

GBD04- Solución a la TAREA

REALIZACIÓN DE CONSULTAS.

A partir de la base de datos **campeonato**, cuyo script se proporciona en la tarea

Realizar las siguientes consultas:

1.- Listado con el código, nombre, fecha de inscripción y cuota de los concursantes masculinos con cuota_inscri entre 50 y 100 euros (incluidos límites), año de inscripción 2019 y mes entre marzo y septiembre (incluidos). Utiliza la función `year()` y `month()` para ello. Ordena el resultado de mayor a menor cuota. El encabezado debe ser 'codigo, jugador, fecha, cuota.

Haz dos versiones:

a) Usando **BETWEEN** donde sea posible **b)** Sin usar **BETWEEN**..

2.- Listado con el código, nombre, comunidad y año de fundación de los equipos de concursantes que son de Andalucía, Murcia o Galicia y su año de fundación es el 2019. Ordena por comunidad alfabéticamente.

Redacta dos versiones:

a) Usando **IN()** donde sea posible **b)** Sin usar **IN()** .

3.- Listado con el código, nombre, dificultad y megusta del o los juegos con mas megusta. (Con subconsulta).

4.- Listado con todos los datos de los juegos sin valor (valor nulo) en megusta y nombre de su equipo organizador.

5.- Listado con el nombre, dificultad, megusta y código de quipo de los juegos organizados por el mismo equipo que el del juego de nombre 'Elvenar'. (Con subconsulta)

6.- Modifica la consulta 5. para que también se muestre el nombre y comunidad del equipo.

7.- Para los concursantes que participan en juegos, listar el código y nombre de concursantes y código del juego en el que participan, mostrando los puntos y fecha de inicio. Ordena el listado alfabéticamente por código de concursante.

8.- Lista donde aparezcan los códigos y nombre de todos los concursantes, nombre de sus equipos y nombre de los juegos en los que participan, así como los puntos que llevan en cada juego. Los concursantes que no participan en juegos aparecerán también en la lista y en el lugar del juego aparecerá “sin juego” y en horas un cero. Debes poner un encabezado apropiado de manera que no aparezcan nulos en el listado.

9.- Para cada concursante que participa en juegos, su código, nombre y la media de sus puntos. Redondear la media de puntos con dos decimales, (usa la función **ROUND()**) y ordenar de más a menos puntos.

10.- Modifica la consulta 9. para que aparezca el código y nombre del ídolo de cada concursante. Limita el listado a los 5 concursantes con la media más alta.

11.- Listado de los nombres de juegos en los que aparezca la palabra “Empire”, indicando los megusta que tiene, dificultad y total de concursantes que participan en el juego.

12.- Listado con el código de juego y total de concursantes que participan en cada juego. En el listado mostrar solo los juegos en los que participan 2 o más concursantes. Ordenar de más a menos participación.

Nota para el tutor:

La SOLUCIÓN propuesta a continuación es orientativa. Algunas consultas pueden resolverse de distintas formas.

SOLUCIÓN EJERCICIO 1.

a) Usando BETWEEN donde sea posible

```
SELECT cdconcur as 'codigo', nombre 'jugador', fecha_inscri as 'fecha', cuota_inscri as cuota
FROM concursante
where sexo='h' and cuota_inscri between 50 and 100 and year(fecha_inscri)=2019 and month(fecha_inscri) between 3
and 9
order by cuota_inscri desc;
```

b) Sin usar BETWEEN.

```
SELECT cdconcur as 'codigo', nombre 'jugador', fecha_inscri as 'fecha', cuota_inscri as cuota
FROM concursante
where sexo='h' and cuota_inscri >= 50 and cuota_inscri <= 100 and year(fecha_inscri)=2019 and month(fecha_inscri) >=
3 and month(fecha_inscri) <= 9
order by cuota_inscri desc;
```

SOLUCIÓN EJERCICIO 2.

a) Usando IN() donde sea posible

```
SELECT cdequipo, nombre, comunidad, anio_funda
FROM equipo
where comunidad in ('Andalucía', 'Murcia', 'Galicia') and anio_funda=2019
order by comunidad;
```

b) Sin usar IN().

```
SELECT cdequipo, nombre, comunidad, anio_funda
FROM equipo
where (comunidad = 'Andalucia' OR comunidad = 'Murcia' OR comunidad = 'Galicia')
and anio_funda=2019
order by comunidad;
```

SOLUCIÓN EJERCICIO 3.

```
SELECT cdjuego, nombre, dificultad, megusta
FROM juego
WHERE megusta = (SELECT MAX(megusta) FROM juego);
```

SOLUCIÓN EJERCICIO 4.

```
SELECT j.*, e.nombre 'equipo'
FROM juego j
INNER JOIN equipo e ON j.cdequipo=e.cdequipo
WHERE j.megusta IS NULL;
```

SOLUCIÓN EJERCICIO 5.

```
SELECT nombre, dificultad, megusta, cdequipo
FROM juego
WHERE cdequipo = (SELECT cdequipo FROM juego WHERE nombre='Elvenar');
```

SOLUCIÓN EJERCICIO 6.

```
SELECT j.nombre, j.dificultad, j.megusta, j.cdequipo, e.nombre 'equipo', e.comunidad
FROM juego j
INNER JOIN equipo e on j.cdequipo=e.cdequipo
WHERE j.cdequipo = (SELECT cdequipo FROM juego WHERE nombre='Elvenar');
```

SOLUCIÓN EJERCICIO 7.

```
SELECT c.cdconcur, c.nombre, p.cdjuego, p.puntos, p.fecha_inicio
FROM participa p
INNER JOIN concursante c on p.cdconcur=c.cdconcur
ORDER BY p.cdconcur;
```

SOLUCIÓN EJERCICIO 8.

```
SELECT c.cdconcur, c.nombre, ifnull(e.nombre, 'sin equipo') 'equipo',
ifnull(j.nombre, 'sin juego') 'juego', IFNULL(p.puntos, 0) 'puntos',
IFNULL(p.fecha_inicio, 'sin fecha') 'fecha'
FROM concursante c
LEFT OUTER JOIN equipo e ON c.cdequipo=e.cdequipo
LEFT OUTER JOIN participa p ON p.cdconcur=c.cdconcur
LEFT OUTER JOIN juego j ON j.cdjuego=p.cdjuego
ORDER BY c.nombre;
```

SOLUCIÓN EJERCICIO 9.

```
SELECT c.cdconcur, c.nombre, ROUND(AVG(p.puntos),2) 'puntos'
FROM concursante c
INNER JOIN participa p ON c.cdconcur=p.cdconcur
GROUP BY p.cdconcur
order by 3 desc;
```

SOLUCIÓN EJERCICIO 10.

```
SELECT c.cdconcur, c.nombre, ROUND(AVG(p.puntos),2) 'puntos', id.cdconcur 'idolo', id.nombre
FROM concursante c
INNER JOIN concursante id on c.cdidolo=id.cdconcur
INNER JOIN participa p ON c.cdconcur=p.cdconcur
GROUP BY p.cdconcur
order by 3 desc
limit 5;
```

SOLUCIÓN EJERCICIO 11.

```
SELECT j.nombre AS 'Nombre juego', j.megusta, j.dificultad, count(j.cdjuego) AS 'participan'
FROM juego j
INNER JOIN participa p ON p.cdjuego=j.cdjuego
where j.nombre like '%empire%'
GROUP BY j.cdjuego;
```

SOLUCIÓN EJERCICIO 12.

```
SELECT cdjuego, count(cdconcur)
FROM participa
GROUP BY cdjuego
HAVING count(cdconcur)>=2
order by 2 desc;
```