

Base de datos de ciclismo

Tenemos una base de datos dedicada a gestionar toda la información de una competición ciclista, con las tablas ciclista, equipo, etapa, llevar, maillot y puerto. Necesitamos realizar consultas para obtener las siguientes informaciones:

1. Obtener el código, el tipo, el color y el premio de todos los maillots que hay.

```
SELECT * FROM maillot;  
SELECT codigo, tipo, color, premio FROM maillot;
```

2. Obtener el dorsal y el nombre de los ciclistas cuya edad sea menor o igual que 25 años.

```
SELECT dorsal, nombre, edad FROM ciclista  
WHERE edad <= 25;
```

3. Obtener el nombre y la altura de todos los puertos de categoría 'E' (Especial).

```
SELECT nompuerto, altura FROM puerto WHERE categoria = 'E';
```

4. Obtener el valor del atributo netapa de aquellas etapas con salida y llegada en la misma ciudad.

```
SELECT netapa FROM etapa WHERE salida = llegada;
```

5. ¿Cuántos ciclistas hay?

```
SELECT COUNT(*) FROM ciclista;
```

6. ¿Cuántos ciclistas hay con edad superior a 25 años?

```
SELECT COUNT(*) FROM ciclista WHERE edad > 25;
```

7. ¿Cuántos equipos hay?

```
SELECT COUNT(*) FROM equipo;
```

8. Obtener la media de edad de los ciclistas.

```
SELECT AVG(edad) FROM ciclista;
```

9. Obtener la altura mínima y máxima de los puertos de montaña.

```
SELECT MAX(altura), MIN(altura) FROM puerto;
```

10. Obtener la altura máxima de los puertos por categoría.

```
SELECT MAX(altura) FROM puerto GROUP BY categoria;
```

11. Obtener la media de edad de ciclistas para cada equipo.

```
SELECT nomeq, avg(edad) FROM ciclista GROUP BY nomeq;
```

12. Obtener los equipos cuya media de edad es mayor de 30.

```
SELECT AVG(edad) AS media FROM ciclista  
GROUP BY nomeq  
having media > 30;
```

Consultas sobre varias tablas

13. Obtener el nombre y la categoría de los puertos ganados por ciclistas del equipo 'Banesto'.

```
SELECT p.nompuerto, p.categoria FROM puerto p, ciclista c  
WHERE p.dorsal = c.dorsal AND c.nomeq = 'Banesto';
```

14. Obtener el nombre de cada puerto indicando el número (netapa) y los kilómetros de la etapa en la que se encuentra el puerto.

```
SELECT p.nompuerto, e.netapa, e.km FROM puerto p, etapa e  
WHERE p.netapa = e.netapa;
```

15. Obtener el nombre y el director de los equipos a los que pertenezca algún ciclista mayor de 33 años.

```
SELECT eq.nomeq, eq.descripcion FROM equipo eq, ciclista c  
WHERE eq.nomeq = c.nomeq  
GROUP BY c.nomeq HAVING c.edad > 33;
```

16. Obtener el nombre de los ciclistas con el color de cada maillot que hayan llevado.

```
SELECT DISTINCT c.nombre, m.color FROM ciclista m, llevar ll, maillot m  
WHERE c.dorsal = ll.dorsal AND ll.codigo = m.codigo.
```

17. Obtener pares de nombre de ciclista y número de etapa tal que ese ciclista haya ganado esa etapa habiendo llevado el maillot de color 'Amarillo' al menos una vez.

```
SELECT DISTINCT c.nombre, e.netapa FROM ciclista c, etapa e, llevar ll, maillot m  
WHERE c.dorsal = e.dorsal AND e.dorsal = ll.dorsal AND ll.codigo = m.codigo  
AND m.color = 'Amarillo';
```

Consultas con subconsultas

18. Obtener el valor del atributo netapa y la ciudad de salida de aquellas etapas que no tengan puertos de montaña.

```
SELECT e.netapa, e.salida FROM etapa e
WHERE e.netapa NOT IN (SELECT netapa FROM puerto);
```

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

```
SELECT AVG(c.edad) FROM ciclista c
WHERE c.dorsal IN (SELECT DISTINCT e.dorsal FROM etapa e);
```

20. Selecciona el nombre de los puertos con una altura superior a la altura media de todos los puertos.

```
SELECT p.nompuerto, p.altura FROM puerto p
WHERE p.altura > (SELECT AVG(p.altura) FROM puerto p);
```

21. Obtener el nombre de la ciudad de salida y de llegada de las etapas donde estén los 3 puertos con mayor pendiente.

```
SELECT ETAPA.SALIDA, ETAPA.LLEGADA FROM ETAPA
WHERE NETAPA IN ( SELECT DISTINCT NETAPA FROM PUERTO
ORDER BY PENDIENTE DESC FETCH FIRST 3 ROWS ONLY);
```

22. Obtener el dorsal y el nombre de los ciclistas que han ganado los 2 puertos de mayor altura.

```
SELECT c.dorsal, c.nombre FROM ciclista c
WHERE c.dorsal IN ( SELECT DISTINCT dorsal FROM PUERTO
ORDER BY altura DESC FETCH FIRST 2 ROWS ONLY);
```

23. Obtener el nombre del ciclista más joven.

```
(En caso de empate mostraría 1 solo)
SELECT c.nombre FROM ciclista c
ORDER BY c.edad ASC
FETCH FIRST 1 ROW ONLY;
(En caso de empate mostraría todos)
SELECT c.nombre FROM ciclista c
WHERE c.edad = (SELECT MIN(c.edad) FROM ciclista c);
```

24. Obtener el nombre del ciclista más joven que ha ganado al menos una etapa.

```
SELECT c.nombre FROM ciclista c
WHERE c.dorsal IN (SELECT e.dorsal FROM etapa e)
ORDER BY c.edad ASC
FETCH FIRST 1 ROW ONLY;
```

25. Obtener el nombre de los ciclistas que han ganado más de un puerto.

```
SELECT c.dorsal, c.nombre FROM ciclista c
WHERE c.dorsal IN (SELECT p.dorsal FROM puerto p
GROUP BY p.dorsal
HAVING COUNT(p.dorsal) > 1);
```

Consultas con cuantificación universal (CON NOT EXISTS O NOT IN)

26. Obtener el valor del atributo netapa de aquellas etapas tales que todos los puertos que están en ellas tienen más de 700 metros de altura.

```
SELECT DISTINCT p.netapa FROM puerto p
WHERE p.netapa NOT IN (SELECT p.netapa FROM puerto p
GROUP BY p.netapa
HAVING MAX(p.altura) < 700);
```

27. Obtener el nombre y el director de los equipos tales que todos sus ciclistas son mayores de 26 años.

```
SELECT e.nomeq, e.descripcion FROM equipo e
WHERE e.nomeq IN (SELECT c.nomeq FROM ciclista c
GROUP BY c.nomeq HAVING MIN(c.edad) > 26);
```

28. Obtener el dorsal y el nombre de los ciclistas tales que todas las etapas que han ganado tienen más de 170 km (es decir que sólo han ganado etapas de más de 170 km).

```
SELECT c.dorsal, c.nombre FROM ciclista c
WHERE c.dorsal IN (SELECT dorsal from etapa
WHERE dorsal = c.dorsal
AND km > 220);
```

29. Obtener el nombre de los ciclistas que han ganado todos los puertos de una etapa y además han ganado esa misma etapa.

```
SELECT c.nombre FROM ciclista c
WHERE c.dorsal IN (SELECT dorsal FROM etapa e
WHERE e.dorsal IN (SELECT dorsal FROM puerto p
WHERE p.netapa = e.netapa));
```

30. Obtener el nombre de los equipos tales que todos sus corredores han llevado algún maillot o han ganado algún puerto.

```
SELECT DISTINCT c.nomeq FROM ciclista c
WHERE c.dorsal IN (SELECT ll.dorsal FROM llevar ll)
OR c.dorsal IN (SELECT p.dorsal FROM puerto p);
```

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

```
SELECT DISTINCT m.codigo, m.color FROM ciclista c, llevar ll, maillot m
WHERE c.dorsal = ll.dorsal AND ll.codigo = m.codigo
GROUP BY m.codigo, m.color
HAVING COUNT(DISTINCT c.nomeq) = 1;
```

32. Obtener el nombre de aquellos equipos tal que sus ciclistas sólo hayan ganado puertos de 1ª categoría.

```
SELECT DISTINCT c.nomeq FROM ciclista c, puerto p
WHERE c.dorsal = p.dorsal
AND c.dorsal NOT IN (SELECT dorsal FROM puerto WHERE categoria != '1')
GROUP BY c.nomeq;
```

Consultas agrupadas

33. Obtener el valor del atributo netapa de aquellas etapas que tienen puertos de montaña indicando cuántos tiene.

```
SELECT p.netapa, COUNT(*) FROM puerto p GROUP BY p.netapa;
```

34. Obtener el nombre de los equipos que tengan ciclistas indicando cuántos tiene cada uno.

```
SELECT c.nomeq, COUNT(*) FROM ciclista c GROUP BY c.nomeq;
```

35. Obtener el nombre de todos los equipos indicando cuántos ciclistas tiene cada uno.

```
SELECT c.nomeq, COUNT(*) FROM ciclista c GROUP BY c.nomeq;
```

36. Obtener el director y el nombre de los equipos que tengan más de 3 ciclistas y cuya edad media sea inferior o igual a 30 años.

```
SELECT e.descripcion, e.nomeq FROM equipo e
WHERE e.nomeq IN (SELECT c.nomeq FROM ciclista c
GROUP BY c.nomeq
HAVING COUNT(*) > 3
AND AVG(c.edad) <= 30);
```

37. Obtener el nombre de los ciclistas que pertenezcan a un equipo que tenga más de cinco corredores y que hayan ganado alguna etapa indicando cuántas etapas ha ganado.

```
SELECT c.nombre, count(*) 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.nombre;
```

38. Obtener el nombre de los equipos y la edad media de sus ciclistas de aquellos equipos que tengan la media de edad máxima de todos los equipos.

```
SELECT c.nomeq, AVG(c.edad) edad FROM ciclista c
GROUP BY c.nomeq HAVING AVG(c.edad) =
(SELECT AVG(edad) FROM ciclista GROUP BY ciclista.nomeq
ORDER BY avg(edad) DESC      FETCH FIRST 1 ROWS ONLY);
```

39. Obtener el director de los equipos cuyos ciclistas han llevado más días maillots de cualquier tipo. Nota: cada tupla de la relación llevar indica que un ciclista ha llevado un maillot un día

```
SELECT eq.nomeq, eq.descripcion FROM equipo eq
WHERE eq.nomeq IN (SELECT c.nomeq FROM ciclista c, llevar ll, maillot m
                  WHERE c.dorsal = ll.dorsal AND ll.codigo = m.codigo
                  GROUP BY c.nomeq ORDER BY COUNT(*) DESC
                  FETCH FIRST 1 ROWS ONLY);
```

Consultas generales

40. Obtener el código y el color del maillot que ha sido llevado por algún ciclista que no ha ganado ninguna etapa.

```
SELECT DISTINCT m.codigo, m.color FROM maillot m, llevar ll
WHERE m.codigo = ll.codigo
AND DORSAL NOT IN (SELECT e.dorsal FROM etapa e);
```

41. Obtener el valor del atributo netapa, la ciudad de salida y la ciudad de llegada de las etapas de más de 190 km. y que tengan por lo menos dos puertos.

```
SELECT e.netapa, e.salida, e.llegada FROM etapa e
WHERE km > 190 AND e.netapa IN (SELECT p.netapa FROM puerto p
                                GROUP BY p.netapa
                                HAVING COUNT(*) >= 2);
```

42. Obtener el dorsal y el nombre de los ciclistas que no han llevado todos los maillots que ha llevado el ciclista de dorsal 20.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal AND ll.codigo != ALL
(SELECT codigo FROM LLEVAR WHERE dorsal = 20);
```

43. Obtener el dorsal y el nombre de los ciclistas que han llevado al menos un maillot de los que ha llevado el ciclista de dorsal 20.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal
AND ll.codigo = ANY (SELECT codigo FROM LLEVAR WHERE dorsal = 20);
```

44. Obtener el dorsal y el nombre de los ciclistas que no han llevado ningún maillot de los que ha llevado el ciclista de dorsal 20.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal AND ll.codigo NOT IN
(SELECT DISTINCT codigo FROM LLEVAR WHERE dorsal = 20);
```

45. Obtener el dorsal y el nombre de los ciclistas que han llevado todos los maillots que ha llevado el ciclista de dorsal 20.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal AND ll.codigo = ANY
(SELECT codigo FROM LLEVAR WHERE dorsal = 20);
```

46. Obtener el dorsal y el nombre de los ciclistas que han llevado exactamente los mismos maillots que ha llevado el ciclista de dorsal 20.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal AND ll.codigo = ALL
(SELECT DISTINCT codigo FROM LLEVAR WHERE dorsal = 20);
```

47. Obtener el dorsal y el nombre del ciclista que ha llevado durante más kilómetros un mismo maillot e indicar también el color de dicho maillot.

```
SELECT c.dorsal, c.nombre, m.color FROM ciclista c, llevar ll, maillot m, etapa e
WHERE c.dorsal = ll.dorsal AND ll.codigo = m.codigo AND ll.netapa = e.netapa
GROUP BY c.dorsal, c.nombre, m.color
ORDER BY SUM(e.km) DESC FETCH FIRST 1 ROWS ONLY;
```

48. Obtener el dorsal y el nombre de los ciclistas que han llevado tres tipos de maillot menos de los que ha llevado el ciclista de dorsal 1.

```
SELECT DISTINCT c.dorsal, c.nombre FROM ciclista c, llevar ll
WHERE c.dorsal = ll.dorsal GROUP BY c.dorsal, c.nombre
HAVING COUNT(DISTINCT c.dorsal) + 4 <= (SELECT COUNT(DISTINCT ll.codigo)
FROM llevar ll WHERE ll.dorsal = '1');
```

49. Obtener el valor del atributo netapa y los km de las etapas que tienen puertos de montaña.

```
SELECT e.netapa, e.km FROM etapa e
WHERE e.netapa IN (SELECT p.netapa FROM puerto p);
```