

Uso delle JOIN

Abbiamo visto nel paragrafo precedente i tre tipi di Relazioni; MySQL offre anche la possibilità di usare, oltre alle semplici Select, le Join sempre per ottenere dati da più tabelle correlate tra loro, ad esempio da una Relazione 1:n.

Esistono diverse tipologie di join, tra cui: la **inner join**, la **outer join** e la **cross join**.

La **cross join** è concettualmente la più semplice, ma non viene usata molto in quanto fa un "prodotto cartesiano" delle due tabelle; ecco come funziona: ogni riga della prima tabella viene combinata con tutte le righe della seconda. Ipotizzando di avere una tabella di 5 righe e una di 6, il risultato sarà una tabella di 30 righe.

La **inner join** si effettua andando a cercare righe corrispondenti nelle due tabelle, basandosi sul valore di determinate colonne; si distinguono in INNER JOIN e LEFT JOIN.

Le **outer join**, come le inner join, vengono effettuate in base alla corrispondenza di alcuni valori sulle tabelle. La differenza è che, nel caso delle outer join, è possibile estrarre anche le righe di una tabella che non hanno corrispondenti nell'altra.

Inner Join

Le query di tipo INNER JOIN ('unione interna') sono le più comuni nonché il tipo predefinito. Corrispondono all'unione cartesiana di due tabelle e all'eliminazione di tutti i record risultanti che non trovano riscontro nelle condizioni della query.

È possibile eseguire query di questo tipo sia in forma esplicita che in forma implicita: la differenza principale è rappresentata dal punto in cui si inseriscono le condizioni dell'unione.

Di seguito è riportato il codice rispettivamente la forma implicita e quella esplicita:

```
SELECT contatti.nome, contatti.cognome, telefoni.telefono FROM contatti, telefoni WHERE contatti.ID = telefoni.ID_persona;
```

```
SELECT contatti.nome, contatti.cognome, telefoni.telefono FROM contatti INNER JOIN telefoni ON contatti.ID = telefoni.ID_persona;
```

Il termine INNER si può omettere, scrivendo quindi una query in cui sia presente solo il termine JOIN. La mancanza di clausole di unione può portare infine ad ottenere una query di tipo **CROSS JOIN** che nella quasi totalità dei casi risulta essere indesiderata e frutto solamente di un errore.

Left Join

Le query di questo tipo permettono di estrarre tutti i record dalla tabella principale (quella specificata appunto a sinistra, prima del termine 'LEFT JOIN') associandoli ai valori collegati provenienti dalla seconda tabella qualora essi siano presenti; nel caso non ci siano valori della seconda tabella corrispondenti a quelli della prima, avrei dei valori nulli.

Vediamo un esempio per chiarire il concetto:

Tabella "Contatti"

ID	Nome	Cognome
1	Andrea	Susi
2	Simone	Rizzo
3	Antonio	De Paolis

Tabella "Telefoni"

ID	Telefono	ID_Persona
1	06681164	1
2	08422021	3

Il risultato sarebbe il seguente se facessi una INNER JOIN (il secondo record non verrebbe visualizzato):

Nome	Cognome	Telefono
Antonio	De Paolis	08422021

mentre se facessi una LEFT JOIN il risultato sarebbe il seguente:

```
SELECT contatti.nome, contatti.cognome, telefoni.telefono FROM contatti LEFT JOIN telefoni ON contatti.ID = telefoni.ID_persona;
```

Nome	Cognome	Telefono
Simone	Rizzo	
Antonio	De Paolis	08422021

Esiste anche la RIGHT JOIN che genera gli stessi risultati prendendo però come tabella principale quella di destra.