

Consultas en más de dos tablas

por eamexicano

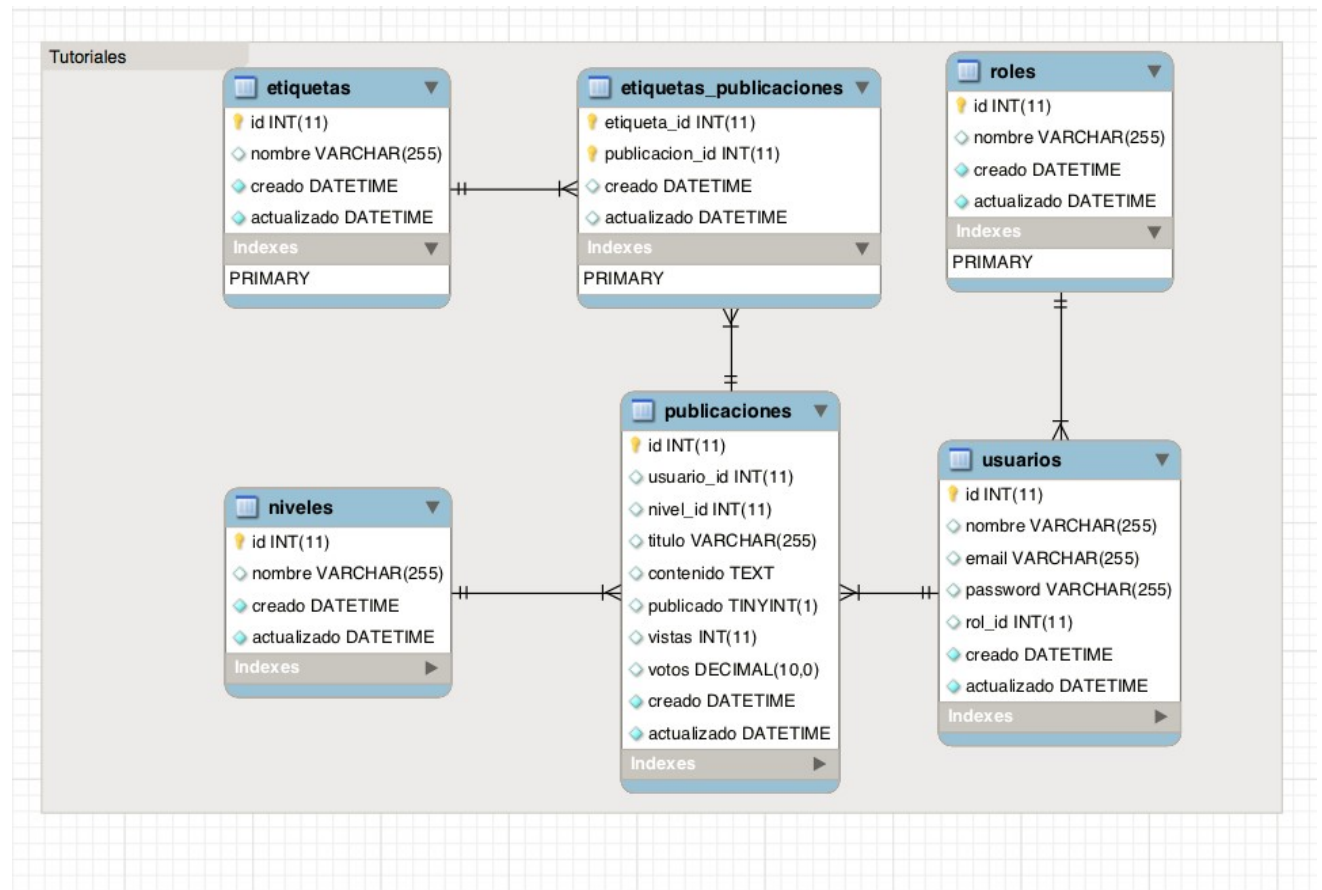
En las últimas tres publicaciones revisamos la manera de consultar datos que están en distintas tablas.

En las [consultas anidadas](#) revisamos cómo consultar más de dos tablas – en una relación M:M – pero eso no lo vimos con los diferentes [INNER JOIN](#) ni con los [OUTER JOIN](#).

Vamos a ver sólo dos casos:

- Relaciones M:M
- Otras relaciones

Incluyo el esquema de la base de datos -tutoriales – para darle contexto a las consultas.



Relaciones M:M

Vamos a comenzar con el ejemplo utilizado en las consultas anidadas donde queremos seleccionar el título de las publicaciones que utilizan la etiqueta HTML.

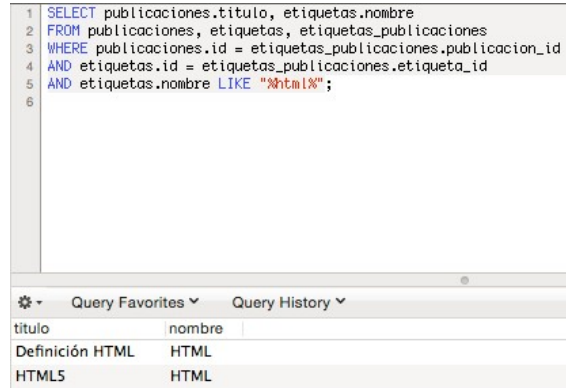
```
SELECT titulo FROM publicaciones WHERE id IN (
  SELECT publicacion_id FROM etiquetas_publicaciones WHERE etiqueta_id IN (
    SELECT id FROM etiquetas WHERE nombre LIKE "%html%"
  )
);
```



Podemos realizar la misma consulta con COMMA JOIN:

```
SELECT publicaciones.titulo, etiquetas.nombre
FROM publicaciones, etiquetas, etiquetas_publicaciones
WHERE publicaciones.id = etiquetas_publicaciones.publicacion_id
AND etiquetas.id = etiquetas_publicaciones.etiqueta_id
AND etiquetas.nombre LIKE "%html%";
```

```
SELECT publicaciones.titulo
FROM publicaciones, etiquetas, etiquetas_publicaciones
WHERE publicaciones.id = etiquetas_publicaciones.publicacion_id
AND etiquetas.id = etiquetas_publicaciones.etiqueta_id
AND etiquetas.nombre LIKE "%html%";
```

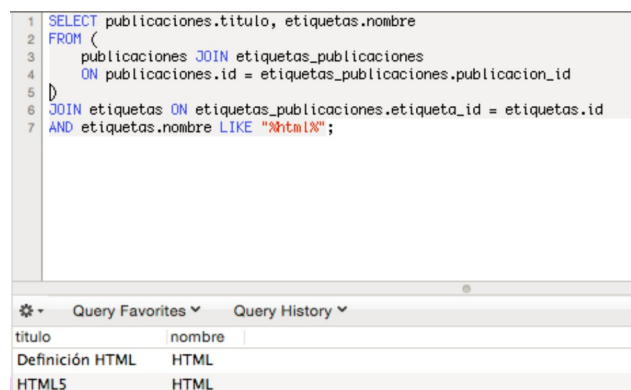


Y con las otras variedades de INNER JOIN:

```
SELECT publicaciones.titulo, etiquetas.nombre
FROM (publicaciones JOIN etiquetas_publicaciones ON publicaciones.id = etiquetas_publicaciones.publicacion_id)
JOIN etiquetas ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
AND etiquetas.nombre LIKE "%html%";
```

```
SELECT publicaciones.titulo, etiquetas.nombre
FROM (publicaciones INNER JOIN etiquetas_publicaciones ON publicaciones.id = etiquetas_publicaciones.publicacion_id)
INNER JOIN etiquetas ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
AND etiquetas.nombre LIKE "%html%";
```

```
SELECT publicaciones.titulo, etiquetas.nombre
FROM (publicaciones CROSS JOIN etiquetas_publicaciones ON publicaciones.id = etiquetas_publicaciones.publicacion_id)
CROSS JOIN etiquetas ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
AND etiquetas.nombre LIKE "%html%";
```



De manera personal prefiero utilizar COMMA JOIN sobre JOIN, INNER JOIN, CROSS JOIN para este tipo de consultas – porque la sintaxis me es más familiar y, el resultado va a ser el mismo –.

No hay diferencia en el desempeño de la consulta – [Percona](#).

Regresando al tema también podemos obtener el mismo resultado con OUTER JOIN – LEFT / RIGHT –.

```
SELECT publicaciones.titulo, etiquetas.nombre
FROM (publicaciones LEFT JOIN etiquetas_publicaciones ON publicaciones.id = etiquetas_publicaciones.publicacion_id)
LEFT JOIN etiquetas
ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
WHERE etiquetas.nombre LIKE "%html%";
```

```
SELECT publicaciones.titulo
FROM (publicaciones LEFT JOIN etiquetas_publicaciones ON publicaciones.id = etiquetas_publicaciones.publicacion_id)
LEFT JOIN etiquetas
ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
WHERE etiquetas.nombre LIKE "%html%";
```

1	SELECT publicaciones.titulo, etiquetas.nombre
2	FROM (
3	publicaciones LEFT JOIN etiquetas_publicaciones
4	ON publicaciones.id = etiquetas_publicaciones.publicacion_id
5)
6	LEFT JOIN etiquetas
7	ON etiquetas_publicaciones.etiqueta_id = etiquetas.id
8	WHERE etiquetas.nombre LIKE "%html%";

Query Favorites	Query History
titulo	nombre
Definición HTML	HTML
HTML5	HTML

Otras relaciones

Se pueden realizar consultas a más de dos tablas aún si la relación no es M:M, basta con que exista una tabla que esté relacionada con dos o más tablas para poder realizar este tipo de consultas.

En este caso queremos mostrar el nivel, el título de la publicación y el nombre del usuario que la escribió. Como se muestran datos de las tres tablas no es posible hacerlo con consultas anidadas.

COMMA JOIN

```
SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
usuarios.nombre AS 'usuario'
FROM niveles, publicaciones, usuarios
WHERE
niveles.id = publicaciones.nivel_id
AND publicaciones.usuario_id = usuarios.id;
```

```

1 SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
2 usuarios.nombre AS 'usuario'
3 FROM niveles, publicaciones, usuarios
4 WHERE niveles.id = publicaciones.nivel_id
5 AND publicaciones.usuario_id = usuarios.id;

```

Query Favorites ▾			Query History ▾		
nivel	titulo	usuario			
Fácil	Definición HTML	Dashawn Renner			
Fácil	HTML5	Dashawn Renner			
Fácil	Definición CSS	Ms. Eileen Waelchi			
Medio	CSS3	Ms. Eileen Waelchi			
Medio	Definición Javascript	Blaise Gaylord			
Difícil	Archivo JS	Blaise Gaylord			
Difícil	XAMPP	Dashawn Renner			

También se puede filtrar el contenido de los resultados obtenidos. En este caso mostrar solo las publicaciones cuyo título sea HTML.

```
SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
usuarios.nombre AS 'usuario'
FROM niveles, publicaciones, usuarios
WHERE
niveles.id = publicaciones.nivel_id
AND publicaciones.usuario_id = usuarios.id
AND publicaciones.titulo LIKE "%html%";
```

```

1 SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
2 usuarios.nombre AS 'usuario'
3 FROM niveles, publicaciones, usuarios
4 WHERE niveles.id = publicaciones.nivel_id
5 AND publicaciones.usuario_id = usuarios.id
6 AND publicaciones.titulo LIKE "%html%";

```

Query Favorites			Query History	
nivel	titulo	usuario		
Fácil	Definición HTML	Dashawn Renner		
Fácil	HTML5	Dashawn Renner		

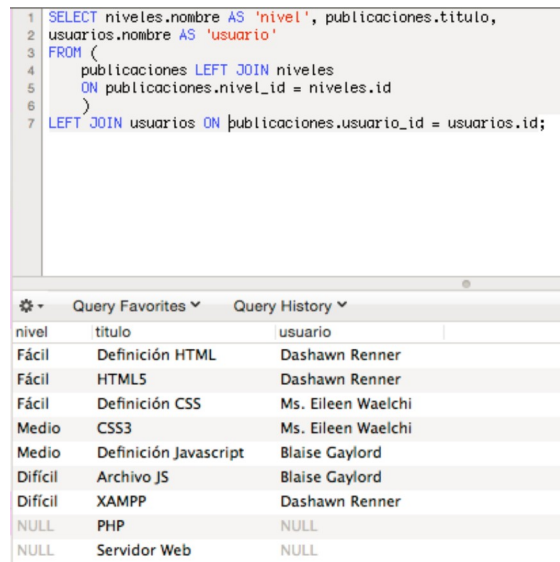
Como vimos, en OUTER JOIN – LEFT / RIGHT – es importante el orden en el que se especifican las tablas.

En este caso la publicación es la tabla principal porque queremos mostrar todas las publicaciones, aunque no tengan nivel ni usuario asociados.

En el caso de tener registros asociados, mostrar el nombre del nivel y el nombre del usuario.

LEFT JOIN

```
SELECT niveles.nombre AS 'nivel', publicaciones.titulo, usuarios.nombre AS 'usuario'
FROM (
    publicaciones LEFT JOIN niveles
    ON publicaciones.nivel_id = niveles.id
)
LEFT JOIN usuarios ON publicaciones.usuario_id = usuarios.id;
```

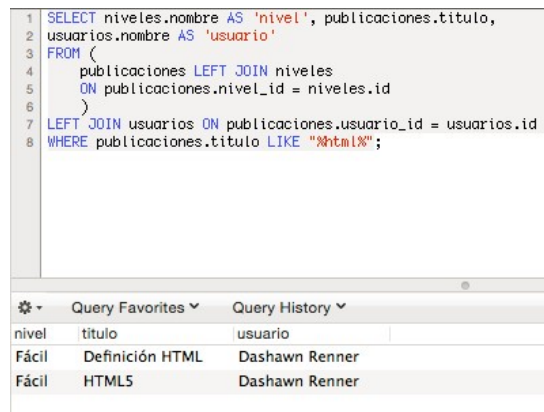


```
1 SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
2 usuarios.nombre AS 'usuario'
3 FROM (
4     publicaciones LEFT JOIN niveles
5     ON publicaciones.nivel_id = niveles.id
6 )
7 LEFT JOIN usuarios ON publicaciones.usuario_id = usuarios.id;
```

nivel	titulo	usuario
Fácil	Definición HTML	Dashawn Renner
Fácil	HTML5	Dashawn Renner
Fácil	Definición CSS	Ms. Eileen Waelchi
Medio	CSS3	Ms. Eileen Waelchi
Medio	Definición Javascript	Blaise Gaylord
Difícil	Archivo JS	Blaise Gaylord
Difícil	XAMPP	Dashawn Renner
NULL	PHP	NULL
NULL	Servidor Web	NULL

Aquí también podemos definir más criterios para filtrar. Hay que tener cuidado de utilizar WHERE en lugar de AND para unir el primer criterio de búsqueda

```
SELECT niveles.nombre AS 'nivel', publicaciones.titulo, usuarios.nombre AS 'usuario'
FROM (
    publicaciones LEFT JOIN niveles
    ON publicaciones.nivel_id = niveles.id
)
LEFT JOIN usuarios
ON publicaciones.usuario_id = usuarios.id
WHERE publicaciones.titulo LIKE "%html%";
```



```
1 SELECT niveles.nombre AS 'nivel', publicaciones.titulo,
2 usuarios.nombre AS 'usuario'
3 FROM (
4     publicaciones LEFT JOIN niveles
5     ON publicaciones.nivel_id = niveles.id
6 )
7 LEFT JOIN usuarios ON publicaciones.usuario_id = usuarios.id
8 WHERE publicaciones.titulo LIKE "%html%";
```

nivel	titulo	usuario
Fácil	Definición HTML	Dashawn Renner
Fácil	HTML5	Dashawn Renner

Recuerda:

Además de poder visualizar los datos que están en distintas tablas puedes aplicar lo que hemos visto hasta ahora – ordenar, limitar, desplazar, agrupar, funciones –.