EXPLAIN QUERY PLAN REPORT on 1_gengrant

generated at 2025-09-15T00:06:34

Table of contents:

1. SQL statement q1_gengrants_query.sql

```
-----------------------------------  SQLITE -------------------------------------------------
-- TABLES_GRANTS_EXIST contains processing groups for the requered access check to generate commands
-- TABLES_GRANTS_MUST_HAVE contains set up for grants of processing groups for generating commands
-- GRANTEES - group to be granted
-- SELECTAUTH, UPDATEAUTH, DELETEAUTH, INSERTAUTH - what should be granted
-- TABLEPREFIX - prefix of table's name or whole table's name for which granted
with TABLES_GRANTS_MUST_HAVE
  (GRANTEES, TABLEPREFIX, SELECTAUTH, UPDATEAUTH, DELETEAUTH, INSERTAUTH)
as
 (select 'CRANTEE_MUST_A', 'TABPR',    'SELECT',         '',        '',        '' from SYSDUMMY1 union
  select 'CRANTEE_MUST_A', 'TABPRM',   'SELECT',    'UPDATE', 'DELETE', 'INSERT' from SYSDUMMY1 union
  select 'CRANTEE_MUST_A', 'TABPRADMT','SELECT',    'UPDATE', 'DELETE', 'INSERT' from SYSDUMMY1 union
  select 'CRANTEE_MUST_B', 'TABPR',    'SELECT',         '',        '',        '' from SYSDUMMY1
--union
-- select 'CRANTEE_MUST_C', 'TABPR',    'SELECT',         '',        '',        '' from SYSDUMMY1
--union
-- select 'CRANTEE_ADM' ,   'TABPR',    'SELECT',         '',        '',        '' from SYSDUMMY1
--union
-- select 'CRANTEE_MUST_E', 'TABPR',          '',         '',        '',        '' from SYSDUMMY1
  ),
--  TABLES_GRANTS_MUST_HAVE
--  (GRANTEES, TABLEPREFIX, SELECTAUTH, UPDATEAUTH, DELETEAUTH, INSERTAUTH)
--as
--  (select GRANTEES, TABLEPREFIX, MAX(SELECTAUTH), MAX(UPDATEAUTH), MAX(DELETEAUTH), MAX(INSERTAUTH)
--   from TABLES_GRANTS_REQUIRED
--   group by GRANTEES, TABLEPREFIX
--  ),
  TABLES_GRANTS_EXIST
  (TCREATOR, TTNAME, GRANTEES, SELECTAUTH, INSERTAUTH, UPDATEAUTH, DELETEAUTH)
as
  (select STABAUTH.TCREATOR, STABAUTH.TTNAME, STABAUTH.GRANTEES,
    case when max(STABAUTH.SELECTAUTH)  in ('Y','G')
      then 'SELECT' else ''
    end as SELECTAUTH,
    case when max(STABAUTH.INSERTAUTH)  in ('Y','G')
      then 'INSERT' else ''
    end as INSERTAUTH,
    case when max(STABAUTH.UPDATEAUTH)  in ('Y','G')
      then 'UPDATE' else ''
    end as UPDATEAUTH,
    case when max(STABAUTH.DELETEAUTH)  in ('Y','G')
      then 'DELETE' else ''
    end as DELETEAUTH
  from SYSTABLES STAB, SYSTABAUTH STABAUTH
    where
        STAB.TYPE    =  'T'
--    and STAB.NAME  like 'TABPR%'
      and STAB.CREATOR =  STABAUTH.TCREATOR
      and STAB.NAME    =  STABAUTH.TTNAME
      and STAB.CREATOR =  'CRANTEE_ADM'
    group by STABAUTH.TCREATOR, STABAUTH.TTNAME, STABAUTH.GRANTEES
  )
------- GENERATE REVOKE STATEMENTS
select " A list of generated SQL statements on GRANTS modifications reflected TABLES_GRANTS_MUST_HAVE
requirements. A processing time stamp: " || strftime( datetime(current_timestamp, 'localtime')) from
SYSDUMMY1 as command
UNION
select  (' REVOKE '
  || rtrim
      (case when (TG_EXIST.SELECTAUTH <> '' and TG_MUST_HAVE.SELECTAUTH = '')
          then TG_EXIST.SELECTAUTH || ', ' else ''
      end ||
      case when (TG_EXIST.DELETEAUTH <> ''  and TG_MUST_HAVE.DELETEAUTH = '')
          then TG_EXIST.DELETEAUTH || ', ' else ''
      end ||
      case when (TG_EXIST.INSERTAUTH <> ''  and TG_MUST_HAVE.INSERTAUTH = '')
          then TG_EXIST.INSERTAUTH || ', ' else ''
      end ||
      case when (TG_EXIST.UPDATEAUTH <> ''  and TG_MUST_HAVE.UPDATEAUTH = '')
          then TG_EXIST.UPDATEAUTH            else ''
      end,' ,')
  || ' ON TABLE ' || trim(TG_EXIST.TTNAME)
  || ' from '     || TG_MUST_HAVE.GRANTEES || ';') as command
from TABLES_GRANTS_MUST_HAVE as TG_MUST_HAVE inner join TABLES_GRANTS_EXIST as TG_EXIST
  on
        instr(trim(TG_EXIST.TTNAME),trim(TG_MUST_HAVE.TABLEPREFIX))>0
    and TG_MUST_HAVE.GRANTEES    = TG_EXIST.GRANTEES
    and TG_MUST_HAVE.TABLEPREFIX = (select max(MH.TABLEPREFIX)
                                    from TABLES_GRANTS_MUST_HAVE as MH
                                      where
                                            instr(trim(TG_EXIST.TTNAME),trim(MH.TABLEPREFIX))>0
                                        and TG_EXIST.GRANTEES = MH.GRANTEES
                                        group by MH.GRANTEES)
    and  ((TG_MUST_HAVE.SELECTAUTH = '' and TG_EXIST.SELECTAUTH <> '')
```

```
        or(TG_MUST_HAVE.DELETEAUTH = '' and TG_EXIST.DELETEAUTH <> '')
        or(TG_MUST_HAVE.INSERTAUTH = '' and TG_EXIST.INSERTAUTH <> '')
        or(TG_MUST_HAVE.UPDATEAUTH = '' and TG_EXIST.UPDATEAUTH <> ''))
---1---- GENERATE GRANT STATEMENTS
union
select  (' GRANT '
  || rtrim
      (case when (TG_EXIST.SELECTAUTH = '' and TG_MUST_HAVE.SELECTAUTH <> '')
          then TG_MUST_HAVE.SELECTAUTH || ', ' else '' end ||
      case when (TG_EXIST.DELETEAUTH = '' and TG_MUST_HAVE.DELETEAUTH <> '')
          then TG_MUST_HAVE.DELETEAUTH || ', ' else '' end ||
      case when (TG_EXIST.INSERTAUTH = '' and TG_MUST_HAVE.INSERTAUTH <> '')
          then TG_MUST_HAVE.INSERTAUTH || ', ' else '' end ||
      case when (TG_EXIST.UPDATEAUTH = '' and TG_MUST_HAVE.UPDATEAUTH <> '')
          then TG_MUST_HAVE.UPDATEAUTH         else '' end,' ,')
  || ' ON TABLE '
  || rtrim (TG_EXIST.TTNAME) || ' TO ' ||
           TG_MUST_HAVE.GRANTEES || ';') as command
from  TABLES_GRANTS_EXIST as TG_EXIST inner join TABLES_GRANTS_MUST_HAVE as TG_MUST_HAVE
  on
       TG_MUST_HAVE.GRANTEES = TG_EXIST.GRANTEES
    and  instr(trim(TG_EXIST.TTNAME),trim(TG_MUST_HAVE.TABLEPREFIX))>0
    and  ((TG_MUST_HAVE.SELECTAUTH <> '' and TG_EXIST.SELECTAUTH = '')
       or (TG_MUST_HAVE.DELETEAUTH <> '' and TG_EXIST.DELETEAUTH = '')
       or (TG_MUST_HAVE.INSERTAUTH <> '' and TG_EXIST.INSERTAUTH = '')
       or (TG_MUST_HAVE.UPDATEAUTH <> '' and TG_EXIST.UPDATEAUTH = ''))
    and TG_MUST_HAVE.TABLEPREFIX = (select  max(MH.TABLEPREFIX)
                                    from TABLES_GRANTS_MUST_HAVE as MH
                                      where
                                           instr(trim(TG_EXIST.TTNAME),trim(MH.TABLEPREFIX))>0
                                       and TG_EXIST.GRANTEES = MH.GRANTEES
                                    group by MH.GRANTEES)
union
---2---- GENERATE GRANT STATEMENTS
select  (' GRANT ' ||
  trim(
    (case when (TG_MUST_HAVE.SELECTAUTH <> '' )
          then TG_MUST_HAVE.SELECTAUTH   else ''
   end ||
     case when (TG_MUST_HAVE.DELETEAUTH <> '')
          then ', '||TG_MUST_HAVE.DELETEAUTH  else ''
   end ||
    case when (TG_MUST_HAVE.INSERTAUTH <> '')
          then ', '||TG_MUST_HAVE.INSERTAUTH  else ''
   end ||
   case when   (TG_MUST_HAVE.UPDATEAUTH <> '' )
        then ', '||TG_MUST_HAVE.UPDATEAUTH         else ''
  end),', ')
  || ' ON TABLE ' || TG_EXIST.TTNAME
  || ' TO ' || TG_MUST_HAVE.GRANTEES || ';') as command
from TABLES_GRANTS_EXIST as TG_EXIST inner join  TABLES_GRANTS_MUST_HAVE as TG_MUST_HAVE
  on
    instr(trim(TG_EXIST.TTNAME), trim(TG_MUST_HAVE.TABLEPREFIX))>0
    and TG_MUST_HAVE.TABLEPREFIX = (select max(MH.TABLEPREFIX)
                                    from TABLES_GRANTS_MUST_HAVE as MH
                                      where
                                           instr(trim(TG_EXIST.TTNAME),trim(MH.TABLEPREFIX))>0
                                       and TG_MUST_HAVE.GRANTEES = MH.GRANTEES
                                    group by MH.GRANTEES)
  group by TG_EXIST.TTNAME,
      TG_MUST_HAVE.GRANTEES, TG_MUST_HAVE.TABLEPREFIX,
      TG_MUST_HAVE.SELECTAUTH, TG_MUST_HAVE.DELETEAUTH, TG_MUST_HAVE.INSERTAUTH ,TG_MUST_HAVE.UPDATEAUTH
   having  sum(instr(trim(TG_EXIST.GRANTEES),trim(TG_MUST_HAVE.GRANTEES))) = 0
     and NOT (  TG_MUST_HAVE.SELECTAUTH = ''
            and TG_MUST_HAVE.DELETEAUTH = ''
            and TG_MUST_HAVE.INSERTAUTH = ''
            and TG_MUST_HAVE.UPDATEAUTH = '')
```

2. Explain query plan generated by <EXPLAINE QUERY PLAN sql_statement>

2.1. Plain report

step	id	parent	unused	detail
1	2	0	0	COMPOUND QUERY
2	3	2	0	LEFT-MOST SUBQUERY
3	5	3	216	SCAN command
4	16	2	0	UNION USING TEMP B-TREE
5	19	16	0	MATERIALIZE TABLES_GRANTS_MUST_HAVE
6	22	19	0	COMPOUND QUERY
7	23	22	0	LEFT-MOST SUBQUERY
8	25	23	216	SCAN SYSDUMMY1
9	36	22	0	UNION USING TEMP B-TREE
10	38	36	216	SCAN SYSDUMMY1
11	49	22	0	UNION USING TEMP B-TREE
12	51	49	216	SCAN SYSDUMMY1
13	62	22	0	UNION USING TEMP B-TREE
14	64	62	216	SCAN SYSDUMMY1
15	90	16	0	MATERIALIZE TABLES_GRANTS_EXIST
16	98	90	216	SCAN STABAUTH
17	118	90	53	SEARCH STAB USING AUTOMATIC PARTIAL COVERING INDEX (TYPE=? AND CREATOR=? AND NAME=?)
18	126	90	0	USE TEMP B-TREE FOR GROUP BY
19	211	16	82	SCAN TG_EXIST
20	217	16	0	BLOOM FILTER ON TG_MUST_HAVE (TABLEPREFIX=?)
21	231	16	54	SEARCH TG_MUST_HAVE USING AUTOMATIC COVERING INDEX (TABLEPREFIX=?)
22	233	16	0	CORRELATED SCALAR SUBQUERY 7
23	242	233	216	SCAN MH
24	356	2	0	UNION USING TEMP B-TREE
25	361	356	216	SCAN TG_EXIST
26	365	356	0	BLOOM FILTER ON TG_MUST_HAVE (TABLEPREFIX=?)
27	379	356	53	SEARCH TG_MUST_HAVE USING AUTOMATIC COVERING INDEX (TABLEPREFIX=?)
28	381	356	0	CORRELATED SCALAR SUBQUERY 9
29	390	381	216	SCAN MH
30	504	2	0	UNION USING TEMP B-TREE
31	513	504	216	SCAN TG_MUST_HAVE
32	523	504	216	SCAN TG_EXIST
33	533	504	0	CORRELATED SCALAR SUBQUERY 11
34	544	533	0	BLOOM FILTER ON MH (GRANTEES=?)
35	554	533	53	SEARCH MH USING AUTOMATIC COVERING INDEX (GRANTEES=?)
36	566	533	0	USE TEMP B-TREE FOR GROUP BY
37	602	504	0	USE TEMP B-TREE FOR GROUP BY

2.2. GRAPH report

```
          EXPLAINE QUERY PLAN
step 1....  |__COMPOUND QUERY...node(id: 2)
step 2....     |--LEFT-MOST SUBQUERY...node(id: 3)
step 3....     |  |__SCAN command...node(id: 5, notused: 216)
step 4....     |--UNION USING TEMP B-TREE...node(id: 16)
step 5....     |  |--MATERIALIZE TABLES_GRANTS_MUST_HAVE...node(id: 19)
step 6....     |  |  |__COMPOUND QUERY...node(id: 22)
step 7....     |  |     |--LEFT-MOST SUBQUERY...node(id: 23)
step 8....     |  |     |  |__SCAN SYSDUMMY1...node(id: 25, notused: 216)
step 9....     |  |     |--UNION USING TEMP B-TREE...node(id: 36)
step 10...     |  |     |  |__SCAN SYSDUMMY1...node(id: 38, notused: 216)
step 11...     |  |     |--UNION USING TEMP B-TREE...node(id: 49)
step 12...     |  |     |  |__SCAN SYSDUMMY1...node(id: 51, notused: 216)
step 13...     |  |     |__UNION USING TEMP B-TREE...node(id: 62)
step 14...     |  |        |__SCAN SYSDUMMY1...node(id: 64, notused: 216)
step 15...     |  |--MATERIALIZE TABLES_GRANTS_EXIST...node(id: 90)
step 16...     |  |  |--SCAN STABAUTH...node(id: 98, notused: 216)
step 17...     |  |  |--SEARCH STAB USING AUTOMATIC PARTIAL COVERING INDEX (TYPE=? AND CREATOR=? AND NAME=?)...node(id: 118, notused: 53)
step 18...     |  |  |__USE TEMP B-TREE FOR GROUP BY...node(id: 126)
step 19...     |  |--SCAN TG_EXIST...node(id: 211, notused: 82)
step 20...     |  |--BLOOM FILTER ON TG_MUST_HAVE (TABLEPREFIX=?)...node(id: 217)
step 21...     |  |--SEARCH TG_MUST_HAVE USING AUTOMATIC COVERING INDEX (TABLEPREFIX=?)...node(id: 231, notused: 54)
step 22...     |  |__CORRELATED SCALAR SUBQUERY 7...node(id: 233)
step 23...     |     |__SCAN MH...node(id: 242, notused: 216)
step 24...     |--UNION USING TEMP B-TREE...node(id: 356)
step 25...     |  |--SCAN TG_EXIST...node(id: 361, notused: 216)
step 26...     |  |--BLOOM FILTER ON TG_MUST_HAVE (TABLEPREFIX=?)...node(id: 365)
step 27...     |  |--SEARCH TG_MUST_HAVE USING AUTOMATIC COVERING INDEX (TABLEPREFIX=?)...node(id: 379, notused: 53)
step 28...     |  |__CORRELATED SCALAR SUBQUERY 9...node(id: 381)
step 29...     |     |__SCAN MH...node(id: 390, notused: 216)
step 30...     |__UNION USING TEMP B-TREE...node(id: 504)
step 31...        |--SCAN TG_MUST_HAVE...node(id: 513, notused: 216)
step 32...        |--SCAN TG_EXIST...node(id: 523, notused: 216)
step 33...        |--CORRELATED SCALAR SUBQUERY 11...node(id: 533)
step 34...        |  |--BLOOM FILTER ON MH (GRANTEES=?)...node(id: 544)
step 35...        |  |--SEARCH MH USING AUTOMATIC COVERING INDEX (GRANTEES=?)...node(id: 554, notused: 53)
step 36...        |  |__USE TEMP B-TREE FOR GROUP BY...node(id: 566)
step 37...        |__USE TEMP B-TREE FOR GROUP BY...node(id: 602)
```