

Operaciones de Conjuntos

Álvaro Herrera

13 de noviembre de 2004

Operaciones de Conjuntos

- ▶ UNION (\cup)
- ▶ INTERSECT (\cap)
- ▶ EXCEPT (\setminus)

Tablas de Ejemplo

tabla **departamentos**

deptoId	nombreDepto
1	Desarrollo
2	Operaciones

tabla **empleados**

empleadoId	nombreEmpleado	deptoId
1	Joel Iturra	1
2	Aldrin Martoq	2

tabla **parientes**

parienteId	empleadoId	nombre
1	1	Elena
2	1	Alejandro
3	2	Ghislaine

UNION

- ▶ Permite unir resultados de varias consultas
- ▶ Resultados deben ser compatibles
 - ▶ misma cantidad de columnas
 - ▶ tipos de cada columna debe ser compatibles

UNION — Ejemplo

```
SELECT nombre FROM empleados  
UNION  
SELECT nombre FROM parientes;
```

nombre

Aldrin Martoq

Alejandro

Elena

Ghislaine

Joel Iturra

UNION — Observaciones

- ▶ ¡El resultado está ordenado!
 - ▶ UNION elimina repetidos
 - ▶ Implementación: para eliminar repetidos, se ordena el conjunto
 - ▶ Para evitarlo, use UNION ALL
 - ▶ Además, es más eficiente
- ▶ ¿Cómo distinguir un pariente de un empleado?
 - ▶ Use constantes!
 - ▶ Consistencia r00lz

UNION — Ejemplo 2 (malo)

```
SELECT empleadoId AS id, NULL, nombre
      FROM empleados
UNION ALL
      SELECT parienteid, empleadoId, nombre
      FROM parientes
ORDER BY 1
```

<i>id</i>	<i>?column?</i>	<i>nombre</i>
1		Joel Iturra
1	1	Elena
2		Aldrin Martoq
2	1	Alejandro
3	2	Ghislaine

UNION — Ejemplo 3 (bueno)

```
SELECT 'p' AS tipo, empleadoId, nombre, NULL as dep
      FROM parientes
UNION ALL
      SELECT 'e', empleadoId, nombre, deptoid
      FROM empleados
ORDER BY 2, 1;
```

<i>tipo</i>	<i>empleadoId</i>	<i>nombre</i>	<i>deptoid</i>
e	1	Joel Iturra	1
p	1	Elena	
p	1	Alejandro	
e	2	Aldrin Martoq	2
p	2	Ghislaine	

INTERSECT

- ▶ Entrega la intersección de dos conjuntos
- ▶ Como UNION, los conjuntos deben ser compatibles
- ▶ Compara filas completas, columna a columna
- ▶ Ejemplo: agregamos una tabla de proyectos, los empleados pueden participar en más de un proyecto

Tablas de Ejemplo

tabla **proyectos**

proyectoId	nombre
1	Plan Z
2	E.P.E. (<i>El Proyecto E.P.E.</i>)

tabla **empleados_proyectos**

empleadoId	proyectoId
2	1
1	1
2	2

Modelo del Ejemplo

```
SELECT e.nombre, p.nombre  
      FROM empleados AS e  
      JOIN empleados_proyectos USING (empleadoId)  
      JOIN proyectos AS p USING (proyectoId);
```

<i>nombre</i>	<i>nombre</i>
Joel Iturra	Plan Z
Joel Iturra	E.P.E.
Aldrin Martoq	E.P.E.

INTERSECT — Ejemplo

Los empleados que trabajan en ambos proyectos:

```
SELECT empleados.nombre
FROM empleados
      JOIN empleados_proyectos USING (empleadoId)
      JOIN proyectos USING (proyectoId)
WHERE proyectos.nombre = 'E.P.E.'
INTERSECT
SELECT empleados.nombre
FROM empleados
      JOIN empleados_proyectos USING (empleadoId)
      JOIN proyectos USING (proyectoId)
WHERE proyectos.nombre = 'Plan Z';
```

nombre

Joel Iturra

INTERSECT — Observaciones

- ▶ Se compara igualdad de cada fila
- ▶ Todas las columnas deben ser iguales!
- ▶ Generalmente se extraerá sólo la llave primaria, en una subconsulta, y luego unirlo usando WHERE llave IN (SELECT ... INTERSECT ...)
- ▶ Como UNION, se eliminan los registros repetidos. Use INTERSECT ALL para impedir esto.

EXCEPT

- ▶ Entrega la diferencia de conjuntos
- ▶ Conjuntos deben ser compatibles
- ▶ Como INTERSECT, compara filas completas, columna a columna
- ▶ Generalmente se usará para extraer la llave primaria en una subconsulta, juntándolo todo con la consulta externa.
- ▶ Como UNION e INTERSECT, elimina registros repetidos. Use EXCEPT ALL para impedirlo.

EXCEPT — Ejemplo

Empleados que trabajan en E.P.E. y no en Plan Z.

```
SELECT empleados.nombre
  FROM empleados
        JOIN empleados_proyectos USING (empleadoId)
        JOIN proyectos USING (proyectoId)
 WHERE proyectos.nombre = 'E.P.E.'
EXCEPT
SELECT empleados.nombre
  FROM empleados
        JOIN empleados_proyectos USING (empleadoId)
        JOIN proyectos USING (proyectoId)
 WHERE proyectos.nombre = 'Plan Z';
```

nombre

Aldrin Martoq