Oracle Data Redaction

Oracle Database 12c and 11g (patch set: 11.2.0.4)

Who Am I

Ivica Arsov

Certifications:

- Oracle Database 11g Administrator Certified Master
- Oracle Certified Expert, Oracle Exadata X3 and X4 Administrator
- Oracle Certified Expert, Oracle Real Application Clusters 11g and Grid Infrastructure Administrator
- Oracle Database 11g Administrator Certified Professional

Blog: http://iarsov.com

Social media:

LinkedIn: https://www.linkedin.com/in/iarsov

Twitter: @IvicaArsov

Agenda

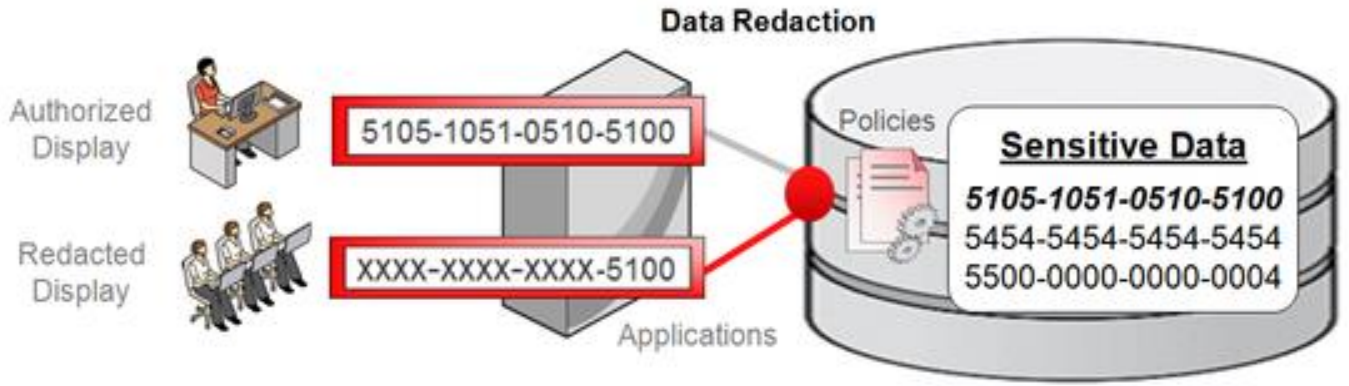
- Introduction to Oracle Data Redaction
- Data Redaction methods
- Caveats
- Virtual Columns - *warning* ⚠

- Available from Oracle Database 12c

  also available for 11g Release 2 (patch set 11.2.0.4)

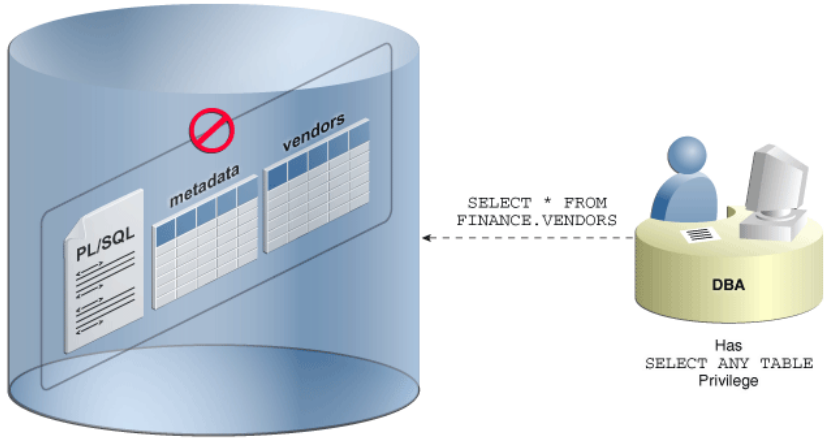
- Data is modified at query-execution time

- Not designed to prevent data exposure


Data Redaction
Authorized Display
Redacted Display
5105-1051-0510-5100
XXXX-XXXX-XXXX-5100
Applications
Policies
Sensitive Data
5105-1051-0510-5100
5454-5454-5454-5454
5500-0000-0000-0004

Other security options

- Database Vault
  - Realms (protection zone)
  - Command rules

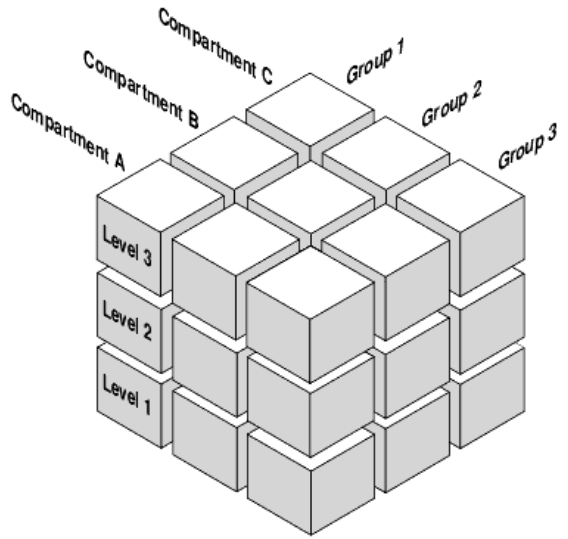


- Oracle Label Security
  - User and Data Labels (levels, compartments, groups)

- Virtual Private Database
  - Policies (similar to Data Redaction)

Other security options

- Database Vault
  - Realms (protection zone)
  - Command rules



- Oracle Label Security
  - User and Data Labels (levels, compartments, groups)

- Virtual Private Database
  - Policies (similar to Data Redaction)

Other security options

- Database Vault
  - Realms (protection zone)
  - Command rules
- Oracle Label Security
  - User and Data Labels (levels, compartments, groups)
- Virtual Private Database
  - Policies are used to modify WHERE clause

Data Redaction vs Data Masking

Data Redaction != Data Masking

With Data Masking:

- Actual data is modified
- Suitable for non-production environments
- It requires analysis to identify sensitive data

How does Data Redaction works?

Policies are defined to determine:

- What to redact
- How to redact
- When to redact

How do we manage policies ?

Interface to Data Redaction: DBMS_REDACT

ADD_POLICY	Defines a Data Redaction policy for a table or view
ALTER_POLICY	Alters a Data Redaction policy for a table or view
DISABLE_POLICY	Disables a Data Redaction policy
DROP_POLICY	Drops a Data Redaction policy
ENABLE_POLICY	Enables a Data Redaction policy
UPDATE_FULL_REDACTION_VALUES	Modifies the default displayed values for a Data Redaction policy for full redaction

```
DBMS_REDACT.ADD_POLICY (

  object_schema         => 'HR',
  object_name           => 'EMPLOYEES',
  policy_name           => 'SALARY_FULL_REDACT',
  column_name           => 'SALARY',
  function_type         => DBMS_REDACT.FULL,
  function_parameters   => NULL,
  expression            => 'SYS_CONTEXT(''USERENV'',''CURRENT_USER'') = ''ORACLE''',
  enable                => TRUE,
  policy_description    => 'Policy for salary redaction in Employees table'

);
```

```
column_name          => 'SALARY',
function_type        => DBMS_REDACT.FULL,
function_parameters  => NULL,
expression           => 'SYS_CONTEXT(''USERENV'',''CURRENT_USER'') = ''ORACLE''',
```

- Some functions that can be used:

SYS_CONTEXT

V, NV

OLS_LABEL_DOMINATES

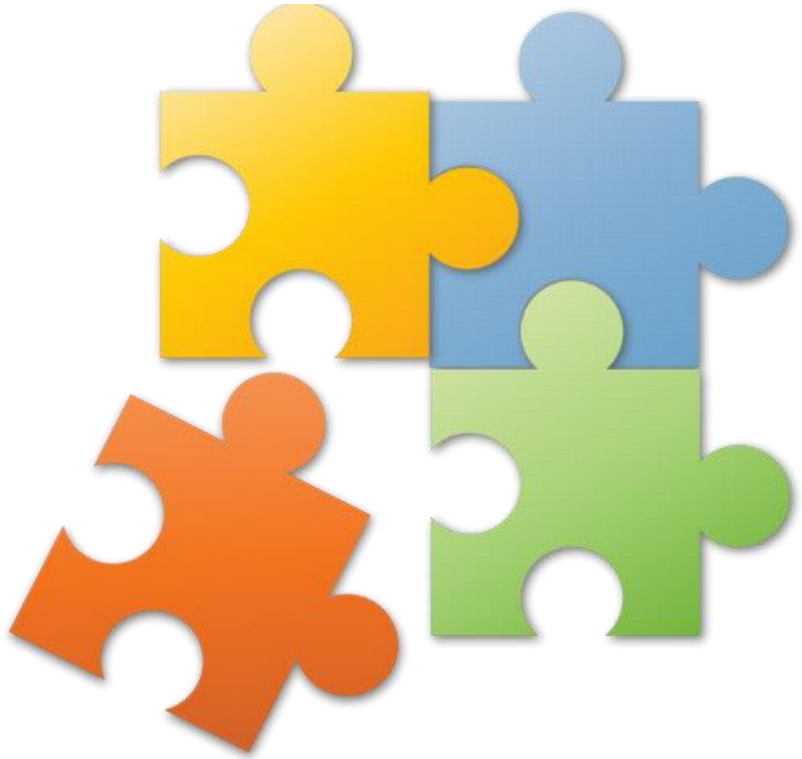
Conditions: =, != , >, <, >=, <=

- User defined functions are not permitted

Data Redaction Methods

Methods for Data Redaction

- Full
- Random
- Partial
- Regular expression
- None



- Full
- Random
- Partial
- Regular expression
- None

- Whole column value is redacted
- Different default values for different data types
  - Character data types: single space
  - Number data types: 0
  - Date data types: 01.01.2001

Determine current default values from dictionary view
`REDACTION_VALUES_FOR_TYPE_FULL`

```
DBMS_REDACT.ADD_POLICY (

  object_schema         => 'HR',
  object_name           => 'EMP',
  policy_name           => 'SSN_FULL_REDACT',
  column_name           => 'SSN',
  function_type         => DBMS_REDACT.FULL,
  function_parameters   => NULL,
  expression            => 'SYS_CONTEXT(''USERENV'',''CURRENT_USER'') = ''ORACLE''',
  enable                => TRUE,
  policy_description    => 'Policy for salary redaction in Employees table'

);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SSN
100	Steven	King	
101	Neena	Kochhar	
102	Lex	De Haan	
103	Alexander	Hunold	
104	Bruce	Ernst	
105	David	Austin	
106	Valli	Pataballa	
107	Diana	Lorentz	
108	Nancy	Greenberg	
109	Daniel	Faviet	

10 rows selected

- Base tables for default values

  Table: `radm_fptm$`

  LOBs are stored in separate table: `radm_fptm_lob$`

- Default values can be changed

  ```
  DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES
  ```

- Database instance must be restarted

Full Redaction - change default values

How to change default values:

1. Login to database with execute privilege on `DBMS_REDACT`
2. Check the default value you want to change
3. Set new default value with `DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES`
4. Restart the database instance

- Full
- Random
- Partial
- Regular expression
- None

- Column value is entirely changed
- Random value is generated each time redacted column is accessed

- Character data types:

CHAR

Character set remains same

Byte length is same as real column definition

VARCHAR2

Character set remains same

Data is limited to real (actual) data length

Number data types

Random non-negative number is generated

Precision is preserved

```
DBMS_REDACT.ADD_POLICY (

  object_schema         => 'HR',
  object_name           => 'EMP',
  policy_name           => 'SSN_RANDOM_REDACT',
  column_name           => 'SSN',
  function_type         => DBMS_REDACT.RANDOM,
  function_parameters   => NULL,
  expression            => 'SYS_CONTEXT(''USERENV'',''CURRENT_USER'') = ''ORACLE''',
  enable                => TRUE,
  policy_description    => 'Policy for salary redaction in Employees table'

);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SSN
100	Steven	King	,]NQ-o<Q4eV
101	Neena	Kochhar	5fFE,{X$=nN
102	Lex	De Haan	(&]We{?u0.e
103	Alexander	Hunold	2?]FG0<s:Ge
104	Bruce	Ernst	~iN,:h]z'qV
105	David	Austin	~QeM\q4\'Ym
106	Valli	Pataballa	y%?2#|Y""-G
107	Diana	Lorentz	]E4#;TF=eM<
108	Nancy	Greenberg	^PJ.3EsgfXR
109	Daniel	Faviet	#KJRd!BV+SR

10 rows selected

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SSN
100	Steven	King	++K$Z>1A33S
101	Neena	Kochhar	+QKsGKLR3YS
102	Lex	De Haan	XV}:g|\u`^&
103	Alexander	Hunold	-%B5(5 .5-J
104	Bruce	Ernst	FCGyK|Z2NKO
105	David	Austin	B+.{c>^JJ36
106	Valli	Pataballa	6sfNaJN/>{n
107	Diana	Lorentz	V0LAhvEF^8T
108	Nancy	Greenberg	"MUHF~1<*U]
109	Daniel	Faviet	Bb\,B$i(Jj

10 rows selected

- Full
- Random
- Partial
- Regular expression
- None

Character data types

- String must be fixed length
- Masking format is explicitly set by the user

Parameters

- Input format
- Output format
- Mask character
- Starting digit position
- Ending digit position

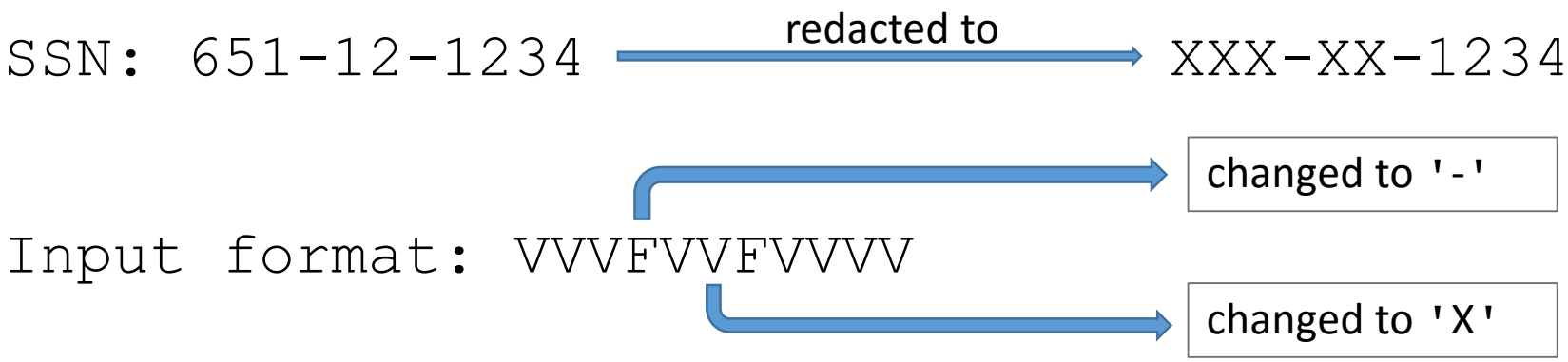
```
DBMS_REDACT.ADD_POLICY(
   object_schema        => 'HR',
   object_name          => 'EMPLOYEES',
   column_name          => 'SSN',
   policy_name          => 'SSN_PARTIAL_REDACT',
   function_type        => DBMS_REDACT.PARTIAL,
   function_parameters  => 'VVVFVVFVVVV,VVV-VV-VVVV,X,1,5',
   expression           => '1=1',
   policy_description   => 'Partial redact for Employee social security number');
```

Input / Output format

`V` - for potential characters to be redacted

`F` - for characters to be formatted using format character

```
function_parameters => 'VVVFVVFVVVV,VVV-VV-VVVV,X,1,5'
```



- Full
- Random
- Partial
- Regular expression
- None

Regular Expression Redaction

- Redaction based on patterns
- Full redaction can take place if:
  - Pattern fails to match
  - If no replacement occurs during regular expression replacement operation

Regular Expression Redaction

Predefined patterns

REGEXP_PATTERN

```
DBMS_REDACT.RE_PATTERN_ANY_DIGIT
DBMS_REDACT.RE_PATTERN_CC_L6_T4
DBMS_REDACT.RE_PATTERN_US_PHONE
DBMS_REDACT.RE_PATTERN_EMAIL_ADDRESS
RE_REDACT_EMAIL_NAME
RE_REDACT_EMAIL_DOMAIN
RE_REDACT_EMAIL_ENTIRE
DBMS_REDACT.RE_PATTERN_IP_ADDRESS
```

REGEXP_REPLACE_STRING

```
DBMS_REDACT.RE_REDACT_WITH_SINGLE_X
DBMS_REDACT.RE_REDACT_WITH_SINGLE_1
DBMS_REDACT.RE_REDACT_CC_MIDDLE_DIGITS
DBMS_REDACT.RE_REDACT_PHONE_L7
DBMS_REDACT.RE_REDACT_EMAIL_NAME
DBMS_REDACT.RE_REDACT_EMAIL_DOMAIN
DBMS_REDACT.RE_REDACT_IP_L3
```

Regular Expression Redaction

Input parameters:

`regexp_pattern` - search pattern

`regexp_replace_string` - replacement value

`regexp_position` - from where to start the search (defaults to 1)

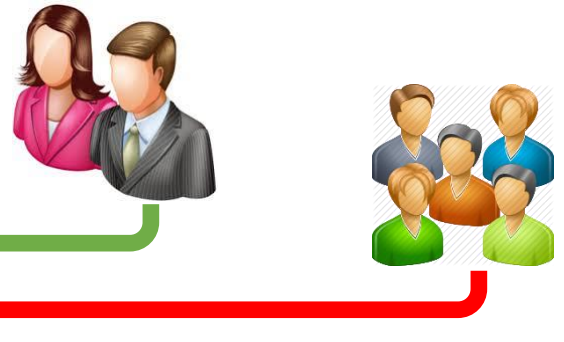
`regexp_occurences` - whether to replace all, first or nth occurrence

`regexp_match_parameter` - changes matching behavior

- Full
- Random
- Partial
- Regular expression
- None

```
SELECT   object_name, object_type
FROM     user_objects
WHERE    object_name in ('EMP_T','EMP_V');
```

```
OBJECT_NAME                    OBJECT_TYPE
------------------------------ ------------------
EMP_V                          VIEW
EMP_T                          TABLE
```



```
DBMS_REDACT.ADD_POLICY (
  object_schema => 'HR',
  object_name   => 'EMP_T',
  policy_name   => 'EMP_SAL_FULL_REDACT',
  column_name   => 'SALARY',
  function_type => DBMS_REDACT.FULL,
  expression    => '1=1');
```

```
DBMS_REDACT.ADD_POLICY (
    object_schema => 'HR',
    object_name   => 'EMP_V',
    policy_name   => 'EMP_VIEW_NOREDACT',
    column_name   => 'SALARY',
    function_type => DBMS_REDACT.NONE,
    expression    => '1=1');
```

```
SELECT object_name, policy_name, enable FROM REDACTION_POLICIES WHERE object_owner = 'HR';

OBJECT_NAME    POLICY_NAME                ENABLE
-------------  -------------------------  -----------
EMP_T          EMP_SAL_FULL_REDACT        YES
EMP_V          EMP_VIEW_NOREDACT          YES
```

```
SELECT first_name, last_name, salary FROM EMP_T fetch first 3 rows only;

FIRST_NAME    LAST_NAME      SALARY
------------  -------------  ----------
Steven        King           0
Neena         Kochhar        0
Lex           De Haan        0
```

FULL redaction on salary column

```
SELECT first_name, last_name, salary FROM EMP_V fetch first 3 rows only;

FIRST_NAME    LAST_NAME      SALARY
------------  -------------  ----------
Steven        King           24000
Neena         Kochhar        17000
Lex           De Haan        17000
```

NONE redaction policy defined

Data Redaction - explain plan

- There is no change to explain plan
- No information for end user that redaction took place

```
SQL> set autotrace trace exp
SQL> select first_name, last_name, salary from emp;

Execution Plan
----------------------------------------------------------
Plan hash value: 3956160932

--------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |   107 |  2033 |     3   (0)| 00:00:01 |
|   1 |  TABLE ACCESS FULL| EMP  |   107 |  2033 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------
```

```
select first_name, salary from emp;
```

FULL redaction
on salary column

Optimizer trace:
==========

```
PARSING IN CURSOR #18446604434619702408 len=57 tim=58985251144 sqlid='7b50t3fpq2fng'
select pname, pexpr, enable_flag from radm$ where obj#=:1
END OF STMT
PARSE #18446604434619702408:c=19340,e=19341,p=0,cr=71, mis=1,dep=1,og=4,plh=0,tim=58985251138
BINDS #18446604434619702408:
 Bind#0
  oacdty=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
  oacflg=00 fl2=1000001 frm=00 csi=00 siz=24 off=0
  kxsbbbfp=ffff80ffbdb29630  bln=22  avl=04  flg=05
  value=92715
EXEC #18446604434619702408:c=2366,e=2367,p=0,cr=12, mis=1, og=4,plh=1091136192,tim=58985253638
```

Optimizer trace:
==========

```
PARSING IN CURSOR #18446604434620453248 len=401 dep=1 uid=0 oct=3 lid=0 tim=58985264150 hv=3348710374 ad='16f570690'
sqlid='fpm1tjb3tkhz6'
select mfunc, mparams, intcol#, regexp_pattern, regexp_replace_string, regexp_position, regexp_occurrence,
regexp_match_parameter, mp_iformat_start_byte, mp_iformat_end_byte, mp_oformat_start_byte, mp_oformat_end_byte,
mp_maskchar_start_byte, mp_maskchar_end_byte, mp_maskfrom, mp_maskto, mp_datmask_Mo, mp_datmask_D, mp_datmask_Y,
mp_datmask_H, mp_datmask_Mi, mp_datmask_S
from radm_mc$ where obj#=:1
END OF STMT
PARSE #18446604434620453248:c=10286,e=10286,p=0,cr=71,cu=0,mis=1,r=0,dep=1,og=4,plh=0,tim=58985264145
BINDS #18446604434620453248:
 Bind#0
  oacdty=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
  oacflg=00 fl2=1000001 frm=00 csi=00 siz=24 off=0
  kxsbbbfp=ffff80ffbdb3ed58  bln=22  avl=04  flg=05
  value=92715
EXEC #18446604434620453248:c=11911,e=11910,p=0,cr=69,cu=0,mis=1,r=0,dep=1,og=4,plh=3522975176,tim=58985276263
FETCH #18446604434620453248:c=29,e=29,p=0,cr=2,cu=0,mis=0,r=1,dep=1,og=4,plh=3522975176,tim=58985276363
```

Optimizer trace:

```
=====================

.......
Bind#0
  oacdty=02 mxl=22(22) mxlc=00 mal=00 scl=00 pre=00
  oacflg=00 fl2=1000001 frm=00 csi=00 siz=24 off=0
  kxsbbbfp=ffff80ffbdb3ed58  bln=22  avl=04  flg=05
  value=92715
.......

SELECT pname, pexpr, enable_flag FROM RADM$ where obj# = 92715;

PNAME        PEXPR      ENABLE_FLAG
-----------  ---------  ------------
emp_redact   1=1        1


SELECT object_id, owner, object_name, object_type FROM dba_objects WHERE object_id = 92715;

OBJECT_ID    OWNER     OBJECT_NAME   OBJECT_TYPE
-----------  --------  ------------  -------------
92715        HR        EMP           TABLE
```

Bypassing Data Redaction Policies

- EXEMPT REDACTION POLICY privilege

- EXEMPT DDL REDACTION POLICY privilege
- EXEMPT DML REDACTION POLICY privilege

- SYS and SYSTEM by default have EXEMPT REDACTION POLICY privilege

Considerations

Recycle Bin

You might see something like
`BIN$C1uN3icECP3gVAgAJ3PSGQ==$0`
under `OBJECT_NAME` in `REDACTION_POLICIES`

```
show parameter recyclebin

NAME                 TYPE         VALUE
-------------------  -----------  ------------
recyclebin           string       on
```

CTAS and Data Redaction

DDL statements not allowed when redacted objects are involved:

- **CREATE TABLE AS SELECT** (CTAS)
- **INSERT AS SELECT**

```
SQL> create table emp1 as select * from emp;
create table emp1 as select * from emp
                             *
ERROR at line 1:
ORA-28081: Insufficient privileges - the command references a redacted
object.
```

CTAS and Data Redaction

```
SQL> !oerr ora 28081
28081, 00000, "Insufficient privileges - the command references a redacted object."
// *Cause: The command referenced a redacted column in an
// object protected by a data redaction policy.
// *Action: If possible, modify the command to avoid referencing any
// redacted columns.  Otherwise, drop the data redaction policies that
// protect the referenced tables and views, or ensure that the user issuing
// the command has the EXEMPT REDACTION POLICY system privilege, then
// retry the operation.  The EXEMPT REDACTION POLICY system privilege
// is required for creating or refreshing a materialized view when the
// materialized view is based on an object protected by a data redaction
// policy.  The EXEMPT REDACTION POLICY system privilege is required for
// performing a data pump schema-level export including any object
// protected by a data redaction policy.  All data redaction policies are
// listed in the REDACTION_COLUMNS catalog view.
```

CTAS and Data Redaction

```
SQL> conn sys/oracle@pdb1 as sysdba
Connected.
SQL>
SQL> grant exempt redaction policy to hr;

Grant succeeded.

SQL> conn hr/hr@pdb1
Connected.
SQL>
SQL> create table emp1 as select * from emp;

Table created.
```

Data Redaction and GROUP BY

- Redacted columns are not allowed to be specified in SQL expression while used in GROUP BY clause

Error **ORA-00979: not a GROUP BY expression** is raised

```
SQL> select salary from hr.employees group by (salary+0);
select salary from hr.employees group by (salary+0)
       *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```

Data Redaction and Data Pump

- DATAPUMP_EXP_FULL_DATABASE role includes EXEMPT_REDACTION_POLICY system privilege

- Data Pump export cannot be performed on redacted objects without EXEMPT_REDACTION_POLICY system privilege

```
Starting "HR"."SYS_EXPORT_SCHEMA_01":  hr/********@pdb1 directory=dbexp schemas=hr access_method=direct_path
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
ORA-31696: unable to export/import TABLE_DATA:"HR"."EMPLOYEES" using client specified DIRECT_PATH method
Total estimation using BLOCKS method: 33.5 MB
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
```

Data Pump Export

Error: ORA-31696: unable to export/import TABLE_DATA:"HR"."EMPLOYEES" using client specified DIRECT_PATH method

Not very clear what the problem is

How to find the problem ?

Re-execute data pump export/import without ACCESS_METHOD parameter or set it to automatic (default) or external_table

```
. . exported "HR"."COUNTRIES"                            6.460 KB      25 rows
. . exported "HR"."DEPARTMENTS"                          7.125 KB      27 rows
ORA-31693: Table data object "HR"."EMPLOYEES" failed to load/unload and is being skipped due to error:
ORA-28081: Insufficient privileges - the command references a redacted object.
. . exported "HR"."JOBS"                                 7.109 KB      19 rows
. . exported "HR"."JOB_HISTORY"                          7.195 KB      10 rows
```

Data Pump Import

Data Redaction Security Considerations

Known limitations:

- Not meant to prevent from privileged users who execute ad hoc queries
- Sensitive data can be revealed by the method of inference
- Not enforced for users logged as SYSDBA administrative privilege

```
declare
  n number default 1;
  v_tmp number default 1;
  v_salary number default 0;
begin

loop
     begin

          select src.num into v_tmp
          from   employees,
                 (select  (rownum-1) num
                  from    dual
                  connect by rownum <= 10) src

          where lower(email) = lower('sking')
                and to_number(substr(salary,n,1)) = src.num;

          v_salary := v_salary || v_tmp;

     exception
        when no_data_found then
           goto gexit;
     end;

n := n + 1;

end loop;
...
```

```
declare
  n number default 1;
  v_tmp number default 1;
  v_salary number default 0;
begin

loop
     begin

          select src.num into v_tmp
          from   employees,
                 (select  (rownum-1) num
                  from    dual
                  connect by rownum <= 10) src

          where lower(email) = lower('sking')
                and to_number(substr(salary,n,1)) = src.num;

          v_salary := v_salary || v_tmp;

     exception
        when no_data_found then
           goto gexit;
     end;

n := n + 1;

end loop;
...
```

```
declare
  n number default 1;
  v_tmp number default 1;
  v_salary number default 0;
begin

loop
     begin

          select src.num into v_tmp
          from   employees,
                 (select  (rownum-1) num
                  from    dual
                  connect by rownum <= 10) src

          where lower(email) = lower('sking')
                and to_number(substr(salary,n,1)) = src.num;

          v_salary := v_salary || v_tmp;

     exception
        when no_data_found then
           goto gexit;
     end;

n := n + 1;

end loop;
...
```

```
Name: Steven, Surname: King, redacted salary: 8758
Name: Steven, Surname: King, salary: 24000

Name: Steven, Surname: King, redacted salary: 10859
Name: Steven, Surname: King, salary: 24000

Name: Steven, Surname: King, redacted salary: 9253
Name: Steven, Surname: King, salary: 24000

Name: Steven, Surname: King, redacted salary: 19116
Name: Steven, Surname: King, salary: 24000

Name: Steven, Surname: King, redacted salary: 18614
Name: Steven, Surname: King, salary: 24000
```

Data Redaction and Function Based Indexes

- Function Based Indexes will break Data Redaction

Lets assume that:

1. We have `SALART_FULL_REDACT` policy
2. We have user defined dummy function `DUMMY_F` used for index on `SALARY` column

```
create function dummy_f(p_val number)
return number deterministic
is
begin
  return p_val;
end dummy_f;
```

```
dbms_redact.add_policy
(object_schema  => 'HR',
 object_name    => 'EMP_FBI',
 policy_name    => 'SALARY_FULL_REDACT',
 function_type  => DBMS_REDACT.FULL,
 column_name    => 'SALARY',
 expression     => SYS_CONTEXT(''USERENV'',''CURRENT_USER'') =''ORACLE''
);
```

Data Redaction and Function Based Indexes

```
SQL> conn oracle/oracle@pdb1
Connected.

SQL> select first_name, last_name, hr.dummy_f(salary) SALARY
     from emp_fbi where email = 'SKING'

FIRST_NAME    LAST_NAME   SALARY
------------  ----------  ---------
Steven        King        0
```

```
SQL> conn hr/hr@pdb1
Connected.

SQL> create index emp_fbi_sal_ix
on emp_fbi(dummy_f(salary));

Index created.
```

```
SQL> conn oracle/oracle@pdb1
Connected.

SQL> select first_name, last_name, hr.dummy_f(salary) SAL
     from emp_fbi where email = 'SKING'

FIRST_NAME    LAST_NAME   SALARY
------------  ----------  --------
Steven        King        24000
```

Data Redaction and User Defined Indexes

```
SQL> conn hr/hr@pdb1
Connected.
SQL>
SQL> create index ss_ix on
     employees(salary+0);
Index created.
```

Without index:

```
select first_name, last_name, salary
from   hr.employees
where email = 'SKING';

FIRST_NAME   LAST_NAME   SALARY
-----------  ----------  ----------
Steven       King        0
```

With index defined:

```
select first_name, last_name, (salary + 0) as salary
from   hr.employees
where email = 'SKING';

FIRST_NAME   LAST_NAME   SALARY
-----------  ----------  ----------
Steven       King        24000
```

Data Redaction and Virtual Columns

ORA-28083: A redacted column was referenced in a virtual column expression.

Cause: This redacted column was referenced in a virtual column expression.

```
SQL> alter table emp_fbi add salary1 as (salary+0);
Table altered.

SQL> select first_name, salary, salary1 from emp_fbi where email = 'SKING';

FIRST_NAME                SALARY    SALARY1
------------------------- ---------- ----------
Steven                             0      24000
```

Data Redaction from Cloud Control

From Database Home Page: **Security** -> **Data Redaction**