Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Práctica 1

Escuela de Ciencias y Sistemas

Sistemas de Bases de Datos 2

Ing. Luis Alberto Arias Solórzano

Ing. Otto Amílcar Rodríguez Acosta

Aux. Diego André Mazariegos Barrientos

Aux. Emiliano José Alexander Velásquez Najera

Aux. Gerson Aaron Quinia Folgar

2do. Semestre 2024

OBJETIVOS

Objetivo general

Comprender y emplear el uso correcto de funciones, procesos, triggers y transacciones para proporcionar secuencias de trabajo fiables, así mismo el conjunto de herramientas que nos proveen los DBMS para ejecutar código dentro de nuestra base de datos.

Objetivos específicos

- Utilizar transacciones para los procesos que modifican el estado de nuestra base de datos (Insert, Update, Delete).
- Conocer las bondades que ofrece Transact-SQL como variante mejorada del estándar por parte de SQL Server.
- Garantizar las propiedades básicas de una transacción (ACID).
- Llamadas de transacciones por medio de procedimientos almacenados.
- Implementación de disparadores para ejecutar acciones al momento de que se produzcan eventos en la base de datos.

ENUNCIADO

Descripción general

Actualmente la Facultad de Ingeniería de la Universidad San Carlos de Guatemala se ha encontrado con el problema que algunos procesos dentro de su modelo de base de datos no son 100% fiables garantizados, debido a que el modelo ha sido modificado a lo largo del tiempo y se le han agregado más tablas al modelo inicial y no se ha garantizado la atomicidad en algunos procesos de inserciones.

Por ejemplo, se ha encontrado que hay usuarios que no poseen ningún rol, asignaciones donde a los tutores no se les informa los estudiantes que tienen asignados e incluso asignaciones con datos incorrectos. Por lo tanto, se le solicita a usted como Administrador de Base de Datos que pueda garantizar que todos los procesos que se realicen cumplan las 4 propiedades básicas de una transacción provocando inserciones, actualizaciones y eliminaciones exitosas o si ocurre algún error la base de datos finalice en un estado fiable y que no se vea comprometida la información que esté almacenando.

Problemas a resolver

A continuación, se le detallan algunos problemas esenciales iniciales que la Universidad desea resolver:

1. **Manejo de un historial**
   Inicialmente se le solicita que cada transacción que se realice en la base de datos debe de ser registrada a manera de tener una bitácora de todos los procesos que se hayan realizado, para ello se debe de implementar una serie de triggers que se ejecuten luego de las siguientes operaciones:

   - *DELETE*
   - *UPDATE*
   - *INSERT*

- **Entidades Implicadas**:
  - *Todas las necesarias*

2. **Creación de roles para estudiantes**

   Actualmente se necesita crear los respectivos roles para los estudiantes que se van a inscribir en la Facultad, para ello se debe implementar un procedimiento almacenado y es necesario que se tengan en cuenta los siguientes roles:

   - *Student*
   - *Tutor*

   - **Entidades Implicadas**:
     - *Roles*

3. **Creación de cursos**

   Se le pide que se creen los cursos que se van a impartir en las distintas escuelas de la Facultad, para ello deben implementar un procedimiento almacenado para realizar dicho requerimiento.

   - **Entidades Implicadas**:
     - *Course*

4. **Registro de Usuarios**

   Actualmente se necesita que cuando un estudiante sea registrado en el sistema se le reconozca internamente con el rol de estudiante y las acciones que haga dentro del sistema sean basadas en su perfil de estudiante. Las cuentas de los usuarios tienen la opción de activar un segundo factor de autenticación por lo tanto cuando el usuario se registre se debe llevar el control si esta opción la tiene activa o no. Por último, se le enviará un correo de notificación al usuario que fue registrado en el sistema.

   - **Entidades Implicadas**:
     - *Usuarios*
     - *Usuariorole*
     - *ProfileStudent*

- *TFA*
- *Notification*

- **Restricciones**: Se debe garantizar que el email no esté asociado con ninguna otra cuenta dentro del sistema. El campo EmailConfirmed es Falso o 0,

5. **Cambio de Roles**

   Algunos estudiantes pueden tomar el rol de tutor ya que la Universidad les da la oportunidad de realizar prácticas finales como auxiliar de cátedra. Por lo tanto, se necesita un proceso mediante el cual el estudiante se le agregue junto a su rol de estudiante un rol de tutor y así mismo se le asigne un perfil de tutor con el cual podrá hacer acciones dentro del sistema. Cuando este proceso se lleve a cabo se debe especificar a qué curso se le asigna como tutor. Por último, se le enviará un correo de notificación al usuario que fue promovido con el rol de tutor.

   - **Entidades Implicadas**:
     - *UsuarioRole*
     - *TutorProfile*
     - *CourseTutor*
     - *Notification*

   - **Restricciones**: Se debe garantizar que el usuario sea un usuario con cuenta activa (se comprueba por medio del campo EmailConfirmed).

6. **Asignación de Cursos**

   Cuando se realiza una asignación se necesita garantizar que esta sea exitosa y fiable ya que ciertos cursos tienen límite de cantidad de estudiantes lo cual ha sucedido que hay algunos casos con más estudiantes del cupo asignado. Por lo tanto, cuando se haga la asignación se necesita que se le notifique al estudiante que fue asignado y también al tutor para que pueda llevar el control de sus estudiantes.

   - **Entidades Implicadas**:
     - *CourseAssignment*
     - *Notification*

- **Restricciones**: Se debe garantizar que la cantidad de créditos que tenga el usuario cumpla los créditos necesarios del curso por llevar.

7. **Validación de datos**

   Al momento de una inserción de información es necesario que dichos datos sean correctos y acordes a la restricción de los campos y el tipo de datos que estén manejando, para ello se le solicita que se cree un procedimiento que maneje la actualización de tablas (Constraints) para cerciorarse que se ingresen datos correctos.

   - **Entidades Implicadas y campos**:
     - *Usuarios*
       - *FirstName, LastName: Validar que solamente sean ingresados letras.*
     - *Course*
       - *Name: Validar que solamente sean ingresados letras.*
       - *CreditsRequired: Validar que solamente sean ingresados números.*

8. **Funciones de verificación**

   A fin de verificar que se tenga la correcta inserción de datos se le solicita al estudiante que implemente una serie de funciones que recibirán determinados parámetros para devolver información conjunta almacenada en las tablas de la base de datos. Las funciones que se piden son las siguientes:

   - *Func_course_usuarios: Función que retornará el listado completo de alumnos que están asignados a un determinado curso.*
   - *Func_tutor_course: Función que retornará la lista de cursos a los cuales los tutores estén designados para dar clase.*
   - *Func_notification_usuarios: Función que retornará la lista de notificaciones que hayan sido enviadas a un usuario.*
   - *Func_logger: Función que retornará la información almacenada en la tabla HistoryLog.*

- *Func_usuarios: Función que retornará el expediente de cada alumno, que incluye los siguientes campos:*
    - *Firstname*
    - *Lastname*
    - *Email*
    - *DateOfBirth*
    - *Credits*
    - *RoleName*

Consideraciones importantes

- No se pueden agregar ni tablas ni campos nuevos del modelo dado.
- Se llevará un historial de las acciones que se realizan dentro de la base de datos por lo cual si una transacción termina de manera exitosa se debe escribir en la tabla de History Log que la transacción fue exitosa y si falla se deberá escribir que la transacción fue fallida.
- No se debe dejar ningún campo vacío.
- Cada transacción será almacenada en su respectivo procedimiento almacenado, función o trigger y cada uno tendrá el nombre y los campos especificados:

Transacción	Nombre Procedimiento	Parámetros
Registro de Usuarios	PR1	PR1 (Firstname, Lastname, Email, DateOfBirth, Password, Credits) Nota: EmailConfirmed será 1
Cambio de Roles	PR2	PR2 (Email, CodCourse)
Asignación de Curso	PR3	PR3 (Email, CodCourse)
Creación de roles para estudiantes	PR4	PR4 (RoleName)
Creación de Cursos	PR5	PR5 (Name, CreditsRequired)
Validación de Datos	PR6	PR6

Función	Nombre Función	Parámetros

Func_course_usuarios	F1	F1 (CodCourse)
Func_tutor_course	F2	F2 (Id [TutorProfile])
Func_notification_usuarios	F3	F3 (Id [Usuarios])
Func_logger	F4	F4
Func_usuarios	F5	F5 (Id [Usuarios])

Trigger	Nombre Trigger	Disparador
History Log	Trigger1	Update, Insert, Delete

ENTREGA

- La entrega será el 17 de Agosto, a más tardar a las 11:59 pm.
- Se debe de exportar la Base de datos y crear un repositorio en Github con nombre BD2S22023_Grupo_#.
- Se debe crear una carpeta en dicho repositorio con el nombre “Practica1”.
- En el repositorio deben incluir el script de la base de datos (script.sql) y el archivo con el código hecho por el estudiante donde estará el DDL de los procedimientos, triggers y funciones creadas.
- Para entregar el proyecto en UEDI se deberá subir un archivo de texto con el link del repositorio, nombre del archivo BD2_Practica1_#Carnet|Grupo.
- Copias totales o parciales entre integrantes del laboratorio o bajadas de internet obtendrán nota de 0 puntos.
- No se recibirán entregas tardes.
- Agregar al respectivo auxiliar al repo.