

4. Aufgabenblatt

Indexierung mit Oracle

Abgabe: 17.01.2018, 23:59 Uhr

Patrick Schäfer

Berlin, 18. Dezember 2017

patrick.schaefer@hu-berlin.de

Vorlesung:

https://hu.berlin/vl_dwhdm17

Übung:

https://hu.berlin/ue_dwhdm17

Daten (Wettbewerbsaufgabe!)

- Gegeben sei eine Datenbank aller Einwohner eines Landes
- Felder der Tabelle EINWOHNER:
 - ID, Name, Vorname, Geburtsdatum, Wohnort, Adresse, PLZ, Bundesland, VaterID, MutterID
- Instanziierung ist vorher **unbekannt!**
 - Ca. 50.000.000 Tupel
 - Die älteste Person wurde 1914 geboren
 - Kinder bekommt man frühestens mit 18
 - VaterID bzw. MutterID ist NULL, wenn das Elternteil nicht im Land wohnt oder gestorben ist!
- 1.000 und 10.000 Beispieldaten (nicht die Wettbewerbsdaten) und CREATE-Skript auf der Übungsseite

Beispieldaten (Auszug)

ID;NAME;VORNAME;GEBURTSDATUM;ADRESSE;WOHNORT;PLZ;BUNDESLAND;VATERID;MUTTERID
0;Heuer;Tilman;1920-10-17;Voßweg 70;Kaub;56349;Rheinland-Pfalz;NULL;NULL
1;Weinhold;Galina;1921-7-2;Pfarrer-Schneider-Straße 100;Remscheid;42853;Nordrhein-Westfalen;NULL;NULL
2;Radermacher;Dietwalt;1921-11-2;Fiedlerstraße 90;Marktleuthen;95168;Bayern;NULL;NULL3;Dittmann;Hasso;1916-12-17;Guntherstr. 58;Schönberg;22929;Mecklenburg-Vorpommern;NULL;NULL
4;Kögler;Mary;1918-1-24;Brönnnerstraße 87;Stuttgart;70190;Baden-Württemberg;NULL;NULL
5;Kracht;Steffen;1918-10-26;Nelkenstraße 88;Grevenbroich;41517;Nordrhein-Westfalen;NULL;NULL
6;Buß;Melita;1921-1-2;Stendaler Straße 60;Manderscheid;54531;Rheinland-Pfalz;NULL;NULL
7;Scheer;Felizitas;1920-3-18;Launhardtstraße 33;Plauen;08523;Sachsen;NULL;NULL
8;Graßl;Meta;1920-1-28;Hellerstr. 58;Wirges;56422;Rheinland-Pfalz;NULL;NULL
9;Burkhart;Auguste;1915-5-24;Am Hirtenacker 100;Ichenhausen;89335;Bayern;NULL;NULL
10;Bluhm;Heidegret;1918-8-29;Kneiselmühle 89;Berlin;12557;Berlin;NULL;NULL
11;Bohm;Alwine;1914-5-1;Laufkötterweg 98b;Peitz;03185;Brandenburg;NULL;NULL
12;Siepmann;Lucia;1914-6-16;Schlinkenweg 47;Hückeswagen;42499;Nordrhein-Westfalen;NULL;NULL
13;Strasser;Neithard;1918-1-17;Moränenende 41;Kelbra;06537;Sachsen-Anhalt;NULL;NULL
14;Heide;Adolph;1915-10-12;Richard-Bernhardt-Weg 34;Oer-Erkenschwick;45739;Nordrhein-Westfalen;NULL;NULL
15;Leder;Katherina;1917-7-1;Langenzugbrücke 33;München;80689;Bayern;NULL;NULL
16;Keitel;Edelfried;1917-6-29;Ooser Gartenstraße 1;Soltau;29614;Niedersachsen;NULL;NULL
[...]

CREATE Table-Script

```
CREATE TABLE EINWOHNER (  
    ID            INTEGER        NOT NULL,  
    NAME          VARCHAR( 25 )  NOT NULL,  
    VORNAME       VARCHAR( 25 )  NOT NULL,  
    GEBURTSDATUM DATE           NOT NULL,  
    ADRESSE       VARCHAR( 60 )  NOT NULL,  
    WOHNORT       VARCHAR( 50 )  NOT NULL,  
    PLZ           VARCHAR( 5 )   NOT NULL,  
    BUNDESLAND   VARCHAR( 25 )  NOT NULL,  
    VATERID      INTEGER,  
    MUTTERID     INTEGER  
);
```

SQL-Anfrage 1

- Die Anzahl der Personen, die in Berlin wohnen und (mindestens) ein Geschwisterteil (auch Halbbruder und Halbschwester) haben, das außerhalb von Berlin lebt
 - Jede Person soll dabei nur einmalig gezählt werden (auch bei mehreren Geschwistern)

SQL-Anfrage 2

- Die Anzahl der Personen, die in einem kalten Monat (Dezember, Januar, Februar) geboren sind und (laut Datenbank) keine Eltern (Vollwaise) haben

SQL-Anfrage 3

- Anzahl der Personen, die in einem Haus mit der Hausnummer 23 wohnen
=> Freitextsuche in Adresse

SQL-Anfrage 4

- Die Knoten (=Personen) des längsten Pfades im Datenbaum
- Struktur des Ergebnisses
 1. ID der ältesten Person
 2. ID von Tochter bzw. Sohn
 3. ...
 - n. ID der jüngsten Person, aka $Ur^{(n-3)}$ – Enkel
- Beispiel:
 - Für 1000: /74/251/494/771/1000
 - Für 10000: /1734/3891/5375/7115/9456

Indexierung / Execution Plan

- (Sinnvolle) Index(e) erstellen
- „**Explain Plan**“ für jede Anfrage ausführen:

```
EXPLAIN PLAN for  
  SELECT ... FROM ...;
```



```
SELECT PLAN_TABLE_OUTPUT  
  FROM TABLE(DBMS_XPLAN.DISPLAY());
```
- Idealerweise sollte kein einziger TABLE ACCESS FULL angezeigt werden (nicht immer möglich)

Wettbewerb

- Es wird die Gesamtzeit aller Queries plus Indexerzeugung betrachtet
- Punkte
 1. Gruppe: 5 Punkte
 2. Gruppe: 3 Punkte
 3. Gruppe: 2 Punkte

Wettbewerb: Zeitmessung

- Skript erstellen, welches ...
 - eure Datenstrukturen vorbereitet,
 - alle SQL-Anfragen 1 Mal ausführt und
 - die Gesamtzeit misst
- Vorlage siehe Übungsseite
 - Hier die SQL-Anweisungen eintragen (nicht die VIEWS)

Anmerkungen

- Ihr dürft im Skript alles Verwenden (im Rahmen eurer Oracle-Berechtigungen)
 - Indizes anlegen, PL/SQL/ Rekursion, Partitionierung, ...
 - Wichtig: die Zeit zählt ab Anfang eures Skripts
- Eure Anfragen müssen bezüglich der Tabelle **EINWOHNER** definiert sein
- Zum finalen Test:
 - a) Lege ich die Tabelle EINWOHNER an
 - b) Lade die Test-Daten in die Tabelle EINWOHNER
 - c) Führ euer Skript aus
 - Ich lege also keine Indizes an – das müsst ihr selber im Skript machen

Abgabe

- Deadline ist **Mittwoch, 17.01.2018, 23:59 Uhr**
- Upload einer Zip-Datei „Gruppenname.zip“ per HU-Box:
https://hu.berlin/dwhdm_ue4
- Die Tabelle/Daten/Indixe müssen in eurer **Oracle DB** liegen
- Für jede Anfrage muss ein **View** in eurer **Oracle DB** existieren
- Die Abgabe (**ZIP-Datei**) im Einzelnen:
 - Text-Datei mit
 - **CREATE INDEX** Anweisungen
 - Für jede Anfrage: **SQL-Befehl, Ergebnis und Execution Plan**
 - **Namen der View** für jede Anfrage in eurer DB
 - **Wettbewerb: Das Skript zum Messen der Laufzeit mit (a) den CREATE INDEX Anweisungen und (b) SQL-Queries**

Präsentation

- Ihr dürft euch aussuchen, was und wann ihr präsentieren wollt (first-come-first-served).
- Montags (Gruppe 1)
https://dudle.inf.tu-dresden.de/dwhdm_mo_ue4/
- Mittwochs (Gruppe 2)
https://dudle.inf.tu-dresden.de/dwhdm_mi_ue4/
- Kurzpräsentation der Lösung am 22.01. (Gruppe 1) bzw. 24.01. (Gruppe 2).

Fragen?