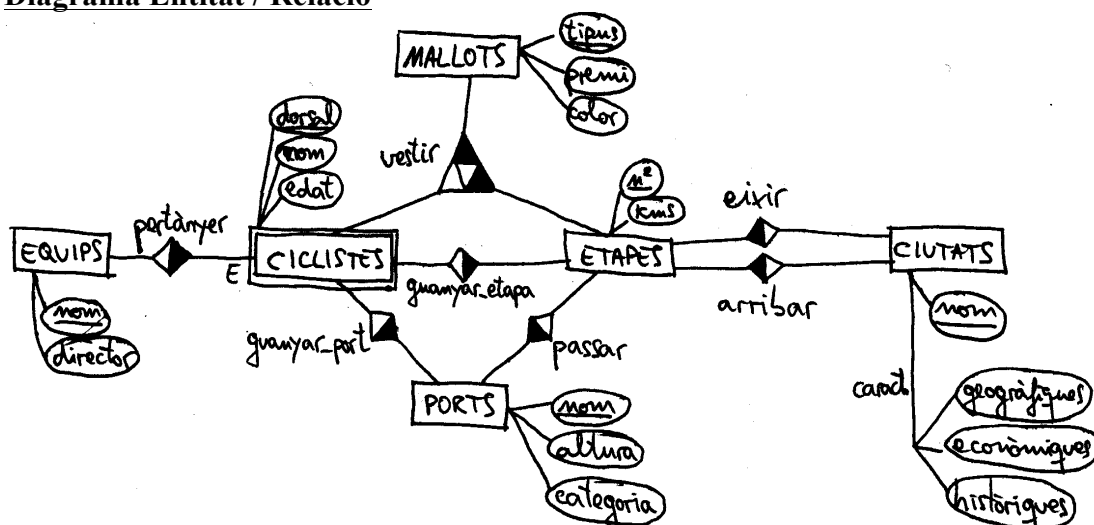


BASE DE DADES *CICLISME* (PER A FER ELS EXERCICIS)

Nota: fet sobre la BD existent *cicli80_2000val.mdb*

Diagrama Entitat / Relació



Nota: en la base de dades no existeix la taula ciutats.

Esquema relacional:

CICLISTES = dorsal + nom + edat + equip

C. Ali: equip → EQUIPS (nom)

EQUIPS = nom + director

ETAPES = numero + kms + eixida + arribada + ciclista

C. Ali: ciclista → CICLISTES (dorsal)

R.I: *ciclista* és el dorsal del ciclista que ha guanyat l'etapa

PORTAR = etapa + mallot + ciclista

C. Ali: etapa → ETAPES (numero)

C. Ali: mallot → MALLOTS (codi)

C. Ali: ciclista → CICLISTES (dorsal)

R.I: esta taula té la informació de quin ciclista vesteix cada mallot al final de cada etapa. És a dir: donada una etapa i un mallot, hi haurà un ciclista guanyador.

MALLOT = codi + tipus + color + premi

PORTS = nom + categoria + etapa + ciclista + altura

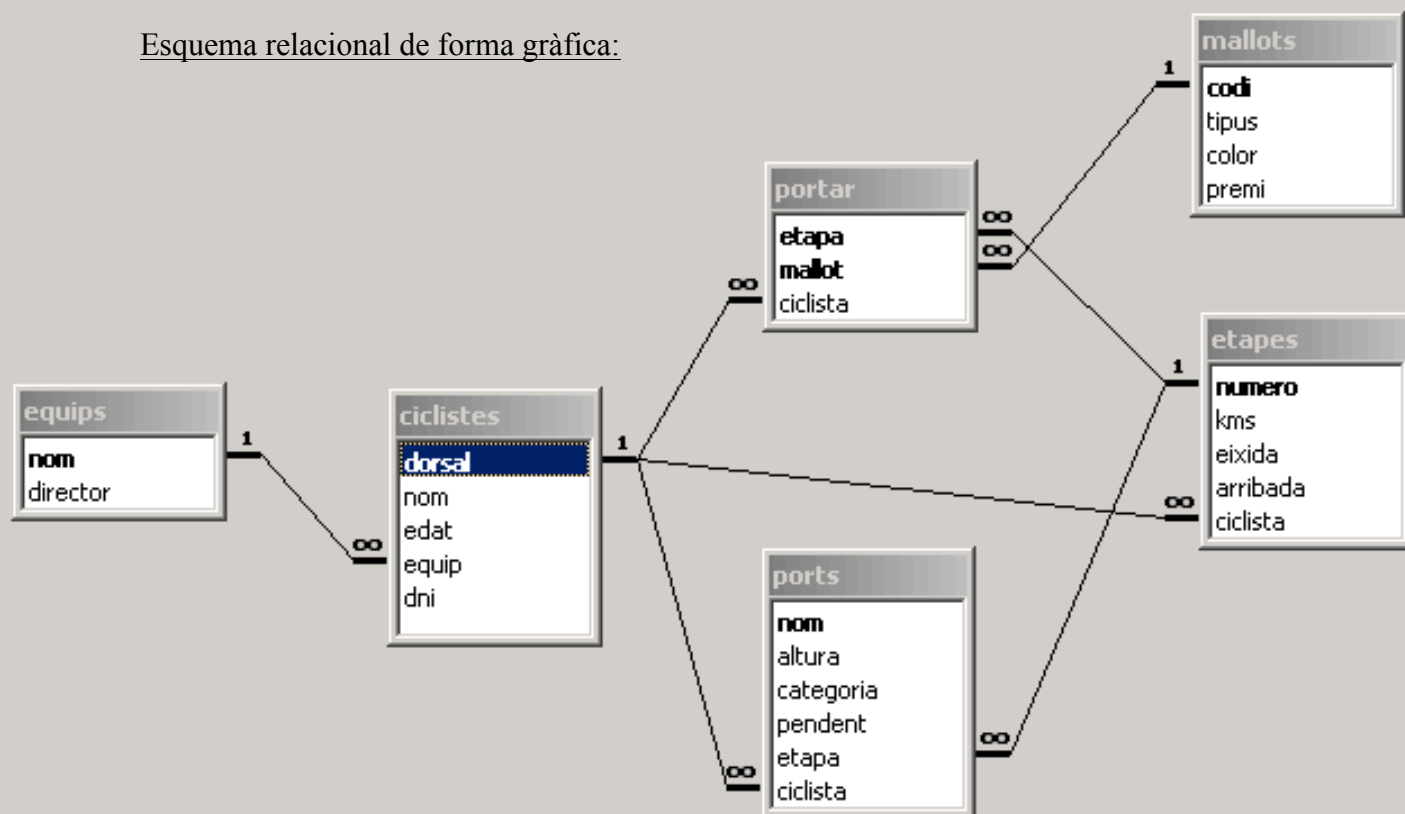
C. Ali: etapa → ETAPES(numero)

C. Ali: ciclista → CICLISTES(dorsal)

R.I: - L'atribut *etapa* indica en quina etapa està eixe port.

- L'atribut *ciclista* indica el ciclista que ha arribat primer al port.

Esquema relacional de forma gràfica:



EXERCICIS BD CICLISME

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

- a1) Obteniu quants ciclistes hi ha
- a2) Mostra l'edat mitjana dels ciclistes
- a3) Obteniu l'altura mínima i màxima dels ports de muntanya.
- a4) Calcula quants kilòmetres té en total la volta ciclista

Falta de dades (valors NULL)

- b1) Obteniu aquells ports que no tenen assignada categoria

Clàusula SELECT

- A1) Selecciona tota la informació dels ports
- A2) Selecciona el nom del port, l'altura en kilòmetres (està guardada en metres) i, al costat, que aparega la paraula 'Km'.

Resultats de consultes

- B1) 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.
- B2) 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".

Clàusula GROUP BY

- C1) Calcula de cada equip l'edat mitjana, màxima, mínima i quants corredors té.
- C2) Calcula quantes etapes ha guanyat cada corredor, ordenat primer per qui ha guanyat més etapes.
- C3) Calcula, de cada categoria de port:
 - L'altura màxima, mínima i mitjana

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

C4) Calcula quantes voltes ha portat un corredor cada mallot

C5) Quants corredors hi ha de cada edat en cada equip?

Clàusula HAVING

D1) Calcula de cada equip amb més de 3 corredors: l'edat mitjana, màxima, mínima i quants corredors té.

D2) 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.

D3) Mostra les categories de ports on l'altura màxima siga igual a la mínima:

D4) Dorsals que han guanyat més d'un port en una mateixa etapa.

Condicions de recerca

E1) Mostra l'etapa i els km de les etapes entre 50 i 100 km

E2) Mostra l'etapa i els km de les altres etapes (<50 i >100)

E3) Obtén el nom dels ciclistes que comencen per 'Al'

E4) Noms d'equips que continguin 'tiac'

E5) Noms de ciutats d'arribada que continguin 'Naran', que tinguin altra lletra al costat, després una 'o' i després més coses.

E6) Noms de ciclistes que continguin 'Induráin' però que no siguin 'Miguel'.

Condicions de recerca compostes: AND, OR i NOT

F1) Selecciona els ports que estiguen entre 1500 i 2000 metres. A més, hauran de complir que no siguin de la categoria 'E' o que tinguin una pendent major que 5.

F2) Ciclistes que comencen per 'Al', 'An' o 'Ar' que tinguin entre 25 i 30 anys.

Ordenació dels resultats d'una consulta (ORDER BY)

G1) 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.

Combinació de resultats (UNION)

H1) Selecciona els dorsals dels corredors de Banesto i també els que han guanyat alguna etapa.

H2) Donada la següent taula:

```
CREATE TABLE amics (  
    nom CHAR(10),  
    cog1 CHAR(10),  
    cog2 CHAR(10))
```

Mostra en una sola columna tots els cognoms dels amics

- a) sense repetir-los
- b) encara que estiguen repetits
- c) els primers cognoms sense repetir però els segons repetits

H3) 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

H4) Seleccioneu 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.

H5) 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.

H6) 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:

| Dorsal | Premis | Tipus |
|--------|--------|---------|
| 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 multitaula

I1) Obtén els noms dels corredors amb el nom dels seus directors.

I2) Mostra entre quines 2 ciutats està cada port

I3) Mostra quants corredors té cada director

I4) 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.

I5) Obtén el nom dels corredors que guanyaren algun port de l'etapa que començava en Igualada.

I6) Per a cada etapa, volem saber la següent informació:

- a. Número
- b. Ciutat origen
- c. Informació dels corredors que duen mallot en eixa etapa:
 - i. Codi del mallot
 - ii. Color del mallot
 - iii. Dorsal del corredor
 - iv. Nom del corredor
 - v. Nom del director del corredor
- d. Informació dels ports de l'etapa:
 - i. Quants ports hi ha en l'etapa
 - ii. L'altura mínima, màxima i mitjana dels ports de l'etapa

I7) Obteniu parelles d'etapes que tenen el 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)

I8) Mostra les dades de les etapes que passen per algun port i que tinguen

l'eixida i l'arribada a la mateixa ciutat.

I9) Mostra el nom del port i nom del ciclista que l'ha guanyat.

I10) Mostra el número d'etapa i l'equip del corredor que l'ha guanyada.

I11) Mostra el número d'etapa i el director de l'equip de qui ha guanyat l'etapa

I12) Mostra de cada etapa:

- a) el número, 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 el nom dels 2 corredors corresponents

I13) Equips que han guanyat alguna etapa portant algun mallot. També cal mostrar el número d'etapa, l'eixida i el mallot.

I14) Obté el nom del ciclista que ha guanyat cada etapa.

I15) 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.

I16) Mostra el codi i el color del mallot que porta cada ciclista (dorsal i nom) en cada etapa.

I17) 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.

I18) Obteniu parelles de nom de ciclista i número d'etapa, tal que eixe ciclista haja guanyat eixa etapa havent portat el mallot de color 'Groc' almenys una vegada.

Subconsultes

J1) Noms de ciclistes que no han guanyat cap etapa

J2) Obteniu número d'etapa i km d'aquelles etapes que no les ha guanyades cap corredor del Banesto.

J3) 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.

J4) Nom de ciclista i voltes que ha portat el mallot groc (MGE) d'aquell ciclista que ha portat més voltes eixe mallot.

J5) Quantitat d'etapes que no tenen ports

J6) Mostra el número d'etapa i quants ports té l'etapa amb més ports.

J7) Mostra el número d'etapa i quants ports té l'etapa amb més ports i la de menys ports.

J8) Nom dels ciclistes que han portat tots els mallots

J9) Qui ha guanyat més etapes i quantes?

J10) Quants ciclistes no han guanyat res (ni etapes, ni ports ni han portat mallot)

J11) Nom del ciclista que han guanyat el port més alt i el del més baix. També cal mostrar el nom del port i l'altura.

J12) Mostra la informació dels ciclistes d'edat màxima de cada equip.

J13) Obteniu el nom dels ports de muntanya que tenen una altura superior a l'altura mitjana de tots els ports.

J14) 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.

J15) Obteniu el dorsal i nom dels ciclistes que han guanyat l'etapa amb el port més alt.

J16) 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.

J17) Obteniu les ciutats que tenen la meta d'alguna etapa però des de les quals no es fa cap eixida.

J18) Obteniu les dades de les etapes que no comencen en la mateixa ciutat en que acaba l'anterior.

J19) Obteniu les dades de les etapes tals que tots els seus ports superen els 1300 metres d'altura.

J20) Obteniu el codi i color dels mallots que només han sigut portats per ciclistes d'un mateix equip.

J21) 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'.

J22) Nom i edat del ciclista més jove.

J23) Obteniu el nom dels equips en els quals els seus ciclistes només hagen guanyat ports de 1a categoria 1.

J24) 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.

J25) Obteniu el codi i el color del mallot que ha sigut portat per algun ciclista que no ha guanyat cap etapa.

J26) Obteniu el dorsal i el nom dels ciclistes que han portat almenys un mallot dels que ha portat el ciclista de dorsal 20.

J27) 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.

J28) Nom dels equips tals que tots els seus corredors han portat algun mallot o han guanyat algun port.

J29) Nom dels ciclistes que han guanyat tots els ports d'una etapa i, a més, han guanyat eixa mateixa etapa.

J30) Nom, edat i equip dels ciclistes més joves de cada equip

J31) Quantes (no quines, sinó quantes) edats diferents tenen els ciclistes?

Exercicis d'actualització

K1) Esborra els equips que no tinguen cap ciclista.

K2) Incrementa en un 10% el premi del mallot amb codi MRE.

K3) Qui va contar els kms de les primeres 10 etapes tenia mal el conta quilòmetres. Cal augmentar un 10 % els kms d'eixes etapes.

K4) Les claus alienes no estan ben posades. Per tant, podria donar-se el cas que algun mallot que ha portat algú, no figure en la taula de mallots. Insereix aquests mallots, de forma que tinguen el codi corresponent i tipus "desconegut".

K5) 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.

K6) A causa d'un terratrèmol, el port 'Portet' ha baixat 100m. Modifica-ho.

K7) Modifica la ciutat d'arribada de l'etapa 3 perquè siga la mateixa que la seua ciutat d'eixida.

K8) Duplica el premi de tots els mallots.

K9) El director d'equip Manuel Sainz no va dir correctament l'edat dels seus corredors. Cal augmentar en 1 l'edat dels seus corredors.

K10) Esborra els corredors que no tinguen cap equip assignat o que estiguen assignats a un equip que no figure en la taula d'equips.

Exercicis de vistes

L1) Vista ports_alts amb totes les dades dels ports de més de 2000 metres.

L2) Insereix un port a ala vista de més de 2000 m i altre de menys de 2000. Deixa? Es veuen a la vista?

L3) Vista guanyadors_etapes_llargues que tinga: número de l'etapa, kilòmetres, dorsal i nom del guanyador, l'equip i el director, d'aquelles etapes de més de 100 kilòmetres.

L4) Inseriu en la vista anterior l'etapa 200 amb el guanyador 1

L5) 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 anterior.

És actualitzable? Per què?

L6) 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 |

Solucions als exercicis:

a1) Obteniu quants ciclistes hi ha

```
Select count(*) AS quant_ciclistes  
From ciclistes
```

a2) Mostra l'edat mitjana dels ciclistes

```
SELECT AVG(edat) AS mitja  
FROM ciclistes
```

a3) Obteniu l'altura mínima i màxima dels ports de muntanya.

```
SELECT MIN(altura) AS mínima, MAX(altura) AS màxima  
FROM ports
```

a4) Calcula quants kilòmetres té en total la volta ciclista

```
Select sum(kms)  
From etapes
```

b1) Obteniu aquells ports que no tenen assignada categoria

```
SELECT *  
FROM ports  
WHERE categoria IS NULL
```

A1) Selecciona tota la informació dels ports

```
SELECT *  
FROM ports
```

A2) 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
```

B1) 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.

Access:

```
SELECT numero, eixida
INTO etapes_ciutat
FROM etapes
WHERE eixida=arribada
```

Mysql:

```
CREATE TABLE etapes_ciutat
SELECT numero, eixida
FROM etapes
WHERE eixida=arribada
```

B2) 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".

```
SELECT color AS "Color del mallot", premi AS "Valor del premi"
INTO premis
FROM mallots
```

C1) 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
```

C2) 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
```


C3) 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
```

C4) Calcula quantes voltes ha portat un corredor cada mallot

```
SELECT ciclista, mallot, count(*)  
FROM portar  
GROUP BY ciclista, mallot
```

C5) Quants corredors hi ha de cada edat en cada equip?

```
SELECT equip, edat, count(*) as quants  
FROM ciclistes  
GROUP BY equip, edat
```

D1) 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
```

D2) 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
```

D3) 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)
```

D4) Dorsals que han guanyat més d'un port en una mateixa etapa.

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

E1) 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)
```

E2) 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)
```

E3) Obtén el nom dels ciclistes que comencen per 'Al'

```
SELECT *
FROM ciclistes
WHERE nom LIKE 'Al%'
```

E4) Noms d'equips que continguin 'tiac'

```
SELECT *
FROM equips
WHERE nom LIKE '%tiac%'
```

E5) 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%'
```

E6) Noms de ciclistes que continguin 'Induráin' però que no siguin 'Miguel'.

```
SELECT *
FROM ciclistes
WHERE nom LIKE '%Induráin%' AND nom NOT LIKE '%Miguel%'
```

F1) 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)
```

F2) 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)
```

O també (almenys en Access):

```
... WHERE nom LIKE 'A[lnr]*'  
AND edat BETWEEN 25 AND 30
```

G1) 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 aparéixer ordenats pel nom del ciclista.

```
SELECT *  
FROM ciclistes  
ORDER BY equip, edat DESC, nom
```

H1) 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
```

H2) Donada la següent taula:

```
CREATE TABLE amics (  
    nom CHAR(10),  
    cog1 CHAR(10),  
    cog2 CHAR(10))
```

Mostra en una sola columna tots els cognoms dels amics

a) sense repetir-los

```
SELECT cog1  
FROM amics
```

UNION

```
SELECT cog2  
FROM amics
```

b) encara que estiguen repetits

```
SELECT cog1  
FROM amics
```

UNION ALL

```
SELECT cog2  
FROM amics
```

c) els primers cognoms sense repetir però els segons repetits

```
SELECT DISTINCT cog1  
FROM amics
```

UNION ALL

```
SELECT cog2  
FROM amics
```

H3) 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 diferència, ja que el cas *b*) no portaria aquells cognoms que apareguen tant en la columna *cog1* com en *cog2*.

H4) Seleccioneu 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
```

H5) 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
```

H6) 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
```


I1) Obten els noms dels corredors amb el nom dels seus directors.

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

I2) Mostra entre quines 2 ciutats està cada port

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

I3) Mostra quants corredors té cada director

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

I4) 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
```

I5) Obten 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'
```

I6) Per a cada etapa, volem saber la següent informació:

- e. Número
- f. Ciutat origen
- g. Informació dels corredors que duen mallot en eixa etapa:
 - i. Codi del mallot
 - ii. Color del mallot
 - iii. Dorsal del corredor
 - iv. Nom del corredor
 - v. Nom del director del corredor
- h. Informació dels ports de l'etapa:
 - i. Quants ports hi ha en l'etapa
 - ii. 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 et.numero, et.eixida, po.mallot, ma.color, po.ciclista, ci.nom,
eq.director
```

I7) 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
```

I8) Mostra les dades de les etapes que passen per algun port i que tinguen 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
```

I9) 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
```

I10) 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
```

I11) 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
```

I12) Mostra de cada etapa:

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

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

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

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

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

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

d) Mostra a) més el nom dels 2 corredors corresponents

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 etapes.numero, etapes.eixida,  
       etapes.ciclista as guanyador,  
       guanyadors.nom as nom_guanyador,  
       portar.ciclista as groc, grocs.nom as nom_groc  
from etapes, ciclistes guanyadors, portar, mallots,  
       ciclistes grocs  
where etapes.numero = portar.etapa  
       and portar.mallot = mallots.codi  
       and mallots.color = 'groc'  
       and etapes.ciclista = guanyadors.dorsal  
       and portar.ciclista = grocs.dorsal  
order by numero
```

I13) Equips que han guanyat alguna etapa portant algun mallot. També cal mostrar el número d'etapa, l'eixida i el mallot.

```
select ciclistes.equip, 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
```

I14) Obté el nom del ciclista que ha guanyat cada etapa.

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

I15) 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 equip, count(*)  
from etapes, ciclistes  
where etapes.ciclista = ciclistes.dorsal  
group by equip  
order by 2 desc, 1
```

I16) 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
```

I17) 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

```

I18) Obteniu parelles de nom de ciclista i número d'etapa, tal que eixe ciclista haja guanyat eixa etapa havent portat el mallot de 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 reflecteix "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)

```

J1) Noms de ciclistes que no han guanyat cap etapa

```

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

```

J2) 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'

```

J3) 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)

```

J4) 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)
```

J5) Quantitat d'etapes que no tenen ports

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

J6) 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)
```

J7) 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)
```

J8) 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)
```

J9) Qui ha guanyat més etapes i quantes?

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

J10) 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)
```

J11) 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)  
)
```

J12) 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)
```

J13) 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)
```

J14) 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
                           )
```

J15) 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))
```

J16) 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
```

J17) 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)
```

J18) Obteniu les dades de les etapes que no comencen en la mateixa ciutat en que acaba l'anterior.

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

La select anterior també mostra l'etapa 1. Aquesta select següent, no:

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

J19) 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
```


J20) 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.equipo = all (select ciclistes.equipo
                           from portar, ciclistes
                           where portar.ciclista = ciclistes.dorsal
                             and portar.mallot = mallots.codi
                           )
```

J21) 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
```

J22) 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)
```

J23) 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.equipo = 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, la següent select (sense sentit) deuria de traure 0 equips, però en trau 22 (tots):

```

select count(*)
  from equips
 where 'a' = all (select 'b'
                  from ciclistes
                  where 1=2)

```

Si de cas, 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)

```

J24) Obteniu el número d'etapa, la ciutat d'eixida i la ciutat d'arribada de les etapes de més de 190 km. i que tinguin 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
```

J25) 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)
```

J26) 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, portar20
WHERE ciclistes.dorsal = portar.ciclista
AND portar.mallot = portar20.mallot
and portar20.ciclista = 20
```

J27) 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)
```

J28) 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 portar)
                  )
```

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)

J29) 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'aquesta 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 corredors que han guanyat etapes i tots els seus ports, caldria afegir:

```
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.

J30) Nom, edat i equip dels ciclistes més joves de cada equip

Nota: es pot fer sense exists, però amb exists és així:

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

J31) Quantes (no quines, sinó quantes) edats diferents tenen els ciclistes?

```
select count(*)
from ciclistes
where dorsal = (select min(dorsal)
                from ciclistes ciclistes2
                where ciclistes2.edat = ciclistes.edat)
```

K1) Esborra els equips que no tinguen cap ciclista.

```
DELETE FROM equips
WHERE nom NOT IN (SELECT equip FROM ciclistes)
```

K2) Incrementa en un 10% el premi del mallot amb codi MRE.

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

K3) 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
```

K4) Les claus alienes no estan ben posades. Per tant, podria donar-se el cas que algun mallot que ha portat algú, no figure en la taula de mallots. Insereix aquests mallots, de forma que tinguin 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)
```

K5) 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');
```

go

```
INSERT INTO ciclistes (dorsal, nom, equip)
VALUES (101, 'Pep Garcia', 'Corremolt');
```

K6) A causa d'un terratrèmol, el port 'Portet' ha baixat 100m. Modifica-ho.

```
UPDATE ports
SET altura = altura - 100
WHERE nom = 'Portet'
```

K7) 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
```

K8) Duplica el premi de tots els mallots.

```
UPDATE mallots
SET premi = premi * 2
```

K9) 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')
```

K10) Esborra els corredors que no tinguin 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)
```

L1) 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
```

L2) 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.

L3) Vista guanyadors_etapes_llargues que tinga: número de l'etapa, kilòmetres, dorsal i nom del guanyador, l'equip i el director, d'aquelles etapes de més de 100 kilòmetres.

```
create view guanyadors_etapes_llargues as
```

```
select etapes.numero, etapes.kms, etapes.dorsal, ciclistes.nom, equips.director
from etapes, ciclistes, equips
where etapes.ciclista = ciclistes.dorsal
and ciclistes.equip = equips.nom
and etapes.kms > 100
```

L4) Inserir en la vista anterior l'etapa 200 amb el guanyador 1

```
insert into guanyadors_etapes_llargues (numero, dorsal) values (200, 1)
```

NO deixa perquè la vista no és actualitzable (té més d'una taula al FROM)

L5) 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 anterior.

És actualitzable? Per què?

```
create view etapes_llargues as
```

```
select etapa, count(*)
from ports_alts
group by etapa
```

No és actualitzable perquè té una funció d'agregat: count(*)

L6) 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 |

```

create view equips2 as
  select equip, count(*) as q_ciclistes
    from ciclistes
   group by equip
;
select equips2.equip, equips2.q_ciclistes, count(*) as q_mallots
  from equips2, ciclistes, portar
 where equips2.equip = ciclistes.equip
   and ciclistes.dorsal = portar.ciclista
 group by equips2.equip, equips2.q_ciclistes
 having count(*) >= all (select count(*)
                        from equips2 e2, ciclistes c, portar p
                       where e2.equip = c.equip
                         and c.dorsal = p.ciclista
                         and e2.q_ciclistes = equips2.q_ciclistes
                      group by e2.equip, e2.q_ciclistes)
 order by 2,1

```