

Uso degli UUID nei database relazionali

Gli UUID (Universally Unique Identifier) stanno diventando sempre più popolari nei database relazionali per l'identificazione univoca di record. In questo articolo, esploreremo cosa sono gli UUID, i loro vantaggi e svantaggi, e come possono essere utilizzati efficacemente nei database relazionali.

Cosa sono gli UUID?

Un UUID è un identificatore a 128 bit, generalmente rappresentato da una stringa di 36 caratteri in formato testuale. Un tipico UUID appare come una serie di caratteri esadecimali divisi da trattini.

Gli UUID sono progettati per essere unici a livello globale, il che significa che l'identificatore generato in un sistema dovrebbe essere unico rispetto a quelli generati in qualsiasi altro sistema, senza la necessità di coordinamento tra i sistemi stessi.

Vantaggi dell'uso degli UUID nei database relazionali

- 1. Unicità Globale:** Uno dei maggiori vantaggi degli UUID è la loro unicità globale. Questo è particolarmente utile in applicazioni distribuite, dove più database possono essere sincronizzati o dove i dati vengono aggregati da diverse fonti.
- 2. Indipendenza dal Database:** Poiché gli UUID possono essere generati indipendentemente dal database, l'applicazione può creare identificatori univoci senza dover fare affidamento sul database per la

generazione di chiavi primarie, riducendo la necessità di operazioni di rete e migliorando le prestazioni.

3. **Scalabilità:** Gli UUID supportano l'inserimento simultaneo di dati da più nodi senza rischio di collisione, rendendoli ideali per sistemi distribuiti o microservizi.
4. **Sicurezza:** Gli UUID possono anche migliorare la sicurezza impedendo il "guessing" di ID sequenziali, una pratica che potrebbe essere utilizzata per sfruttare vulnerabilità nel sistema.

Svantaggi dell'uso degli UUID

1. **Dimensioni:** Gli UUID occupano più spazio rispetto agli identificatori interi. In un database, dove ogni byte conta, l'uso di UUID può aumentare significativamente lo spazio di archiviazione necessario.
2. **Prestazioni:** L'uso di UUID come chiavi primarie può avere un impatto negativo sulle prestazioni, specialmente in termini di indice e ordinamento. Gli identificatori interi autoincrementali sono molto più efficienti per l'ordinamento e l'indicizzazione rispetto agli UUID, che sono essenzialmente casuali.
3. **Leggibilità:** Gli UUID sono meno leggibili rispetto agli identificatori numerici. Questo può complicare il debug e la gestione manuale dei dati, rendendo più difficile per gli sviluppatori e gli amministratori del database lavorare direttamente con i dati.
4. **Ordinamento non sequenziale:** Poiché gli UUID non sono sequenziali, l'inserimento di nuovi record può portare a frammentazione nell'indice del database, riducendo l'efficienza delle query.

Quando utilizzare gli UUID

L'uso degli UUID è consigliato in scenari specifici:

- **Applicazioni Distribuite:** Quando i dati devono essere sincronizzati tra più sistemi senza un controllo centralizzato sulla generazione degli identificatori.
- **Multi-Tenancy:** In sistemi multi-tenant dove gli identificatori devono essere unici non solo all'interno di un singolo database ma attraverso più database o applicazioni.
- **Sincronizzazione di Dati:** Quando i dati da più fonti devono essere uniti o sincronizzati senza rischio di collisione di chiavi.
- **Microservizi:** In architetture a microservizi, dove i servizi sono indipendenti l'uno dall'altro, ma devono eventualmente comunicare e condividere dati.

Conclusione

Gli UUID rappresentano una potente opzione per la gestione di identificatori unici in database relazionali, specialmente in contesti distribuiti o multi-tenant. Tuttavia, la loro adozione deve essere ponderata attentamente, considerando i compromessi in termini di prestazioni, spazio di archiviazione e leggibilità. In scenari dove le prestazioni e l'efficienza dell'archiviazione sono critiche, gli identificatori interi tradizionali potrebbero rimanere la scelta preferibile. D'altro canto, in un mondo sempre più orientato verso il cloud e le applicazioni distribuite, gli UUID possono offrire la flessibilità e la scalabilità necessarie per affrontare le sfide moderne.