BASE DE DATOS **VELOCIPEDOS.sql**

ACTIVIDADES

1. Realiza un procedimiento mediante el cual rellene el tiempo\_total de la tabla ciclista. Este tiempo total será la suma de los tiempos de cada etapa que el ciclista haya participado. Como parámetro se le proporcionará el nombre del ciclista.

**CREATE** **PROCEDURE** `actualizar\_tiempo\_total`(

**IN** `nom\_ciclista` **VARCHAR**(50)

)

**LANGUAGE** **SQL**

**NOT** **DETERMINISTIC**

**CONTAINS** **SQL**

**SQL** **SECURITY** **DEFINER**

**COMMENT** ''

**BEGIN**

**declare** total **int**;

**set** total=(**select** **sum**(tiempo)

**from** ciclistaxetapa **join** ciclista

**on** ciclistaxetapa.ciclista=ciclista.id

**where** ciclista.nombre=nom\_ciclista);

**update** ciclista **set** tiempo\_total=total

**where** nombre=nom\_ciclista;

**END**

1. Realiza un procedimiento mediante el cual rellene el campo puntuacion de la tabla ciclista. Esta puntuación será la suma de todas las posiciones del ciclista en las etapas que haya participado. Se le pasará al procedimiento como parámetro el id del ciclista.

**CREATE** **PROCEDURE** `actualizar\_puntuacion\_ciclista`(

**IN** `id\_ciclista` **INT**

)

**LANGUAGE** **SQL**

**NOT** **DETERMINISTIC**

**CONTAINS** **SQL**

**SQL** **SECURITY** **DEFINER**

**COMMENT** ''

**BEGIN**

**declare** punt **int**;

**set** punt=(**select** **sum**(posicion)

**from** ciclistaxetapa

**where** ciclista=id\_ciclista);

**update** ciclista **set** puntuacion=punt

**where** id=id\_ciclista;

**END**

1. Realiza un procedimiento que inserte en la tabla GANADOR al ciclista resultante de ganar. El ciclista ganador será aquel que haya obtenido un número más bajo de sumar tiempo\_total y puntuacion. No se le pasará al procedimiento ningún parámetro.

**CREATE** **DEFINER**=`root`@`localhost` **PROCEDURE** `actualizar\_ganador`()

**LANGUAGE** **SQL**

**NOT** **DETERMINISTIC**

**CONTAINS** **SQL**

**SQL** **SECURITY** **DEFINER**

**COMMENT** ''

**BEGIN**

**declare** winner **int**;

**set** winner=(**select** id

**from** ciclista

**where** (tiempo\_total+puntuacion)=(**select** **min**(tiempo\_total+puntuacion)

**from** ciclista));

**insert** ganador **values** (winner);

**END**

1. Realiza una función que devuelva el tiempo que le ha llevado a realizar las etapas de ‘montanya’ al ciclista. Se le pasará como parámetro el nombre del ciclista.

**CREATE** **FUNCTION** `calcula\_tiempo\_montanya`(

`nom\_ciclista` **VARCHAR**(50)

)

**RETURNS** **INT**

**LANGUAGE** **SQL**

**NOT** **DETERMINISTIC**

**CONTAINS** **SQL**

**SQL** **SECURITY** **DEFINER**

**COMMENT** ''

**BEGIN**

**declare** tiempo\_montanya **int**;

**set** tiempo\_montanya=(**select** **sum**(ciclistaxetapa.tiempo)

**from** etapa **join** ciclistaxetapa **join** ciclista

**on** etapa.id=ciclistaxetapa.etapa

**and** ciclistaxetapa.ciclista=ciclista.id

**where** etapa.tipo="montanya"

**and** ciclista.nombre=nom\_ciclista);

**return** tiempo\_montanya;

**END**

1. Realiza un procedimiento para rellenar el campo tiempo\_total de la tabla NACION. Este tiempo será la suma de todos los tiempos de los ciclistas de esa nación que han invertido en las etapas. OJO: deberás sacar los tiempos de la tabla CICLISTAxETAPA. Se le pasará como parámetro el nombre de la nación.

**CREATE** **PROCEDURE** `actualizar\_tiempo\_total\_nacion`(

**IN** `nom\_nacion` **VARCHAR**(50)

)

**LANGUAGE** **SQL**

**NOT** **DETERMINISTIC**

**CONTAINS** **SQL**

**SQL** **SECURITY** **DEFINER**

**COMMENT** ''

**BEGIN**

**declare** total\_nacion **int**;

**set** total\_nacion=(**select** **sum**(tiempo)

**from** ciclistaxetapa **join** ciclista **join** nacion

**on** ciclistaxetapa.ciclista=ciclista.id

**and** ciclista.nacion=nacion.id

**where** nacion.nombre=nom\_nacion);

**update** nacion **set** tiempo\_total=total\_nacion

**where** nombre=nom\_nacion;

**END**

