

Capacitación SQL

Nivel inicial

Unidad 3/8

Indice general

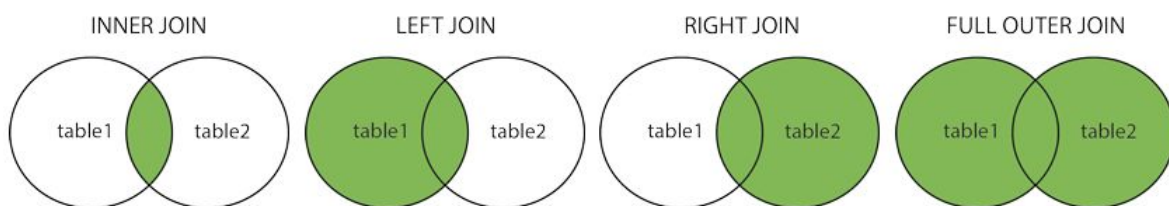
1. Tipos de Joins y Alternativas de sintaxis
 - JOIN o INNER JOIN
 - LEFT JOIN
 - RIGHT JOIN
 - FULL JOIN

1. Tipos de Joins y Alternativas de sintaxis

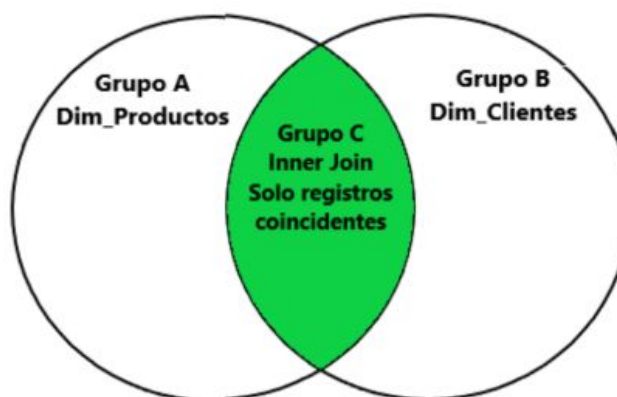
Los JOINS en SQL se utilizan para combinar filas de dos o más tablas, basándose en un campo común entre ellas, devolviendo datos de diferentes tablas, según lo solicitado en la consulta..

A continuación se detallan los diferentes tipos de JOIN en SQL:

- **(INNER) JOIN** : Devuelve registros con valores coincidentes en ambas tablas
- **LEFT (OUTER) JOIN** : Devuelve todos los registros de la tabla izquierda y los registros coincidentes de la tabla derecha
- **RIGHT (OUTER) JOIN** : Devuelve todos los registros de la tabla derecha y los registros coincidentes de la tabla izquierda
- **FULL (OUTER) JOIN** : Devuelve todos los registros cuando hay una coincidencia en la tabla izquierda o derecha



JOIN o INNER JOIN



Una cláusula JOIN o INNER JOIN se utiliza para visualizar registros coincidentes de dos o más tablas, en función de una columna relacionada entre ellas.

Sintaxis:

```
SELECT column_name(s)

FROM table1

INNER JOIN table2

ON table1.column_name = table2.column_name
```

Nota: Como se mencionó en la clase anterior, la utilización de un alias para las tablas será necesario para poder hacer referencia a los campos que se necesitan agregar al resultado de la consulta. Adicionalmente, podría suceder que los campos posean el mismo nombre en ambas Tablas, por lo cual, si no se indica la tabla a la que pertenece el campo, el motor de base de datos no podrá identificar a qué campo se está refiriendo en la consulta.

En la sintaxis, el objetivo de ON es indicar cual es el campo en cada tabla que permite la relación, es decir, que los valores de ambos campos deben ser iguales para que la consulta arroje resultados. Se colocó antes de cada nombre de campo el nombre de la tabla, con el objetivo de hacer referencia a la tabla a la que pertenece cada campo.

A continuación puede observarse una selección de las tablas **"DimProduct"** (Grupo A) y **"FactInternetSales"** (Grupo B). La columna **"ProductKey"** en la tabla **"DimProduct"** se refiere al campo **"ProductKey"** en la tabla **"FactInternetSales"**, esta misma sería la relación entre ambas tablas.

SQLQuery1.sql - qq...dev_user_dw (98))

```

Select Top 5 * From DimProduct
Order by ProductKey desc

Select Top 5 * From FactInternetSales
Order by ProductKey desc
    
```

	ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMeasureCode	SizeUnitMeasureCode	EnglishProductName	SpanishProductName	FrenchProductName	StandardCost
1	606	BK-R19B-52	2	LB	CM	Road-750 Black, 52	Carretera: 750, negra, 52	Vélo de route 750 noir, 52	34
2	605	BK-R19B-48	2	LB	CM	Road-750 Black, 48	Carretera: 750, negra, 48	Vélo de route 750 noir, 48	34
3	604	BK-R19B-44	2	LB	CM	Road-750 Black, 44	Carretera: 750, negra, 44	Vélo de route 750 noir, 44	34
4	603	BB-9108	5	G	NULL	HL Bottom Bracket	Eje de pedalier GA	Axe de pédalier HL	53
5	602	BB-8107	5	G	NULL	ML Bottom Bracket	Eje de pedalier GM	Axe de pédalier ML	44

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	Customer
1	606	20121228	20130109	20130104	22261	2	100	4	SO51187	1	1	...
2	606	20121228	20130109	20130104	12657	1	100	8	SO51188	1	1	...
3	606	20121229	20130110	20130105	21440	1	100	4	SO51199	1	1	...
4	606	20121231	20130112	20130107	21600	1	100	4	SO51241	1	1	...
5	606	20130104	20130116	20130111	29356	1	100	8	SO51297	1	1	...

Nota: La sentencia Inner Join y JOIN es lo mismo, ambos casos arrojan el mismo resultado.

Luego, es posible crear la siguiente instrucción SQL (que contiene una UNIÓN INTERNA), que selecciona registros que tienen valores coincidentes en ambas tablas (Grupo C) en donde se listan los campos SalesOrderNumber, ProductKey, ListPrice, Color y OrderDate, donde el Color del producto es "Silver":

SQLQuery1.sql - qq..._dev_user_dw (98))

```

Select
    B.SalesOrderNumber As "Número de Orden",
    A.ProductKey As Id_Producto,
    A.ListPrice As "Precio de Lista",
    A.Color,
    B.OrderDate As "Fecha de Order"
From DimProduct As A
JOIN FactInternetSales As b
    On A.ProductKey = B.ProductKey
WHERE A.Color = 'Silver'
    
```

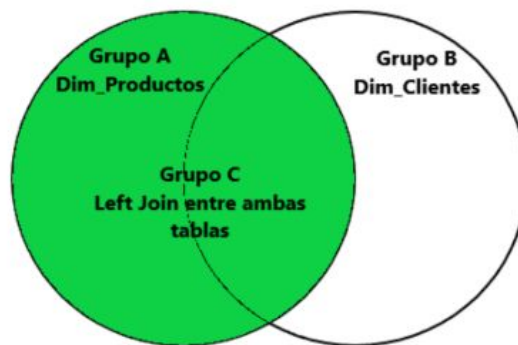
100 %

Results Messages

	Número de Orden	Id_Producto	Precio de Lista	Color	Fecha de Order
1	SO43698	346	3399,99	Silver	2010-12-29 00:00:00.000
2	SO43699	346	3399,99	Silver	2010-12-29 00:00:00.000
3	SO43701	346	3399,99	Silver	2010-12-29 00:00:00.000
4	SO43705	344	3399,99	Silver	2010-12-30 00:00:00.000
5	SO43729	346	3399,99	Silver	2011-01-06 00:00:00.000
6	SO43730	346	3399,99	Silver	2011-01-06 00:00:00.000
7	SO43736	346	3399,99	Silver	2011-01-07 00:00:00.000
8	SO43743	347	3399,99	Silver	2011-01-09 00:00:00.000
9	SO43765	346	3399,99	Silver	2011-01-14 00:00:00.000
10	SO43775	347	3399,99	Silver	2011-01-16 00:00:00.000
11	SO43778	345	3399,99	Silver	2011-01-17 00:00:00.000
12	SO43787	344	3399,99	Silver	2011-01-19 00:00:00.000
13	SO43790	344	3399,99	Silver	2011-01-19 00:00:00.000
14	SO43793	344	3399,99	Silver	2011-01-19 00:00:00.000
15	SO43794	347	3399,99	Silver	2011-01-19 00:00:00.000
16	SO43810	345	3399,99	Silver	2011-01-23 00:00:00.000
17	SO43811	346	3399,99	Silver	2011-01-23 00:00:00.000
18	SO43819	346	3399,99	Silver	2011-01-24 00:00:00.000

Query executed successfully. qq-db-dev

LEFT JOIN



La palabra clave LEFT JOIN devuelve todos los registros de la tabla izquierda **"Dim_Productos"** (Grupo A) y los registros coincidentes de la tabla derecha **"Dim_Clientes"** (Grupo B). El resultado es NULL desde el lado derecho, si no hay coincidencia.

Sintaxis:

```
SELECT column_name(s)

FROM table1

LEFT JOIN table2

ON table1.column_name = table2.column_name
```

Se puede observar una selección de las tablas **"FactInternetSales"** (Grupo A) y **"DimCustomer"** (Grupo B) se puede determinar que la columna **"CustomerKey"** en la tabla **"FactInternetSales"** se refiere al **"CustomerKey"** en la tabla **"DimCustomer"**, esta misma sería la relación entre ambas tablas.

SQLQuery1.sql - qq..._dev_user_dw (98) * - X

```

Select Top 5 * From FactInternetSales
Order by CustomerKey desc

Select Top 5 * From DimCustomer
Order by CustomerKey desc

```

100 %

Results Messages

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	RevisionNumber
1	360	20120909	20120921	20120916	29483	1	100	7	SO49665	1	1
2	358	20120918	20120930	20120925	29482	1	100	7	SO49746	1	1
3	349	20110813	20110825	20110820	29481	1	100	8	SO45427	1	1
4	562	20130717	20130729	20130724	29480	1	98	10	SO62341	1	1
5	479	20130717	20130729	20130724	29480	1	98	10	SO62341	2	1

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	Suffix	Gender	EmailAddress
1	29483	217	AW00029483	NULL	Jésus	L	Navarro	0	1965-06-06	M	NULL	M	jésus9@adventure-works.c
2	29482	179	AW00029482	NULL	Clayton	NULL	Zhang	0	1964-09-01	M	NULL	M	clayton0@adventure-works
3	29481	120	AW00029481	NULL	Ivan	NULL	Suri	0	1965-07-04	S	NULL	M	ivan0@adventure-works.co
4	29480	248	AW00029480	NULL	Nina	W	Raji	0	1977-05-06	S	NULL	F	nina21@adventure-works.c
5	29479	209	AW00029479	NULL	Tommy	L	Tang	0	1969-06-30	M	NULL	M	tommy2@adventure-works.i

Query executed successfully. qq-db-dev.c1kzfhaaypx.us-e... qq_dev_user_dw (98) QG_AdventureWorksDW2017 00:00:01 10 rows

Luego, con la siguiente instrucción SQL (que contiene una UNIÓN IZQUIERDA), se selecciona todos los registros del Grupo A y coincidentes del Grupo B, creando un Grupo C, que selecciona registros donde se listan los registros de los campos CustomerKey, FirstName, OrderDate y ShipDate:

SQLQuery1.sql - qg..._dev_user_dw (98))*

```

Select
    B.CustomerKey As "Id Cliente",
    B.FirstName As Nombre,
    A.OrderDate As "Fecha de Compra",
    A.ShipDate As "Fecha envío de compra"
From FactInternetSales As A
LEFT JOIN DimCustomer As b
    On A.CustomerKey = B.CustomerKey
  
```

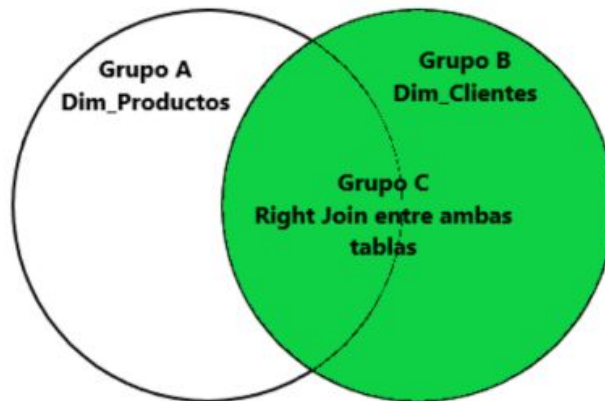
100 %

Results Messages

	Id Cliente	Nombre	Fecha de Compra	Fecha envío de compra
1	21768	Cole	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
2	28389	Rachael	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
3	25863	Sydney	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
4	14501	Ruben	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
5	11003	Christy	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
6	27645	Colin	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
7	16624	Albert	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
8	11005	Julio	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
9	11011	Curtis	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
10	27621	Edward	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
11	27616	Emma	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
12	20042	Brad	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
13	16351	Martha	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
14	16517	Katrina	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
15	27606	Courtney	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
16	13513	Abigail	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
17	27601	Sydney	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000
18	13591	Latasha	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000

Query executed successfully. qg-db-dev.

RIGHT JOIN



La palabra clave RIGHT JOIN devuelve todos los registros de la tabla derecha **"DimCustomer"** (Grupo B) y los registros coincidentes de la tabla izquierda **"FactInternetSales"** (Grupo A). El resultado es NULL desde el lado izquierdo, cuando no hay coincidencia.

Sintaxis:

```
SELECT column_name(s)

FROM table1

RIGHT JOIN table2

ON table1.column_name = table2.column_name
```

Se utilizará el mismo ejemplo de Left Join, para la relación entre las tablas.

Se puede observar una selección de las tablas **"FactInternetSales"** (Grupo A) y **"DimCustomer"** (Grupo B) determinando que la columna **"CustomerKey"** en la tabla **"FactInternetSales"** se refiere al campo **"CustomerKey"** en la tabla **"DimCustomer"**, esta misma sería la relación entre ambas tablas.

Luego, mediante la siguiente instrucción SQL (que contiene una UNIÓN DERECHA), selecciona todos los registros del Grupo A y coincidentes del Grupo B,

creando un Grupo C. Se utilizará el ejemplo anterior, y solamente cambiará la palabra clave, para comprender la diferencia entre ambos.

SQLQuery1.sql - qq..._dev_user_dw (98))*

```

Select
    B.CustomerKey As "Id Cliente",
    B.FirstName As Nombre,
    A.OrderDate As "Fecha de Compra",
    A.ShipDate As "Fecha envío de compra"
From FactInternetSales As A
    RIGHT JOIN DimCustomer As b
    On A.CustomerKey = B.CustomerKey
    
```

100 %

Results Messages

	Id Cliente	Nombre	Fecha de Compra	Fecha envío de compra
1	21768	Cole	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
2	28389	Rachael	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
3	25863	Sydney	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
4	14501	Ruben	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
5	11003	Christy	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
6	27645	Colin	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
7	16624	Albert	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
8	11005	Julio	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
9	11011	Curtis	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
10	27621	Edward	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
11	27616	Emma	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
12	20042	Brad	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
13	16351	Martha	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
14	16517	Katrina	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
15	27606	Courtney	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
16	13513	Abigail	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
17	27601	Sydney	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000
18	13591	Latasha	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000

Query executed successfully. qq-db-dev

En este caso todos los Customer de la tabla tienen órdenes de compra por lo que no se muestran nulos.

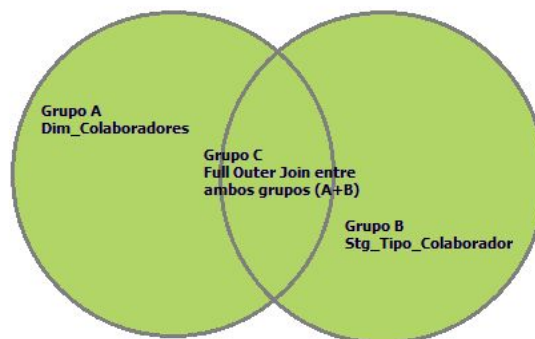
The screenshot shows a SQL query window titled 'SQLQuery1.sql - qq..._dev_user_dw (98)*'. The query is as follows:

```

Select
    B.CustomerKey As "Id Cliente",
    B.FirstName As Nombre,
    A.OrderDate As "Fecha de Compra",
    A.ShipDate As "Fecha envío de compra"
From FactInternetSales As A
RIGHT JOIN DimCustomer As b
    On A.CustomerKey = B.CustomerKey
WHERE OrderDate IS null
    
```

Below the query, the 'Results' tab is active, showing a table with the following columns: 'Id Cliente', 'Nombre', 'Fecha de Compra', and 'Fecha envío de compra'. The table is currently empty.

FULL JOIN



La palabra clave FULL OUTER JOIN o FULL JOIN devuelve todos los registros cuando hay una coincidencia en los registros de la tabla izquierda **"FactInternetSales"** (Grupo A) o derecha **"DimCustomer"** (Grupo B).

Nota: Full outer Join, puede devolver gran volumen de registros. Las palabras claves Full Outer Join y Full Join, son equivalentes. Al devolver todos los registros coincidentes de ambas tablas, habrá registros del grupo A que no coincidan con el grupo B o viceversa, estos igualmente van a ser mostrados.

Sintaxis:

```
SELECT column_name(s)

FROM table1

FULL OUTER JOIN table2

ON table1.column_name = table2.column_name

WHERE condition
```

Se utilizarán las mismas tablas de los ejemplos anteriores, cambiando únicamente las palabras claves.

Luego, se podrá crear la siguiente instrucción SQL (que contiene una UNIÓN COMPLETA), que selecciona todos los registros del Grupo A y coincidentes del Grupo B, o bien todos los registros del Grupo B y coincidentes del Grupo A, creando un Grupo C (en este caso hay 60398 registros devueltos por esta unión):

SQLQuery1.sql - qg..._dev_user_dw (98))*

```

Select
    B.CustomerKey As "Id Cliente",
    B.FirstName As Nombre,
    A.OrderDate As "Fecha de Compra",
    A.ShipDate As "Fecha envío de compra"
From FactInternetSales As A
FULL JOIN DimCustomer As b
    On A.CustomerKey = B.CustomerKey
    
```

100 %

Results Messages

	Id Cliente	Nombre	Fecha de Compra	Fecha envío de compra
1	21768	Cole	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
2	28389	Rachael	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
3	25863	Sydney	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
4	14501	Ruben	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
5	11003	Christy	2010-12-29 00:00:00.000	2011-01-05 00:00:00.000
6	27645	Colin	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
7	16624	Albert	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
8	11005	Julio	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
9	11011	Curtis	2010-12-30 00:00:00.000	2011-01-06 00:00:00.000
10	27621	Edward	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
11	27616	Emma	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
12	20042	Brad	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
13	16351	Martha	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
14	16517	Katrina	2010-12-31 00:00:00.000	2011-01-07 00:00:00.000
15	27606	Courtney	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
16	13513	Abigail	2011-01-01 00:00:00.000	2011-01-08 00:00:00.000
17	27601	Sydney	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000
18	13591	Latasha	2011-01-02 00:00:00.000	2011-01-09 00:00:00.000

Query executed successfully. | qg-db-dev.

----- *Seguí reforzando tu conocimiento en la próxima unidad del programa.