

Der SQL-Workflow

Es ist nicht immer einfach, eine SQL-Query zu schreiben. Je nach Fragestellung kann der Aufwand sehr groß sein und das Risiko, einen Fehler zu machen, steigt schnell an. Es existiert aber ein Workflow, den du immer anwenden kannst und der dir stets Stabilität bietet. Die Idee dahinter ist es, sich nach der Ausführungsreihenfolge der einzelnen Klauseln zu richten. Im Folgenden stellen wir dir diese Reihenfolge vor und geben einige Tipps, um Probleme zu lösen. Gerade am Anfang solltest du versuchen, diesen Workflow zu nutzen. Im Laufe der Zeit kannst du ihn auch personalisieren und deinen eigenen Stil finden.

FROM

Zu Beginn ermittelt MySQL die Gesamttabelle, in welcher dann das finale Resultat steckt. Beachte, dass auch die Joins hier ausgeführt werden. Mein Tipp an dich: Beginne deine Query mit **SELECT * FROM ...** und erzeuge zuerst die Tabelle, die du im Anschluss filterst, gruppierst und aus der du das finale Resultat ziehst. Das kann zwar unübersichtlich sein, aber im Allgemeinen merkst du so sehr schnell, ob du dich auf dem richtigen Weg befindest.

WHERE

Nachdem die Ergebnistabelle erstellt wurde, widmet sich MySQL dem Filtern. In dieser Reihenfolge solltest du auch agieren. Fokussiere die Tabelle auf die Elemente, die du für dein finales Resultat benötigst. Beachte, dass du hier nicht nach aggregierten Größen filtern oder Spaltenaliasse nutzen kannst, da weder eine Gruppierung noch das **SELECT**-Statement ausgeführt wurden.

GROUP BY

Nun werden die Daten gruppiert. Du solltest mittlerweile genauer wissen, welche spezifischen Spalten du benötigst und sie anstelle von **SELECT *** explizit auflisten. Auch wenn immer noch nicht das **SELECT** ausgeführt wurde, kannst du hier in MySQL bereits Spaltenaliasse nutzen. Aber Vorsicht, dieses Verhalten weicht vom SQL-Standard ab und funktioniert deshalb in anderen Datenbanksystemen möglicherweise nicht.

HAVING

Nun kannst du die Tabelle nach gruppierten Werten filtern, was mit der **WHERE**-Klausel noch nicht möglich war. Beachte, dass auch hier wieder die Nutzung der Spaltenaliasse MySQL-spezifisch und nicht standardkonform ist.

SELECT

Erst jetzt, nachdem alle Daten für die Ergebnistabelle zusammengetragen und gefiltert wurden, wird das **SELECT** ausgeführt. Dieser Schritt zieht dein finales Resultat aus den beteiligten Tabellen.

ORDER BY

Am Ende kannst du deine Ergebnisse nach Belieben sortieren. Da dies erst mit der finalen Tabelle Sinn ergibt, wird das **ORDER BY** beinahe zuletzt ausgeführt. Standardmäßig darfst du hier Spaltenaliasse nutzen.

LIMIT

Zu guter Letzt kannst du mit **LIMIT** deine Ergebnistabelle auf eine beliebige Anzahl Zeilen einschränken, wenn du nicht alle Daten benötigst.