***IES Virgen del Espino. DAW1.***

***Bases de Datos. 17-02-2016***

Tenemos una base de datos llamada **almacen** que, mediante 2 tablas, controla el stock de un almacén de cereales. Las tablas son las siguientes:

|  |  |
| --- | --- |
| Tabla: **productos** |  |
| Tabla: **movimientos**  En el campo c\_v se coloca una c si ha ingresado producto en el almacén y una v si sale producto del almacén |  |

Se pide el código que permita realizar las siguientes acciones

1.- Función **cuanto\_queda(varchar). 2 puntos**

Se le pasa la denominación de un producto y devuelve la cantidad de kg que quedan en el almacén.

Por ejemplo, cuanto\_queda(‘Trigo’) devolverá 47600, ya que han entrado 16800 kg y 32500 kg y han salido 1700 kg. 16800+32500-1700=47600

2.- Función **cuanto\_gastado(varchar). 2 puntos**

Se le pasa la denominación de un producto y devuelve el dinero que queda por recuperar en ese producto.

Por ejemplo cuanto\_gastado(‘Trigo’) devolverá 10312,70, ya que

* hemos gastado 16800x0.21+32500x0.221=10710.50
* hemos ingresado 1700x0.234=397.80
* en total 10710.50-397.80=10312.70

3.- Procedimiento **inventario()**. 2 puntos

Coloca en los campos precio\_medio y stock los valores correspondientes para cada tipo de producto. Hacerlo con instrucciones SQL

4.- Procedimiento **inventario2()**. 2 puntos

Hace lo mismo que inventario() pero mediante un cursor que recorra uno a uno los diferentes registros de la tabla productos y le ponga su stock y su precio medio

5.- Realizar el o los **disparadores necesarios** para que cuando se haga cualquier acción de mantenimiento de la tabla movimientos (altas/bajas/modificación) ponga en los campos precio\_medio y stock de la tabla productos los valores correctos

-- Devuelve el numero de kg que queda del producto p

DECLARE c int;

DECLARE entra double;

DECLARE sale double;

-- Recupero el codigo del producto p

SELECT codigo INTO c FROM productos WHERE denominacion=p;

-- Cantidad de producto comprado

SELECT sum(kg) INTO entra FROM movimientos WHERE producto=c AND c\_v='c';

-- Cantidad de producto vendido

SELECT sum(kg) INTO sale FROM movimientos WHERE producto=c AND c\_v='v';

-- Por si los nulos

IF entra is null then set entra=0; end if;

IF sale IS NULL THEN SET sale=0; END IF;

-- Devuelvo la diferencia

RETURN entra-sale;

-- Devuelve el importe gastado en el producto p

DECLARE c int;

DECLARE gastado double;

DECLARE ingresado double;

-- Recupero el codigo del producto p

SELECT codigo INTO c FROM productos WHERE denominacion=p;

-- Pasta gastada en el producto p

SELECT sum(kg\*precio) INTO gastado

FROM movimientos

WHERE producto=c AND c\_v='c';

-- Pasta ingresada por la venta del producto p

SELECT sum(kg\*precio) INTO ingresado

FROM movimientos

WHERE producto=c AND c\_v='v';

-- Por si los nulos

IF gastado is null then set gastado=0; end if;

IF ingresado IS NULL THEN SET ingresado=0; END IF;

-- Devuelvo la diferencia

RETURN gastado-ingresado;

UPDATE productos

SET precio\_medio=(cuanto\_gastado(denominacion)/cuanto\_queda(denominacion)),

stock=cuanto\_queda(denominacion);

DECLARE de varchar(50);

DECLARE cantidad double;

DECLARE pasta double;

DECLARE acabado BOOLEAN DEFAULT FALSE;

DECLARE salchichon CURSOR FOR SELECT denominacion FROM productos;

DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET acabado = TRUE;

OPEN salchichon;

FETCH salchichon INTO de;

WHILE acabado=FALSE DO

SELECT cuanto\_queda(de) INTO cantidad;

SELECT cuanto\_gastado(de) into pasta;

UPDATE productos SET stock=cantidad WHERE denominacion=de;

UPDATE productos SET precio\_medio=pasta/cantidad WHERE denominacion=de;

FETCH salchichon INTO de;

END WHILE;

CLOSE salchichon;

Crear 3 disparadores AFTER INSERT, AFTER DELETE y AFTER UPDATE y poner una única orden:

CALL inventario();