

Avoid „Horror Queries“

Keep Your SQL Simple and Fast

Dani Schnider



@dani_schnider



danischnider.wordpress.com

BASEL | BERN | BRUGG | BUCHAREST | DÜSSELDORF | FRANKFURT A.M. | FREIBURG I.BR. | GENEVA
HAMBURG | COPENHAGEN | LAUSANNE | MANNHEIM | MUNICH | STUTTGART | VIENNA | ZURICH

trivadis

Dani Schnider

- Senior Principal Consultant at Trivadis AG in Glattbrugg/Zurich
- Trainer of several Trivadis courses
- Co-Author of Books “Data Warehousing mit Oracle” and “Data Warehouse Blueprints”
- Oracle ACE



 @dani_schnider

 danischnider.wordpress.com



BASEL | BERN | BRUGG | BUCHAREST | COPENHAGEN | DÜSSELDORF | FRANKFURT A.M. | FREIBURG I. BR.
GENEVA | HAMBURG | LAUSANNE | MANNHEIM | MUNICH | STUTTGART | VIENNA | ZURICH

trivadis

Note on this presentation



- The session presents five examples of complex SQL statements that are simplified step-by-step to improve performance. The slides consist of numerous individual steps that are difficult to understand without a "soundtrack".
- For this reason, many slides have been omitted from this download version. Instead, a brief explanation is given for each example of the actions used to improve query performance.

Reduction of Complexity

Reduction of Complexity

trivadis

```
SELECT CAST('GELMARET Anzahl Records pro ARTLESSART' AS CHAR(50)) AS KRAHLID,
CASE
  WHEN CONVERT(tabell.sinnseite, 'WERRCHD1C500', 'UTF8') < CONVERT(CAST('00000001.0000000' AS CHAR(17)), 'WERRCHD1C500', 'UTF8')
  THEN odatt.LISTAG(CONCAT(UNISTR(0), CAST(TO_CHAR(LTRIM(tabell.sinnseite, '0')) AS CHAR(50)))
  ELSE CAST(TO_CHAR(LTRIM(tabell.sinnseite, '0')) AS CHAR(50))
END
CAST(TO_CHAR(CAST(COUNT(*) AS NUMBER(15, 0)), '0000000000000000') AS CHAR(17)) AS KEYFIELD,
tabell.mandant,
AS WERRCHD,
FROM
(
  SELECT DISTINCT gesped.par_id AS parnr,
        CAST(gesped.gesch_id AS CHAR(11)) AS applnr,
        CAST(SUBSTR(gesped.gesch_id, 2, 2) AS CHAR(2)) AS hbnr,
        CAST(SUBSTR(gesped.gesch_id, 4, 3) AS CHAR(3)) AS sennr,
        CAST(SUBSTR(gesped.gesch_id, 7, 8) AS CHAR(2)) AS geschnr,
        CAST(LTRIM(TO_CHAR(CAST(hfcall.abw_atned_sas_pt AS NUMBER(15, 7)), '000000')) AS sinnseite,
        AS mandant
  FROM
    odatt.mtgesped gesped
  JOIN odatt.mtgesped gesped
    ON gesped.lnh_par_tech_id = gesped.lnh_par_tech_id
    AND gesped.proktrtp_art_cd IN ('000')
    AND gesped.proktrtp_ttyp_cd = '033'
  JOIN odatt.mtgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.gg_vers_ttyp_cd = '000V'
    AND gesped.spl_proktrtp_art_cd = '009'
    AND gesped.spl_proktrtp_ttyp_cd = '077'
  JOIN odatt.lbfcall hfcall
    ON hfcall.kto_gesch_tech_id = gesped.gesch_tech_id
  JOIN odatt.lbfkto hfcall
    ON hfcall.kto_gesch_tech_id = hfcall.kto_gesch_tech_id
  JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
  WHERE gesped.gesch_tech_id IN ('C8K4', 'C8K2')
  AND gesped.gg_vers_ttyp_cd IN ('001', '059')
  AND gesped.proktrtp_art_cd IN ('030', '059')
  UNION
  SELECT DISTINCT gesped.par_id AS parnr,
        CAST(gesped.gesch_id AS CHAR(11)) AS applnr,
        CAST(SUBSTR(gesped.gesch_id, 2, 2) AS CHAR(2)) AS hbnr,
        CAST(SUBSTR(gesped.gesch_id, 4, 3) AS CHAR(3)) AS sennr,
        CAST(SUBSTR(gesped.gesch_id, 7, 8) AS CHAR(2)) AS geschnr,
        CAST(LTRIM(TO_CHAR(CAST(hfcall.abw_atned_sas_pt AS NUMBER(15, 7)), '00000000.0000000')) AS CHAR(17)) AS sinnseite,
        AS mandant
  FROM
    odatt.mtgesped gesped
  JOIN odatt.mtgesped gesped
    ON gesped.lnh_par_tech_id = gesped.lnh_par_tech_id
    AND gesped.proktrtp_art_cd IN ('000')
    AND gesped.proktrtp_ttyp_cd = '033'
  JOIN odatt.mtgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.gg_vers_ttyp_cd = '000V'
    AND gesped.spl_proktrtp_art_cd = '009'
    AND gesped.spl_proktrtp_ttyp_cd = '077'
  JOIN odatt.lbfcall hfcall
    ON hfcall.kto_gesch_tech_id = gesped.gesch_tech_id
  AND hfcall.lfst_Els_dt >= to_date('2018-03-28', 'yyyy-mm-dd')
  JOIN odatt.lbfkto hfcall
    ON hfcall.kto_gesch_tech_id = hfcall.kto_gesch_tech_id
  JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
  WHERE gesped.gesch_tech_id IN ('C8K4', 'C8K2')
  AND gesped.proktrtp_art_cd = '001'
  AND gesped.proktrtp_ttyp_cd IN ('030', '059')
  UNION
  SELECT DISTINCT gesped.par_id AS parnr,
        CAST(gesped.gesch_id AS CHAR(11)) AS applnr,
        CAST(SUBSTR(gesped.gesch_id, 2, 2) AS CHAR(2)) AS hbnr,
        CAST(SUBSTR(gesped.gesch_id, 4, 3) AS CHAR(3)) AS sennr,
        CAST(SUBSTR(gesped.gesch_id, 7, 8) AS CHAR(2)) AS geschnr,
        CAST(LTRIM(TO_CHAR(CAST(gesped.sas_pt AS NUMBER(15, 7)), '00000000.0000000')) AS CHAR(17)) AS sinnseite,
        AS mandant
  FROM
    odatt.mtgesped gesped
  JOIN odatt.mtgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.gg_vers_ttyp_cd IN ('000V', '000V')
    AND gesped.spl_proktrtp_art_cd = '009'
    AND gesped.spl_proktrtp_ttyp_cd = '077'
  JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
  AND gesped.lnh_ttyp_cd IN ('MORTCALL', 'MORTTIME', 'FIDT', 'FIDC', 'FIDT_TIME', 'FIDT_CALL')
  LEFT JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.poa_tech_id = gesped.poa_tech_id
  LEFT JOIN
  (
    SELECT gde.gesch_tech_id,
          poa_tech_id,
          buch_stat_cd,
          FROM odatt.TGDEGMA gdegma
          WHERE gdegma.tr_dt =
            SELECT MAX(gdegma.tr_dt)
            FROM odatt.TGDEGMA gdegma
            WHERE gdegma.poa_tech_id = CAST(gdegma.poa_tech_id AS CHAR(255))
            AND gdegma.abw_atned_sas_pt = to_date('31.12.2999', 'DD.MM.YYYY')
          )
    ON gdegma.gde.gesch_tech_id = gesped.gesch_tech_id
    AND gdegma.poa_tech_id = gesped.poa_tech_id
    AND gdegma.buch_stat_cd = '0'
  LEFT JOIN
  (
    SELECT gde.gesch_tech_id,
          poa_tech_id,
          buch_stat_cd,
          FROM odatt.TGDEGMA gdegma
          WHERE gdegma.tr_dt =
            SELECT MAX(gdegma.tr_dt)
            FROM odatt.TGDEGMA gdegma
            WHERE gdegma.poa_tech_id = CAST(gdegma.poa_tech_id AS CHAR(255))
            AND gdegma.abw_atned_sas_pt = to_date('31.12.2999', 'DD.MM.YYYY')
          )
    ON gdegma.gde.gesch_tech_id = gesped.gde.gesch_tech_id
    AND gdegma.poa_tech_id = gesped.poa_tech_id
    AND gdegma.buch_stat_cd = '0'
  LEFT JOIN odatt.tgsktiv gsktiv
    ON gsktiv.gde.gesch_tech_id = gesped.gesch_tech_id
    AND gsktiv.lao_auss_wing_cd = gesped.lao_poa_wing_cd
  WHERE gesped.proktrtp_art_cd = '001'
  AND gesped.proktrtp_ttyp_cd = '077'
)
)
GROUP BY tabell.sinnseite,
tabell.mandant
```

> 1 hour

```
SELECT CAST('GELMARET Anzahl Records pro ARTLESSART' AS CHAR(50)) AS KRAHLID,
CASE
  WHEN CONVERT(tabell.sinnseite, 'WERRCHD1C500', 'UTF8') < CONVERT(CAST('00000001.0000000' AS CHAR(17)), 'WERRCHD1C500', 'UTF8')
  THEN odatt.LISTAG(CONCAT(UNISTR(0), CAST(TO_CHAR(LTRIM(tabell.sinnseite, '0')) AS CHAR(50)))
  ELSE CAST(TO_CHAR(LTRIM(tabell.sinnseite, '0')) AS CHAR(50))
END
CAST(TO_CHAR(CAST(COUNT(*) AS NUMBER(15, 0)), '0000000000000000') AS CHAR(17)) AS KEYFIELD,
tabell.mandant,
AS WERRCHD,
FROM
(
  SELECT gesped.par_id AS parnr,
        CAST(gesped.gesch_id AS CHAR(11)) AS applnr,
        CAST(SUBSTR(gesped.gesch_id, 2, 2) AS CHAR(2)) AS hbnr,
        CAST(SUBSTR(gesped.gesch_id, 4, 3) AS CHAR(3)) AS sennr,
        CAST(SUBSTR(gesped.gesch_id, 7, 8) AS CHAR(2)) AS geschnr,
        CAST(LTRIM(TO_CHAR(CAST(gesped.sas_pt AS NUMBER(15, 7)), '00000000.0000000')) AS CHAR(17)) AS sinnseite,
        AS mandant
  FROM
    odatt.mtgesped gesped
  JOIN odatt.mtgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.gg_vers_ttyp_cd = '000V'
    AND gesped.spl_proktrtp_art_cd = '009'
    AND gesped.spl_proktrtp_ttyp_cd = '077'
  JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
  AND gesped.lnh_ttyp_cd IN ('MORTCALL', 'MORTTIME', 'FIDT', 'FIDC', 'FIDT_TIME', 'FIDT_CALL')
  LEFT JOIN odatt.tgesped gesped
    ON gesped.gesch_tech_id = gesped.gesch_tech_id
    AND gesped.poa_tech_id = gesped.poa_tech_id
  LEFT JOIN odatt.tgsktiv gsktiv
    ON gsktiv.gde.gesch_tech_id = gesped.gesch_tech_id
    AND gsktiv.lao_auss_wing_cd = gesped.lao_poa_wing_cd
  WHERE gesped.proktrtp_art_cd = '001'
  AND gesped.proktrtp_ttyp_cd = '077'
)
)
GROUP BY tabell.sinnseite,
tabell.mandant
```

2 seconds

Performance Tuning without Database Access

```

SELECT S1.DEPTNO,
       S1.JOB,
       MIN (S1.EMPNO) AS EMPNO,
       LTRIM (MAX (SYS_CONNECT_BY_PATH (S1.EMPNO, ';'))
              KEEP (DENSE_RANK LAST ORDER BY S1.curr), ';') AS EMPFAENGERLISTE
FROM (SELECT DEPTNO,
             JOB,
             EMPNO,
             ROW_NUMBER () OVER (PARTITION BY DEPTNO, JOB
                                  ORDER BY EMPNO) AS curr,
             ROW_NUMBER () OVER (PARTITION BY DEPTNO, JOB
                                  ORDER BY EMPNO) - 1 AS prev
      FROM EMP
      WHERE DEPTNO > 0) S1
GROUP BY S1.DEPTNO, S1.JOB
CONNECT BY      S1.prev = PRIOR S1.curr
               AND S1.DEPTNO = PRIOR S1.DEPTNO
START WITH curr = 1

SELECT DEPTNO,
       JOB,
       MIN (EMPNO) AS EMPNO,
       LISTAGG(EMPNO, ';')
              WITHIN GROUP (ORDER BY EMPNO)
       AS EMPFAENGERLISTE
FROM EMP
WHERE DEPTNO > 0
GROUP BY DEPTNO, JOB

```

In the second example, a complex query was replaced by a simpler statement using LISTAGG, which runs much faster and is easier to understand. This example is described in detail here:

<https://danischnider.wordpress.com/2018/04/03/keep-your-sql-simple-and-fast/>

An Expensive Attribute

Simple Query on a View

```
SELECT state_filing_number,  
       legal_entity_name,  
       toi_desc,  
       sub_toi_desc,  
       state_full_desc,  
       project_lead,  
       state_lead,  
       pr_state_status,  
       response_date,  
       form_type_desc,  
       state_tracking_num,  
       serff_track_nbr,  
       circular_number_txt,  
       request_effective_date,  
       effective_date,  
       status date,  
       state_req_uid  
FROM rah vw_state_tab_data  
WHERE legal_entity_id IN (6)  
ORDER BY state_desc;
```

The calculation of an attribute STATUS_DATE in a view became very inefficient due to several nested subselects. Using a subquery factoring clause (WITH statement) and a CASE statement, the response time has been reduced from several minutes to a few seconds.

Make it Faster Step by Step

Query on daily accounting data too slow

trivadis

By using several performance features in two nested views, it was possible to speed up the frequent query for a single date step by step.

```
SELECT SECURITY_CODE, RATING_AGENCY, RATING, RATING_DATE  
FROM V_IF_SRAM_INSTR_RATINGS  
WHERE ACCOUNTING_DATE = TO_DATE('20180513', 'yyyymmdd');
```

> 2 hours

29 minutes

1 minute

12 seconds

trivadis

A PL/SQL Journey



Dani Schnider
@dani_schnider

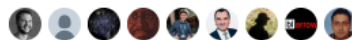
Calling **#PLSQL** functions in a **#SQL** query, and the functions run other **#SQL** queries with **#PLSQL** functions than call **#SQL** statements with **#PLSQL** functions 😞

#Performance tuning issues on a Friday afternoon - and another example for SQL **#HorrorShow**



1:09 PM - 24 May 2019

6 Retweets 46 Likes



5



6



46



It is a very bad idea to call PL/SQL functions containing SQL statements in the SELECT or WHERE part of a SQL statement because the locally called SQL statement is executed for every row of the query. If this local SQL statement calls other PL/SQL functions, the performance of the main query is getting worse and worse.

Summary

Summary

- SQL Performance Tuning is Hard Work
 - ... but effective
 - Massive performance improvements are possible
- Reduce complexity of SQL statements
 - Filtering the data set as soon as possible
 - Avoid multiple executions
 - The easier, the better – for us and for the optimizer



Keep Your SQL Simple and Fast

