**IMPORTANTE:** Al finalizar la prueba la debe enviar por medio de la plataforma, este proceso lo realiza entrando al inicio de la plataforma donde encontrará en la parte superior derecha la opción **MENSAJE**S, allí podrá enviarle al profesor del curso la prueba por medio de la selección de adjuntos, con el asunto: **EVALUACIÓN SQLSERVER.**

**Nombres y apellidos del evaluado:**

**Cargo al que aspira:**

Se requiere de una solución computacional para sistematizar el proceso de radicación de comunicaciones internas y externas para la compañía ALPHA.

La solución deberá permitir la radicación de comunicaciones y su posterior consulta por parte del personal autorizado de la compañía.

Actualmente nuestra compañía realiza un proceso manual donde se ejecutan los siguientes pasos:

1. La comunicación llega físicamente a las instalaciones de nuestra compañía y es entregada al gestor documental.
2. El gestor documental se encarga de clasificarlas según sean internas o externas, les asigna un número de comunicación y las archiva.
3. Finalmente la correspondencia es entregada al funcionario correspondiente.

Adicionalmente, se debe tener en cuenta que se desea que el nuevo sistema permita:

* Almacenar la correspondencia en formato digital
* Auditar todas las transacciones realizadas en la base de datos
* Gestionar la asignación de los consecutivos para la correspondencia teniendo en cuenta los siguientes prefijos y número de dígitos: Correspondencia Interna CI00000000, Correspondencia externas CE00000000, por ejemplo, para el primer comunicado interno radicado el consecutivo asignado será CI00000001.
* Almacenar la información de contacto de los remitentes y destinatarios
* Manejar una seguridad por roles, el administrador debe poder hacer todo tipo de acciones, el gestor puede radicar y consultar correspondencia pero no modificar datos administrativos y los destinarios solo pueden consultar correspondencia radicada dirigida a ellos.

**Evaluación de conocimientos en modelado de base de datos con SQL Server y TSQL**

1. Diseñe el modelo de datos para darle solución al sistema de radicación teniendo en cuenta las indicaciones dadas en el enunciado. Realice el diseño haciendo uso de los tipos de datos de SQL Server e ilustrando las llaves foráneas y llaves primarias. El modelo debe quedar en 3NF (Tercera forma normal). **Hint:** tener en cuenta todo lo indicado, es decir, las necesidades del sistema, auditoría, manejo de consecutivos, información digitalizada, seguridad, etc.

Dos cosas deben salir como resultado de esta pregunta:

* 1. Diagrama ilustrando las tablas y las relacionales entre ellas
  2. Descripción de cada tabla, sus columnas y claves con el siguiente formato:

NombreTabla { Llave Primaria 1 (tipo de dato), Llave Primaria 2 (tipo de dato), Columna 1 (tipo de dato), Columna 2 (tipo de dato), …, Llave foránea 1@(tipo de dato), Llave foránea 2@)(tipo de dato) }

**Como se evidencia en el formato**: al lado de cada columna entre paréntesis va el tipo de datos SQL Server, las llaves primarias deben ir subrayadas, las foráneas con el carácter @.

Ejemplo:

Empleado {

EmpleadoId (smallint), Nombres(Varchar),

FechaNacimiento (Date), CargoId@(smallint) ) }

1. Describa y defienda su modelo de datos indicando como soluciona cada uno de los puntos solicitados en el enunciado del problema.

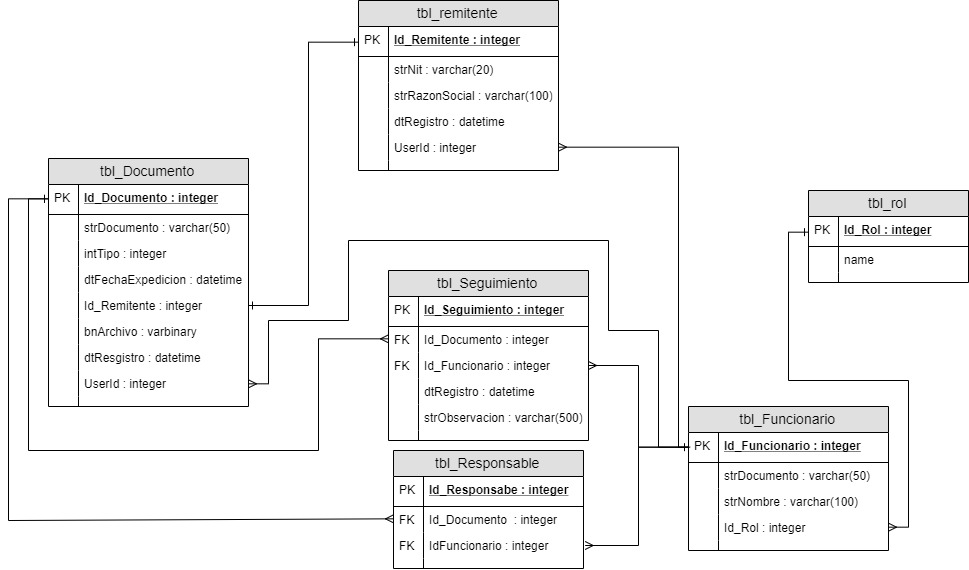
1. Seleccione dos tablas del modelo entidad relación modelado en el punto uno (al menos una de ellas debe tener claves foráneas) y escriba las sentencias DDL usando T-SQL para crear la base de datos y las 2 tablas seleccionadas con sus constraints.

1. Escriba el código T-SQL de un procedimiento almacenado que permita almacenar una correspondencia que acaba de llegar para radicación. **Hint**: tenga en cuenta que debe soportar los dos tipos de correspondencia y hacer la asignación del consecutivo.

1. Escriba el código T-SQL para construir un trigger asociado a la tabla donde se almacena las comunicaciones radicadas, que permita registrar todo cambio en las tablas de auditoría definidas en el modelo de datos.

1. Escriba el código T-SQL de un procedimiento almacenado que permita consultar todas las radicaciones dado un rango de fechas.

1.



Tbl\_Documento{

Id\_Docuemnto(integer), strDocumento(varchar), intTipo(integer), dtFechaExpedicion(datetime), Id\_Remitente@(integer), bnArchivo(varbinary), dtRegistro(datetime), UserId@(integer)

}

Tbl\_Funcionario{

Id\_Funcionario(integer), strDocumento(varchar), strNombre(varchar), Id\_Rol@(integer )

}

Tbl\_remitente{

Id\_Remitente(integer), strNit(varchar),strRazonSocial(varchar), dtResgistro(datetime), UserId@(integer)

}

Tbl\_Seguimiento{

Id\_Seguimiento(integer), Id\_Documento@(integer), Id\_Funcionario@(integer), strObservacion(varchar)

}

Tbl\_Responsabe{

Id\_Responsable(integer), Id\_Documento@(integer), Id\_Funcionario@(integer)

}

Tbl\_rol{

Id\_rol(integer), name(varchar)

}

2.

* Un documento tiene relación con un remitente se debe de ingresar primero la información del remitente.
* Un documento tiene relación con un responsable que sería a quien va dirigido, y también tiene relación con el seguimiento para almacenar todas la transacciones realizadas por los usuarios
* Un Funcionario tiene relación con el seguimiento y con el responsable para tener el registro a quien y las transacciones que realiza
* Un funcionario tiene relación con un rol
* Un funcionario tiene relación con un remitente para tener el registro de quien lo ingreso
* Un funcionario tiene relación con un documento para tener el registro de quien lo ingreso

3.

CREATE DATABASE DB\_Prueba

GO

USE DB\_Prueba

GO/\*\*\*\*\*\* CREACION DE TABLA REMITENTE \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tbl\_Remitente](

[strNit] [varchar](20) NOT NULL,

[strRazonSocial] [varchar](100) NULL,

[dtRegistro] [datetime] NULL,

[UserId] [int] NULL,

CONSTRAINT [PK\_tbl\_Remitente\_1] PRIMARY KEY CLUSTERED

(

[strNit] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* CREACION DE TABLA DOCUMENTO\*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tbl\_Documento](

[Id\_Documento] [int] IDENTITY(1,1) NOT NULL,

[strDocumento] [varchar](50) NULL,

[intTipo] [int] NULL,

[dtFechaExpedicion] [datetime] NULL,

[Id\_Remitente] [varchar](20) NOT NULL,

[bnArchivo] [varbinary](max) NULL,

[dtRegistro] [datetime] NULL,

[UserId] [int] NULL,

CONSTRAINT [PK\_tbl\_Documento] PRIMARY KEY CLUSTERED

(

[Id\_Documento] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

ALTER TABLE [dbo].[tbl\_Documento] WITH CHECK ADD CONSTRAINT [FK\_tbl\_Documento\_tbl\_Documento] FOREIGN KEY([Id\_Documento])

REFERENCES [dbo].[tbl\_Documento] ([Id\_Documento])

GO

ALTER TABLE [dbo].[tbl\_Documento] CHECK CONSTRAINT [FK\_tbl\_Documento\_tbl\_Documento]

GO

/\*\*\*\*\*\* CREACION TABLA SEGUIMIENTO\*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tbl\_Seguimiento](

[Id\_Seguimiento] [int] IDENTITY(1,1) NOT NULL,

[Id\_Documento] [int] NULL,

[Id\_Funcionario] [int] NULL,

[dtRegistro] [datetime] NULL,

[strObservacion] [varchar](500) NULL,

CONSTRAINT [PK\_tbl\_Seguimiento] PRIMARY KEY CLUSTERED

(

[Id\_Seguimiento] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[tbl\_Seguimiento] WITH CHECK ADD CONSTRAINT [FK\_tbl\_Seguimiento\_tbl\_Documento] FOREIGN KEY([Id\_Documento])

REFERENCES [dbo].[tbl\_Documento] ([Id\_Documento])

GO

ALTER TABLE [dbo].[tbl\_Seguimiento] CHECK CONSTRAINT [FK\_tbl\_Seguimiento\_tbl\_Documento]

GO

/\*\*\*\*\*\* PROCEDIMIENTO INSERTAR DOCUMENTO \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[tbl\_DocumentoAdd] --2,'2019-01-01','1152694349',0x20,'2019-01-01',1

(

@intTipo int ,

@dtFechaExpedicion datetime,

@Id\_Remitente varchar(20),

@bnArchivo varbinary,

@dtRegistro datetime,

@UserId int

)

AS

DECLARE @strCodCIF varchar(50)

declare @intConsecutivo int

if not exists(select \* from tbl\_Remitente where strNit=@Id\_Remitente)

begin

insert into tbl\_Remitente (strNit,strRazonSocial,dtRegistro,UserId) values(@Id\_Remitente,@Id\_Remitente+' (Nuevo)',getdate(),@UserId)

end

if @intTipo=1

begin

select @intConsecutivo=ISNULL(CAST(CAST(SUBSTRING(MAX(strDocumento),3,LEN(MAX(strDocumento))) as int) + 1 as varchar),'00000001')

from tbl\_Documento

where intTipo=@intTipo

set @strCodCIF='CI'+ RIGHT('00000000',8-len(cast(@intConsecutivo as varchar)))+CAST(@intConsecutivo as varchar)

end

if @intTipo=2

begin

select @intConsecutivo=ISNULL(CAST(CAST(SUBSTRING(MAX(strDocumento),3,LEN(MAX(strDocumento))) as int) + 1 as varchar),'00000001')

from tbl\_Documento

where intTipo=@intTipo

set @strCodCIF='CE'+ RIGHT('00000000',8-len(cast(@intConsecutivo as varchar)))+CAST(@intConsecutivo as varchar)

end

INSERT INTO tbl\_Documento

(

strDocumento

,intTipo

,dtFechaExpedicion

,Id\_Remitente

,bnArchivo

,dtRegistro

,UserId

)

values

(

@strCodCIF,

@intTipo ,

@dtFechaExpedicion ,

@Id\_Remitente ,

@bnArchivo ,

@dtRegistro ,

@UserId

)

GO

/\*\*\*\*\*\* CREACION DEL TRIGGER\*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [dbo].[TGSeguimiento]

ON [dbo].[tbl\_Documento]

AFTER INSERT

AS

BEGIN

declare @Id\_Documento int

declare @Id\_Remitente varchar(20)

select @Id\_Documento= ROW\_NUMBER() OVER(ORDER BY Id\_Documento ASC),@Id\_Remitente=Id\_Remitente from tbl\_Documento

INSERT INTO tbl\_Seguimiento(Id\_Documento ,Id\_Funcionario ,dtRegistro ,strObservacion ) values (@Id\_Documento,@Id\_Remitente,getdate(),'Se inserto el documento')

END

GO

ALTER TABLE [dbo].[tbl\_Documento] ENABLE TRIGGER [TGSeguimiento]

GO

/\*\*\*\*\*\* CREACION DE PROCEDIMIENTO SELECT \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[tbl\_DocumentoGet]

(

@dtFechaIni datetime,

@dtFechaFin datetime

)

AS

SELECT

Id\_Documento

,strDocumento

,intTipo

,dtFechaExpedicion

,Id\_Remitente

,bnArchivo

,dtRegistro

,UserId

FROM tbl\_Documento

WHERE dtFechaExpedicion>=@dtFechaIni and dtFechaExpedicion<=@dtFechaFin

GO