Fachhochschule Nordwestschweiz
Hochschule für Technik

dbarc Lab 7:
Performanceoptimierung

Ziel dieses Labs ist es, verschiedene SQL-Abfragen durch geeignete Massnahmen zu optimieren. Für die Performanceoptimierung von einzelnen SQL-Statements gibt es verschiedene Möglichkeiten. Dazu gehören unter anderem:

- Umschreiben der SQL-Befehle, um die Abfragen zu vereinfachen. Komplexe SQL-Abfragen sind eine häufige Ursache für Performanceeinbussen.
- Erstellen von zusätzlichen Indizes. Das Ziel ist dabei: so wenig Indizes wie möglich, aber so viele wie nötig.
- Partitionierung von Tabellen. Wichtig dabei ist die Wahl eines geeigneten Partition Keys, damit Partition Pruning (Einschränkung auf eine oder mehrere Partitionen) möglich ist.
- Berechnung von zusätzlichen Statistiken (z.B. Histogramme, Extended Statistics). Dies kann dem Optimizer helfen, schlechte Schätzungen der Kardinalität zu verbessern.
- Verwendung von Hints, um dem Optimizer einen bestimmten Execution Plan vorzugeben. Achtung: Die Kosten des Execution Plans werden typischerweise höher, wenn Hints verwendet werden.

Die Aufgaben werden in Zweierteams gelöst und abgegeben. Jedes Team verwendet das eigene Schema, das in den vorherigen Labs erstellt und verwendet wurde:

Team	Schema
Team 1: Shana + Ramanan	DBARC1
Team 2: Arman + Jonathan	DBARC2
Team 3: Bianca + Danijel	DBARC3_USER
Team 4: Marc + Tobias	DBARC4_USER

Da die Aufgaben für die Leistungsbeurteilung relevant sind, müssen die Lösungen per E-Mail an dani.schnider@fhnw.ch abgegeben werden. Abgabeschluss ist **Freitag, 5. Mai 2023, 23:59 Uhr**.

1. Vorbereitung

Um eine vergleichbare Ausgangslage für alle Teams zu haben, solltet Ihr alle Indexes (ausser Primary Key und Unique Constraints), die in früheren Labs erstellt wurden, aus Eurem Schema löschen. Mit folgender Abfrage könnt Ihr die entsprechenden DROP-Befehle generieren:

```
SELECT 'DROP INDEX '||index_name||';'
FROM user_indexes
MINUS
SELECT 'DROP INDEX '||constraint_name||';'
FROM user_constraints WHERE constraint_type IN ('P', 'U');
```

Nach Ausführung der DROP-Befehle befindet sich das Schema wieder im ursprünglichen Zustand.

2. Aufgabenstellung

Das Script lab7_queries.sql enthält verschiedene Abfragen auf das Demo-Schema, die optimiert werden können. Einige davon sind zwar bereits schnell, können aber verbessert werden. Dies kann zweckmässig sein für Abfragen, die sehr oft ausgeführt werden. Andere Abfragen werden weniger häufig ausgeführt, sind aber langsam und sollen deshalb optimiert werden.

Da die Antwortzeiten der meisten Queries bereits schnell sind, gelten die Kosten der einzelnen Execution Plans als Kriterium für «gute Performance». Je tiefer die Kosten, desto besser ist der erreichte Execution Plan. In der Praxis ist dies nicht immer der Fall, aber für die Abfragen in dieser Übung ist eine Reduktion der Kosten in allen Fällen möglich.

Welche Performancemassnahmen Ihr verwendet, ist Euch überlassen.

3. Bewertungskriterien

Die umgesetzten Massnahmen (z.B. Erstellung von Indexes oder Anpassungen der SQL-Abfragen) sollen in einem SQL-Script dokumentiert werden. Zusätzliche Angaben (z.B. Erklärungen, weshalb Ihr eine bestimmte Massnahme gewählt habt), könnt Ihr direkt als Kommentare im Script ergänzen.

Als zusätzliche Dokumentation werden die Execution Plans der ursprünglichen und optimierten Abfragen festgehalten (z.B. mit dbms_xplan oder mit Screenshots der grafischen Execution Plans von SQL Developer, etc.). Dabei sollen die Kosten und wenn möglich auch die Ausführungszeiten vor und nach der Optimierung ersichtlich sein.

Je nach Komplexität der Aufgabe können unterschiedlich viele Punkte erreicht werden. Für jede Query ist im Kommentar ersichtlich, wie viele Punkte maximal erreicht werden können.