

## Solución actividad de aprendizaje 5-02

### Se trata de:

- Realizar consultas SELECT sobre operaciones de combinación de tablas en la base de datos **concurso musical**.

### Objetivos:

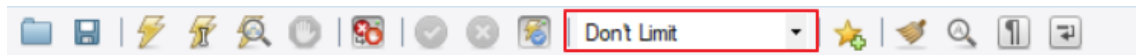
- Editar y ejecutar instrucciones SELECT sobre combinaciones de tablas usando INNER JOIN.
- Editar y ejecutar instrucciones SELECT sobre combinaciones producto cartesiano.
- Comparar la ejecución de instrucciones con producto cartesiano o con INNER JOIN.
- Establecer condiciones de selección sobre consultas aplicadas a combinaciones de tablas.

### Ejercicios

Todos estos ejercicios tienes que realizarlos sobre la base de datos **concurso musical**.

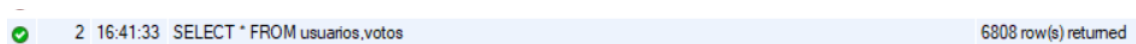
1.- Obtén el resultado del producto cartesiano entre la tabla **votos** y la tabla **usuarios**.

**Ojo!** Para que salgan en **Workbench** todas las filas del resultado, tienes que establecer en la pantalla de consultas que no se limite el número de filas de los resultados.



**SELECT \* FROM usuarios, votos;**

¿Cuántas filas se obtienen?



2.- A partir de una operación de producto cartesiano, obtén los nombres y apellidos de usuarios que han dado votos el día 2 de enero de 2018.

**SELECT nombre, apellidos FROM usuarios, votos WHERE fecha='2018-01-02' AND user=usuario;**

nombre	apellidos
Cristina	Cano Trueba
Guillermo	Ruiz Antolín
Daniel	Tresgallo Ceballos
Adolfo	Díez Cobo

Aunque en el resultado se hayan obtenido unas pocas filas, realmente se han obtenido anteriormente todas las filas de un producto cartesiano hecho en el ejercicio anterior y después a ese resultado intermedio se le ha aplicado la condición WHERE.

3.- Usando la combinación INNER JOIN, obtén los nombres y apellidos de usuarios que han dado votos el día 2 de enero de 2018.

```
SELECT nombre, apellidos FROM usuarios INNER JOIN votos ON user=usuario WHERE fecha='2018-01-02';
```

4.- Obtén un listado en el que se tengan todas las combinaciones posibles entre grupos y usuarios. En el listado debe mostrarse el código del grupo, el identificador de usuario (user) y la fecha actual (**Curdate()**).

```
SELECT user, codgrupo, curdate() FROM usuarios, grupos;
```

5.- Obtén el nombre y apellidos de usuario y el título de la canción correspondientes al último voto que se haya dado antes de la fecha actual.

```
SELECT nombre, apellidos, titulo FROM  
votos INNER JOIN canciones ON cancion=numcancion  
INNER JOIN usuarios ON user=usuario  
WHERE fecha<curdate() ORDER BY fecha DESC limit 1;
```

nombre	apellidos	titulo
Ana Patricia	Fernandez González	No gires

6.- Obtén sin repetir el nombre de todos los grupos de los que se han votado canciones en diciembre de 2017.

```
SELECT DISTINCT nombre FROM  
votos INNER JOIN canciones ON cancion=numcancion  
INNER JOIN grupos ON grupo=codgrupo  
WHERE fecha LIKE '2017-12-%';
```

7.- Obtén una lista con los nombres y apellidos de componentes que sean voz o vocalista de grupos de Madrid.

```
SELECT componentes.nombre, apellido FROM  
componentes INNER JOIN grupos ON grupo=codgrupo  
WHERE localidad='madrid'  
AND (funcion LIKE '%voz%' OR funcion LIKE '%vocalista%');
```

8.- Obtén una lista sin repetir con el nombre y apellidos de usuarios que han votado canciones del grupo 16.

```
SELECT DISTINCT nombre, apellidos FROM  
votos INNER JOIN canciones ON cancion=numcancion  
INNER JOIN usuarios ON user=usuario  
WHERE grupo=16;
```

9.- Obtén una lista sin repetir con el nombre y apellidos de usuarios que han votado canciones del grupo **Cyan**.

```
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='cyan';
```

10.- Obtén una lista sin repetir con el nombre y apellidos de usuarios que han votado canciones de grupos de Madrid o de Zaragoza entre el 20 y el 28 de diciembre, ambos incluidos.

```
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 (localidad='madrid' OR localidad = 'zaragoza')  
AND fecha>='2017-12-20' AND fecha <='2017-12-28';
```