

# Stabilisierung von Ausführungsplänen - Baselines

Klaus Reimers  
ORDIX AG  
Paderborn

## Schlüsselworte

Optimizer, Ausführungsplan, Outline, Baseline, Profile, SPM

## Einleitung

Der Optimizer ist die komplexeste Komponente innerhalb der Oracle Datenbank. Der Ausführungsplan wird in der Parse-Phase erzeugt. In aller Regel wählt der Optimizer den für die aktuelle Situation korrekten Ausführungsplan. Allerdings kann es bei sehr komplexen Statements vorkommen, dass der Plan nicht optimal ist. Wenn alle Strukturanpassungen (z.B. neue Indizes) nicht helfen, dann muss der Ausführungsplan stabilisiert werden.

Oracle stellt zur Stabilisierung von Plänen einige Möglichkeiten zur Verfügung:

- Stored Outlines
- SQL Profiles
- SQL Plan Baselines

Im Vortrag werde ich auf diese Möglichkeiten eingehen, der Schwerpunkt liegt in der Betrachtung der Wirkungsweise der Baselines.

Der Vortrag wird vor allem als Demo gehalten, Folien stehen dabei im Hintergrund.

Abschließend werde ich eine Bewertung geben, welche Variante in welcher Situation sinnvoll sein kann.

## Stored Outlines

Stored Outlines gehören zu den Möglichkeiten, mit denen Ausführungspläne stabilisiert werden können. Dabei sind Stored Outlines im Grunde lediglich eine Ansammlung von Hints, die allerdings keine Anpassung des eigentlichen SQL-Statements benötigen.

Das liegt daran, dass Stored Outlines Objekte des Data Dictionaries sind und dort die Hints für ein spezifisches SQL-Statement gespeichert werden.

Auf Basis der SQL-Signatur – also dem Hashwert eines SQL-Statements – wird in den Tabellen des Users OUTLN nach einer vorhandenen Stored Outline gesucht. Existiert eine, so werden die gespeicherten Hints für das SQL-Statement übernommen und der entsprechende Ausführungsplan generiert.

Stored Outlines sind mit Oracle 12g desupported worden, funktionieren allerdings auch in der aktuellen Version 12.2 immer noch problemlos. Stored Outlines können in allen Editionen verwendet werden, also auch in der Standard Edition.

## SQL Profile

Ein SQL Profile besteht aus ausgelagerten Informationen im Data Dictionary bezüglich eines spezifischen SQL-Statements. Vereinfacht dargestellt ist ein SQL Profile für ein SQL-Statement vergleichbar mit Statistiken für eine Tabelle oder einen Index.

Anders als bei der Verwendung von Hints, muss für die Verwendung von SQL Profiles kein SQL-Statement umgeschrieben werden.

SQL Profiles helfen dem Optimizer bei der Wahl des optimalen Ausführungsplans, legen allerdings keinen konkreten Plan fest. Die SQL Profiles bestehen lediglich aus einer Ansammlung von Hints. Bei neuen Statistiken kann dies dazu führen, dass der tatsächliche Ausführungsplan nicht dem des SQL Profiles entspricht.

Neben eine Reihe von Statistiken beinhalten SQL Profiles vor allem Informationen über die Kardinalität der abgefragten Tabellen auf Basis der Where-Klausel. Die geschätzte Kardinalität des Optimizers auf die zugegriffen werden muss, wird mit den Informationen des SQL Profiles verglichen. Stimmen diese Informationen nicht miteinander überein, verwendet der Optimizer als Einschätzung die Informationen des Profiles.

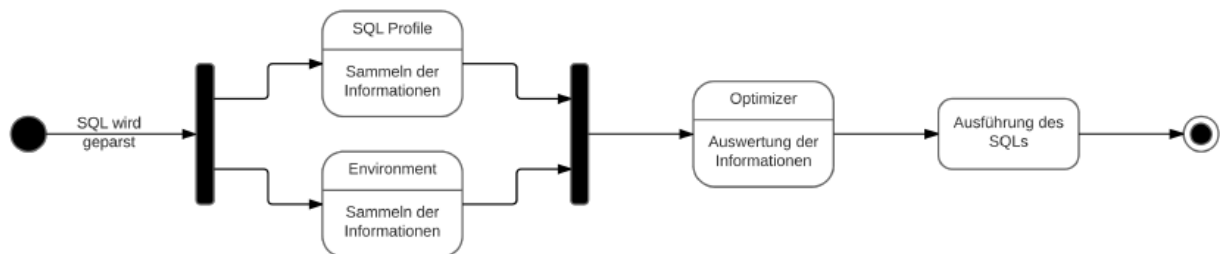


Abb. 1: Ablauf SQL Profile

Diese Abbildung verdeutlicht die Beziehung zwischen einem SQL-Statement und einem SQL Profile. Zunächst wird das Statement geparkt, woraufhin sowohl die Informationen aus dem SQL Profile, als auch die aktuellen Statistiken und Umgebungsparameter gesammelt werden.

Diese Informationen werden vom Optimizer ausgewertet, woraufhin der Ausführungsplan ausgewählt und das Statement ausgeführt wird.

Die offiziell von Oracle unterstützte Möglichkeit zur Erstellung von SQL Profiles ist der SQL Tuning Advisor. Der Advisor ist eine Diagnose Software des Oracle Database Tuning Packs und überprüft ein SQL-Statement auf eine mögliche Optimierung. Anschließend übergibt der Advisor das Statement mit den Optimierungsvorschlägen an den Query Optimizer. Dieser beinhaltet den sogenannten Automatic Tuning Optimizer, welcher die Möglichkeiten vergleicht und dem Tuning Advisor im Anschluss eine konkrete Optimierung vorschlägt.

Folgende Auflistung gibt einen Überblick über die Optimierung, welche durch die Verwendung des SQL Tuning Advisors automatisch durchgeführt werden können:

- Sammeln von Statistiken
- Anlegen von Indizes
- Umschreiben von SQL Statements
- Festschreiben eines Ausführungsplans als SQL Profile

Der Tuning Advisor ist nun dafür verantwortlich, die Optimierung durchzuführen. Dazu berechtigte Anwender können den Tuning Advisor über eine Funktion des Packages DBMS\_SQLTUNE aufrufen und so ein Tuning Task durchführen. Bei der Durchführung einer Tuning Task kann zwischen advise oder accept gewählt werden. Die Option accept würde die Optimierung direkt umsetzen – ein advise würde diese nur vorschlagen.

Wird ein SQL Profile als Optimierung vorgeschlagen, so legt der Advisor dies – bei der Durchführung des Tuning Task mit accept – im Data Dictionary an. Mit der Funktion CREATE\_TUNING\_TASK kann ein solcher Tuning Task auf Basis eines SQL Textes oder einer SQL\_ID gestartet werden.

### SQL Baseline

Mit dem Oracle Release 11g wurde SQL Plan Management (SPM) eingeführt. „Ziel von SPM ist es, stabile und optimale Performance für ausgewählte SQL-Statements zu erlangen.“

Das Konzept hinter SPM besteht aus einer SQL Plan History und SQL Plan Baselines, welche im Tablespace SYSAUX gespeichert werden. Baselines verwenden – anders als SQL Profiles und Stored Outlines – keine Hints.

Damit der Optimizer die Baselines verwenden kann, muss der Datenbankadministrator die Verwendung zunächst aktivieren. Dazu wird der Parameter OPTIMIZER\_USE\_SQL\_PLAN\_BASELINES auf TRUE gesetzt.

SPM besteht aus drei wesentlichen Bestandteilen:

- SQL Plan Baseline Capture
- SQL Plan Baseline Selection
- SQL Plan Baseline Evaluation

Wird ein SQL-Statement ausgeführt, so wird der Ausführungsplan zunächst als akzeptierte (accepted) Baseline in der History des SPM gespeichert. Diese akzeptierten Baselines werden bei erneuter Ausführung des SQL-Statements als möglicher Ausführungsplan hinzugezogen. Der Optimizer wählt aus der Menge der akzeptierten Baselines den kostengünstigsten aus.

Neue Ausführungspläne werden – sofern mindestens eine akzeptierte Baseline vorhanden ist – als unaccepted in der History gespeichert. Bei diesen Baselines wird zwischen unverified und rejected unterschieden.

Der dritte wichtige Bestandteil besteht aus der Evaluierung von Baselines, welche als unaccepted gekennzeichnet sind.

Bei der Evaluierung einer Baseline wird der Ausführungsplan mit dem aktuellen Ausführungsplan verglichen. Wird der Ausführungsplan der Baseline als kostengünstiger betrachtet, so kann die Baseline durch die Evaluierung automatisch akzeptiert und im Anschluss direkt verwendet werden.

Das Besondere an SQL Plan Baselines ist, dass es mehrere „gute Ausführungspläne“ für ein SQL-Statement geben kann und der Optimizer sich – je nach Situation – für eine der akzeptierten Baselines entscheiden kann.

### **Hierarchietest**

Im praktischen Teil werden die einzelnen Methoden vorgestellt, dabei besonderes Gewicht auf die Verwendung von Baselines gelegt. Dabei werden die verschiedenen Möglichkeiten (accept / evaluate / fix) gezeigt und erläutert. Letztendlich werde ich zeigen, welches Verfahren Oracle priorisiert, wenn nicht nur eine Variante genutzt wird.

### **Zusammenfassung**

Die mögliche Verwendung der unterschiedlichen Varianten ist von der genutzten Edition und zusätzlichen Packs abhängig. Sollen alternative Pläne gespeichert werden, so sind die Baselines eine gute Wahl. Soll Cursor Sharing auf SQL Ebene genutzt werden, so müssen SQL Profiles verwendet werden. In jeder Edition ist die Nutzung von Outlines möglich, auch wenn diese seit Oracle 10g nicht mehr supported sind.

### **Kontaktadresse:**

Klaus Reimers  
ORDIX AG  
Karl-Schurz-Straße 19a  
D-33100 Paderborn

Telefon: +49 (0) 5251 1063-0  
Fax: +49 (0) 180 1673490  
E-Mail: [info@ordix.de](mailto:info@ordix.de)  
Internet: [www.ordix.de](http://www.ordix.de)