ISIS PAOLO SARPI

San Vito al Tagliamento

prof. Cristian Virgili

SQL

Lezione 2

SELECT IN

In SQL, la clausola **IN** viene utilizzata per specificare un elenco di valori da confrontare con un valore di una colonna in una tabella. La clausola IN restituisce i record in cui il valore della colonna corrisponde a uno dei valori specificati nell'elenco.

es1: selezionare tutti i clienti che abitano in una delle seguenti città: Roma, Milano, Udine

```
SELECT * FROM clienti WHERE citta IN ('Roma','Milano','Udine');
```

Selezionare tutti i clienti che hanno fatto un ordine con importo superiore a 100€

```
SELECT * FROM clienti WHERE id IN (SELECT id_cliente FROM ordini WHERE importo>100);
```

LIKE

In SQL, la condizione **LIKE** viene utilizzata per selezionare i record in cui il valore di una colonna corrisponde a un determinato pattern di stringa.

La **sintassi** generale della condizione LIKE è la seguente:
SELECT * FROM tabella WHERE colonna LIKE '**pattern**';
.
Il pattern di stringa può contenere caratteri **jolly** che rappresentano uno o più caratteri, ad esempio:

- **%** rappresenta zero o più caratteri
- _ rappresenta esattamente un carattere

LIKE

- Selezionare tutti i record in cui il nome del cliente **inizia** con "A":

SELECT * FROM clienti WHERE nome LIKE 'A%';

- Selezionare tutti i record in cui il nome del cliente **contiene** la stringa "ella":

SELECT * FROM clienti WHERE nome LIKE '%ella%';

- Selezionare tutti i record in cui il nome del cliente **termina** con "son":

SELECT * FROM clienti WHERE nome LIKE '%son';

- Selezionare tutti i record in cui il nome del cliente è **esattamente** "John":

SELECT * FROM clienti WHERE nome LIKE 'John';

- Selezionare tutti i record in cui il nome del cliente è **esattamente** "John" **o** "Jane":

SELECT * FROM clienti WHERE nome LIKE 'John' OR nome LIKE 'Jane';

La condizione LIKE può essere utilizzata anche con la negazione **NOT** per selezionare i record che non corrispondono al pattern specificato, ad esempio:

SELECT * FROM clienti WHERE nome **NOT LIKE** '%ella%';
Questa query restituirà tutti i record in cui il nome del cliente non contiene la stringa "ella".

IS NULL

In SQL, la condizione **IS NULL** viene utilizzata per selezionare i record in cui il valore di una colonna è nullo (ossia non è stato inserito alcun valore).

La **sintassi** generale della condizione IS NULL è la seguente:
SELECT * FROM tabella WHERE colonna IS NULL;

Selezionare tutti i record in cui il valore della colonna email **è nullo**:
SELECT * FROM clienti WHERE email IS NULL;

Selezionare tutti i record in cui il valore della colonna data_nascita **è nullo**:
SELECT * FROM clienti WHERE data_nascita IS NULL;

Selezionare tutti i record in cui il valore della colonna indirizzo **non è nullo**:
SELECT * FROM clienti WHERE indirizzo **IS NOT NULL**;

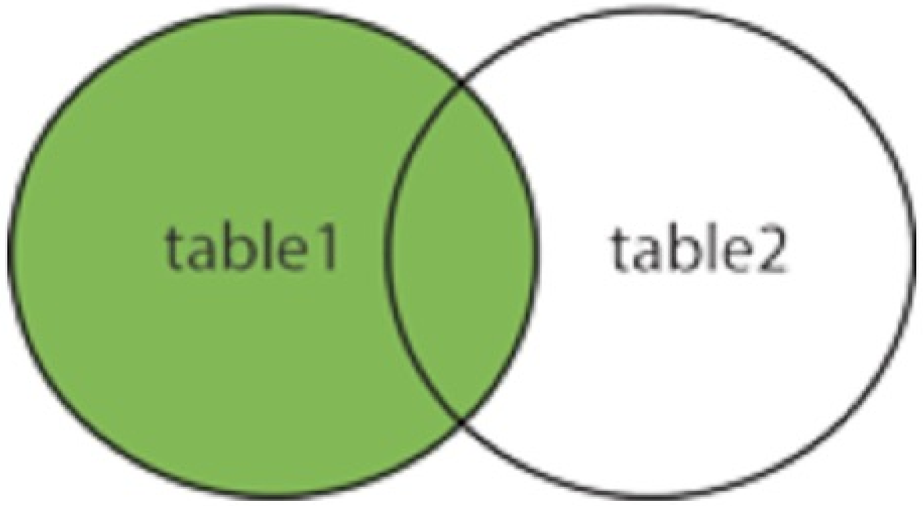
JOIN

In SQL, la clausola **JOIN** viene utilizzata per combinare le righe di due o più tabelle in base alle corrispondenze tra le colonne. La clausola JOIN viene utilizzata insieme alla parola chiave **ON**, che specifica la condizione di unione.

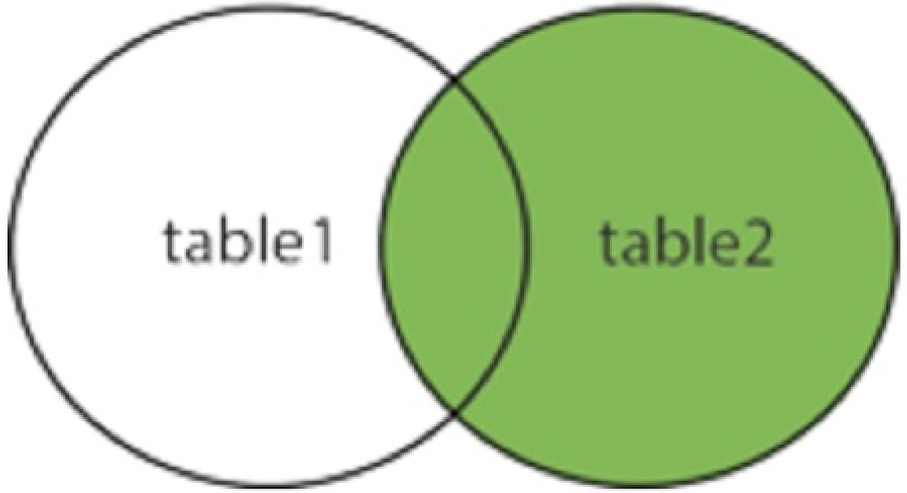
Ci sono diversi tipi di JOIN disponibili in SQL, tra cui:

- **INNER JOIN**: restituisce solo le righe che hanno corrispondenze tra le colonne delle due tabelle.
- **LEFT JOIN:** restituisce tutte le righe della tabella di sinistra e le righe corrispondenti della tabella di destra.
- **RIGHT JOIN**: restituisce tutte le righe della tabella di destra e le righe corrispondenti della tabella di sinistra.
- **FULL JOIN** o **FULL OUTER JOIN**: restituisce tutte le righe di entrambe le tabelle, corrispondenti o meno.

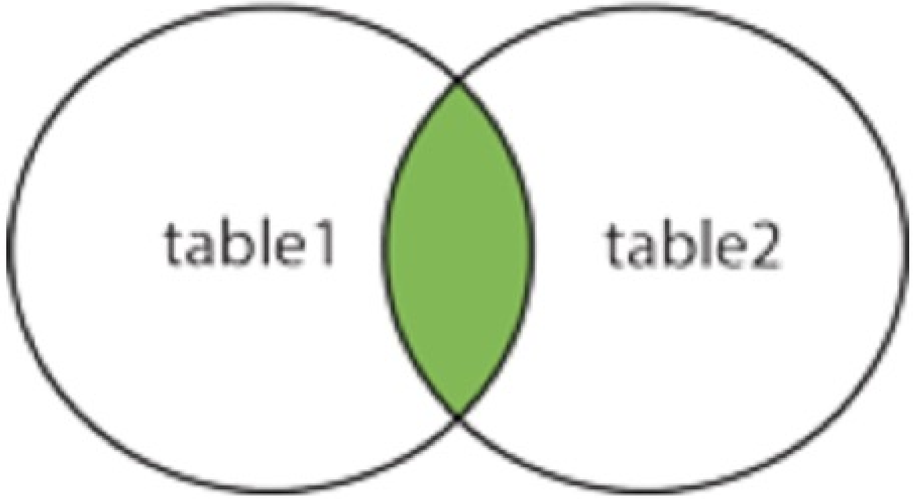
LEFT JOIN



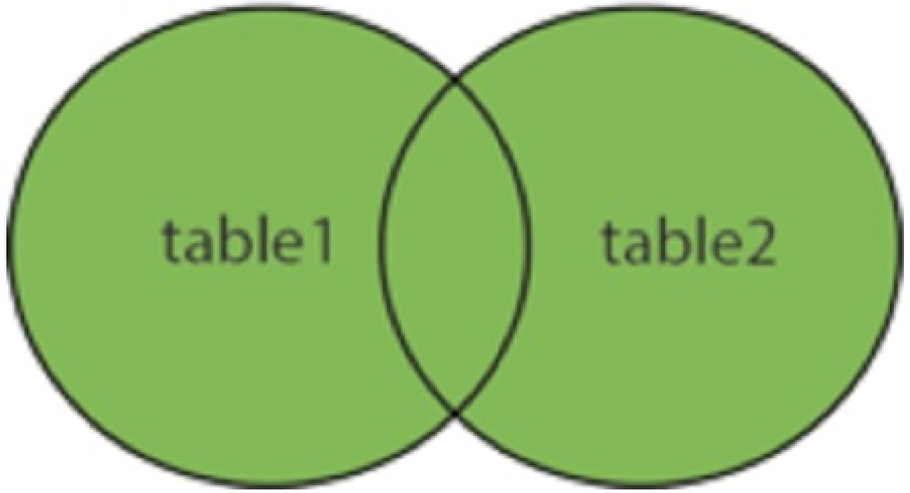
RIGHT JOIN



INNER JOIN

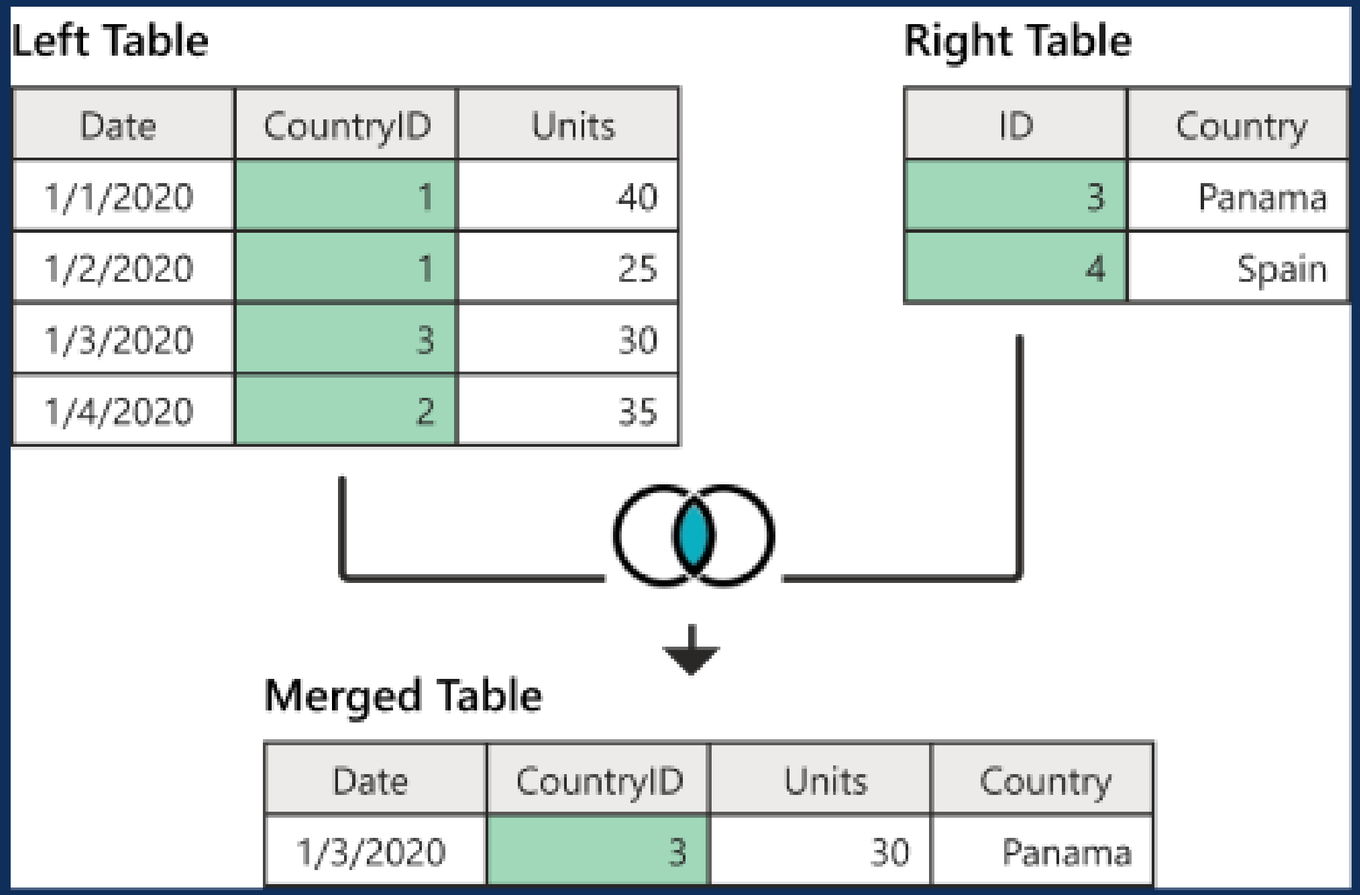


FULL OUTER JOIN



INNER JOIN

```
SELECT l.*, r.* FROM left_table l INNER JOIN right_table r
                ON l.CountryID = r.id;
```



LEFT JOIN

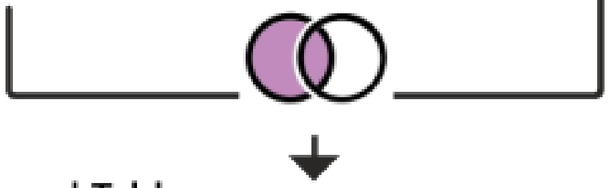
```
SELECT l.*, r.* FROM left_table l LEFT JOIN right_table r
    ON l.CountryID = r.id;
```

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	4	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama

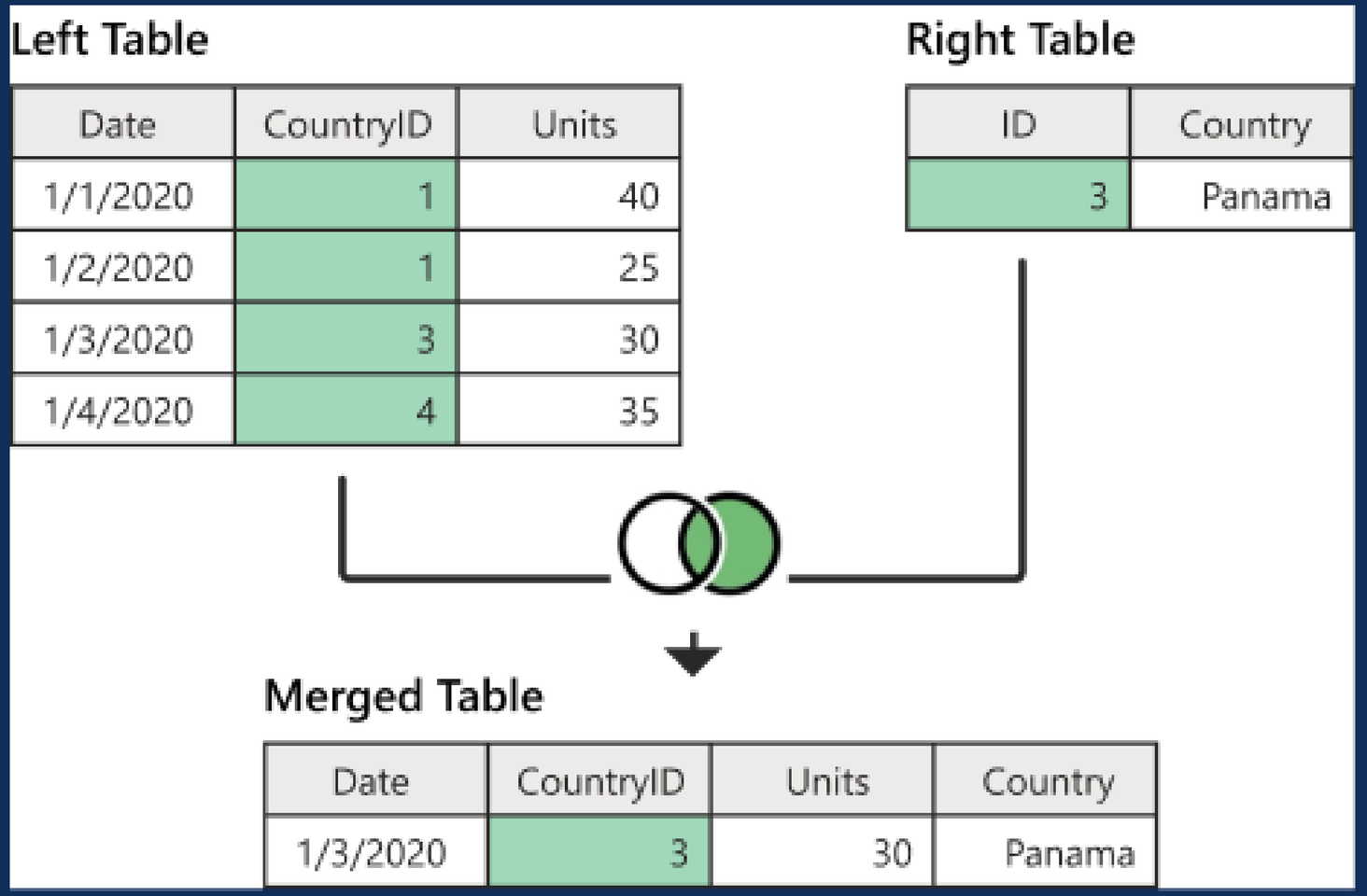


Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/3/2020	3	30	Panama
1/4/2020	4	35	*null*

RIGHT JOIN

```
SELECT l.*, r.* FROM left_table l RIGHT JO JOININ right_table r
                    ON l.CountryID = r.id;
```



FULL JOIN

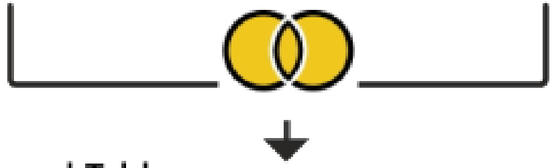
```
SELECT l.*, r.* FROM left_table l  FULL JOIN right_table r
                ON l.CountryID = r.id;
```

Left Table

Date	CountryID	Units
1/1/2020	1	40
1/2/2020	1	25
1/3/2020	3	30
1/4/2020	2	35

Right Table

ID	Country
1	USA
2	Canada
3	Panama
4	Spain



Merged Table

Date	CountryID	Units	Country
1/1/2020	1	40	USA
1/2/2020	1	25	USA
1/4/2020	2	35	Canada
1/3/2020	3	30	Panama
null	*null*	*null*	Spain

Non è supportato da MYSQL

```
SELECT * FROM t1
LEFT JOIN t2 ON
t1.id = t2.id
UNION
SELECT * FROM t1
RIGHT JOIN t2 ON
t1.id = t2.id
```