

Comenzado el	jueves, 18 de julio de 2024, 18:26
Estado	Finalizado
Finalizado en	jueves, 18 de julio de 2024, 19:27
Tiempo empleado	1 hora
Calificación	27,00 de 100,00

Pregunta 1

Incorrecta

Se puntúa -10,00 sobre 20,00

La propiedad ACID de una Transacción que asegurar que sólo se empieza aquellas operaciones que se puede acabar se denomina:

NOTA: Las respuestas incorrectas serán penalizadas

Seleccione una:

- ☐ a. Consistencia
- ☐ b. Durabilidad
- ☐ c. Aislamiento
- ☒ d. Atomicidad ✖
- ☐ e. Ninguna

Respuesta incorrecta.

La respuesta correcta es:
Consistencia

Pregunta 2

Incorrecta

Se puntúa -10,00 sobre 20,00

Al [Esquema de la Base de Datos Preventa](#) se incluye un Procedimiento Almacenado denominado **PA_TotalStockxAlmacen**, el mismo que retorna el stock existente de un producto en un determinado almacén. Es código del PA es el siguiente:

```
CREATE PROCEDURE PA_TotalStockxAlmacen(@xcprd INT, @xcalm INT,@stock INT OUTPUT)
AS
    select @stock=isnull(sum(cant),0) from sumi
    where cprd=@xcprd and calm=@xcalm
RETURN
```

Selecione el Trigger con la lógica correcta que se dispare cada vez que adicione un producto en la pre venta y valide si hay stock suficiente del producto, si no hay Stock se cancele la adición del producto.

NOTA: esta pregunta se penaliza si selecciona una respuesta equivocada.

Selecione una:

- ☐ a.

```
CREATE TRIGGER TR_stock_dventas
ON dventas
FOR INSERT
AS
    DECLARE @cprd INT,@calm INT,@cant DECIMAL,@nvta INT,@xStock DECIMAL
    SELECT @cprd=cprd,@nvta=nvta,@cant=cant FROM INSERTED
    SELECT @calm=calm FROM pventa WHERE nvta=@nvta
    EXECUTE PA_TotalStockxAlmacen @cprd, @calm, @xStock OUTPUT
    IF @cant>@xStock
        BEGIN
            RAISERROR('El producto no tiene Stock suficiente...', 16, 1)
            ROLLBACK -- Aborta el insert realizado en la tabla dventas
            RETURN
        END
```
- ☐ b.

```
CREATE TRIGGER TR_stock_dventas
ON dventas
FOR INSERT
AS
    DECLARE @cprd INT,@calm INT,@cant DECIMAL,@nvta INT,@xStock DECIMAL
    SELECT @cprd=cprd,@nvta=nvta,@cant=cant FROM INSERTED WHERE nvta=@nvta
    SELECT @calm=calm FROM pventa WHERE nvta=@nvta
    EXECUTE PA_TotalStockxAlmacen @cprd, @calm, @xStock OUTPUT
    IF @cant>=@xStock
        BEGIN
            RAISERROR('El producto no tiene Stock suficiente...', 16, 1)
            ROLLBACK -- Aborta el insert realizado en la tabla dventas
            RETURN
        END
```
- ☒ c.

```
CREATE TRIGGER TR_stock_dventas
ON dventas
FOR INSERT
AS
    DECLARE @cprd INT,@calm INT,@cant DECIMAL,@nvta INT,@xStock DECIMAL
    SELECT @cprd=cprd,@nvta=nvta,@cant=cant FROM INSERTED
    SELECT @calm=calm FROM pventa WHERE nvta=@nvta WHERE @itot>0
    EXECUTE PA_TotalStockxAlmacen @cprd, @calm, @xStock OUTPUT
    IF @cant>@xStock
        BEGIN
            RAISERROR('El producto no tiene Stock suficiente...', 16, 1)
            ROLLBACK -- Aborta el insert realizado en la tabla dventas
            RETURN
        END
```

 ✖
- ☐ d. Ninguno de los códigos anteriores

La respuesta correcta es:

```

CREATE TRIGGER TR_stock_dventas
ON dventas
FOR INSERT
AS
    DECLARE @cprd INT,@calm INT,@cant DECIMAL,@nvta INT,@xStock DECIMAL
    SELECT @cprd=cprd,@nvta=nvta,@cant=cant FROM INSERTED
    SELECT @calm=calm FROM pventa WHERE nvta=@nvta
    EXECUTE PA_TotalStockxAlmacen @cprd, @calm, @xStock OUTPUT
    IF @cant>@xStock
        BEGIN
            RAISERROR('El producto no tiene Stock suficiente...', 16, 1)
            ROLLBACK -- Aborta el insert realizado en la tabla dventas
            RETURN
        END

```

Pregunta 3

Correcta

Se puntúa 20,00 sobre 20,00

Para el [esquema de la base de datos demo](#) se debe crear una función que devuelva en una tabla el, código y el nombre de los productos de color **ROJO** suministrados por los proveedores de **Cochabamba** y **Santa Cruz**. Llene los cuadros vacos de la siguiente función para que la misma obtenga los resultados solicitados. Los cuadros deben ser llenados sin espacios en blanco, solamente coloque la tabla por delante de los atributos si existe ambigüedad. Si la función no funciona correctamente se descontara a la calificación obtenida el 30% de los puntos que vale la pregunta.

CREATE FUNCTION ObtenerProductos(@color CHAR(15))

RETURNS ✓

AS

RETURN

(

SELECT ✓

, ✓

FROM ✓

WHERE ✓

IN (SELECT ✓

FROM sumi, ✓

WHERE sumi.

✓

=

✓

AND (ciud='SC'

or

✓)

AND ✓

)

Pregunta 4

Correcta

Se puntúa 20,00 sobre 20,00

En el [Esquema de la Base de Datos Preventa](#) ya esta creada la función **Importe_Descuento**, la misma obtiene el importe a descontar que consiste en calcular el 5% sobre el importe total de la pre venta solamente si el importe total es mayor a 25. El codigo de la funcion es el siguiente:

```
CREATE FUNCTION Importe_Descuento(@nvta INT)
RETURNS DECIMAL(12,5)
AS
BEGIN
    DECLARE @impt DECIMAL(12,5),@porcen DECIMAL(12,5),@ImpDesc DECIMAL(12,5)
    SET @ImpDesc=0.0 SET @porcen=0.0
    SELECT @impt=ISNULL(SUM(impt),0) FROM dventas WHERE nvta=@nvta
    IF @impt>25 SET @ImpDesc=@impt*0.05
    RETURN @ImpDesc
END
```

Seleccione el Trigger con la lógica correcta que actualiza el importe a descontar en una pre venta, cada vez que se adiciona o se elimina un producto de la pre venta:

NOTA: esta pregunta se penaliza si selecciona una respuesta equivocada.

Seleccione una:

- ☒ a.

```
CREATE TRIGGER UpdDescuento
ON dventas
FOR INSERT,DELETE
AS
    DECLARE @nvta INT
    SET @nvta=(SELECT ISNULL(nvta,0) FROM inserted)
    if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
    SET @nvta=(SELECT ISNULL(nvta,0) FROM deleted)
    if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
```

 ✓
- ☐ b.

```
CREATE TRIGGER UpdDescuento
ON dventas
FOR INSERT,DELETE
AS
    DECLARE @nvta INT
    if (SELECT ISNULL(nvta,0) FROM inserted)>0
        UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
    if (SELECT ISNULL(nvta,0) FROM deleted)>0
        UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
```
- ☐ c.

```
CREATE TRIGGER UpdDescuento
ON dventas
FOR INSERT,DELETE
AS
    DECLARE @nvta INT
    if (SELECT COUNT(*) FROM inserted)=1
        UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
    if (SELECT COUNT(*) FROM deleted)=1
        UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
```
- ☐ d. Ninguno de los códigos anteriores

La respuesta correcta es:

```
CREATE TRIGGER UpdDescuento
ON dventas
FOR INSERT,DELETE
AS
    DECLARE @nvta INT
    SET @nvta=(SELECT ISNULL(nvta,0) FROM inserted)
    if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
    SET @nvta=(SELECT ISNULL(nvta,0) FROM deleted)
    if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
```

Pregunta 5

Parcialmente correcta

Se puntúa 7,00 sobre 20,00

Dado el siguiente esquema e instancia de la Base de Datos "Airport"

```
CREATE DATABASE airport;

CREATE TABLE Vuelos
(Nro char(1) not null primary key
Cupo Int);

INSERT INTO Vuelos VALUES ('Y',17)
INSERT INTO Vuelos VALUES ('X',1)
```

Dadas las siguientes transacciones T1 y T2, complete las instrucciones para que el Plan Serie ejecute las operaciones de T1 y T2 , de tal forma de asegurar el Aislamiento y la Consistencia de la Base de Datos.

Transacciones**T1 devuelve 2 asientos del vuelo X**SET TRANSACTION ISOLATION LEVEL ✓

BEGIN TRAN

DECLARE @CupoN INT,@CupoL INT,@CupoU INT,@Asientos INT
WAITFOR DELAY '00:00:06'SELECT @CupoL=cupo FROM Vuelos WITH () ✓

WHERE Nro='X'

BEGIN TRAN

PRINT ' LEER Cupo T1 '+CAST(@cupoL AS CHAR(5))

SET @Asientos=2

SET @CupoN=@Asientos+ ✓

UPDATE vuelos SET cupo=@cupoN WHERE Nro='X'

WAITFOR DELAY '00:00:12'

SELECT @cupoU=cupo FROM Vuelos WITH () ✓

WHERE Nro='X'

PRINT ' UPD Cupo T1 '+CAST(@cupoU AS CHAR(5))

IF (@Asientos+@CupoL)>=0

 ✗

ELSE

 ✗SELECT @cupoL=cupo FROM Vuelos WITH() ✓

WHERE Nro='X'

PRINT ' END Cupo T1 '+CAST(@cupoL AS CHAR(5))

T2 reserva 4 asientos de vuelo XSET TRANSACTION ISOLATION LEVEL ✗

BEGIN TRAN

DECLARE @CupoN INT,@CupoL INT,@CupoU INT,@Asientos INT
WAITFOR DELAY '00:00:03'SELECT @CupoL=cupo FROM Vuelos WITH () ✗

WHERE Nro='X'

BEGIN TRAN

PRINT ' LEER Cupo T1 '+CAST(@cupoL AS CHAR(5))

SET @Asientos=-4

SET @CupoN=@Asientos+ ✗

UPDATE vuelos SET cupo=@cupoN WHERE Nro='X'

WAITFOR DELAY '00:00:09'

SELECT @cupoU=cupo FROM Vuelos WITH () ✗

WHERE Nro='X'

PRINT ' UPD Cupo T1 '+CAST(@cupoU AS CHAR(5))

IF (@Asientos+@CupoL)>=0

 ✗

ELSE

 ✗SELECT @cupoL=cupo FROM Vuelos WITH() ✗

WHERE Nro='X'

PRINT ' END Cupo T1 '+CAST(@cupoL AS CHAR(5))

NOTA: Cada pregunta que seleccione de manera incorrecta tiene una penalización

Ir a...

< Sección anterior