

Die optimierte Abfrage ist schneller, weil sie eine effizientere Methode zum Filtern der Daten verwendet.

Anstelle der NOT IN-Klausel, die alle Film-IDs in der Unterabfrage prüft, verwendet die Abfrage die NOT EXISTS-Klausel, die nur prüft, ob die Film-ID in der Hauptabfrage nicht in der Unterabfrage vorhanden ist. Dadurch wird die Menge der zu verarbeitenden Daten reduziert und die Abfrage beschleunigt.

Außerdem verwendet die Abfrage COUNT(DISTINCT film_id) anstelle von COUNT(film_id), wodurch sichergestellt wird, dass dieselbe film_id nur einmal gezählt wird, wenn sie von demselben Schauspieler wiederholt wird.

Außerdem wird die Abfrage durch die Verwendung von CTEs lesbarer und wartbarer, und die Datenbank kann die Abfrage optimieren, indem sie die Unterabfrage materialisiert und dann in der Hauptabfrage verwendet. Und durch die Verwendung eines Indexes kann die Datenbank beim Verbinden der Tabellen und Filtern der Ergebnisse schnell die relevanten Zeilen finden, was die Abfrageleistung weiter verbessert.