

## Calcolo tavole degli accessi (un-official)

### 1. Introduzione

Per calcolare le tavole degli accessi bisogna ragionare sugli indici che stanno alla base delle tabelle. Quando in MySQL viene creata una tabella con il comando *CREATE TABLE* e viene utilizzata la dicitura *PRIMARY KEY*, per indicare che quell'attributo/i è una chiave primaria, l'engine [InnoDB](#), che da MySQL 8.0 è l'engine di default, crea un **indice** in automatico sulla *PRIMARY KEY* ([link documentazione MySQL](#)).

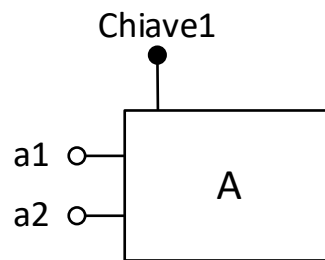
Il calcolo della tavole degli accessi è basato sulle cardinalità massime con cui 2 entità partecipano alla relazione che le collega (si suppone che lo schema E/R rispetti la BCNF, in cui si hanno solo relazioni binarie cioè in cui partecipano due e due sole entità a ciascuna relazione).

Per chiarezza saranno presenti dei disegni annessi alle spiegazioni per favorire la comprensione dei concetti. Inoltre tutte le operazioni andranno dall'entità A verso l'entità B (altrimenti è specificato).

I punti chiave nel calcolo degli accessi sono 2:

- Ragionare per indici → se ho la chiave di una tabella effettuo sempre al massimo 1 accesso puntuale. Al massimo perché se non devo accedere all'entità poiché non mi serve un suo attributo, allora, il numero di accessi sarà nullo (pari a 0)
- Ragionare in termini di schema logico → le cardinalità con cui partecipano due entità a una relazione è fondamentale perché se la relazione che collega le 2 entità (A,B) con cardinalità:
  - **(1:1)** allora a un'occorrenza di A corrisponderà un'occorrenza di B e dunque che vada da A in B o da B in A avrò entrambe le occorrenze di A e di B univocamente determinate (salvo il caso particolare in cui entrambe partecipano opzionalmente alla relazione).
  - **(1:N)** allora a un'occorrenza di A corrisponderanno più occorrenze di B e dunque, se da A vado in B, tutte le occorrenze di B sono determinate da A, viceversa se da B vado in A non so quali occorrenze di B sono associate a ciascuna occorrenza di A perché non ho la chiave della relazione che le collega. Nello schema logico, la chiave va sempre nella relazione che partecipa con cardinalità massima 1 e allora la chiave sarà in A. In questo caso farò sempre un full scan della relazione tra A e B quando passo da B in A e invece accederò puntualmente alla relazione quando vado da A in B.
  - **(N:N)** in questo caso per sapere quali occorrenze di A corrispondono a quali occorrenze di B avrò bisogno della coppia di chiavi, una di A e una di B. Mi servono entrambe le chiavi perché nella ristrutturazione logica di una relazione (N:N) si va a creare una terza tabella che identifica la relazione (N:N), oltre alle 2 tabelle A e B. Avendo necessità della coppia di chiavi, che parta da A o da B non avrò mai entrambe le chiavi alla partenza (salvo casi estremamente particolari) e dunque farò sempre un FULL SCAN della tabella della relazione che tra A e B.

## 1.1. Singola entità



Quando accedo a un'entità le operazioni che posso fare sono le seguenti:

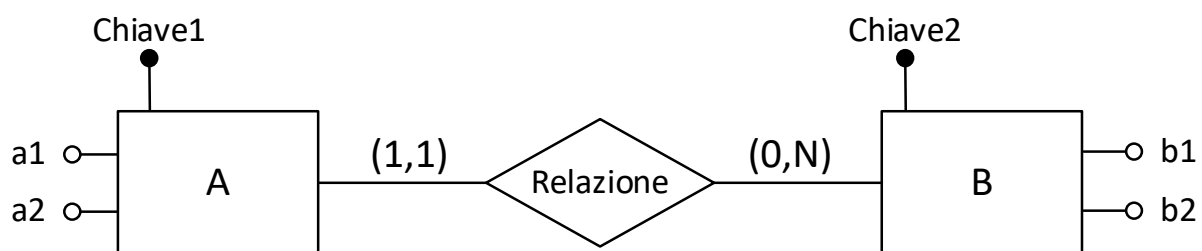
- Verificare che esista un'occorrenza di A dato il valore dell'attributo/i che è **chiave** dell'entità
  - $n^{\circ} \text{accessi} = 1$  (*accesso puntuale*)
- Verificare che esista un'occorrenza di A dato il valore di un attributo che **non sia** la chiave (esempio *a1* oppure *a2*)
  - $n^{\circ} \text{accessi} = n^{\circ} \text{ di occorrenze di A}$  (*full scan*)

Dato che non abbiamo la chiave di A, non possiamo utilizzare l'indice per accedere puntualmente all'occorrenza/e interessata/e, e dunque è necessario andare a scansionare **tutta** la tabella A (tutte le occorrenze) per poter accedere a quelle occorrenze di A che hanno il valore specificato nell'attributo non chiave di A.

In conclusione se ho la chiave effettuo sempre un accesso puntuale altrimenti, se cerco delle occorrenze che hanno un valore specifico, in un attributo non chiave, è sufficiente un full scan della tabella.

## 1.2. Relazioni (1:N)

Le relazioni (1:N) rappresentano il caso più semplice per il calcolo degli accessi nelle relazioni.



Gli accessi vengono calcolati nel seguente modo:

**Se da A andiamo in B**

- Operazione: data la Chiave1 risalire a tutte le occorrenze di B che hanno un valore specifico nell'attributo b1
  - $n^{\circ} \text{accessi in A} = 0$  (*nessuno*)

Non devo effettuare l'accesso in A perché non mi serve nessun attributo di A in questa operazione.

Se questa operazione coinvolgeva uno o più attributi di A  $\Rightarrow n^{\circ} \text{accessi in } A = 1$

Accedo puntualmente perché ho la **chiave** di A.

- $n^{\circ} \text{accessi in Relazione} = \text{Stima}$

*Stima =  $n^{\circ}$  di occorrenze di B che si stima di trovare  
in corrispondenza della chiave A*

Essendo che A partecipa con cardinalità obbligatoria effettuiamo in Relazione un  $n^{\circ}$  di accessi pari al  $n^{\circ}$  di occorrenze di B che ci aspettiamo di trovare in corrispondenza della chiave A specificata.

- $n^{\circ} \text{accessi in } B = 1$

Dato che la relazione è di tipo (1:N) ad ogni occorrenza di A corrisponde una sola occorrenza di B quindi se ho la chiave di A e accedo a Relazione ne identifico univocamente un'occorrenza trovando la rispettiva chiave di B.

#### Se da B andiamo in A

- Operazione: data la Chiave2 risalire a tutte le occorrenze di A che hanno un valore specifico nell'attributo a2

- $n^{\circ} \text{accessi in } B = 0$

Analogo al caso precedente: non devo effettuare l'accesso in A perché non mi serve nessun attributo di A in questa operazione altrimenti se questa operazione coinvolgeva uno o più attributi di B  $\Rightarrow n^{\circ} \text{accessi in } A = 1$

- $n^{\circ} \text{accessi in Relazione} = \text{full scan}$

Non avendo la chiave di A perché B partecipa con cardinalità massima N alla relazione sono costretto a effettuare un FULL SCAN di relazione per determinare quante occorrenze di A corrispondono alla chiave B. Questo caso differenzia dal precedente perché adesso **non ho** entrambe le chiavi come prima e dunque *me le devo andare a prendere* facendo una scansione lineare della tabella Relazione.

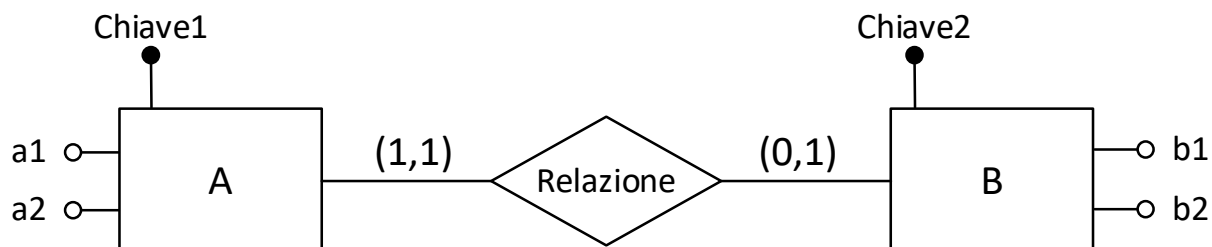
- $n^{\circ} \text{accessi in } A = \text{Stima}$   
*Stima =  $n^{\circ}$  di occorrenze di A che mi aspetto di trovare  
dopo che ho scansionato tutta la tabella Relazione*

Il numero di accessi che andrò a effettuare in A sarà pari a Stima perché devo scansionare tutta le tuple corrispondenti alle chiavi trovate visto che a2 non è chiave di A.

Se invece volevo tutte le chiavi di A corrispondenti alla chiave specificata di B allora non avrei dovuto effettuare nessun accesso A perché le chiavi di A corrispondenti a B le ho già ottenute quando ho scansionato la relazione tra A e B (*Relazione*).

### 1.3. Relazioni (1:1)

Nelle relazioni 1:1 il seguente è il caso più comune:



Nelle relazioni che partecipano con cardinalità (1:1) gli accessi si calcolano nel seguente modo:

- Operazione: Data la **Chiave1** di A si vuole trovare il valore di a1 e si vogliono trovare tutte le occorrenze di B che hanno un valore specifico nell'attributo b2.

- $n^{\circ} \text{accessi in } A = 1$

Effettuo un accesso puntuale in A perché ho la chiave di A e mi serve il valore di un attributo di quell'occorrenza di A.

- $n^{\circ} \text{accessi in Relazione} = 1$

In Relazione effettuo 1 accesso perché per definizione di cardinalità (1:1) a ogni occorrenza di A corrisponde un'occorrenza di B.

- $n^{\circ} \text{accessi in } B = 1$

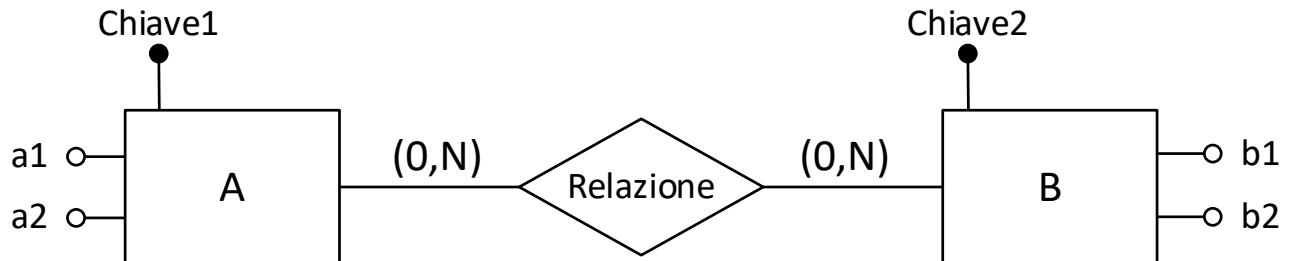
In B effettuo un singolo accesso perché tramite la scansione di Relazione mi sono ricavato la chiave di B e dunque posso accedere puntualmente a B.

Dato che la relazione è di tipo (1:1) effettuerò sempre al massimo un accesso in B perché, per definizione, a ogni occorrenza di A corrisponde al massimo un'occorrenza di B (e viceversa).

Nota: nel caso particolare in cui entrambe le entità partecipano alla relazione con cardinalità minima 0 può avvenire che, in fase di ristrutturazione logica, invece di includere la chiave di un'entità nell'altra, vada a creare una terza tabella che identifica la relazione che le collega. In questo caso, seguendo la logica delle (N:N) dovrò effettuare un FULL scan di Relazione perché in partenza non ho la coppia di chiavi presente in Relazione a seguito della ristrutturazione logica. Invece il n° di accessi nelle entità rimane invariato a quanto detto sopra.

## 1.4. Relazioni (N:N)

Nel calcolo degli accessi le relazioni N:N rappresentano il calcolo più complicato (e più oneroso in termini di n° di accessi) perché non avrò mai in partenza entrambe le chiavi (salvo casi estremamente particolari) della tabella *Relazione* creata in fase di Ristrutturazione logica.



In questo caso non viene fatta la distinzione se parto da A oppure da B perché entrambe le relazioni partecipano con cardinalità massima N e nessuna delle due ha cardinalità massima 1.

Dunque il ragionamento che viene spiegato (in questo caso da A verso B) è il medesimo che se si andasse nel verso opposto (da B verso A).

Nelle relazioni che partecipano con cardinalità (N:N) gli accessi si calcolano nel seguente modo:

- Operazione: Data la **Chiave1** di A si vuole trovare il valore di a2 e si vogliono trovare tutte le occorrenze di B che hanno un valore specifico nell'attributo b2.

- $n^{\circ} \text{accessi in } A = 1$

Effettuo un accesso puntuale in A perché ho la chiave di A e mi serve il valore di un attributo di quell'occorrenza di A.

- $n^{\circ} \text{accessi in Relazione} = \text{full scan}$

Come già detto sopra, non avendo la **coppia di chiavi** che identifica ciascuna tupla (o occorrenza se ragioniamo in termini di progettazione concettuale) sono costretto a effettuare un FULL SCAN di Relazione per andare a cercare tutte le chiavi di B che corrispondono alla chiave di A specificata.

- $n^{\circ} \text{accessi in } B = \text{Stima}$

*Stima = n° di occorrenze di B che stimo di aver trovato  
dopo che ho scansionato tutta la tabella Relazione*

Quelli che vado a effettuare in questo caso sono tutti **accessi puntuali** perché dopo aver scansionato *Relazione* adesso ho le chiavi di B.

In B il n° di accessi corrisponde a una stima di quante occorrenze (a ogni occorrenza corrisponde una e una sola chiave) ritengo di potere aver trovato quando sono andato a scansionare tutta la tabella Relazione. Queste tuple le vado a scorrere tutte perché voglio quelle occorrenze di B che hanno un valore specifico nell'attributo b2.

## 2. Chiarimenti vari e note conclusive

### Singole entità

Dagli esempi soprastanti il lettore dovrebbe aver dedotto che per calcolare gli accessi è fondamentale capire che quando accediamo alla **singola entità** la necessità di un eventuale FULL SCAN è determinata dal fatto che se abbiamo o meno la chiave dell'entità:

- Se abbiamo la **chiave**  $\Rightarrow$  accesso puntuale
- Se non abbiamo la **chiave**  $\Rightarrow$  FULL SCAN

### Relazioni

Quando accediamo alle **relazioni** bisogna ragionare in termini di **progettazione logica**: ogni tupla della relazione che collega le due entità presenta 2 chiavi;

- se la relazione viene accorpata in una delle due entità allora la coppia di chiavi sarà presente in una delle due relazioni dopo la progettazione logica;
- se invece la relazione non viene accorpata ma si crea invece una nuova tabella ecco allora che la coppia di chiavi che identifica ciascuna occorrenza della relazione non è più presente, e dunque, mi ritrovo a dover scansionare tutta questa nuova tabella perché in partenza non è presente più il caso in cui possedevo già la coppia di chiavi (visto che parto sempre da A o da B, entrambe 2 entità).

Per ulteriori esempi consiglio di leggere i progetti caricati su GitHub tra cui quello del sottoscritto sul GitHub di Ing. Informatica nella sezione [Base di dati](#).

### Disclaimer

1. La seguente guida non è da sostituirsi a una spiegazione/ricevimento del professore perché, pur essendo stata scritta da uno studente che ha tenuto un ricevimento con il professore, non sono in primis infallibile, e secondo, per ogni dubbio è sempre importante parlare con il professore.
2. Se trovate eventuali errori e/o correzioni oppure volete ampliare con delle nuove sezioni questo documento, siete liberi di modificare questo documento, a patto che le modifiche siano pensate e ragionate prima di essere scritte.

*Ringrazio lo studente Samuele Riccomini per la revisione di questo documento*