Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Laboratorio Sistemas de Bases de Datos 1
Ing. Luis Fernando Espino
Aux. Luis Danniel Castellanos

Enunciado

PROYECTO #2

Introducción

La integridad de los datos es un término usado para referirse a la exactitud y fiabilidad que estos adquieren. Los datos deben estar completos, sin variaciones o compromisos del original, que se considera confiable y exacto.

Al crear bases de datos, se debe prestar atención a la integridad de los datos y a cómo mantenerlos. Una buena base de datos hará cumplir la integridad de los datos siempre que sea posible.

Como parte del diseño es necesario crear procedimientos que permitan acceder de una manera más confiable a la base de datos, así como también garantizar que las transacciones se realicen correctamente.

Toda función, procedimiento almacenado y triggers, nos permiten poder administrar de una mejor manera cómo debe de comportarse una tabla de acuerdo con cada inserción, validación y disparador al momento de que ocurra cualquier evento o gestionemos nuestra base de datos con información nueva.

Objetivos

General:

- Adquirir nuevos conocimientos y destrezas en la gestión de una base de datos relacional para el uso profesional.

Específicos:

- Que el estudiante diseñe un sistema basado en un caso planteado.
- Aprender a utilizar procedimientos almacenados, funciones y triggers.
- Aplicar las primeras formas normales para la elaboración de un modelo relacional más optimizado.

Descripción del problema

Una cadena de restaurantes famosos de la localidad ha decidido incursionar hacia la transformación digital, por lo que está dispuesto a implementar una sistematización completa de todas sus operaciones, y de funcionalidades nuevas que servirán para el desarrollo de una nueva aplicación Delivery, para que los clientes puedan realizar sus pedidos en línea de manera fácil y sencilla. Para ello es necesario contar con una base de datos relacional que tenga un enfoque adecuado a la lógica de negocio del restaurante.

Se le requiere a usted, como estudiante del curso de Sistemas de Bases de Datos 1, para brindar una solución inicial en fase beta, a nivel de base de datos para que usted sea el encargado de proponer, diseñar e implementar toda la base de datos. Se le dará la libertad de crear sus propios procedimientos, funciones y triggers para el buen funcionamiento de la base de datos del restaurante; pero respetando las principales funcionalidades que se le requieren.

Requerimientos del proyecto

Diseño:

Se debe de realizar todo el proceso de modelado de la base de datos necesario para su creación. Lo cual consistirá en:

1. Modelo conceptual
2. Modelo lógico
3. Modelo relacional (Diagrama ER)

Funcionalidades:

La administración del restaurante ha brindado todos los requerimientos funcionales de manera detallada para que el estudiante los pueda implementar correctamente. Los cuales se detallan a continuación:

1. Registrar restaurante

Es importante poder tener la funcionalidad de registrar los restaurantes que ya se tienen y los nuevos que se podrán abrir en el futuro. Se debe de almacenar los datos del restaurante tales como: el identificador único de cada restaurante, la dirección en que se ubica, el número de teléfono, la cantidad de personal que tiene, y si cuenta con parqueo propio o no.

RegistrarRestaurante

Parámetro	Tipo de dato	Observación
Id	String	
Dirección	String	*Es la dirección completa
Municipio	String	*El parámetro vendrá con el nombre
Zona	Numérico	*Validar que sea entero positivo
Teléfono	Numérico	
Personal	Numérico	*Es el número máximo de personal
Tiene parqueo	Booleano	0=no, 1=sí

2. Crear empleado

Se debe de poder crear un nuevo empleado al momento de que es contratado por recursos humanos, se almacenarán todos sus datos personales, así como el puesto que posee y la fecha de inicio de labores. Se creará un identificador que es de tipo autoincremental que debe llevar exactamente 8 dígitos y que al inicio deberá preceder de dígitos cero. Por ejemplo: 00000001, 00000025, 00004597, 00123456, etc.

CrearEmpleado

Parámetro	Tipo de dato	Observación
Nombres	String	
Apellidos	String	
Fecha de nacimiento	Fecha	
Correo	String	*Validar que sea un formato válido
Teléfono	Numérico	
Dirección	String	
Número de DPI	Bigint	
Puesto	Numérico	*Es el id de puesto
Fecha de inicio	Fecha	
IdRestaurante	String	*Es el id de la función *RegistrarRestaurante*

3. Registrar puesto de trabajo

Sirve para agregar a la base de datos los distintos puestos de trabajo dentro del restaurante, cuyos identificadores serán de tipo numérico autoincremental (este id es el que se guarda en empleado). También se almacena una descripción de las funciones, y el salario del puesto.

RegistrarPuesto

Parámetro	Tipo de dato	Observación
Nombre	String	
Descripción	String	
Salario	Decimal	*Validar que sea positivo

4. Registrar cliente

Existirá dentro de la base de datos un registro de los clientes para poder acceder a su información sin tener que solicitarla cada vez que ordenan. Se sugiere que la llave primaria sea el número de DPI. Es importante mencionar que un cliente puede tener muchas direcciones asociadas, por lo que el registro de direcciones será otro procedimiento aparte.

RegistrarCliente

Parámetro	Tipo de dato	Observación
Número DPI	Bigint	*No pueden haber duplicados
Nombre	String	
Apellidos	String	
Fecha de nacimiento	Fecha	
Correo	String	*Validar que sea un formato válido
Teléfono	Numérico	
NIT	Numérico	*Puede ser nulo

5. Registrar dirección

Se utilizará para ir almacenando direcciones asociadas a los clientes. Por cada dirección se generará una llave numérica autoincremental. Para este caso es permitido guardar el municipio con letras completo, pero también se puede normalizar y guardar la referencia como número.

RegistrarDireccion

Parámetro	Tipo de dato	Observación
DPI cliente	Bigint	*Validar que el cliente exista
Dirección	String	*Va la dirección completa
Municipio	String	*El parámetro vendrá con el nombre
Zona	Numérico	*Validar que sea entero positivo

6. Crear orden

Se utiliza cuando un cliente comienza a realizar su pedido e indica en qué dirección desea recibir su orden. Durante este proceso se debe de crear una orden en la base de datos, el cual tendrá un identificador de tipo numérico autoincremental; este servirá para llevar un control del estado de la orden, la referencia del cliente y del detalle de productos. También se guarda el canal del pedido, en donde L=llamada y A=aplicación.

Es necesario contar con dos campos de fecha y hora exacta en el registro *(no vendrá dentro de los parámetros, sino que se genera con la función del gestor como GETDATE)*, en un campo se guarda el momento en que se inicia la orden y en otro el momento en que se entrega al cliente *(al momento de hacer la primera creación de la orden el campo de fecha final debería de ser NULL)*, ya que servirá para el de analista de datos.

Debe existir un campo para manejar el estado de la orden, la cual al momento de creación será de "INICIADA".

En este procedimiento, se debe hacer una validación importante, para poder atender el pedido es necesario validar que exista un restaurante en la misma zona y municipio que la dirección del cliente, y se debe guardar el id del restaurante que cubre la orden, de lo contrario no se puede continuar por falta de cobertura. Para este caso la orden quedará registrada en estado "SIN COBERTURA" y no se realiza nada más.

CrearOrden

Parámetro	**Tipo de dato**	**Observación**
DPI cliente	Bigint	*Validar que el cliente exista
Id Dirección cliente	Numérico	*Es el id de la entidad dirección
Canal	Char	*Dominio restringido '**L**','**A**'

7. Agregar ítems a la orden

El restaurante ofrece los combos del 1 al 6, extras del 1 al 3, bebidas del 1 al 5 y postres del 1 al 4. (Se adjunta en el apartado de **anexos** el detalle y precios).

Cuando un cliente está realizando un pedido, puede agregar n cantidad de productos por lo que es necesario manejarlo en un detalle. Se debe de especificar el tipo de producto, donde C=Combo, E=Extra, B=Bebida y P=Postre.

Si el estado de la orden se encontraba en "INICIADA" debe cambiarse por "AGREGANDO". No puede ampliarse una orden que ya haya sido finalizada o se encuentre en camino para entregarse al cliente.

AgregarItem

Parámetro	**Tipo de dato**	**Observación**
IdOrden	Numérico	*Validar que la orden exista y su estado sea válido.
Tipo producto	Char	*Dominio restringido '**C**','**E**','**B**','**P**'.
Producto	Numérico	*Validar que sea un producto que exista, por ejemplo: no existe un combo 7, extra 8, bebida -2, postre 5.
Cantidad	Numérico	*Validar que sea un entero positivo.
Observación	String	*Puede ir vacía, sirve para colocar alguna anotación como "Sin cebolla".

8. Confirmar orden

Se hace la confirmación cuando el cliente ya terminó de agregar productos para proceder con la entrega. El estado de la orden pasa a ser "EN CAMINO".

Además, se coloca el id del repartidor que está a cargo de entregar la orden.

Se genera automáticamente un encabezado de factura con lo siguiente:

- ➔ Número de serie: conformado por la concatenación del año en curso y el id de la orden.
- ➔ Monto total de la factura (suma de la orden + el IVA 12%).
- ➔ Lugar: municipio de la dirección del cliente
- ➔ Fecha y hora actual
- ➔ Id de la orden
- ➔ NIT del cliente, si en caso no tiene entonces colocar "C/F"
- ➔ Forma de pago, donde E=Efectivo, T=Tarjeta de débito o crédito.

ConfirmarOrden

Parámetro	Tipo de dato	Observación
IdOrden	Numérico	*Validar que la orden exista, y que su estado sea válido.
Forma de pago	Char	*Dominio restringido '**E**','**T**'.
IdRepartidor	Numérico	*Validar que el trabajador exista.

9. Finalizar orden

Se ejecuta cuando el repartidor ya ha entregado la orden en la dirección del cliente, el estado de la orden pasa a "ENTREGADA". Solamente se puede finalizar una orden cuyo estado sea "EN CAMINO". Además, se debe de colocar en el campo de fecha de entrega del pedido que se encontraba NULL, la fecha y hora exacta del momento en que se finaliza la orden.

FinalizarOrden

Parámetro	Tipo de dato	Observación
IdOrden	Numérico	*Validar que la orden exista, y que su estado sea válido.

Reportería:

El restaurante solicita que se extraiga la información necesaria a través de los datos almacenados con el fin de poder generar reportes y consultas necesarias para los distintos procesos administrativos y de análisis de datos.

Para ello es necesario elaborar los siguientes reportes:

1. Listar restaurantes

Debe retornar un listado de los restaurantes existentes y toda su información. Para el campo de si tiene parqueo debe mostrarse "Sí" o "No". Este procedimiento no lleva parámetros.

2. Consultar empleado

Debe retornar la información de un empleado según su código de personal.

ConsultarEmpleado

Parámetro	Tipo de dato	Observación
IdEmpleado	Numérico	*Si no existe mostrar error

Resultado esperado:

- → IdEmpleado
- → Nombre completo (concatenado)
- → Fecha de nacimiento
- → Correo
- → Teléfono
- → Dirección
- → Número de DPI
- → Nombre del puesto
- → Fecha de inicio
- → Salario

3. Consultar detalle del pedido

Debe retornar el detalle de ítems de una orden hecha por un cliente.

ConsultarPedidosCliente

Parámetro	Tipo de dato	Observación
IdOrden	Numérico	*Si no existe mostrar error

Resultado esperado:

- → Producto
- → Tipo producto: "Combo", "Extra", "Bebida" o "Postre".
- → Precio
- → Cantidad
- → Observación, si no tiene retorna vacío.

4. Consultar el historial de órdenes de un cliente

Debe retornar el listado de órdenes que ha realizado un cliente, en el detalle mostrar qué restaurante y repartidor atendió el pedido y a qué dirección fue enviado.

ConsultarHistorialOrdenes

Parámetro	Tipo de dato	Observación
DPI cliente	Numérico	*Si no existe mostrar error

Resultado esperado:

- → IdOrden
- → Fecha
- → Monto de la orden
- → Restaurante que cubrió la orden
- → Repartidor que hizo la entrega
- → Dirección que fue enviada
- → Canal: "Llamada" o "Aplicación"

5. Consultar direcciones de un cliente

Debe mostrar el conjunto de direcciones de un cliente.

ConsultarDirecciones

Parámetro	Tipo de dato	Observación
DPI cliente	Bigint	*Si no existe mostrar error

Resultado esperado:

- → Dirección completa
- → Municipio
- → Zona

6. Mostrar órdenes según estado

Debe retornar todas las órdenes según el estado que se indica en el parámetro.

MostrarOrdenes

Parámetro	Tipo de dato	Observación
Estado	Numérico	*Se enviará un entero, donde: 1 = INICIADA 2 = AGREGANDO 3 = EN CAMINO 4 = ENTREGADA -1 = SIN COBERTURA

Resultado esperado:

- → IdOrden
- → Estado
- → Fecha
- → DPI cliente
- → Dirección del cliente
- → Restaurante que cubre la orden
- → Canal: "Llamada" o "Aplicación"

7. Retornar encabezados de factura según el día

Debe retornar el listado de las facturas según la fecha indicada.

ConsultarFacturas

Parámetro	Tipo de dato	Observación
Día	Numérico	
Mes	Numérico	
Año	Numérico	

Resultado esperado:

- → Número de serie
- → Monto total de la factura
- → Lugar (municipio)
- → Fecha y hora
- → IdOrden
- → NIT del cliente (o C/F)
- → Forma de pago: "Efectivo" o "Tarjeta".

8. Consultar tiempos de espera

El restaurante busca ofrecer una política de comida gratis para las largas esperas, por lo que el analista de datos necesita conocer las estadísticas. Se envía como parámetro un número y se retornan todas las órdenes que tuvieron una demora mayor o igual a ese número de minutos.

ConsultarTiempos

Parámetro	Tipo de dato	Observación
Minutos	Numérico	*Será un entero positivo

Resultado esperado:

- → IdOrden
- → Dirección de entrega
- → Fecha y hora iniciada la orden
- → Tiempo de espera (en minutos)
- → Repartidor a cargo

Historial de transacciones

Se debe llevar un control de transacciones por medio de una tabla adicional en donde se registrará automáticamente (por medio de triggers) cada vez que ocurra una inserción, modificación o eliminación en una tabla.

Se debe de almacenar la fecha y hora exacta, y el nombre de la tabla que se vio afectada.

Tabla Ejemplo:

Fecha	Descripción	Tipo
YYYY-MM-DD hh:mm:ss	Se ha realizado una acción en la tabla "Nombre".	INSERT/UPDATE/DELETE

Anexos

Menú del restaurante		
Id	Producto	Precio
C1	Cheeseburger	Q.41.00
C2	Chicken Sandwinch	Q.32.00
C3	BBQ Ribs	Q.54.00
C4	Pasta Alfredo	Q.47.00
C5	Pizza Espinator	Q.85.00
C6	Buffalo Wings	Q.36.00
E1	Papas fritas	Q.15.00
E2	Aros de cebolla	Q.17.00
E3	Coleslaw	Q.12.00
B1	Coca-Cola	Q.12.00
B2	Fanta	Q.12.00
B3	Sprite	Q.12.00
B4	Té frío	Q.12.00
B5	Cerveza de barril	Q.18.00
P1	Copa de helado	Q.13.00
P2	Cheesecake	Q.15.00
P3	Cupcake de chocolate	Q.8.00
P4	Flan	Q.10.00

Entregables

- Modelo conceptual, lógico y relacional
- Script de la base de datos y datos de carga
- Funciones, procedimientos y triggers que hayan sido implementados

Restricciones

- El gestor de base de datos a utilizar puede ser **MySQL**, **Oracle** o **SQL Server**.
- El proyecto es individual.
- Sistema operativo libre.
- Pueden utilizar Docker o una instancia local para su base de datos.
- Todo puede ser presentado desde su gestor nativo o IDE.
- El código y todo lo relacionado a su proyecto debe alojarse en un repositorio de **GitHub** agregando a su auxiliar como colaborador: ***luisdacg*.**
- En caso de copias totales o parciales será anulado y reportado a escuela.

Entrega

- Fecha límite: viernes 5 de mayo de 2023 hasta las 23:59 hrs.
- La entrega se hará vía UEDi por medio del enlace a su repositorio.