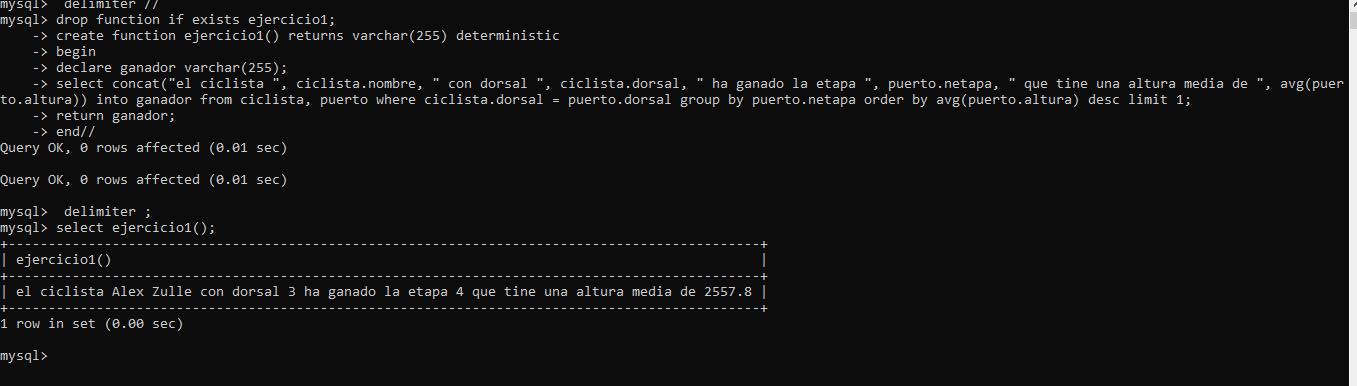
1. **Ganador\_etapa\_max\_altura\_media -** Crea una función que muestre cual es el ganador de la etapa (etapa, dorsal y nombre) que tiene la mayor altura media de sus puertos. **(1 punto).** Si hubiera varias etapas con la misma altura media, elige la que tenga más kilómetros **(0.25 puntos)**

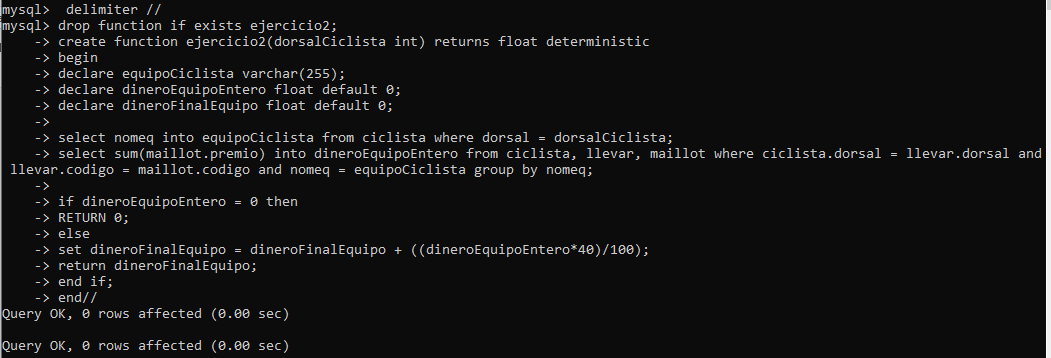
Haz que devuelva la siguiente frase:

***“El ciclista …… con dorsal…. ha ganado la etapa …, que tiene una altura media de … metros”***.



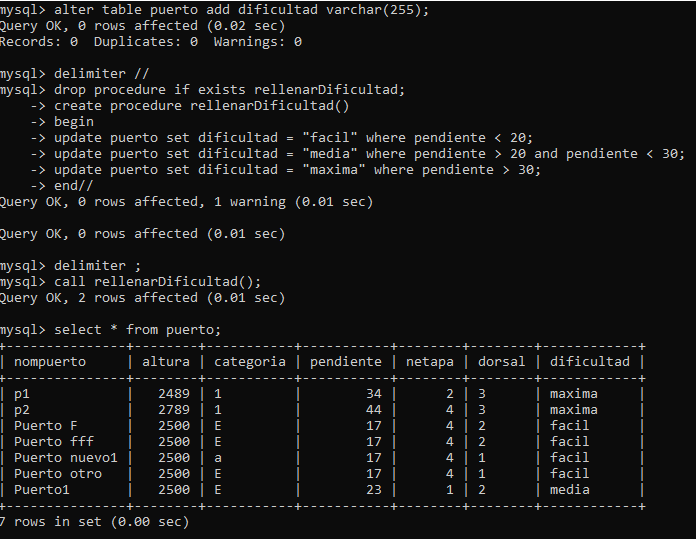
# Ganancias\_por\_equipo (2.25 puntos).

* + 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.
  + Prúebalo con los ciclistas 1,3,4 y 27.

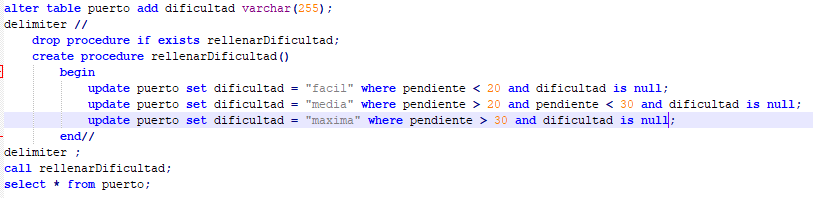


# Rellena\_dificultad() (1 ptos)

* + 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.

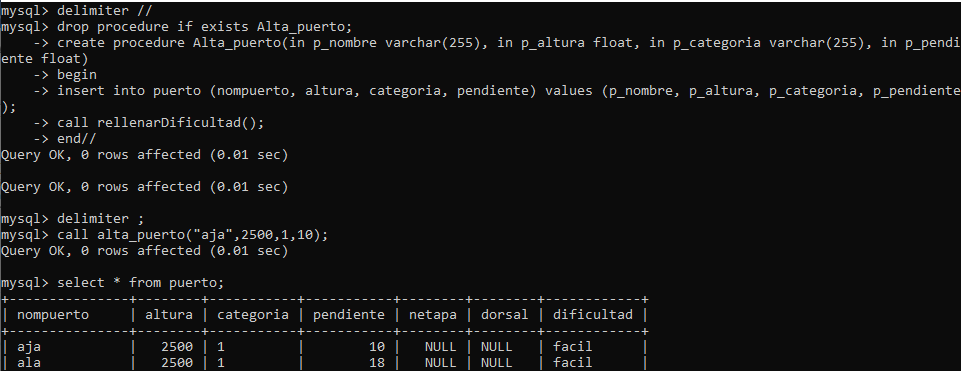


Puse el is null

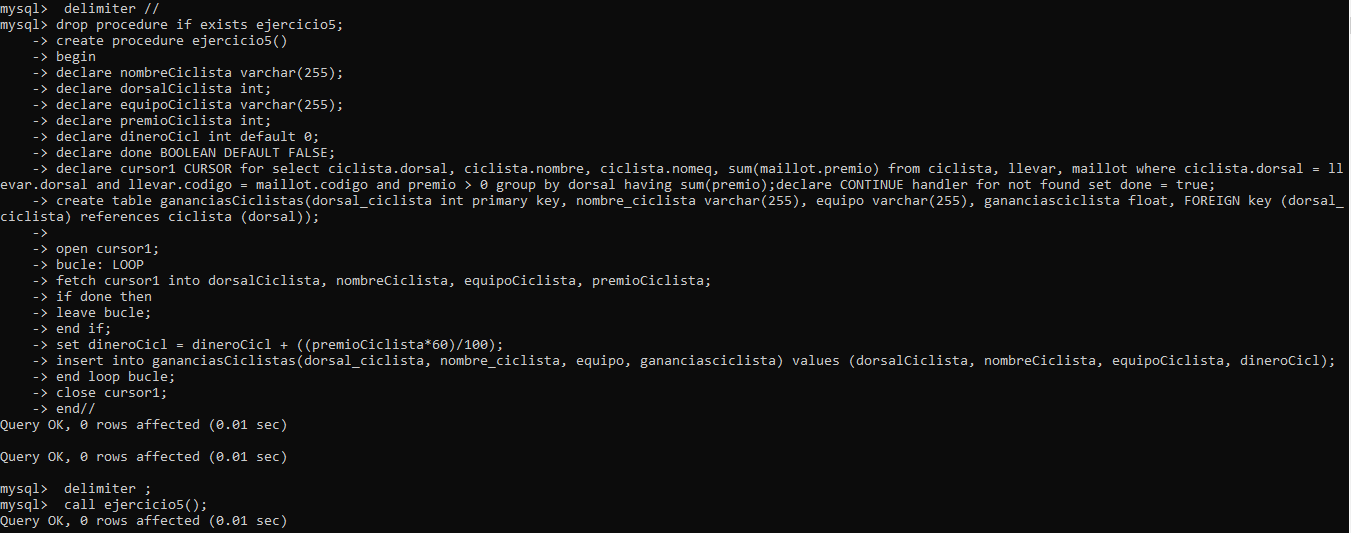


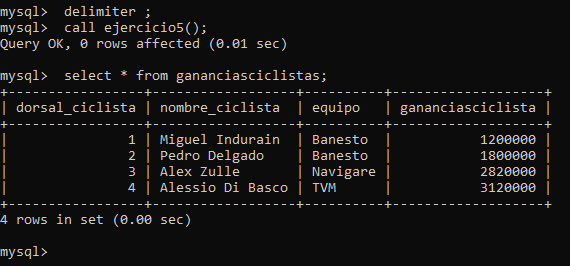
# Alta\_puerto() (1.5 puntos)

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.



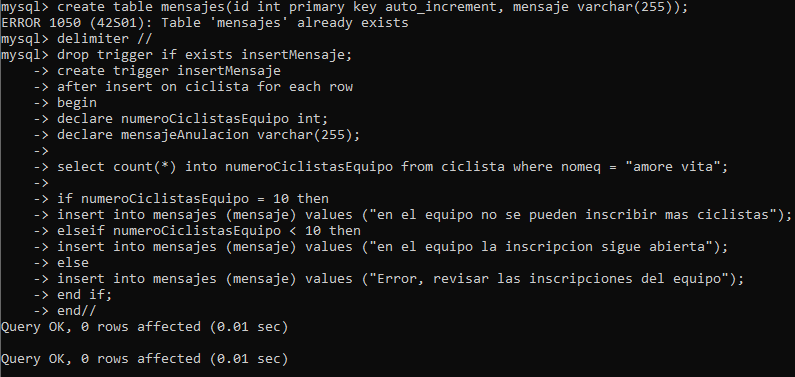
1. **Ganancias\_ciclistas (2.5 ptos) (Utiliza cursores para resolver el *ejercicio. En caso contrario no se corregirá)***

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*.

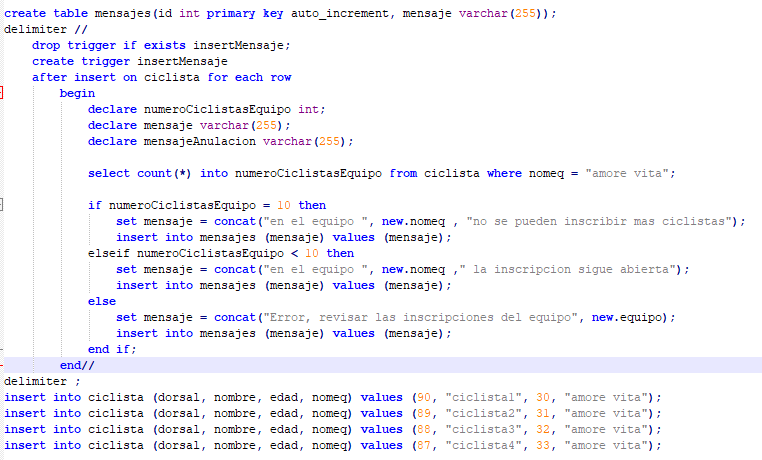


# Trigger (1.5 ptos)

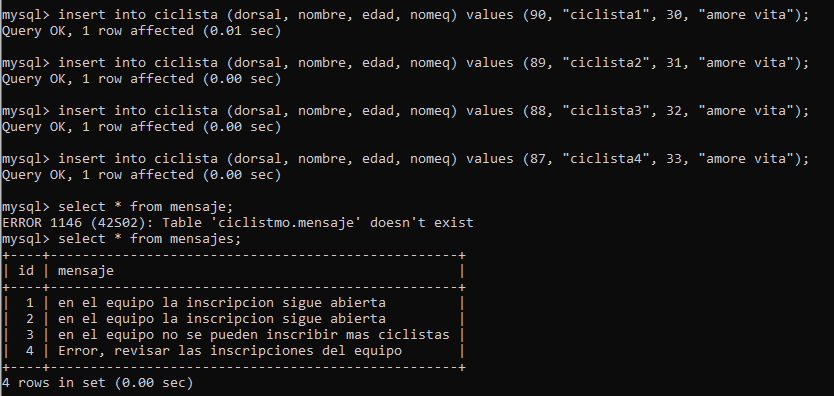
* + Crea una tabla llamada “mensajes” con una columna “id” int auto\_increment y otra llamada “mensaje” de tipo varchar.
  + Una vez hecho esto crea un trigger que al insertar un nuevo ciclista vaya rellenando la tabla mensajes según lo siguiente:
    - Si en el equipo hay 10 ciclistas rellene el mensaje 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”.



Puse los concats



* + 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.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 |  |
|  |  |  |  |  |  |  |