**INSTITUTO SUPERIOR CIBERTEC**

**DIRECCIÓN ACADÉMICA**

**CARRERA PROFESIONALES**

**CURSO :** BASE DE DATOS AVANZADO I

### NOTA

**PROFESOR :** JORGE CUADROS

**SEMESTRE :** 2022

**CICLO :** Tercero

**SECCIÓN :** T3YL

**FECHA :** 13/08/2022

**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** | **2** |  |  |  |
| **3** | **2** |  |  |  |
| **4** | **2** |  |  |  |
| **5** | **2** |  |  |  |
| **6** | **2** |  |  |  |
| **7** | **2** |  |  |  |
| **8** | **2** |  |  |  |
| **9** | **2** |  |  |  |
| **10** | **2** |  |  |  |
| **Nota Recalificada** | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **RÚBRICA** | | | |
| 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: BDTiendas

CREATE TABLE BOLETA (

numeroBoleta CHAR(15) NOT NULL,

fechaEmision DATE NOT NULL,

monto MONEY NOT NULL

)

GO

------------ LLENADO DE REGISTROS A LA TABLA BOLETA ------------

INSERT INTO BOLETA VALUES ('00000001','22/04/2008',200)

INSERT INTO BOLETA VALUES ('00000002','02/01/2008',350)

INSERT INTO BOLETA VALUES ('00000003','03/10/2008',250)

INSERT INTO BOLETA VALUES ('00000004','01/05/2007',150)

INSERT INTO BOLETA VALUES ('00000005','07/12/2008',200)

INSERT INTO BOLETA VALUES ('00000006','16/08/2008',120)

INSERT INTO BOLETA VALUES ('00000007','03/01/2007',50)

INSERT INTO BOLETA VALUES ('00000008','05/03/2007',200)

INSERT INTO BOLETA VALUES ('00000009','14/09/2008',10)

INSERT INTO BOLETA VALUES ('00000010','18/12/2008',50)

INSERT INTO BOLETA VALUES ('00000011','09/11/2007',100)

INSERT INTO BOLETA VALUES ('00000012','06/10/2007',80)

Resolver los siguientes requerimientos

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 BOLETA, pruebe su script*

create trigger ingresar\_fecha

on BOLETA after insert

as begin declare

@numeroBoleta CHAR(15),

@fechaEmision date,

@monto MONEY

if @fechaEmision is null

print 'Falta ingresar fecha de Emision'

rollback transaction

end

--Probando Script

INSERT INTO BOLETA VALUES ('00000013','',80)

create trigger boleta\_actualizar

on BOLETA

for update

as

raiserror('Los datos de la tabla "Boleta" no pueden modificarse',10,1)

rollback transaction

go

--Probando Script

update BOLETA set fechaEmision='06/10/2008' where numeroBoleta='00000013'

go

1. *Implementar un script que permita crear el procedimiento almacenado SP\_TOTALBOLETAS de la tabla BOLETA. Que cuente el total de las boletas x año*

create procedure SP\_TOTALBOLETAS

@fechaEmision varchar(15)

as

begin

select count (\*) as Total\_Boletas\_Año

from BOLETA

where fechaEmision like '%' + @fechaEmision + '%'

end

--Probando Script

exec SP\_TOTALBOLETAS '2008'

1. *Implementar un script que permita crear la funcion FN\_TOTALBOLETAS de un determinado año ingresado por el usuario a la tabla BOLETA. Pruebe su funcion usando una consulta*

CREATE FUNCTION FN\_TOTALBOLETAS (@fechaEmision varchar(15))

RETURNS TABLE

AS

RETURN

(

select \*

from BOLETA

where fechaEmision like '%' + @fechaEmision + '%'

);

--Probando Script

select \* from FN\_TOTALBOLETAS ('2008')

go

1. *Implemente un trigger DDL que permita bloquear la aplicación de las instrucciones ALTER y DROP TABLE en la base de datos* BDTiendas*.*

create trigger TR\_Bloque

on database for drop\_table, alter\_table

as

Begin

raiserror('No esta permitido borrar ni modificar tablas!',16,1)

rollback transaction

end

--Probando script

alter table Boleta

drop column fechaEmision

1. *Implemente un trigger ala tabla Boleta, para cuando se inserte o actualice o se elimine un registro, se muestre un mensaje que indique: se ha realizado una transaccion*

create trigger nueva\_transaccion1

on BOLETA after insert, update, delete

as

begin

print 'se ha realizado una transaccion'

rollback transaction

end

--Prueba de Script

insert into BOLETA VALUES ('00000014','08/10/2009',90)

update BOLETA set fechaEmision='06/10/2008' where numeroBoleta='00000013'

go

delete from BOLETA where fechaEmision='06/10/2008'

1. Crear una base de datos: BDAgencia

Resolver los siguientes requerimientos

1. *Implemente un Trigger que valide la insercion de un nuevo registro de pais siempre y cuando el nombre del pais sea único, emitir un mensaje de error y emitir con que codigo fue registrado.*

alter trigger validar\_insertPais

ON PAIS

FOR INSERT

AS DECLARE @V\_Pais VARCHAR(30),@V\_IdPais CHAR(5)

IF (SELECT COUNT(\*) FROM INSERTED

INNER JOIN PAIS ON PAIS.PAIS=INSERTED.PAIS

WHERE INSERTED.PAIS=PAIS.PAIS)>1

BEGIN

SELECT @V\_Pais=PAIS FROM INSERTED

rollback transaction

SELECT @V\_IdPais=IDPAIS FROM PAIS

WHERE PAIS.PAIS=@V\_Pais

raiserror('El pais ya existe con el codigo %s. ',16,1, @V\_IdPais);

end

GO

--Prueba de Script

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0024','ARGENTINA')

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

create procedure SP\_BuscarReservaPasajero

@pasajeroNombres varchar(15)

as

begin

if (0<(select count (\*)

from RESERVA r

join PAGO p on r.IDRESERVA=p.IDRESERVA

join PASAJERO pa on p.IDPASAJERO=pa.IDPASAJERO

and pa.NOMBRES

like '%' + @pasajeroNombres + '%'))

select r.IDRESERVA, r.FECHA, p.MONTO,

pa.IDPASAJERO, pa.NOMBRES, pa.EMAIL

from RESERVA r

join PAGO p on r.IDRESERVA=p.IDRESERVA

join PASAJERO pa on p.IDPASAJERO=pa.IDPASAJERO

and pa.NOMBRES

like '%' + @pasajeroNombres + '%'

else

print 'No existe reservas para este pasajero.'

end

--Probando Script

exec SP\_BuscarReservaPasajero 'David'

1. *Implemente el procedimiento almacenado que permita actualizar los datos de un detrminado Pasajero para lo cual se debera definir como parametros de entrada lso campos de la tabla, se debe de tener en cuenta que al actualizar no debe enviar el codigo de pais sino su nombre del pais y validar la existencia del mismo con un mensaje de “Pais no registrado en la BD”, si todo es correcto enviar un mesaje “Pasajero actualizado con éxito”*

alter procedure SP\_ActualizarPasajero

@idpasajero char(5),

@nombres varchar(50),

@pais varchar(30),

@telefono char(15),

@email varchar(50)

as

begin

if (0 < (select count(\*) from PAIS where PAIS like '%'+@pais+'%'))

begin

update PASAJERO

set

IDPASAJERO = @idpasajero,

NOMBRES = @nombres,

IDPAIS = (select TOP 1 IDPAIS from PAIS where PAIS like '%'+@pais+'%'),

TELEFONO = @telefono,

EMAIL = @email

where IDPASAJERO = @idpasajero

end

else

print 'Pais '+ @pais +' no registrado en la BD.'

end

--Probando Script

exec SP\_ActualizarPasajero 'P0001', 'Angela', 'Venezuela', '99789933', 'EMAIL'

1. *Implemente un procedimeinto almacenado que permita determinar el numero de reservas y el monto acumulado de las reservas realizadas en un detrminado pais, este debera de ingresarse por su nombre, use parametros de salida*

--Se insertó data complementaria para el ejercicio

INSERT INTO AVION VALUES ('AV01', '10123456789', 'LAN PERU', 'XXX', 200)

INSERT INTO ASIENTO VALUES (1, 'A', 1)

INSERT INTO VUELO VALUES (1,'AE03',1,'AV01','ECONOMICO')

INSERT INTO VUELO VALUES (1,'AE03',2,'AV01','ECONOMICO')

INSERT INTO VUELO VALUES (1,'AE04',3,'AV01','ECONOMICO')

INSERT INTO VUELO VALUES (1,'AE04',4,'AV01','ECONOMICO')

INSERT INTO VUELO VALUES (1,'AE04',6,'AV01','ECONOMICO')

alter procedure SP\_CantidadReserva\_y\_MontoAcumulado\_x\_Pais

@pais varchar(30)

as

begin

Declare @cant\_reserva int, @costo\_reserva money

--Cantidad de reservas por pais.

SET @cant\_reserva = (select count(\*)

from reserva r

join vuelo v on r.IDRESERVA=v.IDRESERVA

join aeropuerto ae on v.IDAERO=ae.IDAERO

join pais p on ae.IDPAIS=p.IDPAIS AND p.PAIS like '%'+@pais+'%')

--Monto acumulado de reservas por pais.

SET @costo\_reserva = (select sum(pag.monto)

from reserva r

join vuelo v on r.IDRESERVA=v.IDRESERVA

join aeropuerto ae on v.IDAERO=ae.IDAERO

join pago pag on r.IDRESERVA=pag.IDRESERVA

join pais p on ae.IDPAIS=p.IDPAIS AND p.PAIS like '%'+@pais+'%')

Select @cant\_reserva as CANTIDAD\_RESERVA, @costo\_reserva as MONTO\_ACUMULADO\_RESERVA

end

--Probando Script

exec SP\_CantidadReserva\_y\_MontoAcumulado\_x\_Pais 'CHI'

1. *Implemente un trigger que permita controlar la eliminacion de un registro de la tabla pasajero, en la cual si dicho pasajero tieneun pago realizado no permita su eliminacion y mostrara un mensaje, caso contrario mostrar el mensaje de eliminacion correcta.*

create trigger validar\_NoEliminarPasajeroSiPago

ON PASAJERO

FOR DELETE

AS BEGIN DECLARE @V\_IdPasajero CHAR(5)

IF (SELECT COUNT(\*) FROM DELETED

INNER JOIN PAGO ON PAGO.IDPASAJERO=DELETED.IDPASAJERO)>1

rollback transaction

raiserror('No puede eliminar al pasajero, ya que tiene pagos realizados.',16,1);

end

GO

--Prueba de Script

DELETE FROM PASAJERO WHERE IDPASAJERO = 'P0001'

GO

--BASE DE DATOS

IF DB\_ID('BDAGENCIA') IS NOT NULL

BEGIN

USE master

DROP DATABASE BDAGENCIA

END

CREATE DATABASE BDAGENCIA

GO

--Activando la base de datos BDAGENCIA

USE BDAGENCIA

GO

--Creando las tablas

CREATE TABLE AEROLINEA(

RUC CHAR(11) NOT NULL,

NOMBRE VARCHAR(40) NOT NULL

)

GO

CREATE TABLE AVION(

IDAVION CHAR(5) NOT NULL,

RUC CHAR(11) NOT NULL,

COMPAÑIA VARCHAR(40) NOT NULL,

TIPO VARCHAR(30) NOT NULL,

PASAJEROS INT NOT NULL

)

GO

CREATE TABLE TARIFA(

CLASE VARCHAR(20) NOT NULL,

PRECIO MONEY NOT NULL,

IMPUESTO MONEY NOT NULL

)

GO

CREATE TABLE RESERVA(

IDRESERVA INT IDENTITY NOT NULL,

COSTO MONEY DEFAULT 0,

FECHA DATE DEFAULT GETDATE()

)

GO

CREATE TABLE PASAJERO(

IDPASAJERO CHAR(5) NOT NULL,

NOMBRES VARCHAR(50) NOT NULL,

IDPAIS CHAR(4) NOT NULL,

TELEFONO CHAR(15) NOT NULL,

EMAIL VARCHAR(50) NOT NULL

)

GO

CREATE TABLE PAIS(

IDPAIS CHAR(4) NOT NULL,

PAIS VARCHAR(30) NOT NULL

)

GO

CREATE TABLE ASIENTO(

NUMVUELO INT NOT NULL,

LETRA CHAR(2) NOT NULL,

FILA INT NOT NULL

)

GO

CREATE TABLE PAGO(

NUMPAGO INT NOT NULL,

IDRESERVA INT NOT NULL,

IDPASAJERO CHAR(5) NOT NULL,

FECHA DATE NOT NULL,

MONTO MONEY NOT NULL

)

GO

CREATE TABLE AEROPUERTO(

IDAERO CHAR(5) NOT NULL,

NOMBRE VARCHAR(40) NOT NULL,

IDPAIS CHAR(4) NOT NULL

)

GO

CREATE TABLE VUELO(

NUMVUELO INT NOT NULL,

IDAERO CHAR(5) NOT NULL,

IDRESERVA INT NOT NULL,

IDAVION CHAR(5) NOT NULL,

CLASE VARCHAR(20) NOT NULL

)

GO

--Implementado las Llaves Primarias

ALTER TABLE AEROLINEA

ADD PRIMARY KEY NONCLUSTERED (RUC)

ALTER TABLE AVION

ADD PRIMARY KEY NONCLUSTERED (IDAVION)

ALTER TABLE AEROPUERTO

ADD PRIMARY KEY NONCLUSTERED (IDAERO)

ALTER TABLE TARIFA

ADD PRIMARY KEY NONCLUSTERED (CLASE)

ALTER TABLE VUELO

ADD PRIMARY KEY NONCLUSTERED (NUMVUELO,IDAERO,IDRESERVA,IDAVION)

ALTER TABLE RESERVA

ADD PRIMARY KEY NONCLUSTERED (IDRESERVA)

ALTER TABLE PAIS

ADD PRIMARY KEY NONCLUSTERED (IDPAIS)

ALTER TABLE ASIENTO

ADD PRIMARY KEY NONCLUSTERED (NUMVUELO)

ALTER TABLE PASAJERO

ADD PRIMARY KEY NONCLUSTERED (IDPASAJERO)

ALTER TABLE PAGO

ADD PRIMARY KEY NONCLUSTERED (NUMPAGO)

GO

--Implementando las llaves Secundarias

ALTER TABLE PAGO

ADD FOREIGN KEY (IDRESERVA) REFERENCES RESERVA

ALTER TABLE PAGO

ADD FOREIGN KEY (IDPASAJERO) REFERENCES PASAJERO

ALTER TABLE AVION

ADD FOREIGN KEY (RUC) REFERENCES AEROLINEA

ALTER TABLE VUELO

ADD FOREIGN KEY (IDAERO) REFERENCES AEROPUERTO

ALTER TABLE VUELO

ADD FOREIGN KEY (IDRESERVA) REFERENCES RESERVA

ALTER TABLE VUELO

ADD FOREIGN KEY (IDAVION) REFERENCES AVION

ALTER TABLE VUELO

ADD FOREIGN KEY (CLASE) REFERENCES TARIFA

ALTER TABLE VUELO

ADD FOREIGN KEY (NUMVUELO) REFERENCES ASIENTO

ALTER TABLE AEROPUERTO

ADD FOREIGN KEY (IDPAIS) REFERENCES PAIS

ALTER TABLE PASAJERO

ADD FOREIGN KEY (IDPAIS) REFERENCES PAIS

GO

------------------------------------------------

--Insertando registros

INSERT INTO AEROLINEA VALUES('10123456789','LAN PERU')

INSERT INTO AEROLINEA VALUES('10123456710','AEROPERU')

INSERT INTO AEROLINEA VALUES('10123456711','TACA')

INSERT INTO AEROLINEA VALUES('10123456712','BIRD PERU')

INSERT INTO AEROLINEA VALUES('10123456713','LAN CUSCO')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0001','PERU')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0002','ARGENTINA')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0003','CHILE')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0004','ECUADOR')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0005','BRASIL')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0006','VENEZUELA')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0007','PARAGUAY')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0008','URUGUAY')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0009','BOLIVIA')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0010','MEXICO')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0011','HONDURAS')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0012','EEUU')

INSERT INTO PAIS (IDPAIS,PAIS) VALUES('0013','PUERTO RICO')

SET DATEFORMAT DMY

GO

INSERT INTO RESERVA (COSTO,FECHA)

VALUES (0,'01/10/11'),

(0,'06/10/11'),

(0,'14/11/11'),

(0,'16/11/11'),

(0,'12/12/11'),

(0,'17/12/11'),

(0,'14/01/12'),

(0,'15/01/12'),

(0,'01/02/12'),

(0,'02/04/12'),

(0,'09/04/12'),

(0,'01/08/12'),

(0,GETDATE())

GO

INSERT INTO PASAJERO

VALUES('P0001','ANGELA TORRES LAZARO','0001','999999999','ATORRES@HOTMAIL.COM'),

('P0002','FERNANDA TORRES LAZARO','0001','999999999','FTORRES@HOTMAIL.COM'),

('P0003','MARIA ZAMORA MEJIA','0005','957564526','MZAMORA@GMAIL.COM'),

('P0004','GUADALUPE ACOSTA FERRER','0002','957564526','GACOSTA@HOTMAIL.COM'),

('P0005','LUZ LAZARO MENOR','0001','999999999','LLAZARO@GMAIL.COM'),

('P0006','KARLA GALLEGOS SILVA','0007','957564526','KGALLEGOS@HOTMAIL.COM'),

('P0007','NERY CALLE DE LA CRUZ','0010','957564526','NCALLE@GMAIL.COM'),

('P0008','HEIDI RENGIFO REATEGUI','0004','957564526','HRENGIFO@HOTMAIL.COM'),

('P0009','MARISOL DIAZ ZAMBRANO','0004','957564526','MDIAZ@GMAIL.COM'),

('P0010','LINDA TUME VARAS','0008','957564526','LTUME@HOTMAIL.COM')

GO

INSERT INTO PAGO

VALUES (1,1,'P0005','01/10/11',500),

(2,2,'P0003','06/10/11',900),

(3,2,'P0008','14/11/11',500),

(4,3,'P0002','16/11/11',1200),

(5,3,'P0001','12/12/11',1500),

(6,5,'P0006','17/12/11',590),

(7,6,'P0003','14/01/12',400),

(8,5,'P0003','15/01/12',1300),

(9,7,'P0008','01/02/12',1000),

(10,3,'P0002','02/04/12',1800),

(11,8,'P0001','09/04/12',1200),

(12,9,'P0006','01/08/12',400),

(13,9,'P0003',GETDATE()+1,800)

GO

INSERT INTO TARIFA

VALUES('SUPER VIP',1200,12),

('VIP',1000,12),

('NACIONAL',800,12),

('ECONOMICO',500,0)

GO

INSERT INTO AEROPUERTO

VALUES('AE01','BARILOCHE','0002'),

('AE02','MAR DEL PLATA','0002'),

('AE03','JORGE CHAVEZ','0001'),

('AE04','SANTIAGO','0003'),

('AE05','AICM','0010'),

('AE06','JOSE JOAQUIN DE OLMEDO','0004'),

('AE07','SIMON BOLIVAR','0006'),

('AE08','SAO PAULO CONGONHAS','0005'),

('AE09','SILVIO PETTIROSSI','0007'),

('AE10','CARRASCO PUERTA DEL SUR','0008')

GO