

Pregunta 1

Parcialmente correcta

Se puntúa 24,00 sobre 60,00

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 (
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 (
Nro='X'
PRINT ' UPD Cupo T1 '+CAST(@cupoU AS CHAR(5))
IF (@Asientos+@CupoL)>=0

COMMIT TRAN
ELSE
ROLLBACK TRAN

SELECT @cupoL=cupo FROM Vuelos WITH(
Nro='X'
PRINT ' END Cupo T1 '+CAST(@cupoL AS CHAR(5))
```

T2 reserva 4 asientos de vuelo X

```
SET 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 (
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 (
Nro='X'
PRINT ' UPD Cupo T1 '+CAST(@cupoU AS CHAR(5))
IF (@Asientos+@CupoL)>=0

COMMIT TRAN
ELSE
ROLLBACK TRAN

SELECT @cupoL=cupo FROM Vuelos WITH(
Nro='X'
PRINT ' END Cupo T1 '+CAST(@cupoL AS CHAR(5))
```

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

Pregunta 2

Parcialmente
correcta

Se puntúa
16,00 sobre
40,00

Esquema e Instancia de la Base de Datos “Bancos”

```
-- Tabla de Cuentas de Ahorro
create table Cuentas
(
ncta char(1) not null primary key,      -- Numero de Cuenta
dcta char(15) not null,                  -- Descripción de la Cuenta
propietario char(40) not null,           -- Nombre del propietario
fecha date not null                      -- Fecha de inicio
)
-- Tabla de Movimientos de la Cuenta de Ahorros
create table MovCuentas
(
ntran integer not null,                  --Nro Movimiento
ncta char(1) not null,                  -- Nro de Cuenta
fecha date not null,                    -- Fecha del Mvimiento
impt decimal,                           -- si impt>0 es abono, si impt<0 es retiro
check (impt<>0),
foreign key(ncta) references Cuentas(ncta),
primary key(ntra,ncta)
)
```

La BD de datos tiene programado los siguientes Trigger:

```
create trigger cargo_abono
on MovCuentas
for insert
as
declare @ntra int,@ncta char(1)
declare @impt decimal(12,2),@imptx decimal(12,2)
select @ntra=ntra,@ncta=ncta,@impt=isnull(impt,0) from inserted
if @impt<0 set @imptx=@impt*-1
if @imptx>50
begin
    set @ntra=@ntra+1 set @impt=(@impt*0.10)
    insert into MovCuentas values (@ntra,@ncta,GETDATE(),@impt)
end
create trigger cargo_retiro
on MovCuentas
for insert
as
declare @ntra int,@ncta char(1)
declare @impt decimal(12,2),@imptx decimal(12,2)
select @ntra=ntra,@ncta=ncta,@impt=isnull(impt,0) from inserted
if @impt>50
begin
    set @ntra=@ntra+2 set @impt=(@impt*0.05)*-1
    insert into MovCuentas values (@ntra,@ncta,GETDATE(),@impt)
end
```



Figura 1. Instancia de la Base de Datos Bancos

Dada la siguiente Transacción T, que transfiere fondos entre dos cuentas, retira 107.15 bs de la cuenta A y los abona a la cuenta B. Para sus respuestas considere la instancia de la BD de la Figura 1 y los Trigger programados.