Indexbasiertes SQL Tuning

Eine Einführung

Sebastian Wittig
Systementwickler
merlin.zwo InfoDesign GmbH & Co. KG
76228 Karlsruhe

Spitzenleistung heißt, sich auf seine Stärken zu konzentrieren.

merlin 2wo InfoDesign

merlin.zwo

Wir machen Oracle - nur Oracle.

Aus gutem Grund.

www.merlin-zwo.de

ORACLE Platinum Partner

Wir kümmern uns!

Übersicht

- Motivation
- Was ist eigentlich ein Index?
- Wann verwende ich welchen Index?
- Ausführungspläne
- Do‘s and Don‘ts

merlin2wo InfoDesign

Motivation

Motivation

```
SELECT count(mitarbeiter_nr)
  INTO l_count
  FROM (SELECT mitarbeiter_nr
        FROM   mitarbeiter_funktion
        WHERE  mitarbeiter_nr = nutzer_auth_pgk.get_user_id(iUSERNAME)
        AND    funktion = 20
        AND    gueltig_von > SYSDATE
        AND    ( gueltig_bis  <=  SYSDATE
                 OR gueltig_bis IS NULL )
        UNION ALL
        SELECT mitarbeiter_nr
        FROM   projektteam pt
        WHERE  pt.mitarbeiter_nr       =     nutzer_auth_pgk.get_user_id(iUSERNAME)
        AND    pt.funktion    =     20
        AND    NVL(pt.gueltig_von, SYSDATE) > SYSDATE
        AND    NVL(pt.gueltig_bis ,SYSDATE)<= SYSDATE
        );
```

Motivation

Ursächlich für solche Statements sind bspw.:

- Unkenntnis über Datenbankmechanismen
- Zeitdruck bei der Umsetzung von Funktionen
- Rückgriff auf objektrelationale Abbildung
- Toolgeneriertes SQL

In der Folge: Es herrscht mitunter unreflektiertes Vorgehen beim Formulieren von SQL.

Was ist eigentlich ein Index?

Was ist ein Index?

- Datenstruktur
- Per *CREATE INDEX* Befehl erzeugt
  - Wann darf man das?
  - Welche Informationen fließen hinein?

Beispiel:

```
CREATE INDEX mitarbeiter_idx
    ON mitarbeiter (nachname, vorname, rufnummer);
```

Was ist ein Index?

merlin2 InfoDesign

- Enthält vorhandene Informationen erneut
- Enthält *nur* diese Informationen
- Diese Informationen sind sortiert
- Belegt zusätzlichen Speicherplatz
- Wird stets aktuell gehalten

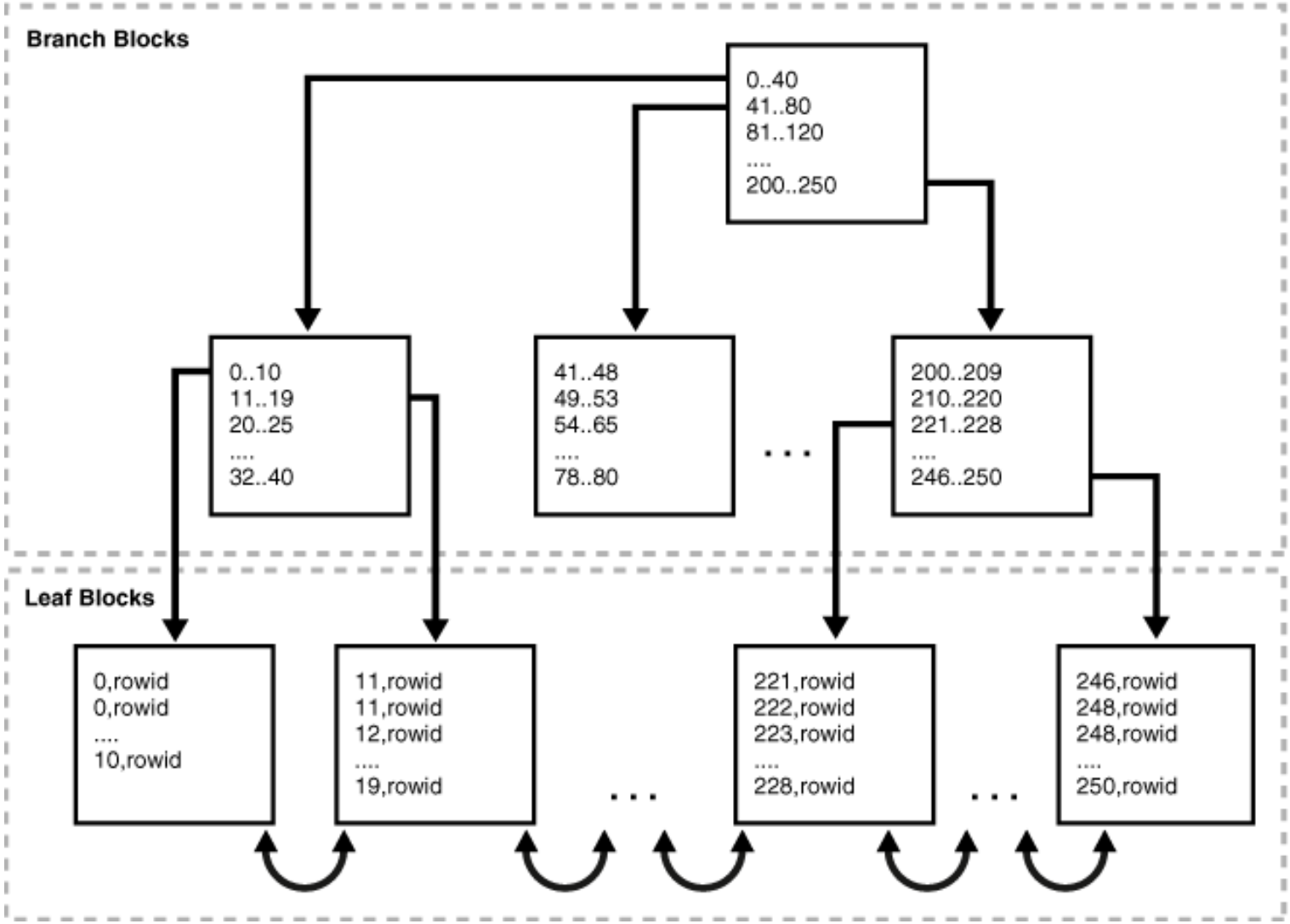
Ein Index ist eine *zweckgebundene redundante Datenstruktur*.

Was ist ein Index?

Es gibt verschiedene Formen die diese Indizierung annehmen kann, dazu gehören u.a. folgende:

- B*Tree Index (Standard)
- Bitmap Index (nur EE)
- Funktionsbasierter Index
- Index Organized Table

Was ist ein Index?



Welche Spalten indizieren?

merlin2 InfoDesign

Am besten wählt man Spalten mit hoher Kardinalität, da diese meist mit starker Selektivität einhergehen.

- Kardinalität ist die Anzahl der Werteausprägungen einer Spalte
- Häufig hat man sich beim Datenmodell schon Überlegungen über die Kardinalität getroffen
  - Flags haben in der Regel eine niedrige Kardinalität
  - Primärschlüssel haben eine besonders hohe Kardinalität

Welche Spalten indizieren?

merlin2 InfoDesign

- Selektivität ist wie folgt definiert: $sel_P = \frac{|\sigma_P(A)|}{|A|}$
- Bevorzugt Spalten mit starker Selektivität wählen
- Spalten mit schwacher Selektivität in mehrspaltige Indizes mitaufnehmen

Selektivität ist nicht nur für die Entscheidung ob indiziert werden soll wichtig, sie hat auch Einfluss auf den Optimizer.

merlin2 InfoDesign

Wann verwende ich einen Index?

B*Tree Index

- Optimal bei vergleichsweise geringer Zeilenzahl in der Rückgabe
- Als Faustregel:
  - <5% ist der Indexzugriff in der Regel schneller
  - 5%<x<15% ist zu prüfen

Mehr Zeilen sorgen dafür, dass der Index schnell langsam wird.

Da für die Ergebnismenge erst ROWIDs gesammelt werden müssen und anschließend ein Zeilenzugriff stattfindet.

B*Tree Index

- sehr gut geeignet für treibende Tabellen in JOINS
  - generell ist es ratsam bei Tabellen früh zu filtern, da späteres filtern die Zeilen die verworfen werden deutlich erhöht
  - gute Indizierung spart bei „Intersection“ Tabellen den Objektzugriff

- Kostenüberlegung nicht überinterpretieren
  - der erste Index ist performancetechnisch der teuerste
  - *sollte* mit dem Primärschlüssel ohnehin erfolgen
  - weitere Indizes verschlechtern die Performance, aber nur relativ wenig

Bitmap Index

- nur Enterprise Edition
- kodiert Spaltenausprägungen in einer Bitmap
- sehr gut wenn:
  - viele Sätze aus großen Tabellen selektiert werden
  - die Prädikate eine niedrige Selektivität aufweisen
  - sehr komplexe Abfragen vorliege
- niedriger Speicheraufwand
- wird häufig im DWH-Umfeld genutzt

Funktionsbasierter Index

- Basiert auf einem Ausdruck wie lower(<String>)
- Ermöglicht eine Vorausberechnung aufwändiger Funktionen
- Funktion muss deterministisch sein
- Suchoptimierung und Großkleinschreibung

- Kein RBO – Support
- Keine Aggregatsfunktionen
- Manueller Rebuild kann nötig sein

Index-Organized-Table

Existiert ein Index der alle Spalten enthält, wird die Tabelle entsorgt.

- kein zusätzlicher Objektzugriff mehr nötig
- Daten liegen vorsortiert vor
- weniger Speicherplatz nötig, bessere Kompression

Aber:

- teure DML Operationen
- zusätzliche weitere Indizes sind komplexer

merlin2wo InfoDesign

Ausführungspläne

Ausführungspläne

- können mit EXPLAIN Plan und den meisten GUIs erzeugt werden
- zeigt den Ablaufplan der Operationen an
- kann zum Verifizieren der eigenen Überlegungen genutzt werden
- für sich aber eine schwierige Entscheidungsbasis
  - Kosten sind ein abstraktes Maß
  - zahlreiche Einflüsse, wie z.B. Optimizerparameter
  - ggfs. mit tkprof oder Trace abgleichen

Ausführungspläne

- Z.B. via `@$ORACLE_HOME/RDBMS/ADMIN/UTLXPLAN.SQL` erstellen
- Aufruf:

```
EXPLAIN PLAN FOR SELECT last_name FROM employees;
```

- Output:

```
Rows   Plan
-----  -------------------------
16957  SELECT STATEMENT
16957  TABLE ACCESS FULL EMPLOYEES
```

Ausführungspläne

- Prüfen ob die Prädikate entsprechend genutzt werden
  - Accessprädikate bestimmen die ausgelesenen Zeilen auf dem Index
  - Filterprädikate werden nach dem Zeilenzugriff genutzt
- JOIN-Algorithmen prüfen
- Werden meine Indizes korrekt genutzt?
  - Ausführungspläne sind instabil und können kippen
  - Insbesondere bei Veränderungen der Statistiken
  - Annahmen müssen ggfs. reevaluiert werden

merlin2wo InfoDesign

Do‘s and Don‘ts

Do‘s

- Spalten indizieren die häufig in WHERE Klauseln und JOINS auftauchen
- Fremdschlüsselspalten indizieren
- Zugriffspfade abstimmen und nutzen
- Funktionsaufrufe in der WHERE Klausel vorsichtig nutzen
- Vor dem Schreiben eines Statements:
  - Gedanken über den optimalen Zugriffspfad machen
  - Theorie verifizieren

Don‘ts

- Indizes zum Selbstzweck erstellen, stets überlegen ob nötig
- niemals prüfen ob Indizes noch genutzt werden
- Geschwindigkeitsvorteile in der Produktion erwarten und untätig bleiben
- Full Table Scans vollständig eliminieren
- unüberlegt HINTs einsetzen

Fragen und Antworten






Haben Sie noch Fragen?

merlin.zwo InfoDesign GmbH & Co. KG
Sebastian Wittig

merlin.zwo
Wir kümmern uns!

merlin.zwo InfoDesign GmbH & Co. KG

Sebastian Wittig
Elsa-Brändström-Straße 14
76228 Karlsruhe
Tel. 0721 132096-44

sebastian.wittig@merlin-zwo.de
http://www.merlin-zwo.de