Criterios para campos repetidos

Dentro de la tabla *Maestra* se encontraron 4 clientes, los cuales tienen repetido el DNI, por lo cual, la estrategia optada por el grupo, fue la de a uno de esos repetidos multiplicarlo por -1. De esta manera, quedan los dos DNI con el valor original inalterado, pero podemos diferenciar de manera unívoca a uno del otro.

Llegado el caso de que uno de estos casos particulares se quiera modificar, no se dejaría modificar el usuario sin antes haber modificado el DNI negativo, dado que la aplicación Desktop, valida que el DNI además de ser numérico, sea positivo.

Criterios para pagos ya realizados

Una de las situaciones con las que nos encontramos en este TP, al migrar era la necesidad de determinar que compras fueron ya realizadas el correspondiente pago a su respectivo Beneficiario.

Es por ello, que se decidió que se agregara el campo *Pagado* dentro de la tabla *Compra*, para determinar las compras ya cobradas para el Beneficiario.

De esta forma, podremos filtrar las compras no acreditadas, para que se pueda proceder a pagarlas.

Además, en la migración pudimos detectar una situación inesperada, las compras realizadas al Beneficiario A eran pagadas al Beneficiario B (esto sucedió en pocos casos). Por lo que el grupo tomo como estrategia saldar los pagos con los Beneficiarios que habían estado involucrados en la transacción inicial.

Cientes Premium

Para poder aprovechar las herramientas que nos otorga SQL, decidimos implementar una función que retorne la consulta, esta recibe como parámetro el año que se quiere consultar y retorna los 30 primeros clientes que más realizaron compras en el plazo de ese año.

En el enunciado se pide que se exprese el monto acumulado y la fecha de la última compra registrada. Nosotros consideramos que el monto acumulado a informar es el total de compras realizadas en el año indicado para la consulta. A su vez, la última fecha de compra registrada, corresponde a la última compra registrada en ese año por ese usuario, pudiendo existir fechas posteriores correspondientes a años siguientes.

Inhabilitación de tarjetas

Para agilizar las consultas de esta funcionalidad, se utilizaron distintas vistas. Esto proporciono, una dinamización de la herramienta. Y haciendo las consultas y las inhabilitaciones mucho más performantes.

Es por ello, que se realizo una function, la cual una vez llamada, devuelve la tabla de los usuarios inactivos.

Relación cliente-tarjeta

Al darse de baja un cliente, no sólo se inhabilita el mismo sino también la tarjeta que tenga asignada, en caso de existir.

Una tarjeta no podrá asignarse a un cliente que ha sido deshabilitado, ya sea al intentar darla de alta o modificarla.

Por lo tanto si nos encontramos en el primer caso, para poder volver a habilitar esa tarjeta primero debe volver a habilitarse al cliente, y luego a su tarjeta.