

SQL results report on 1_gengrant

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

Table of contents:

1. SQL statment 1_gengrant content	3
2. SQL results	5



1. SQL statment 1 gengrant content

```

----- 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 SYSDDUMMY1 union
   select 'CRANTEE_MUST_A', 'TABPRM', 'SELECT', 'UPDATE', 'DELETE', 'INSERT' from SYSDDUMMY1 union
   select 'CRANTEE_MUST_A', 'TABPRADMT', 'SELECT', 'UPDATE', 'DELETE', 'INSERT' from SYSDDUMMY1 union
   select 'CRANTEE_MUST_B', 'TABPR', 'SELECT', '', '', '' from SYSDDUMMY1
  --union
  -- select 'CRANTEE_MUST_C', 'TABPR', 'SELECT', '', '', '' from SYSDDUMMY1
  --union
  -- select 'CRANTEE_ADM', 'TABPR', 'SELECT', '', '', '' from SYSDDUMMY1
  --union
  -- select 'CRANTEE_MUST_E', 'TABPR', '', '', '', '' from SYSDDUMMY1
  ),
  -- 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
SYSDDUMMY1 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 <> '' )

```



SQL results report on l_gengrant

```

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. SQL results

SN	A list of generated SQL statements on GRANTS modifications reflected TABLES_GRANTS_MUST_HAVE requirements. A processing time stamp: 2025-09-15 00:06:09
1	GRANT SELECT ON TABLE TABPR1T_A TO CRANTEE_MUST_A;
2	GRANT SELECT ON TABLE TABPR1T_A TO CRANTEE_MUST_B;
3	GRANT SELECT ON TABLE TABPR1T_B TO CRANTEE_MUST_B;
4	GRANT SELECT ON TABLE TABPRADMT TO CRANTEE_MUST_B;
5	GRANT SELECT ON TABLE TABPRMT_A TO CRANTEE_MUST_B;
6	GRANT SELECT ON TABLE TABPR_T_A TO CRANTEE_MUST_B;
7	GRANT SELECT ON TABLE TABPR_T_B TO CRANTEE_MUST_A;
8	GRANT SELECT ON TABLE TABPR_T_B TO CRANTEE_MUST_B;
9	GRANT SELECT ON TABLE TABPR_T_C TO CRANTEE_MUST_B;
10	GRANT SELECT ON TABLE TABPR_T_D TO CRANTEE_MUST_A;
11	GRANT SELECT ON TABLE TABPR_T_D TO CRANTEE_MUST_B;
12	GRANT SELECT ON TABLE TABPR_T_E TO CRANTEE_MUST_A;
13	GRANT SELECT ON TABLE TABPR_T_F TO CRANTEE_MUST_A;
14	GRANT SELECT ON TABLE TABPR_T_F TO CRANTEE_MUST_B;
15	GRANT SELECT ON TABLE TABPR_T_J TO CRANTEE_MUST_A;
16	GRANT SELECT ON TABLE TABPR_T_J TO CRANTEE_MUST_B;
17	GRANT SELECT ON TABLE TABPR_T_K TO CRANTEE_MUST_A;
18	GRANT SELECT ON TABLE TABPR_T_K TO CRANTEE_MUST_B;
19	GRANT SELECT ON TABLE TABPR_T_L TO CRANTEE_MUST_A;
20	GRANT SELECT ON TABLE TABPR_T_L TO CRANTEE_MUST_B;
21	GRANT SELECT ON TABLE TABPR_T_M TO CRANTEE_MUST_A;
22	GRANT SELECT ON TABLE TABPR_T_M TO CRANTEE_MUST_B;
23	GRANT SELECT, DELETE, UPDATE ON TABLE TABPRMT_A TO CRANTEE_MUST_A;
24	GRANT SELECT, INSERT, UPDATE ON TABLE TABPRADMT TO CRANTEE_MUST_A;
25	REVOKE DELETE, INSERT, UPDATE ON TABLE TABPR1T_B from CRANTEE_MUST_A;
26	REVOKE DELETE, INSERT, UPDATE ON TABLE TABPR_T_A from CRANTEE_MUST_A;
27	REVOKE INSERT ON TABLE TABPR_T_E from CRANTEE_MUST_B;
28	REVOKE INSERT, UPDATE ON TABLE TABPR_T_C from CRANTEE_MUST_A;