

INF-239 Bases de Datos

Tarea 1: SQL Server

Profesores: Sebastián Gallardo - Rodrigo Olavarría

Ayudante de Cátedra: Tomás Guttman

Ayudantes de Tareas: Débora Alayo - Daniela Sánchez - Sebastián Castro

14 de septiembre, 2022

1. Contextualización

MultiUSM corresponde a una multitienda que abrirá al público dentro de poco, pretende ofrecer una variedad de prestigiosos productos para sus clientes a un módico precio. Dichos productos corresponden a bebidas, dulces y postres, cuidado personal, productos congelados, entre otros. Para realizar compras, el cliente contará con un carrito al cual puede añadir los productos que desee comprar, junto con su cantidad. Además con el fin de establecer un orden entre los productos de la tienda, a cada artículo se le asoció con una categoría. Para incentivar el consumo tras la apertura, MultiUSM ha implementado una serie de ofertas en algunos de sus productos, como descuentos y precios especiales al agregar una determinada cantidad de un producto al carrito (por ejemplo, 2x1 en galletas). Para esto, se le solicita crear un programa que simule la interacción de un cliente con la multitienda utilizando una conexión a SQL Server Express.

2. Descripción del problema

Para el desarrollo de esta tarea se le solicita que simule la interacción de un cliente con la multitienda MultiUSM. Para eso se le entregará un dataset **productos.csv** con la siguiente información:

- **prod_id** - Identificador del producto [int]
- **prod_name** - Nombre del producto [str]
- **prod_description** - Descripción del producto [str]
- **prod_brand** - Marca del producto [str]
- **category** - Tipo de producto [str]
- **prod_unit_price** - Precio por unidad [int]

Además debe crear la tabla **Carrito**, que contiene las siguientes columnas:

- **prod_id** - Identificador del producto [int]
- **prod_name** - Nombre del producto [str]
- **prod_brand** - Marca del producto [str]
- **quantity** - Cantidad del producto [int]

la tabla **Boleta**, que contiene las siguientes columnas:

- **prod_id** - Identificador del producto [int]
- **offer** - Oferta del articulo [str]
- **total_value** - Valor total por producto [int]
- **final_value** - Valor total por producto con la oferta aplicada [int]

La tabla **Oferta**, que contiene las siguientes columnas:

- **prod_id** - Identificador del producto [int]
- **offer** - Oferta del articulo [str]

3. To Do List

- Se solicita establecer una conexión entre python y su base de datos, para poder trabajar a través de queries desde el archivo .py
- Se solicita que realice una aplicación que simule la interacción entre un usuario y MultiUSM, que se debe presentar por medio de consola a través de un archivo python.
- Debe cargar el archivo csv en la base de datos con los criterios de la tabla Productos.
- Se debe crear una tabla adicional llamada Carrito que contiene los productos elegidos por el usuario, con los criterios de la tabla Carrito.
- Se debe crear una tabla adicional Boleta que contiene la información de la boleta del usuario.
- Se debe crear una tabla adicional Oferta que contiene la información de las ofertas de los productos. Para esta tabla solo debe considerar aquellas ofertas definidas en el atributo prod_description de la tabla Productos que cumplan con la estructura pague M lleve N. El atributo offer entonces será MxN.
- A la tabla Carrito debe añadirle los registros de los productos seleccionados por el usuario.
- La tabla Boleta debe generar de manera automática los registros respectivos cuando un usuario de por finalizada su compra.
- Su aplicación debe mostrar un menú de navegación que permita realizar las siguientes operaciones:
 1. Mostrar mi carrito: que muestre todos los productos seleccionados hasta el momento por el usuario.
 2. Agregar productos al carrito.
 3. Mostrar Top 5 productos más caros.
 4. Mostrar los 5 productos más caros según categoría. Se debe solicitar la categoría al momento de seleccionar esta opción.
 5. Finalizar compra, momento en el cual automáticamente se debe generar la boleta, considerando las ofertas de la tabla Productos, la cual se encuentra detallada en el atributo prod_description
 6. Mostrar mi boleta: Mostrar la boleta del usuario.
 7. Mostrar valor total: Mostrar el monto total a pagar en base a la boleta del usuario.
 8. Buscar producto según nombre que muestre toda la información asociada a este desde la tabla Productos.
 9. Eliminar todos los productos del carrito del usuario.
 10. Eliminar producto: Se debe solicitar el nombre del producto y eliminarlo del carrito. Si además implementan con éxito lo siguiente, hay un bonus 6 puntos: En caso de existir más de un producto con el mismo nombre, el usuario debe elegir cual eliminar.

- Su aplicación debe ser capaz de gestionar eventuales errores al momento de interactuar con el menú (por ejemplo: eliminar un producto que no ha sido agregado al carrito previamente)
- Se debe crear una **view** a elección, debe ser útil.
- Se debe crear un **trigger** a elección, debe ser funcional.
- Se debe crear un **procedimiento almacenado** a elección, debe ser funcional.
- Se debe crear una **función** a elección, debe ser funcional.

4. Especificaciones y reglas

El desarrollo de esta tarea debe cumplir las siguientes especificaciones, de lo contrario podría existir un descuento en la nota final:

- **Toda interacción con la base de datos debe ser mediante Querys en Python, inclusive el cargar los datos a la tabla Productos.**
- **El código debe ser realizado en Python utilizando la librería pyodbc .**
- **Debe** realizarse en parejas, no se aceptarán tareas individuales.
- En el foro de consultas podrán buscar pareja quienes no tengan, esto es exclusiva responsabilidad del estudiante.
- En caso de problemas con su pareja podrán contactar al profesor explicando su situación.
- La tarea debe ser entregada como un archivo .ZIP comprimido de la forma T1_ROL1_ROL2, este debe contener los archivos **.py** necesarios para el funcionamiento de su programa, además de un archivo README.txt el cual debe contener nombre, rol de los alumnos y las instrucciones para la correcta ejecución de su programa.
- Solo un alumno debe realizar la entrega.
- La entrega será vía AULA y el plazo máximo de entrega es hasta el **Viernes 14 de octubre de 2022 a las 23:55.**
- Las entregas con hasta 5 min. de atraso no tendrán descuento, posterior a las 00:00 existirá un descuento de 10 puntos por cada hora o fracción.
- Las funciones deberán ir comentadas, explicando clara y brevemente lo que realiza, los parámetros que recibe y el return en caso de ser necesario.
- Las copias serán evaluadas con nota 0 y se informarán a las respectivas autoridades.
- En caso que falle la ejecución de algún comando, no se asignará puntaje a éste.
- Las consultas sobre la tarea se deben hacer en las instancias de laboratorios. Consultas puntuales que no se hayan hecho en los laboratorios podrán hacerse a través de un foro creado para eso.
- Los laboratorios son instancia donde se explica y resuelven dudas sobre el enunciado de la tarea y además se hace una introducción a SQL.
- Ante cualquier duda sobre algo que no aparezca especificado en la tarea y no afecte las reglas ni la dificultad de esta, puede asumir lo que estime conveniente **pero debe ser especificado en el README.txt, en caso de que no se haga esto habrá descuento.**
- Esta evaluación debe ser defendida a través del servidor de Discord (el link se compartirá por aula los días previos a las defensas). Quienes no defiendan tendrán un 0 en su tarea 1.
- Existe la posibilidad de que a su defensa, asista su profesor.
- Cada grupo tendrá un horario definido para su defensa, en caso de atraso contarán con un tiempo menor para presentar su trabajo.
- La información respecto a la defensa será eventualmente publicada en Aula, esto considera el detalle sobre los descuentos. Es su obligación estar atento a esta información y cumplir con lo establecido allí.