



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



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
	customer_id	customer_name	segment	age	country	city	state	postal_code	region			
	character varying	character varying	character varying	integer	character varying	character varying	character varying	integer	character varying			
1	CC-12450	Elisea Gutierrez	Consumer	67	United States	Henderson	Kentucky	42450	South			

Tabla de ventas (sales)

Data	Output	Explain	Messages	Notifications
order_line	[PK] integer			
order_id	character varying			
order_date	date			
ship_date	date			
ship_mode	character varying			
customer_id	character varying			
product_id	character varying			
sales	numeric			
quantity	integer			
discount	numeric			
profit	numeric			

Tabla de productos (product)

	product_id PK character varying	category character varying	sub_category character varying	product_name character varying
1	EID-05-10001709	Furniture	Bookcases	Bush Somerset Collection Bookcase

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:

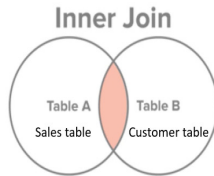
1. CG-12520, DV-13045 y SO-20335 están presentes en las dos tablas.
2. PG-18895 está en ventas pero no en clientes.
3. BH-11710 está en clientes pero no en ventas.



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**.

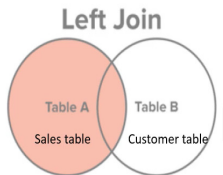
Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335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

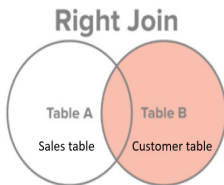
Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
88	CA-2017-155558	26-10-2017	PG-18895	OFF-LA-10000134	6.16	Null	Null	Null
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	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**.

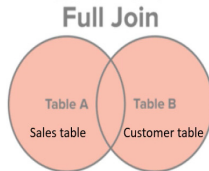
Order Line	Order ID	Order Date	Customer ID	Product ID	Sales	Customer Name	State	Region
1	CA-2016-152156	08-11-2016	CG-12520	FUR-BO-10001798	261.96	Claire Gute	Kentucky	South
3	CA-2016-138688	12-06-2016	DV-13045	OFF-LA-10000240	14.62	Darrin Van Huff	California	West
5	US-2015-108966	11-10-2015	SO-20335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



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.

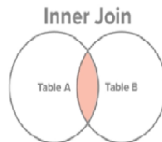
```
SELECT <<columnas>>  
FROM <<tabla_1>>  
INNER JOIN <<tabla_2>>  
ON tabla_1.columna_join = tabla_2.columna_join;
```

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;
```

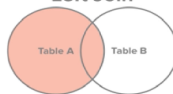
col1A	col2A	col3A
A	dato A	o
B	dato B	p
C	dato C	q
D	dato D	r

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.

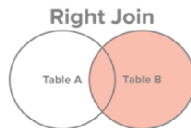
```
SELECT <<columnas>>  
FROM <<tabla_1>>  
RIGHT JOIN <<tabla_2>>  
ON tabla_1.columna_join = tabla_2.columna_join;
```

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;
```

