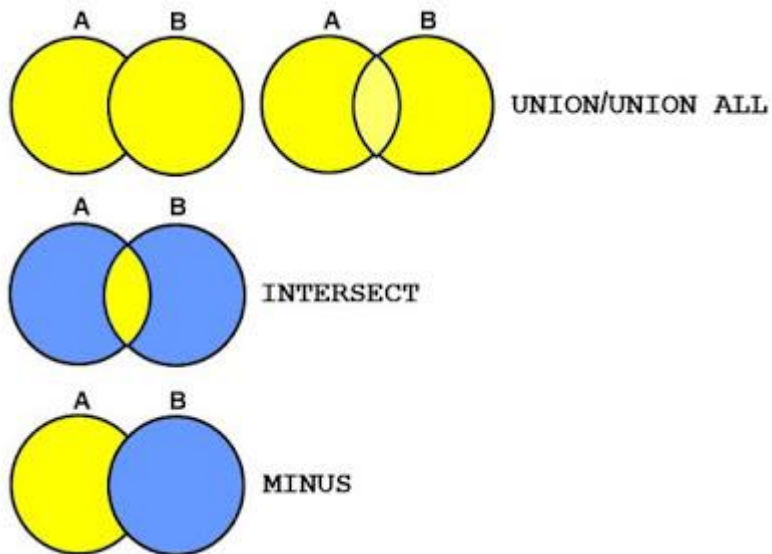


Operadores SET

Antonio Espín Herranz

Operadores SET



UNION

Todas las filas distintas seleccionadas por cualquiera de las dos consultas

UNION ALL

Todas las filas seleccionadas por cualquiera de las dos consultas, incluidos todos los duplicados

INTERSECT

Todas las filas distintas seleccionadas por ambas consultas

MINUS

Todas las filas seleccionadas por la primera sentencia **SELECT** y no seleccionadas en la segunda sentencia **SELECT**

Unión, Intersección, Diferencia

- Podemos realizar operaciones de conjuntos sobre datos de la tablas. Teniendo en cuenta que de todas las consultas tendremos los mismo campos con el mismo tipo:
- Sintaxis:
 - Select lista de columnas from S
 - { union | union all | intersect | minus }
 - Select lista de columnas from T;
 - ...

Unión, Intersección, Diferencia

- **UNIONES**

- La palabra UNION permite añadir el resultado de un SELECT a otro SELECT. Para ello ambas instrucciones tienen que utilizar el mismo número y tipo de columnas.
Ejemplo:
- SELECT nombre FROM provincias
- UNION SELECT nombre FROM comunidades
- El resultado es una tabla que contendrá nombres de provincia y de comunidades. Es decir, UNION crea una sola tabla con registros que estén presentes en cualquiera de las consultas. Si están repetidas sólo aparecen una vez, para mostrar los duplicados se utiliza UNION ALL en lugar de la palabra UNION.

- **INTERSECCIONES**

- De la misma forma, la palabra INTERSECT permite unir dos consultas SELECT de modo que el resultado serán las filas que estén presentes en ambas consultas.

- **DIFERENCIA**

- Con **MINUS** también se combinan dos consultas SELECT de forma que aparecerán los registros del primer SELECT que no estén presentes en el segundo.

Ejemplos

- **SELECT** employee_id, job_id
FROM employees
UNION
SELECT employee_id, job_id
FROM job_history;
/*En el Query anterior notamos algunos códigos de empleados repetidos, pero con el puesto de trabajo diferente, dichos registros pertenecen a los empleados que han cambiado de trabajo en algún momento; Los registros idénticos en ambas tablas no muestran solo una vez.*/
- ---
- **SELECT** employee_id, job_id, department_id
FROM employees
UNION
SELECT employee_id, job_id, department_id
FROM job_history;
/*Este Query es similar al anterior a el, con la pequeña diferencia de que este incluye la columna *department_id*. Este genera mas registros debido a que hay empleados que aparte de haber cambiado de puesto, también han cambiado de departamento.*/

Ejemplos

- ***SELECT*** *employee_id, job_id*
FROM *employees*
UNION ALL
SELECT *employee_id, job_id*
FROM *job_history;*

/*En este ejemplo vemos como UNION ALL muestra filas repetidas ya que no elimina los duplicados, para ver la diferencia correr el mismo Query solo con UNION.*/

Ejemplos

```
SELECT employee_id, job_id  
FROM employees  
INTERSECT  
SELECT employee_id, job_id  
FROM job_history;
```

*/*Este Query muestra los empleados que han cambia de puesto en algún momento pero han vuelto a su misma posición anterior, por dicha razón los registros para cada tabla coinciden y por ello se muestran.*/*

- ---

```
SELECT employee_id, job_id, department_id  
FROM employees  
INTERSECT  
SELECT employee_id, job_id, department_id  
FROM job_history;
```

*/*Al agregar el columna **department_id** notamos que la cantidad de registros disminuye, esto se debe a que este campo es diferente en cada tabla.*/*