

Ejercicio SQL

Dado el siguiente modelo relacional, se pide:

EQUIPO(codigoequipo, nombreequipo, director)
CICLISTA(codigociclista, nombreciclista, fechanacimiento, peso, *Codigoequipo*)
ETAPA(numetapa, km, salida, llegada, *codigociclista*)
TRAMO(codigotramo, nombretramo, altura, categoria, *numetapa*, *codigociclista*)
CAMISETA(codigocamiseta, tipo, premio, color)
LLEVA(*numetapa*, *codigocamiseta*, *codigociclista*)

Nota: Las claves foráneas en los modelos relacionales presentes en este documento se representan en cursiva y negrita.

Equipo: contiene los datos de los distintos equipos: nombre (nomequipo) y nombre de su director (director).

Ciclista: contiene los datos de los ciclistas que componen los distintos equipos: número del dorsal (dorsal), nombre del ciclista (nombre), edad del ciclista (edad) y nombre del equipo al que pertenece (nomequipo).

Etapas: contiene los datos de las etapas que componen la vuelta ciclista: número de la etapa (numetapa) (las etapas se numeran consecutivamente: 1, 2, ...), kilómetros que tiene la etapa (kms), nombre de la población de donde sale la etapa (salida), nombre de la población donde está la meta de la etapa (llegada) y número del dorsal del ciclista que ha ganado la etapa (dorsal). Los atributos salida y llegada están definidos sobre el mismo dominio.

Tramo: contiene los datos de los tramos de montaña que visita la vuelta ciclista: nombre del tramo (nomtramo), altura máxima del tramo (altura), categoría del tramo: primera, especial, etc. (categoría), porcentaje que indica la pendiente media del tramo (pendiente), número de la etapa donde se sube el tramo (numetapa) y número del dorsal que ha ganado el tramo al pasar en primera posición (dorsal).

Camiseta: contiene los datos de los premios que se otorgan mediante las distintas camisetas: código de la camiseta (código), clasificación que premia esa camiseta: general, montaña, etc. (tipo), color de la camiseta asociada (color) e importe del premio que corresponde al ciclista que termine la vuelta llevando la camiseta (premio).

Lleva: contiene la información sobre qué ciclistas (dorsal) han llevado cada camiseta (código) en cada una de las etapas (numetapa)

1. Nombre y el equipo de aquellos corredores menores de 30 años que han ganado alguna etapa.

Solución 1:

```
SELECT DISTINCT nombre, nomequipo
FROM Ciclista C, Etapa E
WHERE edad < 30
AND C.dorsal = E.dorsal;
```

2. Nombre y equipo de los corredores mayores de 35 años que han ganado algún tramo.

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

Solución 1:

```
SELECT DISTINCT nombre, nomequipo
  FROM Ciclista C, Tramo P
 WHERE C.dorsal = P.dorsal
    AND edad > 35;
```

3. Datos de las etapas que pasan por algún tramo de montaña y que tienen salida y llegada en la misma población.

Solución 1:

```
SELECT DISTINCT E.numetapa, E.kms,
                E.salida, E.llegada, E.dorsal
  FROM Etapa E, Tramo P
 WHERE E.numetapa = P.numetapa
    AND E.salida = E.llegada;
```

4. Poblaciones que tienen la meta de alguna etapa, pero desde las que no se realiza ninguna salida

Solución 1:

```
SELECT llegada FROM Etapa
MINUS
SELECT salida FROM Etapa;
```

Solución 2:

```
SELECT DISTINCT llegada
  FROM Etapa
 WHERE llegada NOT IN (SELECT salida FROM Etapa);
```

5. Nombre y el equipo de los ciclistas que han ganado alguna etapa llevando la camiseta de color amarillo, mostrando también el número de etapa.

```
SELECT C.nombre, C.nomequipo, E.numetapa
  FROM Ciclista C, Etapa E, Lleva LL, Camiseta M
 WHERE E.dorsal = LL.dorsal
    AND E.numetapa = LL.numetapa
    AND LL.código = M.código
    AND M.color = 'amarillo'
    AND E.dorsal = C.dorsal;
```

6. Poblaciones de salida y de llegada de las etapas donde se encuentran tramos con altura superior a 1300 metros.

```
SELECT DISTINCT salida, llegada
  FROM Etapa E, Tramo P
 WHERE E.numetapa = P.numetapa
    AND altura > 1300;
```

7. Número de las etapas que tienen algunos tramos de montaña, indicando cuántos tiene cada una de ellas.

```
SELECT numetapa, COUNT(*) AS numero_tramos
  FROM Tramo
GROUP BY numetapa;
```

8. Nombre y edad de los ciclistas que han llevado dos o más camisetas en una misma etapa

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
SELECT DISTINCT LL.dorsal, C.nombre, C.edad
FROM Ciclista C, Lleva LL
WHERE C.dorsal = LL.dorsal
GROUP BY LL.numetapa, LL.dorsal, C.nombre, C.edad
HAVING COUNT(*) >= 2;
```

9. Obtener el nombre y el equipo de los ciclistas que han llevado alguna camiseta o que han ganado algún tramo.

Solución 1:

```
SELECT C.nombre, C.nomequipo
FROM Ciclista C, Lleva LL
WHERE C.dorsal = LL.dorsal
UNION
SELECT C.nombre, C.nomequipo
FROM Ciclista C, Tramo P
WHERE C.dorsal = P.dorsal;
```

Solución 2:

```
SELECT nombre, nomequipo
FROM Ciclista
WHERE dorsal IN (SELECT dorsal FROM Lleva)
OR dorsal IN (SELECT dorsal FROM Tramo);
```

10. Obtener los números de las etapas que no tienen tramos de montaña.

Solución 1:

```
SELECT numetapa FROM Etapa
MINUS
SELECT numetapa FROM Tramo;
```

Solución 2:

```
SELECT numetapa
FROM Etapa
WHERE numetapa NOT IN (SELECT numetapa FROM Tramo);
```

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

```
SELECT AVG(C.edad) AS media_edad_ganadores
FROM Ciclista C, Etapa E
WHERE C.dorsal = E.dorsal;
```

12. Obtener el nombre de los tramos de montaña que tienen una altura superior a la altura media de todos los tramos.

```
SELECT nompuerto FROM Tramo
WHERE altura > (SELECT AVG(altura) FROM Tramo);
```

13. Obtener las poblaciones de salida y de llegada de las etapas donde se encuentran los puertos con mayor pendiente.

Solución 1:

```
SELECT DISTINCT salida, llegada
FROM Etapa E, Tramo P
WHERE E.numetapa = P.numetapa
AND P.pendiente = (SELECT MAX(pendiente) FROM Tramo);
```

Solución 2:

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
SELECT DISTINCT E.salida, E.llegada
FROM Etapa E, Tramo P
WHERE E.numetapa = P.numetapa
AND P.pendiente >= ALL (SELECT pendiente FROM Tramo);
```

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

Solución 1:

```
SELECT DISTINCT C.dorsal, C.nombre
FROM Ciclista C, tramo P
WHERE C.dorsal = P.dorsal
AND P.altura = (SELECT MAX(altura) FROM tramo);
```

Solución 2:

```
SELECT DISTINCT C.dorsal, C.nombre
FROM tramo P, Ciclista C
WHERE P.dorsal = C.dorsal
AND P.altura >= ALL (SELECT altura FROM tramo);
```

15. Obtener los datos de las etapas cuyos puertos (todos) superan los 1300 metros de altura.

Solución 1:

```
SELECT E.*
FROM Tramo P, Etapa E
WHERE P.numetapa = E.numetapa
AND P.altura > 1300
MINUS
SELECT E.*
FROM Tramo P, Etapa E
WHERE P.numetapa = E.numetapa
AND P.altura <= 1300;
```

Solución 2:

```
SELECT DISTINCT E.*
FROM Tramo P, Etapa E
WHERE P.numetapa = E.numetapa
AND P.altura > 1300
AND E.numetapa NOT IN (SELECT numetapa
FROM Tramo WHERE altura <= 1300);
```

16. Obtener el nombre del equipo y el director del ciclista que ha ganado la etapa más larga.

Solución 1:

```
SELECT director, Q.nomequipo
FROM Etapa E, Ciclista C, Equipo Q
WHERE E.dorsal = C.dorsal
AND C.nomequipo = Q.nomequipo
AND kms = (SELECT MAX(kms) FROM Etapa);
```

Solución 2:

```
SELECT TOP(1) director, Q.nomequipo
FROM Etapa E, Ciclista C, Equipo Q
```

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
WHERE E.dorsal = C.dorsal
AND C.nomequipo = Q.nomequipo
ORDER BY kms DESC;
```

17. Obtener el dorsal y el nombre de los ciclistas que hayan ganado alguna etapa, mostrando también el número de etapas que han ganado.

```
SELECT E.dorsal, C.nombre, COUNT(E.dorsal) AS etapas_ganadas
FROM Etapa E, Ciclista C
WHERE C.dorsal = E.dorsal
GROUP BY E.dorsal, C.nombre;
```

18. Obtener el nombre de los ciclistas que pertenecen a un equipo de más de cinco ciclistas y que han ganado alguna etapa, indicando también cuántas etapas han ganado.

```
SELECT C.dorsal, C.nombre, COUNT(*) AS num_etapas_ganadas
FROM Ciclista C, Etapa E
WHERE C.dorsal = E.dorsal
AND C.nomequipo IN (SELECT nomequipo
                     FROM Ciclista
                     GROUP BY nomequipo
                     HAVING COUNT(*) > 5)
GROUP BY C.dorsal, C.nombre;
```

19. Nombre y equipo de los ciclistas que han llevado alguna vez camiseta de color amarillo, indicando durante cuántas etapas lo han llevado.

```
SELECT DISTINCT C.nombre, C.nomequipo, COUNT(*) AS Num_veces
FROM Ciclista C, Lleva L, Camiseta M
WHERE C.dorsal = L.dorsal
AND L.código = M.código
AND color = 'amarillo'
GROUP BY C.nombre, C.nomequipo;
```

20. Por cada equipo, color de las camisetas que han llevado sus ciclistas.

```
SELECT nomequipo, color
FROM camiseta M, Lleva L, Ciclista C
WHERE M.código = L.código
AND L.dorsal = C.dorsal
GROUP BY nomequipo, color;
```

21. Nombre y equipo del ganador de la vuelta (es decir, el que ha lucido la camiseta amarilla en la última etapa).

```
SELECT C.nombre, C.nomequipo
FROM Ciclista AS C, Lleva AS L, Camiseta AS M
WHERE C.dorsal = L.dorsal
AND M.código = L.código
AND color = 'amarillo'
AND L.numetapa = (SELECT MAX(numetapa) FROM Etapa);
```

22. Nombre de los equipos que no han llevado la camiseta roja.

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
SELECT DISTINCT nomequipo
FROM Equipo
WHERE nomequipo NOT IN (SELECT DISTINCT nomequipo
                        FROM Ciclista C, Lleva L, Camiseta M
                        WHERE C.dorsal = L.dorsal
                        AND L.código = M.código
                        AND color = 'rojo');
```

23. Nombre y dorsal de los ciclistas mayores de 30 años que han ganado algún tramo, junto con el número de tramos que han ganado.

```
SELECT C.dorsal, nombre, edad, COUNT(*) AS numeroDeTramos
FROM Ciclista AS C, Tramo AS P
WHERE C.dorsal = P.dorsal
AND edad > 30
GROUP BY C.dorsal, nombre, edad;
```

24. Nombre y director de los equipos que, en alguna etapa, sus ciclistas han llevado tres o más camisetas.

```
SELECT DISTINCT E.nomequipo, E.director
FROM Ciclista AS C, Equipo AS E, Lleva AS L
WHERE C.dorsal = L.dorsal
AND C.nomequipo = E.nomequipo
GROUP BY L.numetapa, E.nomequipo, E.director
HAVING COUNT(*) >= 3;
```

25. Nombre de los equipos que sólo tienen ciclistas menores de 28 años.

```
SELECT nomequipo
FROM Equipo
WHERE nomequipo IN (SELECT nomequipo
                    FROM Ciclista WHERE edad < 28)
AND nomequipo NOT IN (SELECT nomequipo
                     FROM Ciclista WHERE edad >= 28);
```

26. Dorsal y nombre del ciclista que ha llevado la camiseta verde durante más etapas.

Solución 1:

```
SELECT C.dorsal, C.nombre
FROM Ciclista AS C, Lleva AS L, Camiseta AS M
WHERE C.dorsal = L.dorsal
AND M.código = L.código
AND M.color = 'verde'
GROUP BY C.dorsal, C.nombre
HAVING COUNT(*) >= ALL (SELECT COUNT(*) AS veces
                      FROM Lleva LL, Camiseta MA
                      WHERE LL.código = MA.código
                      AND MA.color = 'verde '
                      GROUP BY LL.dorsal);
```

Solución 2:

```
SELECT TOP(1) dorsal, nombre
FROM Ciclista AS C, Lleva AS L, Maillot AS M
WHERE C.dorsal = L.dorsal
```

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
AND M.codigo = L.codigo
AND M.color = 'amarillo'
GROUP BY C.dorsal, C.nombre
ORDER BY COUNT(*) DESC;
```

27. Obtener todos los datos de aquellos ciclistas de los que se desconozca su edad.

```
SELECT * FROM Ciclista WHERE FechaNacimiento IS NULL;
```

28. Obtener de las camisetas, el tipo y el premio en dólares (está en bolívares) de aquellas cuyo premio supere los \$5000.

```
SELECT tipo, premio / 80000 premioEnDolares
FROM Camiseta
WHERE premio / 80000 > 5000;
```

29. Obtener el número de las etapas donde el nombre de la ciudad de llegada tenga por segunda letra una B o donde el nombre de la ciudad de salida lleve dos o más E.

```
SELECT NumEtapa
FROM Etapa
WHERE llegada LIKE '_B%'
OR salida LIKE '%E%E%';
```

30. Obtener el nombre y la fecha de nacimiento de los ciclistas que pertenezcan a equipos cuyo nombre contenga la cadena 100%.

```
SELECT nombreciclista, fechaNacimiento
FROM Ciclista, Equipo
WHERE nombreequipo LIKE '%100\%'
AND equipo.codigoequipo = ciclista.codigoequipo;
```

Posible estructura

```
create database ciclistas
use ciclistas
```

```
CREATE TABLE EQUIPO (
    codigoequipo INT PRIMARY KEY,
    nombreequipo VARCHAR(50),
    director VARCHAR(50)
);
CREATE TABLE CICLISTA (
    codigociclista INT PRIMARY KEY,
    nombreciclista VARCHAR(50),
    fechanacimiento DATE,
    peso FLOAT,
    Codigoequipo INT,
    FOREIGN KEY (Codigoequipo) REFERENCES EQUIPO(codigoequipo)
);
CREATE TABLE ETAPA (
    numetapa INT PRIMARY KEY,
    km FLOAT,
    salida VARCHAR(50),
    llegada VARCHAR(50),
    codigociclista INT,
    FOREIGN KEY (codigociclista) REFERENCES CICLISTA(codigociclista)
);
CREATE TABLE TRAMO (
```

Ingeniería de Datos I (3.4.209)	Ing. Charles Maldonado
septiembre de 2022	Lic. Gustavo Escandell

Ejercicio SQL

```
codigotramo INT PRIMARY KEY,  
nombretramo VARCHAR(50),  
altura FLOAT,  
categoria VARCHAR(50),  
numetapa INT,  
codigociclista INT,  
FOREIGN KEY (numetapa) REFERENCES ETAPA(numetapa),  
FOREIGN KEY (codigociclista) REFERENCES CICLISTA(codigociclista)  
);  
CREATE TABLE CAMISETA (  
    codigocamiseta INT PRIMARY KEY,  
    tipo VARCHAR(50),  
    premio VARCHAR(50),  
    color VARCHAR(50)  
);  
CREATE TABLE LLEVA (  
    numetapa INT,  
    codigocamiseta INT,  
    codigociclista INT,  
    PRIMARY KEY (numetapa, codigocamiseta, codigociclista),  
    FOREIGN KEY (numetapa) REFERENCES ETAPA(numetapa),  
    FOREIGN KEY (codigocamiseta) REFERENCES CAMISETA(codigocamiseta),  
    FOREIGN KEY (codigociclista) REFERENCES CICLISTA(codigociclista)  
);
```