

Facultad de Telemática Ingeniería de Software

7°E Base de Datos Avanzadas

A3 APLICACIÓN DE SQL SENTENCIAS AVANZADAS -JOINS, GROUP, HAVING, UNIONS-

Vizcaino Lupian Alejandro

Profesor Gallardo Heredia Armando Román

Colima, Col., martes 27 de agosto del 2024

BASE DE DATOS 1: PAPELERÍA

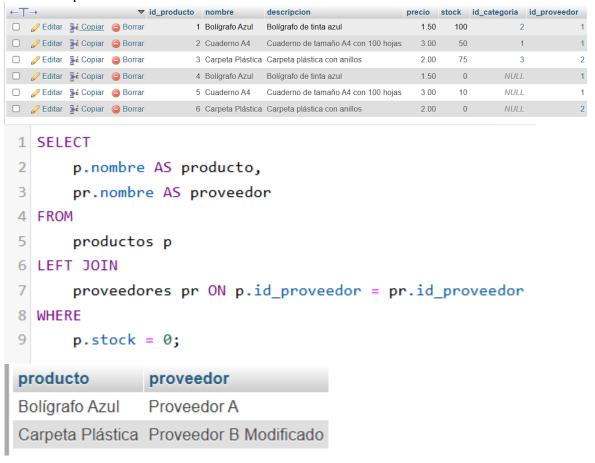
1. Mostrar los nombres de los productos, sus precios, categorías y proveedores correspondientes.

```
1 SELECT
 2
       p.nombre AS producto,
 3
       p.precio,
       c.nombre AS categoria,
 4
 5
       pr.nombre AS proveedor
 6 FROM
 7
       productos p
 8 LEFT JOIN
       categorias c ON p.id_categoria = c.id_categoria
 9
10 LEFT JOIN
       proveedores pr ON p.id_proveedor = pr.id_proveedor;
11
producto
               precio
                      categoria proveedor
                  1.50 Bolígrafos Proveedor A
Bolígrafo Azul
Cuaderno A4
                  3.00 Libretas
                                   Proveedor A
                  2.00 Carpetas Proveedor B Modificado
Carpeta Plástica
```

2. Contar cuántos productos hay en cada categoría.

```
1 SELECT
2
      c.nombre AS categoria,
3
      COUNT(p.id_producto) AS total_productos
4 FROM
5
      categorias c
6 LEFT JOIN
      productos p ON p.id_categoria = c.id_categoria
7
8 GROUP BY
      c.nombre;
categoria
          total_productos
Bolígrafos
Carpetas
Libretas
```

3. Mostrar los productos que tienen un stock de 0 y sus proveedores correspondientes.



4. Encontrar el producto más caro en cada categoría.

```
1 SELECT
       c.nombre AS categoria,
 3
       p.nombre AS producto,
4
       p.precio
5 FROM
       productos p
7 LEFT JOIN
       categorias c ON p.id_categoria = c.id_categoria
9 WHERE
       (p.id_categoria, p.precio) IN (
10
           SELECT id_categoria, MAX(precio)
11
           FROM productos
12
           GROUP BY id categoria
13
       );
14
 categoria producto
                           precio
```

Bolígrafos Bolígrafo Azul 1.50
Libretas Cuaderno A4 3.00
Carpetas Carpeta Plástica 2.00

5. Mostrar los nombres de los proveedores y los nombres de los productos que suministran.

```
1 SELECT
2    pr.nombre AS proveedor,
3    p.nombre AS producto
4 FROM
5    productos p
6 LEFT JOIN
7    proveedores pr ON p.id_proveedor = pr.id_proveedor;
```

proveedor	producto
Proveedor A	Bolígrafo Azul
Proveedor A	Cuaderno A4
Proveedor B Modificado	Carpeta Plástica
Proveedor A	Bolígrafo Azul
Proveedor A	Cuaderno A4
Proveedor B Modificado	Carpeta Plástica

6. Calcular el promedio de precios de productos en cada categoría.

```
1 SELECT
2     c.nombre AS categoria,
3     AVG(p.precio) AS promedio_precio
4 FROM
5     productos p
6 LEFT JOIN
7     categorias c ON p.id_categoria = c.id_categoria
8 GROUP BY
9     c.nombre;
```

categoria	promedio_precio
NULL	2.166667
Bolígrafos	1.500000
Carpetas	2.000000
Libretas	3.000000

7. Mostrar los nombres de los productos agrupados por categoría y proveedor.

categoria	proveedor	productos
NULL	Proveedor A	Bolígrafo Azul, Cuaderno A4
NULL	Proveedor B Modificado	Carpeta Plástica
Bolígrafos	Proveedor A	Bolígrafo Azul
Carpetas	Proveedor B Modificado	Carpeta Plástica
Libretas	Proveedor A	Cuaderno A4

8. Contar cuántos productos suministra cada proveedor.

```
1 SELECT
2    pr.nombre AS proveedor,
3    COUNT(p.id_producto) AS total_productos
4 FROM
5    productos p
6 LEFT JOIN
7    proveedores pr ON p.id_proveedor = pr.id_proveedor
8 GROUP BY
9    pr.nombre;
```

proveedor	total_productos
Proveedor A	4
Proveedor B Modificado	2

9. Mostrar los nombres de los productos en una categoría específica.

```
1 SELECT
2    p.nombre AS producto
3 FROM
4    productos p
5 LEFT JOIN
6    categorias c ON p.id_categoria = c.id_categoria
7 WHERE
8    c.nombre = 'Carpetas';
```

producto

Carpeta Plástica

10. Mostrar los nombres de los productos que no tienen una categoría asignada.

```
1 SELECT
2 p.nombre AS producto
3 FROM
4 productos p
5 WHERE
6 p.id_categoria IS NULL;

← → producto

□ ② Editar ♣ Copiar ⑤ Borrar Bolígrafo Azul
□ ② Editar ♣ Copiar ⑥ Borrar Cuaderno A4
□ ② Editar ♣ Copiar ⑥ Borrar Carpeta Plástica
```

BASE DE DATOS 2: BLOG VIAJERO:

1. Contar cuántos artículos ha escrito cada usuario.

```
SELECT u.Nombre, COUNT(a.ID_Articulo) AS Total_Articulos
FROM Usuarios u

LEFT JOIN Articulos a ON u.ID_Usuario = a.ID_Usuario
GROUP BY u.ID_Usuario, u.Nombre;

Nombre Total_Articulos

Juan Pérez 1

Ana Torres 4
```

2. Calcular el promedio de comentarios por artículo.

3. Mostrar los títulos de los artículos junto con la cantidad de comentarios que tienen.

```
1 SELECT a.Titulo, COUNT(c.ID_Comentario) AS Total_Comentarios
2 FROM Articulos a
3 LEFT JOIN Comentarios c ON a.ID_Articulo = c.ID_Articulo
4 GROUP BY a.ID_Articulo, a.Titulo;
```

Titulo	Total_Comentarios
Viaje a la montaña	0
Descubriendo la playa	0
Aventura en la selva	0
Nueva Aventura en la Ciudad	0
Explorando el desierto	0

4. Mostrar los nombres de los usuarios que tienen más artículos escritos.

```
1 SELECT u.Nombre
2 FROM Usuarios u
3 JOIN Articulos a ON u.ID_Usuario = a.ID_Usuario
4 GROUP BY u.ID_Usuario, u.Nombre
5 ORDER BY COUNT(a.ID_Articulo) DESC
6 LIMIT 1;
```

Nombre

Ana Torres

5. Encontrar el día de la semana en que se publican más artículos.

```
SELECT DAYNAME(a.Fecha_Publicacion) AS Dia_Semana, COUNT(a.ID_Articulo) AS Total_Articulos
FROM Articulos a
GROUP BY DAYNAME(a.Fecha_Publicacion)
ORDER BY Total_Articulos DESC
LIMIT 1;

Dia_Semana Total_Articulos
Thursday 5
```

6. Contar cuántos comentarios se hicieron en cada día.

```
SELECT DATE(c.Fecha) AS Dia, COUNT(c.ID_Comentario) AS Total_Comentarios

FROM Comentarios c

GROUP BY DATE(c.Fecha)

ORDER BY Dia;

✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0,0004 segundos.)

SELECT DATE(c.Fecha) AS Dia, COUNT(c.ID_Comentario) AS Total_Comentarios FROM Comentarios c GROUP BY DATE(c.Fecha) ORDER BY Dia;

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Dia Total_Comentarios
```

7. Calcular el promedio de comentarios por usuario.

```
1 SELECT AVG(ComentarioCount) AS Promedio_Comentarios
 2 FROM (
 3
            SELECT COUNT(c.ID_Comentario) AS ComentarioCount
 4
            FROM Articulos a
 5
            LEFT JOIN Comentarios c ON a.ID Articulo = c.ID Articulo
            GROUP BY a.ID_Articulo
 6
     ) AS Subquery;

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0006 segundos.)

SELECT AVG(ComentarioCount) AS Promedio_Comentarios FROM ( SELECT COUNT(c.ID_Comentario) AS ComentarioCount FROM Articulos a LEFT JOIN Comentarios c ON a.IO_Articulo
 ..ID_Articulo GROUP BY a.ID_Articulo ) AS Subquery;
□ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]
 ☐ Mostrar todo │ Número de filas: 25 	 Filtrar filas: Buscar en esta tabla
Opciones extra
Promedio_Comentarios
           0.0000
```

8. Conteo de Comentarios por Artículo y Usuario: Contar cuántos comentarios ha hecho cada usuario en cada artículo.

```
SELECT u.Nombre, a.Titulo, COUNT(c.ID_Comentario) AS Total_Comentarios

PROM Usuarios u

JOIN Comentarios c ON u.ID_Usuario = c.ID_Usuario

JOIN Articulos a ON c.ID_Articulo = a.ID_Articulo

GROUP BY u.ID_Usuario, a.ID_Articulo, u.Nombre, a.Titulo;

MySQL ha devuello un conjunto de valores vacio (es decir. cero columnas) (La consulta tardó 0,0144 segundos.)

SELECT u.Nombre, a.Titulo, COUNT(c.ID_Comentario) AS Total_Comentarios FROM Usuarios u JOIN Comentarios c ON u.ID_Usuario = c.ID_Usuario JOIN Articulos a ON c.ID_Articulo = a.ID_Articulo GROUP BY u.ID_Usuario, a.ID_Articulo, u.Nombre, a.Titulo;

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Nombre Titulo Total_Comentarios
```

9. Mostrar los títulos de los artículos con la mayor y menor cantidad de comentarios.

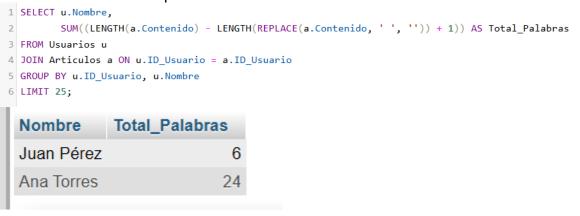
```
SELECT a.Titulo, COUNT(c.ID_Comentario) AS Total_Comentarios
FROM Articulos a

LEFT JOIN Comentarios c ON a.ID_Articulo = c.ID_Articulo
GROUP BY a.ID_Articulo, a.Titulo

ORDER BY Total_Comentarios DESC
LIMIT 1;
```



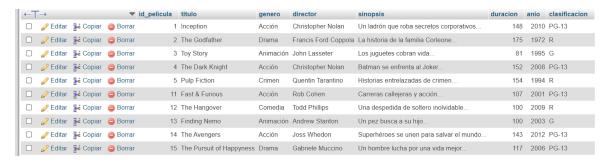
10. Calcular el total de palabras en todos los artículos de cada usuario.



Información importante:

En la segunda base de datos las querys para los comentarios fueron creadas satisfactoriamente, pero hubo un error al intentar agregar comentarios ejemplos, estos no podían ser agregados por restricciones con las llaves foráneas, con el miedo de modificar algo que pueda dañar el proyecto, lo he dejado cómo está; considero que este es el mejor camino ya que demuestra que las querys funcionan, el error es meramente de datos no establecidos con anterioridad.

BASE DE DATOS 3: CARTELERA DE PELICULAS DE CINE



1.- Mostrar los títulos de todas las películas de género "Acción" y "Comedia".

2.- Mostrar los títulos de todas las películas que se proyectan en cines de las ciudades "Ciudad A" y "Ciudad B".

```
SELECT DISTINCT p.titulo FROM peliculas p
JOIN funciones f ON p.id_pelicula = f.id_pelicula
JOIN cines c ON f.id_cine = c.id_cine
WHERE c.ciudad IN ('New York', 'Los Angeles');

titulo
Inception
The Godfather
Toy Story
The Dark Knight
```

3.- Mostrar los títulos de todas las películas con clasificación "PG" y"PG-13".

```
1 SELECT titulo FROM peliculas
2 WHERE clasificacion IN ('PG', 'PG-13');
```



4.-Listar todas las Películas Proyectadas en un Cine Específico: Mostrar los títulos de todas las películas que se proyectan en un cine específico (ID Cine = 3).

```
1 SELECT DISTINCT p.titulo FROM peliculas p
2 JOIN funciones f ON p.id_pelicula = f.id_pelicula
3 WHERE f.id_cine = 3;

titulo
Pulp Fiction
```

5.- Mostrar los títulos de todas las películas de género "Acción" o "Drama" con duración menor a 120 minutos.

```
SELECT titulo FROM peliculas

WHERE genero IN ('Acción', 'Drama') AND duracion < 120;

titulo

□  ② Editar  ♣ Copiar  ⊜ Borrar Fast & Furious

□  ② Editar  ♣ Copiar  ⊜ Borrar The Pursuit of Happyness
```