**INSTITUTO SUPERIOR CIBERTEC**

**DIRECCIÓN ACADÉMICA**

**CARRERA PROFESIONALES**

**CURSO :** BASE DE DATOS AVANZADO I

### NOTA

**PROFESOR :** JORGE CUADROS

**SEMESTRE :** 2022

**CICLO :** Tercero

**SECCIÓN :** T3YL

**FECHA :**

**DURACIÓN :** 45 minutos

|  |
| --- |
| **ALUMNO (A) : Jose David Cardenas Rodriguez** |

**EVALUACIÓN CASO PRACTICO LABORATORIO1 (CL3)**

**Consideraciones generales:**

* Considerar el orden y claridad de las respuestas.
* Guarde el script con el nombre BDAI\_CL3\_T3YL\_XXXX (donde XXXX serán sus apellidos y nombres) o adjunte sus respuestas en este archivo y enviarlo por la plataforma.

**Consolidado**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pregunta** | **Puntaje** | | **Llenar solo en caso de Recalificación justificada** | |
| **Máximo** | **Obtenido** | **Sustento** | **Puntaje** |
| **1** | **2** |  |  |  |
| **2** | **4** |  |  |  |
| **3** | **4** |  |  |  |
| **4** | **4** |  |  |  |
| **5** | **4** |  |  |  |
| **6** | **2** |  |  |  |
| **Nota Recalificada** | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 1** | | | |
| Puntaje |  | | |
| **2 puntos** | 2 Puntos | 01 Puntos | 0 Puntos |
| Diseña la operación según indicaciones. | Diseña la operación en parte. | No desarrolla el requerimiento. |

1. Crear una base de datos BDLibreria con dos archivos de transacciones, crearla en C:\biblioteca. Asignar a dos archivos con tamaño inicial de 20MB, tamaño máximo a 200 mb y a dos archivos un crecimiento de 70%.

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 2** | | | |
| Puntaje |  | | |
| **4 puntos** | 4 Puntos | 02 Puntos | 0 Puntos |
| Diseña la operación según indicaciones. | Diseña la operación con no menos de 3 TABLAS | No desarrolla el requerimiento. |
|  |  |  |  |

create database BDLibreria

on

(

Name=BDLibreria\_Data, Filename='C:\biblioteca\BDLibreria\_Data.MDF',

Size=20MB, MaxSize=200MB, Filegrowth=70%

)

Log on

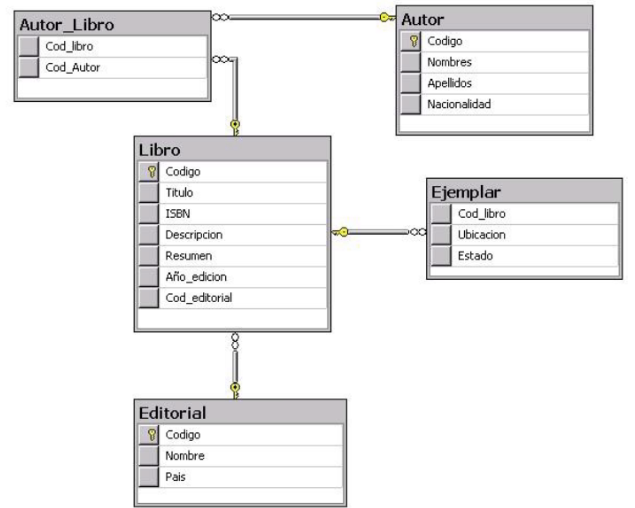
(

Name=BDLibreria\_Log, Filename='C:\biblioteca\BDLibreria\_Log.LDF',

Size=20MB, MaxSize=200MB, Filegrowth=70%

)

Go

1. Cree un esquema y diseñe tablas y relaciones según diagrama. Utilice restricciones PK, FK, Unique, Default y Check. Defina índices. Cree las tablas y relaciones adjunte el script 

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 3** | | | |
| Puntaje |  | | |
| **4 puntos** | 4 Puntos | 02 Puntos | 0 Puntos |
| Ingrese datos según indicaciones. | Ingresa datos de no menos de 3 TABLAS | No desarrolla el requerimiento. |

Create table Libro

(

Codigo varchar(20) not null Primary Key ,

Titulo varchar (200),

ISBN varchar(100),

Descripcion varchar (300),

Resumen varchar (300),

Año\_edicion int,

Cod\_editorial varchar(20)

)

Go

Create table Autor\_Libro

(

Cod\_libro varchar(20),

Cod\_Autor varchar(20)

)

Go

Create table Autor

(

Codigo varchar(20) not null primary Key,

Nombres varchar (100),

Apellidos varchar (100),

Nacionalidad varchar (70)

)

Go

Create table Ejemplar

(

Cod\_libro varchar(20),

Ubicacion varchar (70),

Estado varchar (50)

)

Go

Create table Editorial

(

Codigo varchar(20) not null primary key,

Nombre varchar (50),

Pais varchar (50)

)

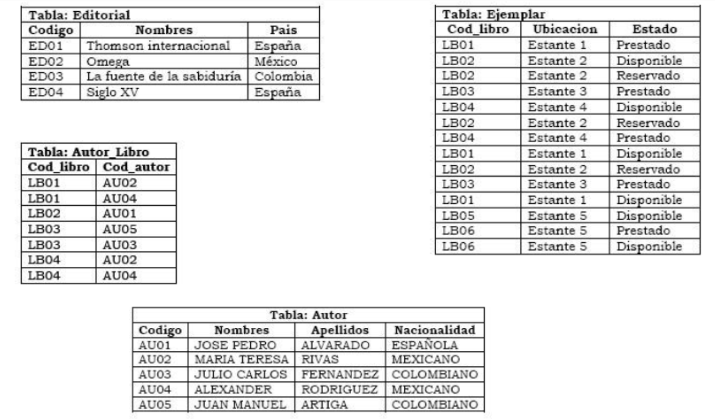
Go

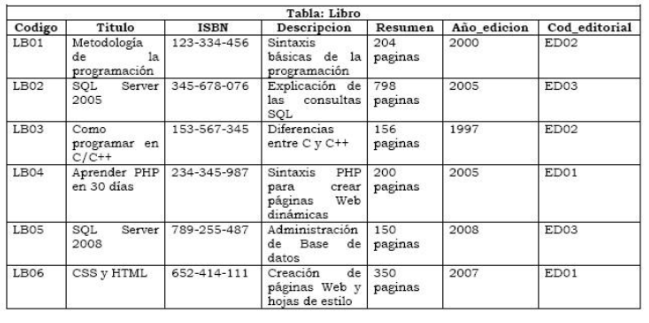
alter table Libro add foreign key (Cod\_editorial) references Editorial(Codigo);

alter table Autor\_Libro add foreign key (Cod\_libro) references Libro(Codigo);

alter table Autor\_Libro add foreign key (Cod\_Autor) references Autor(Codigo);

alter table Ejemplar add foreign key (Cod\_libro) references Libro(Codigo);

1. Ingrese datos, Defina reglas de validación para el ingreso de Estados de la tabla Ejemplar solo se ingresaran (Prestado | Disponible | Reservado) . El año de edición será mayor a 1995



Insert Editorial values ('ED01', 'Thomson internacional', 'España');

Insert Editorial values ('ED02', 'Omega', 'Mexico');

Insert Editorial values ('ED03', 'La Funte de la Sabiduria', 'Colombia');

Insert Editorial values ('ED04', 'Siglo XV', 'España');

Select \* from Editorial;

insert Libro values ('LB01', 'Metodologia de la programacion', '123-334-456','Sintaxis basicas de la programacion', '204 paginas', 2000, 'ED02');

insert Libro values ('LB02', 'SQL Server 2005', '345-678-076','Explicacion de las consultas SQL ', '798 paginas', 2005, 'ED03');

insert Libro values ('LB03', 'Como programar en C/C++', '153-567-345','Diferencias entre C y C++', '156 paginas', 1997, 'ED02');

insert Libro values ('LB04', 'Aprender PHP en 30 dias', '234-345-987','Sintaxis PHP para crear paginas web dinamicas', '200 paginas', 2005, 'ED01');

insert Libro values ('LB05', 'SQL Server 2008', '789-255-487','Administracion de Base de datos', '150 paginas', 2008, 'ED03');

insert Libro values ('LB07', 'CSS y HTML', '652-414-111','Creacion de paginas Web y hojas de estilo', '350paginas', 2007, 'ED01');

Select \* from Libro;

Insert Ejemplar values ('LB01', 'Estante 1', 'Prestado');

Insert Ejemplar values ('LB02', 'Estante 2', 'Disponible');

Insert Ejemplar values ('LB02', 'Estante 2', 'Reservado');

Insert Ejemplar values ('LB03', 'Estante 3', 'Prestado');

Insert Ejemplar values ('LB04', 'Estante 4', 'Disponible');

Insert Ejemplar values ('LB02', 'Estante 2', 'Reservado');

Insert Ejemplar values ('LB04', 'Estante 4', 'Prestado');

Insert Ejemplar values ('LB01', 'Estante 1', 'Disponible');

Insert Ejemplar values ('LB02', 'Estante 2', 'Reservado');

Insert Ejemplar values ('LB03', 'Estante 3', 'Prestado');

Insert Ejemplar values ('LB01', 'Estante 1', 'Disponible');

Insert Ejemplar values ('LB05', 'Estante 5', 'Disponible');

Insert Ejemplar values ('LB06', 'Estante 5', 'Prestado');

Insert Ejemplar values ('LB06', 'Estante 5', 'Disponible');

Select \* from Ejemplar;

Insert Autor values ('AU01', 'Jose Pedro', 'Alvarado', 'Española');

Insert Autor values ('AU02', 'Maria Teresa', 'Rivas', 'Mexicano');

Insert Autor values ('AU03', 'Julio Carlos', 'Fernandez', 'Colombiano');

Insert Autor values ('AU04', 'Alexander', 'Rodriguez', 'Mexicano');

Insert Autor values ('AU05', 'Juan Manuel', 'Artiga', 'Colombiano');

Select \* from Autor;

Insert Autor\_Libro values('LB01', 'AU02');

Insert Autor\_Libro values('LB01', 'AU04');

Insert Autor\_Libro values('LB02', 'AU01');

Insert Autor\_Libro values('LB03', 'AU05');

Insert Autor\_Libro values('LB03', 'AU03');

Insert Autor\_Libro values('LB04', 'AU02');

Insert Autor\_Libro values('LB04', 'AU04');

Select \* from Autor\_Libro;

-- Crear sp de registro de datos para tabla Ejemplar.

Create procedure sp\_ingresoEjemplar

@Codlib varchar(20),

@Ubi varchar (70),

@Est varchar (50)

as

Begin

Begin Try

If @Est = 'Disponible' OR @Est = 'Prestado' OR @Est = 'Reservado'

insert into dbo.Ejemplar

values (@Codlib, @Ubi, @Est)

Else

RaisError ('Error... Solo puede registrar estados: Disponible, Prestado o Reservado. Revise!!', 16, 1)

End Try

Begin Catch

Print Error\_message()

End Catch

end

--Comprobación

Exec sp\_ingresoEjemplar 'LB01', 'Estante 2', 'Disponibless'

select \* from dbo.Ejemplar order by Cod\_libro asc

-- Crear sp de registro de datos para tabla Libro.

Create procedure sp\_ingresoLibro

@Cod varchar(20),

@Tit varchar (200),

@ISBN varchar (100),

@Des varchar (300),

@Res varchar (300),

@AñoEdi int,

@CodEdi varchar (20)

as

Begin

Begin Try

If @AñoEdi > 1995

insert into dbo.Libro

values (@Cod, @Tit, @ISBN, @Des, @Res, @AñoEdi, @CodEdi)

Else

RaisError ('Error... Solo puede registrar libros con años de edición mayor a 1995. Revisar...!!', 16, 1)

End Try

Begin Catch

Print Error\_message()

End Catch

end

--Comprobación

Exec sp\_ingresoLibro 'LB08', 'SQL 2022', '999-333-134', 'Sintaxis SQL', '100 páginas', '1996', 'ED02'

select \* from dbo.Libro order by Codigo asc

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 4** | | | |
| Puntaje |  | | |
| **4 puntos** | 4 Puntos | 02 Puntos | 0 Puntos |
| Implementa script según indicaciones. | Implementa en parte el script | No desarrolla el requerimiento. |

1. *Implementar un script que permita crear el TRIGGER\_MENSAJE que muestre un mensaje al usuario cuando se realice una insercion y la actualizacion a la tabla Libro, pruebe su script*

Create trigger nueva\_transac\_libro

on Libro for insert, update

as

begin

print 'Tabla: Libro. se ha realizado una transaccion.'

end

--Comprobación

insert Libro values ('LB10', 'CSS y HTML', '652-414-111','Creacion de paginas Web y hojas de estilo', '350paginas', 2007, 'ED01');

*Probando script*

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 5** | | | |
| Puntaje |  | | |
| **4 puntos** | 4 Puntos | 02 Puntos | 0 Puntos |
| Implementa script según indicaciones. | Implementa en parte el script | No desarrolla el requerimiento. |

1. *Implemente un procedimiento almacenado que permita mostrar las libros de un determinado autor, se debera de considerar como parametro de entrada el nombre del autor, defina la cabecera del listado e implemente un mensaje de error en el caso que el autor no exista.*

- Crear sp de registro de datos para tabla Ejemplar.

Create procedure sp\_ConsultaLibro\_X\_Autor

@NomAut varchar(100)

as

Begin

If Exists (Select \* from Autor where upper(Nombres) = upper(@NomAut))

select l.Codigo, l.Titulo, l.Descripcion, l.Resumen, a.Nombres, a. Apellidos, a.Nacionalidad

from Libro l

join Autor\_Libro al on l.Codigo=al.Cod\_libro

join Autor a on al.Cod\_Autor=a.Codigo AND upper(a.Nombres) = upper(@NomAut)

Else

Raiserror ('El nombre del autor %s no existe. Revisar.', 16, 1, @NomAut);

end

--Comprobación

Exec sp\_ConsultaLibro\_X\_Autor 'Alexanders'

*Probando script*

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA PREGUNTA 6** | | | |
| Puntaje |  | | |
| **2 puntos** | 2 Puntos | 01 Puntos | 0 Puntos |
| Implementa script según indicaciones. | Implementa en parte el script | No desarrolla el requerimiento. |

1. *Implemente un procedimiento almacenado que Muestre la siguiente informacion de un libro especifico, el codigo del libro debera de ingresarse*

*LIBRO ISBN AñoEdicion Nombre Autor AñosTrascurridos NombreEditorial Estado Ubicación.*

-- Crear sp de registro de datos para tabla Ejemplar.

Create procedure sp\_ConsultaLibro\_X\_Codigo

@Cod varchar(20)

as

Begin

select l.Titulo AS 'LIBRO', l.ISBN AS 'ISBN',

l.Año\_edicion AS 'AñoEdicion',

year(getdate()) - l.Año\_edicion AS 'AñosTranscurridos',

a.Nombres AS 'NombreAutor',

e.Nombre AS 'NombreEditorial', ej.Estado AS 'Estado',

ej.Ubicacion AS 'Ubicación'

from Libro l

join Autor\_Libro al on l.Codigo=al.Cod\_libro

join Autor a on al.Cod\_Autor=a.Codigo

join Editorial e on l.Cod\_editorial=e.Codigo

join Ejemplar ej on l.Codigo=al.Cod\_libro

where l.Codigo = @Cod

end

--Comprobación

Exec sp\_ConsultaLibro\_X\_Codigo 'LB02'

*Probando script*