

Proyecto final de la asignatura de bases de datos

Ing. Fernando Arreola

Noviembre 2022

Objetivo

El alumno analizará una serie de requerimientos y propondrá una solución que atienda a los mismos, aplicando los conceptos vistos en el curso.

Descripción del problema

El problema se divide en dos partes:

Parte uno:

Consiste en el diseño de una base de datos. Una mueblería quiere digitalizar su forma de operar para evitar tener sus registros de forma física y tener una vista completa de la información de todas sus sucursales, por lo que se plantea el siguiente requerimiento:

Se debe almacenar el código de barras, nombre, el precio de venta, el precio de compra, stock y fotografía de cada artículo, clasificándolos teniendo en cuenta que un artículo sólo pertenece a una categoría. Un artículo puede ser surtido por más de un proveedor, manteniendo registro de la fecha en la que se comenzó a surtir cada artículo y datos como el rfc, razón social, dirección, teléfono y cuenta para pago. De las ventas debe tenerse registro de su folio, fecha, monto total, monto por artículo, cantidad total de artículos, cantidad por artículo, quién concretó la venta y quién la cobró. Para facturación y programas de lealtad debe tenerse registro de datos de clientes, tales como rfc, nombre, razón social, dirección, email y teléfono. Debe tenerse una visibilidad completa de los empleados y el rol que desempeñan, por lo que debe asignarse un número de empleado (irrepetible) y tener registro de datos como el supervisor directo, rfc, curp, nombre, teléfonos, dirección, email, su fecha de ingreso y tipo de empleado (debe ser cajero, vendedor, administrativo, seguridad o limpieza; sólo puede ser un tipo). Respecto a las sucursales, debe tenerse conocimiento de su ubicación,

teléfono, año de fundación y debe considerarse que un empleado sólo trabaja en una sucursal. Adicionalmente, deben cumplirse los siguientes puntos:

- Cada que se agregue un artículo a una venta, debe actualizarse los totales (por artículo, venta y cantidad de artículos), así como validar que el artículo esté disponible.
- Crear al menos, un índice, del tipo que se prefiera y donde se prefiera. Justificar el porqué de la elección en ambos aspectos.
- Lista de artículos no disponibles o con stock < 3 . Si el artículo no está disponible, debe aparecer el mensaje "No disponible". Esta lista debe almacenarse en una tabla independiente, la cual debe actualizarse cada que se desee obtener esta lista.
- De manera automática se genere una vista que contenga información necesaria para asemejarse a un ticket de venta, incluyendo un folio para facturación (caracteres aleatorios irrepetibles).
- Al iniciar una venta, debe validarse que el vendedor y el cajero, pertenezcan a la misma sucursal, en caso de que no sea así, debe mostrarse un mensaje de error.
- Dada una fecha, o una fecha de inicio y fecha de fin, regresar el total vendido por sucursal y la cantidad de ventas efectuadas. Si no se especifica fecha, debe tomarse el día en que se hace la consulta.

Tomar en cuenta las siguientes consideraciones:

- Puede haber distintas soluciones al problema.
- Los requerimientos enlistados anteriormente, deberán ser realizados por medio de PostgreSQL, con los elementos que se consideren adecuados para resolverlos.
- Considerar que no necesariamente los clientes deben estar registrados para hacer una venta, pero la venta sí debe quedar registrada.
- El folio de la venta debe tener un formato similar a MBL-001, prefijo MBL, seguido de un guión y un número secuencial.
- Donde este presente el atributo *domicilio*, está compuesto por estado, código postal, colonia, calle y número.
- La razón social en el caso de los clientes es el nombre completo y debe ser un valor *default*.
- Donde este presente el atributo *nombre*, está compuesto por nombre, apellido paterno y materno.

- La propuesta de solución debe satisfacer todos los principios de diseño, los requerimientos anteriores y agregar las restricciones que se consideren necesarias para asegurar la consistencia e integridad de la información.

Parte dos:

Una vez diseñada y lista la base de datos, se debe crear una interfaz gráfica vía app móvil o web, que permita:

1. Consultar la información de un artículo, incluyendo su fotografía.
2. Ingresar los datos de un nuevo empleado.

Entregables

Es necesario entregar lo que se solicita a continuación:

Documento

Se deberá entregar un documento FORMAL elaborado en Latex, que contenga las siguientes secciones:

- Introducción: Breve descripción del análisis del problema, su(s) objetivos, su propuesta(s) de solución.
- Plan de trabajo: Queda a criterio de cada equipo el grado de detalle que se le dará a esta sección, pero mínimo debe contener descripción general de las actividades a realizar y su correspondiente plan de actividades, así como un breve detalle de qué hizo cada miembro del equipo.
- Diseño: Descripción de lo realizado en las correspondientes fases de diseño de las bases de datos, agregando los resultados de cada una de ellas.
- Implementación: Descripción de funcionamiento y código de los stored procedures, triggers, funciones, etc. Empleados para cumplir con los requerimientos del problema. También debe incluirse el DDL de la base de datos.
- Presentación: Descripción de lo que hace la modalidad seleccionada como forma de conexión hacia la base de datos.
- Conclusiones: Personales, detallando las dificultades, retos, aciertos, etc. que se presentaron en el proyecto.

Código

A través de github, se deberá crear una carpeta por equipo donde se anexe lo siguiente, asignando nombres ADECUADOS para cada documento/archivo:

- El documento del punto anterior
- Códigos fuente de MER, MR
- Script de creación de la base de datos y tablas
- Script para el agregado de información
- Script de toda la programación a nivel BD
- Códigos de lo implementado como parte de la etapa de presentación
- La presentación a emplear a la hora de exponer

Exposición

Como parte de la evaluación del curso, se deberá realizar una presentación del proyecto. La idea es simular la presentación de una solución de software a un posible cliente, lo que implica descripción del software, muestras de funcionalidad, solución de dudas, etc. Cada equipo es libre de decidir cómo hacer su presentación, el unico detalle es que debe ser TOTALMENTE formal.

Puntos a considerar

Se tomará en cuenta lo siguiente, partiendo del porcentaje de calificación estipulado al inicio del semestre:

- Equipos que se copien en alguna parte del proyecto, su calificación del curso será 5.
- Equipo(s) que reprueben el proyecto o la parte de exposición, deberán presentar examen final sin importar lo obtenido en los otros rubros del curso.
- La calificación del primer examen final será 30 % lo obtenido en el proyecto, 70 % lo obtenido en el examen.
- La seriedad, formalidad, presentación y calidad del proyecto, si bien no tendrán un valor numérico específico, pueden jugar a favor o en contra de la calificación de los rubros correspondientes.
- Cualquier punto no contemplado en el presente documento será determinado por el profesor.

Recomendaciones

No es obligatorio, pero se sugieren las siguientes puntos:

- Delegar correctamente responsabilidades
- No demorar en formar sus equipos y agregar su carpeta en GitHub
- Empleo de herramientas colaborativas
- Empleo de arquitecturas cliente servidor remotas o servicios en la nube
- Redacción y presentación en inglés