

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
Gestor de contenidos (Alfresco)  
Aplicaciones híbridas

Tareas programadas (Quartz)  
Gestor documental (Alfresco)  
Inversión de control (Spring)


Control de autenticación y  
acceso (Spring Security)  
UDDI  
Web Services  
Rest Services  
Social SSO  
SSO (Cas)

JPA-Hibernate, MyBatis  
Motor de búsqueda empresarial (Solr)  
ETL (Talend)


Dirección de Proyectos Informáticos.  
Metodologías ágiles  
Patrones de diseño  
TDD

BPM (jBPM o Bonita)  
Generación de informes (JasperReport)  
ESB (Open ESB)

**adictos**  
altrabajo



Encuentra un trabajo que te guste y no volverás a trabajar ni un sólo día de tu vida  
*Confucio*



E-mail:

Contraseña:

Entrar


Deseo registrarme

He olvidado mis datos de acceso

[Inicio](#) [Quiénes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#) [Nuestro libro](#)[Charlas](#) [Más](#)

✖ Estás en:

[Inicio](#) [Tutoriales](#) SQL Joins explicados de forma gráfica con diagramas Venn

DESARROLLADO POR:  
 [Francisco Ferri Pérez](#)

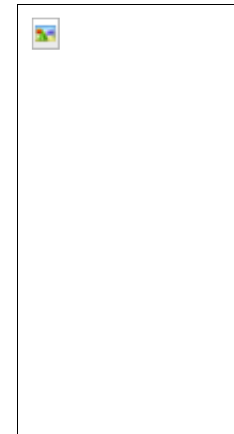
Consultor tecnológico de desarrollo de proyectos informáticos.

Desarrollador de proyectos informáticos

Puedes encontrarme en [Autentia](#):  
Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Catálogo de servicios Autentia

[Anuncios Google](#)[SQL](#)[Inner Joins](#)**Fecha de publicación del tutorial: 2010-09-03**  345[Share](#) |[Regístrate para votar](#)

## SQL Joins explicados de forma gráfica con diagramas Venn

**SQL92 Joins, INNER JOIN, FULL OUTER JOIN, LEFT OUTER JOIN como nunca los habías visto.**

Objetivo del documento:

**Presentar de forma fácil y clara el ámbito de registros dada una entidad que cubren cada una de las sentencias JOIN del SQL92.**

Autor: **Francisco Ferri Pérez**

Fecha de publicación 02/09/2010


**Lecturas recomendadas:**

<http://www.w3schools.com/sql>

Últimas Noticias  
[Tutoriales](#)

 [X Charla Autentia - Talend - Vídeos y Material](#)

 [Comic Flash sobre la decadencia del software](#)

 [Comentando el Libro: Todo va a cambiar de Enrique Dans](#)

 [Java Specialist Master Course](#)

 [Corto sobre Metodologías Ágiles](#)



[Histórico de NOTICIAS](#)

## Una simple ayuda

El lenguaje SQL es aveces el gran olvidado por los desarrolladores, cada vez abundan mas los frameworks que abstraen al desarrollador del contacto con el modelo de negocio.

He escrito este documento, basándome en otros similares para ayudar a entender con un diagrama de Vann, los diferentes tipos de Join's que SQL92 soporta.

Existen tecnologías que abstraen completamente del modelo de negocio, para el desarrollador funcionan creando una serie de clases de Dominio que define el modelo, sin importar qué base de datos o de qué fabricante sea esta. Por ejemplo Ruby on Rails, GRails, ... usando un conjunto de tecnologías, como Hibernate configuradas por [convención](#) dentro del propio framework.

También es muy extendido el uso de aplicaciones que permiten modelar el negocio de forma gráfica, ERM (como DBSchema), y normalmente después se usa un ORM's que les hacen el trabajo sucio de forma elegante y segura.

Todo vale, pero la realidad de las empresas TIC es que necesitan profesionales serios y conscientes que entiendan y controlen todas las capas para que un sistema funciona como se espera y está diseñado, y no se deje nada al azar.

El desarrollador, muy frecuentemente recurre a activar trazas o aplicaciones de monitorización de actividad que nos desvelen, qué está haciendo nuestro framework con nuestro modelo de negocio. Existen aplicaciones que esnifan directamente de la base de datos esta información para que podamos analizarla, el obsoleto Profile y Analyzer que incluía Microsoft SQL Server es un ejemplo de ellos.

Finalmente cuando tenemos delante la query sucede que hace tanto tiempo que no trabajamos con SQL que no entendemos qué hace exactamente, sobre todo si está trabajando con tablas relacionadas y nos encontramos con una consulta que afecta a varias entidades.

Espero que este documento ayude a reducir el tiempo invertido a descubrir porqué se comporta una SQL de un modo u otro.


## Convenciones


Este documento asume que siempre que la "TABLA A" esta a la izquierda y la "TABLA B" a la derecha de las sentencias.

Para los ejemplos vamos a utilizar 2 tablas que van a contener los siguientes datos:

id	name	id	name
1	Roberto	1	Alex
2	<b>Juan</b>	2	<b>Carlos</b>
3	Rubén	3	<b>Juan</b>
4	<b>Carlos</b>	4	Saúl


Últimos Tutoriales


 [Contratos Ágiles y TDD](#)

 [Introducción básica a la herramienta DBSchema](#)

 [MediaWiki -](#)

NamespacePermissions

 [Prey, localizador de dispositivos móviles robados](#)

 [Edición de vídeo \(III\): Añadir un logo de fondo transparente](#)

Últimos Tutoriales del Autor


 [Introducción básica a la herramienta DBSchema](#)

Síguenos a través de:



Últimas ofertas de empleo

2010-08-30  
 [Otras - Electricidad - BARCELONA.](#)

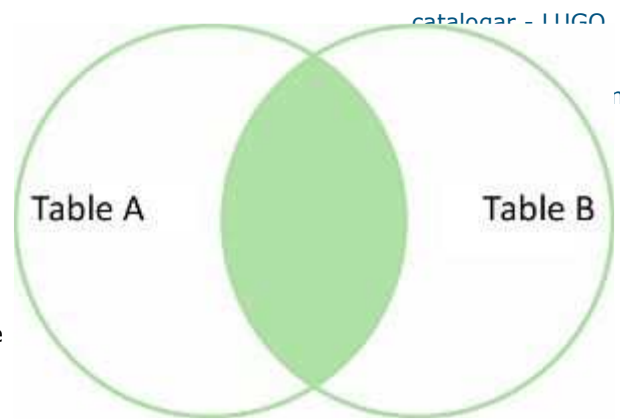
2010-08-24  
 [Otras Sin](#)

**INNER JOIN**

```
SELECT * FROM TableA INNER JOIN TableB ON
TableA.name = TableB.name
```

<u>id</u>	<u>name</u>		<u>id</u>	<u>name</u>
2	Juan		2	Carlos
4	Carlos		3	Juan

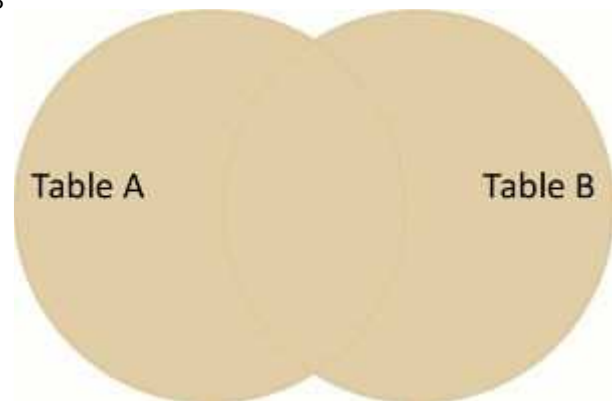
El resultado son solo el conjunto de registros que coinciden en ambas tablas.

**FULL OUTER JOIN**

```
SELECT * FROM TableA FULL OUTER JOIN TableB
ON TableA.name = TableB.name
```

<u>id</u>	<u>name</u>		<u>id</u>	<u>name</u>
1	Roberto		null	null
2	Juan		3	Juan
3	Rubén		null	null
4	Carlos		2	Carlos
null	null		1	Alex
null	null		4	Saúl

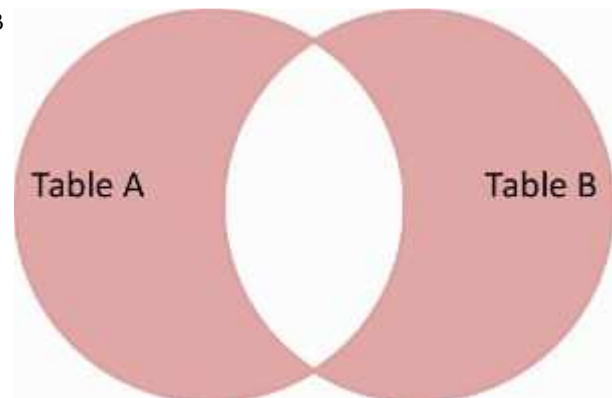
El resultado es el conjunto total de registros de ambas tablas, coincidiendo aquellos registros cuando sea posible. Si no hay coincidencias, se asignan nulos.

**FULL OUTER JOIN WHERE**

```
SELECT * FROM TableA FULL OUTER JOIN TableB
ON TableA.name = TableB.name WHERE
TableA.id IS null OR TableB.id IS null
```

<u>id</u>	<u>name</u>		<u>id</u>	<u>name</u>
1	Roberto		null	null
3	Rubén		null	null
null	null		1	Alex
null	null		4	Saúl

El resultado es un conjunto de records únicos en la TablaA y en la TablaB, hacemos el Full Outer Join y excluimos los registros que no queremos con el Where

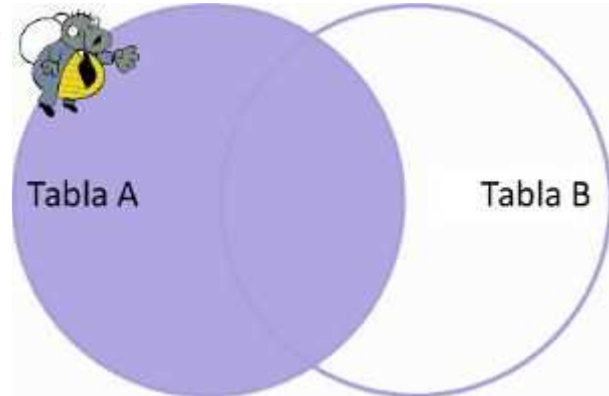


## LEFT OUTER JOIN

```
SELECT * FROM TableA LEFT OUTER JOIN TableB
ON TableA.name = TableB.name
```

<u>id</u>	<u>name</u>		<u>id</u>	<u>name</u>
1	Roberto		null	null
2	Juan		3	Juan
3	Rubén		null	null
4	Carlos		2	Carlos

El resultado son todos los registros de la TablaA, y si es posible las coincidencias de la TablaB. Si no hay coincidencias, el lado derecho mostrará nulos.

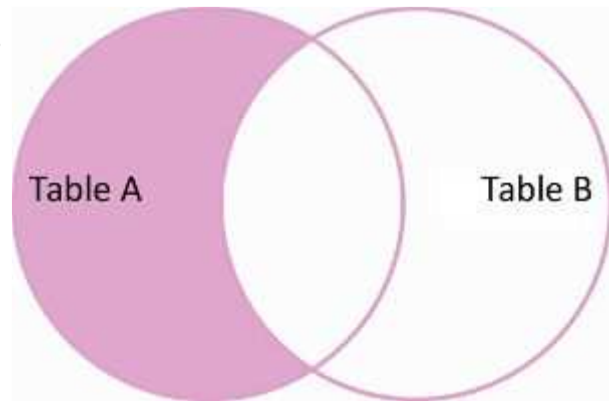


## LEFT OUTER JOIN WHERE

```
SELECT * FROM TableA LEFT OUTER JOIN TableB
ON TableA.name = TableB.name WHERE
TableB.id IS null
```

<u>id</u>	<u>name</u>		<u>id</u>	<u>name</u>
1	Roberto		null	null
3	Rubén		null	null

El resultado es un conjunto de registros que solo están en la TablaA, no en la TablaB. Hacemos lo mismo que en un Left Outer Join, pero eliminamos los registros que no queremos de la TablaB con el Where



## CROSS JOIN

Existe también la posibilidad de cruzar todos los registros con todos (producto cartesiano), imposible de dibujar con un [diagramas Venn](#)



```
SELECT * FROM TableA CROSS JOIN TableB
```

**¡Ojo!** al hacer esto en tablas con muchos registros!

Imaginarse el resultado de todos los registros por todos es muy fácil, si tenemos 4 registros en cada tabla  $4 \times 4 = 16$ .

## Nota del autor

Este documento está escrito con el único fin de informar de forma abierta y gratuita al lector respecto del tema que trata, no está exento de contener imprecisiones o información incorrecta.

Si usted quiere añadir cualquier información al documento porfavor sientase libre de ponerse en contacto con el autor directamente en su correo electrónico [fferri@autentia.com](mailto:fferri@autentia.com). Así todos conseguimos tener la mejor información posible.

Si desea comentar el documento dispone de los comentarios habilitados a continuación del mismo. Estamos encantados de saber que opina o si le ha sido de utilidad.

Muchas gracias por su tiempo, espero que esta lectura le haya sido como mínimo de tanto provecho como para mí fue escribir el documento.

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

## COMENTARIOS



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2010 © All Rights Reserved. [Terms of Use](#) | [Privacy Policy](#) | [Contact Us](#) | [Banners](#) |

