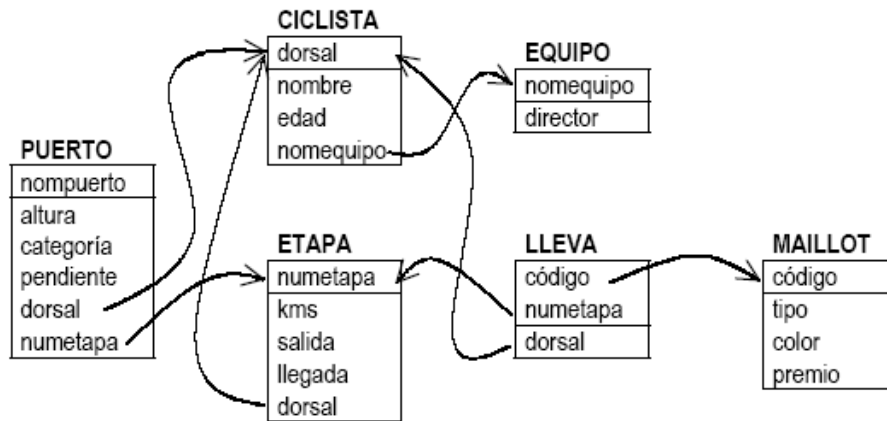


1.- Teniendo en cuenta el siguiente esquema relacional



### Realizar las siguientes consultas:

1. Obtener los datos de las etapas que pasan por algún puerto de montaña y que tienen salida y llegada en la misma población.
2. Obtener las poblaciones que tienen la meta de alguna etapa, pero desde la que no se realiza ninguna salida
3. Obtener el nombre y el equipo de los ciclistas que han ganado alguna etapa llevando el maillot “AMARILLO”, mostrando también el número de etapa.
4. Obtener los datos de las etapas que no comienzan en la misma ciudad en que acaba la etapa anterior.
5. Obtener el numero de las etapa que tienen algún puerto de montaña, indicando cuántos tiene cada una de ellas.
6. Obtener el nombre y la edad de los ciclistas que han llevado dos o más maillots en una misma etapa.
7. Obtener el nombre y el equipo de los ciclistas que han llevado algún maillot o que han ganado algún puerto.
8. Obtener el nombre y el equipo de los ciclistas que han llevado algún maillot y que han ganado algún puerto.
9. Obtener los datos de los ciclistas que han vestido todos los maillots (no necesariamente en la misma etapa)
10. Obtener el código y el color de aquellos maillots que sólo han sido llevados por ciclista de un mismo equipo.
11. Obtener los números de las etapas que no tienen puertos de montaña.
12. Obtener el nombre y la edad de los ciclista que han ganado alguna etapa
13. Obtener la edad media de los ciclistas que han ganado alguna etapa.
14. Obtener el nombre de los puertos de montaña que tienen una altura superior a la altura media de todos los puertos.

15. Obtener los número de etapas, poblaciones de salida y de llegada de las etapas donde se encuentran los puertos con mayor pendiente.
16. Obtener el dorsal y el nombre de los ciclistas que han ganado los puertos de mayor altura
17. Obtener los datos de las etapas cuyos puertos (todos) superan los 1300 metros de altura
18. Obtener el dorsal, nombre de los ciclistas que pertenecen a un equipo de más de cinco ciclistas y que han ganado alguna etapa, indicando también cuántas etapas han ganado

#### ACTUALIZACIONES, INSERCIONES Y BORRADOS

19. Todas las etapa que tienen 2 puertos de montaña han pasado a tener los mismos kms que la etapa número 5 (basada en la nº 5)
20. Todas las etapa que tienen 3 o 4 puertos de montaña y mas de 170 km han pasado a Tener los mismos kms que la etapa número 5 (similar a la anterior).
21. Crea una tabla llamada nuevamaillot con los mismo campos que maillot pero que tenga los datos de aquellos maillots que sólo han sido llevados por ciclista de un mismo equipo. (basada en la nº 10).
22. Pon el nombre en mayúscula y aumenta un año a los ciclista que han ganado los puertos de mayor alturas (basada en la 16).
23. Da de alta un ciclista con el dorsal 101, tu nombre, la misma edad y equipo del ciclista que ha ganado la etapa 8.
24. Crea una tabla llamada ganadores que contenga el dorsal y nombre de los ciclistas que han llevado dos o mas maillote en una misma etapa. La tabla también ha de contener el equipo del ciclista, la etapa y el número de de maillot que han llevado en esa etapa. Los campos de la nueva tabla han de tener los siguientes nombres: Dorsal, Nombre, Equipo, Etapa, Nmaillot. (basada en la 6).
25. Borra los maillot que han sido llevados por Melchor Mauri.  
Nota: has de poner el borrado en cascada en la tabla llevar que referencia a maillot

### Solución ciclista

1. Obtener los datos de las etapas que pasan por algún puerto de montaña y que tienen salida y llegada en la misma población.

```
SELECT DISTINCT E.netapa, km, salida, llegada, E.dorsal
FROM ETAPA E, PUERTO P
WHERE E.netapa = P.netapa
AND E.salida = E.llegada;
```

2. Obtener las poblaciones que tienen la meta de alguna etapa, pero desde la que no se realiza ninguna salida

```
select llegada from etapa
minus
select salida from etapa;
```

3. Obtener el nombre y el equipo de los ciclistas que han ganado alguna etapa llevando el maillot “AMARILLO”, mostrando también el número de etapa.

```
select nombre, nomeq, e.netapa
from ciclista c, etapa e, llevar l, maillot m
where c.dorsal=e.dorsal and
e.netapa=l.netapa and
l.dorsal=c.dorsal
and l.codigo=m.codigo
and m.color='Amarillo';
```

4. Obtener los datos de las etapas que no comienzan en la misma ciudad en que acaba la etapa anterior.

```
SELECT E1.*
FROM ETAPA E1, ETAPA E2
WHERE E1.netapa = E2.netapa + 1
AND E1.salida <> E2.llegada;
```

5. Obtener el numero de las etapa que tienen algún puerto de montaña, indicando cuántos tiene cada una de ellas.

```
select netapa, count(*) "Nº Puertos"
from puerto
group by netapa;
```

6. Obtener el nombre y la edad de los ciclistas que han llevado dos o más maillots en una misma etapa.

```
SELECT DISTINCT nombre, edad
FROM CICLISTA C, LLEVAR L
WHERE C.dorsal = L.dorsal
GROUP BY L.netapa, nombre, edad
HAVING COUNT(*) >= 2;
```

7. Obtener el nombre y el equipo de los ciclistas que han llevado algún maillot o que han ganado algún puerto.

```
SELECT nombre, nomeq
  FROM CICLISTA C, LLEVAR L
 WHERE C.dorsal = L.dorsal
UNION
SELECT nombre, nomeq
  FROM CICLISTA C, PUERTO P
 WHERE C.dorsal = P.dorsal;
```

8. Obtener el nombre y el equipo de los ciclistas que han llevado algún maillot y que han ganado algún puerto.

```
SELECT nombre, nomeq
  FROM CICLISTA C, LLEVAR L
 WHERE C.dorsal = L.dorsal
INTERSECT
SELECT nombre, nomeq
  FROM CICLISTA C, PUERTO P
 WHERE C.dorsal = P.dorsal;
```

9. Obtener los datos de los ciclistas que han vestido todos los maillots (no necesariamente en la misma etapa)

```
SELECT C.dorsal, nombre, edad, nomeq
  FROM CICLISTA C, LLEVAR L
 WHERE C.dorsal = L.dorsal
GROUP BY C.dorsal, nombre, edad, nomeq
HAVING COUNT(DISTINCT L.codigo) = (SELECT COUNT(*)
                                   FROM MAILLOT);
```

10. Obtener el código y el color de aquellos maillots que sólo han sido llevados por ciclista de un mismo equipo.

```
SELECT L.codigo, M.color
  FROM LLEVAR L, CICLISTA C, MAILLOT M
 WHERE L.dorsal=C.dorsal AND L.codigo=M.codigo
GROUP BY L.codigo, M.color
HAVING COUNT(DISTINCT C.nomeq)=1;
```

11. Obtener los números de las etapas que no tienen puertos de montaña.

```
SELECT netapa FROM ETAPA
MINUS
SELECT netapa FROM PUERTO;
```

12. Obtener el nombre y la edad de los ciclista que han ganado alguna etapa

```
SELECT distinct nombre,edad
FROM CICLISTA C, ETAPA E
WHERE C.dorsal=E.dorsal;
```

13. Obtener la edad media de los ciclistas que han ganado alguna etapa.

```
SELECT AVG(C.edad)
FROM CICLISTA C, ETAPA E
WHERE C.dorsal=E.dorsal;
```

14. Obtener el nombre de los puertos de montaña que tienen una altura superior a la altura media de todos los puertos.

```
SELECT nompuerto
FROM PUERTO
WHERE altura > (SELECT AVG(altura)
FROM PUERTO);
```

15. Obtener los número de etapas, poblaciones de salida y de llegada de las etapas donde se encuentran los puertos con mayor pendiente.

```
SELECT DISTINCT E.netapa,E.salida,E.llegada
FROM ETAPA E,PUERTO P
WHERE E.netapa=P.netapa
AND P.pendiente = (SELECT MAX(pendiente)
FROM PUERTO);
```

16. Obtener el dorsal y el nombre de los ciclistas que han ganado los puertos de mayor altura.

```
SELECT DISTINCT C.dorsal,C.nombre
FROM PUERTO P,CICLISTA C
WHERE P.dorsal=C.dorsal
AND P.altura = (SELECT MAX(altura) FROM PUERTO);
```

17. Obtener los datos de las etapas cuyos puertos (todos) superan los 1300 metros de altura.

```
SELECT E.* FROM PUERTO P, ETAPA E
WHERE P.netapa=E.netapa AND P.altura>1300
MINUS
SELECT E.* FROM PUERTO P, ETAPA E
WHERE P.netapa=E.netapa AND P.altura<=1300;
```

```
SELECT * FROM ETAPA E
WHERE 1300 < (SELECT MIN(P.altura) FROM PUERTO P
WHERE P.netapa=E.netapa);
```

18. Obtener el dorsal, nombre de los ciclistas que pertenecen a un equipo de más de cinco ciclistas y que han ganado alguna etapa, indicando también cuántas etapas han ganado.

```
SELECT C.dorsal,C.nombre,COUNT(*) "num_etapas_ganadas"  
FROM CICLISTA C,ETAPA E  
WHERE C.dorsal=E.dorsal  
AND C.nomeq IN (SELECT nomeq  
                FROM CICLISTA  
                GROUP BY nomeq  
                HAVING COUNT(*)>5)  
GROUP BY C.dorsal,C.nombre;
```

### ACTUALIZACIONES, INSERCIONES Y BORRADOS

- 19.- Todas las etapa que tienen 2 puertos de montaña han pasado a tener los mismos kms que la etapa número 5 (basada en la nº 5)

```
update etapa set km = (select km from etapa where netapa=5)  
where netapa in  
(select netapa from puerto p  
 group by netapa  
 having count(netapa)=2);
```

- 20.- Todas las etapa que tienen 3 o 4 puertos de montaña y mas de 170 km han pasado a tener los mismos kms que la etapa número 5

```
update etapa set km = (select km from etapa where netapa=5)  
where netapa in  
(select e.netapa from puerto p, etapa e  
 where p.netapa=e.netapa and km>170  
 group by e.netapa  
 having count(e.netapa)in (3,4));
```

- 21.- Crea una tabla llamada nuevamaillot con los mismo campos que maillot pero que tenga los datos de aquellos maillots que sólo han sido llevados por ciclista de un mismo equipo. (basada en la nº 10)

```
create table nuevamaillot as  
SELECT M.codigo, M.tipo, M.color, M.premio  
FROM LLEVAR L, CICLISTA C, MAILLOT M  
WHERE L.dorsal=C.dorsal AND L.codigo=M.codigo  
GROUP BY M.codigo, M.tipo, M.color, M.premio  
HAVING COUNT(DISTINCT C.nomeq)=1;
```

22.- Poner el nombre en mayúscula y aumenta un año a los ciclista que han ganado los puertos de mayor alturas (basada en la 16)

```
update ciclista set nombre=upper(nombre), edad=edad+1
where dorsal in
(select dorsal from puerto where altura =
(select max(altura) from puerto) );
```

23.- Da de alta un ciclista con el dorsal 101, tu nombre, la misma edad y equipo del ciclista que ha ganado la etapa 8

```
insert into ciclista
select distinct 101,'tu nombre', edad, nomeq from ciclista c, etapa e
where c.dorsal=e.dorsal and e.netapa=8;
```

24.- Crea una tabla llamada ganadores que contenga el dorsal y nombre de los ciclistas que han llevado dos o mas maillote en una misma etapa.  
La tabla también ha de contener el equipo del ciclista, la etapa y el número de de maillot que han llevado en esa etapa. (basada en la 6)  
Los campos de la nueva tabla han de tener los siguientes nombres:  
Dorsal, Nombre, Equipo, Etapa, Nmaillot

```
CREATE TABLE GANADORES (dorsal, nombre, equipo, etapa, nmaillot) AS
SELECT DISTINCT c.dorsal, nombre, nomeq, netapa, count(*)
FROM CICLISTA C, LLEVAR L
WHERE C.dorsal = L.dorsal
GROUP BY netapa, c.dorsal, nombre, nomeq, netapa
HAVING COUNT(*) >=2;
```

25.- Borra los maillot que han sido llevados por Melchor Mauri.

Nota: has de poner el borrado en cascada en la tabla llevar que referencia a maillot

```
ALTER TABLE LLEVAR DROP CONSTRAINT FK_LLEVAR_MAI;
```

```
ALTER TABLE LLEVAR ADD CONSTRAINT FK_llevar_mai FOREIGN KEY
(codigo) REFERENCES maillot (codigo) on delete cascade;
```

```
delete maillot where codigo in (
select distinct codigo from ciclista c, llevar l
where nombre='Melchor Mauri' and
c.dorsal=l.dorsal );
```

19.- todas las etapa que tienen 2 puertos de montaña han pasado a tener los mismos kms que la etapa número 5 (basada en la nº 5)

20.- todas las etapa que tienen 3 o 4 puertos de montaña y mas de 170 km han pasado a tener los mismos kms que la etapa número 5

21.- crea una tabla llamada nuevamaillot con los mismo campos que maillot pero que tenga los datos de aquellos maillots que sólo han sido llevados por ciclista de un mismo equipo. (basada en la nº 10)

22.- poner el nombre en mayuscula y aumentar un año a los ciclista que han ganado los puertos de mayor alturas (basada en la 16)

23.- Da de alta un ciclista con el dorsal 101, tu nombre, la misma edad y equipo del ciclista que ha ganado la etapa 8

24.- Crea una tabla llamada ganadores que contenga el dorsal y nombre de los ciclistas que han llevado dos o mas maillote en una misma etapa.  
La tabla también ha de contener el equipo del ciclista, la etapa y el número de de maillot que han llevado en esa etapa. (basada en la 6)  
Los campos de la nueva tabla han de tener los siguientes nombres:  
Dorsal, Nombre, Equipo, Etapa, Nmaillot

25.- Borra los maillot que han sido llevados por Melchor Mauri.

Nota: has de poner el borrado en cascada en la tabla llevar que referencia a maillot