CARAJAN DB

Index oder nicht Index

Johannes Ahrends
Geschäftsführer
CarajanDB GmbH

ORACLE ACE

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Agenda

- Vorstellung CarajanDB
- I
- Index oder nicht Index – das ist doch keine Frage, oder?
- Was kann der DBA tun?
- Was kann der Entwickler tun?

CarajanDB

- **Experten mit über 30 Jahren Oracle Erfahrung**
- **Spezialisten für**
  - Backup & Recovery
  - Hochverfügbarkeit
  - Healthchecks
  - Performance Optimierung
  - Einsatz von Oracle Standard Edition
  - Oracle in virtuellen Umgebungen und in der Cloud
  - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
  - Monitoring (Grid / Cloud Control, HLMM, Foglight, Spotlight)
- **Schulung und Workshops (Oracle, Toad)**

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Einfache Indizierung

- **Sind diese Indizes sinnvoll?**

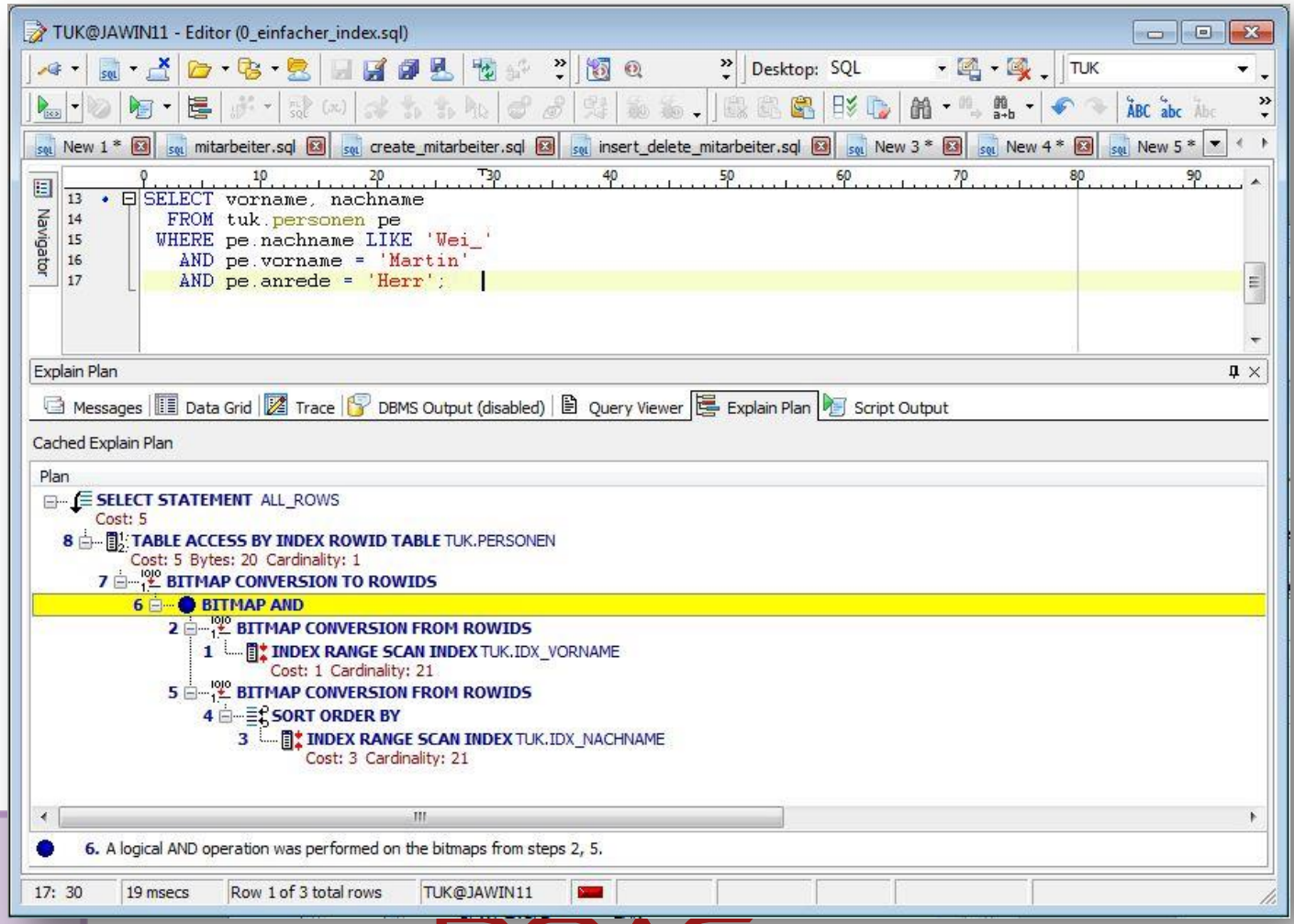
```
CREATE INDEX idx_nachname
   ON personen (nachname);
CREATE INDEX idx_vorname
   ON personen (vorname);
```

- **Abfrage 1:**

```
SELECT anrede, vorname, nachname
  FROM tuk.personen pe
 WHERE pe.nachname LIKE 'Wei_'
   AND pe.vorname = 'Martin'
   AND pe.anrede = 'Herr';
```

0_einfacher_index.bat

Bitmap Conversion



Zusammengesetzter Index

- **Ist dieser Index sinnvoll?**

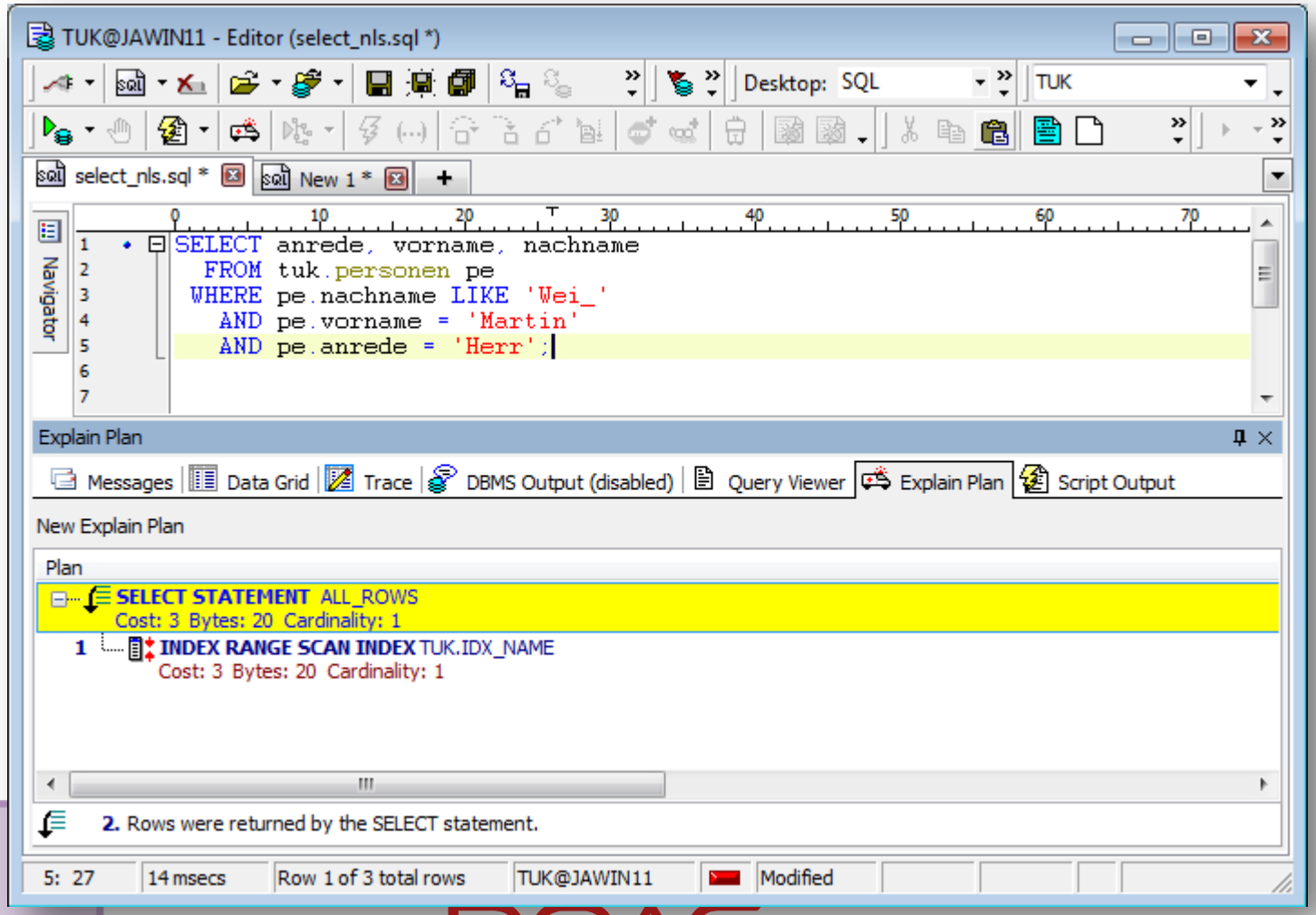
```
CREATE INDEX idx_name
   ON personen (anrede, vorname, nachname);
```

- **Abfrage 1:**

```
SELECT anrede, vorname, nachname
  FROM tuk.personen pe
 WHERE pe.nachname LIKE 'Wei_'
   AND pe.vorname = 'Martin'
   AND pe.anrede = 'Herr';
```

1_concat_index.bat

Index Range Scan



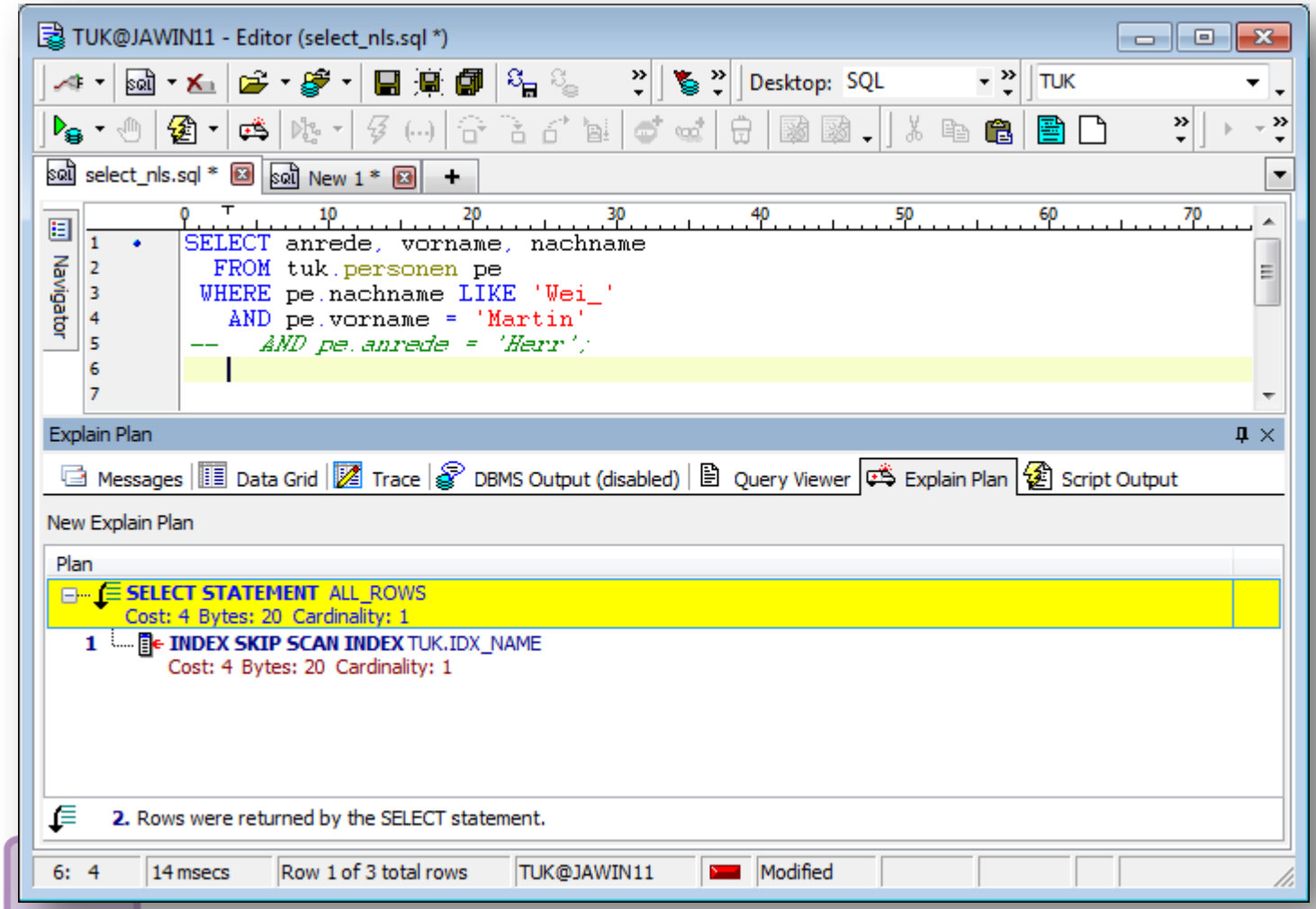
Index Aufbau

- **Abfrage 2:**

```
SELECT anrede, vorname, nachname
  FROM tuk.personen pe
 WHERE pe.nachname LIKE 'Wei_'
   AND pe.vorname = 'Martin'
--   AND pe.anrede = 'Herr';
```

1b_concat_index.bat

Index Skip Scan



Index Benutzung oder nicht?

```
SQL> SELECT anrede, vorname, nachname
  2    FROM tuk.personen pe
  3   WHERE pe.nachname LIKE 'wei_'
  4     AND pe.vorname LIKE 'Martin';

ANRED VORNAME                NACHNAME
----- ---------------------- --------------------
Herr  Martin                 Weiß
Herr  Martin                 Weis
Herr  Martin                 Weiz

3 Zeilen ausgewählt.
```

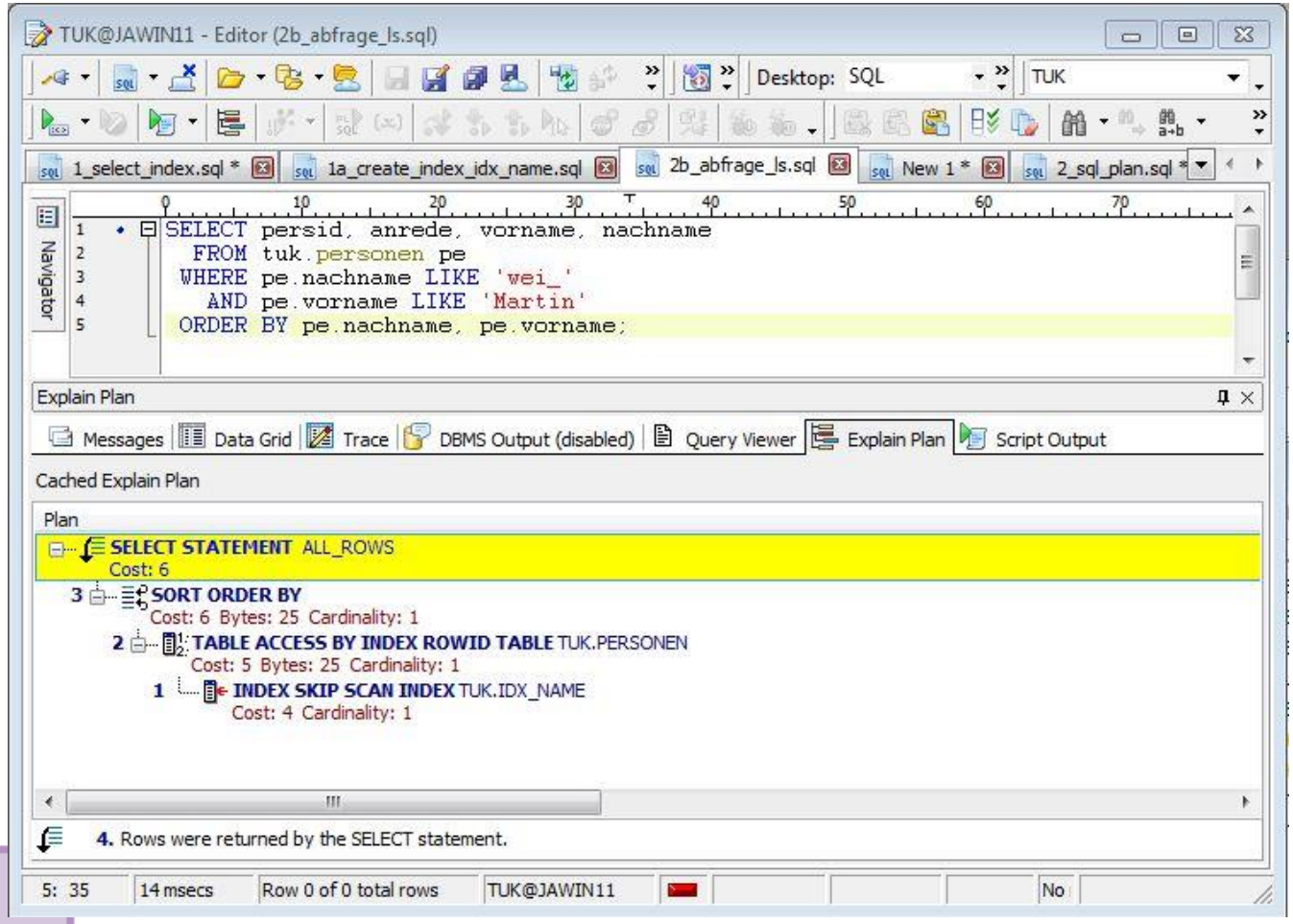
2_abfrage_ls.bat

Warum kein Index?

```
PLAN_TABLE_OUTPUT
-----------------------------------------
Plan hash value: 1826680655

-----------------------------------------
| Id  | Operation           | Name      |
-----------------------------------------
|   0 | SELECT STATEMENT    |           |
|   1 |  SORT ORDER BY      |           |
|   2 |   TABLE ACCESS FULL| PERSONEN  |
-----------------------------------------
```

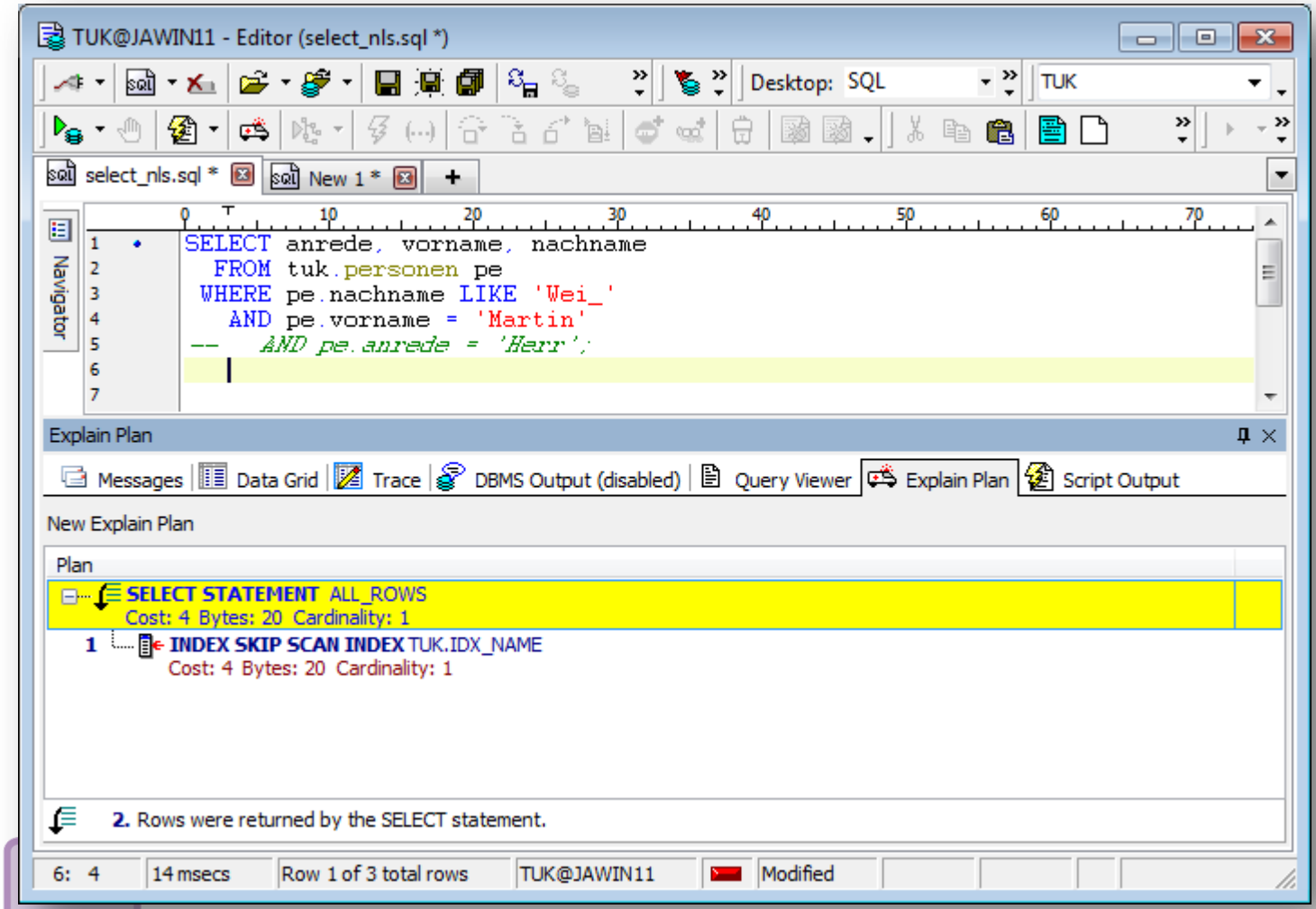
… oder doch ein Index?



Deutsche ORACLE-Anwendergruppe e.V.

… den hatten wir doch schon einmal!

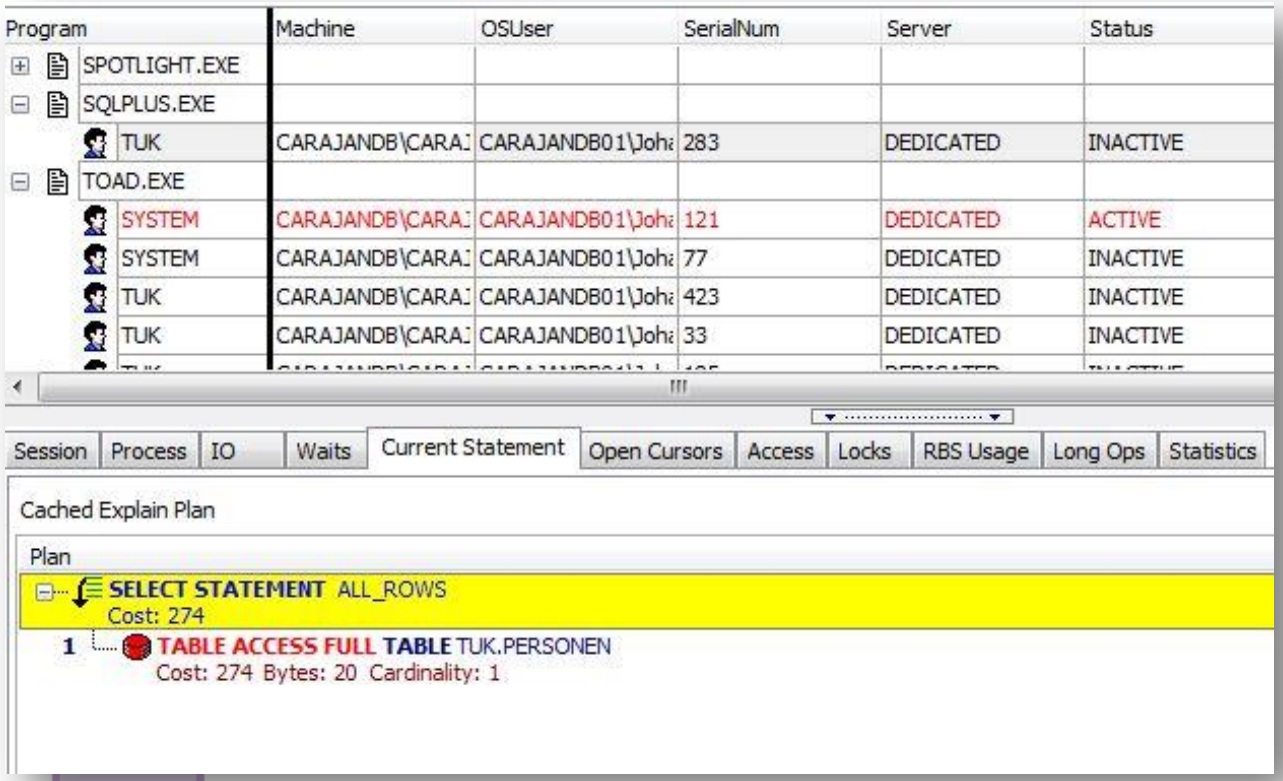
CARAJANDB



DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Ausführungsplan

- Warum Full-Table-Scan?



Warum?

- **Versuch einer Analyse:**

```
SQL> SELECT anrede, vorname, nachname
  2    FROM tuk.personen pe
  3   WHERE pe.nachname LIKE 'wei_'
  4     AND pe.vorname LIKE 'Martin';

ANRED VORNAME              NACHNAME
----- -------------------- --------------------
Herr  Martin               Weiß
Herr  Martin               Weis
Herr  Martin               Weiz

3 Zeilen ausgewählt.
```

Warum?

- Versuch einer Analyse:

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
Plan hash value: 1826680655

-------------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |     1 |    25 |   275   (2)| 00:00:04 |
|   1 |  SORT ORDER BY     |          |     1 |    25 |   275   (2)| 00:00:04 |
|*  2 |   TABLE ACCESS FULL| PERSONEN |     1 |    25 |   274   (1)| 00:00:04 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("PE"."NACHNAME" LIKE 'wei_' AND
NLSSORT("PE"."VORNAME",'nls_sort=''BINARY_CI''') =HEXTORAW('6D617274696E00') )
```

Linguistische Suche

- **`ALTER SESSION SET nls_sort=binary_ci;`**
  - Sortierung ist unabhängig von Groß- / Kleinschreibung aber abhängig von Akzenten
  - Alternativen:
    - binary_ai → Case und Akzent insensitiv
    - german_ai → Deutsche Sortierung, Case und Akzent insensitiv
    - …
- **`ALTER SESSION SET nls_comp=linguistic;`**
  - Filter verwenden die gleiche Funktion wie `NLS_SORT`, d.h. in diesem Fall ist die WHERE-Clausel unabhängig von Groß- / Kleinschreibung und von Akzenten
  - Alternativen:
    - BINARY oder ANSI

Lösung

- Entweder Linguistische Suche ausschalten
- … oder Index auf Linguistische Suche

```
CREATE INDEX idx_name2
   ON personen (NLSSORT (vorname, 'nls_sort=binary_ci'),
                NLSSORT (nachname, 'nls_sort=binary_ci'));
```

- Aber jetzt sind z.B. keine Bitmapped Indizes auf den Spalten möglich

2c_nls_index.sql

Index oder nicht?

```
SELECT anrede, vorname, nachname, geburtstag
  FROM tuk.personen pe
 WHERE pe.nachname like 'Weisse%'
   and pe.anrede = 'Herr';
```

4_abfrage_mb.bat

Index Zugriff

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
Plan hash value: 1706350694

------------------------------------------------------------------------------------------
| Id  | Operation                    | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |          |    26 |   728 |   405   (1)| 00:00:05 |
|   1 |  TABLE ACCESS BY INDEX ROWID| PERSONEN |    26 |   728 |   405   (1)| 00:00:05 |
|*  2 |   INDEX RANGE SCAN           | IDX_NAME |    26 |       |   379   (1)| 00:00:05 |
------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("PE"."ANREDE"='Herr' AND "PE"."NACHNAME" LIKE 'Weisse%')
       filter("PE"."NACHNAME" LIKE 'Weisse%')
```

… oder doch nicht?

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------
Plan hash value: 2673218719

------------------------------------------------------------------------------
| Id  | Operation          | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |          |    26 |   728 |   274   (1)| 00:00:04 |
|*  1 |  TABLE ACCESS FULL| PERSONEN |    26 |   728 |   274   (1)| 00:00:04 |
------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("PE"."ANREDE"='Herr' AND "PE"."NACHNAME" LIKE 'Weisse%')
```

Serverparameter

- **db_file_multiblock_read_count**
  - Anzahl Blöcke die mit einem I/O beim table oder index scan gelesen werden
  - Default 128 (ev. 64)
  - Je höher der Wert umso wahrscheinlicher ein Full-Table-Scan
- **optimizer_index_cost_adj**
  - Verhältnis zwischen Full-Table-Scan und Index Benutzung
  - Default 100
  - Je geringer der Wert umso wahrscheinlicher ein Index Zugriff
  - Schlecht bei hohen Clustering Faktoren!

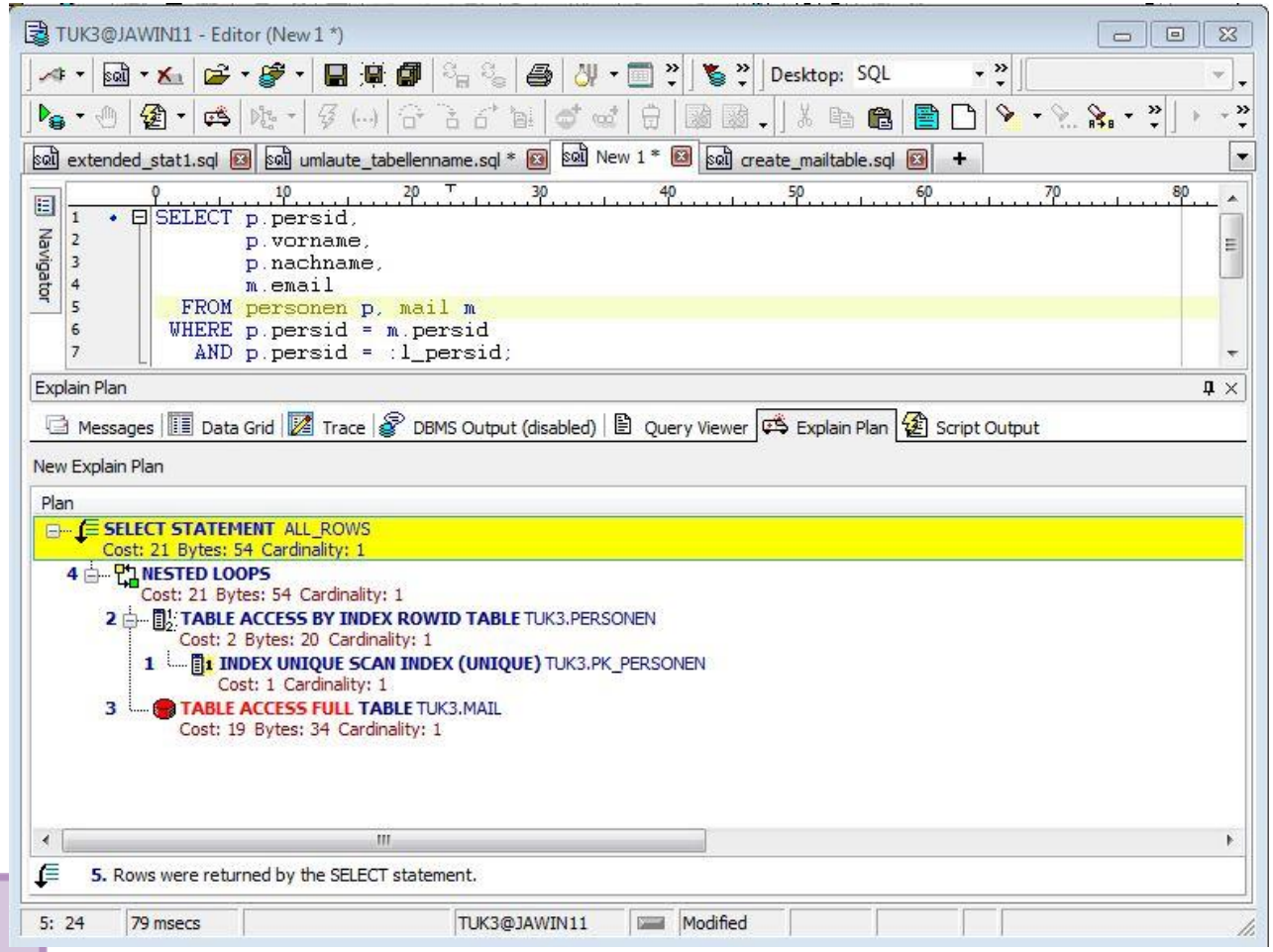
Index oder nicht?

```
SELECT p.persid,
       p.vorname,
       p.nachname,
       m.email
  FROM personen p, mail m
 WHERE p.persid = m.persid
   AND p.persid = :l_persid;
```

- **Beide Tabellen persid als Primary Key!**

5_select_mail.bat

Warum kein Index Zugriff?



Warum kein Index Zugriff?

CARAJANDB

```
-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name         | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |              |     1 |    54 |    21   (0)| 00:00:01 |
|   1 |  NESTED LOOPS                |              |     1 |    54 |    21   (0)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID| PERSONEN     |     1 |    20 |     2   (0)| 00:00:01 |
|*  3 |    INDEX UNIQUE SCAN         | PK_PERSONEN  |     1 |       |     1   (0)| 00:00:01 |
|*  4 |   TABLE ACCESS FULL          | MAIL         |     1 |    34 |    19   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("P"."PERSID"=TO_NUMBER(:L_PERSID))
   4 - filter(TO_NUMBER("M"."PERSID")=TO_NUMBER(:L_PERSID))
```

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Datentyp Konvertierung

```
CREATE TABLE MAIL
(
  PERSID      VARCHAR2(10 BYTE),
  EMAIL       VARCHAR2(50 BYTE),
  BEMERKUNG   VARCHAR2(200 BYTE)
)
TABLESPACE USERS;
```

Statistiken

```
CREATE INDEX idx_ort
    ON adressen (plz,ort);
```

```
CREATE INDEX idx_name
    ON personen(nachname,vorname);
```

```
SELECT pe.anrede, pe.vorname, pe.nachname,
       ad.plz, ad.ort, ad.strasse
  FROM adressen ad, personen pe
 WHERE pe.persid = ad.persid
   AND pe.nachname = :nachname
   AND ad.plz = :plz
   AND ad.ort = :ort;
```

7_statistiken.bat

Ausführungsplan 1

```
Ausführungsplan
----------------------------------------------------------
Plan hash value: 2938594640

-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name         | Rows  | Bytes | Cost (%CPU)| Time
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |              |     2 |   140 |     5   (0)| 00:00:
|   1 |  NESTED LOOPS                 |              |       |       |            |
|   2 |   NESTED LOOPS                |              |     2 |   140 |     5   (0)| 00:00:
|   3 |    TABLE ACCESS BY INDEX ROWID| ADRESSEN     |     2 |    76 |     3   (0)| 00:00:
|*  4 |     INDEX RANGE SCAN          | IDX_ORT      |     2 |       |     1   (0)| 00:00:
|*  5 |    INDEX UNIQUE SCAN          | PK_PERSONEN  |     1 |       |     0   (0)| 00:00:
|*  6 |   TABLE ACCESS BY INDEX ROWID | PERSONEN     |     1 |    32 |     1   (0)| 00:00:
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   4 - access("AD"."PLZ"=:L_PLZ AND "AD"."ORT"=:L_ORT)
   5 - access("PE"."PERSID"="AD"."PERSID")
   6 - filter("PE"."NACHNAME"=:L_NACHNAME)
```

Statistiken

```
Statistiken
-----------------------------------------------------------
          1  recursive calls
          0  db block gets
        474  consistent gets
          1  physical reads
          0  redo size
        922  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

Extended Statistics

CARAJANDB

- Erstellen von erweiterten Statistiken

```
SELECT dbms_stats.create_extended_stats (
        OWNNAME          => 'TUK3',
        TABNAME          => 'ADRESSEN',
        EXTENSION        => '(PLZ,ORT)')
  FROM DUAL;

BEGIN
    dbms_stats.gather_table_stats (
      OWNNAME           => 'TUK3',
      TABNAME           => 'ADRESSEN'
    ,ESTIMATE_PERCENT   => 100
    ,METHOD_OPT         => 'FOR ALL COLUMNS SIZE SKEWONLY'
    ,DEGREE             => NULL
    ,CASCADE            => TRUE
    ,NO_INVALIDATE      => FALSE);
END;
```

Ausführungsplan 2

```
Ausführungsplan
----------------------------------------------------------
Plan hash value: 3657448784

-------------------------------------------------------------------------------------------
| Id  | Operation                     | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |          |     4 |   280 |    75   (2)| 00:00:01 |
|*  1 |  HASH JOIN                    |          |     4 |   280 |    75   (2)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID| PERSONEN |     4 |   128 |     6   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN           | IDX_NAME |     4 |       |     2   (0)| 00:00:01 |
|*  4 |   TABLE ACCESS FULL           | ADRESSEN |   206 |  7828 |    68   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - access("PE"."PERSID"="AD"."PERSID")
   3 - access("PE"."NACHNAME"=:L_NACHNAME)
   4 - filter("AD"."PLZ"=:L_PLZ AND "AD"."ORT"=:L_ORT)
```

Statistiken

```
Statistiken
----------------------------------------------------------
          8  recursive calls
          0  db block gets
        196  consistent gets
          0  physical reads
          0  redo size
        922  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

Extended Statistics

- Löschen von erweiterten Statistiken

```
BEGIN
  dbms_stats.drop_extended_stats(
    OWNNAME   => 'TUK2',
    TABNAME   => 'ADRESSEN',
    EXTENSION => '(PLZ,ORT)');
END;
```

Delete eines Satzes

```
SQL> DELETE FROM mitarbeiter
  2  WHERE mit_id=102075;

1 Zeile wurde gelöscht.
```

Ausführungsplan

```
Plan hash value: 3730998038

-------------------------------------------------------------------------------
| Id  | Operation          | Name        | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | DELETE STATEMENT   |             |     1 |    26 |     1   (0)| 00:00:01 |
|   1 |  DELETE            | MITARBEITER |       |       |            |          |
|*  2 |   INDEX UNIQUE SCAN| PK_MIT_ID   |     1 |    26 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("MIT_ID"=102075)
```

… und die Statistiken?

```
Statistiken
-----------------------------------------------------------
         14  recursive calls
         10  db block gets
       1197  consistent gets
          0  physical reads
        816  redo size
        839  bytes sent via SQL*Net to client
        800  bytes received via SQL*Net from client
          3  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

Index auf Foreign Key

- Kein Index:
  - Table Lock auf die Detail Tabelle, wenn Update auf Primary Key der Master Tabelle
- Updates auf den Primary Key sind selten

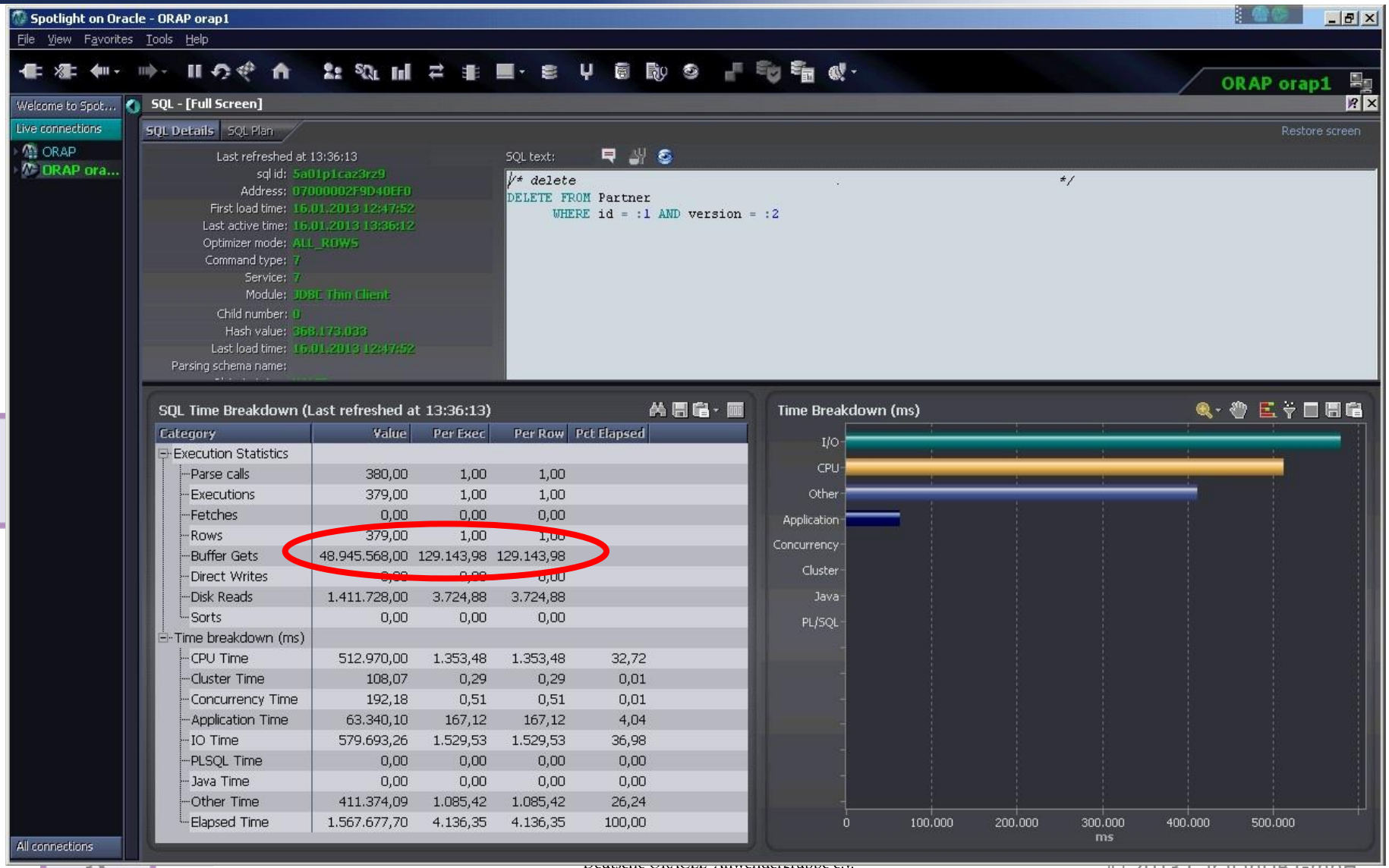
  → Index auf Foreign Key nicht erforderlich

Aber:

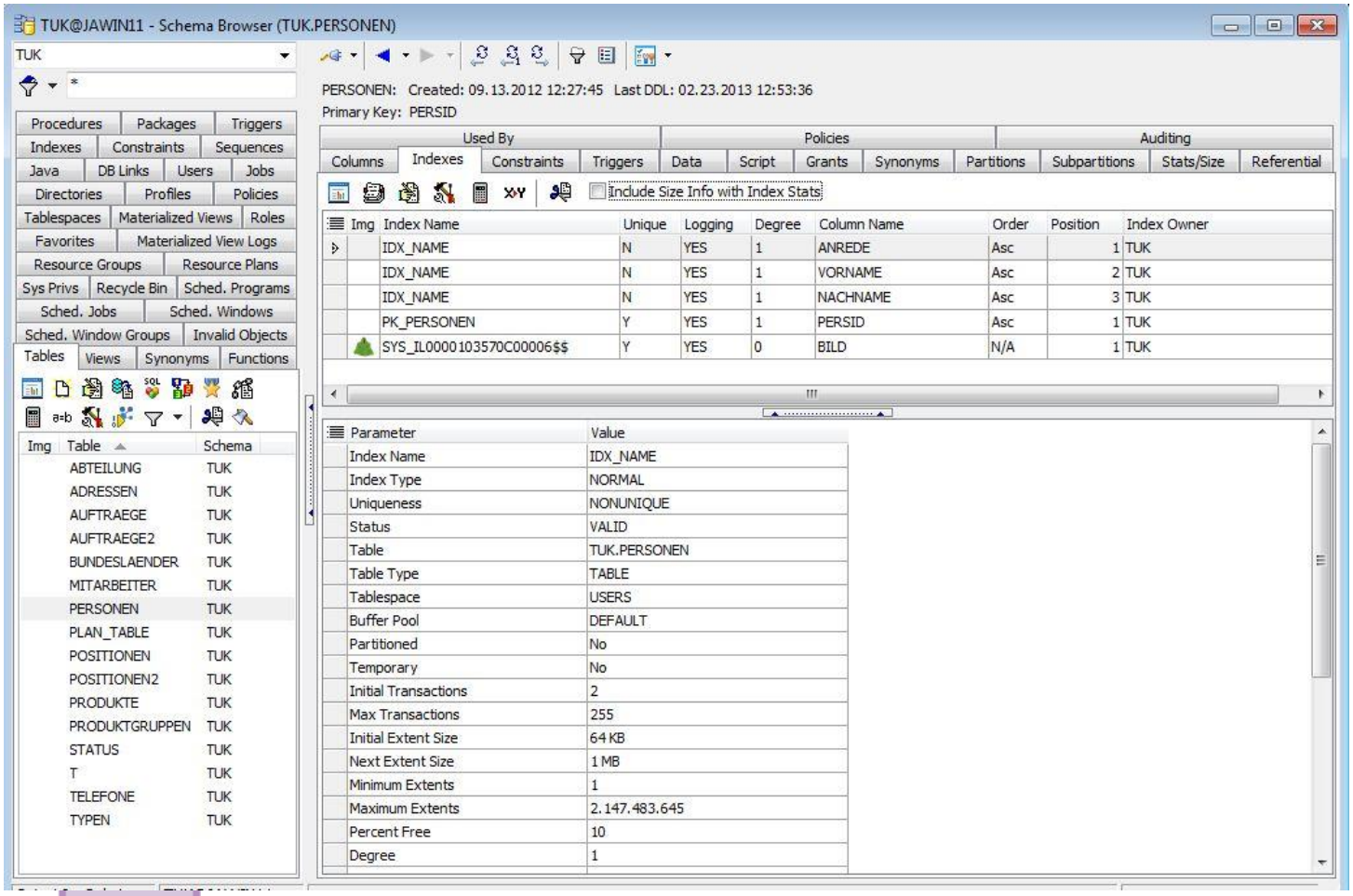
- DELETE FROM <Master Tabelle>

  → FULL TABLE SCAN auf die <Detail Tabelle>

Index auf Foreign Key

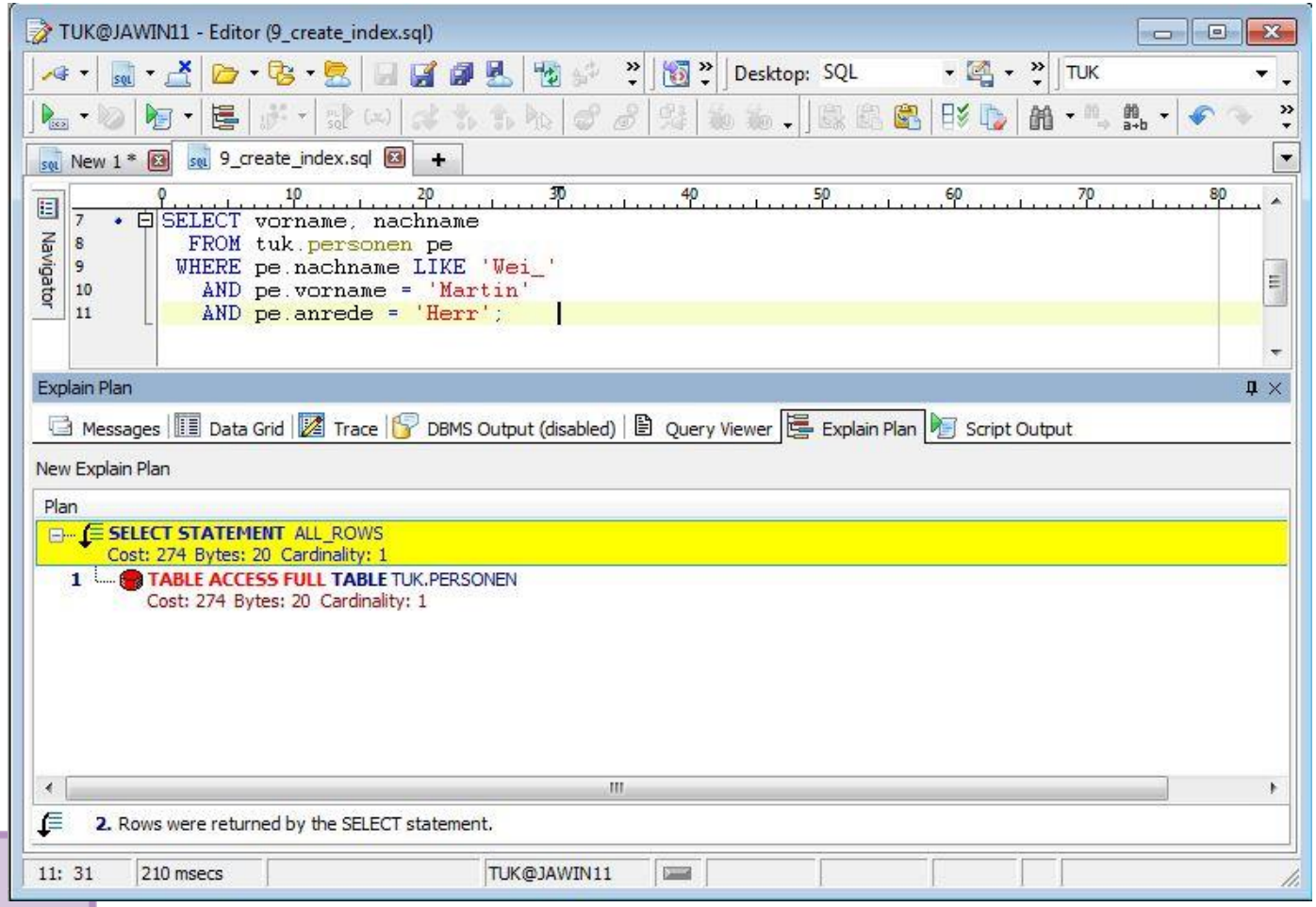


Zusammengesetzter Index



9_concat_iindex.bat

… und trotzdem Full Table Scan!?



Invisible Index

- Erstellen von Indizes zum Testen

1. Test, ob Index tatsächlich genutzt wird
   → CREATE INDEX … INVISIBLE
2. Test, ob Index notwendig ist
   → ALTER INDEX … INVISIBLE

Histogrammme

- **Was passiert, wenn ein Wert nicht gleich verteilt ist?**
- **Z.B. AUFTRAGSSTATUS**
  - Geliefert hoffentlich über 90%
  - Storniert hoffentlich nur 1 – 2 %

```
SELECT …
  FROM personen p, auftraege a
 WHERE p.persid = a.persid
   AND a.aufstatus = 'G';
```

  - Wenn l_aufstatus = ‘G‘ ist, dann besser Full-Tables Scan
  - Wenn l_aufstatus = ‘S‘ ist, dann besser Index Range Scan

Zusammenfassung

- Auch mehrere Indizes können gleichzeitig verwendet werden → Bitmap conversion
- Bei zusammengesetzten Indizes Reihenfolge der Spalten beachten (trotz „Skip Scan“)
- Funktionen setzen Indizes außer Kraft (z.B: NLS Parameter)
- Vorsicht bei der Annahme eine „ID“ oder gar ein „NR“ sei eine Nummer!
- Indizes auf Foreign Keys sind durchaus sinnvoll (DELETE)
- Invisible Indizes helfen nicht nur beim Anlegen sondern auch beim Löschen von Indizes
- Erweiterte Statistiken sind hilfreich
- Histogramme nutzen aber cursor_sharing=FORCE vermeiden
- … und immer die Systemparameter und –statistiken beachten!

Wenn Sie Hilfe brauchen →

CARAJANDB

- **Experten mit über 30 Jahren Oracle Erfahrung**
- **Spezialisten für**
  - Backup & Recovery
  - Hochverfügbarkeit
  - Healthchecks
  - Performance Optimierung
  - Einsatz von Oracle Standard Edition
  - Oracle in virtuellen Umgebungen und in der Cloud
  - Oracle Migrationen (HW, Unicode, Konsolidierung, Standard Edition)
  - Monitoring (Grid / Cloud Control, HLMM, Foglight, Spotlight)
- **Schulung und Workshops (Oracle, Toad)**

DOAG
Deutsche ORACLE-Anwendergruppe e.V.

Oracle Standard Edition

CARAJANDB

Funktion	Standard Edition	Enterprise Edition
Data Guard	**NEIN**	**JA**
Active Data Guard	**NEIN**	**Option**
Online Table und Index Rebuild	NEIN	JA
Parallel DML and DDL	NEIN	JA
Flashback Query	JA	JA
Flashback Table, Database, Transaction Query	NEIN	JA
Flashback Data Archive (Total Recall)	NEIN	Option
Streams (inklusive Capture)	NEIN	JA
Online und Incremental Backup and Recovery	**JA**	**JA**
Advanced Compression	NEIN	Option
Bitmapped Index und Bitmapped Join Index	NEIN	JA
Oracle Real Application Clusters	**JA**	**Option**
Partitioning	NEIN	Option
Transportable Tablespaces	NEIN	JA
AWR, ADDM, ASH	**NEIN**	**Option**

Vollständige Liste: Oracle Database Licensing Information 11g Release 2 (11.2) Part Number E10594-18

Lizensierung für VMware

- **Zitat DOAG Lizensierung Competence Center:**
  - Der gesamte Cluster muss lizenziert werden! Dies ist eine grundsätzliche Regel bei Softpartitionierung, der VMWare und HyperV zugerechnet werden. Die Art der Automatisierung ist hierbei für die Lizenzierung unerheblich. Sofern in dem Cluster kein Server über mehr als zwei Prozessorsockel verfügt, kann – wenn es von der Funktionalität her reicht – die DB SE1 lizenziert werden. Dies muss allerdings für alle bestückten Prozessorsockel geschehen.

    http://www.doag.org/doag/competence-center/lizenz/fragen-und-antworten.html

CARAJAN DB

Fragen?

Johannes Ahrends

ORACLE ACE

www.carajandb.com

Johannes.ahrends@carajandb.com

DOAG

Deutsche ORACLE-Anwendergruppe e.V.