**REALIZA LAS SIGUIENTES CONSULTAS EN SQL:**

**1. OBTENER EL CÓDIGO, EL TIPO, EL COLOR Y EL PREMIO DE TODOS LOS MAILLOTS QUE HAY.**

SELECT \* FROM MAILLOT;

**2. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS CUYA EDAD SEA MENOR O IGUAL QUE 25 AÑOS.**

SELECT DORSAL, NOMBRE FROM CICLISTA WHERE EDAD <= 25;

**3. OBTENER EL NOMBRE Y LA ALTURA DE TODOS LOS PUERTOS DE CATEGORÍA**

**ʻEʼ (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 LLEGADA = SALIDA;

**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) AS ALTURA\_MAXIMA, MIN(ALTURA) AS ALTURA\_MINIMA FROM PUERTO;

**10. OBTENER EL NOMBRE DE CADA CICLISTA JUNTO CON EL NOMBRE DEL EQUIPO AL QUE PERTENECE.**

SELECT NOMBRE, NOMEQ FROM CICLISTA;

**11. OBTENER EL NOMBRE DE LOS CICLISTAS QUE SEAN DE BANESTO.**

SELECT NOMBRE FROM CICLISTA WHERE NOMEQ ='BANESTO';

**12. ¿CUÁNTOS CICLISTAS PERTENECEN AL EQUIPO AMORE VITA?**

SELECT COUNT(\*) FROM CICLISTA WHERE NOMEQ = 'AMORE VITA';

**13. EDAD MEDIA DE LOS CICLISTAS DEL EQUIPO TVM.**

SELECT AVG(EDAD) FROM CICLISTA WHERE NOMEQ='TVM';

**14. NOMBRE DE LOS CICLISTAS QUE PERTENEZCAN AL MISMO EQUIPO QUE MIGUEL**

**INDURAIN.**

SELECT NOMBRE FROM CICLISTA WHERE NOMEQ = (SELECT NOMEQ FROM CICLISTA WHERE NOMBRE = 'MIGUEL INDURAIN');

**15. NOMBRE DE LOS CICLISTAS QUE HAN GANADO ALGUNA ETAPA.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DISTINCT(DORSAL) FROM ETAPA);

SELECT NOMBRE FROM CICLISTA, ETAPA WHERE CICLISTA.DORSAL=ETAPA.DORSAL;

**16. NOMBRE DE LOS CICLISTAS QUE HAN LLEVADO EL MAILLOT GENERAL.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DISTINCT(DORSAL) FROM LLEVAR WHERE CODIGO = (SELECT CODIGO FROM MAILLOT WHERE TIPO = 'GENERAL'));

SELECT NOMBRE FROM CICLISTA, LLEVAR, MAILLOT WHERE CICLISTA.DORSAL = LLEVAR.DORSAL AND LLEVAR.CODIGO = MAILLOT.CODIGO AND MAILLOT.TIPO = 'GENERAL';

**17. OBTENER EL NOMBRE DEL CICLISTA MÁS JOVEN**

SELECT NOMBRE FROM CICLISTA WHERE EDAD = (SELECT MIN(EDAD) FROM CICLISTA);

**18. OBTENER EL NÚMERO DE CICLISTAS DE CADA EQUIPO.**

SELECT NOMEQ, COUNT(\*) FROM CICLISTA GROUP BY NOMEQ;

**19. OBTENER EL NOMBRE DE LOS EQUIPOS QUE TENGAN MÁS DE 5 CICLISTAS.**

SELECT NOMEQ, COUNT(\*) FROM CICLISTA GROUP BY NOMEQ HAVING COUNT(\*)>5;

**20. OBTENER EL NÚMERO DE PUERTOS QUE HA GANADO CADA CICLISTA.**

SELECT CICLISTA.NOMBRE, COUNT(PUERTO.DORSAL) FROM PUERTO,CICLISTA WHERE PUERTO.DORSAL = CICLISTA.DORSAL GROUP BY PUERTO.DORSAL;

**21. OBTENER EL NOMBRE DE LOS CICLISTAS QUE HAN GANADO MÁS DE UN PUERTO.**

SELECT CICLISTA.NOMBRE FROM CICLISTA, PUERTO WHERE PUERTO.DORSAL = CICLISTA.DORSAL GROUP BY PUERTO.DORSAL HAVING COUNT(\*) > 1;

**22. OBTENER EL NOMBRE Y EL DIRECTOR DE LOS EQUIPOS A LOS QUE PERTENEZCA ALGÚN CICLISTA MAYOR DE 33 AÑOS.**

SELECT DISTINCT EQUIPO.NOMEQ, EQUIPO.DIRECTOR FROM EQUIPO, CICLISTA WHERE CICLISTA.NOMEQ = EQUIPO.NOMEQ AND CICLISTA.EDAD > 33;

**23. NOMBRE DE LOS CICLISTAS QUE NO PERTENEZCAN A KELME**

SELECT NOMBRE FROM CICLISTA WHERE NOMEQ <> 'KELME';

**24. NOMBRE DE LOS CICLISTAS QUE NO HAYAN GANADO NINGUNA ETAPA.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL NOT IN (SELECT DORSAL FROM ETAPA);

SELECT NOMBRE FROM CICLISTA LEFT JOIN ETAPA ON CICLISTA.DORSAL=ETAPA.DORSAL WHERE ETAPA.DORSAL IS NULL;

**25. NOMBRE DE LOS CICLISTAS QUE NO HAYAN GANADO NINGÚN PUERTO DE MONTAÑA.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL NOT IN (SELECT DORSAL FROM PUERTO);

**26. NOMBRE DE LOS CICLISTAS QUE HAYAN GANADO MÁS DE UN PUERTO DE MONTAÑA.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM PUERTO GROUP BY DORSAL HAVING COUNT(DORSAL) > 1);

SELECT NOMBRE, COUNT(\*) FROM PUERTO, CICLISTA WHERE CICLISTA.DORSAL=PUERTO.DORSAL GROUP BY PUERTO.DORSAL HAVING COUNT(\*)>1;

**27. ¿QUÉ CICLISTAS HAN LLEVADO EL MISMO MAILLOT QUE MIGUEL INDURAIN?**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL = (SELECT DORSAL FROM CICLISTA WHERE NOMBRE = 'MIGUEL INDURAIN')) AND NOMBRE<>'MIGUEL INDURAIN');

SELECT CICLISTA.NOMBRE FROM CICLISTA, LLEVAR WHERE CICLISTA.DORSAL = LLEVAR.DORSAL AND LLEVAR.CODIGO IN (SELECT LLEVAR.CODIGO FROM LLEVAR WHERE DORSAL = (SELECT DORSAL FROM CICLISTA WHERE NOMBRE = 'MIGUEL INDURAIN')) AND NOMBRE<>'MIGUEL INDURAIN';

SELECT NOMBRE FROM CICLISTA, MAILLOT, LLEVAR WHERE CICLISTA.DORSAL=LLEVAR.DORSAL AND LLEVAR.CODIGO=MAILLOT.CODIGO AND NOMBRE<>’MIGUEL INDURAIN’ AND MAILLOT.CODIGO IN (SELECT MAILLOT.CODIGO FROM CICLISTA, MAILLOT, LLEVAR WHERE CICLISTA.DORSAL=LLEVAR.DORSAL AND LLEVAR.CODIGO=MAILLOT.CODIGO AND NOMBRE='MIGUEL INDURAIN');

**28. DE CADA EQUIPO OBTENER LA EDAD MEDIA, LA MÁXIMA EDAD Y LA MÍNIMA EDAD.**

SELECT NOMEQ, AVG(EDAD) AS MEDIA\_EDAD, MAX(EDAD) AS MAXIMA\_EDAD, MIN(EDAD) AS MINIMA\_EDAD FROM CICLISTA GROUP BY NOMEQ;

**29. NOMBRE DE AQUELLOS CICLISTAS QUE TENGAN UNA EDAD ENTRE 25 Y 30 AÑOS Y QUE NO PERTENEZCAN A LOS EQUIPOS KELME Y BANESTO.**

SELECT NOMBRE FROM CICLISTA WHERE EDAD BETWEEN 25 AND 30 AND NOMEQ NOT IN ('KELME','BANESTO');

**30. NOMBRE DE LOS CICLISTAS QUE HAN GANADO LA ETAPA QUE COMIENZA EN ZAMORA.**

SELECT NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM ETAPA WHERE SALIDA = 'ZAMORA');

SELECT NOMBRE FROM CICLISTA, ETAPA WHERE CICLISTA.DORSAL=ETAPA.DORSAL AND SALIDA='ZAMORA';

**31. OBTÉN EL NOMBRE Y LA CATEGORÍA DE LOS PUERTOS GANADOS POR CICLISTAS DEL EQUIPO ʻBANESTOʼ.**

SELECT NOMPUERTO, CATEGORIA FROM PUERTO WHERE DORSAL IN (SELECT DORSAL FROM CICLISTA WHERE NOMEQ = 'BANESTO');

**32. OBTENER EL NOMBRE DE CADA PUERTO INDICANDO EL NÚMERO (NETAPA) Y LOS**

**KILÓMETROS DE LA ETAPA EN LA QUE SE ENCUENTRA EL PUERTO.**

SELECT PUERTO.NOMPUERTO, PUERTO.NETAPA, ETAPA.KM FROM PUERTO, ETAPA WHERE PUERTO.NETAPA = ETAPA.NETAPA;

**33. OBTENER EL NOMBRE DE LOS CICLISTAS CON EL COLOR DE CADA MAILLOT QUE HAYAN LLEVADO.**

SELECT CICLISTA.NOMBRE, MAILLOT.COLOR FROM CICLISTA, LLEVAR, MAILLOT WHERE CICLISTA.DORSAL = LLEVAR.DORSAL AND LLEVAR.CODIGO = MAILLOT.CODIGO;

**34. 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.** INTERPRETABLE

¡ESTO SERÍA QUE HA LLEVADO EL AMARILLO ALGUNA VEZ AUNQUE NO SEA EN LA ETAPA QUE HA GANADO!

SELECT CICLISTA.NOMBRE, ETAPA.NETAPA FROM CICLISTA, ETAPA WHERE ETAPA.DORSAL = CICLISTA .DORSAL AND ETAPA.DORSAL IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO = ( SELECT CODIGO FROM MAILLOT WHERE COLOR = 'AMARILLO'));

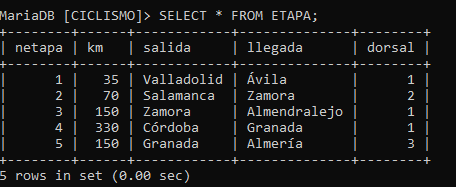
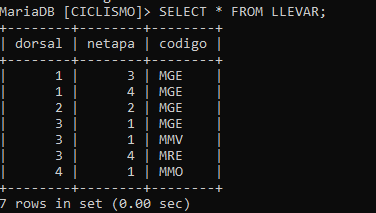
Para mi esta no es válida

¡ESTO SERÍA QUE HA LLEVADO EL AMARILLO EN LA ETAPA QUE DEVUELVE PERO NO QUIERE DECIR QUE ESA ETAPA LA HAYA GANADO!

SELECT CICLISTA.NOMBRE, LLEVAR.NETAPA FROM CICLISTA, MAILLOT, LLEVAR

WHERE CICLISTA.DORSAL=LLEVAR.DORSAL AND MAILLOT.CODIGO=LLEVAR.CODIGO AND COLOR='AMARILLO';

**LO MÁS CORRECTO PARA MI SERÍA COMPROBAR EN CADA ETAPA QUE EL CICLISTA QUE LA HA GANADO HA LLEVADO EL AMARILLO(EN ESA ETAPA)**



|  |  |  |
| --- | --- | --- |
| Etapa ganada | dorsal | ¿LLEVABA AMARILLO ESE DORSAL EN ESA ETAPA? |
| 1 | 1 | No |
| **3** | **1** | 1-3 ->SI |
| **4** | **1** | 1-4 ->SI |
| **2** | **2** | 2-2 ->SI |
| 5 | 3 | 4-1->NO |

Serían el dorsal 1 que ha ganado y ha llevado el amarillo en las etapa 3 y 4 y el dorsal 2 en la etapa2

**SELECT CICLISTA.NOMBRE, ETAPA.NETAPA FROM CICLISTA, ETAPA WHERE ETAPA.DORSAL = CICLISTA .DORSAL AND ETAPA.DORSAL IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO = ( SELECT CODIGO FROM MAILLOT WHERE COLOR = 'AMARILLO') AND ETAPA.NETAPA=LLEVAR.NETAPA);**

**35. OBTENER EL VALOR DEL ATRIBUTO NETAPA DE LAS ETAPAS QUE NO COMIENZAN EN LA MISMA CIUDAD EN QUE ACABÓ LA ANTERIOR ETAPA.**

SELECT E1.NETAPA, E1.LLEGADA, E2.NETAPA, E2.SALIDA FROM ETAPA E1, ETAPA E2 WHERE E1.NETAPA+1=E2.NETAPA AND E1.LLEGADA <>E2.SALIDA;

SELECT E2.NETAPA, E2.SALIDA FROM ETAPA E1, ETAPA E2 WHERE E1.NETAPA+1=E2.NETAPA AND E1.LLEGADA <>E2.SALIDA;

**36. OBTENER EL VALOR DEL ATRIBUTO NETAPA Y LA CIUDAD DE SALIDA DE AQUELLAS ETAPAS QUE NO TENGAN PUERTOS DE MONTAÑA.**

SELECT NETAPA, SALIDA FROM ETAPA WHERE NETAPA NOT IN (SELECT NETAPA FROM PUERTO);

**37. OBTENER LA EDAD MEDIA DE LOS CICLISTAS QUE HAN GANADO ALGUNA ETAPA.**

SELECT AVG(EDAD) FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM ETAPA);

**38. SELECCIONA EL NOMBRE DE LOS PUERTOS CON UNA ALTURA SUPERIOR A LA ALTURA MEDIA DE TODOS LOS PUERTOS.**

SELECT NOMPUERTO FROM PUERTO WHERE ALTURA > (SELECT AVG(ALTURA) FROM PUERTO);

**39. OBTENER EL NOMBRE DE LA CIUDAD DE SALIDA Y DE LLEGADA DE LAS ETAPAS DONDE ESTÉN LOS PUERTOS CON MAYOR PENDIENTE.**

SELECT SALIDA, LLEGADA FROM ETAPA WHERE NETAPA IN (SELECT NETAPA FROM PUERTO WHERE PENDIENTE = (SELECT MAX(PENDIENTE) FROM PUERTO);

**40. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS QUE HAN GANADO LOS PUERTOS DE MAYOR ALTURA.**

SELECT CICLISTA.NOMBRE FROM PUERTO, CICLISTA WHERE PUERTO.DORSAL = CICLISTA.DORSAL AND PUERTO.ALTURA = (SELECT MAX(ALTURA) FROM PUERTO);

**41. OBTENER EL NOMBRE DEL CICLISTA MÁS JOVEN QUE HA GANADO AL MENOS UNA ETAPA.**

SELECT NOMBRE FROM CICLISTA, ETAPA WHERE CICLISTA.DORSAL=ETAPA.DORSAL AND CICLISTA.EDAD = (SELECT MIN(EDAD) FROM CICLISTA, ETAPA WHERE CICLISTA.DORSAL=ETAPA.DORSAL);

SELECT NOMBRE FROM CICLISTA NATURAL JOIN ETAPA WHERE EDAD= (SELECT MIN(EDAD) FROM CICLISTA NATURAL JOIN ETAPA);

**42. 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 NETAPA FROM PUERTO GROUP BY NETAPA HAVING MIN(ALTURA) > 700;

**NOTA**: TODOS LOS PUERTOS TIENEN UNA ALTURA SUPERIOR A 700 METROS POR LO QUE ES ESTA CONSULTA SIEMPRE DARÁ COMO RESULTADO TODAS LAS ESTAPAS CON AL MENOS UN PUERTO DE MONTAÑA EN SU ITINERARIO. LA CONSULTA SE PUEDA COMPROBAR FÁCILMENTE CAMBIANDO LA ALTURA SEGÚN LOS REGISTROS PRESENTES EN LA TABLAS PUERTO.

**43. OBTENER EL NOMBRE Y EL DIRECTOR DE LOS EQUIPOS TALES QUE TODOS SUS CICLISTAS SON MAYORES DE 20 AÑOS.**

SELECT NOMEQ, DIRECTOR FROM EQUIPO WHERE NOMEQ NOT IN (SELECT DISTINCT NOMEQ FROM CICLISTA WHERE EDAD<=20);

SELECT EQUIPO.NOMEQ, DIRECTOR FROM CICLISTA, EQUIPO WHERE CICLISTA.NOMEQ = EQUIPO.NOMEQ GROUP BY EQUIPO.NOMEQ HAVING MIN(EDAD) > 20;

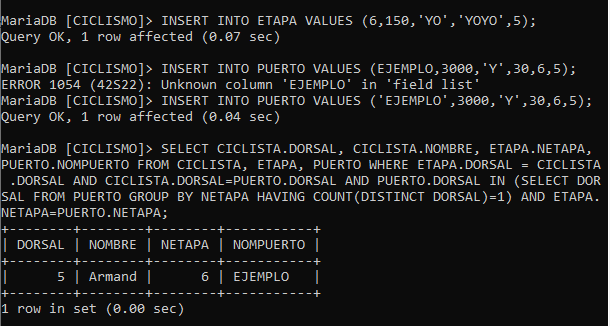
**44. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS TALES QUE TODAS LAS ETAPAS QUE HAN GANADO TIENEN MÁS DE 170 KM (ES DECIR QUE SÓLO HAN GANADO ETAPAS DE MÁS DE 170 KM).**

SELECT CICLISTA.DORSAL, CICLISTA.NOMBRE FROM CICLISTA, ETAPA WHERE ETAPA.DORSAL = CICLISTA.DORSAL GROUP BY ETAPA.DORSAL HAVING MIN(KM) > 170;

**45. OBTENER EL NOMBRE DE LOS CICLISTAS QUE HAN GANADO TODOS LOS PUERTOS DE UNA ETAPA Y ADEMÁS HAN GANADO ESA MISMA ETAPA. (NO HAY NINGUNO)**

**SELECT CICLISTA.NOMBRE, ETAPA.NETAPA, PUERTO.NOMPUERTO FROM CICLISTA, ETAPA, PUERTO WHERE ETAPA.DORSAL = CICLISTA .DORSAL AND CICLISTA.DORSAL=PUERTO.DORSAL AND PUERTO.DORSAL IN (SELECT DORSAL FROM PUERTO GROUP BY NETAPA HAVING COUNT(DISTINCT DORSAL)=1) AND ETAPA.DORSAL=PUERTO.DORSAL;**

Nota: voy a insertar una nueva etapa y un nuevo puerto para esa etapa con el mismo dorsal ganador para ver si la consulta es correcta:



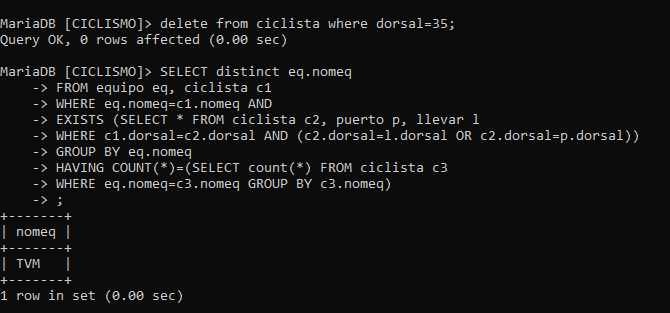
**Esta es la que os dije que no acabo de ver. La solución está bien, cuando tenga un rato la pensaré bien y os la explicaré en clase.**

**46. OBTENER EL NOMBRE DE LOS EQUIPOS TALES QUE TODOS SUS CORREDORES HAN**

**LLEVADO ALGÚN MAILLOT O HAN GANADO ALGÚN PUERTO. (**No hay ninguno)

SELECT distinct eq.nomeq FROM equipo eq, ciclista c1  
WHERE eq.nomeq=c1.nomeq AND EXISTS (SELECT \* FROM ciclista c2, puerto p, llevar l WHERE c1.dorsal=c2.dorsal AND (c2.dorsal=l.dorsal OR c2.dorsal=p.dorsal)) GROUP BY eq.nomeq HAVING COUNT(\*)=(SELECT count(\*) FROM ciclista c3 WHERE eq.nomeq=c3.nomeq GROUP BY c3.nomeq);

Para comprobar que funciona me cargo el 35 y luego lo añadiré otra vez.



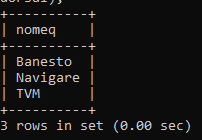
Explicación:

1. Cuento los ciclistas que hay en cada equipo.

SELECT nomeq, count(\*) FROM ciclista group by nomeq

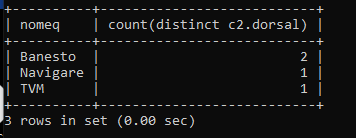
1. Por otro lado le digo que una las tablas ciclista, puerto y llevar para poner la condición de que el dorsal aparezca en las tablas llevar o puerto (es decir que haya llevado un maillot o haya ganado un puerto)

SELECT distinct nomeq FROM ciclista c2, puerto p, llevar l WHERE (c2.dorsal=l.dorsal OR c2.dorsal=p.dorsal)



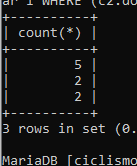
1. Ahora he de contar los ciclistas que hay en cada equipo de los obtenidos en la consulta2 para ello agrupo por equipo y añado el count(distinct dorsal)

SELECT nomeq, count(distinct c2.dorsal) FROM ciclista c2, puerto p, llevar l WHERE (c2.dorsal=l.dorsal OR c2.dorsal=p.dorsal) group by nomeq;



1. Cuento los ciclistas que hay en esos equipos

Select count(\*) from ciclista where nomeq in (SELECT distinct nomeq FROM ciclista c3, puerto p, llevar l WHERE (c3.dorsal=l.dorsal OR c3.dorsal=p.dorsal))group by nomeq;



1. Ahora INTENTO UNIR TODO (pensarlo…estoy agotada)

SELECT distinct eq.nomeq FROM equipo eq, ciclista c1  
WHERE eq.nomeq=c1.nomeq AND EXISTS (SELECT \* FROM ciclista c2, puerto p, llevar l WHERE c1.dorsal=c2.dorsal AND (c2.dorsal=l.dorsal OR c2.dorsal=p.dorsal)) GROUP BY eq.nomeq HAVING COUNT(\*)=(SELECT count(\*) FROM ciclista c3 WHERE eq.nomeq=c3.nomeq GROUP BY c3.nomeq);

**46. Más fácil así:**

**Ciclistas que han llevado un dorsal o han ganado un puerto**

select distinct c2.dorsal, c2.nomeq from ciclista c2, puerto p, llevar l where (c2.dorsal=l.dorsal or c2.dorsal=p.dorsal);

**Cuento cuantos ciclistas de cada equipo han llevado un dorsal o han ganado un puerto**

select count(distinct c2.dorsal) as conteo, c2.nomeq from ciclista c2, puerto p, llevar l where (c2.dorsal=l.dorsal or c2.dorsal=p.dorsal) group by c2.nomeq;

**Cuento los ciclistas que hay en cada equipo.**

SELECT nomeq, count(\*) as conteo general FROM ciclista group by nomeq

Tengo que seleccionar los equipos de esta última consulta que estén en la anterior y cuyo conteo coincida

En el where pongo la condición de que los equipos sean los que he sacado antes. Asi solo tengo Banesto, navigare y Tvm

Ahora agrupo por equipos y cuento

Y en el having pongo la condición de que el conteo sea igual al conteo anterior pero tengo que añadir lo que está en amarillo para que me compare en las dos consultas el mismo equipo

SELECT c1.nomeq, count(\*) as conteogeneral FROM ciclista c1

where c1.nomeq in (select c2.nomeq from ciclista c2, puerto p, llevar l where (c2.dorsal=l.dorsal or c2.dorsal=p.dorsal))

group by c1.nomeq

having conteogeneral=(select count(distinct c3.dorsal) from ciclista c3, puerto p, llevar l where c1.nomeq=c3.nomeq and (c3.dorsal=l.dorsal or c3.dorsal=p.dorsal) group by c3.nomeq)

**47. OBTENER EL CÓDIGO Y EL COLOR DE AQUELLOS MAILLOTS QUE SÓLO HAN SIDO LLEVADOS POR CICLISTAS DE UN MISMO EQUIPO.**

select codigo,color,nomeq from ciclista natural join llevar natural join maillot group by codigo having count(distinct nomeq)<=1;

**48. OBTENER EL NOMBRE DE AQUELLOS EQUIPOS TAL QUE SUS CICLISTAS SÓLO HAYAN GANADO PUERTOS DE 1ª CATEGORÍA. (explicada en clase----2 partes)**

Equipos con Ciclistas que han ganado puertos de categoría 1

Select distinct equipo.nomeq

From equipo, ciclista, puerto

Where equipo.nomeq=ciclista.nomeq and ciclista.dorsal=puerto.dorsal

And puerto.categoria=ʼ1ʼ

And not exists (select \*

Y que no tengan otros ciclistas que hayan ganado puertos que no sean de categoria 1

From puerto p, ciclista c

Where c.nomeq=ciclista.nomeq and c.dorsal <>ciclista.dorsal

and c.dorsal=p.dorsal and

p.categoria<>ʼ1ʼ);

**49. OBTENER EL VALOR DEL ATRIBUTO NETAPA DE AQUELLAS ETAPAS QUE TIENEN PUERTOS DE MONTAÑA INDICANDO CUÁNTOS TIENE.**

SELECT NETAPA, COUNT(\*) AS NUM\_PUERTOS FROM PUERTO GROUP BY NETAPA;

**50. OBTENER EL NOMBRE DE TODOS LOS EQUIPOS INDICANDO CUÁNTOS CICLISTAS TIENE CADA UNO.**

SELECT NOMEQ, COUNT(\*) AS NUM\_CICLISTAS FROM CICLISTA GROUP BY NOMEQ;

**51. OBTENER EL DIRECTOR Y EL NOMBRE DE LOS EQUIPOS QUE TENGAN MÁS DE 3**

**CICLISTAS Y CUYA EDAD MEDIA SEA IGUAL O INFERIOR A 30 AÑOS.**

SELECT EQUIPO.NOMEQ,DIRECTOR FROM EQUIPO,CICLISTA

WHERE EQUIPO.NOMEQ=CICLISTA.NOMEQ

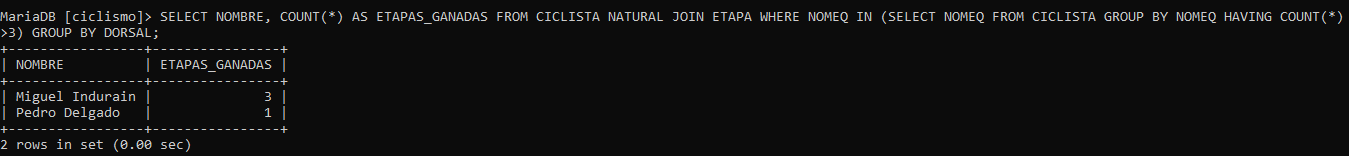
GROUP BY EQUIPO.NOMEQ

HAVING COUNT(\*)>3 AND AVG(EDAD)>30;

SELECT EQUIPO.NOMEQ, DIRECTOR FROM CICLISTA INNER JOIN EQUIPO ON CICLISTA.NOMEQ = EQUIPO.NOMEQ GROUP BY EQUIPO.NOMEQ HAVING COUNT(CICLISTA.DORSAL) >= 3 AND AVG(EDAD) <= 30;

**52. 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.** (NO HAY, LO PRUEBO CON MÁS DE 3 CICLISTAS PARA VER QUE ES CORRECTA)

SELECT NOMBRE, COUNT(\*) AS ETAPAS\_GANADAS FROM CICLISTA NATURAL JOIN ETAPA WHERE NOMEQ IN (SELECT NOMEQ FROM CICLISTA GROUP BY NOMEQ HAVING COUNT(\*)>5) GROUP BY DORSAL;



**53. OBTENER EL NOMBRE DE LOS EQUIPOS Y LA EDAD MEDIA DE SUS CICLISTAS DE**

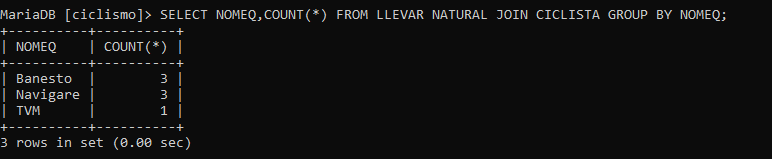
**AQUELLOS EQUIPOS QUE TENGAN LA MEDIA DE EDAD MÁXIMA DE TODOS LOS EQUIPOS.**

SELECT NOMEQ, AVG(EDAD) FROM CICLISTA GROUP BY NOMEQ HAVING AVG(EDAD) >= ALL(SELECT AVG(EDAD) FROM CICLISTA GROUP BY NOMEQ);

SELECT NOMEQ, AVG(EDAD) AS MEDIA FROM CICLISTA GROUP BY NOMEQ HAVING AVG(EDAD)=(SELECT MAX(MEDIA) FROM (SELECT AVG(EDAD) AS MEDIA FROM CICLISTA GROUP BY NOMEQ) AS TABLA1);

**54. OBTENER EL DIRECTOR DE LOS EQUIPOS CUYOS CICLISTAS HAN LLEVADO MÁS DÍAS**

**MAILLOTS DE CUALQUIER TIPO.** (INTERPRETO DIAS=VECES QUE LO LLEVAN)



NUMERO DE VECES QUE HA LLEVADO CADA EQUIPO UN MAILLOT DEL TIPO QUE SEA.

SELECT NOMEQ, COUNT(\*) AS DIAS\_MAILLOT, DIRECTOR FROM LLEVAR NATURAL JOIN CICLISTA NATURAL JOIN EQUIPO GROUP BY NOMEQ HAVING DIAS\_MAILLOT=( SELECT MAX(DIA) FROM (SELECT COUNT(\*) AS DIA FROM LLEVAR NATURAL JOIN CICLISTA GROUP BY NOMEQ )AS TABLE1);

**55. OBTENER EL CÓDIGO Y EL COLOR DEL MAILLOT QUE HA SIDO LLEVADO POR ALGÚN CICLISTA QUE NO HA GANADO NINGUNA ETAPA**

SELECT CODIGO, COLOR FROM MAILLOT WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL NOT IN (SELECT DORSAL FROM ETAPA));

**56. 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 NETAPA, SALIDA, LLEGADA FROM ETAPA WHERE KM>190 AND NETAPA IN (SELECT NETAPA FROM PUERTO GROUP BY NETAPA HAVING COUNT(\*)>=2);

**57. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS QUE NO HAN LLEVADO TODOS LOS MAILLOTS QUE HA LLEVADO EL CICLISTA DE DORSAL 2.**

**(incluidos los dorsales que nunca han llevado un maillot)**

SELECT C.DORSAL, NOMBRE FROM CICLISTA C WHERE EXISTS (SELECT CODIGO FROM LLEVAR WHERE DORSAL=2 AND CODIGO NOT IN (SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=C.DORSAL));

**SELECT NOMBRE, DORSAL FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM (SELECT DORSAL, CODIGO FROM LLEVAR GROUP BY DORSAL HAVING CODIGO NOT IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL=2)) AS RESULTADO) OR DORSAL NOT IN (SELECT DORSAL FROM LLEVAR);**

**58. 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 2.**

SELECT DORSAL, NOMBRE FROM CICLISTA WHERE DORSAL IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL=2)) AND DORSAL<>2;

**59. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS QUE NO HAN LLEVADO NINGÚN MAILLOT DE LOS QUE HA LLEVADO EL CICLISTA DE DORSAL 2.**

SELECT DORSAL, NOMBRE FROM CICLISTA WHERE DORSAL NOT IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL=2));

**60. OBTENER EL DORSAL Y NOMBRE DE LOS CICLISTAS QUE HAN LLEVADO EXACTAMENTE LOS MISMOS MAILLOTS QUE HA LLEVADO EL CICLISTA DE DORSAL 1.**

SELECT C.DORSAL, NOMBRE FROM CICLISTA C WHERE NOT EXISTS (SELECT CODIGO FROM LLEVAR WHERE DORSAL=1 AND CODIGO NOT IN (SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=C.DORSAL)) AND NOT EXISTS (SELECT CODIGO FROM LLEVAR WHERE DORSAL=C.DORSAL AND CODIGO NOT IN (SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=1));

Otra forma sin el exist explicada:

* Veo que maillot ha llevado el 1

SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=1;

* Veo que maillot no ha llevado el 1

SELECT CODIGO FROM LLEVAR L WHERE codigo not in (SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=1);

* Veo que dorsales han llevado esos códigos que no ha llevado el 1

SELECT DORSAL FROM LLEVAR WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR L WHERE codigo not in (SELECT CODIGO FROM LLEVAR L WHERE L.DORSAL=1));

* Veo que dorsal ha llevado los mismos dorsales que el 1 y no ha llevado ninguno de los otros

SELECT C.DORSAL, NOMBRE FROM CICLISTA C NATURAL JOIN LLEVAR L WHERE L.CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE DORSAL=1) AND L.DORSAL NOT IN (SELECT DORSAL FROM LLEVAR WHERE CODIGO IN (SELECT CODIGO FROM LLEVAR WHERE CODIGO NOT IN(SELECT CODIGO FROM LLEVAR WHERE DORSAL=1))) AND DORSAL<>1;

**61. 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 ll.dorsal, ll.codigo, m.color, sum(km) as kms from etapa e,llevar ll, maillot m where e.netapa=ll.netapa and ll.codigo=m.codigo group by ll.dorsal, ll.codigo having sum(km)=(select max(kms) from (select ll.dorsal, ll.codigo, m.color, sum(km) as kms from etapa e,llevar ll, maillot m where e.netapa=ll.netapa and ll.codigo=m.codigo group by ll.dorsal, ll.codigo) as tabla1);

**ESTO ES LO MISMO PERO EN DOS PASOS**

También se puede hacer creando la tabla intermedia (Tabla1) en una tabla virtual llamada **vista que nos muestre los km que se han hecho por maillot**

create view kmspormaillot

as

select l.dorsal, l.codigo, sum(km) as kms

from etapa e, llevar l, maillot m

where e.netapa=l.netapa

and m.codigo=l.codigo

group by l.dorsal, l.codigo;

Select l.dorsal, l.codigo, m.color, sum(km)

from etapa e, llevar l, maillot m

where e.netapa =l.netapa and m.codigo=l.codigo

group by l.dorsal, l.codigo

having sum(km)=(select max(kms) from kmspormaillot);

**62. OBTENER EL DORSAL Y EL NOMBRE DE LOS CICLISTAS QUE HAN LLEVADO DOS TIPOS DE MAILLOT MENOS DE LOS QUE HA LLEVADO EL CICLISTA DE DORSAL 3.**

select dorsal, nombre from ciclista where dorsal in (select dorsal from llevar group by dorsal having count(\*)=(select count(codigo) from llevar where dorsal=3)-2);

select dorsal, nombre from ciclista where dorsal in (select dorsal from llevar group by dorsal having count(\*) + 2 =(select count(distinct codigo)from llevar where dorsal=3));

**63. OBTENER EL VALOR DEL ATRIBUTO NETAPA Y LOS KM DE LAS ETAPAS QUE TIENEN**

**PUERTOS DE MONTAÑA.**

Select netapa, km from etapa where netapa in (select netapa from puerto);

select e.netapa, km from etapa e where exists (select p.netapa from puerto p where p.netapa=e.netapa);