

Operazioni insiemistiche

1. UNION e UNION ALL

L'operatore **UNION** combina i set di risultati di due o più istruzioni **SELECT** in un singolo set di risultati.

UNION: Rimuove i duplicati.

UNION ALL: Mantiene tutti i duplicati.

```
SELECT 'A', Nome, Reparto FROM Impiegati_A
UNION
SELECT 'B', Nome, Reparto FROM Impiegati_B
ORDER BY Nome;
```

lista unica di tutti i nomi e reparti da entrambe le tabelle

	A-Z ?column?	A-Z nome	A-Z reparto
1	A	Anna Verdi	HR
2	B	Giorgia Neri	Marketing
3	B	Luca Bianchi	IT
4	A	Luca Bianchi	IT
5	A	Mario Rossi	Vendite
6	B	Paolo Gialli	Vendite

Operazioni insiemistiche

2. INTERSECT

L'operatore INTERSECT restituisce solo le righe che sono presenti in entrambi i set di risultati delle istruzioni SELECT.

```
SELECT ID, Nome, Reparto FROM Impiegati_A
```

```
INTERSECT
```

```
SELECT ID, Nome, Reparto FROM Impiegati_B
```

```
ORDER BY Nome,
```

trovare gli impiegati che lavorano sia per il team A che per il team B

	123 id ▼	A-Z nome ▼	A-Z reparto ▼
1	102	Luca Bianchi	IT

Operazioni insiemistiche

3. EXCEPT

L'operatore EXCEPT (chiamato MINUS in Oracle) restituisce le righe che sono presenti nel primo set di risultati ma non sono presenti nel secondo.

```
SELECT ID, Nome, Reparto FROM Impiegati_A  
  
EXCEPT  
  
SELECT ID, Nome, Reparto FROM Impiegati_B  
  
ORDER BY Nome;
```

trovare gli impiegati che sono solo nel team A, e non nel team.
Invertendo le select si ottiene l'elenco degli impiegati che sono solo nel team B, e non nel team A.

	123 id	A-Z nome	A-Z reparto
1	103	Anna Verdi	HR
2	101	Mario Rossi	Vendite

	123 id	A-Z nome	A-Z reparto
1	104	Giorgia Neri	Marketing
2	105	Paolo Gialli	Vendite

Operazioni insiemistiche

Per usare UNION, INTERSECT, ed EXCEPT, le istruzioni SELECT coinvolte devono avere:

Lo stesso numero di colonne.

Le colonne devono avere tipi di dato compatibili (lo stesso tipo, oppure tipi che PostgreSQL può convertire implicitamente).

L'ordinamento (ORDER BY) deve essere applicato alla fine dell'intera operazione insiemistica.

E' la prima query che guida l'aliasing delle colonne

Prodotto cartesiano

L'operazione di Prodotto Cartesiano (detta anche CROSS JOIN) tra due insiemi A e B ($A \times B$) crea tutte le possibili coppie ordinate dove il primo elemento appartiene ad A e il secondo a B.

Se il prodotto cartesiano viene applicato ad una singola tabella, consente di trovare tutte le possibili combinazioni di elementi all'interno di un unico insieme, ad esempio per effettuare confronti oppure realizzare accoppiamenti. (vedere le select relative al calendario calcistico)

JOIN

INNER JOIN

Ritorna solo le righe che fanno match.

Il Match avviene sempre per tutte le tabelle.

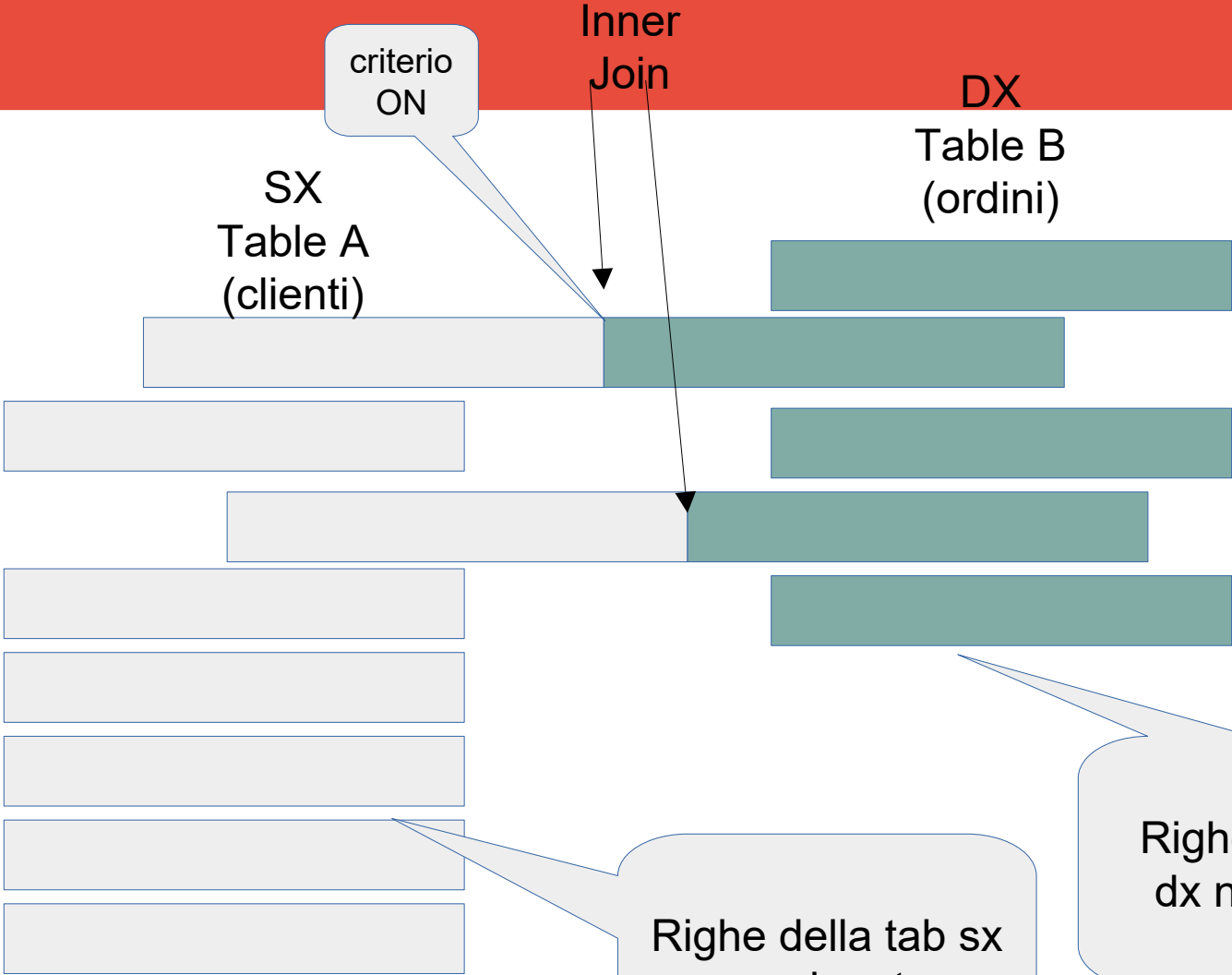
OUTER JOIN

Ritorna tutte le righe di una tabella e le righe della seconda tabella che fanno match.

Le colonne di comparazione devono essere le medesime su entrambe le tabelle.

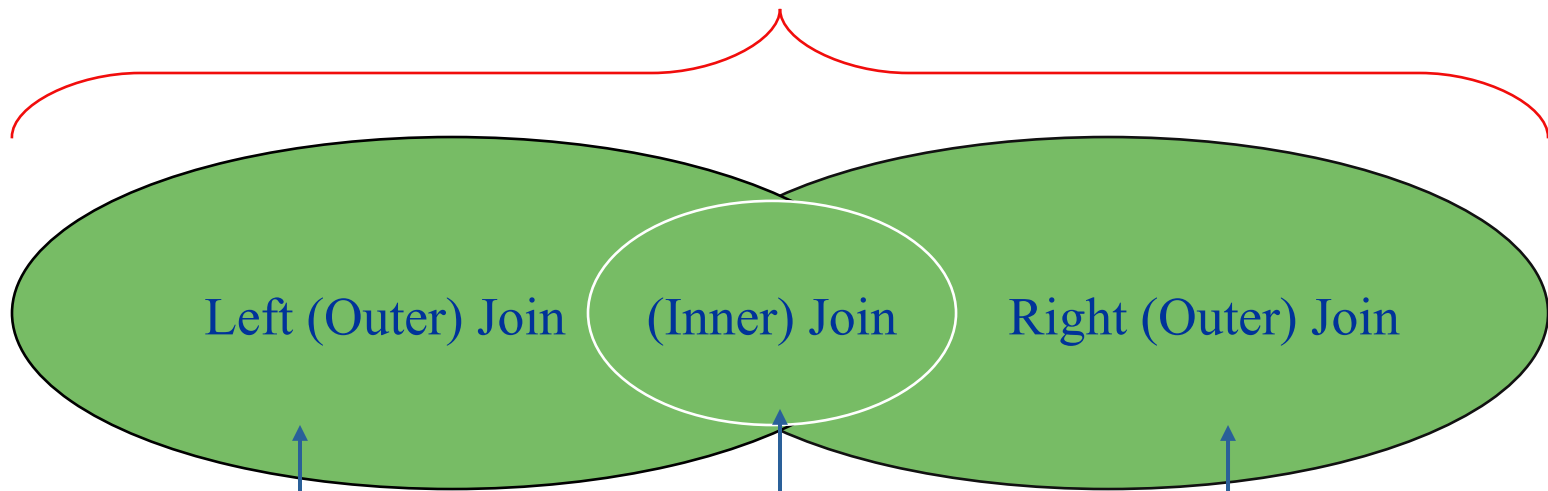


Join



Join

Full (outer) join



Le righe escluse
dalla join della
tabella a sx

Le righe che
soddisfano la
join

Le righe escluse
dalla join della
tabella a dx

Cross Join

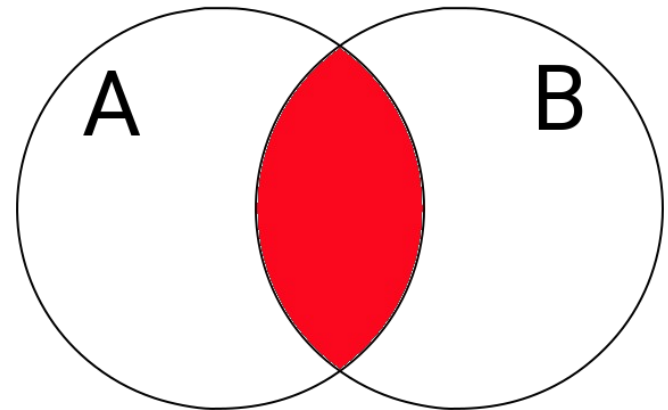
**select nome, cognome, data, valore from clienti a
cross join ordini b;**

Questo tipo di JOIN ritorna il prodotto cartesiano delle righe prese dalle tabelle in Join. Ritorna una table formata dai record combinazione di ogni riga della prima table con ogni riga della seconda tabella table.

Inner Join

```
select a.nome, a.cognome, b.data,  
b.valore from clienti a inner join ordini  
b on a.id_cliente = b.id_cliente;
```

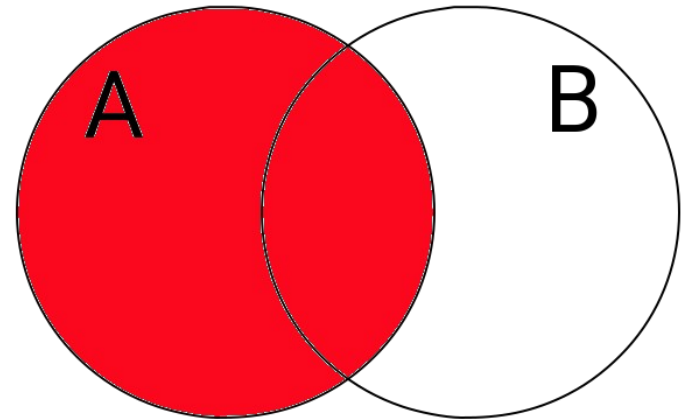
nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2018-11-24 00:00:00	4589.00



Left Join

```
select nome, cognome, data, valore from  
clienti a left join ordini b on  
a.id_cliente = b.id_cliente;
```

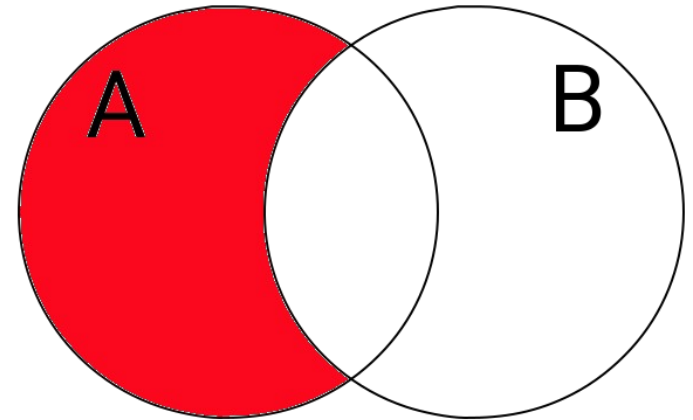
nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL



Left Join nulls

**select nome, cognome, data, valore from clienti a
left join ordini b on a.id = b.id_cliente where
b.data is null;**

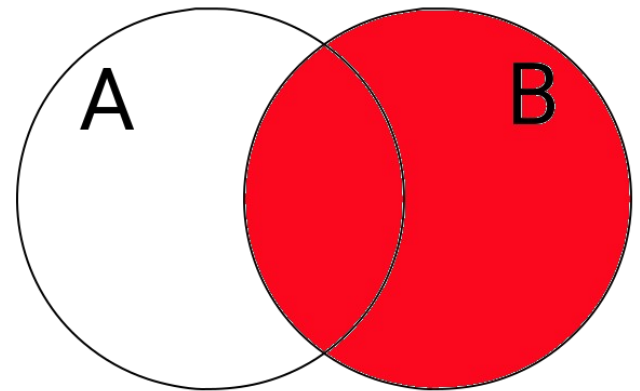
nome	cognome	data	valore
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL



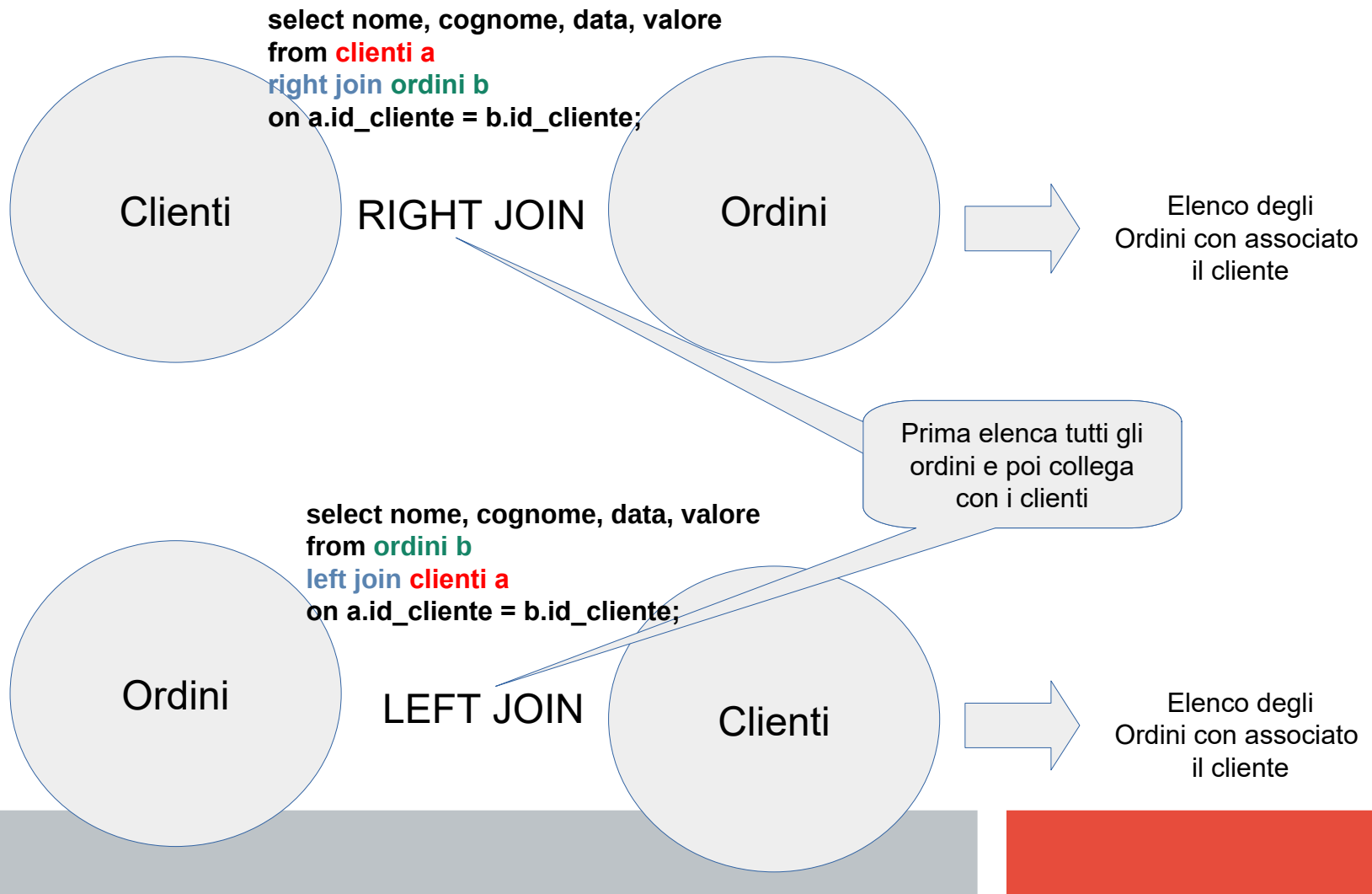
Right Join

**select nome, cognome, data, valore from clienti a
right join ordini b on a.id_cliente = b.id_cliente;**

nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Right Join by Left Join Inversion

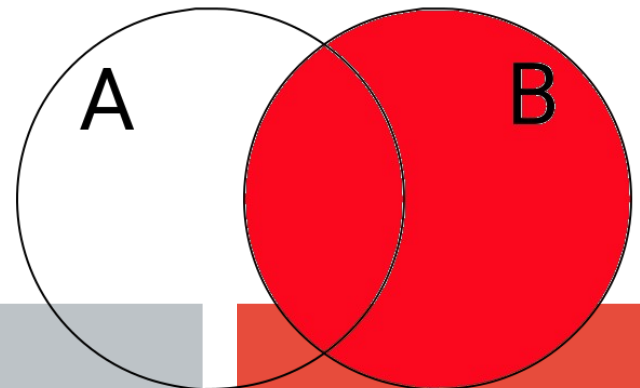


Right Join by Left Join Inversion

```
select nome, cognome, data, valore from  
clienti a right join ordini b on a.id_cliente  
= b.id_cliente;
```

```
select nome, cognome, data, valore from  
ordini b left join clienti a on a.id_cliente  
= b.id_cliente;
```

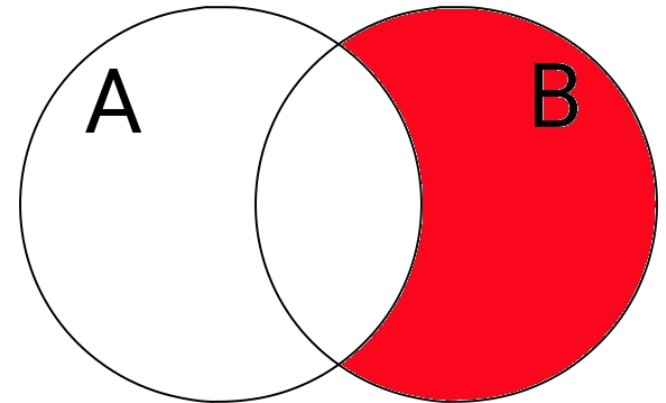
nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Right Join nulls

**select nome, cognome, data, valore from clienti a
right join ordini b on a.id = b.id_cliente where
a.nome is null;**

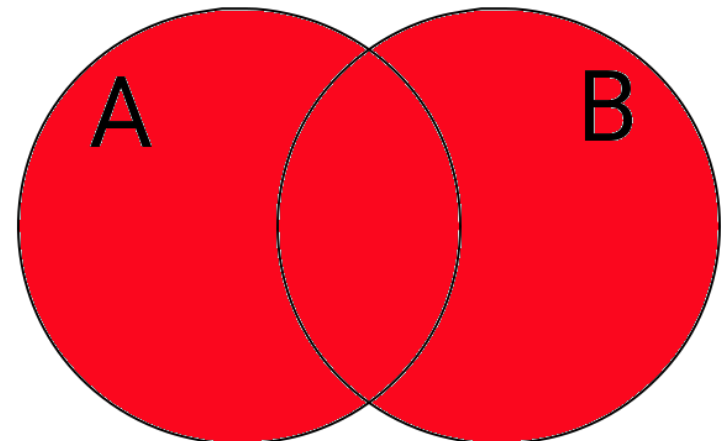
nome	cognome	data	valore
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Full Outer Join

```
select nome, cognome, data, valore  
from clienti a  
full join ordini b  
on a.id_cliente = b.id_cliente;
```

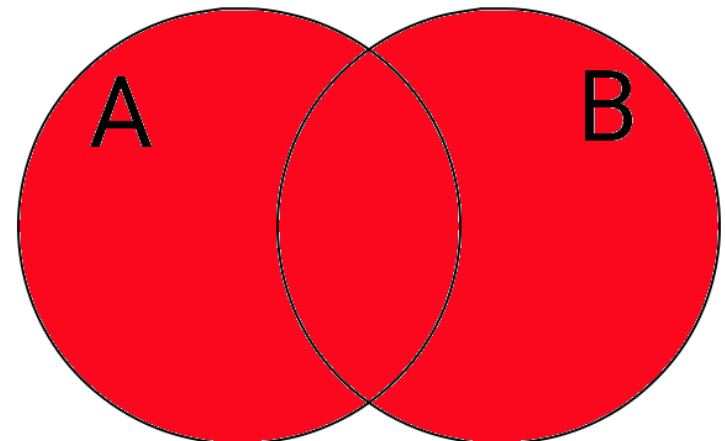
nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Full Outer Join

**select nome, cognome, data, valore from clienti a
left join ordini b on a.id = b.id_cliente **union** select
nome, cognome, data, valore from clienti a right
join ordini b on a.id = b.id_cliente;**

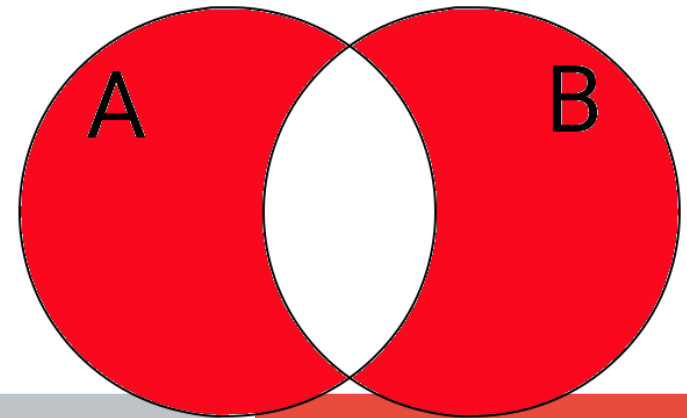
nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Full Join nulls

```
select nome, cognome, data, valore  
from clienti a  
full join ordini b  
on a.id_cliente = b.id_cliente  
where nome is null or data is null;
```

nome	cognome	data	valore
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00

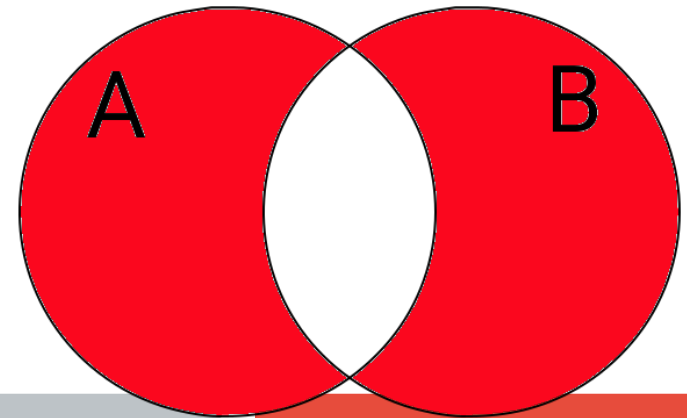


Full Join nulls

**select nome, cognome, data, valore from clienti a left
join ordini b on a.id = b.id_cliente where b.data is null
union**

**select nome, cognome, data, valore from clienti a
right join ordini b on a.id = b.id_cliente where a.nome
is null;**

nome	cognome	data	valore
Gaetano	Donizetti	NULL	NULL
Vincenzo	Bellini	NULL	NULL
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00



Right Join by Left join inversion

```
select nome, cognome, data, valore from  
clienti a right join ordini b on  
a.id_cliente = b.id_cliente;
```

```
select nome, cognome, data, valore from  
ordini b left join clienti a on  
a.id_cliente = b.id_cliente;
```

nome	cognome	data	valore
Giuseppe	Verdi	2018-10-10 00:00:00	345.67
Gioacchino	Rossini	2019-01-01 00:00:00	33.88
Giacomo	Puccini	2017-12-31 00:00:00	176.00
Giacomo	Puccini	2018-11-24 00:00:00	4589.00
NULL	NULL	2018-07-13 00:00:00	230.00
NULL	NULL	2018-06-01 00:00:00	144.00

