

ASGBD03-Solución a la TAREA

AUTOMATIZACIÓN DE TAREAS: CONSTRUCCIÓN DE GUIONES DE ADMINISTRACIÓN.

Esta es una propuesta de solución orientativa, ya que los ejercicios pueden ser resueltos de diferentes maneras.

1.- Crea una única rutina que sume todos los tantos marcados por los jugadores de cada clase, muestre el resultado con la suma y el nombre del grupo ordenado de mayor a menor y finalmente sume dos puntos al primer grupo y uno al segundo.

Créala para que se utilicen los permisos del usuario 'limitado'@'localhost' para ejecutarla. ¿Qué permisos necesitará tener asociados ese usuario?

NOTA: El ejercicio deberá estar debidamente documentado.

```
-- Ejercicio 1
-- =====
-- El procedimiento muestra un listado en pantalla de la clasificación del torneo de baloncesto de las clases,
-- ordenado de mayor a menor, según los tantos marcados por los jugares de cada clase.
-- Se suma a la puntuación de cada clase dos puntos para la primera y uno para la segunda.
-- Programador.
-- Fecha.
-- No hay parámetros de entrada ni de salida.
-- Sólo se muestra por pantalla la información recuperada por la primera sentencia SELECT.
-- Se emplean 3 variables internas de apoyo:
--     * 2 para almacenar las clases a las que hay que modificar la puntuación.
--     * 1 para almacenar la suma de tantos para poder realizar la SELECT...INTO.
-- Debe existir previamente el usuario limitado en localhost, con los permisos: SELECT y UPDATE.

DELIMITER //
DROP PROCEDURE if exists ejercicio1 //
CREATE DEFINER = 'limitado'@'localhost' PROCEDURE ejercicio1() -- Es necesario que exista este usuario
previamente.
SQL SECURITY DEFINER -- La rutina se ejecutará con los permisos de "limitado". El usuario debe existir
previamente.
BEGIN
-- Necesito dos variables para almacenar los códigos de clase del primero y segundo y luego sumar los puntos.
DECLARE grupo1, grupo2 char(3); -- Variables en las que se almacenan los grupos 1º y 2º en la
clasificación.
DECLARE tantos int; -- Variable de apoyo para poder realizar la SELECT...INTO.
    SELECT clase, sum(tantos_marcados) AS "Tantos totales" FROM Jugadores GROUP BY clase
ORDER BY sum(tantos_marcados) DESC;
    SELECT clase, sum(tantos_marcados) INTO grupo1, tantos FROM Jugadores GROUP BY clase
ORDER BY sum(tantos_marcados) DESC Limit 1;
    SELECT clase, sum(tantos_marcados) INTO grupo2, tantos FROM Jugadores GROUP BY clase
ORDER BY sum(tantos_marcados) DESC Limit 1,1;
    UPDATE Clases SET puntuacion = puntuacion + 2 WHERE codigo = grupo1;
    UPDATE Clases SET puntuacion = puntuacion + 1 WHERE codigo = grupo2;
END //
DELIMITER;
```

El usuario 'limitado'@'localhost' necesitará tener los permisos para **SELECT** y **UPDATE**.

2.- Crea un evento que llame cada semana al procedimiento anterior.

```
CREATE EVENT IF NOT EXISTS ejercicio2
ON SCHEDULE EVERY 1 WEEK
DO CALL ejercicio1();
```

3.- Crea la tabla HCO_CAPITANES con la estructura siguiente:

id -> entero, autoincremental, PK
nomantc -> Nombre del capitán anterior.
apeantc -> Apellido del capitán anterior.
nomactc -> Nombre del capitán actual.
apeactc -> Apellido del capitán actual.
clase -> Nombre de la clase en la que cambia el capitán.
fecambio -> Fecha del cambio de capitán.

Todos los campos son obligatorios.

Codifica un disparador que inserte un registro en HCO_CAPITANES tras cambiar el capitán de una clase.

```
drop table if exists Hco_capitanes;  
create table Hco_capitanes( /* Los únicos registros que puede tener la tabla son los que se han dado de alta */  
id tinyint(3) primary key not null auto_increment,  
nomantc varchar(20) not null, /* Nombre del capitán anterior. */  
apeantc varchar(20) not null, /* Apellido del capitán anterior. */  
nomactc varchar(20) not null, /* Nombre del capitán actual. */  
apeactc varchar(20) not null, /* Apellido del capitán actual. */  
clase char(3) not null, /* Clase en la que cambia el capitán. */  
fecambio date not null /* Fecha del cambio de capitán. */  
) engine = innodb;
```

```
create trigger ejercicio3 after update on Clases For Each Row  
if (OLD.capitan != new.capitan) then  
insert into Hco_capitanes set  
nomantc = (SELECT nombre FROM Jugadores WHERE codalumno = OLD.capitan),  
apeantc = (SELECT apellido FROM Jugadores WHERE codalumno = OLD.capitan),  
nomactc = (SELECT nombre FROM Jugadores WHERE codalumno = NEW.capitan),  
apeactc = (SELECT apellido FROM Jugadores WHERE codalumno = NEW.capitan),  
clase = NEW.codigo,  
fecambio = CURRENT_DATE();  
end if;
```