Departamento de Cs. e Ingeniería de la Computación

Universidad Nacional del Sur

Bases de Datos

Segundo Cuatrimestre de 2020

Proyecto N$^{\circ}$ 3

Transacción para abrir y cerrar estacionamientos

Cuando una tarjeta se conecta a un parquímetro es necesario determinar si trata de la apertura o cierre de un estacionamiento. Para esto se deberá controlar si la tarjeta tiene un estacionamiento abierto, esto es, el estacionamiento tiene fecha y hora de entrada pero no tiene fecha y hora de salida (valores nulos).

En caso de tener un estacionamiento abierto se deberá cerrar dicho estacionamiento actualizando el saldo de la tarjeta, tomando como fecha y hora de salida la fecha y hora actual. En caso contrario, si el saldo de la tarjeta es mayor a cero se abrirá un nuevo estacionamiento para la tarjeta y parquímetro involucrados, tomando como fecha y hora de entrada la fecha y hora actual.

Para actualizar el *saldo* de la tarjeta se utilizará la siguiente fórmula, pudiendo obtenerse un saldo negativo: $nuevo\_saldo = saldo - (tiempo * tarifa * (1 - descuento))$, donde:

- *tiempo*: cantidad de minutos desde la fecha y hora de entrada hasta la fecha y hora actual.
- *tarifa*: costo por minuto asociado a la ubicación donde se abrió el estacionamiento.
- *descuento*: bonificación asociada al tipo de la tarjeta.

Debido a que esta operación actualiza más de una tabla y debe realizarse de forma atómica es necesario implementarla mediante una transacción.

Ejercicios

1. Mediante un *stored procedure* implemente una transacción que realice, según corresponda, la apertura o cierre de un estacionamiento. Dicho *stored procedure* deberá llamarse *conectar* y recibirá como parámetros el id de una tarjeta y el id de un parquímetro, y deberá tener el siguiente encabezado: `conectar(IN id_tarjeta INTEGER , IN id_parq INTEGER)`

   En el caso de producirse una *apertura* el s.p. deberá devolver una tabla con la siguiente información:

   - el tipo de operación realizada (`apertura`),
   - si la operación se realizó con éxito o no (el saldo de la tarjeta es menor o igual a cero) y
   - la cantidad de *tiempo* en minutos que dispone para estar estacionado: $tiempo = \frac{saldo}{tarifa*(1-descuento)}$

   En el caso de producirse un *cierre* el s.p. deberá devolver una tabla con la siguiente información:

   - el tipo de operación realizada (`cierre`),
   - la cantidad de tiempo transcurrido en minutos y
   - el saldo actualizado de la tarjeta.

   Los datos de salida se retornarán a través de una tabla (utilizando una sentencia de la forma: `SELECT 'apertura' as operacion, ...`) conteniendo la información necesaria según la operación sea una apertura, cierre o se produzca algún error que impida hacer la transacción (como

por ejemplo que id_tarjeta o id_parq no existan en la B.D., el saldo de la tarjeta sea menor o igual a cero, etc.)

El *stored procedure* deberá definirse dentro de la B.D. *parquimetros*. La sentencia de creación de este procedimiento deberá *añadirse* al archivo *"parquimetros.sql"* creado en el proyecto 2 .

2. Incorpore un nuevo usuario al servidor MySQL llamado *parquímetro*. Este usuario está destinado a permitir el acceso de los parquímetros a la base de datos, para abrir y cerrar estacionamientos. Este usuario deberá tener privilegios para ejecutar el *stored procedure* definido en el ejercicio 1. El password de este usuario deberá ser *parq* y la sentencia de creación de este usuario deberá *añadirse* al archivo *"parquimetros.sql"*.

3. Incorpore a la base de datos una nueva tabla llamada *ventas* con los siguientes campos: *id_tarjeta, tipo_tarjeta, saldo, fecha y hora*. La sentencia de creación de dicha tabla deberá *añadirse* al archivo *"parquimetros.sql"*.

   Implemente un *trigger* para llevar una traza de todas las ventas de tarjetas realizadas. Dicho trigger se activará cuando se inserta una nueva tarjeta y guardará en la tabla *ventas* la siguiente información: el id de la tarjeta, el tipo de la tarjeta, el saldo inicial, la fecha y hora actual. La sentencia de creación de dicho trigger deberá *añadirse* al archivo *"parquimetros.sql"*.

4. Extienda la aplicación desarrollada en Java en el proyecto 2, para permitir simular la conexión de una tarjeta a cualquier parquímetro.

   La aplicación deberá permitir lo siguiente:

   - Seleccionar una ubicación de las existentes en la base de datos.
   - Seleccionar un parquímetro dentro de la ubicación seleccionada.
   - Seleccionar una tarjeta de los existentes en la base de datos.

   Además deberá proveer un botón que al ser presionado simulará la conexión de la tarjeta elegida al parquímetro seleccionado, invocando al stored procedure desarrollado en ejercicio 1 a través del usuario *parquímetro* creado en el ejercicio 2. A continuación se deberá mostrar un mensaje con la información devuelta por el stored procedure.

Fechas y condiciones de entrega

- **Fecha límite de entrega: 27 de Noviembre de 2020**: a través del curso **Moodle** de la materia y utilizando la tarea correspondiente habilitada para tal fin, se deberá subir un archivo comprimido (zip o rar) que contenga:
  - el archivo "parquimetros.sql" extendido con lo solicitado en los ejercicios 1, 2 y 3
  - los archivos fuentes de la aplicación extendida con lo solicitado en el ejercicio 4, junto con el archivo JAR ejecutable correspondiente. **La aplicación deberá incluir las correcciones a las observaciones realizadas por la cátedra sobre la entrega del proyecto 2.**
- **Fecha límite de re-entrega: 7 de Diciembre de 2020**: a través del curso **Moodle** de la materia y utilizando la tarea correspondiente habilitada para tal fin, se deberá subir un archivo comprimido (zip o rar) con lo solicitado en la entrega anterior, **que incluya las correcciones a todas las observaciones indicadas en la corrección de la entrega del proyecto 3**.
- **Comisiones:** Los proyectos deben realizarse en comisiones de *dos alumnos* cada una. Las comisiones deberán ser *las mismas* para todas las entregas. Para las entregas a **solo es necesario que uno de los integrantes de cada comisión suba los archivos solicitados** en la tarea.
- **Importante:** La entrega en *tiempo y forma* de este proyecto es *condición de cursado* de la materia.