

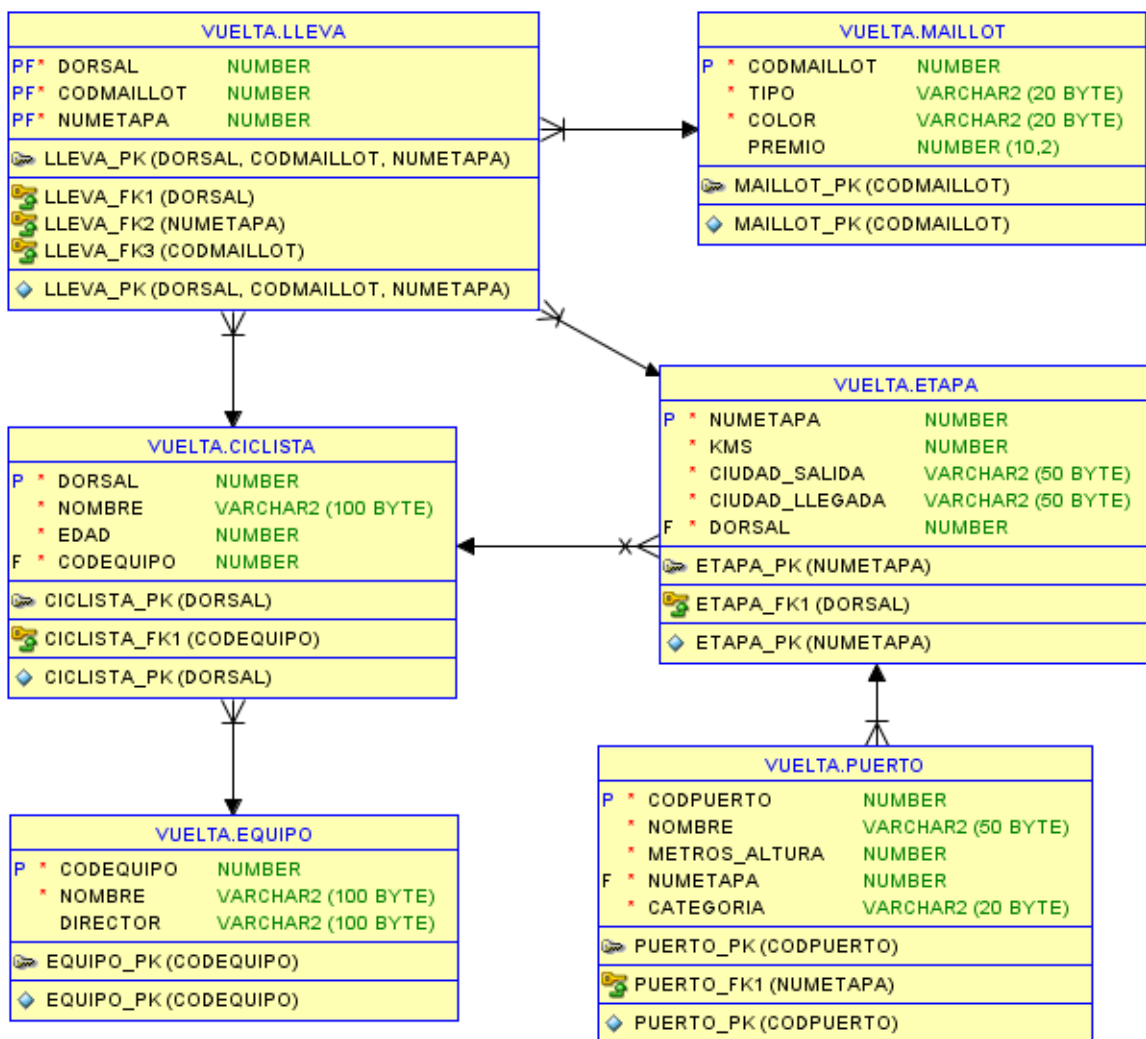


BASES DE DATOS - 06/05/2024		 
DA1D1E	3ª EVALUACIÓN	
NOMBRE		

#### INDICACIONES:

- Se deben controlar todas las excepciones:
  - Parámetros obligatorios en procedimientos y/o funciones (excepciones de usuarios)
  - Excepción general
- Siempre que se pueda poner salida (put\_line) en las excepciones del error.
- Hay que sacar un mínimo de 2 para hacer media con la parte a ordenador.



- Equipo**, contiene información de los equipos que intervienen en la Vuelta, nombre del equipo y su director
- Ciclista**, información de los ciclistas, dorsal, nombre (en este campo se guarda tanto el nombre como los apellidos), edad y equipo al que pertenece.

- **Etapas**, información de las etapas de la vuelta, número de etapa, los kilómetros que tiene, las ciudades de salida llegada de dicha etapa y el dorsal del ciclista ganador de la etapa.
- **Maillot**, información de los distintos maillots que pueden llevar los ciclistas durante la vuelta, su código, su tipo, su color y el premio que le corresponde.
- **Lleva**, contiene información sobre qué ciclistas (su dorsal) llevan qué maillot en qué etapa (numetapa).
- **Puerto**, información de los puertos por los que se pasa en la Vuelta, su altura, su categoría ('Primera', 'Segunda', 'Tercera' y 'Especial'), los metros de altura del puerto y la etapa en la que se pasa por ellos (por el mismo puerto no se puede pasar en 2 etapas).

### Ejercicio 1 (0,5 puntos).

Crea una función que reciba un equipo y me devuelva el nº de etapas que ha ganado ese equipo.

### Ejercicio 2 (0,5 puntos)..

Crea un procedimiento o bloque de código anónimo que haga un listado de todos los equipos y por cada equipo escriba nombre del equipo, director y número de etapas ganadas. Debe utilizar la función anterior para saber el número de etapas ganadas.

### Ejercicio 3 (1 punto).

Crea una función que reciba un código de etapa (numetapa) y me calcule si la etapa es de:

- Alta Montaña: debe tener al menos 2 puertos de categoría 1ª (categoria = PRIMERA) o tener al menos 4 puertos de categorías 1ª (PRIMERA) o 2ª (SEGUNDA)
- Llana: no tiene ningún puerto de PRIMERA y como máximo tiene 2 puertos.
- Media Montaña: Resto de circunstancias

### Ejercicio 4 (0,75 puntos).

Haz un procedimiento que se llamará listar\_ganadores\_etapas. Este procedimiento recibe un parámetro de tipo Varchar2 que es el tipo\_etapa (que corresponde con Alta Montaña, Llana o Media Montaña).

El procedimiento debe escribir un listado (put\_line) con el nombre de los corredores que han ganado una etapa del tipo que nos piden, la ciudad de inicio de esta etapa, la ciudad de final y el número de kilómetros.

### Ejercicio 5 (0,75 puntos).

Durante el transcurso de la vuelta, en la tabla de ciclista se ha añadido un campo de "premio acumulado por el ciclista", como un numérico de 10 posiciones, 2 de ellas decimales. No admite nulos y el valor por defecto es 0.

VUELTA.CICLISTA		
P	* DORSAL	NUMBER
	* NOMBRE	VARCHAR2 (100 BYTE)
	* EDAD	NUMBER
F	* CODEQUIPO	NUMBER
	* PREMIO_ACUMULADO	NUMBER (10,2)
CICLISTA_PK (DORSAL)		
CICLISTA_FK1 (CODEQUIPO)		
CICLISTA_PK (DORSAL)		

Cada vez que un ciclista lleva un maillot en una etapa le corresponde un premio (el maillot, según sea el color tiene un premio u otro y está definido en la tabla de maillots).

El premio que acumula un ciclista (un dorsal a fin de cuentas) se calcula a partir de la tabla "LLEVA" y corresponderá con el total de los premios por haber llevado los diferentes maillots durante las distintas etapas.

Haz un procedimiento o bloque de pl que rellene este campo.

#### Ejercicio 6 (1,5 puntos).

Haz un trigger de inserción sobre la tabla LLEVA y que cada vez que se inserte un registro, busque el premio que corresponde al maillot insertado y lo acumule en el campo "premio\_acumulado" para el ciclista.

El trigger no debe fallar.