



Arquitectura y Administración de bases de datos con SQL 2021

Héctor Manuel Garduño Castañeda

Diciembre, 2021



Contenido

Álgebra relacional

Inner Join

Left Join

Right Join

Full Outer Join

Cross Join



Motivación del problema



Para resolver el problema anterior, observemos la información que tenemos en cada una de nuestras tablas:

Tabla de clientes (customer)

Data Output	Explain	Messages	Notifications						
<div>customer_id</div> <div>character varying</div>	<div>customer_name</div> <div>character varying</div>	<div>segment</div> <div>character varying</div>	<div>age</div> <div>integer</div>	<div>country</div> <div>character varying</div>	<div>city</div> <div>character varying</div>	<div>state</div> <div>character varying</div>	<div>postal_code</div> <div>integer</div>	<div>region</div> <div>character varying</div>	
1	PG-12430	Claine Gate	Consumer	67	United States	Hamden	Kentucky	40420	South

Tabla de ventas (sales)

Data Output				Explain	Messages	Notifications				
Tabla de Ventas (Sales)										
order_line	order_id	order_date	ship_date	ship_mode	customer_id	product_id	sales	quantity	discount	profit
(PK) Integer	character varying			character varying	character varying	character varying	numeric	integer	numeric	numeric
1	PG-12430	2017-11-05	2017-11-11	Standard	PG-12430	PG-12430	12430	1	0	12430

Tabla de productos (product)

Data Output	Explain	Messages	Notifications
product_id	category	sub_category	product_name
1	PG-12430	PG-12430	PG-12430

Como puede observarse, en la tabla de ventas no aparece la información por región. Las regiones las tenemos en la tabla de clientes.



Supongamos que las tablas contienen únicamente los siguientes registros.

Order Line	Order ID	Order Date	Customer ID	Product ID	Sales
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96
88	CA-2017-155558	26-10-2017	PG-18895	OFF-LA-10000134	6.16
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62
5	US-2015-108966	11-10-2015	SO-20335	OFF-ST-10000760	22.368

Tabla de ventas (sales)

Customer ID	Customer Name	State	Region
CG-12520	Claire Gute	Kentucky	South
DV-13045	Darrin Van Huff	California	West
SO-20335	Sean O'Donnell	Florida	South
BH-11710	Brosina Hoffman	California	West

Tabla de clientes (customer)

Note que:

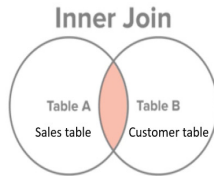
1. CG-12520, DV-13045 y SO-20335 están presentes en las dos tablas.
2. PG-18895 está en ventas pero no en clientes.
3. BH-11710 está en clientes pero no en ventas.



Inner Join

Una unión interna de tablas se refiere a intersectar las tablas. Por lo tanto tendremos como resultado una tabla con **tal vez menos registros**.

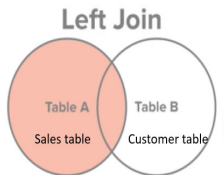
Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335	OFF-ST-10000760	22.368	Sean O'Donnell	Florida	South



Left Join

Una unión izquierda de tablas se refiere a unir los registros de la tabla derecha a la de la izquierda. Por lo tanto tendremos como resultado una tabla con **con los mismos registros que la tabla izquierda y valores nulos**.

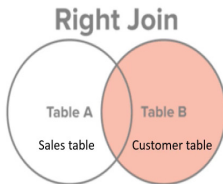
Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
88	CA-2017-155558	26-10-2017	PG-18895	OFF-LA-10000134	6.16	Null	Null	Null
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335	OFF-ST-10000760	22.368	Sean O'Donnell	Florida	South



Right Join

Una unión izquierda de tablas se refiere a unir los registros de la tabla izquierda a la de la derecha. Por lo tanto tendremos como resultado una tabla con **con los mismos registros que la tabla derecha y valores nulos**.

Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335	OFF-ST-10000760	22.368	Sean O'Donnell	Florida	South
Null	Null	Null	BH-11710	Null	Null	Brosina Hoffman	California	West

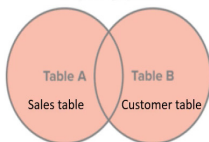


Full Join

Una unión completa de tablas se refiere a unir todos los registros de ambas tablas. Por lo tanto tendremos como resultado una tabla con **con probablemente mas registros que las originales y valores nulos**.

Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
88	CA-2017-155558	26-10-2017	PG-18895	OFF-LA-10000134	6.16	Null	Null	Null
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335	OFF-ST-10000760	22.368	Sean O'Donnell	Florida	South
Null	Null	Null	BH-11710	Null	Null	Brosina Hoffman	California	West

Full Join



El comando **INNER JOIN** *compara* cada renglón de la tabla 1 (izquierda) con cada renglón de la tabla 2 (derecha) y viceversa, para encontrar todos los renglones que tienen en *común*. En realidad, lo que hace es tomar la columna *join* de cada tabla y buscar los valores comunes. Luego, arma una sola tabla uniendo en un solo renglón los correspondientes renglones de 1 y 2 compartiendo las columnas join.

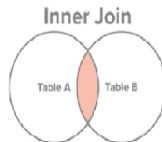
```
SELECT <<columnas>>
FROM <<tabla_1>>
INNER JOIN <<tabla_2>>
ON tabla_1.columna_join = tabla_2.columna_join;
```

col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

col1B	col2B
C	s
E	t
F	u

=

col1A	col2A	col3A	col1B	col2B
C	dato C	q	C	s



Práctica

Mostrar todos los números de orden, identificador de productos, identificador de clientes, ingresos, nombre del cliente y edad de todos los clientes entre 20 y 60 años que realizaron compras en 2015, ordenados por identificador de cliente.

SELECT

a.order_line,
a.product_id,
a.customer_id,
a.sales,
b.customer_name,
b.age

FROM sales_2015 **AS** a

INNER JOIN customer_20_60 **AS** b

ON a.customer_id = b.customer_id

ORDER BY customer_id;



El comando **LEFT JOIN** nos regresa **todas las filas de la tabla izquierda**, aun cuando no exista coincidencia con las de la derecha.

```
SELECT <<columnas>>  
FROM <<tabla_1>>  
LEFT JOIN <<tabla_2>>  
ON tabla_1.columna_join = tabla_2.columna_join;
```

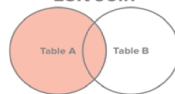
col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

col1B	col2B
C	s
E	t
F	u

=

col1A	col2A	col3A	col1B	col2B
A	dato A	o	NULL	NULL
B	dato B	p	NULL	NULL
C	dato C	q	C	s
D	dato D	r	NULL	NULL

Left Join



Práctica

Mostrar todos los números de orden, identificador de productos, identificador de clientes, ingresos y también, si se dispone, nombre del cliente y edad de todos los clientes entre 20 y 60 años, ordenados por identificador de cliente.

SELECT

a.order_line,
a.product_id,
a.customer_id,
a.sales,
b.customer_name,
b.age

FROM sales_2015 **AS** a

LEFT JOIN customer_20_60 **AS** b

ON a.customer_id = b.customer_id

ORDER BY customer_id;



El comando **RIGHT JOIN** nos regresa **todas las filas de la tabla derecha**, aun cuando no exista coincidencia con las de la izquierda.

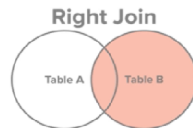
```
SELECT <<columnas>>  
FROM <<tabla_1>>  
RIGHT JOIN <<tabla_2>>  
ON tabla_1.columna_join = tabla_2.columna_join;
```

col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

col1B	col2B
C	s
E	t
F	u

=

col1A	col2A	col3A	col1B	col2B
C	dato C	q	C	s
NULL	NULL	NULL	E	t
NULL	NULL	NULL	F	u



Práctica

Mostrar todos los nombre de clientes y edad de todos los clientes entre 20 y 60 años y, si se dispone, números de orden, identificador de productos, identificador de clientes, ingresos de quienes realizaron compras en 2015, ordenados por identificador de cliente.

SELECT

a.order_line,
a.product_id,
a.customer_id,
a.sales,
b.customer_name,
b.age

FROM sales_2015 **AS** a

RIGHT JOIN customer_20_60 **AS** b

ON a.customer_id = b.customer_id

ORDER BY customer_id;



El comando **FULL JOIN** nos regresa **todas las filas de ambas tablas**, aun cuando no exista coincidencia.

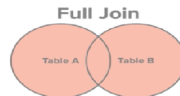
```
SELECT <<columnas>>  
FROM <<tabla_1>>  
FULL JOIN <<tabla_2>>  
ON tabla_1.columna_join = tabla_2.columna_join;
```

col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

col1B	col2B
C	s
E	t
F	u

=

col1A	col2A	col3A	col1B	col2B
A	dato A	o	NULL	NULL
B	dato B	p	NULL	NULL
C	dato C	q	C	s
D	dato D	r	NULL	NULL
NULL	NULL	NULL	E	t
NULL	NULL	NULL	F	u



Práctica

Mostrar todos los nombres de clientes y edad de todos los clientes entre 20 y 60 años y todos los números de orden, identificador de productos, identificador de clientes, ingresos de quienes realizaron compras en 2015, ordenados por identificador de cliente.

SELECT

a.order_line,
a.product_id,
a.customer_id,
a.sales,
b.customer_name,
b.age

FROM sales_2015 **AS** a

FULL JOIN customer_20_60 **AS** b

ON a.customer_id = b.customer_id

ORDER BY a.customer_id, b.customer_id;



El *producto cartesiano* entre dos tablas es una tabla resultado de combinar todas las filas de la tabla 1 con todas las filas de la tabla 2.

SELECT <<columnas>>
FROM <<tabla 1>>, <<tabla 2>>,...

col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

col1B	col2B
C	s
E	t
F	u

=

col1A	col2A	col3A	col1B	col2B
A	dato A	o	C	s
A	dato A	o	E	t
A	dato A	o	F	u
B	dato B	p	C	s
B	dato B	p	E	t
B	dato B	p	F	u
C	dato C	q	C	s
C	dato C	q	E	t
C	dato C	q	F	u
D	dato D	r	C	s
D	dato D	r	E	t
D	dato D	r	F	u



Práctica

Crear una tabla con los años del 2011 al 2021 con cada uno de los meses.

```
CREATE TABLE month(MM int);
```

```
CREATE TABLE year(YYYY int);
```

```
INSERT INTO month VALUES
```

```
(1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12);
```

```
INSERT INTO year VALUES
```

```
(2011),(2012),(2013),(2014),(2015),(2016),(2017),(2018),(2019),(2020),(2021);
```

```
SELECT a.YYYY, b.MM FROM year AS a, month AS b;
```

