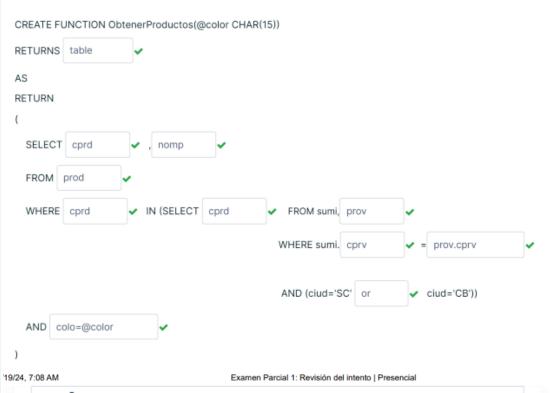
```
La respuesta correcta es:
CREATE TRIGGER UpdDescuento
ON dventas
FOR INSERT, DELETE
        DECLARE @nvta INT
        SET @nvta=(SELECT ISNULL(nvta,0) FROM inserted)
        if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
        SET @nvta=(SELECT ISNULL(nvta,0) FROM deleted)
        if @nvta>0 UPDATE pventas SET ides=dbo.Importe_Descuento(@nvta) WHERE nvta=@nvta
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
                            RAISERROR('El producto no tiene Stock suficiente...', 16, 1)
                            ROLLBACK -- Aborta el insert realizado en la tabla dventas
                            RETURN
              END
CREATE FUNCTION ListaPromedios()
RETURNS table
AS
RETURN
(
   SELECT sumi.cprd, avg(sumi.cant)
   FROM sumi,prod
   WHERE sumi.cprd=prod.cprd
   AND (SELECT count
                                             distinct
           FROM
                                 WHERE prod.cprd=sumi.cprd)>
                                 sumi.cprd
         group by
```

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 vacíos 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 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.





Para el <u>esquema de la base de datos demo</u> se debe nacer una función que retorne una lista con el codigo del proveedor, nombre del proveedor, total cantidad y el total importe suministrado, la lista debe contemplar solamente aquellos proveedores que suministraron productos a los almacenes fuera de su ciudad. Llene los cuadros de la siguiente función para obtener los resultados solicitados. Los cuadros deben ser llenados sin espacios en blanco, solamente coloque la tabla por delante de los 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 ListaProveedores RETURNS table AS BEGIN SELECT sumi.cprv, nomb,SUM(cant),SUM(impt) FROM sumi, prov WHERE sumi. cprv=prov.cprv ✓ IN (SELECT calm FROM alma WHERE alma.ciud AND calm != GROUP cprv,nomb

Comentario: 35-12



Si dos transacciones únicamente leen un determinado elemento de datos, no entran en conflicto entre sí y el orden de las operaciones no es importante

Seleccione una:

Verdadero

Falso

Pregunta 2

Correcta

Se puntúa 10,00 sobre 10,00

Marcar pregunta

Toda Planificación Serie es correcta y deja la Base de Datos consistente, sin embargo, en general, son inaceptables en la práctica (ineficiencia).

Seleccione una:



● Verdadero ✓



Falso



Pregunta 4

Correcta

Se puntúa 10,00 sobre 10,00

Marcar pregunta

Cuando varias transacciones introducidas por usuarios, que se ejecutan de manera concurrente, pueden leer/modificar el mismo elemento de datos X almacenados en la Base de Datos, a esto se denomina Transacciones ...

Respuesta:

concurrentes



```
RETURN
(SELECT cprd, nomp
 FROM prod
 WHERE cprd IN (SELECT cprd
              FROM sumi, prov
              WHERE sumi.cprv=prov.cprv
                     AND (ciud='SC' or ciud='CB')
                     AND colo='ROJO'
)
CREATE FUNCTION Suministro (@cprv INT, @color CHAR(15))
returns DECIMAL(12,3)
as BEGIN
DECLARE @cant DECIMAL(12,3) SET @cant=(SELECT SUM(sumi.cant)
                                   FROM sumi
                                   WHERE exists (SELECT *
                                   FROM prod
                                   WHERE sumi.cprd=prod.cprd
                                          AND prod.colo=@color
AND sumi.cprv=@cprv)
                                   )
       RETURN @cant
END
```

```
CREATE FUNCTION Suministro(@cprv INT)
returns DECIMAL(12,3)
as BEGIN
DECLARE @total DECIMAL(12,3) SET @total=(SELECT SUM(impt)
                              FROM sumi
                              WHERE exists (SELECT nomb
                              FROM prod
                              WHERE sumi.cprd=prod.cprd
                                    AND colo='VERDE')
                                    AND sumi.cprv=@cprv)
      RETURN @total
END
*********************
CREATE FUNCTION ListaProveedores
RETURNS table
AS
BEGIN
SELECT sumi.cprv, nomb, sum(cant), sum(impt)
FROM sumi, prov
WHERE sumi.cprv=prov.cprv AND calm IN(SELECT calm
                              FROM alma
                              WHERE alma.ciud != prov.ciud)
            group by sumi.cprv, nomb
************************
CREATE FUNCTION ObtenerProductos(@color CHAR(15))
RETURNS table
```

AS

```
RETURN

(SELECT cprd, nomp

FROM prod

WHERE cprd IN (SELECT cprd

FROM sumi,prov

WHERE sumi.cprv=prov.cprv

AND (ciud='SC' or ciud='CB')

AND colo='ROJO'

)
```