

GABRIELE ROMANATO

Gestione del tipo di dato booleano in MySQL

In MySQL, il tipo di dato booleano non esiste in modo nativo, ma viene gestito attraverso alias e conversioni. Questo articolo illustra come lavorare con valori booleani nel database.

Uso di TINYINT(1) come Booleano

In MySQL, il tipo `BOOLEAN` è semplicemente un alias di `TINYINT(1)`. Ciò significa che MySQL memorizza i valori booleani come numeri interi, tipicamente 0 per `FALSE` e 1 per `TRUE`.

```
CREATE TABLE utenti (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(50),  
    attivo BOOLEAN -- Alias di TINYINT(1)  
);
```

Nonostante MySQL permetta di dichiarare il tipo come `BOOLEAN`, internamente il valore viene trattato come un numero intero.

Inserimento e Recupero di Valori Booleani

Durante l'inserimento dei dati, è possibile specificare direttamente `TRUE` o `FALSE`, che MySQL convertirà automaticamente in 1 o 0.

```
INSERT INTO utenti (nome, attivo) VALUES ('Mario Rossi',  
TRUE);  
INSERT INTO utenti (nome, attivo) VALUES ('Luigi Bianchi',  
FALSE);
```

Per recuperare i dati booleani:

```
SELECT nome, attivo FROM utenti;
```

Considerazioni sulle Query Boolean

Quando si lavora con condizioni booleane nelle query, è possibile utilizzare direttamente TRUE e FALSE:

```
SELECT * FROM utenti WHERE attivo = TRUE;
```

Oppure, utilizzando la rappresentazione numerica:

```
SELECT * FROM utenti WHERE attivo = 1;
```

Conversione Esplicita

Se necessario, è possibile convertire esplicitamente i valori booleani in numeri interi:

```
SELECT nome, attivo+0 AS attivo_numerico FROM utenti;
```

Questo aiuta a garantire la compatibilità con applicazioni esterne che potrebbero non interpretare direttamente i booleani.

Conclusione

Anche se MySQL non fornisce un vero tipo booleano, la gestione tramite TINYINT(1) e alias come BOOLEAN è sufficiente per la maggior parte delle esigenze. Comprendere il funzionamento interno aiuta a scrivere query più efficienti e a evitare errori nell'elaborazione dei dati.