

## EXAMEN SQL ENERO 2023

[0.5] Lista de equipos con un presupuesto mayor de 12000000.

```
SELECT *  
FROM equipo  
WHERE _____
```

[0.8] Lista de partidos en los que el equipo 'R.C. Celta' jugó como local. Muestra para cada partido, la jornada, el código del adversario y la diferencia de goles marcados menos encajados.

```
SELECT p.jornada _____  
FROM equipo e _____  
WHERE _____  
AND e.nombre = 'R.C.Celta'
```

[0.8] Muestra los partidos en los que el equipo 'R.C.Celta' jugó contra el equipo 'R.C.Deportivo'. Para cada partido muestra la jornada, los códigos de los equipos local y visitante y el resultado con el formato 'goleslocal - golesvisitante'. Ordena el resultado de forma ascendente por jornada.

```
SELECT p.jornada, p.equipo_local, p.equipo_visitante,  
cast (_____) || '-' _____ AS resultado  
FROM equipo l, _____  
WHERE _____  
_____  
AND l.nombre IN ('R.C.Celta', 'R.C.Deportivo')  
AND ....(aquí había algo más pero IDK)
```

[0.9] Obtén el número de goles anotados en cada una de las jornadas. Ordena el resultado de forma inversa por dicho número de goles.

```
SELECT _____  
FROM _____  
_____  
_____
```

[1,1] Obtén los equipos que, jugando como local, han marcado al menos el mismo número de goles que han encajado. Muestra para cada equipo, su nombre y los goles marcados y encajados. Ordena el resultado de forma descendente por la diferencia de goles marcados menos encajados.

```
SELECT e.nombre _____  
_____  
FROM _____  
WHERE _____  
_____
```

\_\_\_\_\_  
\_\_\_\_\_  
[1.2] Obtén el nombre del equipo que en las primeras 3 jornadas ha marcado más goles como visitante.

```
SELECT e.nombre
FROM equipo e _____
WHERE _____
GROUP BY _____
HAVING _____ (SELECT max(goles_visitantes)
FROM equipo e1 , partido p
WHERE e1.idequipo = p.equipoVisitante
group by idequipo)
```

[1.3] Obtén los datos de los equipos que no han perdido ningún partido.

```
SELECT *
FROM equipo e
WHERE _____ (SELECT *
FROM _____
WHERE _____)
_____ (SELECT *
FROM _____
WHERE _____)
```

[1.3] Obtén el nombre de cada equipo y el número de partidos que ha ganado. Ordena el resultado de forma descendente por el número de partidos ganados.

```
SELECT codigo, nombre _____
FROM (
    SELECT e.codigo, e.nombre
    FROM equipo e _____
    WHERE _____
    _____
    SELECT _____
    FROM _____
    WHERE _____ Inserta el texto aquí
    _____) AS partidos
ORDER BY _____
```

[1.3] Obtén un informe con la clasificación de la liga, en la que muestres el número de partidos ganados, empatados y perdidos, goles a favor y goles en contra y puntos (3 por

victoria y 1 por empate). Ordena los equipos por el número de puntos de forma descendente.

```

SELECT equipos.nombre,
       COALESCE (ganados_local.partidos, 0) +
       COALESCE (ganados_visitante.partidos, 0) AS ganados,
       ... (aquí había algo más pero IDK)
       COALESCE (perdidos_local.gc, 0) +
       COALESCE (ganados_visitante.gc, 0) +
       COALESCE (empatados_visitante.gc, 0) +
       COALESCE (perdidos_visitante.gc, 0) AS gc,

_____,
_____,
_____,
_____) AS puntos
FROM (SELECT codigo, nombre FROM equipo) AS equipos
LEFT JOIN
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as ganados_local
_____,
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as empatados_local
_____,
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as perdidos_local
_____,
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as ganados_visitante
_____,
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as empatados_visitante
_____,
(SELECT e.codigo, e.nombre _____
_____
FROM equipo e _____
WHERE _____
GROUP BY _____) as perdidos_visitante
_____

```

```
FROM equipo e _____  
WHERE _____  
GROUP BY _____) as perdidos_visitante  
_____  
...(IDK si seguía o no, pero esa es la esencia jejej)
```