BASES DE DATOS

**CONSULTAS SQL PARTE 2**

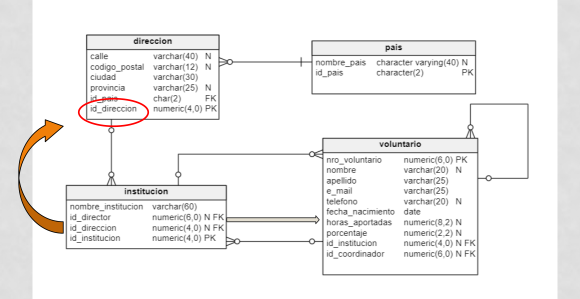
**Consultas de más de una tabla**

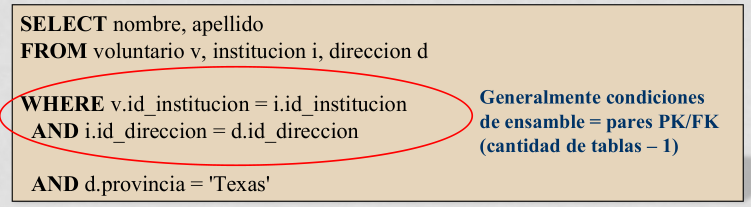
Se re refiere a las consultas de varias tablas a la vez

**Ejemplo:**

Seleccionar el nombre y apellido de los voluntarios del estado (provincia) de Texas.

Hay que revisar desde el esquema las condiciones de ensamble entre las distintas tablas.



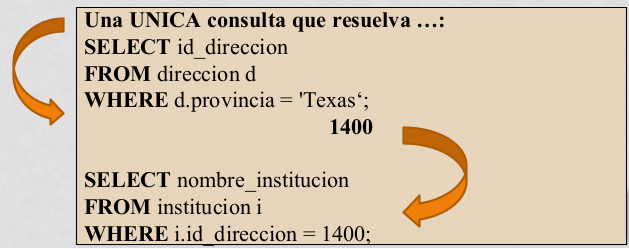


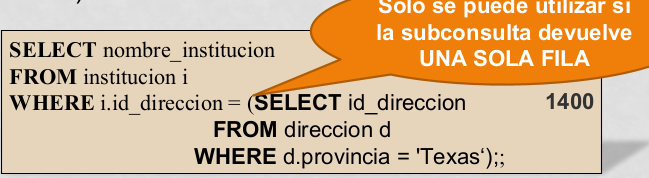
**Consultas anidadas**

La cláusula WHERE puede contener un SELECT anidado, como una consulta conjunta en 2 pasos.

**Ejemplo:**

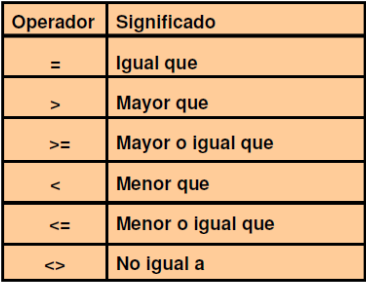
Seleccionar el nombre de la/s instituciones del estado (provincia) de Texas.



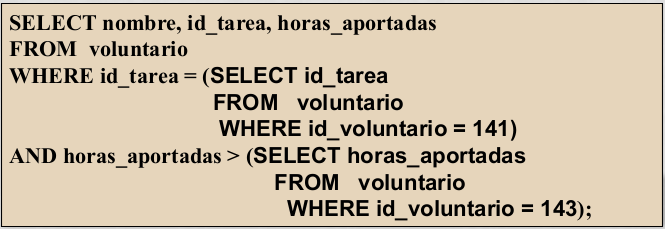


Si hay más de una fila para “Texas”, hay que utilizar operadores para subqueries que retornan **multiple-rows** (IN, ANY, ALL).

**Subconsultas de una sola fila**

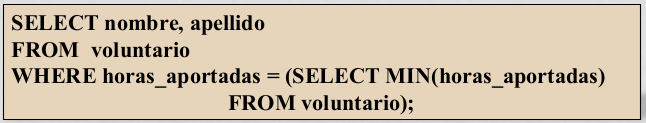
 Es posible utilizar los siguientes operadores de comparación:

**Ejemplo:**

Se desea seleccionar los voluntarios que realizan la misma tarea que el voluntario 141 y que aportan mas horas que el voluntario 143.

**Ejemplo con uso de funciones de grupo:**

Seleccionar todos los voluntarios que aportan la mínima cantidad de horas:

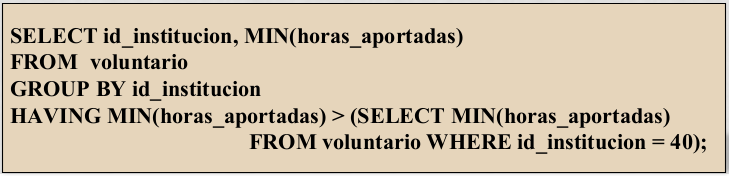


Es responsabilidad de quien escribe el query asegurar que el subquery devolverá una sola fila. Si el subquery devuelve mas de 1 fila, da error.

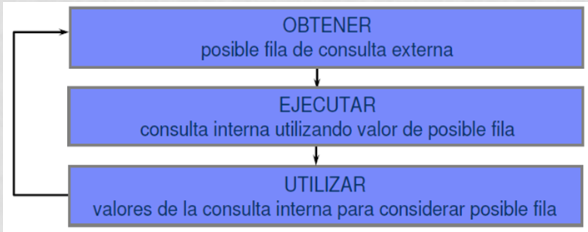
**Clausula HAVING en subconsultas**

Se ejecuta en primer lugar la subconsulta. Devuelve resultados a la clausula **HAVING** que luego se usaran para chequear la condición de grupo.

**Ejemplo:**

Instituciones donde la **mínima cantidad de horas que aportan sus voluntarios** es mayor que la mínima cantidad de horas que aportan los de la institución 40.

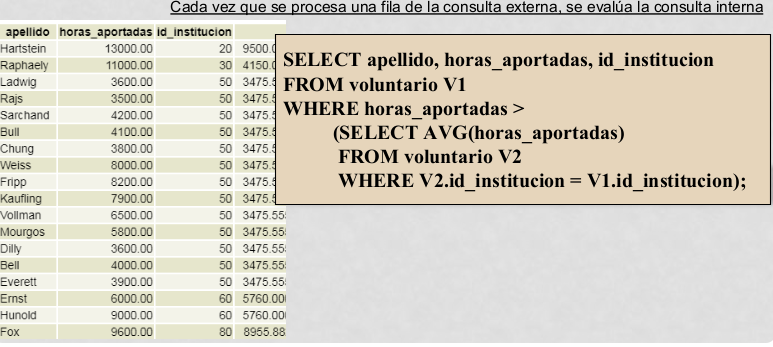
**Subconsultas correlacionadas**

Se utiliza para el procesamiento fila a fila. Cada subconsulta se ejecuta una vez por cada fila de la consulta externa

**Uso de subconsulta correlacionada**

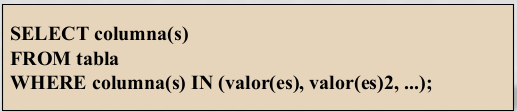
**Ejemplo:**

Buscar todos los voluntarios que aportan mas horas que el promedio de horas aportadas por los voluntarios de la institución a la que pertenecen.



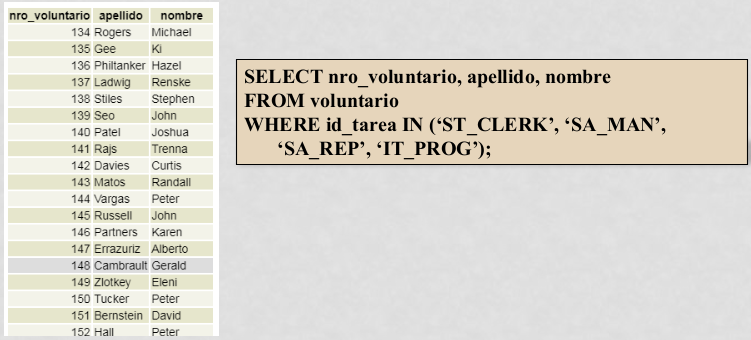
**Operador IN**

El operador IN permite especificar múltiples valores en una clausula WHERE. Podría decirse que es una forma abreviada de varias condiciones OR.

La forma es:

**Ejemplo:**

listar los voluntarios que realizan las tareas ST\_CLERK, SA\_MAN, SA\_REP o IT\_PROG



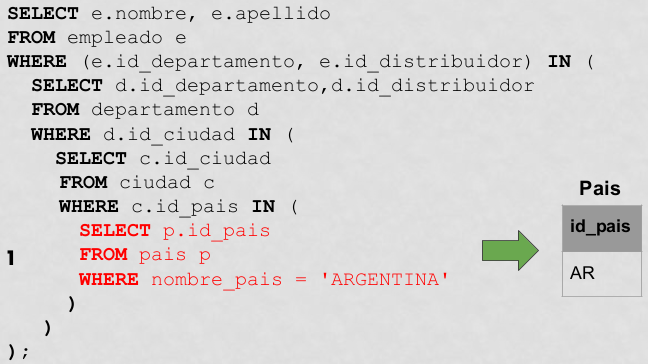
**Uso del operador IN**

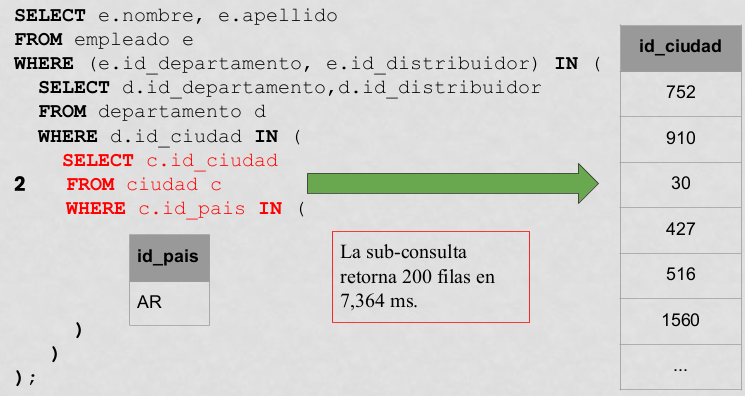
La lista de valores puede ser reemplazada por una subconsulta (consulta interna), que retorna una lista de valores que la consulta exterior luego emplea para recuperar los datos

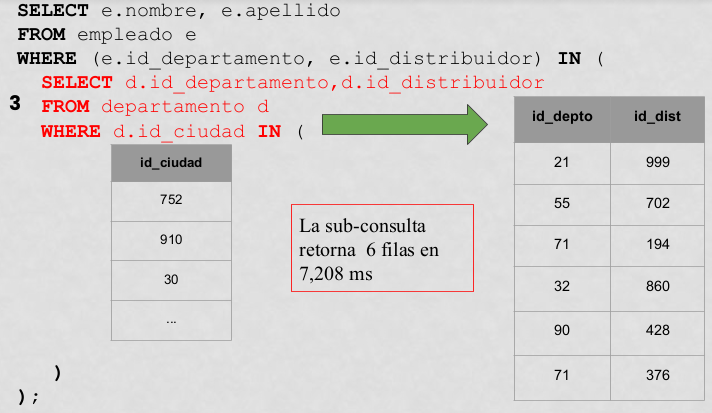
Se recomienda probar las subconsultas antes de incluirlas en una consulta exterior, así puede verificar que retorna lo necesario, porque a veces resulta difícil verlo como consultas anidadas.

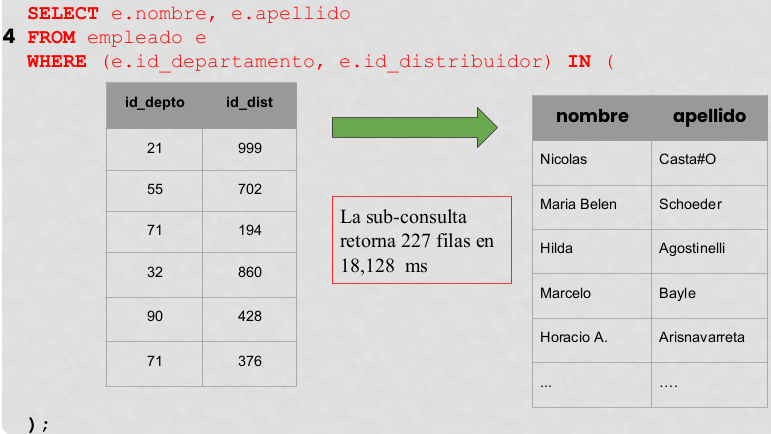
**Ejemplo:**

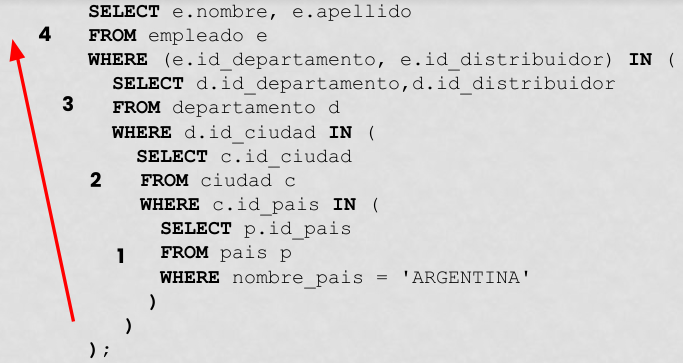
Liste el apellido y nombre de los empleados que trabajan en departamentos de Argentina.





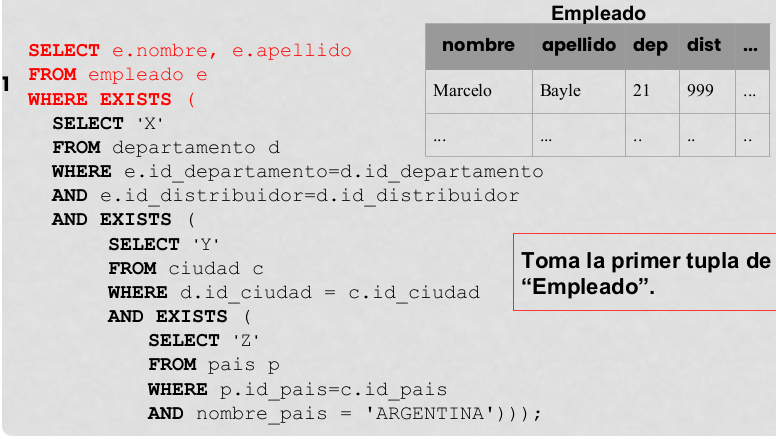


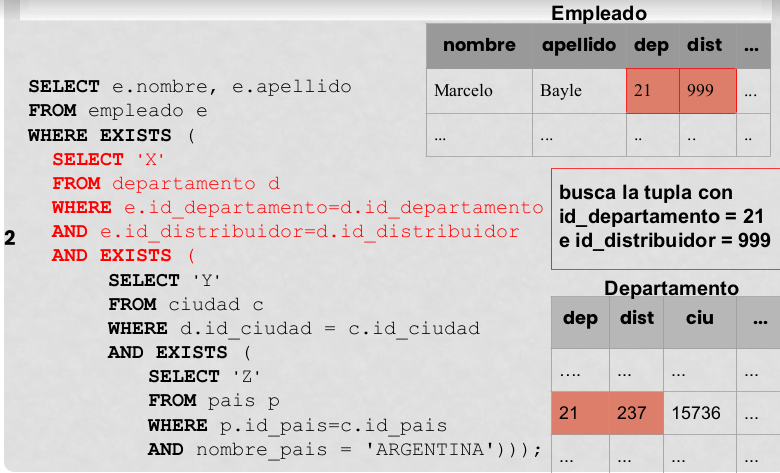


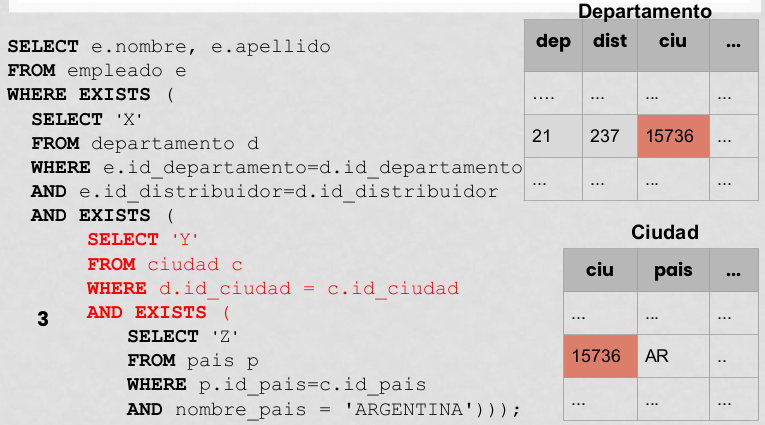


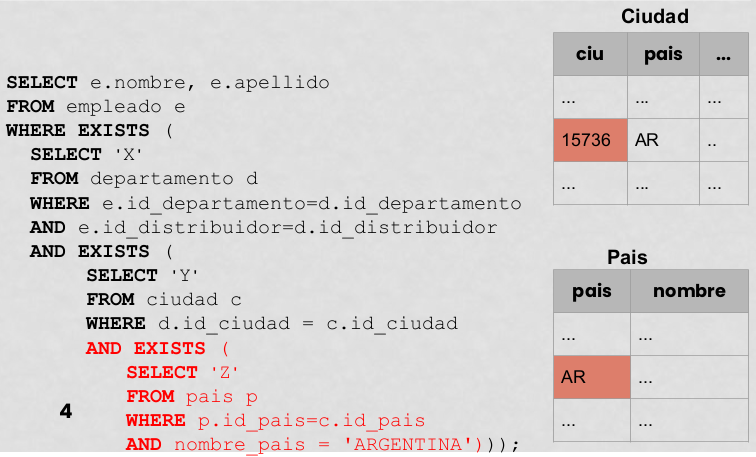
**Uso del operador EXISTS**

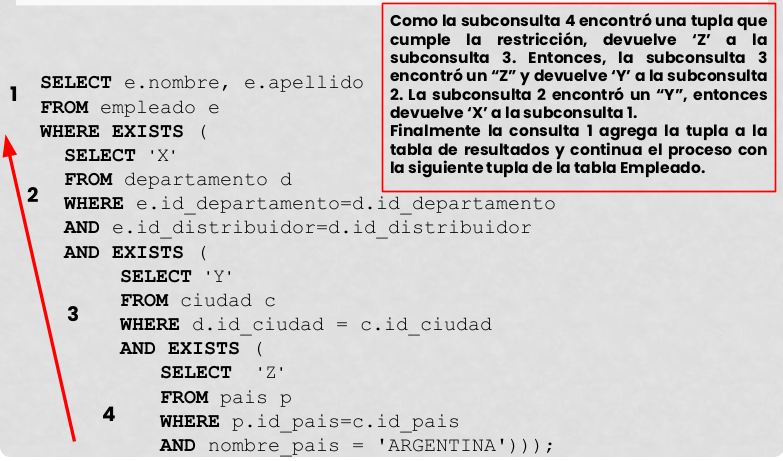
* El operador **EXISTS** comprueba la existencia de filas en el conjunto de filas del resultado de la consulta.
* Si se encuentra un valor de fila de la subconsulta:
  + La búsqueda no continua en la consulta interna.
  + Se señaliza a la condición como **TRUE**.
* Si no se encuentra un valor de fila de la subconsulta:
  + Se señaliza a la condición como **FALSE.**
  + La búsqueda continua en la consulta interna.













**Ensambles (JOIN)**

El operador **JOIN** combina dos tablas según una condición para obtener registros compuestos por atributos de las dos tablas combinadas. Existen diferentes maneras de hacerlo: **JOIN INTERNO.**

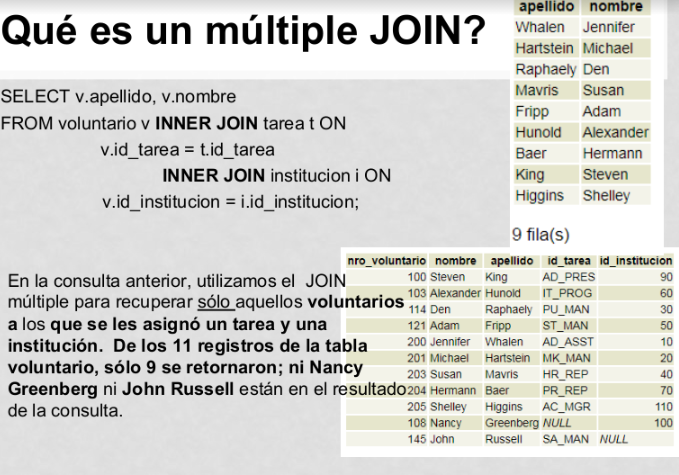
* **INNER JOIN** donde la condición que acota el resultado es una comparación de igualdad.
* **NATURAL JOIN:** Es un caso especial de equi-join en el que en el caso de existir columnas con el mismo nombre en las relaciones que se combinan, solo se incluirá una de ellas en el resultado de la combinación.
* Si los nombres de columnas se repiten, hay que anteponer el nombre de la tabla para evitar ambigüedades.

El tipo de JOIN anterior solo se queda con los registros que tienen valores iguales en las columnas que compara.

Pero puede suceder que se pierdan los registros que nos interesan de alguna de las dos tablas por ejemplo si quisiéramos recuperar todos los voluntarios.

Por eso es que SQL dispone de un **JOIN EXTERNO (OUTER JOIN)**

**Múltiple JOIN**

****

**JOIN**

Los operadores JOIN se realizan de izquierda a derecha.

Primer paso, las tablas del primer JOIN coinciden (tablas voluntario y tarea). Como resultado, se crea una tabla intermedia.

Segundo paso, esta tabla intermedia (tratada como la tabla izquierda) se une con la otra tabla (tabla institución) utilizando el segundo JOIN.

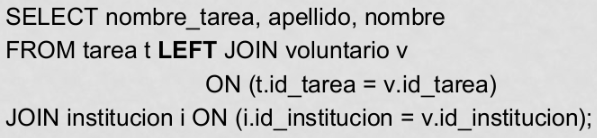
Un solo JOIN produce una sola tabla intermedia durante una consulta de varios JOINs.

**LEFT-RIGHT-FULL JOIN**

Es posible hacer diferentes tipos de combinaciones en una consulta de ensambles múltiple.

***Supongamos que queremos consultar todas las tareas que realizan los voluntarios de alguna institución y también las tareas que no realiza nadie.***

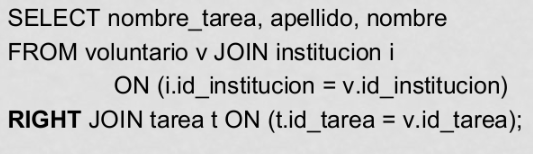
Se podría hacer esto:



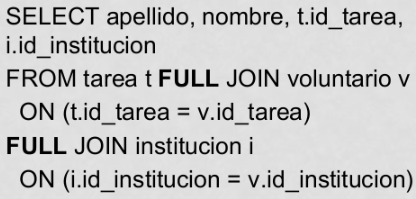


¿Qué paso? La tabla derivada producto del LEFT JOIN entre tarea y voluntario, posee **todas** las tareas, pero al ensamblarla con institución (**INNER JOIN**) quedan solo las que coinciden en el id\_institucion.

¿Como se resuelve ese problema?



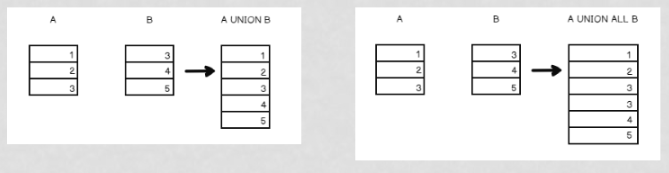
Para el caso FULL JOIN completara con nulos para aquellas tareas que no sean realizadas por ningún voluntario, al igual que las instituciones.



**Clausula UNIÓN**

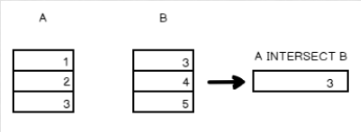
Unión combina los resultados de dos o mas consultas en un único conjunto de resultados que incluye todas las filas que pertenecen a todas las consultas en la Unión.

Hay dos opciones para hacerlo UNION y UNION ALL



**Cláusula INTERSECT**

Recupera las filas que son comunes a todas las tablas.



**Cláusula EXCEPT**

Recupera las filas que están en la primera tabla y que no están en la segunda.

