

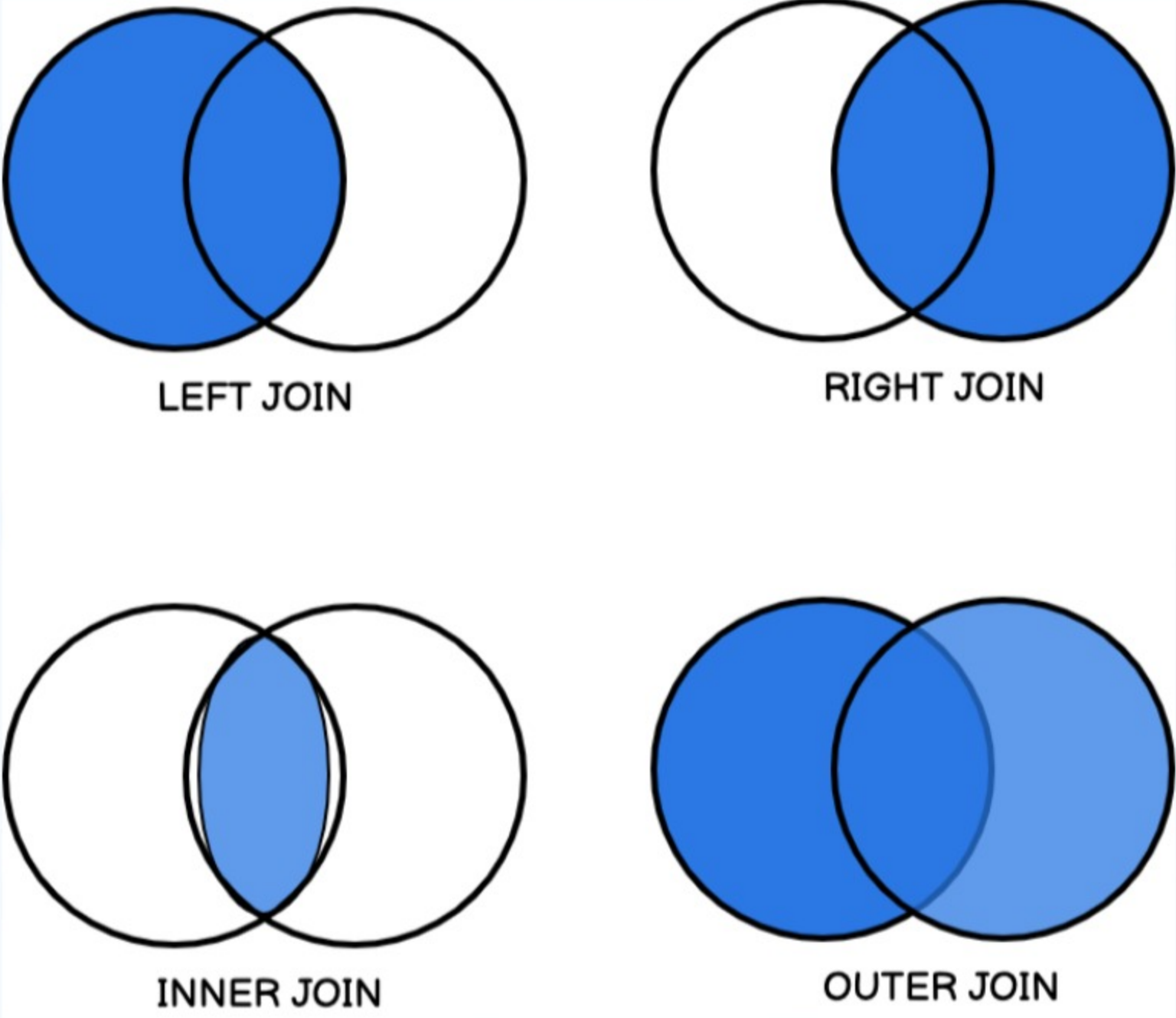
```
SELECT * FROM pedidos
JOIN pedidos_has_productos ON pedidos.id = pedido_id
JOIN productos ON productos.id = producto_id;
```

#En el primer JOIN: pedidos.id se refiere al id de la tabla de pedidos. pedido\_id se refiere a la columna de la tabla pedidos\_has\_productos.

#En el segundo JOIN: productos.id se refiere al id de la tabla de productos. producto\_id se refiere a la columna de la tabla pedidos\_has\_productos.

pedidos					pedidos_has_productos		productos		
id	fecha	cantidad	precio	id_cliente	pedido_id	producto_id	id	nombre	descripción
551	15/07/22	\$500.10	3		551	222	222	lapis	lapis que te ayudará a escribir ERDs
552	10/08/23	\$250.50	2		551	224	224	clip	sostiene tus hojas
553	18/12/23	\$303.13	1		551	227	227	tijeras	recorta todo lo que necesites
554	23/12/23	\$407.00	3		552	223	223	libreta	para escribir todas tus notas de MySQL
					552	225	225	boligrafo	tus notas no se borran con nada
					553	224	224	clip	sostiene tus hojas
					554	224	224	clip	sostiene tus hojas
					554	225	225	boligrafo	tus notas no se borran con nada
					554	226	226	marcatextos	para subrayar lo más importante
					554	227	227	tijeras	recorta todo lo que necesites
union									
id	fecha	cantidad	precio	id_cliente	id	nombre	descripción		
551	15/07/22	\$500.10	3		222	lapis	lapis que te ayudará a escribir ERDs		
551	15/07/22	\$500.10	3		224	clip	sostiene tus hojas		
551	15/07/22	\$500.10	3		227	tijeras	recorta todo lo que necesites		
552	10/08/23	\$250.50	2		223	libreta	para escribir todas tus notas de MySQL		
552	10/08/23	\$250.50	2		225	boligrafo	tus notas no se borran con nada		
553	18/12/23	\$303.13	1		224	clip	sostiene tus hojas		
554	23/12/23	\$407.00	3		224	clip	sostiene tus hojas		
554	23/12/23	\$407.00	3		225	boligrafo	tus notas no se borran con nada		
554	23/12/23	\$407.00	3		226	marcatextos	para subrayar lo más importante		
554	23/12/23	\$407.00	3		227	tijeras	recorta todo lo que necesites		

Los ejemplos que presentamos utilizan un **JOIN** que también se le conoce como **INNER JOIN**. Sin embargo existen otros distintos que pueden ser de gran utilidad. Por ejemplo, al tener una relación de uno a muchos un JOIN nos regresa la información de todos los pedidos y los clientes que realizaron pedidos, sin embargo no nos da información de aquellos clientes que no han realizado pedidos aún. Para incluir en los resultados clientes que no están asociados a ningún pedido utilizamos un **LEFT JOIN**. Te presentamos una representación gráfica de algunos JOINS.



### GROUP BY

En el capítulo anterior aprendimos a manipular valores a través de funciones. Con la sentencia **GROUP BY** podemos agrupar múltiples filas para hacer una acción que combine los valores de estas. El resultado de esta agrupación será una sola fila para el grupo. Te presentamos algunas funciones que puedes utilizar a menudo. A estas les llamamos *Funciones de agrupación* o *Funciones de agregación*.

Función	Resultado
AVG(columna)	Promedio del valor de una columna
COUNT(columna)	La cantidad de valor para una columna
GROUP_CONCAT(columna)	La concatenación de los valores de la columna
MAX(columna)	El valor máximo en una columna
MIN(columna)	El valor mínimo en una columna
SUM(columna)	La sumatoria de todos los valores de la columna

También puedes seguir practicando con este otro arreglo. Descarga el archivo [practica\\_joins\\_2.sql](#)