

Evaluación: TERCERA EVALUACIÓN	Fecha:16-5-2024	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

1. **Ganador_etapa_max_altura_media** - Crea una función que muestre cual es el ganador de la etapa (etapa, dorsal y nombre) que tiene la mayor altura media de sus puertos. **(1 punto)**.
Si hubiera varias etapas con la misma altura media, elige la que tenga más kilómetros **(0.25 puntos)**

Haz que devuelva la siguiente frase:

"El ciclista con dorsal.... ha ganado la etapa ..., que tiene una altura media de ... metros".

```
mysql> delimiter //
mysql> drop function if exists ejercicio1;
-> create function ejercicio1() returns varchar(255) deterministic
-> begin
-> declare ganador varchar(255);
-> select concat("el ciclista ", ciclista.nombre, " con dorsal ", ciclista.dorsal, " ha ganado la etapa ", puerto.netapa, " que tine una altura media de ", avg(puerto.altura)) into ganador from ciclista, puerto where ciclista.dorsal = puerto.dorsal group by puerto.netapa order by avg(puerto.altura) desc limit 1;
-> return ganador;
-> end//
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> select ejercicio1();
+-----+
| ejercicio1() |
+-----+
| el ciclista Alex Zulle con dorsal 3 ha ganado la etapa 4 que tine una altura media de 2557.8 |
+-----+
1 row in set (0.00 sec)

mysql>
```

2. **Ganancias_por_equipo** (2.25 puntos).

- Cada equipo se queda el 40% de las ganancias de sus ciclistas, sabiendo esto, crea una función a la que pasando por consola un dorsal de un ciclista nos devuelva cuánto dinero ha ganado su equipo (teniendo en cuenta a todos los miembros del equipo).
- Si de un equipo no ha ganado dinero nadie, debe aparecer que la cantidad es 0 no puede aparecer null.
- Pruébalo con los ciclistas 1,3,4 y 27.

```
mysql> delimiter //
mysql> drop function if exists ejercicio2;
-> create function ejercicio2(dorsalCiclista int) returns float deterministic
-> begin
-> declare equipoCiclista varchar(255);
-> declare dineroEquipoEntero float default 0;
-> declare dineroFinalEquipo float default 0;
->
-> select nomeq into equipoCiclista from ciclista where dorsal = dorsalCiclista;
-> select sum(maillot.premio) into dineroEquipoEntero from ciclista, maillot where ciclista.dorsal = llevar.dorsal and llevar.codigo = maillot.codigo and nomeq = equipoCiclista group by nomeq;
->
-> if dineroEquipoEntero = 0 then
-> RETURN 0;
-> else
-> set dineroFinalEquipo = dineroEquipoEntero + ((dineroEquipoEntero*40)/100);
-> return dineroFinalEquipo;
-> end if;
-> end//
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

Evaluación: TERCERA EVALUACIÓN	Fecha:16-5-2024	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

3. Rellena_dificultad() (1 ptos)

- Antes de seguir añade una columna a la tabla puerto llamada “dificultad”.
- Una vez hecho esto crea el procedimiento que rellenará la dificultad de los registros existentes en función de la pendiente si la pendiente es de menos de 20 dificultad=fácil, entre 20 y 30 dificultad=media y mayor de 30 dificultad=máxima.

```
mysql> alter table puerto add dificultad varchar(255);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> delimiter //
mysql> drop procedure if exists rellenarDificultad;
-> create procedure rellenarDificultad()
-> begin
-> update puerto set dificultad = "facil" where pendiente < 20;
-> update puerto set dificultad = "media" where pendiente > 20 and pendiente < 30;
-> update puerto set dificultad = "maxima" where pendiente > 30;
-> end//
Query OK, 0 rows affected, 1 warning (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call rellenarDificultad();
Query OK, 2 rows affected (0.01 sec)

mysql> select * from puerto;
```

nompuerto	altura	categoria	pendiente	netapa	dorsal	dificultad
p1	2489	1	34	2	3	maxima
p2	2789	1	44	4	3	maxima
Puerto F	2500	E	17	4	2	facil
Puerto fff	2500	E	17	4	2	facil
Puerto nuevo1	2500	a	17	4	1	facil
Puerto otro	2500	E	17	4	1	facil
Puerto1	2500	E	23	1	2	media

```
7 rows in set (0.00 sec)
```

Puse el is null

```
alter table puerto add dificultad varchar(255);
delimiter //
drop procedure if exists rellenarDificultad;
create procedure rellenarDificultad()
begin
update puerto set dificultad = "facil" where pendiente < 20 and dificultad is null;
update puerto set dificultad = "media" where pendiente > 20 and pendiente < 30 and dificultad is null;
update puerto set dificultad = "maxima" where pendiente > 30 and dificultad is null;
end//
delimiter ;
call rellenarDificultad;
select * from puerto;
```

Evaluación: TERCERA EVALUACIÓN	Fecha:16-5-2024	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

4. Alta_puerto() (1.5 puntos)

Crea un procedimiento para dar de alta un puerto dados su nombre, altura, categoría y pendiente. El campo de la dificultad deberá calcularse dentro del procedimiento, es decir no se pasará por parámetro.

```
mysql> delimiter //
mysql> drop procedure if exists Alta_puerto;
-> create procedure Alta_puerto(in p_nombre varchar(255), in p_altura float, in p_categoria varchar(255), in p_pendiente float)
-> begin
-> insert into puerto (nompuerto, altura, categoria, pendiente) values (p_nombre, p_altura, p_categoria, p_pendiente);
-> call rellenarDificultad();
-> end//
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call alta_puerto("aja",2500,1,10);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from puerto;
+-----+-----+-----+-----+-----+-----+-----+
| nompuerto | altura | categoria | pendiente | netapa | dorsal | dificultad |
+-----+-----+-----+-----+-----+-----+-----+
| aja       | 2500  | 1         | 10        | NULL  | NULL  | facil      |
| ala       | 2500  | 1         | 18        | NULL  | NULL  | facil      |
+-----+-----+-----+-----+-----+-----+-----+
```

5. Ganancias_ciclistas (2.5 pts) (Utiliza cursores para resolver el ejercicio. En caso contrario no se corregirá)

Como hemos dicho antes cada equipo se queda el 40% de las ganancias de sus ciclistas, por tanto, el ciclista se quedará el 60%, sabiendo esto, crea un procedimiento que cree una tabla (la clave primaria debe ser el dorsal del ciclista y deberá tener foráneas) para guardar las ganancias de todos los ciclistas. Dicha tabla deberá tener las siguientes columnas: *dorsal*, *nombre_ciclista*, *equipo*, *gananciasciclista*.

```
mysql> delimiter //
mysql> drop procedure if exists ejercicio5;
-> create procedure ejercicio5()
-> begin
-> declare nombreCiclista varchar(255);
-> declare dorsalCiclista int;
-> declare equipoCiclista varchar(255);
-> declare premioCiclista int;
-> declare dineroCicl int default 0;
-> declare done BOOLEAN DEFAULT FALSE;
-> declare cursor1 CURSOR for select ciclista.dorsal, ciclista.nombre, ciclista.nomeq, sum(maillot.premio) from ciclista, llevar, maillot where ciclista.dorsal = 11
-> evar.dorsal and llevar.codigo = maillot.codigo and premio > 0 group by dorsal having sum(premio);declare CONTINUE handler for not found set done = true;
-> create table gananciasCiclistas(dorsal_ciclista int primary key, nombre_ciclista varchar(255), equipo varchar(255), gananciasciclista float, FOREIGN key (dorsal_ciclista) references ciclista (dorsal));
->
-> open cursor1;
-> bucle: LOOP
-> fetch cursor1 into dorsalCiclista, nombreCiclista, equipoCiclista, premioCiclista;
-> if done then
-> leave bucle;
-> end if;
-> set dineroCicl = dineroCicl + ((premioCiclista*60)/100);
-> insert into gananciasCiclistas(dorsal_ciclista, nombre_ciclista, equipo, gananciasciclista) values (dorsalCiclista, nombreCiclista, equipoCiclista, dineroCicl);
-> end loop bucle;
-> close cursor1;
-> end//
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> call ejercicio5();
Query OK, 0 rows affected (0.01 sec)
```

Evaluación: TERCERA EVALUACIÓN	Fecha:16-5-2024	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

```
mysql> delimiter ;
mysql> call ejercicio5();
Query OK, 0 rows affected (0.01 sec)

mysql> select * from gananciasciclistas;
+-----+-----+-----+-----+
| dorsal_ciclista | nombre_ciclista | equipo | gananciasciclista |
+-----+-----+-----+-----+
| 1 | Miguel Indurain | Banesto | 1200000 |
| 2 | Pedro Delgado | Banesto | 1800000 |
| 3 | Alex Zulle | Navigare | 2820000 |
| 4 | Alessio Di Basco | TVM | 3120000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

6. Trigger (1.5 ptos)

- Crea una tabla llamada “mensajes” con una columna “id” int auto_increment y otra llamada “mensaje” de tipo varchar.
- Una vez hecho esto crea un trigger que al insertar un nuevo ciclista vaya rellenando la tabla mensajes según lo siguiente:
 - Si en el equipo hay 10 ciclistas rellene el mensaje que diga: “En el equipoX no se pueden inscribir más ciclistas”
 - Si hay menos de 10 ciclistas el mensaje dirá: “La inscripción sigue abierta en el equipoX”
 - Si hay más de 10 ciclistas el mensaje dirá: “Error, revisar las inscripciones del equipoX”.

```
mysql> create table mensajes(id int primary key auto_increment, mensaje varchar(255));
ERROR 1050 (42S01): Table 'mensajes' already exists
mysql> delimiter //
mysql> drop trigger if exists insertMensaje;
-> create trigger insertMensaje
-> after insert on ciclista for each row
-> begin
-> declare numeroCiclistasEquipo int;
-> declare mensajeAnulacion varchar(255);
->
-> select count(*) into numeroCiclistasEquipo from ciclista where nomeq = "amore vita";
->
-> if numeroCiclistasEquipo = 10 then
-> insert into mensajes (mensaje) values ("en el equipo no se pueden inscribir mas ciclistas");
-> elseif numeroCiclistasEquipo < 10 then
-> insert into mensajes (mensaje) values ("en el equipo la inscripcion sigue abierta");
-> else
-> insert into mensajes (mensaje) values ("Error, revisar las inscripciones del equipo");
-> end if;
-> end//
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Evaluación: TERCERA EVALUACIÓN	Fecha:16-5-2024	Calificación	IES SEGUNDO DE CHOMÓN
Módulo/Materia: Bases de Datos	Curso:1º DAM		
Departamento: INFORMÁTICA			
Nombre y apellidos:			

Puse los concats

```

create table mensajes(id int primary key auto_increment, mensaje varchar(255));
delimiter //
drop trigger if exists insertMensaje;
create trigger insertMensaje
after insert on ciclista for each row
begin
    declare numeroCiclistasEquipo int;
    declare mensaje varchar(255);
    declare mensajeAnulacion varchar(255);

    select count(*) into numeroCiclistasEquipo from ciclista where nomeq = "amore vita";

    if numeroCiclistasEquipo = 10 then
        set mensaje = concat("en el equipo ", new.nomeq , "no se pueden inscribir mas ciclistas");
        insert into mensajes (mensaje) values (mensaje);
    elseif numeroCiclistasEquipo < 10 then
        set mensaje = concat("en el equipo ", new.nomeq , " la inscripcion sigue abierta");
        insert into mensajes (mensaje) values (mensaje);
    else
        set mensaje = concat("Error, revisar las inscripciones del equipo", new.equipo);
        insert into mensajes (mensaje) values (mensaje);
    end if;
end//
delimiter ;
insert into ciclista (dorsal, nombre, edad, nomeq) values (90, "ciclista1", 30, "amore vita");
insert into ciclista (dorsal, nombre, edad, nomeq) values (89, "ciclista2", 31, "amore vita");
insert into ciclista (dorsal, nombre, edad, nomeq) values (88, "ciclista3", 32, "amore vita");
insert into ciclista (dorsal, nombre, edad, nomeq) values (87, "ciclista4", 33, "amore vita");

```

- Una vez hecho el trigger para comprobar si funciona inserta 4 ciclistas al equipo 'Amore Vita' y muestra la tabla mensajes para ver cómo va cambiando.

```

mysql> insert into ciclista (dorsal, nombre, edad, nomeq) values (90, "ciclista1", 30, "amore vita");
Query OK, 1 row affected (0.01 sec)

mysql> insert into ciclista (dorsal, nombre, edad, nomeq) values (89, "ciclista2", 31, "amore vita");
Query OK, 1 row affected (0.00 sec)

mysql> insert into ciclista (dorsal, nombre, edad, nomeq) values (88, "ciclista3", 32, "amore vita");
Query OK, 1 row affected (0.00 sec)

mysql> insert into ciclista (dorsal, nombre, edad, nomeq) values (87, "ciclista4", 33, "amore vita");
Query OK, 1 row affected (0.00 sec)

mysql> select * from mensaje;
ERROR 1146 (42S02): Table 'ciclistmo.mensaje' doesn't exist
mysql> select * from mensajes;
+----+-----+
| id | mensaje                                     |
+----+-----+
| 1  | en el equipo la inscripción sigue abierta |
| 2  | en el equipo la inscripción sigue abierta |
| 3  | en el equipo no se pueden inscribir mas ciclistas |
| 4  | Error, revisar las inscripciones del equipo |
+----+-----+
4 rows in set (0.00 sec)

```

EJERCICIO 1	EJERCICIO 2	EJERCICIO 3	EJERCICIO 4	EJERCICIO 5	EJERCICIO 6	NOTA FINAL
1.25	2.25	1	1.5	2.5	1.5	