1.

SELECT

codigo AS COD\_TIPO, color, premio

FROM

Maillot

2.

SELECT

dorsal, nombre

FROM

ciclista

WHERE

edad <= 25

3.

SELECT

nompuerto AS NOMBRE, altura

FROM

puerto

WHERE

categoria = 'E'

4.

SELECT

netapa

FROM

etapa

WHERE

salida=llegada

5.

SELECT

count(\*)

FROM

Ciclista

6.

SELECT

count(\*)

FROM

ciclista

where

edad>25

7.

SELECT

count(\*)

FROM

Equipo

8.

SELECT

AVG(edad)

FROM

Ciclista

9.

SELECT

MAX(altura), MIN(altura)

FROM

Puerto

10.

SELECT

nompuerto, categoria

FROM

puerto P,

ciclista C

WHERE

c.dorsal = p.dorsal && nomeq = 'Banesto'

order by nompuerto

11.

SELECT

nompuerto, e.netapa, km

FROM

puerto P,

etapa E

WHERE

e.netapa = p.netapa

ORDER BY nompuerto

**12.**

SELECT

e.nomeq, director

FROM

equipo E,

ciclista C

WHERE

e.nomeq = c.nomeq && edad > 33

GROUP BY e.nomeq , director //porque sino cogerá a dos ciclistas o más de cada equipo que tengo 33 años entonces por ejemplo en banesto y gatorado hay dos ciclistas que tienen mas de 33 años por lo que se repite dos veces, entonces hay que poner el group by e.nomeq, director con tal de que solo sala ALGÚN ciclista no todos ellos

SELECT

e.nomeq, director

FROM

equipo E,

ciclista C

WHERE **nomeq IN( select nomeq from ciclista where edad >33) = EXISTS (Select dorsal from ciclista where edad>33 AND c.equipo=e.equipo) quitamos la condicion del IN y la metemos en el exists**

**Si no hay funcion de agregado puedes utilizar GROUP BY or DISTINCT**

13.

SELECT

c.nombre, m.color

FROM

maillot M,

ciclista C,

llevar L

WHERE

l.dorsal = c.dorsal

&& l.codigo = m.codigo

GROUP BY c.nombre , m.color

order by c.nombre

14.

SELECT

c.nombre, e.netapa

FROM

maillot M,

ciclista C,

llevar L,

etapa E

WHERE

l.dorsal = c.dorsal

&& l.codigo = m.codigo

&& color = 'Amarillo'

&& c.dorsal = e.dorsal //Este lo ponemos para saber quien si coincide el dorsal ganador de la etapa con el dorsal del ciclista

GROUP BY c.nombre , e.netapa

15.

SELECT

netapa

FROM

etapa E

WHERE

salida NOT LIKE (select llegada from etapa where netapa= e.netapa-1) //CHAT GPT

**16. Solución vista**

select netapa, salida

from etapa

where netapa not in (select netapa from puerto) **= not exists (select 1 from puerto p where p.netapa=n.etapa)**

**con not in SELECT**

**netapa, salida**

**FROM**

**etapa e1**

**WHERE**

**e1.netapa NOT IN (SELECT**

**p.netapa**

**FROM**

**puerto p)**

17. No coincide con la solución

SELECT

avg(edad)

FROM

ciclista C, etapa E

where

c.dorsal=e.dorsal

SELECT count(\*) AS ciclistas, AVG(edad) AS edad\_media VERSION BUENA

FROM ciclista

WHERE dorsal IN (select dorsal from etapa) // Tenemos que poner esto con tal de que nos coja aquellos dorsales que esten en etapa SIN REPETIR CICLISTAS

18.

SELECT

nompuerto

FROM

puerto

WHERE

(SELECT

AVG(altura)

FROM

puerto) < altura

19. Solución vista

SELECT

salida, llegada

FROM

etapa E,

puerto P

WHERE

e.netapa = p.netapa

&& p.pendiente = (SELECT

MAX(pendiente)

FROM

puerto)

20.

SELECT

p.dorsal,c.nombre

FROM

etapa E,

puerto P, ciclista C

WHERE

e.netapa = p.netapa && c.dorsal=p.dorsal

&& p.altura= (SELECT

MAX(altura)

FROM

puerto)

21. Solución vista mas o menos

SELECT

nombre

FROM

ciclista

where edad=

(select min(edad) from ciclista)

22.

SELECT

nombre

FROM

ciclista C1, etapa E1

WHERE

edad = (SELECT

MIN(edad)

FROM

ciclista C2,

etapa E2

WHERE

e2.dorsal = c2.dorsal) && e1.dorsal = c1.dorsal

**23. Solución vista**

SELECT

nombre

FROM

ciclista C

WHERE

1 < (SELECT

COUNT(\*)

FROM

puerto P

WHERE

p.dorsal = c.dorsal)

**24. Solución vista**

**SELECT e.netapa**

**FROM etapa e**

**WHERE e.netapa IN**

**(SELECT netapa FROM puerto)**

**AND NOT EXISTS**

**(SELECT \* FROM puerto p WHERE p.altura <=700 AND e.netapa =p.netapa);**

25. Fijandome en el anterior ej

SELECT

e.nomeq, director

FROM

equipo E

WHERE

e.nomeq In (select nomeq from ciclista C)

AND NOT exists (select \* from ciclista C where edad <26 AND e.nomeq =c.nomeq )

26.

SELECT

dorsal, nombre

FROM

ciclista C

WHERE

dorsal IN (SELECT

dorsal

FROM

etapa E)

AND NOT EXISTS( SELECT

\*

FROM

etapa E

WHERE

km < 171 AND e.dorsal = c.dorsal)

27 Solucion vista

SELECT DISTINCT

nombre

FROM

ciclista C,

etapa E,

puerto P

WHERE

e.dorsal = c.dorsal

AND p.netapa = e.netapa

AND NOT EXISTS( SELECT

\*

FROM

puerto p2

WHERE

p2.netapa = e.netapa

AND c.dorsal <> p.dorsal);

!CICLISMO #28. Obtener el nombre de los equipos tales que todos sus corredores han llevado algún maillot o han ganado algún puerto. Equipo | Para todo ciclista perteneciente a Equipo -> (Ciclista llevar maillot) o (Ciclista gana puerto) !============================

SELECT e.nomeq FROM equipo e WHERE NOT EXISTS ( SELECT \* FROM ciclista c WHERE c.nomeq=e.nomeq AND NOT EXISTS (SELECT \* FROM llevar ll WHERE c.dorsal=ll.dorsal ) AND NOT EXISTS (SELECT \* FROM puerto p WHERE p.dorsal=c.dorsal ) ) AND EXISTS ( SELECT \* FROM ciclista c WHERE c.nomeq=e.nomeq);

!CICLISMO #29. 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 maillot m, llevar l, ciclista c WHERE c.dorsal=l.dorsal AND m.codigo=l.codigo AND NOT EXISTS (SELECT \* FROM llevar l2, ciclista c2 WHERE c2.dorsal=l2.dorsal AND c2.nomeq<>c.nomeq AND l2.codigo=l.codigo);

30. Casi lo tengo bien pero me sale un equipo demás

select e.nomeq

from equipo E, ciclista C, puerto P

where categoria IN (select categoria

from puerto

where categoria=1) AND p.dorsal= c.dorsal AND c.nomeq=e.nomeq

group by e.nomeq

**31.**

**select e.netapa, count(nompuerto)**

**from etapa E, puerto P**

**where p.netapa=e.netapa**

**group by e.netapa**

32.

select e.nomeq, count(c.dorsal)

from equipo E left join ciclista C

on e.nomeq=c.nomeq

group by e.nomeq

**33.**

**select e.nomeq, count(c.dorsal)**

**from equipo E LEFT JOIN ciclista C on c.nomeq=e.nomeq**

**group by e.nomeq**

**Pongo LEFT JOIN YA QUE QUIERO MANTENER EL TOTAL DE LOS EQUIPOS A SABIENDAS DE QUE HAY UN EQUIPO CON 0 CICLISTAS**

**SALE UNION TMBN**

**S a.nombre, e.cod**

**F alumnos A, examen E**

**W a.dni=e.dni=**

**UNION**

**S AQUI VA LO MISMO , pero hay que poner un null para igualar campos NULL**

**F alumno**

**W not exists (select 1 from examen where a.dni=e.dni)**

34. Tuve que mirar lo del having count … no entiendo porque se agrupa por una cosa y luego se condiciona por otra, se puede hacer con subqueries?

select e.nomeq, director

from equipo E, ciclista C

where e.nomeq=c.nomeq

group by e.nomeq

having count(c.dorsal)>3 AND avg(edad)<=30

36. David solution

SELECT

(SELECT

AVG(edad) AS media

FROM

ciclista C,

equipo E

WHERE

c.nomeq = e.nomeq

GROUP BY c.nomeq

ORDER BY 1 DESC

LIMIT 1) AS media,

e1.nomeq

FROM

equipo E1,

ciclista C1

WHERE

e1.nomeq = c1.nomeq

GROUP BY c1.nomeq

HAVING AVG(c1.edad) = (SELECT

AVG(edad)

FROM

ciclista C

GROUP BY c.nomeq

ORDER BY AVG(edad) DESC

LIMIT 1)

SELECT

MAX**(media) (cogemos el alias del AVG(EDAD) sino tendríamos que poner MAX(AVG(EDAD))**

FROM

(SELECT

AVG(edad) as **media**, e.nomeq

FROM

ciclista C, equipo E

WHERE

c.nomeq = e.nomeq

GROUP BY e.nomeq

ORDER BY 1 DESC) **t (en el from no se pone AS)**

37. Solucion vista

SELECT

e.director

FROM

llevar L,

equipo E,

ciclista C

WHERE

l.dorsal = c.dorsal

AND e.nomeq = c.nomeq

GROUP BY e.director , e.nomeq

HAVING COUNT(\*) >= ALL (SELECT

COUNT(\*)

FROM

ciclista C1,

llevar l1

WHERE

l1.dorsal = c1.dorsal

GROUP BY c1.nomeq)

38.

SELECT

m.codigo, m.color

FROM

llevar L,

maillot M,

ciclista C,

etapa E

WHERE

NOT EXISTS( SELECT

m.codigo

FROM

maillot M2,

llevar L2,

ciclista C2,

etapa E2

WHERE

m.codigo = l.codigo

AND l.dorsal = c.dorsal

AND l.netapa = e.netapa)

GROUP BY m.codigo , m.color

39,

SELECT

e.netapa, e.salida, e.llegada

FROM

etapa E,

puerto P

WHERE

km IN (SELECT

km

FROM

etapa

WHERE

km > 190)

AND e.netapa = p.netapa

GROUP BY e.netapa

HAVING COUNT(nompuerto) >= 2

40,

SELECT

c.dorsal, c.nombre

FROM

ciclista C

WHERE

EXISTS( SELECT

l.dorsal

FROM

llevar l

WHERE

l.dorsal = 20

AND NOT EXISTS( SELECT

l2.dorsal

FROM

llevar L2

WHERE

l2.dorsal = c.dorsal

AND l2.codigo = l.codigo))

41,

SELECT

c.dorsal, c.nombre

FROM

ciclista c,

llevar l

WHERE

l.codigo IN (SELECT

codigo

FROM

llevar

WHERE

dorsal = 20)

AND c.dorsal != 20

AND c.dorsal = l.dorsal

group by c.dorsal, c.nombre

42.

SELECT

c.dorsal, c.nombre

FROM

ciclista c

WHERE

c.dorsal NOT IN (SELECT

l.dorsal

FROM

llevar l

WHERE

l.codigo IN (SELECT

l2.codigo

FROM

llevar l2

WHERE

l2.dorsal = 20))

**Con esta querie sabemos aquellos dorsales que han llevado el MISMO CODIGO de MAILLOT que DORSAL=20**

**SELECT distinct**

**dorsal**

**FROM**

**llevar L**

**WHERE**

**codigo IN (SELECT**

**l2.codigo**

**FROM**

**ciclista C2,**

**llevar L2**

**WHERE**

**c2.dorsal = 20 AND c2.dorsal=l2.dorsal) and dorsal!=20**

**GROUP BY dorsal**

43.

SELECT c.dorsal, c.nombre

FROM ciclista c

WHERE NOT EXISTS (

SELECT codigo

FROM llevar

WHERE dorsal = 20

AND codigo NOT IN (

SELECT codigo

FROM llevar

WHERE dorsal = c.dorsal

)

)

AND c.dorsal !=20;

44.

SELECT c.dorsal, c.nombre

FROM ciclista c

WHERE NOT EXISTS (

SELECT codigo

FROM llevar l1

WHERE l1.dorsal = 20 AND NOT EXISTS (

SELECT codigo

FROM llevar l2

WHERE l2.dorsal = c.dorsal AND l2.codigo = l1.codigo

)

) AND NOT EXISTS (

SELECT codigo

FROM llevar l3

WHERE l3.dorsal = c.dorsal AND NOT EXISTS (

SELECT codigo

FROM llevar l4

WHERE l4.dorsal = 20 AND l4.codigo = l3.codigo

)

) and c.dorsal!=20

45.

create view kms

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

from etapa E, llevar L, maillot M

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

group by l.dorsal, m.codigo

select l.dorsal, c.nombre, m.color, sum(e.km) as kmss

from etapa E, maillot M, llevar L, ciclista C

where e.netapa=l.netapa and m.codigo=l.codigo AND c.dorsal=l.dorsal

group by l.dorsal, c.nombre, m.color

having sum(e.km)= (select max(kms) from kms)

46.

SELECT dorsal, nombre

FROM ciclista

WHERE dorsal IN (

SELECT dorsal

FROM llevar

GROUP BY dorsal

HAVING COUNT(DISTINCT codigo) = (

SELECT COUNT(DISTINCT codigo) - 3

FROM llevar

WHERE dorsal = 1

)

AND dorsal != 1

);

47,

SELECT

e.netapa, e.km

FROM

etapa E,

puerto P

WHERE

p.netapa = e.netapa

GROUP BY e.netapa , e.km

**como saber la edad maxima de una clase**

**SELECT a1.nom, a1.edad**

**from alu A1**

**where a1.edad =(select max(a2.edad) from alumnos)**