

## **Actividad de aprendizaje 5-08**

### **Se trata de:**

- Realizar consultas SELECT que de diversos tipos contra la base de datos **concursumusica**. Fundamentalmente se utilizarán subconsultas, aunque no es obligatorio usarlas.

### **Objetivos:**

- Editar y ejecutar instrucciones SELECT sobre combinaciones de tablas.
- Obtener los resultados de funciones de agrupamiento sobre todas las filas agrupadas por una o varias columnas.
- Obtener resultado de consultas condicionadas a lo obtenido en una función de agrupamiento.
- Realizar subconsultas para obtener valores usados en cláusulas WHERE y HAVING
- Realizar subconsultas que generen una hoja de resultados fuente para otra consulta.

### **Ejercicios**

1.- Obtener los nombres y apellidos de los usuarios que han votado canciones del grupo **Lori Meyers**

```
SELECT DISTINCT usuarios.nombre, apellidos  
FROM votos INNER JOIN canciones ON cancion=numcancion  
INNER JOIN grupos ON grupo=codgrupo  
INNER JOIN usuarios ON user=usuario  
WHERE grupos.nombre='Lori Meyers';
```

2.- Obtener el título de la canción que tiene más votos. Si hay varias canciones con el máximo de votos, deben salir todas ellas.

Una idea para realizarlo es obtener en una subconsulta el número de votos de cada canción ordenado por votos descendientemente y limitada a una fila.

```
SELECT titulo, count(*)  
FROM canciones INNER JOIN votos  
ON cancion=numcancion  
GROUP BY numcancion
```

```
HAVING count(*) =  
(SELECT count(*)  
FROM canciones INNER JOIN votos ON cancion=numcancion  
GROUP BY numcancion ORDER BY count(*) DESC LIMIT 1);
```

3.- A partir de la **tabla votos**, obtener el título de todas las canciones existentes junto con el número total de votos que han recibido ordenadas por título de canción. Tienes que unir dos consultas, una que obtiene los votos de las canciones votadas y una que obtiene las canciones no votadas junto con cero votos.

```
SELECT titulo, count(*)  
FROM canciones INNER JOIN votos ON cancion=numcancion  
GROUP BY numcancion  
UNION  
SELECT titulo, 0  
FROM canciones LEFT JOIN votos ON cancion=numcancion  
WHERE usuario IS NULL  
GROUP BY numcancion  
ORDER BY titulo;
```

También podríamos obtener el mismo resultado con

```
SELECT titulo, COUNT(cancion) as total  
FROM canciones LEFT JOIN votos ON cancion=numcancion  
GROUP BY titulo  
ORDER BY titulo;
```

Pero esto **no es correcto atendiendo al enunciado**.

4.- Obtener cuantas canciones han recibido votos.

```
SELECT count(DISTINCT cancion) AS votadas  
FROM votos;
```

5.- Obtener los nombres, apellidos y función de los componentes que forman el grupo al que pertenece **Jorge Guirao**.

```
SELECT componentes.nombre, apellido, funcion  
FROM componentes  
WHERE grupo=  
(SELECT grupo FROM componentes WHERE nombre='jorge' AND apellido='guirao');
```

6.- Obtener el nombre y apellidos del usuario registrado más joven, su edad (al finalizar el año actual) y los títulos de las canciones que ha votado.

```
SELECT nombre, apellidos, year(curdate())-year(fechanac) AS edad, titulo  
FROM canciones INNER JOIN votos ON numcancion=cancion  
INNER JOIN usuarios ON user=usuario  
WHERE fechanac=  
(SELECT max(fechanac) FROM usuarios);
```

7.- Obtener el nombre de cada grupo y la duración menor de sus canciones.

```
SELECT nombre, min(duracion)  
FROM canciones INNER JOIN grupos ON grupo=codgrupo  
GROUP BY grupo;
```

8.- Obtener el nombre de cada grupo y el título de su canción de duración menor. Para hacer esto requieres usar una subconsulta que genera una tabla derivada (subconsulta en FROM).

```
SELECT nombre, titulo, tiempo  
FROM (select min(duracion) as tiempo, grupo from canciones group by grupo) AS d  
INNER JOIN grupos ON d.grupo=codgrupo  
INNER JOIN canciones ON canciones.grupo=codgrupo  
WHERE duracion=tiempo;
```