



UD3

CONSULTA Y MODIFICACIÓN DE BASES DE DATOS

MP_0484
Bases de Datos

3.5 Consultas
avanzadas II

Operaciones de conjuntos sobre consultas

UNION e INTERSECT son operadores de conjuntos, si entendemos como conjunto las filas resultantes de una sentencia SELECT válida. Por lo tanto, estos operadores nos van a permitir combinar los resultados de varias consultas SELECT para obtener un único conjunto o resultado.

Formato

SELECT ... FROM ... WHERE ...

UNION | INTERSECT

SELECT ... FROM ... WHERE ...

Estos operadores presentan una serie de restricciones que debemos tener en cuenta en su uso:

- Ambas consultas deben devolver el mismo número de columnas.
- El tipo de datos de cada columna en la primera tabla debe ser el mismo que el tipo de datos de la columna correspondiente en la segunda tabla.
- Ninguna de las dos consultas puede estar ordenada con la cláusula ORDER BY. Sin embargo, los resultados combinados pueden ser ordenados.

Para entender estas operaciones de conjuntos, vamos a realizar ejemplos con las tablas que se presentan en la figura 1.

| SociosAntiguos | | | | Socios | | | SociosNuevos | | | | |
|----------------|---------|--------|------------|--------|---------|--------|-----------------|--|---------|--------|------------|
| | NombreA | SexoA | FechaNacA | | nombre | sexo | FechaNacimiento | | NombreN | SexoN | FechaNacN |
| | ANDRÉS | HOMBRE | 2000-12-14 | | ANA | MUJER | 2005-10-12 | | ANA | MUJER | 2005-10-12 |
| | ERNESTO | HOMBRE | 2006-10-10 | | ERNESTO | HOMBRE | 2006-10-10 | | ERNESTO | HOMBRE | 2006-10-10 |
| | IRENE | MUJER | 2004-03-15 | | JUAN | HOMBRE | 2002-12-12 | | JUAN | HOMBRE | 2002-12-12 |
| | MARÍA | MUJER | 2004-01-01 | | LUISA | MUJER | 2000-12-12 | | MAITE | MUJER | 2003-12-12 |
| | NULL | NULL | NULL | | MARÍA | MUJER | 2004-01-01 | | SOFÍA | MUJER | 2004-13-12 |
| | | | | | PEDRO | HOMBRE | 2005-10-05 | | | | |
| | | | | | RAQUEL | MUJER | 2008-10-10 | | | | |

Figura 1. Base de datos de socios

UNION

Esta cláusula combina los resultados de dos o más consultas (figura 2). En el entorno de MySQL esta es una herramienta realmente útil, pues como recordamos, este software no implementa nativamente el FULL OUTER JOIN que vimos en la sesión anterior. Es por ello que como solución alternativa para obtener el conjunto podemos optar por unir un LEFT JOIN y un RIGHT JOIN empleando UNION.

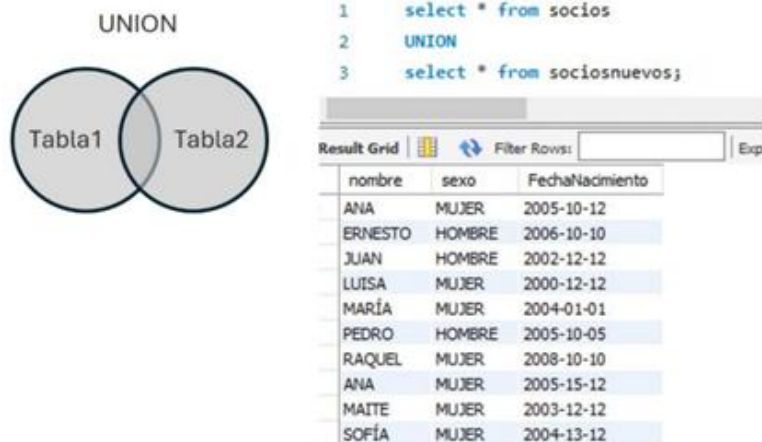


Figura 2. Ejemplo de uso de la cláusula UNION

Los nombres de las columnas de las dos consultas no tienen por qué ser idénticos, lo habitual es mostrar el nombre del campo de la primera tabla/consulta. Por defecto, la operación UNION elimina las filas duplicadas como parte de su procesamiento. Existe UNION ALL, que muestra todos los registros si se sabe que no va a tener filas duplicadas. Esto se debería especificar, ya que el sistema gestor de bases de datos podrá optimizar la consulta en gran medida y esta se ejecutará mucho más rápido.

La cláusula WHERE no puede aparecer en ninguna de las dos sentencias SELECT. UNION puede ser utilizado para combinar tres o más conjuntos de resultados. Se podrían utilizar paréntesis para establecer un orden en las uniones en caso de utilizar tres o más tablas.

INTERSECT

Por su parte, esta cláusula devuelve los valores comunes de ambas, obtenemos una intersección, es decir, únicamente se nos devolverán los resultados comunes a ambas consultas (figura 3).

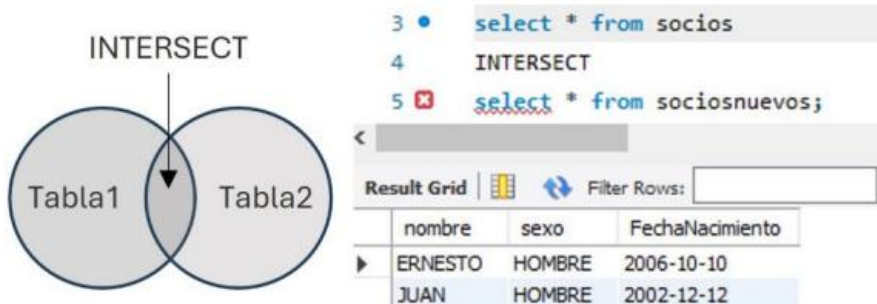


Figura 3. Ejemplo de uso de la cláusula INTERSECT

Consultas que generan valores resumen

Fuera del ámbito de las consultas que agregan datos de varias tablas, hasta ahora hemos utilizado:

- El bloque SELECT para recuperar filas de una tabla.
- La cláusula WHERE para acotar el número de filas a las que nos interesan.
- Funciones de grupo para trabajar con el conjunto completo de filas seleccionadas y calcular valores.

Ahora vamos a darle un matiz adicional: queremos que se realicen grupos dentro del conjunto de filas seleccionadas y realizar cálculos para esos grupos de filas. Es diferente calcular el salario medio de todos los empleados, que el salario medio de los empleados por departamento. Para ello, vamos a utilizar la cláusula group by; si quisiéramos establecer condiciones sobre el resultado de la agrupación, usaríamos la cláusula having. Es importante destacar que no puede existir el having sin un group by antes.

Vamos a ver el ejemplo de la figura 4 en el que consultamos el número de empleados que hay por cada departamento.

```
select DptoNo, count(EmpNo)
from emple
group by DptoNo;
```

| | DptoNo | count(EmpNo) |
|---|--------|--------------|
| ▶ | 10 | 3 |
| | 20 | 5 |
| | 30 | 6 |

Figura 4. Ejemplo de uso de la cláusula GROUP BY

Nótese que para hacer esta consulta hemos tenido que agrupar las filas de la tabla por DptoNo y una vez realizados los grupos contar cuántos empleados hay en cada grupo. Por el contrario, si de la consulta anterior solo queremos sacar los departamentos que tienen más de cuatro empleados, habría que poner una condición en el HAVING (figura 5).

```
select DptoNo, count(EmpNo)
from emple
group by DptoNo
having count(EmpNo)>4;
```

| | DptoNo | count(EmpNo) |
|---|--------|--------------|
| ▶ | 20 | 5 |
| | 30 | 6 |

Figura 5. Ejemplo de uso de la cláusula HAVING

Si queremos que nos cuente los empleados de los departamentos 10 y 20, la condición sería en el WHERE, ya que antes de realizar la agrupación hay que elegir las filas con las que queremos trabajar (empleados del dpto. 10 y 20) y luego armar la agrupación (figura 6).

```
select DptoNo, count(EmpNo)
from emple
where DptoNo In (10,20)
group by DptoNo;
```

| | DptoNo | count(EmpNo) |
|---|--------|--------------|
| ▶ | 10 | 3 |
| | 20 | 5 |

Figura 6. Ejemplo de uso de las cláusulas WHERE y GROUP BY

Se podría llegar a tener dos condiciones: una en la WHERE que use solo las filas de los dptos. 10 y 20 y, una vez realizada la agrupación, que muestre los dptos. que tienen más de cuatro empleados (figura 7).

```
select DptoNo, count(EmpNo)
from emple
where DptoNo In (10,20)
group by DptoNo
having count(DptoNo)>4;
```

| | DptoNo | count(EmpNo) |
|---|--------|--------------|
| ▶ | 20 | 5 |

Figura 7. Ejemplo de uso de todas las cláusulas anteriores

El orden de evaluación de las cláusulas en tiempo de ejecución es la siguiente:

1. **WHERE** selecciona las filas.
2. **GROUP BY** agrupa las filas.
3. **HAVING** selecciona solo los grupos que nos interesan.
4. **ORDER BY** muestra el resultado según se indique.