

Trabajo Práctico

Bases de Datos I — Segundo Semestre 2020

1 Modelo de Datos

A continuación se presenta el modelo de datos que se usa para almacenar la información relativa a una tarjeta de crédito.

```
cliente(  
    nrocliente:int,  
    nombre:text,  
    apellido:text,  
    domicilio:text,  
    telefono:char(12)  
)  
  
tarjeta(  
    nrotarjeta:char(16),  
    nrocliente:int,  
    validadesde:char(6), --e.g. 201106  
    validahasta:char(6),  
    codseguridad:char(4),  
    limitecompra:decimal(8,2),  
    estado:char(10) --`vigente', `suspendida', `anulada'  
)  
  
comercio(  
    nrocomercio:int,  
    nombre:text,  
    domicilio:text,  
    codigopostal:char(8),  
    telefono:char(12)  
)  
  
compra(  
    nrooperacion:int,  
    nrotarjeta:char(16),  
    nrocomercio:int,  
    fecha:timestamp,  
    monto:decimal(7,2),  
    pagado:boolean  
)
```

```

rechazo(
    nrorechazo:int,
    nrotarjeta:char(16),
    nrocomercio:int,
    fecha:timestamp,
    monto:decimal(7,2),
    motivo:text
)

cierre(
    año:int,
    mes:int,
    terminacion:int,
    fechainicio:date,
    fechacierre:date,
    fechavto:date
)

cabecera(
    nroresumen:int,
    nombre:text,
    apellido:text,
    domicilio:text,
    nrotarjeta:char(16),
    desde:date,
    hasta:date,
    vence:date,
    total:decimal(8,2)
);
detalle(
    nroresumen:int,
    nrolinea:int,
    fecha:date,
    nombrecomercio:text,
    monto:decimal(7,2)
);

alerta(
    nroalerta:int,
    nrotarjeta:char(16),
    fecha:timestamp,
    nrorechazo:int,
    codalerta:int, --0:rechazo, 1:compra 1min, 5:compra 5min, 32:límite
    descripcion:text
);

```

```
-- Esta tabla no es parte del modelo de datos, pero se incluye para
-- poder probar las funciones.
consumo(
    nrotarjeta:char(16),
    codseguridad:char(4),
    nrocomercio:int,
    monto:decimal(7,2)
)
```

Esta tarjeta no permite a los usuarios financiar una compra en cuotas, todo es en un solo pago. No existen las extensiones, sin embargo una persona puede tener más de una tarjeta.

2 Creación de la Base de Datos

Se deberán crear las tablas respetando los nombres de tablas, atributos y tipos de datos especificados.

Se deberán agregar las PK's y FK's de todas las tablas, por separado de la creación de las tablas. Además, el usuario deberá tener la posibilidad de borrar todas las PK's y FK's, si lo desea.

3 Instancia de los Datos

Se deberán cargar 20 clientes y 20 comercios. Todos los clientes tendrán una tarjeta, excepto dos clientes que tendrán dos tarjetas cada uno. Una tarjeta deberá estar expirada en su fecha de vencimiento.

La tabla `cierre` deberá tener los cierres de las tarjetas para todo el año 2020.

4 Stored Procedures y Triggers

El trabajo práctico deberá incluir los siguientes stored procedures ó triggers:

- **autorización de compra** se deberá incluir la lógica que reciba los datos de una compra—número de tarjeta, código de seguridad, número de comercio y monto—y que devuelva `true` si se autoriza la compra ó `false` si se rechaza. El procedimiento deberá validar los siguientes elementos antes de autorizar:
 - Que el número de tarjeta sea existente, y que corresponda a alguna tarjeta vigente. En caso de que no cumpla, se debe cargar un rechazo con el mensaje `?tarjeta no válida ó no vigente`.
 - Que el código de seguridad sea el correcto. En caso de que no cumpla, se debe cargar un rechazo con el mensaje `?código de seguridad inválido`.

- Que el monto total de compras pendientes de pago más la compra a realizar no supere el límite de compra de la tarjeta. En caso de que no cumpla, se debe cargar un rechazo con el mensaje **?supera límite de tarjeta**.
- Que la tarjeta no se encuentre vencida. En caso de que no cumpla, se debe cargar un rechazo con el mensaje **?plazo de vigencia expirado**.
- Que la tarjeta no se encuentre suspendida. En caso de que no cumpla, se debe cargar un rechazo con el mensaje **la tarjeta se encuentra suspendida**.

Si se aprueba la compra, se deberá guardar una fila en la tabla **compra**, con los datos de la compra.

- **generación del resumen** el trabajo práctico deberá contener la lógica que reciba como parámetros el número de cliente, y el periodo del año, y que guarde en las tablas que corresponda los datos del resumen con la siguiente información: nombre y apellido, dirección, número de tarjeta, periodo del resumen, fecha de vencimiento, todas las compras del periodo, y total a pagar.
- **alertas a clientes** el trabajo práctico deberá proveer la lógica que genere alertas por posibles fraudes. Existe un Call Centre que ante cada alerta generada automáticamente, realiza un llamado telefónico a le cliente, indicándole la alerta detectada, y verifica si se trató de un fraude ó no. Se supone que la detección de alertas se ejecuta automáticamente con cierta frecuencia—e.g. de una vez por minuto. Se pide detectar y almacenar las siguientes alertas:
 - Todo rechazo se debe ingresar automáticamente a la tabla de alertas. No puede haber ninguna demora para ingresar un rechazo en la tabla de alertas, se debe ingresar en el mismo instante en que se generó el rechazo.
 - Si una tarjeta registra dos compras en un lapso menor de un minuto en comercios distintos ubicados en el mismo código postal.
 - Si una tarjeta registra dos compras en un lapso menor de 5 minutos en comercios con diferentes códigos postales.
 - Si una tarjeta registra dos rechazos por exceso de límite en el mismo día, la tarjeta tiene que ser suspendida preventivamente, y se debe grabar una alerta asociada a este cambio de estado.

Se deberá crear una tabla con consumos virtuales para probar el sistema, la misma deberá contener los atributos: **nrotarjeta**, **codseguridad**, **nrocomercio**, **monto**. Y se deberá hacer un procedimiento de testeo, que pida autorización para todos los consumos virtuales.

Todo el código SQL escrito para este trabajo práctico, deberá poder ejecutarse desde una aplicación CLI escrita en Go.

5 JSON y Bases de datos NoSQL

Por último, para poder comparar el modelo relacional con un modelo no relacional NoSQL, se pide guardar los datos de clientes, tarjetas, comercios, y compras (tres por cada entidad) en una base de datos NoSQL basada en JSON. Para ello, utilizar la base de datos BoltDB. Este código, también deberá ejecutarse desde una aplicación CLI escrita en Go.

6 Condiciones de Entrega

- El trabajo es grupal, en grupos de, *exactamente*, 3 integrantes. Se debe realizar en un repositorio privado `git`, hosteado en [Gitlab](#) con el apellido de los tres integrantes, separados con guiones, seguidos del string ‘-tp’ como nombre del proyecto, e.g. `maradona-riquelme-tevez-tp`. Agregar a los docentes de la materia: los usuarios `hdr` y `hernancz` en el repo como `maintainer`’s.
- La fecha de entrega *máxima* es el 3 de diciembre de 2020 a las 1800hs, con una defensa presencial del trabajo práctico por cada grupo, en la cual los docentes de la materia van a mirar lo que se encuentre en el repo `git` hasta el momento.
- El informe del trabajo práctico se debe presentar en formato AsciiDoc. Para ello, cuentan con una guía en hdr.gitlab.io/adoc.