

ADMINISTRACIÓN DE BASE DE DATOS

Guía de Trabajo



VISIÓN

Ser la mejor organización de educación superior posible para unir personas e ideas que buscan hacer realidad sueños y aspiraciones de prosperidad en un entorno incierto

MISIÓN

Somos una organización de educación superior que conecta personas e ideas para impulsar la innovación y el bienestar integral a través de una cultura de pensamiento y acción emprendedora.



Presentación

La presente guía, brinda al estudiante de la asignatura Administración de Base de Datos, los casos y ejercicios prácticos de Laboratorio, usando los conocimientos, metodología, técnicas y herramientas instruidas; que le permitirán realizar una adecuada administración de bases de datos en las organizaciones.

Los autores



Índice

VISIÓN	2
MISIÓN	2
PRESENTACIÓN	3
ÍNDICE	4
Primera unidad	
Práctica N°1: Implementación de Base de Datos	5
Práctica N°2: Índices	7
Práctica N°3: Disparadores	12
Segunda unidad	
Práctica N°4: Automatización de Trabajos-1	15
Práctica N°5: Automatización de Trabajos-2	18
Práctica N°6: Monitoreo y Rendimiento	19
Tercera unidad	
Práctica N°7: Transacciones y Bloqueos	21
Práctica N°8: Inicios de Sesión y Usuarios	24
Práctica N°9: Copias de Seguridad	26
Cuarta unidad	
Práctica N°10: Migración de Datos	28
Práctica N°11: Bases de Datos Distribuidas	29
Práctica N°12: Auditoría de Base de Datos	30
Referencias bibliográficas	34



Primera unidad

Práctica N° 1: Implementación de Base de Datos

1. PRÁCTICA DE LABORATORIO

- a) Para la siguiente práctica se debe utilizar dos discos duros (o dos particiones), por ejemplo "C:\BD" y "D:\BD".
- b) Restaurar la base de datos 'Comercial'. Utilice el *Management Studio*. Restaurarla en la carpeta "C:\BD"
- c) Incrementar el tamaño del archivo de datos, y del archivo de registro de transacciones, ambos a 30 Mb. Luego comprobar físicamente el tamaño de ambos archivos y también en la ficha de propiedades.

```
ALTER DATABASE Comercial
MODIFY FILE
(NAME = Comercial,
SIZE = 30,
MAXSIZE = 30)
```

```
ALTER DATABASE Comercial
MODIFY FILE
(NAME = Comercial_log,
SIZE = 30,
MAXSIZE = 30)
```

- d) Agregar un segundo archivo de base de datos para la BD **Comercial** en la carpeta "D:\BD":

```
ALTER DATABASE Comercial
ADD FILE
( NAME = Comercial1,
FILENAME = 'D:\BD\Comercial1.ndf',
SIZE = 2,
MAXSIZE = 10,
FILEGROWTH = 2)
```

- e) Reduzca la base de datos, para ahorrar espacio en disco. Luego compruebe el tamaño físico de los archivos.

```
DBCC SHRINKDATABASE (Comercial)
```

- f) Separe la base de datos **Comercial**. Apuntar a la base de datos **Master**. Asegurarse que no haya conexiones abiertas.

```
EXEC sp_detach_db 'Comercial'
```

- g) Mueva los archivos de la base de datos **Comercial**, a la unidad "D:\BD". Desde esa nueva ubicación, adjunte la base de datos nuevamente al servidor.

```
EXEC sp_attach_db 'Comercial',
'D:\BD\Comercial.mdf',
'D:\BD\Comercial_log.ldf',
'D:\BD\Comercial1.ndf'
```



- h) Restaurar la base de datos '**Farmacia**'. Utilice el *Management Studio*. Restaurarla en la carpeta "C:\BD"
- i) Incrementar el tamaño del archivo de datos, y del archivo de registro de transacciones, ambos a 100 Mb. Luego comprobar físicamente el tamaño de ambos archivos y también en la ficha de propiedades.
- j) Agregar un segundo archivo de base de datos para la BD Farmacia en la carpeta "D:\BD":
- k) Reduzca la base de datos, para ahorrar espacio en disco. Luego compruebe el tamaño físico de los archivos.
- l) Separe la base de datos **Farmacia**. Apuntar a la base de datos **Master**. Asegurarse que no haya conexiones abiertas.
- m) Mueva los archivos de la base de datos **Farmacia**, a la unidad "D:\BD". Desde esa nueva ubicación, adjunte la base de datos nuevamente al servidor.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 2: Índices

1. PRÁCTICA DE LABORATORIO

- Para la siguiente práctica crear y utilizar la carpeta "C:\BD".
- Restaurar la base de datos 'ClassNorthwind.bak'. Utilice el *Management Studio*. Restaurarla en la carpeta "C:\BD"
- Crear un índice no agrupado llamado OrdersCustomerslink en la columna CustomerID de la tabla Orders con un valor de FILLFACTOR de 75.

```
CREATE NONCLUSTERED INDEX Orders_Customers_link ON Orders(CustomerID)
WITH FILLFACTOR = 75
GO
```

- Comprobar en la ventana browser del *Management Studio*, que se haya creado el índice.
- Crear índices agrupados y no agrupados para todas las referencias de clave foránea a la tabla **Products**, y comprobar su creación mediante comandos.

```
CREATE CLUSTERED INDEX Products_CategoryID_link ON Products(CategoryID)
CREATE NONCLUSTERED INDEX Products_SupplierID_link ON Products(SupplierID)
GO
```

```
PRINT 'CREATED INDEXES:'
SELECT name
FROM sysindexes
WHERE name IN ('Products_CategoryID_link', 'Products_SupplierID_link')
GO
```

- Restaurar la base de datos 'CreditDB.bak'. Utilice el *Management Studio*. Restaurarla en la carpeta "C:\BD"
- Obtener información estadística de la tabla **member** de la base de datos **credit**.

```
USE credit
GO
EXEC sp_spaceused member
SELECT * FROM sysindexes WHERE id = OBJECT_ID('member')
DBCC SHOWCONTIG ('member')
```

La información a recolectar debe ser (revisar en la pestaña mensajes):

Información	Origen	Resultado
Número de filas	sp_spaceused: filas	10.000
Número de índices	SELECT * FROM sysindexes WHERE id = OBJECT_ID('member')	Ninguna. Una fila en sysindexes con un valor de 0 en <i>indid</i> representa la tabla propiamente dicha.
Número de páginas	SHOWCONTIG: Páginas recorridas	145
Número de filas por página	Cálculo y redondeo de los resultados. (n° de filas/n° de páginas) = n° de filas por página	69
Número de extensiones	SHOWCONTIG: Cambios de extensión	18
Promedio de relleno de extensión	SHOWCONTIG: Promedio de páginas por extensión	7,6



Promedio de relleno de página	SHOWCONTIG: Promedio de densidad de página (completa)	98.82%
-------------------------------	---	--------

- h) Crear un índice agrupado único y observará los cambios en la estructura de la tabla. También obtener información acerca de la estructura del índice.

USE credit

**CREATE UNIQUE CLUSTERED INDEX mem_no_CL
ON member (member_no)**

USE credit

**SELECT * FROM sysindexes WHERE id = OBJECT_ID('member')
DBCC SHOWCONTIG ('member')**

Información	Origen	Resultado
Número de páginas del índice agrupado	fila de sysindexes: used	147
Número de páginas de datos en el índice agrupado.	fila de sysindexes: dpages	145
Número de páginas sin datos en el índice agrupado.	(used - dpages)	147-145 = 2
Número de índices	SELECT * FROM sysindexes	Uno. Una fila en sysindexes con un valor de 1 en <i>indid</i> representa el índice agrupado.
Número de páginas	SHOWCONTIG: Páginas recorridas	145
Número de filas por página	Cálculo y redondeo de los resultados. (n° de filas/ n° de páginas) = n° de filas por página	69
Número de extensiones	SHOWCONTIG: Cambios de extensión	18
Promedio de relleno de extensión	SHOWCONTIG: Promedio de páginas por extensión	7,6
Promedio de relleno de página	SHOWCONTIG: Promedio de densidad de página (completa)	98.82%

- i) En este procedimiento creará un índice no agrupado único y obtendrá información acerca de la estructura del índice. Escriba y ejecute esta instrucción para eliminar el índice creado anteriormente.

USE credit

EXEC index_cleanup member

- j) Escriba y ejecute esta instrucción para crear un índice no agrupado con la columna **firstname** de la tabla **Member**, sin especificar un valor de FILLFACTOR:

USE credit

**CREATE NONCLUSTERED INDEX indx_fname
ON member(firstname)**

USE credit

SELECT * FROM sysindexes WHERE id = OBJECT_ID('member')

Información	Origen	Resultado
Número de páginas del Índice no agrupado en la columna <i>firstname</i>	fila de sysindexes: used	39
Número de páginas en el nivel de hoja	fila de sysindexes: dpages	37



Número aproximado de filas por página de hoja	(n° de filas en la tabla/n° de páginas de nivel de hoja)	(10000 / 37) = 270
---	--	--------------------

- k) En este procedimiento creará un índice no agrupado y observará los cambios en la estructura de la tabla. Escriba y ejecute esta instrucción para eliminar el índice no agrupado de la tabla Member:

```
USE credit
EXEC index_cleanup member
```

- l) Escriba y ejecute esta instrucción para crear el mismo índice, pero ahora con un valor de FILLFACTOR del 25 por ciento:

```
USE credit
CREATE NONCLUSTERED INDEX indx_fname
ON member(firstname)
WITH FILLFACTOR=25
```

- m) Escriba y ejecute esta instrucción SELECT que devuelve las filas de sysindexes de la tabla Member:

```
USE credit
SELECT * FROM sysindexes WHERE id = OBJECT_ID('member')
```

Información	Origen	Resultado
Número de páginas en este índice	fila de sysindexes: used	293
Número de páginas en el nivel de hoja.	fila de sysindexes: dpages	145
Número aproximado de filas por página de hoja	(n° de filas en la tabla/n° de páginas de nivel de hoja)	(10000/145) = 70

NOTA: Con un valor de FILLFACTOR de 0 (predeterminado), pueden caber 270 filas en cada página de nivel de hoja. Multiplique 270 por el 25 por ciento y el resultado es aproximadamente 70 filas. Éste es el número de filas que cabrán en una página de nivel de hoja que sólo está llena un 25 por ciento.

- n) En este ejercicio creará varios índices con la tabla Member, obtendrá sus estadísticas y evaluará si un índice es útil para el optimizador de consultas según su selectividad. Eliminar los índices existentes de la tabla Member:

```
USE credit
EXEC index_cleanup member
```

- o) Escriba y ejecute estas instrucciones para crear tres índices basados en la tabla Member:

```
USE credit
CREATE UNIQUE INDEX indx_member_no ON member (member_no)
CREATE INDEX indx_corp_lname ON member (corp_no,lastname)
CREATE INDEX indx_lastname ON member (lastname)
GO
```

- p) En este procedimiento obtendrá estadísticas de los índices nuevos, registrará la información estadística y evaluará la selectividad de los índices. Escriba y ejecute esta instrucción para mostrar la información estadística de índices acerca de la columna member_no de la tabla Member:

```
USE credit
DBCC SHOW_STATISTICS (member,indx_member_no)
```



Información	Resultado
Rows	10.000
Steps	3
Density	1
All density	0.0001

¿Qué selectividad tiene el índice de la columna *member_no*?

Es muy selectivo. El índice *indx_member_no* se crea con la columna *member_no*, que contiene valores únicos. Si una consulta especifica un número de miembro en la cláusula WHERE con una igualdad, sólo se devolverá una fila.

- q) Escriba y ejecute esta instrucción para mostrar la información estadística de índices de un índice compuesto basado en las columnas **corp_no** y **lastname** de la tabla **Member**:

USE credit

DBCC SHOW_STATISTICS (member,indx_corp_lname)

Información	Resultado
Rows	10,000
Steps	201
Density	0.3257
All density (corp no)	0.0024
All density (corp_no, lastname)	0.0006

¿Qué selectividad tiene este Índice?

Sigue siendo selectivo, pero no tanto como la columna *member_no*, ya que existen valores duplicados.

- r) Escriba y ejecute esta instrucción para mostrar la información estadística de índices acerca de la columna **lastname** de la tabla **Member**:

USE credit

DBCC SHOW_STATISTICS (member,indx_lastname)

Información	Resultado
Rows	10,000
Steps	26
Density	0
All density	0.0384

¿Qué selectividad tiene este Índice?

Este índice no es muy selectivo. El uso de la fórmula para calcular la densidad muestra que este índice tiene baja selectividad. Debido a que no puede dividir por cero, el resultado es cero. All density indica la densidad total.

$$((26/0)/10000) = 0$$

- s) Restaurar la base de datos "Supermercado.bak"
- t) Crear un índice no agrupado para el campo NombreArticulo de la tabla Artículo.

**CREATE NONCLUSTERED INDEX IDXNombreArticulo
ON Artículo (NombreArticulo)**

- u) Crear un índice con valores únicos, no agrupado, para el campo FechaEmision de la tabla DocVenta.

**CREATE UNIQUE NONCLUSTERED INDEX IDXFechaEmision
ON DocVenta (Fecha)**



- v) Crear un índice con valores no únicos, no agrupado, de múltiples campos, para los campos ApPatContacto y ApMatContacto, de la tabla Clientes.

**CREATE NONCLUSTERED INDEX IDXContacto
ON Cliente (ApPatContacto,ApMatContacto)**

- w) Visualizar y registrar en una tabla, la información y estadísticas más importantes de las tablas e índices creados. Escribir un análisis cualitativo del significado de la información encontrada por cada índice.

```
SELECT * FROM sysindexes WHERE id = OBJECT_ID('articulo')
DBCC SHOW_STATISTICS (articulo,idxnombrearticulo)
DBCC SHOWCONTIG ('articulo')
```

NOTA: Falta extraer la información para las tablas DocVenta y Cliente.

- x) Restaurar la base de datos "Farmacia.bak"
y) Crear un índice no agrupado para el campo *Descripción* de la tabla *Artículos*.
z) Crear un índice con valores únicos no agrupado, para el campo *Fecha* de la tabla *Ventas*.
aa) Crear un índice con valores no únicos, no agrupado, de múltiples campos, para los campos *Apellidos y Nombres*, de la tabla *Clientes*.
bb) Visualizar y registrar en una tabla, la información y estadísticas más importantes de las tablas e índices creados. Escribir un análisis cualitativo del significado de la información encontrada por cada índice.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 3: Disparadores

1. PRÁCTICA DE LABORATORIO

- Restaurar la base de datos proporcionada "Supermercado.bak".
- El siguiente disparador alerta si se ha agregado algún registro en la tabla Cliente:

```
CREATE TRIGGER SeAgregoCliente  
ON Cliente  
FOR INSERT  
AS  
IF UPDATE(IDCliente)  
PRINT 'Se agregó un nuevo cliente... Trigger disparado'
```

Pruebe el disparador con la siguiente instrucción:

```
INSERT INTO Cliente (IDCliente, RazonSocial, ApPatContacto, ApMatContacto,  
NombreContacto)  
VALUES ('JPF','Restaurante El Remanso','Parker','Ferrer','José')
```

- Analice y ejecute el siguiente disparador que no permita ingresar un Artículo con precio negativo:

```
CREATE TRIGGER VerificaPrecio  
ON Articulo  
FOR INSERT  
AS  
IF (SELECT PrecioUnidad FROM Inserted)<0  
BEGIN  
PRINT 'No se permite precios negativos... Tigger Disparado...'  
ROLLBACK  
END
```

Pruebe el disparador con la siguiente instrucción:

```
INSERT INTO  
Articulo(IDArticulo,NombreArticulo,UnidExistencia,PrecioUnidad,UnidMinimo)  
VALUES(114,'Spaguetti',50,-5,10)
```

- Crear un disparador que no permita modificar las Descripciones almacenadas en la tabla unidad.

```
CREATE TRIGGER NoCambiosUnidad  
ON Unidad  
FOR UPDATE  
AS  
IF UPDATE(Descripcion)  
BEGIN  
PRINT 'No se deben cambiar las descripciones... Trigger Disparado...'  
ROLLBACK  
END
```

Pruebe el disparador con la siguiente instrucción:

```
UPDATE Unidad  
SET Descripcion='Carnes/Embutidos'  
WHERE IDUnidad=6
```

- Restaurar la base de datos ClassNorthwind desde el archivo Northwind.bak
- Este desencadenador actualiza la columna UnitsInStock de la tabla Products cuando se inserta una fila en la tabla Order Details (cuando se recibe un pedido).



```
CREATE TRIGGER OrdDet_Insert
ON [Order Details]
FOR INSERT
AS
UPDATE P SET
UnitsInStock = (P.UnitsInStock - I.Quantity)
FROM Products AS P INNER JOIN Inserted AS I
ON P.ProductID = I.ProductID
GO
```

- g) Ejecute la instrucción SELECT siguiente para seleccionar una fila de la tabla Products y determinar las unidades de un producto que hay en almacén:

```
SELECT * FROM Products WHERE ProductID = 22
```

La columna UnitsInStock debe contener el valor de 104. Si la columna UnitsInStock muestra un número diferente, anótelo.

- h) Inserte una fila en la tabla Order Details que pida 50 unidades del producto:

```
INSERT [Order Details] (OrderID, ProductID, UnitPrice, Quantity, Discount)
VALUES (11077, 22, 21.00, 50, 0.0)
```

- i) Haga una consulta en la tabla Products para comprobar que el valor de la columna UnitsInStock para el ProductID determinado ha cambiado a 54.
- j) En este ejercicio va a crear dos tablas nuevas denominadas NewCategories y NewProducts. Ninguna de las dos tiene las restricciones PRIMARY y FOREIGN KEY de las tablas Categories y Products. Creará un desencadenador en la tabla NewCategories. Este desencadenador actualiza la columna Discontinued de la tabla NewProducts cuando se elimina una categoría (cuando se elimina un registro de la tabla NewCategories). Todos los productos afectados se marcan con 1, lo que indica que ya no se suministran.
- k) Escriba y ejecute la consulta siguiente para crear dos tablas nuevas denominadas NewCategories y NewProducts:

```
USE ClassNorthwind
GO
--Crea una tabla NewCategories
SELECT * INTO NewCategories FROM Categories
--Crea una tabla NewProducts
SELECT * INTO NewProducts FROM Products
GO
```

- l) Escriba y ejecute la consulta siguiente para crear un desencadenador en la tabla NewCategories. Este desencadenador actualiza la columna Discontinued de la tabla NewProducts a 1 cuando se elimina la categoría principal de un producto (cuando se elimina una fila de la tabla NewCategories).

```
CREATE TRIGGER Category_Delete
ON NewCategories
FOR DELETE
AS
UPDATE P SET Discontinued = 1
FROM NewProducts AS P INNER JOIN Deleted AS d
ON P.CategoryID = D.CategoryID
```

- m) Escriba y ejecute la siguiente instrucción SELECT que consulta la tabla NewProducts para



determinar si se siguen suministrando los productos con CategoryID "7".

**SELECT ProductID, CategoryID, Discontinued
FROM NewProducts WHERE CategoryID = 7**

- n) Escriba una instrucción DELETE que quite una fila de la tabla NewCategories y compruebe que el desencadenador se ejecuta correctamente.

DELETE NewCategories WHERE CategoryID = 7

- o) Restaurar la base de datos Farmacia desde el archivo FARMACIA.BAK.
- p) Crear y ejecutar un disparador que alerte si se ha agregado algún registro en la tabla Cliente.
- q) Crear y ejecutar un disparador que alerte si se ha ingresado un Artículo con stock mayor de 100 unidades.
- r) Crear y ejecutar un disparador que no permita ingresar un Artículo con stock mínimo menor de 10 unidades.
- s) Crear y ejecutar un disparador que no permita modificar el stock de un artículo.
- t) Crear y ejecutar un disparador que no permita eliminar un cliente.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 4: Automatización de Trabajos

1. PRÁCTICA DE LABORATORIO

- a) Utilizando la herramienta de SQL Server, Correo electrónico de Bases de Datos, configurar un perfil nuevo de correo electrónico, para SQL Server:

Asistente para configuración de Correo electrónico de base de datos - FELIPEI7\MSSQLSE...

Administrar perfil existente
Seleccionar el perfil para ver, cambiar o eliminar.

Nombre de perfil: SQL Server Email [Eliminar]

Descripción:

Un perfil puede estar asociado a varias cuentas SMTP. Si se produce un error de la cuenta al enviar un mensaje de correo electrónico, el perfil usa la siguiente cuenta en la lista de prioridades. Especifique las cuentas asociadas al perfil y mueva las cuentas para establecer la prioridad de conmutación por error.

Cuentas SMTP:

Prionid...	Nombre de cuenta	Dirección de correo electrónico
1	Felipe SQL Server	faliagac@hotmail.com

[Agregar...]
[Quitar]
[Subir]
[Bajar]

- b) Utilizando la herramienta de SQL Server, Correo electrónico de Bases de Datos, agregar una cuenta de correo electrónico (de su uso personal), en el perfil creado anteriormente:

Asistente para configuración de Correo electrónico de base de datos - FELIPEI7\MSSQLSE...

Administrar cuenta existente
Elegir la cuenta para ver, cambiar o eliminar.

Nombre de cuenta: Felipe SQL Server [Eliminar]

Descripción:

Servidor de correo saliente (SMTP)

Dirección de correo electrónico: faliagac@hotmail.com

Nombre para mostrar: Felipe SQL Server

Correo electrónico de respuesta: faliagac@hotmail.com

Nombre del servidor: smtp.live.com Número de puerto: 587

El servidor requiere una conexión segura (SSL)

Autenticación SMTP

Autenticación de Windows con credenciales del servicio Motor de base de datos

Autenticación básica

Nombre de usuario: faliagac@hotmail.com

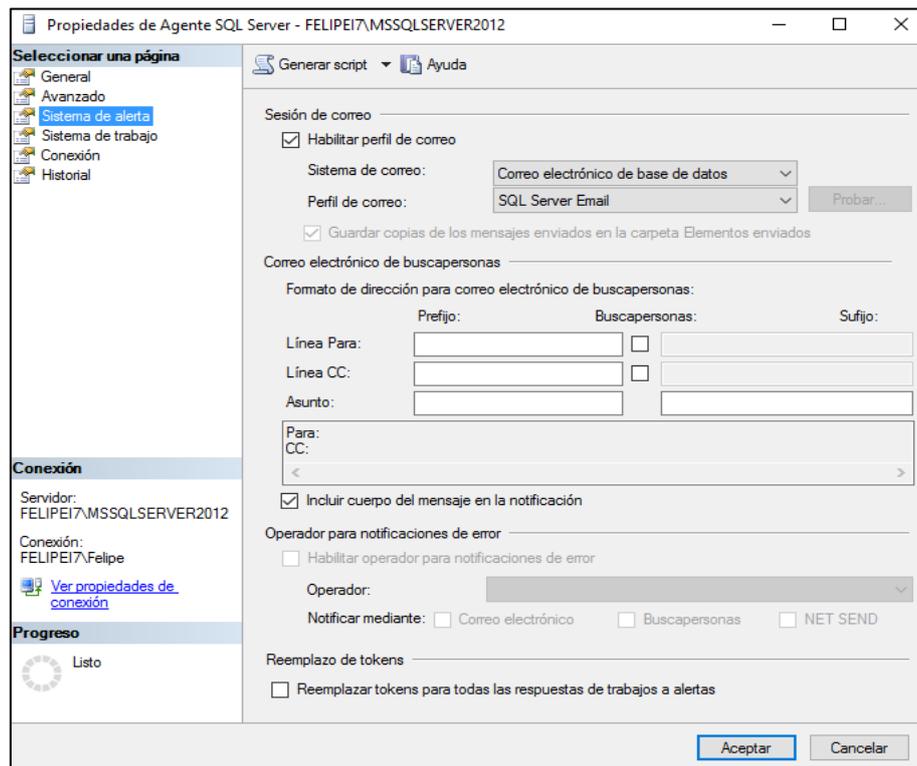
Contraseña: *****

Confirmar contraseña: *****

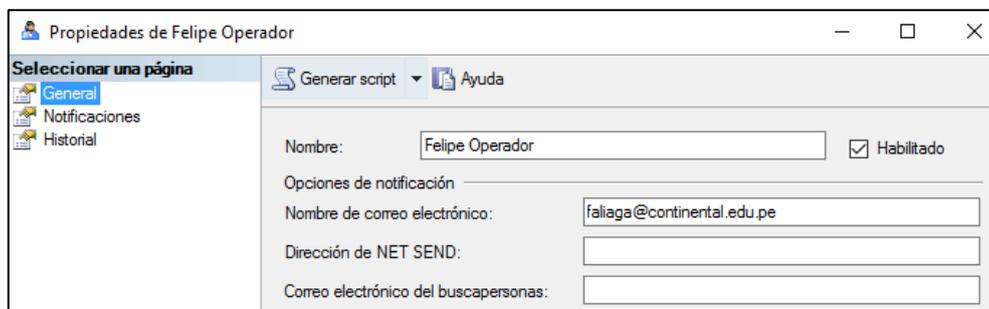
Autenticación anónima

[Ayuda] < Atrás [Siguiente > Finalizar >> Cancelar

- c) En las propiedades del Agente SQL Server, configurar el uso del correo electrónico, creado en los pasos anteriores. Seleccionar el perfil creado.



- d) Restaurar la base de datos Supermercado.bak para realizar pruebas.
- e) Crear en el Agente de SQL Server, un operador que reciba las notificaciones de correo electrónico. Debe utilizar su propio correo electrónico de continental.



- f) Crear en el Agente de SQL Server, un trabajo que liste los clientes de la base de datos supermercado, y establecer una notificación vía e-mail al operador antes creado, en cuanto el trabajo termine.
- g) Ejecutar el trabajo y comprobar si llegó el e-mail de notificación al operador; vale decir a su cuenta de continental.
- h) Crear en el Agente de SQL Server, un trabajo que extraiga un backup de la base de datos supermercado, y crear una notificación vía e-mail al operador antes creado, en cuanto el trabajo termine. El paso incluido en el trabajo debe ser:

BACKUP DATABASE Supermercado to disk='c:\bd\super.bak' WITH INIT

- i) Ejecutar el trabajo y comprobar si llegó el e-mail de notificación al operador; vale decir a su cuenta de continental. Comprobar también si se extrajo el backup respectivo.



- j) Crear en el Agente de SQL Server, una alerta que notifique vía e-mail al operador antes creado, una alerta de tipo (14-Permisos insuficientes).
- k) Pruebe la alerta tratando con el siguiente comando:

```
raiserror('Este es un error simulado de SQL Server',14,1) with log
```
- l) Verifique que haya llegado e-mail de la alerta.
- m) Automatice el trabajo de backup del ejercicio (h), para que se repita diariamente a una hora determinada.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 5: Automatización de Trabajos

1. PRÁCTICA DE LABORATORIO

- a) Utilizando la herramienta de SQL Server, Correo electrónico de Bases de Datos, configurar un perfil nuevo de correo electrónico, para SQL Server.
- b) Utilizando la herramienta de SQL Server, Correo electrónico de Bases de Datos, agregar una cuenta de correo electrónico (de su uso personal de Gmail u Hotmail), en el perfil creado anteriormente:
- c) En las propiedades del Agente SQL Server, configurar el uso del correo electrónico, creado en los pasos anteriores. Seleccionar el perfil creado.
- d) Restaurar la base de datos Farmacia.bak.
- e) Crear en el Agente de SQL Server, un operador que reciba las notificaciones de correo electrónico. Debe utilizar el correo del profesor (**faliagac@hotmail.com**)
- f) Crear en el Agente de SQL Server, un trabajo que extraiga un backup de la base de datos farmacia, y crear una notificación vía e-mail al operador antes creado, en cuanto el trabajo termine.
- g) Ejecutar el trabajo. El profesor calificará el presente laboratorio, de acuerdo al correo electrónico recibido.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- El profesor calificará el presente trabajo, de acuerdo al correo electrónico recibido



Práctica N° 6: Monitoreo y Rendimiento

1. PRÁCTICA DE LABORATORIO

- Restaurar la base de datos "Northwindcopy.bak" en el servidor SQL Server.
- Copiar la carpeta "Monitor" a la unidad C:\
- Abrir la herramienta "Monitor de Rendimiento" (de Windows), y agregar los siguientes contadores:

Contador	Instancia	Objeto
Lecturas de pagina/ s	NorthWindCopy	SQLServer: Administrador de buffer
Transacciones activas	NorthWindCopy	SQLServer: Databases
Porcentaje de registro usado	NorthWindCopy	SQLServer: Databases
Transacciones/s		SQLServer: Databases
Procesos bloqueados		SQLServer: General Statistics
Transacciones		SQLServer: General Statistics
Búsquedas de índices/s		SQLServer: Métodos de acceso

- Observe el valor que reflejan los contadores. (Capturar la pantalla del monitor de rendimiento)
- Desde una ventana de comandos, ejecute el archivo "Monitor.bat" de la carpeta "C:\Monitor". Este conjunto de archivos en lotes, simularán actividades de los usuarios en el servidor SQL Server, en la base de datos "NorthwindCopy".
- Vuelva a observar el valor que reflejan ahora los contadores. ¿Qué tendencias se observan? (Capturar la pantalla del monitor de rendimiento)
- Cerrar la herramienta Monitor de rendimiento, para eliminar los contadores seleccionados.
- Abrir la herramienta "Monitor de Rendimiento" (de Windows), y agregar los siguientes contadores:

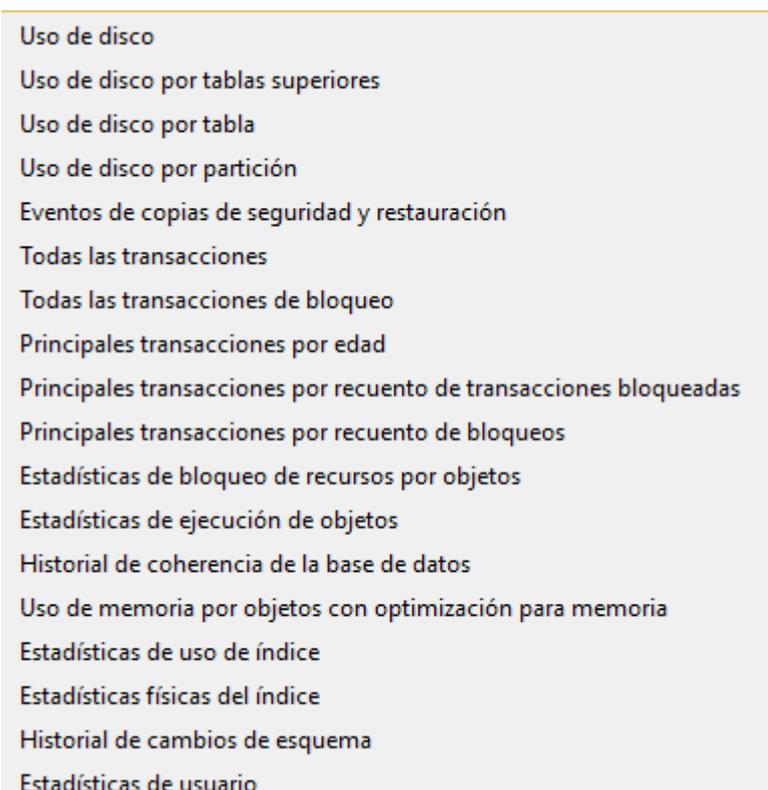
Contador	Instancia	Objeto
Errores de página/s		Memoria
Páginas/s		Memoria
Mbytes disponibles		Memoria
% de tiempo de procesador	Total	Proceso
Errores de página/s	Total	Proceso
Memoria de conexión (KB)		SQLServer: Memory Manager
Memoria total del servidor (KB)		SQLServer: Memory Manager

- Observar el valor que reflejan ahora los contadores. ¿Qué tendencias se observan? - (Capturar la pantalla del monitor de rendimiento)
- Iniciar el monitor de actividad del servidor SQL Server.
- Visualizar e interpretar la información de los rubros del monitor de actividad: (Capturar las pantallas del monitor de actividad)



- Información general
- Procesos**
- Esperas de recursos
- E/S de archivo de datos
- Consultas costosas recientes

- l) Visualizar e interpretar los informes estándar de la base de datos “Northwindcopy”: (Capturar las pantallas de cada reporte)



2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 7: Transacciones y Bloqueos

1. PRÁCTICA DE LABORATORIO

- Restaurar la base de datos proporcionada "Northwind.bak".
- Ejecutar el siguiente Script:

```
BEGIN TRAN
PRINT 'trancount value:'
PRINT @@trancount
PRINT 'Before update:'
SELECT ContactName FROM Customers WHERE CustomerID = 'GREAL'
UPDATE Customers SET ContactName = 'Howard Snyder_Updated' WHERE CustomerID
='GREAL'
PRINT 'After update:'
SELECT ContactName FROM Customers WHERE CustomerID = 'GREAL'
PRINT 'trancount value:'
PRINT @@trancount
-- COMMIT TRANSACTION
```

- En este momento, ¿se han confirmado los cambios efectuados con la instrucción UPDATE en esta transacción? ¿Cómo puede determinar esto?

No, la transacción se debe completar con la instrucción COMMIT TRAN. La transacción sigue estando activa y sigue manteniendo los bloqueos que haya adquirido. El valor de la variable global @@trancount es 1, lo que indica que en esta sesión se ha enviado una instrucción BEGIN TRAN.

- ¿Podrían otras transacciones consultar o actualizar los datos modificados? Otras transacciones no podrían consultar o modificar los datos modificados hasta que se haya confirmado (o deshecho) la transacción.
- Escriba una instrucción COMMIT TRANSACTION en la ventana de consulta y, después, resáltela y ejecútela para completar la transacción y hacer que los cambios sean definitivos.
 - Resalte y ejecute una de las instrucciones SELECT de la tabla Customers para comprobar que ahora sí se ha completado el cambio.
 - Ejecute el siguiente script:

```
BEGIN TRAN
PRINT 'trancount value:'
PRINT @@trancount
PRINT 'Before update:'
SELECT ContactName FROM Customers WHERE CustomerID = 'GREAL'
UPDATE Customers SET ContactName = 'Howard Snyder' WHERE CustomerID = 'GREAL'
PRINT 'After update:'
SELECT ContactName FROM Customers WHERE CustomerID = 'GREAL'
PRINT 'trancount value:'
PRINT @@trancount
```

```
ROLLBACK TRAN
```

```
Print 'ROLLBACK issued'
SELECT ContactName FROM Customers WHERE CustomerID = 'GREAL'
PRINT 'trancount value:'
PRINT @@trancount
```

- ¿Se ha almacenado de forma permanente en la base de datos la modificación efectuada por la instrucción UPDATE?



No. La transacción se ha cancelado, de modo que los cambios efectuados durante la transacción se han deshecho.

- ¿Se ha completado la transacción?
Sí. La instrucción ROLLBACK TRAN completa la transacción y libera los bloqueos adquiridos por la transacción. La consulta de la variable global @@trancount devuelve cero.

- f) Cierre los scripts anteriores, y abra dos nuevas consultas, apuntando a la base de datos Northwind2. En el primer script ejecute el procedimiento almacenado de sistema:

```
exec sp_lock
```

Describa los bloqueos activos.

- g) En el segundo script ejecute lo siguiente, y revise los resultados:

```
BEGIN TRAN
  UPDATE Customers
    SET ContactName = 'Howard Snyder'
    WHERE CustomerID = 'GREAL'
-- For the purpose of the exercise, COMMIT TRAN or ROLLBACK TRAN are not used.

PRINT 'Server Process ID (spid)'
PRINT @@spid
-- Use the SPID to identify the connection when using sp_lock.
```

- h) En el primer script ejecute el procedimiento almacenado de sistema:

```
exec sp_lock
```

Describa los bloqueos activos actuales, y explique con los bloqueos que existían en el ejercicio 6.

- i) En el segundo script ejecute ROLLBACK TRAN.
j) En el primer script ejecute el procedimiento almacenado de sistema, y revise los bloqueos liberados.

```
exec sp_lock
```

- k) Cierre los scripts anteriores, y abra tres nuevas consultas, apuntando a la base de datos Northwind2. En el primer script ejecute el procedimiento almacenado de sistema:

```
exec sp_lock
```

Describa los bloqueos activos.

- l) En el segundo script ejecute lo siguiente, y revise los resultados:

```
BEGIN TRAN
  SELECT * FROM Customers(TABLOCKX) WHERE CustomerID = 'GREAL'
-- TABLOCKX bloquea la tabla Customers
-- For the purpose of the exercise, COMMIT TRAN or ROLLBACK TRAN are not used.

PRINT 'Server Process ID (spid)'
PRINT @@spid
-- Use the SPID to identify the connection when using sp_lock.
```

- m) En el tercer script ejecute lo siguiente, y revise los resultados:

```
BEGIN TRAN
  UPDATE Customers
    SET ContactName = 'Howard Snyder'
```



```
WHERE CustomerID = 'GREAL'  
-- For the purpose of the exercise, COMMIT TRAN or ROLLBACK TRAN are not used.  
  
PRINT 'Server Process ID (spid)'  
PRINT @@spid
```

- ¿Hay una transacción que no se puede ejecutar? Si es así, ¿por qué?
Sí, la segunda transacción está intentando actualizar los datos que están bloqueados por la primera transacción, de modo que la segunda transacción debe esperar a que termine la primera. Puede determinar si una transacción no se puede ejecutar debido a un conflicto de bloqueo; para ello, utilice el procedimiento almacenado de sistema `sp_lock` y busque la palabra `WAIT` en la columna de estado. En este caso, la transacción de la conexión 3 debe esperar a que termine la transacción de la conexión 2.
- ¿Cuánto tiempo espera una transacción un recurso bloqueado?
Una transacción espera indefinidamente un recurso bloqueado, siempre y cuando no haya interbloqueos. Se puede establecer el período de tiempo de espera de la sesión para controlar cuánto tiempo esperará SQL Server a que se liberen los recursos bloqueados.

n) En el segundo script cancele la transacción:

```
ROLLBACK TRAN
```

o) En el primer script ejecute el procedimiento almacenado de sistema:

```
exec sp_lock
```

Revise si el bloqueo ha sido desactivado y si se ejecutó el tercer script.

- p) Restaure la base de datos supermercado. Cree un script donde se bloquee la tabla **Cliente**. Cree un segundo script donde se trate de modificar la tabla `Cliente`, pero no se pueda porque se encuentra bloqueada. Capture las pantallas necesarias para demostrar el bloqueo.
- q) Libere el bloqueo para que se ejecute el segundo script. Capture las pantallas necesarias para demostrar el desbloqueo.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 8: Inicios de Sesión y Usuarios

1. PRÁCTICA DE LABORATORIO

- a) Restaurar la base de Datos Supermercado desde el archivo SUPERMERCADO.BAK.
- b) Crear los siguientes inicios de sesión (login), con sus respectivas contraseñas:

```
exec sp_addlogin 'usuario1', 'abc123'  
exec sp_addlogin 'usuario2', 'abc123'  
exec sp_addlogin 'usuario3', 'abc123'
```

- c) Crear los usuarios que representen a los 03 inicios de sesión en la base de datos Supermercado:

```
use Supermercado  
exec sp_grantdbaccess 'usuario1'  
exec sp_grantdbaccess 'usuario2'  
exec sp_grantdbaccess 'usuario3'
```

- d) Verificar los usuarios creados:

```
sp_helpuser
```

- e) Conceder la función de lectura de datos **db_datareader**, al **usuario1**:

```
sp_addrolemember 'db_datareader','usuario1'
```

- f) Denegar la función de lectura de datos **db_denydatareader**, al **usuario2**:

```
sp_addrolemember 'db_denydatareader','usuario2'
```

- g) Conceder la función de lectura y escritura de datos **db_datareader** y **db_datawriter**, al **usuario3**:

```
exec sp_addrolemember 'db_datareader','usuario3'  
exec sp_addrolemember 'db_datawriter','usuario3'
```

- h) Desde el object explorer conecte una instancia del servidor SQL para el **usuario1**, otra para el **usuario2**, y otra para el **usuario 3**.
- i) Seleccione al **usuario1**. Compruebe si el usuario puede leer y escribir datos en las tablas.
- j) Seleccione al **usuario2**. Compruebe si el usuario puede leer y escribir datos en las tablas.
- k) Seleccione al **usuario3**. Compruebe si el usuario puede leer y escribir datos en las tablas.
- l) Otorgar al **usuario2**, permisos de lectura sobre la tabla **Ciente**:

```
GRANT select ON cliente to usuario2
```

- m) Otorgar al **usuario1**, y al **usuario2**, permisos de INSERT Y UPDATE, sobre la tabla **Cajero**.

```
GRANT insert,update ON cajero to usuario1,usuario2
```

- n) Seleccionar al **usuario1**. Compruebe si el usuario puede leer y escribir datos en las tablas.
- o) Seleccionar al **usuario2**. Compruebe si el usuario puede leer y escribir datos en las tablas.
- p) Restaurar la base de Datos Farmacia desde el archivo FARMACIA.BAK.
- q) Crear 03 inicios de sesión (usuario1, usuario2 y usuario3), con sus respectivas contraseñas.
- r) Crear los usuarios que representen a los 03 inicios de sesión en la base de datos Farmacia.
- s) Conceder la función de lectura de datos **db_datareader**, al **usuario1**.
- t) Denegar la función de lectura de datos **db_denydatareader**, al **usuario2**.
- u) Conceder las funciones de lectura y escritura de datos **db_datareader** y **db_datawriter**, al **usuario3**.



- v) Desde el object explorer conecte una instancia del servidor SQL para el **usuario1**, otra para el **usuario2**, y otra para el **usuario 3**.
- w) Seleccione al **usuario1**. Comprobar si el usuario puede leer y escribir datos en las tablas.
- x) Seleccione al **usuario2**. Comprobar si el usuario puede leer y escribir datos en las tablas.
- y) Seleccione al **usuario3**. Comprobar si el usuario puede leer y escribir datos en las tablas.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar las instrucciones en T-SQL y las capturas de pantalla en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 9: Copias de Seguridad

1. PRÁCTICA DE LABORATORIO

- a) Iniciar un nuevo query. Crear un dispositivo de copia de seguridad para la base de datos **Supermercado**:

```
sp_addumpdevice 'disk' , 'Supermercado_backup', 'C:\Supermercado.bak'
```

- b) Realice una copia de seguridad completa de la base de datos **Supermercado**, inicializando el dispositivo:

```
BACKUP DATABASE Supermercado to Supermercado_backup WITH INIT
```

- c) Ingrese un nuevo registro para la tabla **UNIDAD**.
d) Realice la primera copia de seguridad diferencial de la base de datos **Supermercado**, en el dispositivo **Supermercado_backup**:

```
BACKUP DATABASE Supermercado to Supermercado_backup WITH differential,  
NOINIT
```

- e) Ingrese un nuevo registro para la tabla **UNIDAD**.
f) Realice la segunda copia de seguridad diferencial:

```
BACKUP DATABASE Supermercado to Supermercado_backup WITH differential,  
NOINIT
```

- g) Ingrese un último registro a la tabla **UNIDAD**. Realice una tercera copia de seguridad diferencial.
h) Genere un segundo dispositivo para realizar la copia de seguridad del registro de transacciones:

```
sp_addumpdevice 'disk' , 'Supermercado_log', 'C:\SupermercadoLog.bak'
```

- i) Realice la copia de seguridad del registro de transacciones:

```
BACKUP DATABASE Supermercado TO Supermercado_log WITH INIT  
BACKUP LOG Supermercado TO Supermercado_log WITH NOINIT
```

- j) Verifique la base de datos y archivo de transacciones que se respaldaron en el dispositivo de copia de seguridad **Supermercado_backup**:

```
RESTORE FILELISTONLY FROM Supermercado_backup  
RESTORE VERIFYONLY FROM Supermercado_backup
```

- k) Restaure la primera copia de seguridad diferencial:

```
RESTORE DATABASE Supermercado FROM Supermercado_backup WITH  
NORECOVERY, REPLACE  
RESTORE DATABASE Supermercado FROM Supermercado_backup WITH  
FILE=2, RECOVERY
```

- l) Verifique el contenido de la tabla **Unidad**.
m) Restaure la segunda copia de seguridad diferencial
n) Verifique el contenido de la tabla **Unidad**.
o) Restaure la tercera copia de seguridad diferencial
p) Verifique el contenido de la tabla **Unidad**.
q) Restaure la base de datos completa y el registro de transacciones:

```
RESTORE DATABASE Supermercado FROM Supermercado_LOG WITH
```



**NORECOVERY, REPLACE
RESTORE LOG Supermercado FROM Supermercado_LOG WITH RECOVERY**

- r) Verifique el contenido de la tabla **Unidad**.
- s) Iniciar un nuevo query. Crear un dispositivo de copia de seguridad para la base de datos **Farmacia**.
- t) Realice una copia de seguridad completa de la base de datos **Farmacia**, inicializando el dispositivo.
- u) Genere un segundo dispositivo para realizar la copia de seguridad del registro de transacciones de la base de datos **Farmacia**.
- v) Realice la copia de seguridad del registro de transacciones de la base de datos **Farmacia**.
- w) Elimine la base de datos **Farmacia**.
- x) Restaure la base de datos **Farmacia** completa desde el primer dispositivo.
- y) Elimine la base de datos **Farmacia**.
- z) Restaure la base de datos **Farmacia** completa desde la copia de seguridad del registro de transacciones.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar el código T-SQL en un documento de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 10: Migración de Datos

1. PRÁCTICA DE LABORATORIO

- a) Utilizando la herramienta importación-exportación de datos de SQL Server, importar las bases de datos "COMERCIAL" y "BOTICA", desde los archivos de Microsoft Excel proporcionados. Implementar también las relaciones para cada tabla de base de datos.
- b) Exportar hacia Microsoft Access, ambas bases de datos implementadas: "COMERCIAL" y "BOTICA".
- c) Descargar de internet, una base de datos de tema libre (3 tablas en promedio) en cualquier formato, que no sea de SQL Server.
- d) Implementar e importar la base de datos descargada hacia SQL Server. Implementar también las relaciones para cada tabla de base de datos.

2. FECHA DE ENTREGA

.....

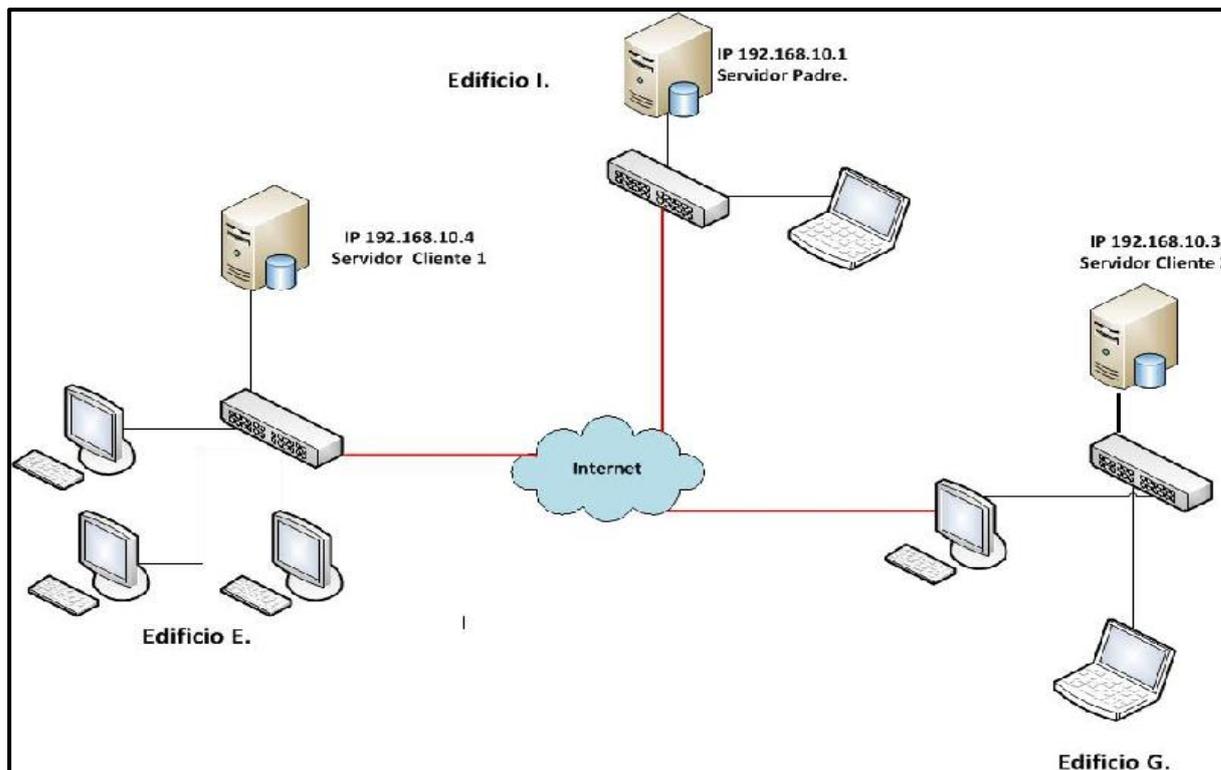
3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar en archivos de Microsoft Access, base de datos original y backup de SQL Server, todo en un solo archivo comprimido.
- Subir al enlace en el aula virtual

Práctica N° 11: Bases de Datos Distribuidas

1. PRÁCTICA DE LABORATORIO

- a) Instalar las instancias necesarias del servidor SQL Server, para conformar la siguiente base de datos distribuida:



2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar capturas de pantallas, todo en un solo archivo de Microsoft Word.
- Subir al enlace en el aula virtual



Práctica N° 12: Auditoría de Bases de Datos

1. PRÁCTICA DE LABORATORIO

a) **CASO:** Usted es el auditor informático de una importante institución financiera. Un día lo convocan a una reunión con el gerente general, donde se tratan temas relativos a la problemática originada por la utilización de cajeros automáticos por parte de los clientes. En particular se han detectado los siguientes inconvenientes:

1. Actualmente el banco entrega una tarjeta de plástico a cada uno de los clientes que contiene el número de cuenta (codificado en forma magnética) y un PIN. Desafortunadamente, hay clientes que escriben el PIN en la tarjeta, y en consecuencia cuando pierden la tarjeta o son robados han ocurrido extracciones de fondos no autorizadas. Este número está en aumento.
2. Los cajeros automáticos permiten hasta tres intentos de ingresar el PIN actual. Luego la máquina retiene la tarjeta. Muchos clientes olvidan sus PIN's. Recientemente, un cliente enojado intentó recuperar su tarjeta utilizando una palanca.
3. El banco envía por correo las tarjetas y los PIN's, sin embargo, recientemente los correos han sido robados y se han producido extracciones de fondo no autorizadas.

El Gerente General, le solicita a Ud. encontrar por lo menos 05 soluciones respecto a los temas presentados. Escriba un reporte breve exponiendo sus 05 soluciones.

b) Para el caso anterior, se le solicita redactar el plan de auditoría de bases de datos.

PLAN DE AUDITORÍA DE BASE DE DATOS	
1. Antecedentes	(Describir los datos generales de la organización tomada como modelo, y resumir los síntomas iniciales detectados, que justifiquen realizar una auditoría informática)
2. Objetivos	(Plantear un objetivo general y 04 objetivos específicos)
3. Cronograma	(Diseñar un cronograma de actividades genérico, donde se estipulen las actividades de auditoría. Incluir fechas tentativas)
4. Presupuesto	(Formular una lista de recursos humanos, materiales y otros que se necesitarían para llevar a cabo la auditoría, con sus respectivos costos tentativos)
5. Anexos	(Considerar la información adicional relevante, que considere pertinente)

c) Seleccione una organización de tamaño mediano real o virtual, que requiera una auditoría de base de datos. (Puede ser la organización donde Ud. trabaja o hace prácticas pre profesionales). Aplique el siguiente cuestionario de auditoría, y encuentre las observaciones y recomendaciones que sean necesarias.

1. ¿Existen procedimientos formales para la operación del sistema de cómputo?
SI () NO ()
2. ¿Están actualizados los procedimientos?
SI () NO ()



3. Indique la periodicidad de la actualización de los procedimientos:
Semestral ()
Anual ()
Cada vez que haya cambio de equipo ()
4. Indique el contenido de los instructivos de operación para cada aplicación:
Identificación del sistema ()
Identificación del programa ()
Periodicidad y duración de la corrida ()
Especificación de formas especiales ()
Fechas de creación y expiración Instructivo sobre datos de entrada y salida ()
Altos programados y la acciones requeridas ()
Instructivos específicos a los operadores en caso de falla del equipo ()
Instructivos de reinicio ()
Procedimientos de recuperación para proceso de gran duración o criterios ()
Identificación de todos los dispositivos de la máquina a ser usados ()
Especificaciones de resultados (cifras de control, registros de salida por archivo, etc.) ()
5. ¿Existen órdenes de proceso para cada corrida en la computadora (incluyendo pruebas, compilaciones y producción)?
SI () NO ()
6. ¿Son suficientemente claras para los operadores estas órdenes?
SI () NO ()
7. ¿Existe una estandarización de las ordenes de proceso?
SI () NO ()
8. ¿Existe un control que asegure la justificación de los procesos en el computador?
SI () NO ()
9. ¿Cómo programan los operadores los trabajos dentro del departamento de cómputo?
Primero que entra, primero que sale ()
se respetan las prioridades, ()
Otra (especifique) ()
10. ¿Los retrasos o incumplimiento con el programa de operación diaria, se revisa y analiza?
SI () NO ()
11. ¿Quién revisa este reporte en su caso?
12. Analice la eficiencia con que se ejecutan los trabajos dentro del departamento de cómputo, tomando en cuenta equipo y operador, a través de inspección visual, y describa sus observaciones.
13. ¿Existen procedimientos escritos para la recuperación del sistema en caso de falla?
14. ¿Cómo se actúa en caso de errores?
15. ¿Existen instrucciones específicas para cada proceso, con las indicaciones pertinentes?
16. ¿Se tienen procedimientos específicos que indiquen al operador que hacer cuando un programa interrumpa su ejecución u otras dificultades en proceso?



17. ¿Puede el operador modificar los datos de entrada?
18. ¿Se prohíbe a analistas y programadores la operación del sistema que programo o analizo?
19. ¿Se prohíbe al operador modificar información de archivos o bibliotecas de programas?
20. ¿El operador realiza funciones de mantenimiento diario en dispositivos que así lo requieran?
21. Las intervenciones de los operadores:
¿Son muy numerosas? SI () NO ()
Se limitan los mensajes esenciales? SI () NO ()
Otras (especifique)_____
22. ¿Se tiene un control adecuado sobre los sistemas y programas que están en operación?
SI () NO ()
23. ¿Cómo controlan los trabajos dentro del departamento de cómputo?
24. ¿Se rota al personal de control de información con los operadores procurando un entrenamiento cruzado y evitando la manipulación fraudulenta de datos?
SI () NO ()
25. ¿Cuentan los operadores con una bitácora para mantener registros de cualquier evento y acción tomada por ellos?
SI ()
por máquina ()
escrita manualmente ()
NO ()
26. ¿Existen procedimientos para evitar las corridas de programas no autorizados?
SI () NO ()
27. ¿Existe un plan definido para el cambio de turno de operaciones que evite el descontrol y discontinuidad de la operación?
28. Verificar que sea razonable el plan para coordinar el cambio de turno.
29. ¿Se hacen inspecciones periódicas de muestreo?
SI () NO ()
30. Enuncie los procedimientos mencionados en el inciso anterior:
31. ¿Se controla estrictamente el acceso a la documentación de programas o de aplicaciones rutinarias?
SI () NO ()
¿Cómo?_____
32. Verifique que los privilegios del operador se restrinjan a aquellos que le son asignados a la clasificación de seguridad de operador.
33. ¿Existen procedimientos formales que se deban observar antes de que sean aceptados en operación, sistemas nuevos o modificaciones a los mismos?
SI () NO ()



34. ¿Estos procedimientos incluyen corridas en paralelo de los sistemas modificados con las versiones anteriores?
SI () NO ()

35. ¿Qué precauciones se toman durante el periodo de implantación?

36. ¿Cómo se controlan las llaves de acceso (Password)?.

2. FECHA DE ENTREGA

.....

3. FORMA DE ENTREGA

- El trabajo es individual.
- Entregar todo lo trabajado en un archivo de Microsoft Word.
- Subir al enlace en el aula virtual



Referencias bibliográficas

Básica

Pakhira, M. (2013). *Database Management System*. New Delhi, India: PHI Learning Private Limited.

Complementaria:

Gabillaud, J. (2015). *SQL Server 2014, Administración de una base de datos transaccional*. España: ENI.

Rajiv, E. (2013). *Database Management System (DBMS)*. (3ª ed.). India: S. Chand & Company LTD.

Ricardo, C. (2015). *Bases de Datos*. España: McGraw-Hill.

Silberschatz, A., Korth H., Sudarshan S. (2013). *Fundamentos de Bases de Datos*. España: McGraw-Hill.