

深入了解Oracle动态数据脱敏

--- 数据安全实战演练系列(三)

公益讲座11: 00准时开始, 请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布, 已入群客户请勿重复入群!



20-21

数据库和云讲座群



甲骨文云技术公众号



B站专家系列课程



基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

400-699-8888

* 活动最终解释权归甲骨文公司所有

Oracle 动态数据脱敏

甲骨文技术公益课 - 数据库专场

2023 年 8 月 25 日 11:00

线上直播

Steven Li



主页 动态 投稿 245 合集和列表 15 收藏 0

TA的合集和视频列表 > 数据安全实战演练系列

2个视频 | 8-4更新



【Oracle 公益课堂】Oracle透明数据加密(TDE)

617

6-30



【Oracle 公益课堂】深入了解Oracle审计仓库和数据库防火墙

438

8-4



主页 动态 投稿 245 合集和列表 15 收藏 0 订阅 搜索视频、动态

TA的合集和视频列表 > 数据安全系列

5个视频 | 2-27更新

Oracle数据安全系列



【Oracle 数据安全系列】Oracle 数据安全
455 2020-6-13



【Oracle 数据安全系列】数据库如何被攻破，怎样防护
329 2022-9-16



【Oracle数据安全系列】Oracle 如何防范网络勒索软件
262 2022-1-21



【Oracle 数据安全系列】利用Exadata进行数据脱敏
363 2021-11-8



【Oracle 数据安全系列】Oracle审计仓库与数据库防火墙
503 2021-10-11

内容

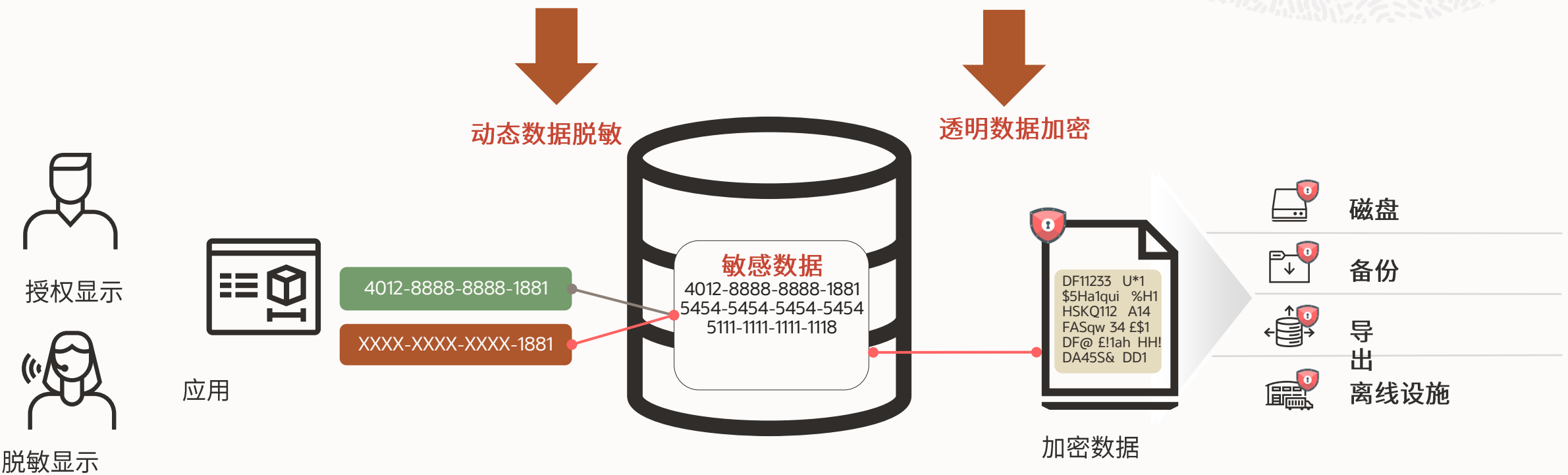
深入了解Oracle 动态数据脱敏(Data Redaction)

- 概述
- 动态数据脱敏策略
- 动手实践
- 常见问题
- 使用指南



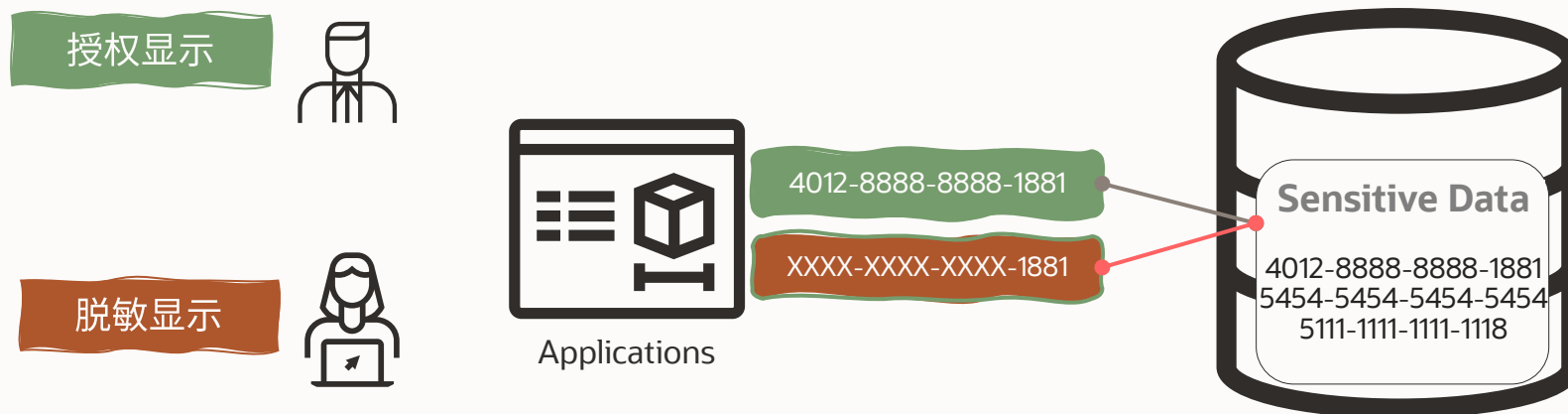
概述

高级安全选件



动态数据脱敏

动态数据脱敏在 Oracle Database 12c 中首次引入



- 动态数据脱敏可基于用户名称、IP、应用程序上下文和其他会话因素
- 通过PLSQL 或 OEM创建动态脱敏策略

动态数据脱敏vs. 静态数据脱敏

动态数据脱敏

- 旨在**实时**保护**实时生产**和**非实时**系统的敏感数据
- 数据在到达应用程序之前在数据库服务器上被屏蔽
- **未经授权的用户无法**通过直接数据库访问读取真实数据
- 对真实的数据不会做更改

静态数据脱敏

- 通常用于创建**开发**和**测试**环境，或向第三方分享数据
- 敏感数据被复制并被**改写**
- 处理过程：
将数据还原到一个与生产分离的数据库，
使用数据子集提取所需的数据量
应用脱敏规则，改写数据
把脱敏后的数据推到目标库

19

5

28

6

* Customer

Edward Larsos

?

* Name

Edward L. Larsos

?

Contact Type

Customer

?

Title

Architect

?

Company

My Company

?

Email

e.larsos.ll@my-mail.com

?

Phone

320-492-****

?

Cell Phone

356-566-****

?

Fax

255-556-****

?

Notes

?

SSN

469-48-****

?

明文显示
基本信息

脱敏显示
支付信息

Address / Country

Address

1850 Timberbreak Line

City

Maytonvilles

State

MN

Country

USA

Zip

55072

?

Credit Card Information

Type

CC-Visa

?

Number

*****7091

?

Exp. Date

03/2023

?

Cardholder Name

Ed Larsos

?

Billing Zipcode

55072

?

Billing Phone

320-492-****

?



动态数据脱敏策略

动态数据脱敏策略

脱敏策略

- 提供多种脱敏方案
- 在运行时进行动态脱敏，适用于数据不断变化的环境
- 可以在一个集中的位置创建脱敏策略，并轻松管理这些策略
- 基于SYS_CONTEXT()创建各种不同的条件

脱敏类型

- 脱敏表或视图中某个列上的全部敏感数据
- 脱敏一部分列中的数据
- 以随机值的方式脱敏列中数据
- 利用正则表达式寻找匹配的内容并替换

策略表达式

- 哪些用户能看到真实数据，哪些用户看到的是脱敏后的数据
- 是一个逻辑表达式，返回true或 false
- 出于安全的考虑，只能使用有限的函数和运算符，如 sys_context 函数，查看命名空间

规划Oracle动态数据脱敏策略

创建策略前的准备

- 确保创建者拥有足够的权限
 - ✓ 如果在创建者拥有的schema上，需要create table 或 create any table 权限，以及在DBMS_REDACT 包上的 Execute权限
 - ✓ 如果是另一个用户拥有的schema上， 需要create any table权限， 以及在DBMS_REDACT 包上的 Execute权限
- 确定需要脱敏的数据类型
- 确定脱敏类型
- 确定脱敏策略应用到哪些用户上
- 调用DBMS_REDACT.ADD_POLICY， 创建脱敏策略
- 调用DBMS_REDACT.ALTER_POLICY， 向脱敏策略中添加其他的列（每个表或视图只能定义一个动态数据脱敏策略）

```
BEGIN
  DBMS_REDACT.ADD_POLICY(
    object_schema => 'HR',
    object_name   => 'EMPLOYEES',
    column_name   => 'SALARY',
    column_description => 'employee salary and commission',
    policy_name   => 'redact_emp_sal_comm',
    policy_description => 'Partially redacts',
    function_type  => DBMS_REDACT.PARTIAL,
    function_parameters => '9,1,4',
    expression     => '1=1');
END;
/

BEGIN
  DBMS_REDACT.ALTER_POLICY(
    object_schema => 'HR',
    object_name   => 'EMPLOYEES',
    policy_name   => 'redact_emp_sal_comm',
    action        => DBMS_REDACT.ADD_COLUMN,
    column_name   => 'COMMISSION_PCT',
    function_type  => DBMS_REDACT.PARTIAL,
    function_parameters => '9,1,1',
    expression     => '1=1');
END;
/
```


定义策略所需的 DBMS_REDACT包

存储过程	描述
DBMS_REDACT.ADD_POLICY	在一个表或视图上添加数据脱敏策略
DBMS_REDACT.ALTER_POLICY	修改脱敏策略
DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL	在一个表或视图的列上，建立策略表达式
DBMS_REDACT.CREATE_POLICY_EXPRESSION	创建策略表达式
DBMS_REDACT.DISABLE_POLICY	禁用脱敏策略
DBMS_REDACT.DROP_POLICY	删除策略
DBMS_REDACT.DROP_POLICY_EXPRESSION	删除策略表达式
DBMS_REDACT.ENABLE_POLICY	激活脱敏策略
DBMS_REDACT.UPDATE_POLICY_EXPRESSION	更新策略表达式



OEM 管理动态数据脱敏

ORACLE Enterprise Manager Cloud Control 13c

企业E

cdb1 / PDB1

Oracle 数据库 性能 可用性 安全性 方案 管理

创建数据编写策略: policy1

* 方案EMPLOYEESEARCH_PROD

* 表/视图DEMO_HR_EMPLOYEES

* 策略名policy1

* 默认表达式1=1

对象列

查看 格式 添加 修改 删除 冻结 分离 换行

列	列数据类型	表达式名称	编写函数	函数属性
EMAIL	VARCHAR2		REGEX	([A-Z0-9_%+-]+)@([A-Z0-9-]+\.[A-Z]{2,4}),xxxx@1,1,i
SALARY	NUMBER		RANDOM	

说明

1. 通过选择数据库方案以及要编写的表或视图, 并为策略

2. 使用下面的列列表选取要编写的特定列并指定其编写

3. 复查并更新编写策略表达式。此表达式默认为 1=1 (T

有关编写策略表达式的帮助, 请单击铅笔图标以显示“策

情况下编写敏感数据的空白列表并在满足所指定的异常

添加

* 列EMAIL

列数据类型VARCHAR2

敏感列类型UNDEFINED

表达式名称默认表达式

编写格式Email Addresses

* 编写函数REGEX

基于正则表达式的编写。指定表示将编写的列数据的正则表达式。

函数属性

* 模式([A-Z0-9_%+-]+)@([A-Z0-9-]+\.[A-Z]{2,4})

* 替换字符串xxxx@1

* 位置1

* 出现1

匹配参数忽略大小写

确定 取消



OEM 管理动态数据脱敏 – 模版

1

ORACLE Enterprise Manager Cloud Control 13c

cdb1 / PDB1

Oracle 数据库 性能 可用性 安全性 方案 管理

数据编写

Oracle Data Redaction 提供了一种简单方法来快速编写应用程序中显示的敏感信息, 而无需变更磁盘或高速缓存中的基础数据库块。数据根据灵活的多因素策略实时编写。作为 Oracle Advanced Security

策略 表达式 格式

查看 格式 创建 类似创建 编辑 查看 删除 刷新 管理敏感列类型

格式名称	敏感列类型	函数类型	说明
American Express Credit Card Numbers - Formatted	CREDIT_CARD_NUMBER	PARTIAL	Redact the American Express Credit Card Number by replacing
American Express Credit Card Numbers - NUMBER	CREDIT_CARD_NUMBER	PARTIAL	Redact the American Express Credit Card Number by replacing
American Express Credit Card Numbers - Partially Redacted	CREDIT_CARD_NUMBER	REGEX	Redact the American Express Credit Card Number by replacing
American Express Credit Card Numbers - Random	CREDIT_CARD_NUMBER	RANDOM	Redact the American Express Credit Card Number by replacing
Canadian Social Insurance Numbers - Formatted	SOCIAL_INSURANCE_NUMBER	PARTIAL	Redact the Canadian Social Insurance Number by replacing the
Canadian Social Insurance Numbers - NUMBER	SOCIAL_INSURANCE_NUMBER	PARTIAL	Redact the Canadian Social Insurance number by replacing the
Canadian Social Insurance Numbers - Random	SOCIAL_INSURANCE_NUMBER	RANDOM	Redact the Canadian Social Insurance Number by replacing all
Canadian Social Insurance Numbers - VARCHAR	SOCIAL_INSURANCE_NUMBER	PARTIAL	Redact the Canadian Social Insurance number by replacing the
Credit Card Numbers - Formatted	CREDIT_CARD_NUMBER	PARTIAL	Redact the Credit Card Number by replacing everything but the
Credit Card Numbers - NUMBER	CREDIT_CARD_NUMBER	PARTIAL	Redact the Credit Card Number by replacing everything but the
Credit Card Numbers - Partially Redacted	CREDIT_CARD_NUMBER	REGEX	Redact the Credit Card Number by replacing everything but the
Credit Card Numbers - Random	CREDIT_CARD_NUMBER	RANDOM	Redact the Credit Card Number by replacing all digits with rand
Date to Epoch	UNDEFINED	PARTIAL	Redacts all dates to Jan 1st, 1970
Date to Millennium	UNDEFINED	PARTIAL	Redacts all dates to Jan 1st, 2000
Email Addresses	EMAIL_ID	REGEX	Redact the Email address by replacing the username with "xxx
IP Addresses	IP_ADDRESS	REGEX	Redact the IP address by replacing the machine/last quadrant o
North American Phone Numbers - Formatted	PHONE_NUMBER	REGEX	Redact the North American Phone Number by leaving the area c
North American Phone Numbers - NUMBER	PHONE_NUMBER	PARTIAL	Redact the North American Phone Number by leaving the area c
North American Phone Numbers - Random	PHONE_NUMBER	RANDOM	Redact the North American Phone Number by replacing all digit

创建

* 格式名称

* 说明

敏感列类型

UNDEFINED

* 编写函数

PARTIAL

基于正则表达式的编写。指定表示将编写的列数据的正则表达式。

函数属性

* 模式

指定要搜索的正则表达式模式。示例: '\d\d\d\d\d\d678' 用于搜索类似 '012345678' 的数字

* 替换字符串

示例: 使用 'XXXXXX13' (替换字符串) 将匹配 '(\d\d\d\d) (\d\d\d\d) (\d\d\d\d)' (正则表达式模式) 的 '012345678' (实际值) 编写为 'XXXXXX678' (编写的值)。请注意, 替换字符串中的 '13' 将保留模式中第三组括号内的实际数据。

* 位置

1

指定字符串搜索的起始位置。默认值 1 表示从列数据的第一个字符开始搜索。

* 出现

0

指定如何执行搜索和替换操作。零表示它将替换所有匹配项。正整数 'n' 表示将替换字符串的第 n 个匹配项。

匹配参数

忽略大小写

为 REGEX 编写函数指定匹配的参数。

确定

取消

2

脱敏类型

全部

- DBMS_REDACT.FULL
- 脱敏表或视图列的全部内容
- 默认情况下，字符数据类型: 输出文本为单个空格
- 数字数据类型: 输出文本为零
- 日期时间数据类型: 输出文本设置为2001年1月1日
- DBMS_REDACT.UPDATE_FULL_REDACTION_VALUES 可以更改默认值

部分

- DBMS_REDACT.PARTIAL
- 脱敏显示列中的部分内容
- 可以设置起始位置，替换需要的字符，以及替换多少个字符
- 987-65-4320 ***_**_4320
'VVVFVVFVVVV,VVV-VV-VVVV',*,1,5'
- 适用于信用卡和身份证号的脱敏

正则表达式

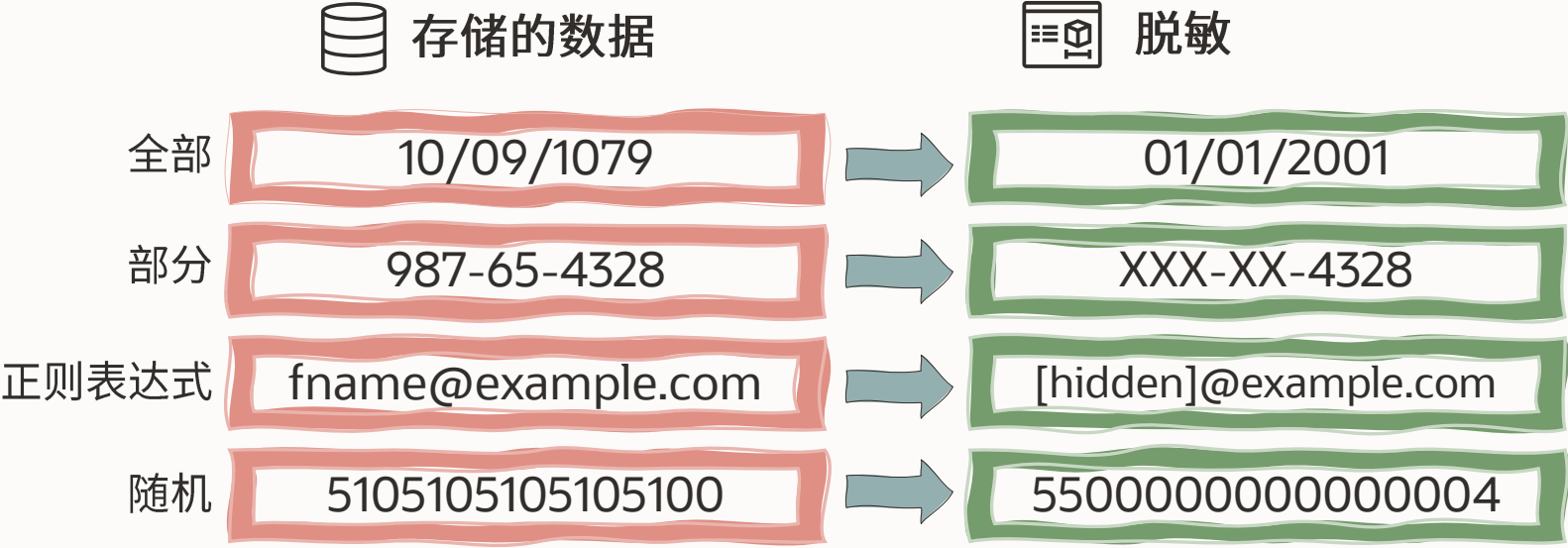
- DBMS_REDACT.REGEXP
- 根据模式搜索在列数据值中脱敏匹配的数据
- regexp_pattern 指定匹配模式
- regexp_replace_string 指定用什么字符串来替换匹配的数据
- regexp_position 指定模式搜索的起始位置
- regexp_occurrences 参数用于指定替换出现的匹配项序号

随机值

- DBMS_REDACT.RANDOM
- 整个值都会被屏蔽并替换为一个随机值
- 如果列上的数据类型是 char(20)，脱敏时会返回一个长度为20的随机串
- 如果列上的数据类型是 varchar2(20)，脱敏时生成的随机字符串长度会和实际值长度相同
- 数字类型：随机替换后，不会超过原数据的精度



脱敏类型一举例



Oracle数据库内置数据类型的脱敏支持

数据类型	全部	部分	正则	随机
CHAR, VARCHAR2, LONG VARCHAR2, NCHAR, NVARCHAR2	是	是	是	是
NUMBER, FLOAT, BINARY_FLOAT, BINARY_DOUBLE	是	是	否	是
RAW , LONG RAW	否	否	否	否
DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIME ZONE	是	是	否	是
BFILE	否	否	否	否
BLOB	是	否	否	否
CLOB	是	否	是	否
NCLOB	是	否	是	否
ROWID	否	否	否	否
UROWID	否	否	否	否



脱敏策略表达式

关于表达式

- 创建和修改脱敏策略时，需要指定表达式
- 当表达式返回true时，数据库才会运行脱敏逻辑

编辑表达式

- AND, OR, IN, NOT IN, =, !=, <>, <, >, >=, <=
- 可以使用sys_context() 函数，如USERENV命名空间，包含session_user, client_identifier 等信息
- 无法在表达式中使用自定义函数，可以使用SUBSTR(), LENGTH(), 部分Oracle Application Express 函数，部分Oracle Label Security 函数

表达式举例

- 应用于用户psmith的表达式

expression =>
'SYS_CONTEXT("USERENV","SESSION_USER") =
"PSMITH"'

- 应用于角色的表达式

expression =>
'SYS_CONTEXT("SYS_SESSION_ROLES","SUPERVISOR'
) = "FALSE"'

- 应用于所有用户的表达式

expression => '1=1'

```
BEGIN
DBMS_REDACT.ADD_POLICY(
  object_schema => 'HR',
  object_name   => 'EMPLOYEES',
  column_name   => 'SALARY',
  column_description => 'employee salary and commission',
  policy_name   => 'redact_emp_sal_comm',
  policy_description => 'Partially redacts',
  function_type => DBMS_REDACT.PARTIAL,
  function_parameters => '9,1,4',
  expression    => 'SYS_CONTEXT(''USERENV'', ''SESSION_USER'') = ''PSMITH''';
END;
```



命名表达式

什么是命名表达式

- 命名表达式用于替代现有动态脱敏策略中使用的策略表达式。
- 没有命名表达式之前，策略表达式应用于动态数据脱敏策略中的所有列
- 命名表达式出现后，可以为表或视图中的列定义各自的表达式，而不是共用一个表达式

- 可以与同一表或不同表的多个列相关联
- 可以与同一策略或不同策略中的列相关联
- 对命名表达式的任何更新都将应用到关联列上
- 无法为同一列关联多个命名表达式
- 在多租户环境中,不能将命名表达式与不同PDB中的列关联。
- 命名表达式的评估结果，优先于默认表达式

```
BEGIN
DBMS_REDACT.ADD_POLICY(
  object_schema      => 'HR',
  object_name        => 'EMPLOYEES',
  column_name        => 'SALARY, commission',
  column_description  => 'employee salary and commission',
  policy_name        => 'redact_emp_sal_comm',
  policy_description  => 'Partially redacts',
  function_type       => DBMS_REDACT.PARTIAL,
  function_parameters => '9,1,4',
  expression          => '1=0');
END;
```

```
BEGIN
DBMS_REDACT.CREATE_POLICY_EXPRESSION (
  policy_expression_name => 'redact_pol',
  expression              => 'SYS_CONTEXT(''USERENV'', ''SESSION_USER'') = ''PSMITH''',
  policy_expression_description => 'Determines whether the column will be redacted');
END;
/
BEGIN
DBMS_REDACT.APPLY_POLICY_EXPR_TO_COL (
  object_schema      => 'hr',
  object_name        => 'employees',
  column_name        => 'salary',
  policy_expression_name => 'redact_pol');
END;
/
```



动态数据脱敏



动态数据脱敏策略举例

```
BEGIN↓
DBMS_REDACT.add_policy (object_schema => 'ANYBANK'↓
, object_name      => 'COS'↓
, policy_name      => 'COS_REDACTION'↓
, expression       => 'SYS_CONTEXT(''USERENV'', 'SESSION_USER') != 'USER_ALL' ' '↓
, column_name      => 'GEO'↓
, function_type    => DBMS_REDACT.RANDOM↓
);↓

DBMS_REDACT.ALTER_POLICY(object_schema => 'ANYBANK'↓
, object_name      => 'COS'↓
, policy_name      => 'COS_REDACTION'↓
, action          => DBMS_REDACT.ADD_COLUMN↓
, column_name      => 'LEGAL_NAME'↓
, function_type    => DBMS_REDACT.RANDOM↓
);↓

DBMS_REDACT.ALTER_POLICY(object_schema => 'ANYBANK'↓
, object_name      => 'COS'↓
, policy_name      => 'COS_REDACTION'↓
, action          => DBMS_REDACT.ADD_COLUMN↓
, column_name      => 'CREDIT_CARD_NUM'↓
, function_type    => DBMS_REDACT.PARTIAL↓
, function_parameters => 'VVVVVVVVVVVVVVVV,VVV-VVV-VVV-VVV,*,1,12'↓
);↓
END;↓
/↓
↓
```

随机

部分

动态数据脱敏策略



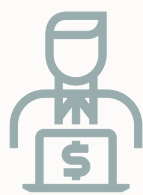
****_****_****-4368

4451-2172-9841-4368

呼叫中心
接线员



授权用户



SESSION_USER != "USER_ALL"

SESSION_USER = "USER_ALL"



如何能免受动态数据脱敏策略的影响

如何免受影响

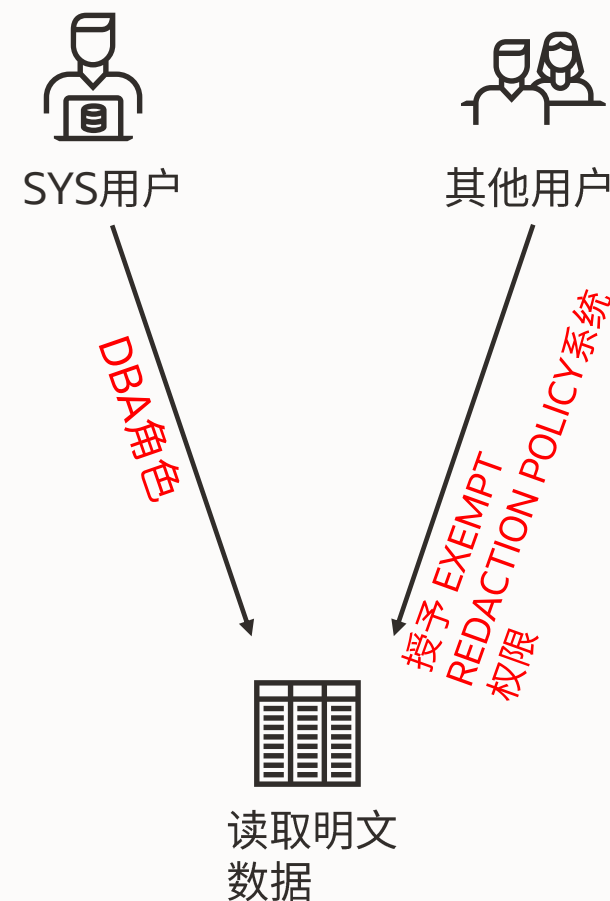
- 向可信用户授予 EXEMPT REDACTION POLICY 系统权限

默认情况下哪些用户会免受影响

- SYS用户
- 携带有EXEMPT REDACTION POLICY 系统权限的用户

注意

- 动态数据脱敏策略仅影响用户发出的SELECT语句，不影响用户发出的其他DML，包括INSERT、UPDATE或DELETE语句
- 如果执行CREATE TABLE AS SELECT涉及的列受脱敏策略保护，用户无法执行这些操作，除非用户被授予了EXEMPT REDACTION POLICY系统权限。
- DBA 角色中包含的EXP_FULL_DATABASE角色拥有EXEMPT REDACTION POLICY 系统权限
- EXEMPT REDACTION POLICY系统权限包含在DBA角色中,但无法通过with admin option 传递权限，需要显式授权



动手实践

实验内容



1. 创建脱敏策略
2. 创建全部类型的脱敏规则
3. 创建部分类型的脱敏规则
4. 创建正则表达式类型的脱敏规则
5. 创建随机值类型的脱敏规则
6. 创建针对sql developer工具的命名表达式
7. 授权用户系统权限EXEMPT REDACTION POLITY

常见问题

常见问题1

问： Oracle动态数据脱敏是怎么工作？

答：• Oracle动态数据脱敏是Oracle高级安全功能中的一个组件，它可以在SQL查询结果返回应用程序显示前，对敏感数据进行选择性的即时脱敏，从而未经授权的用户就无法查看敏感数据。

问： Oracle动态数据脱敏可以防范什么安全威胁？

答：• 使用Oracle动态数据脱敏后，可以限制向未授权的用户暴露敏感数据的风险。

- 不能防范针对特权数据库用户的攻击。例如,拥有SYSDBA等超级用户特权的用户或程序如果想看到未屏蔽的数据是可以的。另外,也不能防范SQL注入或推理攻击,攻击者可以通过反复的ad hoc数据库查询来猜测和最终发现真实的值。
- 建议采用深度防御策略来保护数据库,通过评估、预防和检测。

问： 与第三方代理解决方案相比,Oracle数据屏蔽的优势是什么？

答：• Oracle动态数据脱敏是Oracle数据库的一部分,与基于代理的解决方案相比,它的设置和维护非常简单。

- 能够提供更准确的结果，因为它允许所有查询处理正常完成，然后再屏蔽结果，而不是试图修改复杂的入站SQL语句。

常见问题2

问：Oracle动态数据脱敏性能如何？

答：• Oracle动态数据脱敏是Oracle数据库内核的一部分，提供了**内核级的速度**。对应用程序或数据库本身的**性能影响很小**。在大多数情况下,完全脱敏和部分脱敏的性能开销**接近为零**。

问：Oracle动态数据脱敏需要更改应用程序吗？

答：• 对于使用**一个大数据库用户**建立数据库连接的应用系统，**需要做适当调整**，比如数据库连接建立时注入应用中的当前用户信息，用户名或角色，然后动态数据脱敏策略会基于此信息进行脱敏判断

- 对于使用应用**程序中的登录用户**作为数据库连接用户的系统，**不需要做任何改动**。

问：在脱敏数据时，可以使用哪些运行时因素？

答：• 可以根据数据库会话中的几个运行时因素来脱敏数据，比如**数据库用户、os用户、数据库角色、IP地址、主机名和应用程序标识符**。

使用指南

Oracle动态数据脱敏使用指南

- 不要在GROUP BY子句的SQL expression中包含脱敏列。
ORA-00979: not a GROUP BY expression
- 不要在DISTINCT和ORDER BY**同时使用**的子句的SELECT expression中包含脱敏列。ORA-01791: not a SELECTed expression
- 在视图定义中不建议包含被脱敏列的SQL表达式
(如 concat, sum, trim, min, max等)
- WHERE子句不会被脱敏，可用于比较
只脱敏SELECT列表中的数据。
- 区分动态数据脱敏(PII)和访问控制场(按需访问)景的差别

```
select userid, firstname, sin, ssn, nino from employeesearch_prod.demo_hr_employees
2  where sin='170-042-126';
```

USERID	FIRSTNAME	SIN	SSN	NINO
77	Alice			

Sample example that works

```
1 SELECT COLUMN_NAME_1
2 FROM TABLE_NAME
3 WHERE COLUMN_NAME_ID BETWEEN 500 AND 600
4 GROUP BY COLUMN_NAME_1
```

Sample example that doesn't work

```
1 SELECT SUBSTR(COLUMN_NAME_1, 1, 5)
2 FROM TABLE_NAME
3 WHERE COLUMN_NAME_ID BETWEEN 500 AND 600
4 GROUP BY SUBSTR(COLUMN_NAME_1, 1, 5)
```

Case 1. Throw ORA-01791

```
SELECT DISTINCT pca_pca_id
FROM SA_PRESTACIONES_FACTURADAS
ORDER BY 1
```

Case 2. Finished successfully

```
SELECT DISTINCT vda_drv_proceso
FROM SA_PRESTACIONES_FACTURADAS presfa
ORDER BY 1
```

```
SELECT DISTINCT sensitive_column FROM table_name
ORDER BY sensitive_column;

SELECT sensitive_column FROM
(SELECT DISTINCT sensitive_column FROM table_name ORDER BY sensitive_column);
```


MLE助力APEX全栈开发

数据库和云系列讲座

邢红

- 资深企业应用解决方案专家
- 丰富的应用开发经验
- Oracle APEX (什)
OCI OpenSearch (么)
Oracle区块链平台 (都)
Oracle 分析云 (干)



内容简介

- Oracle MLE介绍
- MLE JavaScript 助力APEX
- MLE Demo分享



Zoom直播

直播时间: 9月1日 11:00 - 12:00

扫描二维码进入直播

Zoom ID: 957 9669 6723

密码: 20212023



微信扫一扫预约



20-21

数据库和云讲座群



甲骨文云技术公众号



技术专家1V1深入交流