

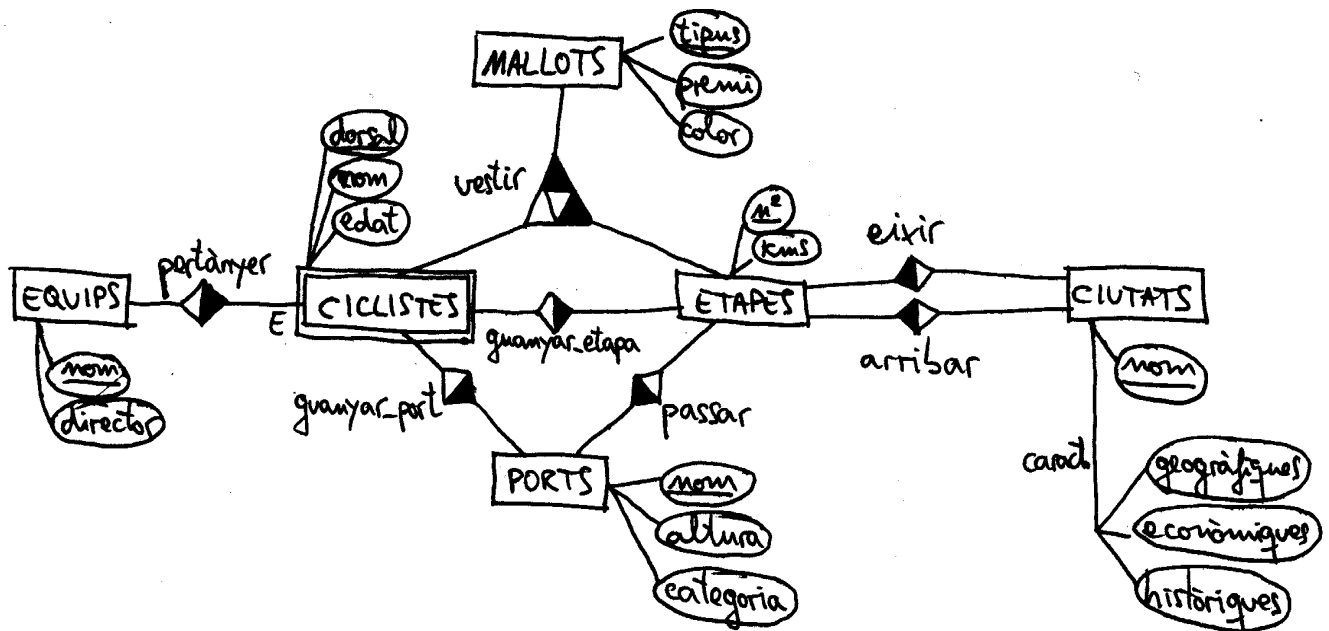
EXERCICIS SQL RESOLTS

SOBRE LA BD CICLISME



Diagrama Entitat - Relació	2
Esquema relacional	3
Disseny físic	3
Enunciats dels exercicis	4
Clàusula SELECT	4
Clàusula ORDER BY	4
Resultats de consultes	4
Operador DISTINCT	5
Clàusula WHERE	5
Falta de dades (valors NULL)	5
Clàusula GROUP BY	6
Clàusula HAVING	6
Combinació de consultes (UNION)	7
Consultes multi taula	8
Subconsultes	10
Actualitzacions (INSERT, UPDATE, DELETE)	13
Vistes (VIEW)	14
Solucions dels exercicis	15
Clàusula SELECT	15
Clàusula ORDER BY	16
Resultats de consultes	16
Operador DISTINCT	16
Clàusula WHERE	17
Falta de dades (valors NULL)	18
Clàusula GROUP BY	18
Clàusula HAVING	20
Combinació de consultes (UNION)	20
Consultes multi taula	23
Subconsultes	28
Actualitzacions (INSERT, UPDATE, DELETE)	41
Vistes (VIEW)	43

Diagrama Entitat - Relació



Notes:

- En la BD no existirà la taula ciutats.
- La relació "vestir" està en la BD en la taula "portar". Té la informació de quin ciclista vesteix (porta) cada mallot al final de cada etapa. És a dir: donada una etapa i un mallot, hi haurà un ciclista qui el portarà.
- Els exercicis estan classificats segons la temàtica o forma de resoldre'ls. Però cal tindre en compte que molts exercicis tenen distintes formes de resoldre'ls.

Esquema relacional

EQUIPS = nom + director

CICLISTES = dorsal + nom + edat + equip

C. Ali: equip → EQUIPS (nom)

VNN: equip

ETAPES = numero + kms + eixida + arribada + ciclista

C. Ali: ciclista → CICLISTES (dorsal)

PORTS = nom + categoria + etapa + ciclista + altura

C. Ali: etapa → ETAPES(numero)

C. Ali: ciclista → CICLISTES(dorsal)

MALLOTS = codi + tipus + color + premi

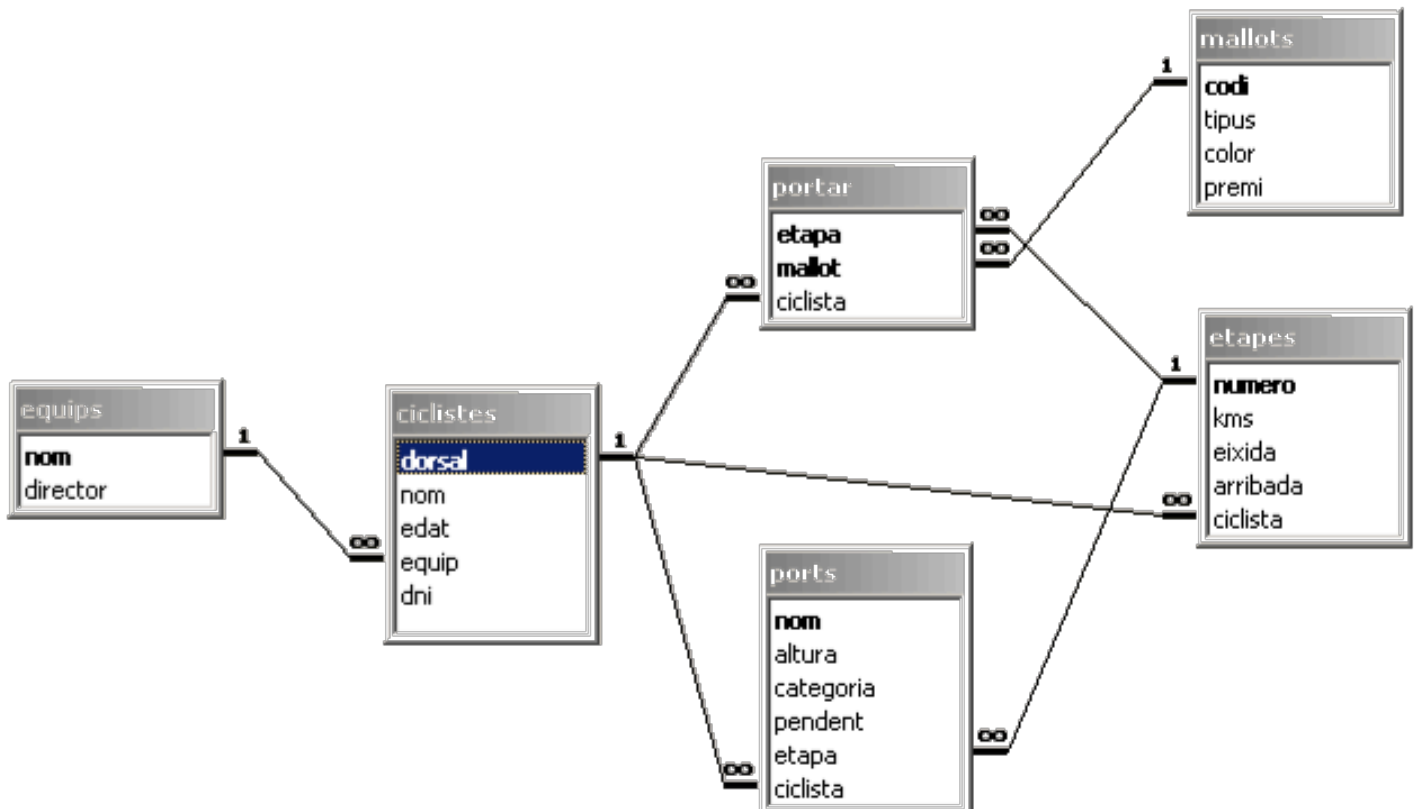
PORTAR = etapa + mallot + ciclista

C. Ali: etapa → ETAPES (numero)

C. Ali: mallot → MALLOTS (codi)

C. Ali: ciclista → CICLISTES (dorsal)

Disseny físic



Enunciats dels exercicis

Clàusula SELECT

sel_1) Selecciona tota la informació dels ports

sel_2) Selecciona el nom del port, l'altura en kilòmetres (està guardada en metres) i, al costat, que aparega la paraula 'Km'.

Funcions d'agregat (MAX, MIN, SUM, AVG, COUNT)

agr_1) Obteniu quants ciclistes hi ha

agr_2) Mostra l'edat mitjana dels ciclistes

agr_3) Obteniu l'altura mínima i màxima dels ports de muntanya.

agr_4) Calcula quants kilòmetres té en total la volta ciclista

Clàusula ORDER BY

ord_1) Ordena els ciclistes pel nom de l'equip. Dins de cada equip, primer eixiran els més vells i després els més joves. Si diversos ciclistes del mateix equip tenen la mateixa edat, deuran aparèixer ordenats pel nom del ciclista.

Resultats de consultes

res_1) Crea la taula *etapes_ciutat* que continga el número d'etapa i la ciutat d'aquelles etapes que comencen i acaben en la mateixa ciutat.

res_2) Crea la taula *premis* que continga el nom del color del mallot i el premi. Els noms de les columnes han de ser "Color del mallot" i "Valor del premi".

Operador DISTINCT

dis_1) Mostra quines edats tenen els ciclistes en general. És a dir, només s'han de mostrar les edats (sense repetir), no els noms dels ciclistes.

dis_2) Quantes edats diferents tenen els ciclistes?

Clàusula WHERE

whe_1) Mostra l'etapa i els km de les etapes entre 50 i 100 km

whe_2) Mostra l'etapa i els km de les altres etapes (<50 i >100)

whe_3) Obtén el nom dels ciclistes que comencen per 'Al'

whe_4) Noms d'equips que continguin 'tiac'

whe_5) Noms de ciutats d'arribada que continguin 'Naran', que tinguen altra lletra al costat, després una 'o' i després més coses.

whe_6) Noms de ciclistes que continguin 'Induráin' però que no siguin 'Miguel'.

whe_7) Selecciona els ports que estiguen entre 1500 i 2000 metres. A més, hauran de complir que no siguin de la categoria 'E' o que tinguen una pendent major que 5.

whe_8) Ciclistes que comencen per 'Al', 'An' o 'Ar' que tinguen entre 25 i 30 anys.

Falta de dades (valors NULL)

nul_1) Obteniu aquells ports que no tenen assignada categoria

Clàusula GROUP BY

gro_1) Calcula de cada equip l'edat mitjana, màxima, mínima i quants corredors té.

gro_2) Calcula quantes etapes ha guanyat cada corredor, ordenat primer per qui ha guanyat més etapes.

gro_3) Calcula, de cada categoria de port:

- L'altura màxima, mínima i mitjana
- La pendent màxima, mínima i mitjana
- Quants ports hi ha

gro_4) Calcula quantes voltes ha portat un corredor cada mallot

gro_5) Quants corredors hi ha de cada edat en cada equip?

Clàusula HAVING

hav_1) Calcula de cada equip amb més de 3 corredors: l'edat mitjana, màxima, mínima i quants corredors té.

hav_2) Calcula quantes etapes ha guanyat cada corredor, ordenat primer per qui ha guanyat més etapes. Només es mostraran els corredors que hagen guanyat més d'1 etapa.

hav_3) Mostra les categories de ports on l'altura màxima siga igual a la mínima:

hav_4) Dorsals que han guanyat més d'un port en una mateixa etapa.

Combinació de consultes (UNION)

uni_1) Selecciona els dorsals dels corredors de Banesto i també els que han guanyat alguna etapa.

uni_2) Seleccionen els ciclistes que han guanyat alguna etapa o algun port. També cal mostrar els kms de l'etapa o l'altura del port però tot expressat en metres. També cal mostrar si es tracta d'un port o d'una etapa. Primer deuen aparèixer els ports i després, les etapes. Les etapes han d'eixir ordenades per la longitud. I els ports, per l'altura.

uni_3) Donada la següent taula:

```
create table amics (  
    nom char(10),  
    cog1 char(10),  
    cog2 char(10))
```

- a) Mostra en una sola columna tots els cognoms dels amics, sense que isquen repetits.
- b) Fes el mateix però mostra ara han d'eixir tots els cognoms, encara que isquen repetits.
- c) Fes el mateix però ara els primers cognoms sense repetir; i els segons, repetits.

uni_4) Diques si hi ha alguna diferència entre aquestes 2 sentències:

- a)

```
select cog1 from amics  
  
union  
  
select cog2 from amics
```
- b)

```
select distinct cog1 from amics  
  
union all  
  
select distinct cog2 from amics
```

uni_5) Selecciona els números d'etapa de més de 200 km, les etapes amb més de 2 ports i les etapes en què algun corredor ha portat més d'un mallot.

uni_6) Obté el total de premis que ha aconseguit cada ciclista, classificant els tipus de premis (ports guanyats, etapes guanyades i mallots portats). Ordenat pel dorsal i per la quantitat de premis. Tindrà el següent aspecte:

<u>Dorsal</u>	<u>Premis</u>	<u>Tipus</u>
1	1	ports
1	2	etapes
1	14	mallots
2	3	etapes
2	4	mallots
2	4	ports
3	2	mallots
4	1	etapes
...		

Consultes multi taula

mul_1) Obtén els noms dels corredors amb el nom dels seus directors.

mul_2) Mostra entre quines 2 ciutats està cada port

mul_3) Mostra quants corredors té cada director

mul_4) Mostra el nom del port i nom del ciclista que l'ha guanyat.

mul_5) Obté el nom del ciclista que ha guanyat cada etapa.

mul_6) Mostra el número d'etapa i l'equip del corredor que l'ha guanyada.

mul_7) Mostra el número d'etapa i el director de l'equip de qui ha guanyat l'etapa

mul_8) Obtén el nom dels corredors que guanyaren algun port de l'etapa que començava en Igualada.

mul_9) Mostra el codi i el color del mallot que porta cada ciclista (dorsal i nom) en cada etapa.

mul_10) Mostra les dades de les etapes que passen per algun port i que tinguen l'eixida i l'arribada a la mateixa ciutat.

mul_11) Equips que han guanyat alguna etapa portant algun mallot. També cal mostrar el número d'etapa, l'eixida i el mallot.

mul_12) Obté el número d'etapes guanyades per cada equip. Primer eixiran els equips que més etapes han guanyat. En cas d'igualtat, eixiran ordenats alfabèticament pel nom de l'equip.

mul_13) Mostra els noms dels corredors que han guanyat més d'una etapa i quantes etapes ha guanyat cadascú. Primer han d'eixir els que han guanyat més etapes. En cas d'empat, ordena-ho pel nom del corredor.

mul_14) Obteniu el nom dels ciclistes que han portat més d'un mallot en alguna etapa. Mostreu també en quina etapa, i quants mallots portava en eixa etapa. Primer eixiran els que han portat més mallots en una etapa. En cas d'empat, ordenat pel nom alfabèticament.

mul_15) Per a cada etapa, volem saber la següent informació:

- Número
- Ciutat origen
- Informació dels corredors que duen mallot en eixa etapa:
 - Codi del mallot
 - Color del mallot
 - Dorsal del corredor
 - Nom del corredor
 - Nom del director del corredor
- Informació dels ports de l'etapa:
 - Quants ports hi ha en l'etapa
 - L'altura mínima, màxima i mitjana dels ports de l'etapa

mul_16) Obteniu parelles d'etapes que tenen els mateix nombre de km. Cal mostrar també el nombre de km. Però tenint en compte que:

- no han d'eixir parelles de la mateixa etapa (p.e: 10 10)
- ni tampoc han d'eixir parelles repetides (p.e: 9 6 i 6 9)

mul_17) Obtén parelles de nom de ciclista i número d'etapa, tal que eixe ciclista haja guanyat eixa etapa havent portat el mallot color 'groc' almenys una vegada.

mul_18) Fes una *select* per a cadascun dels següents apartats:

- a) Mostra el número de cada etapa, la ciutat d'eixida, el dorsal del guanyador i el dorsal de qui portava el mallot groc d'eixa etapa.
- b) Mostra a) més el nom del guanyador de l'etapa
- c) Mostra a) més el nom de qui portava el mallot groc
- d) Mostra a) més nom guanyador etapa i nom portador mallot groc

Subconsultes

sub_1) Noms de ciclistes que no han guanyat cap etapa

sub_2) Quantitat d'etapes que no tenen ports

sub_3) Obteniu número d'etapa i km d'aquelles etapes que no les ha guanyades cap corredor del Banesto.

sub_4) Volem mostrar qui ha guanyat cada mallot al final de la volta ciclista. Per tant, mostrarem el color de cada mallot, el premi corresponent i el nom del corredor que l'ha guanyat. Nota: tindrem en compte que el guanyador d'un mallot és el ciclista que l'ha portat en l'última etapa.

sub_5) Quants ciclistes no han guanyat res (ni etapes, ni ports ni han portat mallot)

sub_6) Nom del ciclista que han guanyat el port més alt i el del mes baix. També cal mostrar el nom del port i l'altura.

sub_7) Obteniu les ciutats que tenen la meta d'alguna etapa però des de les quals no es fa cap eixida.

sub_8) Mostra la informació dels ciclistes d'edat màxima de cada equip.

sub_9) Obteniu el nom dels ports de muntanya que tenen una altura superior a l'altura mitjana de tots els ports.

sub_10) Obteniu el nom dels ciclistes que pertanyen a un equip de més de 5 ciclistes i que han guanyat alguna etapa. Mostra també quantes etapes han guanyat.

sub_11) Obteniu les dades de les etapes que no comencen en la mateixa ciutat en que acaba l'anterior.

sub_12) Nom de ciclista i voltes que ha portat el mallot groc (MGE) d'aquell ciclista que ha portat més voltes eixe mallot.

sub_13 Mostra el número d'etapa i quants ports té l'etapa amb més ports.

sub_14) Mostra el número d'etapa i quants ports té l'etapa amb més ports i la de menys ports.

sub_15) Nom dels ciclistes que han portat tots els mallots

sub_16) Qui ha guanyat més etapes i quantes?
`select ciclista, count(*) as "Etapes guanyades"`

sub_17) Obtenir les etapes (número, eixida i arribada) que tenen els ports amb major pendent (és a dir, que la mitja dels pendents d'eixa etapa siga major que els de les altres etapes). També cal mostrar la mitja del pendent de l'etapa.

sub_18) Obteniu el dorsal i nom dels ciclistes que han guanyat l'etapa amb el port més alt.

sub_19) Obteniu les dades de les etapes tals que tots els seus ports superen els 1300 metres d'altura.

sub_20) Obteniu el codi i color dels mallots que només han sigut portats per ciclistes d'un mateix equip.

sub_21) Ports més alts o més baixos de cada etapa. Ha d'eixir l'etapa, nom dels ports, altures i una indicació de si es tracta d'un port 'alt' o 'baix'.

sub_22) Nom i edat del ciclista més jove.

sub_23) Obteniu el nom dels equips en els quals els seus ciclistes només hagen guanyat ports de 1a categoria 1.

sub_24) Obteniu el número d'etapa, la ciutat d'eixida i la ciutat d'arribada de les etapes de més de 190 km. i que tinguen almenys 2 ports.

sub_25) Obteniu el codi i el color del mallot que ha sigut portat per algun ciclista que no ha guanyat cap etapa.

sub_26) Obteniu el dorsal i el nom dels ciclistes que han portat almenys un mallot dels que ha portat el ciclista de dorsal 20.

sub_27) Obteniu el dorsal i el nom del ciclista que ha portat durant més quilòmetres un mateix mallot i indiqueu també el color d'eixe mallot.

sub_28) Nom dels equips tals que tots els seus corredors han portat algun mallot o han guanyat algun port.

sub_29) Nom dels ciclistes que han guanyat tots els ports d'una etapa i, a més, han guanyat eixa mateixa etapa.

sub_30) Nom, edat i equip dels ciclistes més joves de cada equip

Actualitzacions (INSERT, UPDATE, DELETE)

dml_1) Esborra els equips que no tinguen cap ciclista.

dml_2) Incrementa en un 10% el premi del mallot amb codi MRE.

dml_3) Qui va contar els kms de les primeres 10 etapes tenia mal el conta quilòmetres. Cal augmentar un 10 % els kms d'eixes etapes.

dml_4) Si no estiguera posada la clau aliena de *portar cap a mallots*, podria donar-se el cas que algun mallot que ha portat algú no figure en la taula de *mallots*. Insereix en la taula de mallots els mallots que han sigut portats però no figuren en la taula de mallots, de forma que tinguen el codi corresponent i tipus "desconegut".

dml_5) A última hora s'ha apuntat un nou corredor, el 101, de nom Pep Garcia (de qui no sabem l'edat) que és de l'equip Corremolt, el director del qual és Miquel Marqués. Fes les instruccions SQL necessàries per a actualitzar la base de dades.

dml_6) A causa d'un terratrèmol, el port 'Portet' ha baixat 100m. Modifica-ho.

dml_7) Modifica la ciutat d'arribada de l'etapa 3 perquè siga la mateixa que la seua ciutat d'eixida.

dml_8) Duplica el premi de tots els mallots.

dml_9) El director d'equip Manuel Sainz no va dir correctament l'edat dels seus corredors. Cal augmentar en 1 l'edat dels seus corredors.

dml_10) Esborra els corredors que no tinguen cap equip assignat o que estiguen assignats a un equip que no figure en la taula d'equips.

Vistes (VIEW)

vis_1) Vista *ports_alts* amb totes les dades dels ports de més de 2000 metres.

vis_2) Insereix un port a la vista de més de 2000 m i altre de menys de 2000. Deixa? Es veuen a la vista?

vis_3) Vista *guanyadors_etapes_llargues* que tinga: número de l'etapa, kms, dorsal i nom del guanyador, l'equip i el director, d'aquelles etapes de més de 100 kms. Els camps de la vista es diran: *etapa*, *kms*, *dorsal*, *nom*, *equip* i *director*.

vis_4) Inseriu en la vista anterior l'etapa 200 amb el guanyador 1

vis_5) Vista *etapes_altes* que tinga per cada etapa: número d'etapa i quantitat de ports de més de 2000 m. Fes-ho a partir de la vista *ports_alts* que has creat. Els camps de la vista seran: *etapa* i *ports_alts*. És actualitzable? Per què?

vis_6) Volem mostrar, per a cada equip, el nom de l'equip, quants corredors té i quants mallots ha portat. Però només per als equips que hagen portat més mallots entre els equips amb igual quantitat de corredors. Caldrà mostrar primer els equips amb menys corredors. En cas d'empat, pel nom de l'equip:

equip	q_ciclistes	q_mallots
Castorama	2	4
Jolly Club	2	4
Seguros Amaya	3	7
Gatorade	4	3
ONCE	5	7
TVM	6	7
Artiach	7	16
Gewiss	8	12
Banesto	11	38

De tots els equips amb 5 ciclistes, l'equip ONCE és el que més mallots han portat entre tots els seus ciclistes: 7

Com és una consulta complicada, fes-te primer una vista que tinga el codi de l'equip i quants corredors té cadascun. Després fes la consulta que es demana ajudant-te de la vista que has creat.

Solucions dels exercicis

Clàusula SELECT

sel_1) Selecciona tota la informació dels ports

```
select *  
from ports;
```

sel_2) Selecciona el nom del port, l'altura en kilòmetres (està guardada en metres) i, al costat, que aparega la paraula 'Km'.

```
select nom, (altura / 1000) as altura_kms, 'kms'  
from ports;
```

Funcions d'agregat (MAX, MIN, SUM, AVG, COUNT)

agr_1) Obteniu quants ciclistes hi ha

```
select count(*) as quant_ciclistes  
from ciclistes;
```

agr_2) Mostra l'edat mitjana dels ciclistes

```
select avg(edat) as mitja  
from ciclistes;
```

agr_3) Obteniu l'altura mínima i màxima dels ports de muntanya.

```
select min(altura) as mínima, max(altura) as màxima  
from ports;
```

agr_4) Calcula quants kilòmetres té en total la volta ciclista

```
select sum(kms)  
from etapes;
```

Clàusula ORDER BY

ord_1) Ordena els ciclistes pel nom de l'equip. Dins de cada equip, primer eixiran els més vells i després els més joves. Si diversos ciclistes del mateix equip tenen la mateixa edat, deuran aparèixer ordenats pel nom del ciclista.

```
select *  
from ciclistes  
order by equip, edat desc, nom;
```

Resultats de consultes

res_1) Crea la taula *etapes_ciutat* que continga el número d'etapa i la ciutat d'aquelles etapes que comencen i acaben en la mateixa ciutat.

```
create table etapes_ciutat  
select numero, eixida  
from etapes  
where eixida=arribada;
```

res_2) Crea la taula *premis* que continga el nom del color del mallot i el premi. Els noms de les columnes han de ser "Color del mallot" i "Valor del premi".

```
create table premis  
select color as "color del mallot", premi as "valor del premi"  
into premis  
from mallots;
```

Operador DISTINCT

dis_1) Mostra quines edats tenen els ciclistes en general. És a dir, només s'han de mostrar les edats (sense repetir), no els noms dels ciclistes.

```
select distinct edat  
from ciclistes;
```

dis_2) Quantes edats diferents tenen els ciclistes?

```
select count(distinct edat)  
from ciclistes;
```


Clàusula WHERE

whe_1) Mostra l'etapa i els km de les etapes entre 50 i 100 km

```
select numero, kms
from etapes
where kms between 50 and 100;
```

Altra forma:

```
select numero, kms
from etapes
where (kms >= 50) and (kms <= 100);
```

whe_2) Mostra l'etapa i els km de les altres etapes (<50 i >100)

```
select numero, kms
from etapes
where kms not between 50 and 100;
```

Altra forma:

```
select numero, kms
from etapes
where (kms < 50) or (kms > 100);
```

whe_3) Obtén el nom dels ciclistes que comencen per 'Al'

```
select *
from ciclistes
where nom like 'al%';
```

whe_4) Noms d'equips que continguin 'tiac'

```
select *
from equips
where nom like '%tiac%';
```

whe_5) Noms de ciutats d'arribada que continguin 'Naran', que tinguin altra lletra al costat, després una 'o' i després més coses.

```
select *
from etapes
where arribada like '%Naran_o%';
```

whe_6) Noms de ciclistes que continguin 'Induráin' però que no siguin 'Miguel'.

```
select *
from ciclistes
where nom like '%Induráin%' and nom not like '%Miguel%';
```

whe_7) Selecciona els ports que estiguen entre 1500 i 2000 metres. A més, hauran de complir que no siguin de la categoria 'E' o que tinguen una pendent major que 5.

```
select *  
from ports  
where (altura between 1500 and 2000)  
and (categoria <> 'e' or pendent > 5);
```

whe_8) Ciclistes que comencen per 'Al', 'An' o 'Ar' que tinguen entre 25 i 30 anys.

```
select *  
from ciclistes  
where (nom like 'Al%' or nom like 'An%' or nom like 'Ar%')  
and (edat between 25 and 30);
```

Falta de dades (valors NULL)

nul_1) Obteniu aquells ports que no tenen assignada categoria

```
select *  
from ports  
where categoria is null;
```

Clàusula GROUP BY

gro_1) Calcula de cada equip l'edat mitjana, màxima, mínima i quants corredors té.

```
select equip, round(avg(edat)) as mitjana, max(edat) as maxima,  
min(edat) as minima, count(*) as quants  
from ciclistes  
group by equip;
```

gro_2) Calcula quantes etapes ha guanyat cada corredor, ordenat primer per qui ha guanyat més etapes.

```
select ciclista, count(*) as etapes_guanyades  
from etapes  
group by ciclista  
order by 2 desc;
```

gro_3) Calcula, de cada categoria de port:

- L'altura màxima, mínima i mitjana
- La pendent màxima, mínima i mitjana
- Quants ports hi ha

```
select categoria,  
       max(altura) as max_a, min(altura) as min_a,  
       round(avg(altura)) as mitja_a,  
       max(pendent) as max_p, min(pendent) as min_p,  
       round(avg(pendent)) as mitja_p,  
       count(*) as n_ports  
from ports  
group by categoria;
```

gro_4) Calcula quantes voltes ha portat un corredor cada mallot

```
select ciclista, mallot, count(*)  
from portar  
group by ciclista, mallot;
```

gro_5) Quants corredors hi ha de cada edat en cada equip?

```
select equip, edat, count(*) as quants  
from ciclistes  
group by equip, edat;
```

gro_1) Calcula de cada equip amb més de 3 corredors: l'edat mitjana, màxima, mínima i quants corredors té.

```
select equip, round(avg(edat)) as mitjana, max(edat) as maxima,  
       min(edat) as minima, count(*) as quants  
from ciclistes  
group by equip  
having count(*) >3;
```

gro_2) Calcula quantes etapes ha guanyat cada corredor, ordenat primer per qui ha guanyat més etapes. Només es mostraran els corredors que hagen guanyat més d'1 etapa.

```
select ciclista, count(*) as etapes_guanyades  
from etapes  
group by ciclista  
having count(*) > 1  
order by 2 desc;
```

Clàusula HAVING

hav_1) Mostra les categories de ports on l'altura màxima siga igual a la mínima:

```
select categoria
from ports
group by categoria
having max(altura) = min(altura);
```

hav_2) Dorsals que han guanyat més d'un port en una mateixa etapa.

```
select ciclista
from ports
group by ciclista, etapa
having count(*) > 1;
```

Combinació de consultes (UNION)

uni_1) Selecciona els dorsals dels corredors de Banesto i també els que han guanyat alguna etapa.

```
select dorsal
from ciclistes
where equip = 'Banesto'
```

union

```
select ciclista
from etapes;
```

uni_2) Seleccionen els ciclistes que han guanyat alguna etapa o algun port. També cal mostrar els kms de l'etapa o l'altura del port però tot expressat en metres. També cal mostrar si es tracta d'un port o d'una etapa. Primer deuen aparèixer els ports i després, les etapes. Les etapes han d'eixir ordenades per la longitud. I els ports, per l'altura.

```
select ciclista, kms*1000 as metres, 'Etapa' as tipus
from etapes
```

union

```
select ciclista, altura, 'Port'
from ports
```

```
order by 3 desc, 2;
```

uni_3) Donada la següent taula:

```
create table amics (  
    nom char(10),  
    cog1 char(10),  
    cog2 char(10))
```

a) Mostra en una sola columna tots els cognoms dels amics, sense que isquen repetits.

```
select cog1 from amics  
union  
select cog2 from amics;
```

b) Fes el mateix però mostra ara han d'eixir tots els cognoms, encara que isquen repetits.

```
select cog1 from amics  
union all  
select cog2 from amics;
```

c) Fes el mateix però ara els primers cognoms sense repetir; i els segons, repetits.

```
select distinct cog1 from amics  
union all  
select cog2 from amics;
```

uni_4) Digues si hi ha alguna diferència entre aquestes 2 sentències:

```
a)    select cog1 from amics  
      union  
      select cog2 from amics
```

```
b)    select distinct cog1 from amics  
      union all  
      select distinct cog2 from amics
```

Sí que hi ha diferencia:

a) No eixirà cap cognom repetit

b) Eixiran repetits els cognoms que apareguen tant en cog1 com en cog2.

uni_5) Selecciona els números d'etapa de més de 200 km, les etapes amb més de 2 ports i les etapes en què algun corredor ha portat més d'un mallot.

```
select numero  
from etapes  
where kms > 200
```

union

```
select etapa  
from ports  
group by etapa  
having count(*) > 2
```

union

```
select etapa  
from portar  
group by etapa, ciclista  
having count(*) > 1;
```

uni_6) Obté el total de premis que ha aconseguit cada ciclista, classificant els tipus de premis (ports guanyats, etapes guanyades i mallots portats). Ordenat pel dorsal i per la quantitat de premis. Tindrà el següent aspecte:

<u>Dorsal</u>	<u>Premis</u>	<u>Tipus</u>
1	1	ports
1	2	etapes
1	14	mallots
2	3	etapes
2	4	mallots
2	4	ports
3	2	mallots
4	1	etapes
...		

```
select ciclista as dorsal, count(*) as premis,  
'etapes' as tipus  
from etapes  
group by ciclista
```

union

```
select ciclista, count(*), 'ports'  
from ports  
group by ciclista
```

union

```
select ciclista, count(*), 'mallots'  
from portar  
group by ciclista
```

```
order by 1, 4;
```

Consultes multi taula

mul_1) Obtén els noms dels corredors amb el nom dels seus directors.

```
select ciclistes.nom, equips.director
from ciclistes, equips
where ciclistes.equip = equips.nom;
```

mul_2) Mostra entre quines 2 ciutats està cada port

```
select nom, eixida, arribada
from ports, etapes
where ports.etapa = etapes.numero;
```

mul_3) Mostra quants corredors té cada director

```
select director, count(*)
from equips, ciclistes
where equips.nom = ciclistes.equip
group by director;
```

mul_4) Mostra el nom del port i nom del ciclista que l'ha guanyat.

```
select ports.nom, ciclistes.nom
from ports, ciclistes
where ports.ciclista = ciclistes.dorsal;
```

mul_5) Obté el nom del ciclista que ha guanyat cada etapa.

```
select etapes.numero, ciclistes.nom
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
```

mul_6) Mostra el número d'etapa i l'equip del corredor que l'ha guanyada.

```
select etapes.numero, ciclistes.equip
from etapes, ciclistes
where etapes.ciclista = ciclistes.dorsal
```

mul_7) Mostra el número d'etapa i el director de l'equip de qui ha guanyat l'etapa

```
select etapes.numero, equips.director
from etapes, ciclistes, equips
where etapes.ciclista = ciclistes.dorsal
and ciclistes.equip = equips.nom
```

mul_8) Obtén el nom dels corredors que guanyaren algun port de l'etapa que començava en Igualada.

```
select ciclistes.nom
from ciclistes, ports, etapes
where ciclistes.dorsal = ports.ciclista
and ports.etapa = etapes.numero
and etapes.eixida = 'Igualada'
```

mul_9) Mostra el codi i el color del mallot que porta cada ciclista (dorsal i nom) en cada etapa.

```
select portar.etapa, portar.mallot, mallots.color, portar.ciclista, ciclistes.nom
from portar, mallots, ciclistes
where portar.mallot = mallots.codi
and portar.ciclista = ciclistes.dorsal
order by portar.etapa, portar.mallot
```

mul_10) Mostra les dades de les etapes que passen per algun port i que tinguin l'eixida i l'arribada a la mateixa ciutat.

```
select distinct etapes.numero
from etapes, ports
where etapes.numero = ports.etapa
and etapes.eixida = etapes.arribada
```

mul_11) Equips que han guanyat alguna etapa portant algun mallot. També cal mostrar el número d'etapa, l'eixida i el mallot.

```
select ciclistes.equipo, etapes.numero, etapes.eixida, portar.mallot
from ciclistes, portar, etapes
where ciclistes.dorsal = portar.ciclista
and ciclistes.dorsal = etapes.ciclista
and etapes.numero = portar.etapa
```

mul_12) Obté el número d'etapes guanyades per cada equip. Primer eixiran els equips que més etapes han guanyat. En cas d'igualtat, eixiran ordenats alfabèticament pel nom de l'equip.

```
select equipo, count(*)
from etapes, ciclistes
where etapes.ciclista = ciclistes.dorsal
group by equipo
order by 2 desc, 1
```


mul_13) Mostra els noms dels corredors que han guanyat més d'una etapa i quantes etapes ha guanyat cadascú. Primer han d'eixir els que han guanyat més etapes. En cas d'empat, ordena-ho pel nom del corredor.

```
select nom, count(*) as etapes_guanyades
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
group by nom
having count(*) > 1
order by 2 desc, 1
```

mul_14) Obteniu el nom dels ciclistes que han portat més d'un mallot en alguna etapa. Mostreu també en quina etapa, i quants mallots portava en eixa etapa. Primer eixiran els que han portat més mallots en una etapa. En cas d'empat, ordenat pel nom alfabèticament.

```
select ciclistes.nom, portar.etapa, count(*) as quants
from ciclistes, portar
where ciclistes.dorsal = portar.ciclista
group by nom, etapa
having count(*) > 1
order by 3 desc, 1
```

mul_15) Per a cada etapa, volem saber la següent informació:

- Número
- Ciutat origen
- Informació dels corredors que duen mallot en eixa etapa:
 - Codi del mallot
 - Color del mallot
 - Dorsal del corredor
 - Nom del corredor
 - Nom del director del corredor
- Informació dels ports de l'etapa:
 - Quants ports hi ha en l'etapa
 - L'altura mínima, màxima i mitjana dels ports de l'etapa

```
select et.numero, et.eixida, po.mallot, ma.color, po.ciclista, ci.nom, eq.director,
       count(*) as n_ports, min(altura) as min_alt, max(altura) as max_alt,
       avg(altura) as mitj_alt
from etapes et, portar po, mallots ma, ciclistes ci, equips eq, ports pu
where et.numero = po.etapa
and po.mallot = ma.codi
and po.ciclista = ci.dorsal
and ci.equip = eq.nom
and et.numero = pu.etapa
group by 1, 2, 3, 4, 5, 6, 7;
```

mul_16) Obteniu parelles d'etapes que tenen els mateix nombre de km. Cal mostrar també el nombre de km. Però tenint en compte que:

- no han d'eixir parelles de la mateixa etapa (p.e: 10 10)
- ni tampoc han d'eixir parelles repetides (p.e: 9 6 i 6 9)

```
select e1.numero, e2.numero, e1.kms
from etapes e1, etapes e2
where e1.kms = e2.kms
and e1.numero < e2.numero
```

mul_17) Obtén parelles de nom de ciclista i número d'etapa, tal que eixe ciclista haja guanyat eixa etapa havent portat el mallot color 'groc' almenys una vegada.

```
select distinct ciclistes.nom, etapes.numero
from ciclistes, etapes, portar, mallots
where ciclistes.dorsal = etapes.ciclista
and ciclistes.dorsal = portar.ciclista
and portar.etapa < etapes.numero
and portar.mallot = mallots.codi
and mallots.color = 'groc';
```

Nota: and portar.etapa < etapes.numero és un truquet per a reflectir esta part: "havent portat ...almenys 1 vegada"

O bé, amb subselects:

```
select ciclistes.nom, etapes.numero
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
and ciclistes.dorsal in ( select portar.ciclista
                        from portar, mallots
                        where portar.mallot = mallots.codi
                        and mallots.color = 'groc'
                        and portar.etapa < etapes.numero);
```

mul_18) Fes una select per a cadascun dels següents apartats:

- a) Mostra el número de cada etapa, la ciutat d'eixida, el dorsal del guanyador i el dorsal de qui portava el mallot groc d'eixa etapa.

```
select e.numero, e.eixida,
       e.ciclista as 'guanyador',
       p.ciclista as 'groc'
from etapes e, portar p, mallots m
where e.numero = p.etapa
and p.mallot = m.codi
and m.color = 'groc'
order by e.numero;
```

b) Mostra a) més el nom del guanyador de l'etapa

```
select e.numero, e.eixida,  
       e.ciclista as 'guanyador', c.nom as 'nom guanyador',  
       p.ciclista as 'groc'  
from etapes e, ciclistes c, portar p, mallots m  
where e.numero = p.etapa  
      and p.mallot = m.codi  
      and m.color = 'groc'  
      and e.ciclista = c.dorsal  
order by e.numero;
```

c) Mostra a) més el nom de qui portava el mallot groc

```
select e.numero, e.eixida,  
       e.ciclista as guanyador,  
       p.ciclista as 'groc', c.nom as 'nom groc'  
from etapes e, ciclistes c, portar p, mallots m  
where e.numero = p.etapa  
      and p.mallot = m.codi  
      and m.color = 'groc'  
      and p.ciclista = c.dorsal  
order by e.numero;
```

d) Mostra a) més nom guanyador etapa i nom portador mallot groc

Nota: ací necessitem 2 noms de ciclistes que no tenen res a veure entre ells.
Per tant, “cal utilitzar 2 vegades” la taula de ciclistes per a fer els enllaços corresponents. Per a això, caldrà nomenar-les amb noms diferents, mitjançant l'ús d'àlies:

```
select e.numero, e.eixida,  
       e.ciclista as 'guanyador', c_gua.nom as 'nom guanyador',  
       p.ciclista as 'groc', c_groc.nom as 'nom groc'  
from etapes e, ciclistes c_gua, portar p, mallots m, ciclistes c_groc  
where e.numero = p.etapa  
      and p.mallot = m.codi  
      and m.color = 'groc'  
      and e.ciclista = c_gua.dorsal  
      and p.ciclista = c_groc.dorsal  
order by e.numero;
```

Subconsultes

sub_1) Noms de ciclistes que no han guanyat cap etapa

```
select nom
from ciclistes
where dorsal not in (select ciclista from etapes)
```

sub_2) Quantitat d'etapes que no tenen ports

```
select count(*)
from etapes
where numero not in (select etapa from ports)
```

sub_3) Obteniu número d'etapa i km d'aquelles etapes que no les ha guanyades cap corredor del Banesto.

```
select numero, kms
from etapes
where ciclista not in (select dorsal
                        from ciclistes
                        where equip = 'Banesto')
```

També es podria fer sense subselects:

```
select numero, kms
from etapes, ciclistes
where etapes.ciclista = ciclistes.dorsal
      and ciclistes.equip <> 'Banesto'
```

sub_4) Volem mostrar qui ha guanyat cada mallot al final de la volta ciclista. Per tant, mostrarem el color de cada mallot, el premi corresponent i el nom del corredor que l'ha guanyat. Nota: tindrem en compte que el guanyador d'un mallot és el ciclista que l'ha portat en l'última etapa.

```
select color, premi, nom
from mallots, portar, ciclistes
where mallots.codi = portar.mallot
      and portar.ciclista = ciclistes.dorsal
      and portar.etapa = (select max(numero) from etapes)
```

sub_5) Quants ciclistes no han guanyat res (ni etapes, ni ports ni han portat mallot)

```
select count(*)
from ciclistes
where dorsal not in (select ciclista from etapes)
and dorsal not in (select ciclista from ports)
and dorsal not in (select ciclista from portar)
```

sub_6) Nom del ciclista que han guanyat el port més alt i el del mes baix. També cal mostrar el nom del port i l'altura.

```
select ciclistes.nom, ports.nom, ports.altura
from ciclistes, ports
where ciclistes.dorsal = ports.ciclista
and ( ports.altura = (select max(altura) from ports)
or
ports.altura = (select min(altura) from ports)
);
```

sub_7) Obteniu les ciutats que tenen la meta d'alguna etapa però des de les quals no es fa cap eixida.

```
select arribada
from etapes
where arribada not in (select eixida from etapes)
```

sub_8) Mostra la informació dels ciclistes d'edat màxima de cada equip.

```
select *
from ciclistes as c1
where edat = (select max(edat)
from ciclistes as c2
where c2.equip = c1.equip)
```

sub_9) Obteniu el nom dels ports de muntanya que tenen una altura superior a l'altura mitjana de tots els ports.

```
select nom
from ports
where altura > (select avg(altura) from ports)
```

sub_10) Obteniu el nom dels ciclistes que pertanyen a un equip de més de 5 ciclistes i que han guanyat alguna etapa. Mostra també quantes etapes han guanyat.

```
select ciclistes.nom, count(*) as etapes_guanyades
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
      and equip in (select equip
                    from ciclistes
                    group by equip
                    having count(*) > 5)
group by ciclistes.nom
```

sub_11) Obteniu les dades de les etapes que no comencen en la mateixa ciutat en que acaba l'anterior.

Sol. 1. No eixirà la primera etapa (ja que no hi ha una etapa anterior):

```
select *
from etapes
where eixida != (select arribada
                from etapes et_ant
                where et_ant.numero = etapes.numero - 1)
```

Sol. 2. Sí que eixirà la primera etapa:

```
select *
from etapes
where eixida not in (select arribada
                    from etapes et_ant
                    where et_ant.numero = etapes.numero - 1)
```

Sol. 3. Esta tampoc mostra l'etapa 1:

```
select etap_ant.*
from etapes etap_ant, etapes
where etap_ant.numero = etapes.numero + 1
      and etap_ant.eixida <> etapes.arribada
```

sub_12) Nom de ciclista i voltes que ha portat el mallot groc (MGE) d'aquell ciclista que ha portat més voltes eixe mallot.

```
select ciclistes.nom, count(*) AS max_grocs
from ciclistes, portar
where ciclistes.dorsal = portar.ciclista
  and portar.mallot = 'MGE'
group by ciclistes.nom
having count(*) >= all (select count(*)
                      from ciclistes, portar
                      where ciclistes.dorsal = portar.ciclista
                        and portar.mallot = 'MGE'
                      group by ciclistes.nom)
```

sub_13) Mostra el número d'etapa i quants ports té l'etapa amb més ports.

```
select etapa, count(*)
from ports
group by etapa
having count(*) >= all (select count(*)
                      from ports
                      group by etapa)
```

sub_14) Mostra el número d'etapa i quants ports té l'etapa amb més ports i la de menys ports.

```
select etapa, count(*)
from ports
group by etapa
having count(*) >= all (select count(*)
                      from ports
                      group by etapa)

union
```

```
select etapa, count(*)
from ports
group by etapa
having count(*) <= all (select count(*)
                      from ports
                      group by etapa)
```

sub_15) Nom dels ciclistes que han portat tots els mallots

```
select nom
from ciclistes
where not exists (select *
                  from mallots
                  where codi not in (select mallot
                                     from portar
                                     where ciclista = ciclistes.dorsal)
                  )
```

Nota: la següent solució està mal perquè compta quants mallots ha portat cada ciclista, però poden estar repetits:

```
select nom
from ciclistes, portar
where ciclistes.dorsal = portar.ciclista
group by nom
having count(*) = (select count(*) from mallots)
```

sub_16) Qui ha guanyat més etapes i quantes?

```
select ciclista, count(*) as "Etapas guanyades"
from etapes
group by ciclista
having count(*) >= all (select count(*)
                       from etapes
                       group by ciclista)
```

sub_17) Obtenir les etapes (número, eixida i arribada) que tenen els ports amb major pendent (és a dir, que la mitja dels pendents d'eixa etapa siga major que els de les altres etapes). També cal mostrar la mitja del pendent de l'etapa.

```
select numero, eixida, arribada, avg(pendent)
from etapes, ports
where etapes.numero = ports.etapa
group by numero, eixida, arribada
having avg(pendent) >= all (select avg(pendent)
                           from ports
                           group by etapa
                           )
```


sub_18) Obteniu el dorsal i nom dels ciclistes que han guanyat l'etapa amb el port més alt.

```
select dorsal, nom
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
      and etapes.numero in (select etapa
                             from ports
                             where altura = (select max(altura) from ports))
```

sub_19) Obteniu les dades de les etapes tals que tots els seus ports superen els 1300 metres d'altura.

```
select *
from etapes
where numero in ( select etapa
                  from ports
                  group by etapa
                  having min(altura) > 1300)
```

O bé, sense subselects:

```
select numero, kms, eixida, arribada, etapes.ciclista
from etapes, ports
where etapes.numero = ports.etapa
group by numero, kms, eixida, arribada, etapes.ciclista
having min(altura) > 1300
```

sub_20) Obteniu el codi i color dels mallots que només han sigut portats per ciclistes d'un mateix equip.

```
select distinct codi, color
from mallots, portar, ciclistes
where mallots.codi = portar.mallot
      and portar.ciclista = ciclistes.dorsal
      and ciclistes.equip = all (select ciclistes.equip
                                 from portar, ciclistes
                                 where portar.ciclista = ciclistes.dorsal
                                   and portar.mallot = mallots.codi
                                 )
```

sub_21) Ports més alts o més baixos de cada etapa. Ha d'eixir l'etapa, nom dels ports, altures i una indicació de si es tracta d'un port 'alt' o 'baix'.

```
select etapa, altura, nom, 'alt'
from ports p1
where altura = (select max(altura)
                from ports p2
                where p2.etapa = p1.etapa)
union
```

```
select etapa, altura, nom, 'baix'
from ports p1
where altura = (select min(altura)
                from ports p2
                where p2.etapa = p1.etapa)
order by 1,4
```

Altra forma (cada etapa, 1 sola línia):

```
select baix.etapa, baix.nom as baix, baix.altura, alt.nom as alt, alt.altura
from ports baix, ports alt
where baix.etapa = alt.etapa
    and baix.altura = (select min(altura)
                      from ports
                      where ports.etapa = baix.etapa)
    and alt.altura = (select max(altura)
                    from ports
                    where ports.etapa = baix.etapa)
order by baix.etapa
```

sub_22) Nom i edat del ciclista més jove.

```
select nom, edat
from ciclistes
where edat <= all (select edat from ciclistes)
```

O bé:

```
select nom, edat
from ciclistes
where edat = (select min(edat) from ciclistes)
```

sub_23) Obteniu el nom dels equips en els quals els seus ciclistes només hagen guanyat ports de categoria 1.

```
select nom
from equips
where nom in ( select equip
               from ciclistes, ports
               where ciclistes.dorsal = ports.ciclista
               and ports.categoria = '1')
and nom not in ( select equip
                from ciclistes, ports
                where ciclistes.dorsal = ports.ciclista
                and ports.categoria <> '1')
```

Compte! La solució següent **no** serveix:

```
select nom
from equips
where '1' = all (select categoria
                from ports, ciclistes
                where ports.ciclista = ciclistes.dorsal
                and ciclistes.equip = equips.nom)
```

Nota: no serveix eixa solució ja que si la subselect no retorna cap fila per a algun equip, per a eixe equip s'avalua com a vertadera l'expressió '1' = all i, per tant, eixe equip sí que és mostrat. És a dir, també mostra els equips que no han guanyat cap port.

És a dir, en este exemple (ciclistes que han guanyat tots els ports de l'etapa 555) hauria de traure 0 ciclistes però els trau tots (100)

```
select count(*)
from ciclistes
where dorsal = all (select ciclista
                   from ports
                   where etapa=555);
```

Si de cas, en la solució de l'exercici, caldria afegir la condició de que han guanyat algun port:

```
and nom in (select equip
            from ciclistes, ports
            where ciclistes.dorsal = ports.ciclista)
```

Altra solució:

```
select nom
from equips
where not exists (select *
                  from ports, ciclistes
                  where ports.ciclista = ciclistes.dorsal
                     and ciclistes.equip = equips.nom
                     and categoria <> '1')
and nom in (select equip
            from ciclistes, ports
            where ciclistes.dorsal = ports.ciclista)
```

sub_24) Obteniu el número d'etapa, la ciutat d'eixida i la ciutat d'arribada de les etapes de més de 190 km. i que tinguen almenys 2 ports.

```
select numero, eixida, arribada
from etapes
where kms > 190
and numero in (select etapa
               from ports
               group by etapa
               having count(*) >= 2)
```

O bé:

```
select numero, eixida, arribada
from etapes
where kms > 190
and (select count(*)
     from ports
     where etapa = etapes.numero) >= 2
```

sub_25) Obteniu el codi i el color del mallot que ha sigut portat per algun ciclista que no ha guanyat cap etapa.

```
select codi, color
from mallots
where codi in (select mallot
              from portar
              where ciclista not in (select ciclista from etapes) )
```

O bé:

```
select codi, color
from mallots, portar
where mallots.codi = portar.mallot
and ciclista not in (select ciclista from etapes)
```

sub_26) Obteniu el dorsal i el nom dels ciclistes que han portat almenys un mallot dels que ha portat el ciclista de dorsal 20.

```
select distinct dorsal, nom
from ciclistes, portar
where ciclistes.dorsal = portar.ciclista
      and portar.mallot in (select mallot
                           from portar
                           where ciclista = 20)
```

o bé:

```
select distinct dorsal, nom
from ciclistes, portar, portar portar20
where ciclistes.dorsal = portar.ciclista
      and portar.mallot = portar20.mallot
      and portar20.ciclista = 20
```

sub_27) Obteniu el dorsal i el nom del ciclista que ha portat durant més quilòmetres un mateix mallot i indiqueu també el color d'eixe mallot.

```
select ciclistes.dorsal, ciclistes.nom, mallots.color
from ciclistes, mallots
where (select sum(etapes.kms)
      from etapes, portar
      where etapes.numero = portar.etapa
            and portar.mallot = mallots.codi
            and portar.ciclista = ciclistes.dorsal
      )
      >= all (select sum(kms)
            from etapes, portar
            where etapes.numero = portar.etapa
            group by portar.ciclista, portar.mallot
      );
```

O bé:

```
select ciclistes.dorsal, ciclistes.nom, mallots.color
from ciclistes, portar, mallots, etapes
where ciclistes.dorsal = portar.ciclista
      and portar.mallot = mallots.codi
      and portar.etapa = etapes.numero
group by ciclistes.dorsal, nom, color
having sum(kms) >= all (select sum(kms)
                      from etapes, portar
                      where etapes.numero = portar.etapa
                      group by portar.ciclista, portar.mallot)
```

sub_28) Nom dels equips tals que **tots** els seus corredors han portat algun mallot o han guanyat algun port.

“Traduït”: equips tals que no existeix un ciclista d’eixe equip que ni ha portat mallot ni ha guanyat port.

```
select nom
from equips
where not exists (select *
                  from ciclistes
                  where ciclistes.equip = equips.nom
                     and ciclistes.dorsal not in (select ciclista from portar)
                     and ciclistes.dorsal not in (select ciclista from ports)
                  )
```

Nota: trau 2 files, en compte d’1 perquè també trau l’equip PDM, que no té ciclistes.

O bé:

“Traduït”: equips tals que no estan en el conjunt dels equips d’aquells ciclistes que ni han portat mallot ni han guanyat port.

```
select nom
from equips
where nom not in (select equip
                  from ciclistes
                  where dorsal not in (select ciclista from portar)
                     and dorsal not in (select ciclista from ports)
                  )
```

Nota: també trau 2 files

O bé:

“Traduït”: equips tals que la quantitat de ciclistes que té eixe equip és la mateixa quantitat de ciclistes d’eixe equip que han portat mallot o que han guanyat port).

```
select nom
from equips
where (select count(*)
      from ciclistes
      where equip = equips.nom)
=
(select count(*)
 from ciclistes
 where ciclistes.equip = equips.nom
    and (ciclistes.dorsal in (select ciclista from portar)
        or
        ciclistes.dorsal in (select ciclista from ports)
    )
)
```

Nota: també trau 2 files, pel mateix motiu.

O bé:

“Traduït”: equips de ciclistes que han guanyat port o han portat mallot. Es compten eixos ciclistes de cada equip i es compara amb la quantitat de ciclistes d'eixe equip.

```
select equip
from ciclistes
where dorsal in (select ciclista from portar)
      or dorsal in (select ciclista from ports)
group by ciclistes.equip
having count(*) = (select count(*)
                  from ciclistes as c2
                  where c2.equip = ciclistes.equip)
```

Nota: esta solució només trau un equip (Castorama)

sub_29) Nom dels ciclistes que han guanyat tots els ports d'una etapa i, a més, han guanyat eixa mateixa etapa.

Solució 1: amb el ALL

```
select ciclistes.nom, etapes.numero
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
      and ciclistes.dorsal = all (select ciclista
                                from ports
                                where ports.etapa = etapes.numero)
order by nom
```

Nota: D'esta forma també traurà ciclistes que han guanyat etapes sense ports ja que el **all** funciona així: si no retorna res la subselect, la condició és certa. Per tant, si volem només els ciclistes que han guanyat etapes i tots els seus ports, caldria afegir que les etapes tinguen ports:

```
and etapes.numero in (select etapa from ports)
```

Solució 2: amb comparació de COUNT

```
select ciclistes.nom, etapes.numero
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
    and ( select count(*)
          from ports
          where etapa = etapes.numero
        )
    =
    ( select count(*)
      from ports
      where etapa = etapes.numero
        and ciclista = ciclistes.dorsal
    )
    and etapes.numero in (select etapa from ports)
order by nom
```

Nota: Si no posàrem la condició última (and etapes.numero in (select etapa from ports)), passaria igual que abans, que també trauria els ciclistes que han guanyat etapes sense ports.

Solució 3: amb doble negació (NOT EXISTS i , en este cas, el <>)

```
select ciclistes.nom, etapes.numero
from ciclistes, etapes
where ciclistes.dorsal = etapes.ciclista
    and not exists (select *
                   from ports
                   where ports.etapa = etapes.numero
                      and ports.ciclista <> ciclistes.dorsal
                  )
    and etapes.numero in (select etapa from ports)
order by nom
```

Nota: ídem de la solució 2.

sub_30) Nom, edat i equip dels ciclistes més joves de cada equip

```
select nom, edat, equip
from ciclistes c1
where edat = (select min(edat)
             from ciclistes c2
             where c2.equip = c1.equip
            );
```


O bé:

```
select nom, edat, equip
from ciclistes c1
where not exists (select *
                  from ciclistes c2
                  where c2.equip = c1.equip
                  and c2.edat < c1.edat);
```

Actualitzacions (INSERT, UPDATE, DELETE)

dml_1) Esborra els equips que no tinguen cap ciclista.

```
delete from equips
where nom not in (select equip from ciclistes);
```

dml_2) Incrementa en un 10% el premi del mallot amb codi MRE.

```
update mallots
set premi = premi * 1.1
where codi = 'MRE';
```

dml_3) Qui va contar els kms de les primeres 10 etapes tenia mal el conta quilòmetres. Cal augmentar un 10 % els kms d'eixes etapes.

```
update etapes
set kms = kms * 1.1
where numero <= 10;
```

dml_4) Si no estiguera posada la clau aliena de *portar* cap a *mallots*, podria donar-se el cas que algun mallot que ha portat algú no figure en la taula de *mallots*. Insereix en la taula de mallots els mallots que han sigut portats però no figuren en la taula de mallots, de forma que tinguen el codi corresponent i tipus “desconegut”.

```
insert into mallots (codi, tipus)
select distinct mallot, 'desconegut'
from portar
where mallot not in (select codi from mallots);
```

dml_5) A última hora s'ha apuntat un nou corredor, el 101, de nom Pep Garcia (de qui no sabem l'edat) que és de l'equip Corremolt, el director del qual és Miquel Marqués. Fes les instruccions SQL necessàries per a actualitzar la base de dades.

```
insert into equips values  
( 'Corremolt', 'Miquel Marqués');
```

```
insert into ciclistes (dorsal, nom, equip) values  
(101, 'Pep Garcia', 'Corremolt');
```

dml_6) A causa d'un terratrèmol, el port 'Portet' ha baixat 100m. Modifica-ho.

```
update ports  
set altura = altura - 100  
where nom = 'Portet'
```

dml_7) Modifica la ciutat d'arribada de l'etapa 3 perquè siga la mateixa que la seua ciutat d'eixida.

```
update etapes  
set arribada = eixida  
where numero = 3
```

dml_8) Duplica el premi de tots els mallots.

```
update mallots  
set premi = premi * 2
```

dml_9) El director d'equip Manuel Sainz no va dir correctament l'edat dels seus corredors. Cal augmentar en 1 l'edat dels seus corredors.

```
update ciclistes  
set edat = edat + 1  
where equip in (select nom  
                 from equips  
                 where director = 'Manuel Sainz')
```

dml_10) Esborra els corredors que no tinguen cap equip assignat o que estiguen assignats a un equip que no figure en la taula d'equips.

```
delete from ciclistes  
where equip is null  
or equip not in (select nom from equips);
```

Vistes (VIEW)

vis_1) Vista ports_alts amb totes les dades dels ports de més de 2000 metres.

```
create view ports_alts as
select * from ports
where altura > 2000
```

vis_2) Insereix un port a la vista de més de 2000 m i altre de menys de 2000.

Deixa? Es veuen a la vista?

```
Insert into ports_alts (port, etapa, km) values ('Port Alt', 1, 3000);
Insert into ports_alts (port, etapa, km) values ('Port Baix', 1, 200);
```

Deixa inserir els 2 (ja que la vista no està definida amb *with check option*), però a la vista només es veu el Port Alt, ja que l'altre no compleix la condició de la vista.

vis_3) Vista guanyadors_etapes_llargues que tinga: número de l'etapa, kms, dorsal i nom del guanyador, l'equip i el director, d'aquelles etapes de més de 100 kms. Els camps de la vista es diran: etapa, kms, dorsal, nom, equip i director.

```
create view guanyadors_etapes_llargues
(etapa, kms, dorsal, nom, equip, director) as
select et.numero, et.kms, et.ciclista, ci.nom, ci.equip, eq.director
from etapes et, ciclistes ci, equips eq
where et.ciclista = ci.dorsal
and ci.equip = eq.nom
and et.kms > 100;
```

vis_4) Inseriu en la vista anterior l'etapa 200 amb el guanyador 1

```
insert into guanyadors_etapes_llargues (etapa, dorsal) values (200, 1);
```

En principi, la vista no és actualitzable (perquè té més d'una taula al from) però sí que deixa (potser depenga de versions de MySQL). Eixe guanyador no es vorà en la vista ja que no és una etapa llarga (no consten els kms) però sí que ha inserit en taula *etapes* l'etapa 200, que l'ha guanyada el dorsal 1.

vis_5) Vista *etapes_altes* que tinga per cada etapa: número d'etapa i quantitat de ports de més de 2000 m. Fes-ho a partir de la vista *ports_alts* que has creat. Els camps de la vista seran: *etapa* i *ports_alts*. És actualitzable? Per què?

```
create view etapes_altes as
select etapa, count(*) as 'ports_alts'
from ports_alts
group by etapa; -- No és actualitzable perquè té una funció d'agregat: count(*)
```

vis_6) Volem mostrar, per a cada equip, el nom de l'equip, quants corredors té i quants mallots ha portat. Però només per als equips que hagen portat més mallots entre els equips amb igual quantitat de corredors. Caldrà mostrar primer els equips amb menys corredors. En cas d'empat, pel nom de l'equip:

equip	q_ciclistes	q_mallots
Castorama	2	4
Jolly Club	2	4
Seguros Amaya	3	7
Gatorade	4	3
ONCE	5	7
TVM	6	7
Artiach	7	16
Gewiss	8	12
Banesto	11	38

De tots els equips amb 5 ciclistes, l'equip ONCE és el que més mallots han portat entre tots els seus ciclistes: 7

Com és una consulta complicada, fes-te primer una vista que tinga el codi de l'equip i quants corredors té cadascun. Després fes la consulta que es demana ajudant-te de la vista que has creat.

- Primer ens creem una vista que guarde cada equip quants corredors té (ja que esta informació l'usarem diverses vegades en la consulta que volem).

```
create view eqqcs_qciclistes as
select equip, count(*) as qciclistes
from ciclistes
group by equip;
```

- Després fem una select ajudant-nos d'eixa vista.

```
select eqqc.equip, eqqc.qciclistes, count(*) as q_mallots
from eqqcs_qciclistes eqqc, ciclistes, portar
where eqqc.equip = ciclistes.equip
and ciclistes.dorsal = portar.ciclista
group by eqqc.equip, eqqc.qciclistes
having count(*) >= all (select count(*)
from eqqcs_qciclistes eqqc2, ciclistes c, portar p
where eqqc2.equip = c.equip
and c.dorsal = p.ciclista
and eqqc2.qciclistes = eqqc.qciclistes
group by eqqc2.equip, eqqc2.qciclistes)
order by 2,1
```