

PostgreSQL ist eine moderne relationale Datenbank unter Open-Source-Lizenz, die vielseitig und weit verbreitet ist.

Bei deren Nutzung will – oder muss – man aus verschiedenen Gründen hin und wieder SQL-Abfragen optimieren. Dabei sollte jedoch erst einmal klar werden, wie das Datenbanksystem die jeweilige Abfrage ausführt, und wie es zu dieser Entscheidung gekommen ist.

Die Entscheidungen, welche PostgreSQL trifft, um eine gegebene SQL-Abfrage beantworten zu können, werden in einem Abfrageplan (engl. Query plan) dargestellt. Da es die Voraussetzung für alles nachfolgende ist, muss natürlich zuerst einmal erklärt werden, wie dieser Abfrageplan angezeigt werden kann.

Das „Explain“-Kommando liefert einen gängigen und einfachen Weg, den Abfrageplan anzusehen, doch mit der `auto_explain` Erweiterung lässt sich dies auch automatisieren. In beiden Fällen können mehrere Optionen angegeben werden, um festzulegen, welche Aspekte des Abfrageplans wie ausgegeben werden sollen.

Einen solchen Abfrageplan zu lesen, ist für unerfahrene Anwender nicht gerade leicht. Dieser Vortrag soll hier ansetzen, um vor allem Inhalt und Struktur von Abfrageplänen zu klären. Da Abfragepläne darstellen, wie die in einer PostgreSQL Datenbank gespeicherten Daten abgerufen werden, wird dabei auch ein wenig beleuchtet, wie diese Daten überhaupt gespeichert werden.

Vorher müssen jedoch die „Kosten“ erklärt sein – Kosteneinheiten im Bezug auf PostgreSQL Abfragen stellen ein vereinfachtes Modell zur Verfügung, mit dem verschiedene Abfragepläne bewertet werden können. Die Standardwerte bieten einen guten Start, können jedoch überschrieben werden – und das kann sinnvoll sein. Auch auf die automatisch erhobenen Statistiken soll eingegangen werden, denn diese können Abfragepläne maßgeblich beeinflussen.

Danach wird schließlich die Grundstruktur eines jeden Abfrageplans erklärt (Baum, Blätter, Knoten), sowie die wichtigsten darin angezeigten Werte. Anschließend werden anhand mehrerer Beispiele verschiedene Knotentypen erläutert: Woher die Daten erst einmal kommen (Sequential scan, Index (/only) scan, Bitmap index scan) und verschiedene Operationen auf diesen Datensätzen (Schleifen, Hash Join, Merge join, Sortierung, Limit, ...). Auf einige von ihnen wird zudem auch tiefer eingegangen – um ein paar interessante Details der PostgreSQL Interna zu aufzuzeigen, oder/und um Optimierungsmöglichkeiten aufzuzeigen.

Um dem Vortrag folgen zu können, ist SQL-Grundwissen sehr empfehlenswert (Select, From, Where, Join, Order by, Limit). Wissen spezifisch für PostgreSQL oder dessen Interna sind nicht erforderlich.