



Universidad Nacional
del Nordeste

LICENCIATURA EN SISTEMAS DE INFORMACIÓN

BASE DE DATOS I

PROYECTO INTEGRADOR: CONSORCIO

GRUPO 6 – COMISIÓN 3

PROFESOR:

VALLEJOS, WALTER OSCAR

ALUMNOS:

RODRIGUEZ, JOSEFINA FERNANDA

LOPEZ, ENZO FABIÁN

SACK CEPPI, GUILLERMO HUGO

AÑO 2023

Contenido

CAPITULO I: INTRODUCCION.....	3
1.1 TEMA	3
1.2 PLANTEAMIENTO DEL PROBLEMA	3
1.3 OBJETIVO DEL TRABAJO.....	3
CAPITULO II: MARCO CONCEPTUAL Y REFERENCIAL.....	6
CAPITULO III: METODOLOGIA	6
CAPITULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DERESULTADOS.....	7
4.1 MODELO DE DATOS.....	8
4.2 DICCIONARIO DE DATOS.....	11
CAPITULO V: TEMAS DE INTEGRACIÓN DE OTROS GRUPOS / RESUMEN	15
5.1 BACKUP Y RESTORE - GRUPO 1	15
5.2 MANEJO DE PERMISOS A NIVEL DE USUARIO DE BASE DE DATOS –	15
GRUPO 7	15
CAPITULO VI: CONCLUSIÓN	17
BIBLIOGRAFIA	17

CAPITULO I: INTRODUCCION

Un sistema de base de datos, es en general, un conjunto de información relevante organizada de forma sistemática que representan entidades y sus interrelaciones, los datos almacenados pueden ser de diversa índole, generalmente está estructurada por tablas y estas a su vez por campos y registros, además contiene procedimientos necesarios para la administración de los datos (Triggers, etc.).

Las bases de datos son importantes para la toma de decisiones y para ejercer acciones.

1.1 TEMA

El tema de investigación que se desarrollara es triggers o disparador, siendo éste un script que se usa en lenguaje de programación SQL.

Consiste en una serie de reglas predefinidas que se asocian a una tabla. Estas reglas se aplican a la base de datos cuando se realizan determinadas operaciones en la tabla, por ejemplo, al añadir, actualizar o eliminar registros.

Dicho de otra manera, el trigger desencadena determinadas acciones de forma automática en las tablas de la base de datos cuando se insertan, modifican y se añaden nuevos datos. Generalmente es muy utilizado para auditorias en empresas

Por otra parte, entre sus principales ventajas es que todas estas funciones se pueden realizar desde la propia base de datos, es decir, no es necesario recurrir a lenguajes externos de programación.

1.2 PLANTEAMIENTO DEL PROBLEMA

En el manejo de bases de datos es muy frecuente realizar operaciones como insertar, modificar o borrar registros de tablas en nuestra base de datos y de forma involuntaria cometer errores. Por ello, se precisa el uso de procedimientos de control del manejo de estos datos a través de triggers o disparadores.

1.3 OBJETIVO DEL TRABAJO

I. OBJETIVOS GENERALES

Mejorar la administración de la Base de datos, sin necesidad de contar con que el usuario ejecute la sentencia de SQL, es decir de forma automatizada, ayudando a, maximizar el tiempo de ejecución de las acciones, aumentar la seguridad e integridad de la información.

II. OBJETIVOS ESPECÍFICOS

Utilizaremos en este proyecto triggers (disparadores) de auditorías por cada insert, update y delete sobre la tabla conserje, que a su vez de forma automática registrara esos movimientos en una tabla auxiliar llamada auditoria_conserje, incorporando la fecha y hora en que se registró la acción, usuario de la base de datos que realizó la operación y la acción que realizo.

Pasos a seguir para la creación y utilización del trigger.

1. Creo la tabla auxiliar auditoria_conserje, la cual tendrá los mismos atributos que la tabla conserje, agregando tres atributos más los cuales serán explicados en el diccionario de datos

```
create table auditoria_conserje(  
idconserje int,  
apeynom varchar(50),  
tel varchar(50),  
fechnac date,  
estciv varchar (1),  
fechaModif datetime,  
usuario_accion varchar(100),  
accion varchar (20)  
);
```

2. Luego creo los triggers, para que sean activados de forma automáticamente al realizar un insert, update o delete sobre la tabla conserje. Estos triggers pueden hacerlos todo juntos, pero a modo de mejor comprensión de los hará de forma separada

```
create trigger trg_auditoria_conserje_insertar  
on conserje for insert  
as  
declare @idconserje int, @apeynom varchar(50), @tel varchar(50), @fechnac date,  
@estciv varchar (1),  
@fechaModif date,@usuario varchar(50),@accion varchar(20)  
select @idconserje= idconserje,@apeynom=ApeyNom,  
@tel=tel,@fechnac=fechnac,@estciv=estciv,  
@fechaModif=getdate(),@usuario=SYSTEM_USER from inserted  
insert into auditoria_conserje values (@idconserje, @apeynom,@tel, @fechnac,  
@estciv,  
getdate(),@usuario,'accion insert')
```

```
create trigger trg_auditoria_conserje_modificar  
on conserje for update  
as  
declare @idconserje int, @apeynom varchar(50), @tel varchar(50), @fechnac date,  
@estciv varchar (1),  
@fechaModif date,@usuario varchar(50)  
select @idconserje= idconserje,@apeynom=ApeyNom,  
@tel=tel,@fechnac=fechnac,@estciv=estciv,  
@fechaModif=getdate(),@usuario=SYSTEM_USER from inserted  
insert into auditoria_conserje values (@idconserje, @apeynom,@tel, @fechnac,  
@estciv,  
getdate(),@usuario,'accion modify')
```

```

create trigger trg_auditoria_conserje_borrar
on conserje for update
as
declare @idconserje int, @apeynom varchar(50), @tel varchar(50), @fechnac date,
@estciv varchar(1),
@fechaModif date, @usuario varchar(50)
select @idconserje= idconserje, @apeynom=ApeyNom,
@tel=tel, @fechnac=fechnac, @estciv=estciv,
@fechaModif=getdate(), @usuario=SYSTEM_USER from deleted
insert into auditoria_conserje values (@idconserje, @apeynom, @tel, @fechnac,
@estciv,
getdate(), @usuario, 'accion delete')

```

3. Para corroborar si los triggers están activados sobre la tabla conserje ejecuto la siguiente sentencia:

```
sp_helptrigger conserje
```

4. Luego pruebo con datos las diversas sentencias

```

--insertar
Insert into conserje (ApeyNom,tel,fechnac,estciv) values ('carmelo encinas.',
'000000001', '19870223', 'C');
select* from auditoria_conserje;

--actualizacion
UPDATE conserje SET ApeyNom = 'gerardo edgar' where tel = '000000001';
select* from auditoria_conserje;

--borrado
delete from conserje where tel = '000000001';

```

A su vez definiremos un trigger que al intentar realizar un delete sobre la tabla (administrador) emita un mensaje y no lo permita.

```

--no permitir eliminar registros de la tabla administrador

create trigger trigger_administrador_no_borrar
ON administrador
INSTEAD OF DELETE
AS
    if (select count(*) from deleted) > 0
    begin

        raiserror('No se puede eliminar ningun registro de la tabla
administrador',16,1)
        rollback transaction

    end;

DELETE FROM administrador
WHERE idadmin = 1

```

CAPITULO II: MARCO CONCEPTUAL Y REFERENCIAL

La investigación que se desarrolla en el presente informe se basa en los conceptos básicos que rodean al manejo de base de datos, que nos enmarcan en un camino de construcción y comprensión de lo trabajado. Por ello, debemos conocer con que estamos trabajando y para qué.

Considerando que en la actualidad se requiere de la manipulación sistemática de grandes masas de información y con ello de datos. Una base de datos ayuda al registro de los mismos, análisis, manejo, recopilación de éstos de una forma automatizada con ventajas como ser:

- Mejor organización de datos en tablas
- Mayor eficacia en el manejo/manipulación de datos:
- Conexión de las tablas con claves.

El tema en el que ahondamos en el proyecto. "Triggers de auditoría" el la cual utilizamos la base de datos "Consortio"

- trabaja sobre el control de insertar, eliminar y modificar datos en dicha base, detallando las sentencias utilizadas, las cuales hemos ido recopilando de la bibliografía, internet, videos, etc. Lo cual genera una mejor comprensión de lo que efectivamente estamos aplicando, tanto para los que desarrollamos el tema como para quien desea conocerlo a través del proyecto.

CAPITULO III: METODOLOGIA

La metodología empleada para el desarrollo del proyecto es la conocida como SCRUM, dado que es la más optima de acuerdo al desenvolvimiento del desarrollo y organización de las tareas realizar. Por su esquema estructural de rapidez, eficiencia y productividad es una de las mas utilizada en el mundo actual, y en nuestro caso al ser un proyecto chico y de pocos integrantes, fue fácil emplearla en cuanto se refiere al trabajo grupal y colaborativo.

Cuando hablamos de SCRUM, sabemos que ella se basa en la utilización de Sprint, siendo estos periodos breves de tiempo fijo en el que nuestro equipo de scrum trabajo para completar una cantidad de trabajo establecida. Nuestro proyecto esta dividido en dos Sprint, los cuales siguen una serie de fases para el desarrollo de cada tarea.

Para el desarrollo de cada tarea se desarrollaron cuatro fases para su mejor comprensión y desarrollo a la hora de distribuir las tareas a realizar.

Sprint Planning

Esta es la primera fase de la metodología Scrum y se basa en la planificación, es decir, se describen las tareas a realizar y a su vez se designa quien la llevara a cabo y se establece el pazo para su ejecución.

Scrum team meeting

El grupo de trabajo Scrum, que somos los integrantes del grupo, hacíamos videollamadas diarias, para hacer conocer la tarea que se debía hacer, los resultados que se esperaban lograr, el tiempo con el que contaba cada uno para resolver la tarea a realizar y luego de ser necesarias llamadas de emergencia para

comentar dificultades, para luego acordar reuniones presenciales para resolverlas y mostrar los avances.

Backlog refinement

Al tener directivas de la catedra del trabajo a realizar y lo que se esperaba como producto final, una base de datos consistente que pueda realizar diferentes tipos de funcionalidades, entre ellas triggers que nos fue asignado, optamos por adecuar la realización de tareas llegando a tiempo con las fechas asignadas por la catedra de entrega. Utilizando herramientas como github para subir en ese repositorio lo investigado sobre el tema tanto por medio de un informe detallado del proyecto como su correspondiente desarrollo en sql. Además de en el código sql, utilizar comentarios y señaladores para mejorar la comprensión del código.

Sprint review

Son reuniones que teníamos tanto presencialmente como de modo virtual para mostrar los resultados de las tareas que cada uno tenía asignado realizar, en caso de problemas solucionarlos entre todos, para seguir con el desarrollo del proyecto real y avanzar.

CAPITULO IV: DESARROLLO DEL TEMA / PRESENTACIÓN DE RESULTADOS

A lo largo del proyecto se habla de trigger, siendo este nuestro tema de investigación, para ello debemos tener en cuenta que Un trigger o disparador es un script que se usa en lenguaje de programación SQL. Dicho de otra manera, el trigger desencadena determinadas acciones de forma automática en las **tablas** de la base de datos cuando se insertan, modifican y se añaden nuevos datos.

Los trigger se pueden ejecutar cuando el usuario realizar alguna acción relacionada con añadir, actualizar o eliminar información de una tabla. Generalmente es utilizada en millones de empresas en todo el mundo como un sistema de auditorias de tablas o incluso de permisos para ciertas acciones al utilizar comandos como INSERT, UPDATE o DELETE. Por tanto, para poder usar un trigger es necesario que el usuario posea permisos para hacerlas en la base de datos.

La estructura y modo de **funcionamiento de un trigger** se puede resumir en tres pasos:

Se produce una llamada de activación al código que se ha de ejecutar.

Aplica las restricciones necesarias para poder realizar la acción, por ejemplo, una determinada condición o una nulidad.

Una vez verificadas las restricciones, se ejecuta la acción, en base a las instrucciones recibidas en el primer punto.

Existen diferentes **tipos de disparadores**, en función de las ejecuciones que realizan.

Disparadores de fila: también llamados row triggers, son aquellos cuya ejecución se realiza a través de llamadas desde una tabla asociada al trigger. Estos son los que se desarrollaran más adelante.

Disparadores de secuencia: también llamados statement triggers, son aquellos que se ejecutan solo una vez, independientemente de la cantidad de veces que se cumplan las condiciones para su ejecución.

Los triggers pueden ejecutar su acción en diferentes momentos.

Antes de ejecutar la sentencia (**before statement**)

Después de ejecutar la sentencia y de comprobar las restricciones y condiciones aplicables (**after statement**)

Antes de modificar la fila de la tabla afectada por la sentencia del trigger, y de comprobar las restricciones y condiciones de ejecución (**before row**)

Después de modificar la fila de la tabla afectada por la sentencia del trigger, y de comprobar las restricciones y condiciones de ejecución (**after row**)

4.1 MODELO DE DATOS

NOMBRE	ATRIBUTOS	TIPO
gasto	idgasto	Entero
	idprovincia	Entero
	idlocalidad	Entero
	idconsorcio	Entero
	periodo	Entero
	fechapago	Fecha
	idtipogasto	Entero
	importe	Decimal

tipo gasto	idtipogasto descripción	entero Alfanumérico (50)
consorcio	idprovincia	Entero
	idlocalidad	Entero
	idconsorcio	Entero
	nombre	Alfanumérico (50)
	dirección	Alfanumérico (50)
	idzona	Entero
	idconserje	Entero
	idadmin	Entero

provincia	idprovincia	Entero
	descripción	Alfanumérico (50)
	km2	Entero
	cantdptos	Entero
	nomcabe	Alfanumérico (50)
	población	Entero
localidad	Idprovincia	Entero
	Idlocalidad	Entero
	Descripción	Alfanumérico (50)
zona	idzona	Entero
	descripcion	Alfanumérico (50)
conserje	idconserje	Entero
	apeynom	Alfanumérico (50)
	tel	Alfanumérico (20)
	fechnac	Fecha
	estciv	Alfanumérico (1)
administrador	Idadmin	Entero
	Apeynom	Alfanumérico (150)
	Viveahi	Alfanumérico (1)
	Tel	Alfanumérico (20)
	Sexo	Alfanumérico (1)
	Fechnac	fecha
auditoria_conserje	idconserje	Entero
	apeynom	Alfanumérico (50)

	tel	Alfanumérico (20)
	fechnac	Fecha
	estciv	Alfanumérico (1)
	fechaModif	Fecha
	usuario_accion	Alfanumérico (100)
	accion	Alfanumérico (50)

4.2 DICCIONARIO DE DATOS

Tabla: provincia			Fecha	30/10/2023
Descripcion: Contiene datos de todas las provincias del pais.				
Campo	Tamaño	Tipo de dato	Descripcion	
idprovincia		numerico	identificador numerico que referencia a cada provincia del pais	
descripcion	50	alfanumerico	nombre de la provincia	
km2		numerico	representa dimensiones en kilometros cuadrados de la provincia	
cantdptos		numerico	cantidad de departamentos que posee la provincia	
poblacion		numerico	cantidad de poblacion en la provincia	
nomcabe	50	alfanumerico	nombre de la capital de la provincia	

Tabla: localidad			Fecha	30/10/2023
Descripcion: Contiene datos de todas las localidades del pais, organizadas por provincias				
Campo	Tamaño	Tipo de dato	Descripcion	
idprovincia		numerico	identificador numerico que referencia a cada provincia del pais	
idlocalidad		numerico	identificador numerico que referencia a cada localidad de una provincia	
descripcion	50	alfanumerico	nombre de la localidad	

Tabla: zona			Fecha	30/10/2023
Descripcion: Contiene datos de una division territorial zoificada de las localidades.				
Campo	Tamaño	Tipo de dato	Descripcion	
idzona		numerico	identificador numerico que referencia a cada zona del pais	
descripcion	50	alfanumerico	nombre de la zona del pais	

Tabla: conserje			Fecha	30/10/2023
Descripcion: Se registran datos de las personas que actuan como conserjes de cada edificio.				
Campo	Tamaño	Tipo de dato	Descripcion	
idconserje		numerico	identificador numerico que referencia a cada conserje	
apecnom	50	alfanumerico	nombre y apellido completo del conserje	
tel	20	alfanumerico	numero telefonico del conserje. Telefono fijo o celular	
fechnac		fecha	fecha de nacimiento del conserje	
estciv	1	alfanumerico	estado civil del conserje. Solo valores "C", "S", "D", "V", "O"	

Tabla: administrador			Fecha	30/10/2023
Descripcion: Se registran datos de las personas que actuan como administradores de cada edificio. Se registra si el administrador vive en edificio el edificio				
Campo	Tamaño	Tipo de dato	Descripcion	
idadmin		numerico	identificador numerico que referencia a cada administrador	
apecnom	50	alfanumerico	nombre y apellido completo del administrador	
viveahi	1	alfanumerico	identifica si el administrador vive o no en el edificio. Solo valores "S" o "N"	
tel	20	alfanumerico	numero telefonico del administrador. Telefono fijo o celular	
sexo	1	alfanumerico	identifica si el administrador es hombre o mujer. Solo valores "M" o "F"	
fechnac	50	alfanumerico	fecha de nacimiento del administrador	

Tabla: consorcio			Fecha	30/10/2023
Descripcion: Contiene los datos de cada uno de los consorcios (edificios) que son administrados.				
Campo	Tamaño	Tipo de dato	Descripcion	
idprovincia		numerico	identificador numerico que referencia a cada provincia del pais	
idlocalidad		numerico	identificador numerico que referencia a cada localidad de una provincia	
idconsorcio		numerico	identificador numerico que referencia a cada consorcio (edificio)	
nombre	50	alfanumerico	nombre del consorcio (edificio)	
direccion	50	alfanumerico	direccion del consorcio (edificio). Esta compuesta por la calle y numero	
idzona		numerico	identificador numerico que referencia a cada zona del pais	
idconserje		numerico	identificador numerico que referencia a cada conserje	
idadmin		numerico	identificador numerico que referencia a cada administrador	

Tabla: gasto			Fecha	30/10/2023
Descripcion: Se registran los gastos de cada edificio, segun el tipo de gasto.				
Campo	Tamaño	Tipo de dato	Descripcion	
idgasto		numerico	identificador numerico que referencia el gasto	
idprovincia		numerico	identificador numerico que referencia a cada provincia del pais	
idlocalidad		numerico	identificador numerico que referencia a cada localidad de una provincia	
idconsorcio		numerico	identificador numerico que referencia a cada consorcio (edificio)	
periodo		numerico	periodo de pago	
fechapago		fecha	fecha en que figura el pago del gasto	
idtipogasto		numerico	identificador numerico que referencia el tipo de pago	
importe	8,2	decimal	importe a pagar de acuerdo al periodo	

Tabla: tipogasto			Fecha	30/10/2023
Descripcion: Contiene los tipos de gastos que se pueden registrar en cada consorcio.				
Campo	Tamaño	Tipo de dato	Descripcion	
idtipogasto		numerico	identificador numerico que referencia el tipo de pago	
descripcion	50	alfanumerico	nombre del tipo de gasto	

Tabla:inmueble			Fecha	30/10/2023
Descripcion: contiene datos propios del edificio				
Campo	Tamaño	Tipo de dato	Descripcion	
idprovincia		numerico	identificador numerico que referencia a cada provincia del pais	
idlocalidad		numerico	identificador numerico que referencia a cada localidad de una provincia	
idconsorcio		numerico	identificador numerico que referencia a cada consorcio (edificio)	
sup		decimal	superficie total. cubierto, que ocupan todos los departamentos del inmueble	
nro_pisos		numerico	cantidad de pisos que tiene el inmueble	
cant_dpto		numerico	cantidad de departamentos en el inmueble	
espacio_comun		numerico	si cuenta con espacios de uso comun (0-no, 1- si)	

Tabla:auditoria_conserje			Fecha	30/10/2023
Descripcion: tabla auxiliar que sirve de auditoria a la tabla conserje				
Campo	Tamaño	Tipo de dato	Descripcion	
idconserje		numerico	identificador numerico que referencia a cada conserje	
apeynom	50	alfanumerico	nombre y apellido completo del conserje	
tel	20	alfanumerico	numero telefonico del conserje. Telefono fijo o celular	
fechnac		fecha	fecha de nacimiento del conserje	
estciv	1	alfanumerico	estado civil del conserje. Solo valores "C", "S", "D", "V", "O"	
fechaModif		fecha	fecha actual del sistema en la cual se realizo la accion	
usuario_accion	100	alfanumerico	toma del sistema el usuario que hace la carga del registro	
accion	50	alfanumerico	cadena que define que accion se realizo	

CAPITULO V: TEMAS DE INTEGRACIÓN DE OTROS GRUPOS / RESUMEN

5.1 BACKUP Y RESTORE - GRUPO 1

La copia de seguridad de las bases de datos, la ejecución de procedimientos de restauración de prueba de las copias de seguridad y el almacenamiento de las copias en una ubicación segura y fuera del sitio contribuyen a proteger los datos ante posibles accidentes. Las copias de seguridad son la única forma de proteger los datos.

Con las copias de seguridad válidas de una base de datos se puede recuperar los datos en caso de que se produzcan errores, por ejemplo:

- Errores de medios.
- Errores de usuario. Por ejemplo, quitar una tabla por error.
- Errores de hardware. Por ejemplo, una unidad de disco dañada o la pérdida permanente de un servidor.
- Desastres naturales.

En primer lugar, se inició realizando una copia de seguridad completa de la base de datos utilizando la instrucción "BACKUP DATABASE" para respaldar todos los datos y objetos asociados. Para poder hacer esto, primero se tuvo que verificar que el modo de recuperación de bases de datos se encuentre en el modo adecuado para realizar un backup en línea. Por ello, se cambió dicho modo al estado "FULL" mediante la instrucción "SET RECOVERY FULL;". Luego, se estableció una estrategia de respaldo para asegurar que los cambios posteriores se capturarán de manera efectiva. En caso de una falla o pérdida de datos, el proceso de restauración se lleva a cabo mediante la instrucción "RESTORE DATABASE", que recupera la base de datos a un estado consistente con el último backup válido. Es fundamental documentar y automatizar este proceso para garantizar una recuperación eficiente y minimizar el tiempo de inactividad en situaciones críticas.

La parte práctica se mostrara en el archivo sql para su mayor entendimiento y ejecución.

5.2 MANEJO DE PERMISOS A NIVEL DE USUARIO DE BASE DE DATOS – GRUPO 7

Este tema se centra en la investigación del problema de cómo se manejan los permisos a nivel de usuario en una base de datos.

Un "rol" es un conjunto de derechos predefinidos que se asignan a un grupo de usuarios en una base de datos. Siendo útil para simplificar la gestión de permisos y accesos en lugar de otorgar permisos individuales a cada usuario en la base de datos.

Definición de roles de usuario en bases de datos: Los roles de usuario en las bases de datos son entidades de seguridad que agrupan a otras entidades de seguridad, similares a los grupos del sistema operativo Microsoft Windows. Un rol es un objeto de base de datos que agrupa uno o más privilegios y puede ser asignado a un usuario. Los roles de nivel de base de datos se aplican a toda la base de datos en lo que respecta a su ámbito de permisos.

Tipos de roles: Existen dos tipos de roles en el nivel de base de datos: los roles fijos de base de datos que están predefinidos en la base de datos y los roles de base de datos definidos por el usuario que el usuario puede crear.

Roles fijos en la base de datos:

db_owner: pueden realizar todas las actividades de configuración y mantenimiento en la base de datos, y también pueden drop la base de datos en SQL Server.

db_securityadmin: pueden modificar la pertenencia a roles únicamente para roles personalizados y administrar permisos.

db_accessadmin: pueden agregar o eliminar el acceso a la base de datos para inicios de sesión de Windows, etc

db_backupoperator: pueden crear copias de seguridad de la base de datos.

db_ddladmin: pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.

db_datawriter: pueden agregar, eliminar o cambiar datos en todas las tablas de usuario.

db_datareader: pueden leer todos los datos de todas las tablas y vistas de usuario.

db_denydatawriter: no pueden agregar, modificar ni eliminar datos de tablas de usuario de una base de datos.

db_denydatareader: no pueden leer datos de las tablas y vistas de usuario dentro de una base de datos.

Roles especiales

Los roles de base de datos solo existen en la base de datos "master" que es una base de datos especial en SQL Server.

dbmanager: Puede crear y eliminar bases de datos, convirtiéndose en el propietario. Tiene todos los permisos en las bases de datos que crea, pero no en otras.

db_exporter: Aplicable a grupos de SQL dedicados de Azure Synapse Analytics. Permite actividades de exportación de datos con permisos como CREATE TABLE, ALTER ANY SCHEMA, ALTER ANY EXTERNAL DATA SOURCE, y ALTER ANY EXTERNAL FILE FORMAT.

loginmanager: Puede crear y eliminar inicios de sesión en la base de datos "master" virtual

Ejemplo practicos dentro del archivo sql de integracion

CAPITULO VI: CONCLUSIÓN

Llegamos a la conclusión como grupo que es imprescindible conocer de qué se trata los triggers, su aplicación y uso, ya que en una base de datos no solo se insertan, modifica y eliminan datos, sino que también en la actualidad los usos de las mismas requieren de la implementación de dicho disparador para la optimización del tiempo al automatizar controles y auditorías que sean necesarias al realizar.

Asimismo, el hecho de solo crear una base de datos desconociendo los diversos temas de investigaciones propuestos por la materia, se desaprovecharía todo el potencial que se puede desarrollar aplicándolos. Abriendo un panorama de base para profundizar más aun en conocimientos de sus usos y aplicaciones en la vida real, ya sea en proyectos de mayor escala o proyectos empresariales.

BIBLIOGRAFIA

Paginas web:

<https://learn.microsoft.com/es-es/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver16>

<https://www.tutorialesprogramacionya.com/sqlserverya/temarios/descripcion.php?cod=147&punto=141&inicio=>

https://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/base-de-datos-ii/2019/ii/guia-7.pdf

[https://es.wikipedia.org/wiki/Trigger_\(base_de_datos\)](https://es.wikipedia.org/wiki/Trigger_(base_de_datos))

<https://learn.microsoft.com/es-es/sql/relational-databases/triggers/create-dml-triggers?view=sql-server-ver16>

<https://www.sqlservertutorial.net/sql-server-triggers/sql-server-create-trigger/>