

INSERIMENTO DEI DATI

L'inserimento dei dati in una tabella avviene tramite l'istruzione **INSERT**. Ovviamente dovremo avere il permesso di INSERT sulla tabella.

Vediamo quali sono le diverse sintassi che possiamo utilizzare:

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] nome_tabella [(nome_colonna,...)]
VALUES ({espressione | DEFAULT},...), (...), ...
[ ON DUPLICATE KEY UPDATE nome_colonna=espressione, ... ]
```

Con questa sintassi possiamo inserire una o più righe nella tabella. Prima della clausola VALUES è possibile indicare i nomi delle colonne interessate dalla INSERT: a questi nomi corrisponderanno i valori compresi in ogni parentesi dopo VALUES. Per inserire più righe useremo più coppie di parentesi tonde dopo VALUES. Se non indichiamo la lista delle colonne, dovremo fornire un valore per ogni colonna della tabella, nell'ordine in cui sono definite.

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] nome_tabella
SET nome_colonna={espressione | DEFAULT}, ...
[ ON DUPLICATE KEY UPDATE nome_colonna=espressione, ... ]
```

In questo caso usiamo la clausola SET per assegnare esplicitamente un valore ad ogni colonna indicata. Con questa sintassi è possibile inserire una sola riga.

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] nome_tabella [(nome_colonna,...)]
SELECT ...
[ ON DUPLICATE KEY UPDATE nome_colonna=espressione, ... ]
```

Qui utilizziamo direttamente una SELECT per fornire i valori alla nuova tabella. Con questo sistema si possono inserire più righe. Come nel primo caso, è possibile elencare le colonne interessate; in caso contrario la SELECT dovrà fornire valori per tutte le colonne.

Vediamo ora la funzione delle varie clausole utilizzabili:

LOW PRIORITY: l'inserimento non viene effettuato fino a quando esistono client che leggono sulla tabella interessata; questo può portare ad attese anche lunghe.

DELAYED: anche in questo caso l'inserimento viene ritardato fino a quando la tabella non è libera. La differenza rispetto al caso precedente è che al client viene dato immediatamente l'ok, e le righe da inserire vengono mantenute in un buffer gestito dal server fino al momento della effettiva scrittura. Ovviamente le righe non saranno visibili con una SELECT fino a quando non saranno inserite sulla tabella.

HIGH PRIORITY: annulla l'effetto di una eventuale opzione `–low-priority-updates` che fosse attiva sul server.

IGNORE: permette di gestire eventuali errori che si verificano in fase di inserimento (chiavi duplicate o valori non validi); invece di generare errori bloccanti, i record con chiavi doppie vengono semplicemente scartati, mentre i valori non validi vengono "aggiustati" al valore più prossimo.

ON DUPLICATE KEY UPDATE: nel caso in cui si verifichi una chiave doppia, l'istruzione specificata viene eseguita sulla riga preesistente. Con questa opzione non è possibile usare DELAYED.

I **valori** da inserire nella tabella con le prime due sintassi possono essere indicati da costanti o espressioni, oppure richiamando esplicitamente il DEFAULT. Il default viene usato anche per le colonne non specificate. Tuttavia, in **strict mode**, è obbligatorio specificare valori per tutte le colonne che non hanno un default esplicito; in caso contrario si avrà un errore.

Le colonne di tipo AUTO_INCREMENT vengono valorizzate automaticamente indicando NULL (oppure tralasciandole). Per conoscere il valore generato si può usare, dopo l'inserimento, la funzione LAST_INSERT_ID(), che restituisce l'ultimo valore creato nel corso della connessione attuale.

Oltre alla INSERT, MySQL offre l'istruzione **REPLACE**, che è un'estensione allo standard SQL e che consente di sostituire le righe preesistenti con le righe inserite qualora si verifichi una situazione di chiave doppia. In pratica, usando REPLACE, qualora non sia possibile inserire una riga perchè una PRIMARY KEY o un indice UNIQUE esistono già sulla tabella, MySQL cancella la riga vecchia ed inserisce la nuova. Questo comportamento è opposto a quello di INSERT IGNORE, con il quale è la nuova riga ad essere scartata.

Per effettuare una REPLACE dovremo avere i permessi di INSERT e DELETE; le sintassi sono pressochè identiche a quelle della INSERT.