有客户问，Oracle数据库里面能不能实现这么一个功能：

符合条件的用户可以看到表里面的所有数据，而另外一些人（业务权限低的用户）也需要查询表数据，但是某些敏感数据全部用星号替换。

redact 编写、修订、改写

》Oracle数据编写介绍

12c及以上版本，Oracle提供了数据编写功能。Oracle数据编写提供了重新编写数据的能力，实时隐藏、掩盖敏感数据。

同时也能够使公司遵循法规和行业规则

针对列数据编写的方式

\*Full redaction：列数据内容全部编写，编写后的值依赖于数据类型。如：NUMBER》0，字符》空格

\*Partial redaction：列数据内容部分编写，如：将社保卡账号前8个号码编写为\*，只显示后4位。\*\*\*\*\*\*\*\*4320

\*Regular expressions：类似部分数据编写，只适用于可变字符类型中的部分编写

\*Random redaction：列数据随机编写，每次返回的值都是随机的。依赖于数据类型不同

\*No redaction：列数据不编写，不影响表的查询结果，可用于测试编写功能

数据编写只在运行时间生效。不影响现有数据,不对现有数据做更改。如：查询数据期间。

datafile

table

buffer

数据编写特别适合读操作，针对更改操作要特别小心，编写后的数据会写入数据库中

UPDATE hr.employees

SET

    salary = (

        SELECT

            salary

        FROM

            hr.employees AS OF TIMESTAMP to\_timestamp('2018-05-09 10:47:56','YYYY-MM-DD HH:MI:SS')

        WHERE

            employee\_id = 100

    )

WHERE

    employee\_id = 100;

适用于：

1 数据编写对性能影响很低，可适用于多个场景

主要适用于应用程序，分析工具，监视工具等对数据库敏感数据的查询。

2 也可用于对不同的应用程序用户返回不同的结果

===================================

》Oracle数据编写特性和功能

调用PLSQL程序包DBMS\_REDACT实现不同的编写功能

1 Full redaction

列的整个内容被编写，这是默认的编写方式

function\_type    => DBMS\_REDACT.FULL

Character data types》空格

Number data types 》 0

Date-time data types》01-JAN-01

用DBMS\_REDACT.UPDATE\_FULL\_REDACTION\_VALUES将改写值自定义

2 Partial redaction

对显示出来的数据做部分编写，可以指定别写的起始位置。常用于 银行卡号和身份证号

注意：Partial redaction需要列宽度固定，对于可变字符长度列，需要使用Regular expressions

function\_type    => DBMS\_REDACT.PARTIAL

function\_parameters

Character data types ：987-65-4320

XXX-XX-4320    function\_parameters => DBMS\_REDACT.REDACT\_US\_SSN\_F5,

\*\*\*-\*\*-4320    function\_parameters => 'VVVFVVFVVVV,VVV-VV-VVVV,\*,1,5',

Number data types: 987654328

XXXXX4328    function\_parameters => DBMS\_REDACT.REDACT\_NUM\_US\_SSN\_F5,

999994328    function\_parameters => '9,1,5',

Date-time data types：29-AUG-11 10.20.50.000000 AM

02-AUG-11 10.20.50.000000 AM   function\_parameters  =>  'Md02YHMS',

29-DEC-11 10.20.50.000000 AM   function\_parameters  =>  'm12DYHMS',

3 Regular expressions

用正则表达式查找部分特定数据（search pattern）用于编写    --可变长度中的部分数据

如：xianshijie@highgo.com   改写为xsj@higogh.com

4 Random redaction

整个列数据编写成随机值，每次产生不同的随机值。常用于无规律表达，不体现数据编写格式情况

特别适合 number and datetime

function\_type    => DBMS\_REDACT.RANDOM

Character data types：

字符随机组合（如：HTU[G{\pjkEWcK）

    CHAR data type: 编写输出 字符集与列字符集一致，随机字符宽度与列定义相同

    VARCHAR2 data type:编写输出 字符集与列字符集一致，随机字符宽度与实际数据宽度相同

Number data types:

编写后为完全随机非负数，长度不超过实际值。不是对原数字编辑

123456》1/234/888888

唯一的例外是：实际值为-1~9之间的整数，用10编写

1》10，8》10

Date-time data types

编写后时间总是不同于当前时间

5 No redaction

创建不执行改写的 数据改写策略，不影响数据。

》配置数据编写策略

编写策略定义了编写的条件

需要有以下权限

EXECUTE privilege on the DBMS\_REDACT PL/SQL package

需要确定

what    ：确定遍写类型 -full，Partial

how    ：确定编写发生的方式 -expression

when    ：确定编写发生的时间  -在语句运行时间发生

使用DBMS\_REDACT.ADD\_POLICY程序包

DBMS\_REDACT.ADD\_POLICY (

object\_schema               IN VARCHAR2 := NULL,

object\_name                 IN VARCHAR2 := NULL,

policy\_name                 IN VARCHAR2,

policy\_description          IN VARCHAR2 := NULL,

column\_name                 IN VARCHAR2 := NULL,

column\_description          IN VARCHAR2 := NULL,

function\_type               IN BINARY\_INTEGER := DBMS\_REDACT.FULL,

function\_parameters         IN VARCHAR2 := NULL,

expression                  IN VARCHAR2,

enable                      IN BOOLEAN := TRUE,

regexp\_pattern              IN VARCHAR2 := NULL,

regexp\_replace\_string       IN VARCHAR2 := NULL,

regexp\_position             IN BINARY\_INTEGER :=1,

regexp\_occurrence           IN BINARY\_INTEGER :=0,

regexp\_match\_parameter      IN VARCHAR2 := NULL);

expression: Specifies a Boolean SQL expression to determine how the policy is applied. Redaction takes place only if this policy expression evaluates to TRUE.

有关编写策略中的expression

expression基于两个函数

SYS\_CONTEXT             --基于用户环境应用改写

SYS\_SESSION\_ROLES   --基于用户权限应用改写

expression  => 'SYS\_CONTEXT(''USERENV'',''SESSION\_USER'') = ''HR'''

expression  => 'SYS\_CONTEXT(''SYS\_SESSION\_ROLES'',''ADMIN'') = ''FALSE'''

expression  => '1=1'  --表示没有过滤条件，适用于所有用户（除了sys及拥有EXEMPT REDACTION POLICY权限的用户）

1 Full redaction

创建策略

BEGIN

DBMS\_REDACT.ADD\_POLICY(

   object\_schema    => 'hr',

   object\_name      => 'employees',

   column\_name      => 'commission\_pct',

   policy\_name      => 'redact\_com\_pct',

   function\_type    => DBMS\_REDACT.FULL,

   expression       => '1=1');

END;

/

更改默认显示的编写值

select \* from REDACTION\_VALUES\_FOR\_TYPE\_FULL

Non-LOB数据类型：

EXEC DBMS\_REDACT.UPDATE\_FULL\_REDACTION\_VALUES (number\_val => 8);  --sys

重启生效

SHUTDOWN IMMEDIATE

STARTUP

LOG数据类型：

更改数据字典表RADM\_FPTM\_LOB$

删除策略

BEGIN

  DBMS\_REDACT.DROP\_POLICY (

    object\_schema  => 'hr',

    object\_name    => 'employees',

    policy\_name    => 'redact\_phone\_num');

END;

/

禁用策略

BEGIN

  DBMS\_REDACT.DISABLE\_POLICY (

    object\_schema  => 'hr',

    object\_name    => 'employees',

    policy\_name    => 'redact\_phone\_num');

END;

/

BEGIN

  DBMS\_REDACT.ENABLE\_POLICY (

    object\_schema  => 'hr',

    object\_name    => 'employees',

    policy\_name    => 'redact\_phone\_num');

END;

/

2 Partial redaction

使用固定字符快捷方式

BEGIN

DBMS\_REDACT.ADD\_POLICY(

   object\_schema       => 'hr',

   object\_name         => 'employees',

   column\_name         => 'PHONE\_NUMBER',

   policy\_name         => 'redact\_phone\_num',

   function\_type       => DBMS\_REDACT.PARTIAL,

   function\_parameters => DBMS\_REDACT.REDACT\_US\_SSN\_F5,

   expression          => '1=1',

   policy\_description  => 'first 5 numbers of column,VARCHAR2 data type',

   column\_description  => 'phone number');

END;

/

function\_parameters => DBMS\_REDACT.REDACT\_US\_SSN\_F5

固定字符快捷方式：预定义方式，有很多

DBMS\_REDACT.REDACT\_US\_SSN\_F5

Redacts the first 5 numbers of Social Security numbers when the column is a VARCHAR2 data type. For example, the number 987-65-4320 becomes XXX-XX-4320.

DBMS\_REDACT.REDACT\_US\_SSN\_L4

Redacts the last 4 numbers of Social Security numbers when the column is a VARCHAR2 data type. For example, the number 987-65-4320 becomes 987-65-XXXX.

BEGIN

  DBMS\_REDACT.DROP\_POLICY (

    object\_schema  => 'hr',

    object\_name    => 'employees',

    policy\_name    => 'redact\_phone\_num');

END;

/

使用自定义

function\_parameters  => 'VVVFVVVFVVV,VVV-VVV-VVV,\*,1,6',

function\_parameters  => '9,1,5',

BEGIN

DBMS\_REDACT.ADD\_POLICY(

   object\_schema       => 'hr',

   object\_name         => 'employees',

   column\_name         => 'PHONE\_NUMBER',

   policy\_name         => 'redact\_phone\_n',

   function\_type       => DBMS\_REDACT.PARTIAL,

   function\_parameters  => 'VVVFVVVFVVV,VVV-VVV-VVV,\*,1,6',

   expression          => '1=1',

   policy\_description  => 'first 6 numbers of column,VARCHAR2 data type',

   column\_description  => 'phone number');

END;

/

改写多列！！！

BEGIN

DBMS\_REDACT.ALTER\_POLICY(

  object\_schema       => 'hr',

  object\_name         => 'employees',

  policy\_name         => 'redact\_phone\_n',

  action              => DBMS\_REDACT.ADD\_COLUMN,

  column\_name         => 'salary',

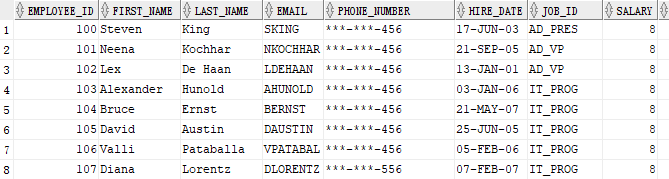
  function\_type       => DBMS\_REDACT.FULL,

  function\_parameters => '',

  expression  => 'SYS\_CONTEXT(''USERENV'',''SESSION\_USER'') = ''HR''');

END;

/



查看编写策略信息

REDACTION\_COLUMNS

REDACTION\_POLICIES

REDACTION\_VALUES\_FOR\_TYPE\_FULL