
ORACLE®

ORACLE

Oracle Database 12*c* Release 1

セキュリティ

日本オラクル株式会社

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。以下の事項は、マテリアルやコード、機能を提供することをコミットメント（確約）するものではないため、購買決定を行う際の判断材料になさらないで下さい。オラクル製品に関して記載されている機能の開発、リリースおよび時期については、弊社の裁量により決定されます。

- **Oracle Advanced Security新機能**
  - **TDEマスター暗号鍵管理**
  - **Data Redaction**

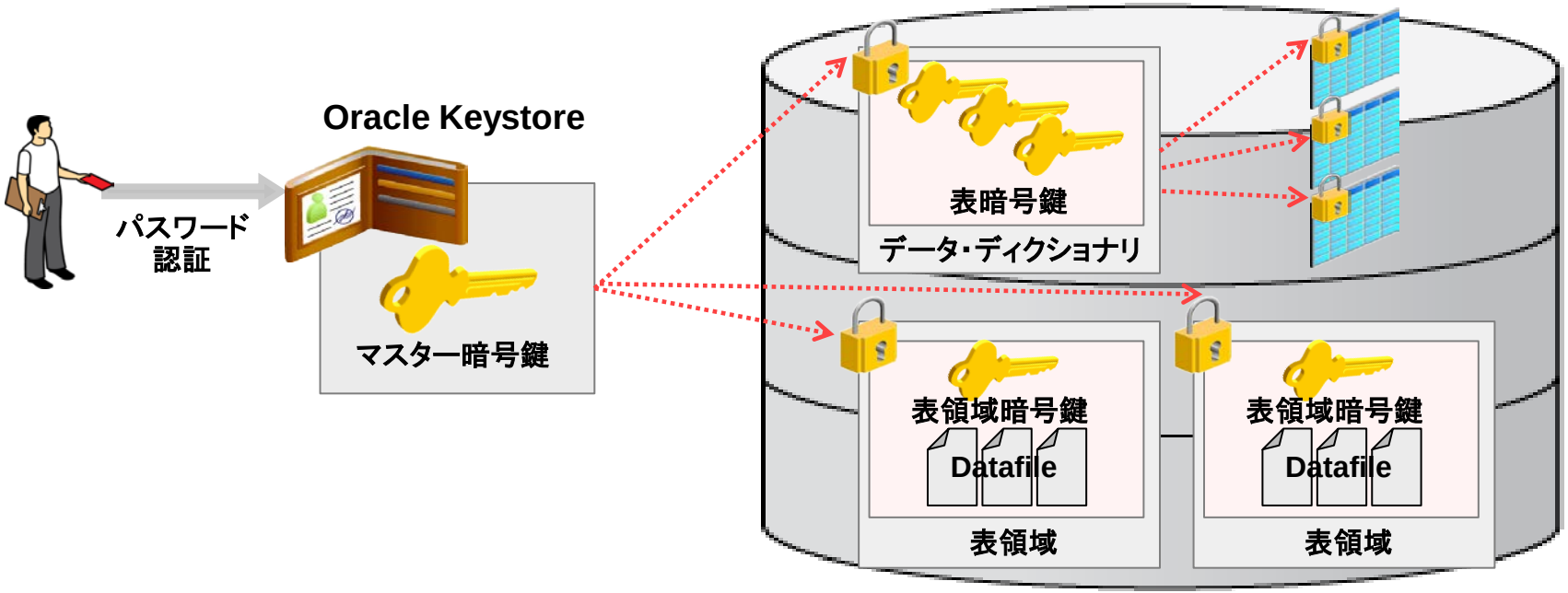
Oracle Advanced Securityで実現する2つの機能

データの暗号化
Transparent Data Encryption

リアルタイムアクセス制御
Oracle Data Redaction

Transparent Data Encryptionの暗号鍵の仕組み

- Oracle Keystoreに、マスター暗号鍵が格納される
- マスター暗号鍵は、それぞれの列暗号鍵と表領域暗号鍵を暗号化する
- 表ごとの暗号鍵、表領域ごとの暗号鍵でそれぞれの実データを暗号化する



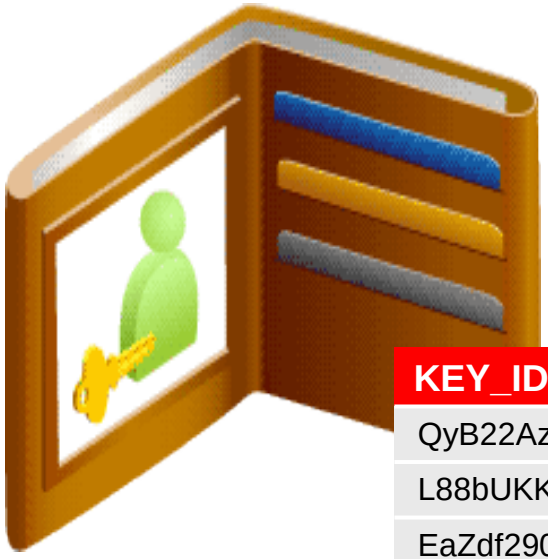
新しくなったマスター暗号鍵管理

暗号鍵のローテーションをサポート

- 鍵管理の専用権限SYSKMが追加
- 従来のOracle Walletユーティリティ(mkstore,orapki)から、SQL*PLUSで実行するADMINISTER KEY MANAGEMENT文に統合
- V$ENCRYPTION_KEYSでマスター暗号鍵の情報を参照可能
- KeystoreをASM上に配置することが可能
- KEY_IDに対するタグ情報の追加が可能
- マスター暗号鍵の利便性が向上
  - Merge, Export, Import
  - Backup
  - Activation
- マルチテナント・アーキテクチャに対応

マスター暗号鍵の状態・履歴を参照

V$ENCRYPTION_KEY



- KEY_ID ・・・マスター暗号鍵
- TAG ・・・任意のタグ（コメント）
- CREATION_TIME ・・・マスター暗号鍵の作成日
- ACTIVATION_TIME ・・・マスター暗号鍵のアクティブ日
- BACKED_UP ・・・バックアップの有無
- ・・・・

KEY_ID	Tag	ACTIVATION_TIME	
QyB22Az…	システム移行のため	2012-12-30 08:02:00	In Use
L88bUKK…	2012年定期変更	2012-07-01 07:58:32	Rotated
EaZdf290…	バックアップ作業	2012-03-15 06:52:17	Rotated
ddJwwWJ…	2011年定期変更	2011-07-01 10:11:02	Rotated
GGgSB36…	システムC/O	2011-01-01 08:25:33	Rotated

12cのマスタ暗号鍵の構成手順

- sqlnet.oraにKeystoreを作成するロケーションを記述する

```
ENCRYPTION_WALLET_LOCATION=
  (SOURCE=
    (METHOD=FILE)
      (METHOD_DATA=
        (DIRECTORY=keystore_location)))
```

- SYSKM権限でSQL*PLUSにログインする

```
SQLPLUS / AS SYSKM
Enter password: password
Connected.
```

- Keystoreを作成する。成功すると指定のロケーションにewallet.p12ファイルが作成される

```
ADMINISTER KEY MANAGEMENT CREATE KEYSTORE ‘keystore_location’
IDENTIFIED BY password;
```

12cのマスタ暗号鍵の構成手順

- Keystoreをオープンする。

```
ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
password;
```

- マスター暗号鍵を作成する。※変更・再作成の場合も同様

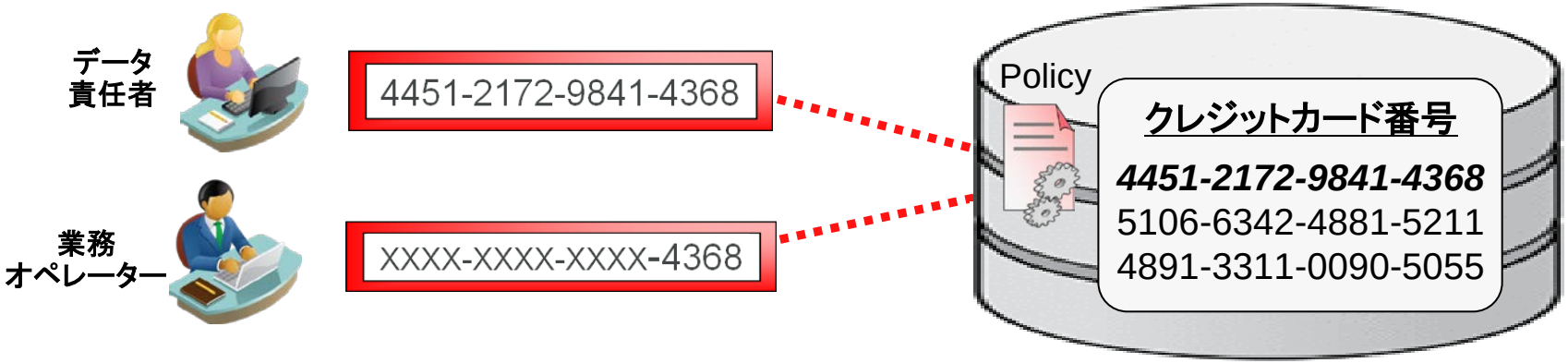
```
ADMINISTER KEY MANAGEMENT SET KEY [USING TAG ‘xxxxxxxx']
IDENTIFIED BY password WITH BACKUP;
```

- Keystoreにマスター暗号鍵が生成され、Keystoreのあるロケーションにバックアップが作成される。以降は、通常通りの暗号表、暗号化表領域の作成手順へ

```
SELECT KEY_ID,ACTIVATION_TIME FROM V$ENCRYPTION_KEYS;

KEY_ID                                                                                    CREATION_TIME
----------------------------------------------------------------------------------------  ---------------------------
AfrZm0w5EE9kv2NNme6cpwIAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  13-02-07 06:59:41
```

Oracle Data Redaction

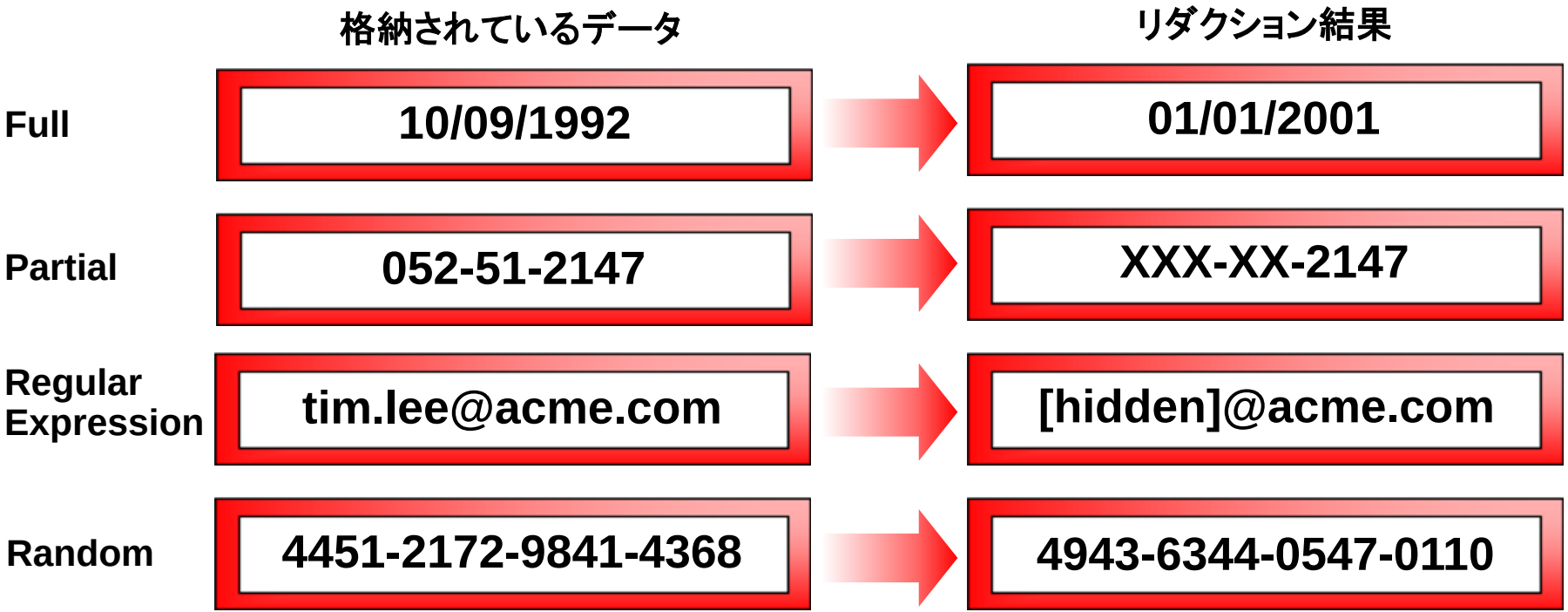
ユーザーの権限に応じたリアルタイム・アクセスコントロール



- ユーザーの権限やクライアント情報に応じてリアルタイムにデータをリダクション
- アプリケーションのコード修正は必要のないデータベース内で完結する列アクセス制御
- コールセンターやサポート業務などの職責に応じた顧客情報へのアクセス制御の実現やPCIDSSに対応したクレジットカード番号の表示、アプリ開発者の直接アクセスも制御

サポートされるリダクションの種類

用途の応じたデータの参照範囲を定義



Oracle Databaseのアクセス制御の特徴

	Data Redaction	Virtual Private Database	Database Vault
機能概要	列のアクセス制御 & リダクション	行・列のアクセス制御	表のアクセス制御 特権ユーザー管理
必要ライセンス	Advanced Security Option	Enterprise Edition	Database Vault Option
バージョン	12c～	8i～(列は10gR1～)	10gR2～
対象アクセス	列(SELECT)	列・行(DML)	オブジェクト・SQLコマンド
説明	表に定義したリダクションポリシーの条件に応じて、列を表示させないまたは、任意の値にリダクション	表に定義したVPDポリシーの条件に応じてWHERE句を自動的に付与することで行を表示させない。 その際に特定の列をNULL表示させることもできる	レルム、ルール、コマンドルールの各要素を使って、オブジェクト(表やビュー、PL/SQL等)へのアクセス、SQLコマンド自体の実行を詳細に強制アクセス制御することができる
特権ユーザー	ポリシーは適用されない	ポリシーは適用されない	どのユーザーでも ポリシーは適用される
設定	DBMS_REDACTパッケージ または、Oracle Enterprise Manager	DBMS_RLSパッケージ または、Oracle Enterprise Manager	DVSYS.DBMS_MACADMパッケージ または、Oracle Enterprise Manager

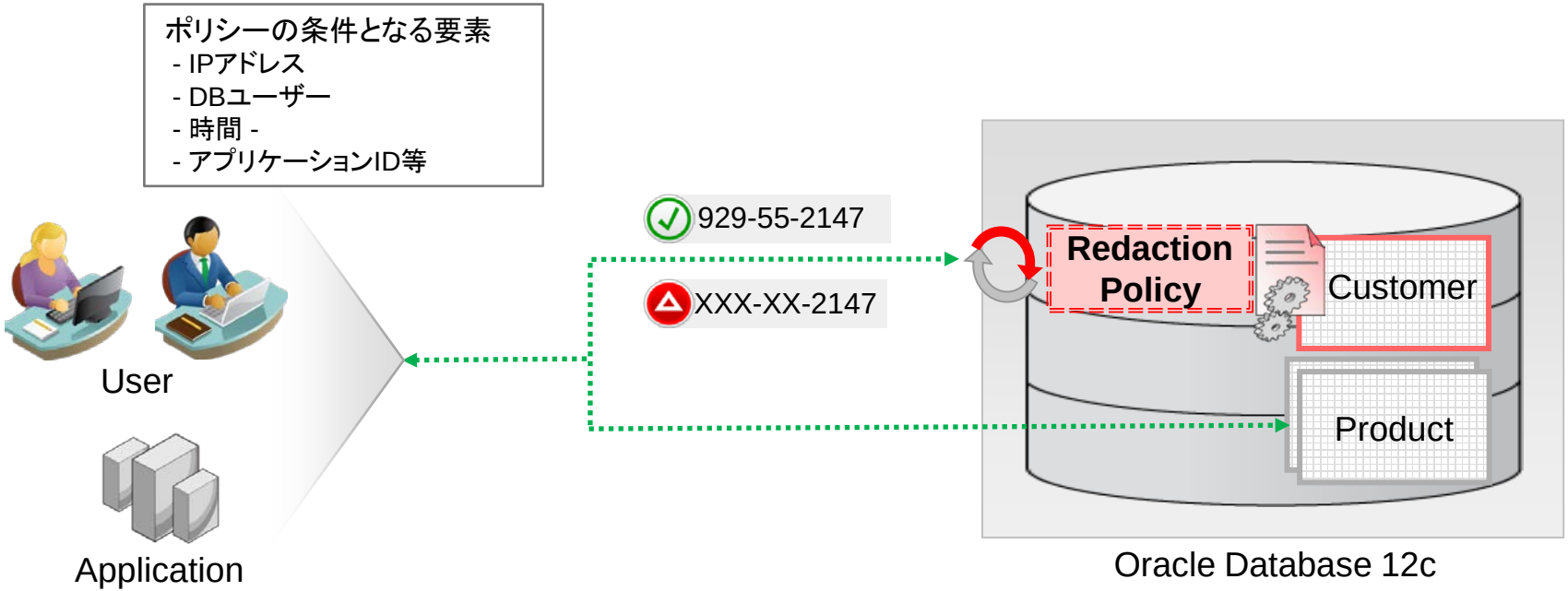
Oracle Data Masking との違い

データをマスクする結果は同じだが、用途が異なる

	Oracle Data Masking	Oracle Data Redaction
実装方式	Oracle Enterprise Manager	DBMS_REDACTパッケージ または、Oracle Enterprise Manager
目的	表を直接マスキングし、 本番に近いテストデータを作成	ユーザーの権限に応じて表やビュー をリダクションするアクセス制御
実行タイミング	オフライン 表・データベースのクローン作成後に マスキングを実行	オンライン 問い合わせ結果にリアルタイムで マスク処理を実行
格納データへの影響	永続的にデータを変更	影響なし

Oracle Data Redactionのアーキテクチャ

- リダクション・ポリシーを表やビューに対してDBMS_REDACTプロシージャで定義
- 対象にできる列は、CHAR/VARCHAR2、NUMBER、DATE、BLOB/CLOB型
- リダクション・ポリシーの条件に応じて、列の値を任意にリダクションする



リダクション・ポリシーの作成

DBMS_REDACT.ADD_POLICYプロシージャ

DBMS_REDACT.ADD_POLICY	
object_schema	リダクション・ポリシーを適用するスキーマ名
object_name	リダクション・ポリシーを適用する表、またはビュー名
policy_name	作成するリダクション・ポリシー名
column_name	リダクション・ポリシーを適用する列名 ※複数指定したい場合は、DBMS_REDACT.ADD_POLICYで別途追加する
function_type	DBMS_REDACT.FULL DBMS_REDACT.RANDOM DBMS_REDACT.PARTIAL DBMS_REDACT.REGEXP
expression	SYS_CONTEXTの値に基づく、Boolean型の条件式を定義。 条件の結果値が“True”である場合のみ、リダクションが実行される
function_parameters	DBMS_REDACT.PARTIALを使用する場合のデータのINとOUTの定義
regexp…….	function_typeがDBMS_REDACT.REGEXPの場合のオプション群

Expression(条件式)の作成方法

- SYS_CONTEXTでセッション情報を取り出し、比較する条件の値を取得する
- 結果がTRUE or FALSEで評価できるように作成し、TRUEの場合にリダクションが行われる
  - DBユーザー名がSCOTTの場合

```
SYS_CONTEXT('USERENV','SESSION_USER') = 'SCOTT'
```

  - IPアドレスがNULLの場合

```
SYS_CONTEXT('SERENV','IP_ADRESS')  IS NULL
```

  - クライアント情報にMGRのユーザー名が含まれていなかった場合

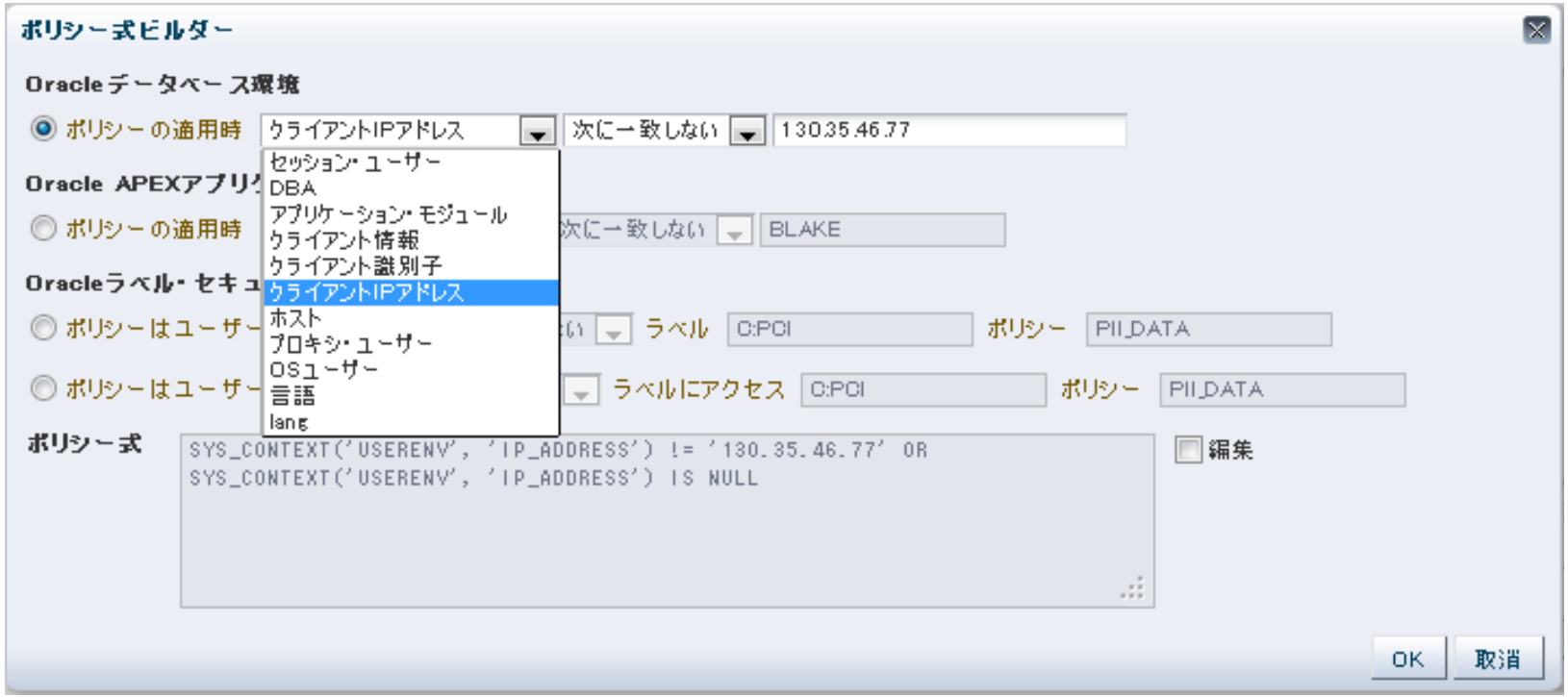
```
SYS_CONTEXT('USERENV', CLIENT_IDENTIFIER') not like 'MGR%'
```

  - ユーザーがMGRロールを持っていなかった場合

```
SYS_CONTEXT('SYS_SESSION_ROLES','MGR') = FALSE
```

ポリシー式ビルダーで条件作成をサポート

Oracle Enterprise Manager 12c



Full Redaction (フル・リダクション)

- HRユーザー以外がEMPLOYEES表のSALARY列にアクセスした際にリダクションさせる

```
BEGIN
  DBMS_REDACT.ADD_POLICY(
    object_schema       => ‘HR',
    object_name         => ‘EMPLOYEES',
    policy_name         => ‘EMPLOYEE _POLICY_SAL’,
    expression          => 'SYS_CONTEXT("USERENV","SESSION_USER") != ‘’HR"',
    column_name         => ‘SALARY',
    function_type       => DBMS_REDACT.FULL);
END;
```

実行結果

```
SELECT SALARY  FROM EMPOYEES;

SALARY
--------------
             0
```

データ型によって固定値でリダクションが行われる
以下、初期値。任意の値への変更は可能

文字列: シングルスペース
数値: ゼロ（0）
日付: 01-JAN-01
LOB: [redacted]

Random Redaction (ランダム・リダクション)

- HRユーザー以外がEMPLOYEES表のEMPLOYEE_ID列のアクセスした際リダクションさせる

```
BEGIN
  DBMS_REDACT.ADD_POLICY(
    object_schema       => ‘HR',
    object_name         => ‘EMPLOYEES',
    policy_name         => ‘EMPLOYEE _ POLICY_EMPID’,
    expression          => 'SYS_CONTEXT("USERENV","SESSION_USER") != ‘’HR"',
    column_name         => ‘EMPLOYEE_ID ',
    function_type       => DBMS_REDACT.RANDOM);
END;
```

実行結果

```
SELECT EMPLOYEE_ID FROM EMPOYEES;

EMPLOYEE_ID
----------------------
                167
```

データ型によってそれぞれの形式でリダクション

文字列: ランダム文字

数値: ランダム数値

日付: ランダム日付

LOB: 使用できない

Partial Redaction (部分リダクション)

- SQL*PLUSでEMPLOYEES表のPHONE_NUMBER列のアクセスした際リダクションさせる

```
BEGIN
  DBMS_REDACT.ADD_POLICY (
    object_schema        => ‘HR',
    object_name          => ‘EMPLOYEE ',
    policy_name          => ‘EMPLOYEE _ POLICY_PHONE',
    expression           => 'UPPER(SYS_CONTEXT("USERENV","MODULE")) like "%SQL*PLUS%"',
    column_name          => ' PHONE_NUMBER ',
    function_type        =>  DBMS_REDACT.PARTIAL,
    function_parameters => ‘VVVFVVVFVVVV,VVV-VVV-VVVV,*,1,6 ');
END;
```

実行結果

```
SELECT PHONE_NUMBER FROM EMPOYEES;

PHONE_NUMBER
----------------------------
 ***-***-8080
```

データ型によってそれぞれの形式でリダクション

- **文字列**: 部分的に任意の文字列
- **数値**: 部分的に任意の数値
- **日付**: 部分的に任意の日付
- **LOB** : 使用できない

Function_parametersの設定方法

文字列を部分リダクションする場合

3528 3589 1231 0001 ▶ ******-****-****-0001**

function_parameters => 'VVVVFVVVVFVVVVFVVVV,VVVV-VVVV-VVVV-VVVV,*,1,12',

- Input format --> 現在のフォーマットを定義。Vはリダクション可能、Fはフォーマット固定
- Output format --> リダクション後のフォーマット定義。Vはリダクション可能、ハイフンなどフォーマット固定する文字
- Mask Character --> リダクション結果を表示する文字
- Starting digit position --> リダクションの開始位置
- Ending digit position --> リダクション終了位置。InputにFが含まれる場合はカウントしない

数値を部分リダクションする場合

0123456789 ▶ **9999456789**

function_parameters => ‘9,1,4’

- Mask Character --> リダクション結果を表示する文字
- Starting digit position --> リダクションの開始位置
- Ending digit position --> リダクション終了位置。InputにFが含まれる場合はカウントしない

Regular Expression-based Redaction（正規表現 リダクション)

```
BEGIN
 DBMS_REDACT.ADD_POLICY(
   object_schema            => ‘HR',
   object_name              => ‘'EMPLOYEES',
   policy_name              => ‘'EMPLOYEE _POLICY_REG',
   expression               => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ‘HR'',
   column_name              => ‘'PHONE_NUMBER',
   function_type            => DBMS_REDACT.REGEXP,
   regexp_pattern           => '([0-3][0-3][0-3])',
   regexp_replace_string    => ‘***',
   regexp_position          => 1,
   regexp_occurrence        => DBMS_REDACT.RE_ALL,
   regexp_match_parameter => 'i');
END;
```

実行結果

```
SELECT  PHONE_NUMBER FROM EMPOYEES
PHONE_NUMBER
------------------------
650.###.5234
650.124.###4
```

EMPLOYEES表のPHONE_NUMBER**列の値に**0-3の連続する3桁の数字がある場合は、その値のみにリダクションを行う

正規表現の指定方法

- 特定の数値のみをリダクションさせる場合

603.123.6666  **603.###.6666**

- regexp_pattern => '([0-3][0-3][0-3])'

適合するデータの検索パターンを定義。0から3の連続した数字が連続して3つ並んでいることを意味する

- regexp_replace_string => '#'

適合した場合のリダクションする文字を定義

- regexp_position => 1

検索の開始位置を指定します

- regexp_occurrence => 0

リダクション回数。0であれば適合する部分をすべて置き換える

- regexp_match_parameter => 'i‘

適合方法を指定。例えば、iは大文字と小文字を識別することを示す

正規表現の詳細な使い方は、
Oracle® Database SQL言語リファレンス
REGEXP_REPLACEを参考にして下さい

リダクション・ポリシーの列追加

DBMS_REDACT.ALTER_POLICYプロシージャ

DBMS_REDACT.ADD_POLICY	
object_schema	リダクション・ポリシーに追加するスキーマ名
object_name	リダクション・ポリシーを追加する表、またはビュー名
policy_name	追加するリダクション・ポリシー名
action	DBMS_REDACT.ADD_COLUMN
column_name	リダクション・ポリシーを追加する列名
function_type	DBMS_REDACT.FULL DBMS_REDACT.RANDOM DBMS_REDACT.PARTIAL DBMS_REDACT.REGEXP
function_parameters	DBMS_REDACT.PARTIALを使用する場合のデータのINとOUTの定義
regexp…….	function_typeがDBMS_REDACT.REGEXPの場合のオプション群

※条件は、DBMS_REDACT.ADD_POLICYで作成したものが使用される

リダクション・ポリシーの列追加

EMPLOYEES表のEMPLOYEE_ID列にリダクション・ポリシー作成

```
BEGIN
  DBMS_REDACT.ADD_POLICY(
    object_schema    =>'HR ',
    object_name      =>'EMPLOYEES',
    policy_name      =>'EMPLOYEE _POLICY_EMPID',
    expression       =>'SYS_CONTEXT("USERENV","SESSION_USER") !="HR"',
    column_name      => 'EMPLOYEE_ID',
    function_type    => DBMS_REDACT.RANDOM);
END;
PL/SQLプロシージャが正常に完了しました。
```

PHONE_NUMBER列を追加

```
BEGIN
  DBMS_REDACT.ALTER_POLICY (
    object_schema       =>'HR ',
    object_name         =>'EMPLOYEES',
    policy_name         =>'EMPLOYEE _POLICY_EMPID',
    action              => DBMS_REDACT.ADD_COLUMN,
    column_name         => 'PHONE_NUMBER',
    function_type       => DBMS_REDACT.PARTIAL,
    function_parameters => 'VVVFVVVFVVVV,VVV-VVV-VVVV,*,1,6');
END;
PL/SQLプロシージャが正常に完了しました。
```

リダクション・ポリシーの削除

DBMS_REDACT.DROP_POLICYプロシージャ

DBMS_REDACT.DROP_POLICY	
object_schema	リダクション・ポリシーを削除スキーマ名
object_name	リダクション・ポリシーを削除する表、またはビュー名
policy_name	削除するリダクション・ポリシー名

```
BEGIN
  DBMS_REDACT.DROP_POLICY (
    object_schema  => 'HR',
    object_name    => 'EMPLOYEES',
    policy_name    => 'EMPLOYEE _POLICY_SAL');
END;

PL/SQLプロシージャが正常に完了しました。
```

Oracle Data Redactionの制限事項

- 表に定義できるのはひとつのリダクション・ポリシーのみ
- 表にリダクション・ポリシーを設定している場合は、その表から派生するビューは
  すべてリダクションされる
- 以下のデータベースの操作に対してはリダクションは行われない
  - Backup, Restore
  - Export, Import
  - Upgrade, Patch
  - Dataguard, Replication
- SYSDBA権限を保持するユーザーはリダクションされない
- システム権限　EXEMPT REDACTION POLICYを持つユーザーは、リダクションされない
- マテリアライズド・ビューを作成、リフレッシュする際にはリダクションされない

データベースの適切な権限管理が前提
SYSDBAのアクセスを強制的に制御したい場合には Oracle Database Vault
との組み合わせで実現

実際にアプリケーションで使用した場合

- 接続するユーザーの所有するロールで制御しており、**アプリケーションの修正は必要なし**
  expression => 'SYS_CONTEXT(''SYS_SESSION_ROLES'','''MGR''') = ''FALSE'''でコントロール

- データベース監査 新機能
  - Unified Auditing

Oracle Databaseの監査機能

	①必須監査(オペレーティング・システム監査)	②DBA監査	③標準監査（任意監査）	④ファイングレイン監査（任意監査）
必要Edtion	全エディション	全エディション	全エディション	Enterprise Edition
対象バージョン	すべて	Oracle 9i～11gR2	Oracle 9i～11gR2	Oracle 9i～11gR2
監査対象	・インスタンス起動 ・インスタンス停止 ・管理者権限によるデータベース接続	・データベース管理者としてログインしたユーザーのデータベース操作	・データベースへの操作（ログイン、CREATE/ALTER/DROPなどのアクション、UPDATE、DELETEなどのオブジェクトへの操作）	・特定のデータ（列名、条件指定可能）へのアクセス（SELECT） ・Oracle10gからはUPDATE、DELETE、INSERTへも可能
監査証跡出力先	・OSファイル	・OSﾌｧｲﾙ / ｼｽﾃﾑﾋﾞｭｰｱ(Win) ・Syslog(10gR2～)	・DBA_AUDIT_TRAILビュー ・OSﾌｧｲﾙ / ｼｽﾃﾑﾋﾞｭｰｱ(Win) ・Syslog(10gR2～) ・XMLファイル(10gR2～)	・DBA_FGA_AUDIT_TRAILビュー ・ユーザー定義表 ・メール送信も可能 ・XMLファイル(10gR2～)
取得可能な監査証跡	・OSによって生成された監査レコード ・データベース監査証跡レコード ・常に監査されるデータベース関連のアクション ・管理ユーザー（SYS）用の監査レコード	・時刻 ・操作(SQL文全体) ・データベースユーザー名/権限 ・OSユーザー名/端末 ・終了コード	・時刻 ・操作(SQL文の種類) ・データベースユーザー名/権限 ・OSユーザー名/端末 ・終了コード	・時刻 ・データベースユーザー ・OSユーザー名/端末 ・アクセスしたオブジェクト名 ・ファイングレイン監査ポリシー名 ・操作(SQL文全体) ・ユーザー定義アクション

従来のデータベース監査機能の課題

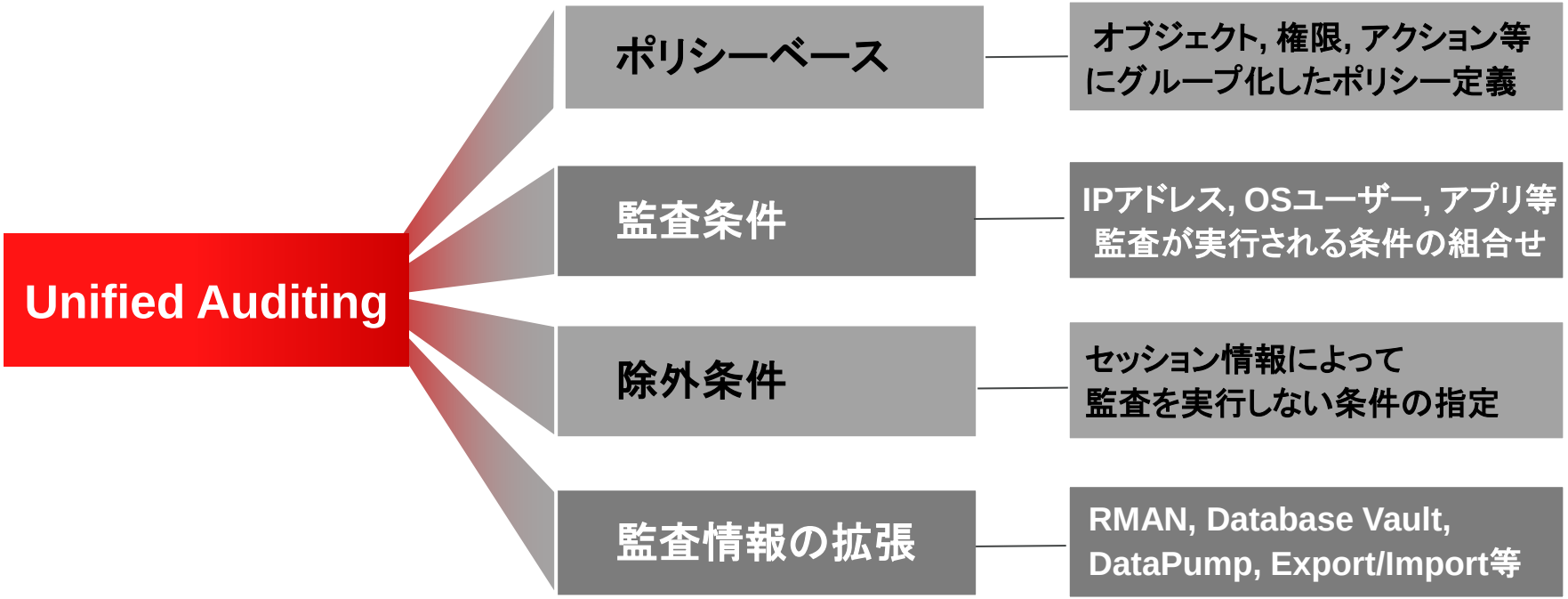
audit_trailアーキテクチャ

- Auditコマンドによる細かな設定が必要
- 特定のセッション情報に限定したログ取得ができない
- Audit取得によるパフォーマンスへの影響が心配
- データベースの機能やユーティリティによってログの出力先が異なる

従来のAUDIT機能の拡張ではなく、
“使いやすさ”＆”速さ”を目指した新しいアーキテクチャが必要

Unified Auditing

シンプル＆高速, 新たにデザインされたデータベース監査機能



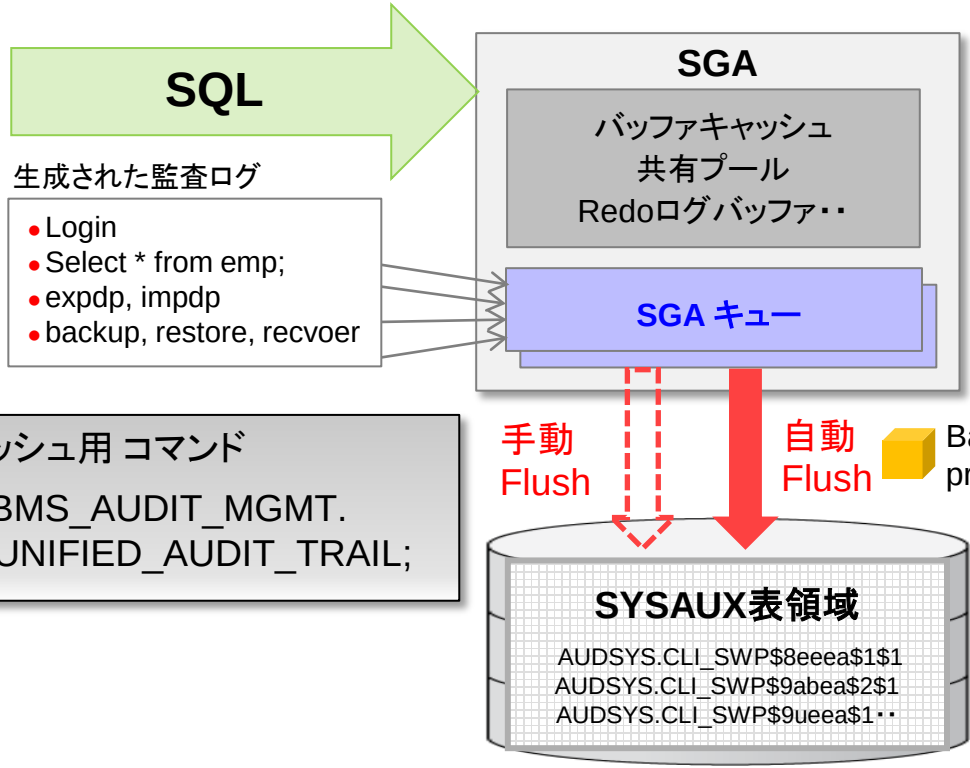
Unified Auditing

従来の監査機能との比較

	従来	Unified Auditing(12c)
1. 監査の定義	監査対象毎で定義	ポリシーで定義 一つのポリシーでDB内のすべてを監査可能
2. 監査条件	指定不可	条件と監査頻度指定可能
3. 監査ユーザーの指定	BYで監査ユーザー指定可能	BYで監査ユーザー指定可能 EXCEPTで免除ユーザーも指定可能
4.初期化パラメータ	設定必須	設定不要
5.監査証跡レコード	・SYS.AUD$とSYS.FGA_LOG$ ・OS監査レコード・ファイル ・DB監査レコード・ファイル ・XML形式でOS監査レコード・ファイル	・AUDSYSスキーマとしてSYSAUX表領域に一元的に格納 ・DB書込み不可の場合、$ORACLE_BASE/audit/$ORACLE_SIDに監査レコード・ファイルに格納、UNIFIED_AUDIT_TRAILにインポート可

Unified Auditingの新しいアーキテクチャ

2つのSGAキューによる並列処理・非同期書き込み



SGAキュー

- SGAに監査ログを保存しておく領域
- UNIFIED_AUDIT_SGA_QUEUE_SIZE サイズで指定
- 接続クライアントごとに2つのキューを持つ
- SGAキューがSYSAUXに書き込みをしている間は、もうひとつのSGAキューが監査ログを保存する

手動フラッシュ用 コマンド

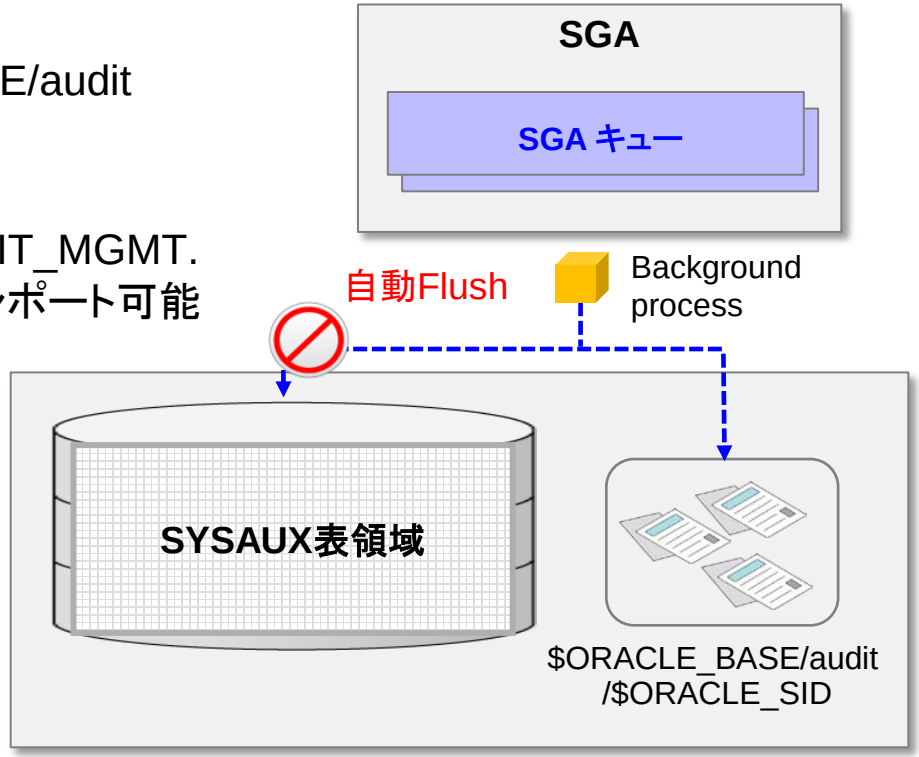
```
EXEC DBMS_AUDIT_MGMT.
FLUSH_UNIFIED_AUDIT_TRAIL;
```

バックグランドプロセスが3秒、
キューのしきい値に達したタイミングで
SYSAUX表領域にフラッシュ
-> AUDSYSユーザーの読み取り専用表

フラッシュできなかった場合のアーキテクチャ

ログの取り漏れを防ぐ

- SYSAUXにフラッシュできない場合は、バイナリファイルとして$ORACLE_BASE/audit/$ORACLE_SIDに書き込まれる
- 出力された監査ファイルは、UNIFIED_AUDIT_TRAILDBMS_AUDIT_MGMT.LOAD_UNIFIED_AUDIT_FILESでインポート可能
- インポート完了後には、監査ファイルは削除される



監査ログの書込み方式

同期・非同期モード

書込み方式	特性
Queued-write mode **キュー書込み**	• SGAキューによる非同期書込み方式 • デフォルト • UNIFIED_AUDIT_SGA_QUEUE_SIZEパラメータにより、SGAキューのサイズを1MBから30MBの範囲で指定できる。デフォルト値は1MB • インスタンス障害やSHUTDOWN ABORTなどデータベースが正常に停止しなかった場合、SGAキューにある監査ログが失われる可能性がある
Immediate-write mode **即時書込み**	• 従来の同期書込み方式 • すべてのログは取得されるが、パフォーマンスへの影響を与える可能性がある

監査ログの書込み方式の設定

DBMS_AUDIT_MGMTパッケージ

- Queued-write modeの設定

```
BEGIN
    DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_WRITE_MODE,
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_IMMEDIATE_WRITE);
 END;
```

- Immediate-write modeの設定

```
BEGIN
    DBMS_AUDIT_MGMT.SET_AUDIT_TRAIL_PROPERTY(
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_WRITE_MODE,
    DBMS_AUDIT_MGMT.AUDIT_TRAIL_QUEUED_WRITE);
 END;
```

監査ポリシーの作成～有効化

- 監査ポリシーの作成

CREATE AUDIT POLICY ポリシー名

ROLES　ロール名,・・

PRIVILEGES　システム権限,・・

ACTIONS オブジェクト権限 ON オブジェクト名,・・

ACTIONS COMPONENT = DATAPUMP, DV, DIRECT_LOAD, OLS

WHEN　監査を実行する条件

EVALUATE PER [STATEMENT, SESSION, INSTANCE]

CONTAINER = [CURRENT, ALL]

- 監査ポリシーの有効化

AUDIT POLICY ポリシー名　[BY,EXCEPT] ユーザ名

条件式の作成例

WHEN句で監査条件の指定

- ファイングレイン監査でしか出来なかった監査条件の指定が、Unified Auditingでも可能に

監査条件	
ローカル接続のみ	SYS_CONTEXT('USERENV','IP_ADDRESS') IS NULL
アプリケーションがSQL*Plus	SYS_CONTEXT('USERENV','MODULE') ='SQL*Plus'
接続クライアントがClient001以外	SYS_CONTEXT(''USERENV',HOST') <> 'Client001'
OSユーザーがOracle以外	SYS_CONTEXT(''USERENV',OS_USER') <> 'oracle'
クライアント識別子がID_XXXXのフォーマット	SYS_CONTEXT('USERENV', CLIENT_IDENTIFIER') like 'ID_%'
ADMINのロールを持っていない	SYS_CONTEXT('SYS_SESSION_ROLES','ADMIN') =FALSE

※ 列単位での監査条件、監査実行後のアクション(例メール配信等)はファイグレイン監査のみ可能

アプリケーション情報の監査

- アプリケーション固有の情報をデータベースの監査ログに出力させる場合は、以下の手順・コードを含める必要がある

SQL*PLUSの場合

SQL*PLUSの場合、接続後に以下を実行

execute dbms_session.set_identifier('任意の値')

Ex)
execute dbms_session.set_identifier('user=tanaka id=001234')

JDBCの場合

DBへ接続オープン後、以下を追加

```
String metrics[] =
new String[OracleConnection.END_TO_END_STATE_INDEX_MAX];
metrics[OracleConnection.END_TO_END_CLIENTID_INDEX] = "任意の値";
conn.setEndToEndMetrics(metrics, (short) 0);
```

.NETの場合

DBへ接続オープン後、以下を追加

```
conn.ClientId = "任意の値" '
```

Oracle Audit Vault and Database Firewallで参照した場合

監査ポリシー作成例

- データベースのすべての操作を対象

```
CREATE AUDIT POLICY all_actions ACTIONS ALL;
AUDIT POLICY all_actions ;
```

- HRユーザーのEMPPLOYEES表へのすべての操作を対象

```
CREATE AUDIT POLICY all_actions_emp ACTIONS ALL ON HR.EMPLOYEES;
AUDIT POLICY all_actions_emp;
```

- (条件)ローカル接続の場合に、特定のシステム権限、表へのアクセスを対象

```
CREATE AUDIT POLICY custom_audit
  PRIVILEGES  SELECT ANY TABLE, UPDATE ANY TABLE, DELETE ANY TABLE
  ACTIONS ALL ON SCOTT.EMP, ALL ON SCOTT.DEPT
  WHEN 'SYS_CONTEXT("USERENV","IP_ADDRESS") IS NULL'
  EVALUATE PER STATEMENT;
AUDIT POLICY custom_audit;
```

ひとつのビューからすべての監査ログにアクセス

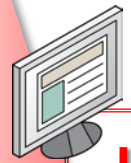
UNIFIED_AUDIT_TRAIL

監査ログ基本情報

イベント時間、SessionID、SCN、DBID、
OSユーザ名、DBユーザ名
IPアドレス、ホスト名、プログラム名
アクション名、オブジェクト名、SQL文、バインド値
クライアント識別子、アプリケーションコンテキスト等

監査ログ拡張情報

ファイングレイン監査
Data Pump, SQL*Loader
Recovery Manager
Database Vault
Oracle Label Security, Real Application Security



UNIFIED_AUDIT_TRAIL

UNIFIED_AUDIT_TRAIL の主な項目 1/3

	説明	例
AUDIT_TYPE	監査タイプ	Standard, Fine Grained Audit Database Vault RMAN AUDIT Data Pump
SESSIONID	監査セッションに割り当てられる識別ID	650971863
OS_USERNAME	OSユーザ名 (接続クライアント)	oracle
USERHOST	ホスト名 (接続クライアント)	client001.jp.oracle.com
TERMINAL	端末の識別子 (接続クライアント)	pts/1
INSTANCE_ID	インスタンス番号	1
DBID	Databaseの識別ID	1417811312
AUTHENTICATION_TYPE	セッションユーザの認証タイプ	(TYPE=(DATABASE));(CLIENT ADDRESS=((ADDRESS=(PROTOCOL=tcp)(HOST=10.185.146.20)(PORT=50713)));
DBUSERNAME	データベースユーザ名	SCOTT

UNIFIED_AUDIT_TRAIL の主な項目 2/3

	説明	例
CLIENT_PROGRAM_NAME	クライアントプログラム名	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)
DBLINK_INFO	データベースリンク情報	SOURCE_GLOBAL_NAME=dblink_src_global_name…..
EVENT_TIMESTAMP	イベント時間 (UTC)	13-04-25 15:16:45.513780000
ACTION_NAME	アクション名	SELECT,INSERT, UPDATE,EXECUTE…. CREATE USER, LOGOFF,LOGON…..
RETURN_CODE	エラー番号 (ORA-XXXXX)	1031
OS_PROCESS	OSのプロセス番号	30422
SCN	System Change Number	5742707
OBJECT_SCHEMA	アクションによって影響するスキーマ名	HR
OBJECT_NAME	アクションによって影響するオブジェクト名	EMPLOYEES
SQL_TEXT	実行されたSQL	select count(*) from emp where empno=:v1
SQL_BINDS	SQL_TEXTに含まれるバインド変数の値	#1(7):1001
APPLICATION_CONTEXTS	アプリケーションコンテキスト値	custno_ctx
CLIENT_IDENTIFIER	セッションに設定されたクライアント識別子	app001

UNIFIED_AUDIT_TRAIL の主な項目 3/3

	説明	例
UNIFIED_AUDIT_POLICIES	監査ログの出力に起因したポリシー名	ORA_SECURECONFIG
FGA_POLICY_NAME	監査ログの出力に起因したFGA名	FGA_EMP_POLICY
DV_XXXXX (略)	Database Vaultに関連するログ情報	
RMAN_XXXX (略)	Recovery Managerに関連するログ情報	
DP_XXXX (略)	Data Pumpに関連するログ情報	
DIRECT_PATH_NUM_COLUMNS_LOADED	SQL*Loader Direct Path Load に関連するログ情報	
OLS_XXXX (略)	Oracle Lable Securityに関連するログ情報	
XS_XXXX (略)	Oracle Real Application Securityに関連するログ情報	

監査ポリシーの無効化～削除

- 監査ポリシーの無効化

```
NOAUDIT POLICY ポリシー名
```

- 監査ポリシーの削除

```
DROP AUDIT POLICY ポリシー名
```

```
NOAUDIT POLICY all_actions_emp;
監査取消しが成功しました。

drop audit policy all_actions_emp;
監査ポリシーが削除されました。
```

定義済みのデフォルトポリシー

ORA_SECURECONFIG

PRIVILEGES	ALTER ANY TABLE CREATE ANY TABLE DROP ANY TABLE	CREATE ANY PROCEDURE DROP ANY PROCEDURE ALTER ANY PROCEDURE,	GRANT ANY PRIVILEGE GRANT ANY OBJECT PRIVILEGEGRANT ANY ROLE
	AUDIT SYSTEM	CREATE EXTERNAL JOB CREATE ANY JOB	CREATE ANY LIBRARY
	EXEMPT ACCESS POLICY	CREATE USER DROP USER	ALTER DATABASE ALTER SYSTEM
	CREATE PUBLIC SYNONYM DROP PUBLIC SYNONYM	CREATE ANY SQL TRANSLATION PROFILE ALTER ANY SQL TRANSLATION PROFILE DROP ANY SQL TRANSLATION PROFILE	TRANSLATE ANY SQL
	EXEMPT REDACTION POLICY ADMINISTER KEY MANAGEMENT	PURGE DBA_RECYCLEBIN	LOGMINING
ACTIONS	ALTER USER	CREATE ROLE ALTER ROLE DROP ROLE SET ROLE	CREATE PROFILE ALTER PROFILE DROP PROFILE
	CREATE DATABASE LINK ALTER DATABASE LINK DROP DATABASE LINK	LOGON LOGOFF	CREATE DIRECTORY DROP DIRECTORY

強制的に監査されるユーザー、コマンド

管理者のアクセス、監査設定の変更履歴はデフォルト監査

- ユーザー
  - SYS, SYSDBA, SYSOPER
  - SYSASM, SYSBACKUP, SYSDG, SYSKM
- コマンド
  - CREATE AUDIT POLICY
  - ALTER AUDIT POLICY
  - DROP AUDIT POLICY
  - AUDIT, NOAUDIT
  - EXECUTE DBMS_FGA, DBMS_AUDIT_MGMT
  - ALTER TABLE (AUDSYSユーザーが保有する表)

RMANイベントの監査

- RMANコマンドの実行

```
$ rman target /
RMAN> backup tablespace users;
RMAN> restore tablespace users;
RMAN> recover tablespace users;
```

- UNIFIED_AUDIT_TRAILのRMAN列を参照

```
SELECT event_timestamp,action_name,rman_operation,rman_object_type FROM unified_audit_trail
WHERE rman_operation IS NOT NULL;

EVENT_TIMESTAMP      ACTION_NAME          RMAN_OPERATION       RMAN_OBJECT_TYPE
-------------------------------------------------------------------------------------------------------------------------
13-02-14 02:19:26    RMAN ACTION          Backup               DF Full
13-02-14 02:19:26    RMAN ACTION          Restore              DF Full
13-02-14 02:19:26    RMAN ACTION          Recover              DF Full
```

Datapumpイベントの監査

- Datapumpのポリシーの設定

```
SQL> CREATE AUDIT POLICY audit_dp_all_pol ACTIONS COMPONENT=DATAPUMP ALL;
SQL> AUDIT POLICY audit_dp_all_pol;
```

- EXPORTの実行

```
$ expdp scott/tiger dumpfile=scott_tables tables=emp,dept directory=dp_dir

Export: Release 12.1.0.1.0 - Production on 木 2月 14 11:44:52 2013
Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.
接続先: Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real Application Testing  and Unified Auditing options
"SCOTT"."SYS_EXPORT_TABLE_01"を起動しています:
     scott/******** dumpfile=scott_tables tables=emp,dept directory=dp_dir
・・
. . "SCOTT"."DEPT"                          6 KB       4行がエクスポートされました
. . "SCOTT"."EMP"                       8.671 KB      12行がエクスポートされました
マスター表"SCOTT"."SYS_EXPORT_TABLE_01"は正常にロード/アンロードされました
******************************************************************************
```

Datapumpイベントの監査

- UNIFIED_AUDIT_TRAILのDP列を参照

```
SELECT event_timestamp,dp_text_parameters1,dp_boolean_parameters1
FROM unified_audit_trail WHERE dp_text_parameters1 is not null

EVENT_TIMESTAMP
---------------------------------
13-02-14 11:44:56

DP_TEXT_PARAMETERS1
--------------------------------------------------------------------------------------------------------------------------------------------
MASTER TABLE:  "SCOTT"."SYS_EXPORT_TABLE_01" , JOB_TYPE: EXPORT, METADATA_JOB_MODE:
TABLE_EXPORT,JOB VERSION: 12.0.0.0.0, ACCESS METHOD: AUTOMATIC,
DATA OPTIONS: 0, DUMPER DIRECTORY: NULL  REMOTE LINK: NULL, TABLE EXISTS: NULL,
PARTITION OPTIONS: NONE

DP_BOOLEAN_PARAMETERS1
--------------------------------------------------------------------------------------------------------------------------------------------
MASTER_ONLY: FALSE, DATA_ONLY: FALSE, METADATA_ONLY: FALSE,
DUMPFILE_PRESENT: TRUE, JOB_RESTARTED: FALSE
```

SQL*Loader Direct Path Load イベントの監査

- SQL*Loader Direct Path Loadのポリシーの設定

```
SQL> CREATE AUDIT POLICY audit_sqlldr_load_pol ACTIONS COMPONENT=DIRECT_LOAD LOAD;
SQL> AUDIT POLICY audit_sqlldr_load_pol ;
```

- SQL*Loader Direct Path Loadの実行

```
$ sqlldr userid=hr/hr control=emp.ctl data=emp.csv direct=y
```

- UNIFIED_AUDIT_TRAILを参照

```
select event_timestamp,audit_type,dbusername,action_name,object_schema,object_name,sql_text,
direct_path_num_columns_loaded from unified_audit_trail where audit_type='Direct path API‘

EVENT_TIMESTAMP    AUDIT_TYPE       ACTION_NAME  OBJECT_SCHEMA      OBJECT_NAME
-------------------------------------------------------------------------------------------------------------------------------
13-02-14 13:05:31        Direct path API      LOAD              HR                          EMP

SQL_TEXT
-------------------------------------------------------------------------------------------------------------------------------
INSERT /*+ SYS_DL_CURSOR */ INTO "HR"."EMP" ("EMP_ID","EMP_NAME") VALUES (NULL,NULL)

DIRECT_PATH_NUM_COLUMNS_LOADED
----------------------------------------------------------
2
```

Database Vaultイベントの監査

- Database Vaultのポリシーの設定
  - 対象オブジェクト: レルム、ルールセット、ファクター
  - アクセス違反、成功等

```
SQL> CREATE AUDIT POLICY audit_dv  ACTIONS COMPONENT=DV Realm Violation
          ON "HR Application";
SQL> AUDIT POLICY audit_dv;
```

- UNIFIED_AUDIT_TRAILのDV列を参照

```
SELECT dbusername,object_name,sql_text,dv_action_name FROM unified_audit_trail
 WHERE db_return_code <> 0;

DBUSERNAME          OBJECT_NAME         SQL_TEXT                DV_ACTION_NAME
-------------------------------------------------------------------------------------------------------------------------
HR                  EMPLOYEE            select * from hr.employee  Realm Violation Audit
```

Mixed Mode Auditing

下位互換性サポート

- 12cのデータベースは、従来のAuditとUnified Auditingの両方が使用可能
- インストール時は、いずれも使用可能なMixed モードで動作
- Mixed モードの場合は、AUDIT_SYS_OPERATIONのSYSDBAのログファイルは従来通りOSのディレクトリ上に出力される
- また、RMANやDatapump等のユーティリティのログは、Unified Auditingに統合されない

監査モードの確認

```
SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';
----------------------------------------
TRUE   -->  Unified Auditing
FALSE -->  Mixed Mode
```

Mixed Mode Auditing

下位互換性サポート

- Mixedモードは、Unified Auditingの対象範囲がAudit Policyに限定されているので、以下の手順で完全に切り替えることを推奨

1. データベースのシャットダウン、リスナーの停止
2. cd $ORACLE_HOME/rdbms/lib
3. make -f ins_rdbms.mk uniaud_on ioracle ORACLE_HOME=$ORACLE_HOME
4. リスナー、データベースの起動

- 従来のAuditだけ使用したい場合は、Mixedモードですべての Unified Auditポリシーを無効化し、audit_trailパラメータとAuditコマンドで設定する

追加されたAUDIT専用ロール

データベース利用者と監査の権限分離

従来　自分の保有するオブジェクトに対して自由に監査設定をすることが出来る

12c　明示的にAUDITロールを持たない限り、監査ポリシーを設定・参照できない

AUDIT_ADMIN	Unified Auditing, ファイングレイン監査の作成 AUDIT, NOAUDIT文の実行 UNIFIED_AUDIT_TRAILの参照 監査ログの管理
AUDIT_VIEW	UNIFIED_AUDIT_TRAILの参照

インターバルを過ぎた監査ログをパージ

DBMS_AUDIT_MGMTパッケージ

- 基準日となる日時を指定

```
BEGIN
  DBMS_AUDIT_MGMT.SET_LAST_ARCHIVE_TIMESTAMP(
    AUDIT_TRAIL_TYPE           =>  DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    LAST_ARCHIVE_TIME          =>  '2013-02-15 10:00:00.00');
END;
```

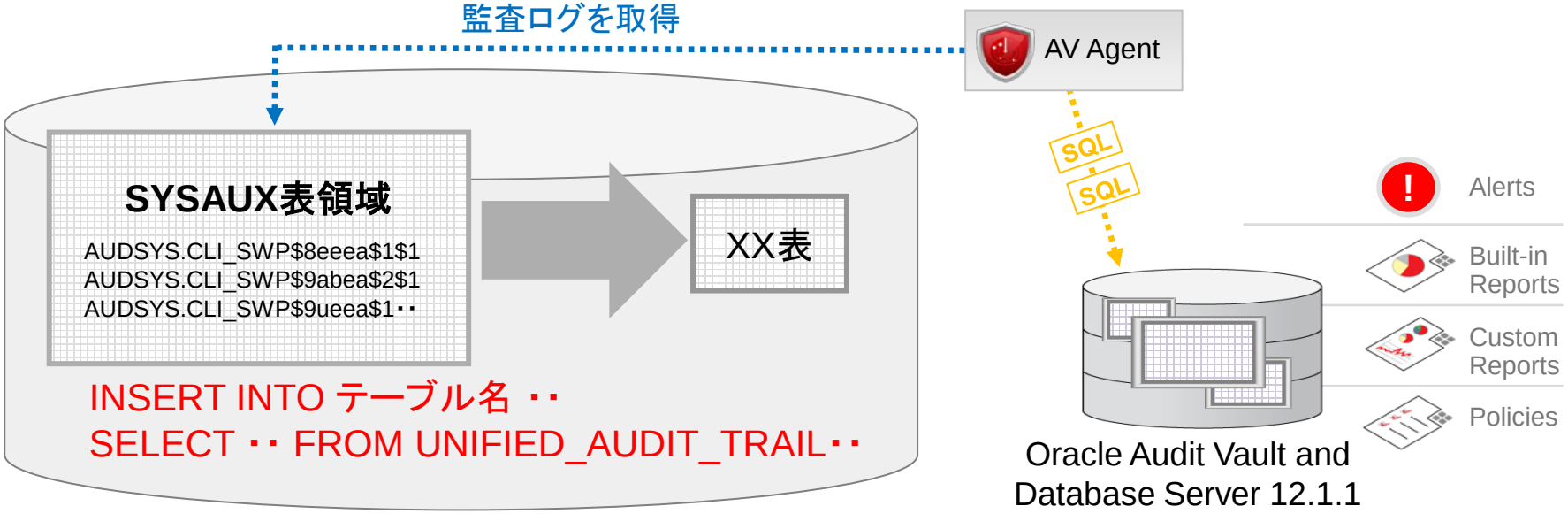
- 2週間(336H)を過ぎた監査ログをパージするジョブを作成

```
BEGIN
  DBMS_AUDIT_MGMT.CREATE_PURGE_JOB (
    AUDIT_TRAIL_TYPE           => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    AUDIT_TRAIL_PURGE_INTERVAL          => 336,
    AUDIT_TRAIL_PURGE_NAME              => 'Audit_Trail_Purge_Job',
    USE_LAST_ARCH_TIMESTAMP             => TRUE);
END;
```

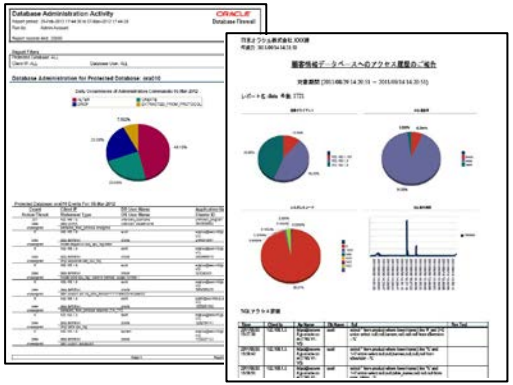
監査ログのアーカイブ

パージする前に監査ログをバックアップする方法

- UNIFIED_AUDIT_TRAILビューで別表として抽出し、EXPDP等で取り出す
- Oracle Audit Vault and Database Firewallにアーカイブする (12.1.1対応)



AVDFなら詳細な検索、レポートの自動配信が可能



Activity Overview Report

実行　アクション

イベント時間が過去の24時間内にある

イベント時間	ターゲット・オブジェクト	ユーザー名	クライアントIP	クライアント・プログラム	コマンド・テキスト
2013/01/02 1:01:56	DUAL	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	SELECT DECODE('A','A','1','2') FROM DUAL
2013/01/02 1:01:56	DUAL	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	SELECT USER FROM DUAL
2013/01/02 1:00:38	DUAL	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	SELECT DECODE('A','A','1','2') FROM DUA
2013/01/02 1:00:38	DUAL	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	SELECT USER FRO
2013/01/02 1:02:05	EMP	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	select * from emp
2013/01/02 1:02:05	EMP	scott	192.168.1.3	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)	select * from emp

Event

Server Time	12/31/2013 10:54:45 PM
Event Time	12/31/2013 10:52:05 PM
User Name	scott
Event Status	statement fail
Error Code	942
Error Message	ORA-00942: 表またはビューが存在しません。
Event Name	statement
Command Class	SELECT
Action Taken	pass
Threat Severity	undefined
Log Cause	unseen
Location	Network

Client/User Information

OS User Name	oracle
Client Host Name	
Client IP	192.168.1.3
Network Connection	192.168.1.3:58691,192.168.1.5:1521
Client Program	sqlplus@secvm3.jp.oracle.com (TNS V1-V3)

Statement

Command Text	select * from emp where empno = :1
Command Param	#1(4):7566

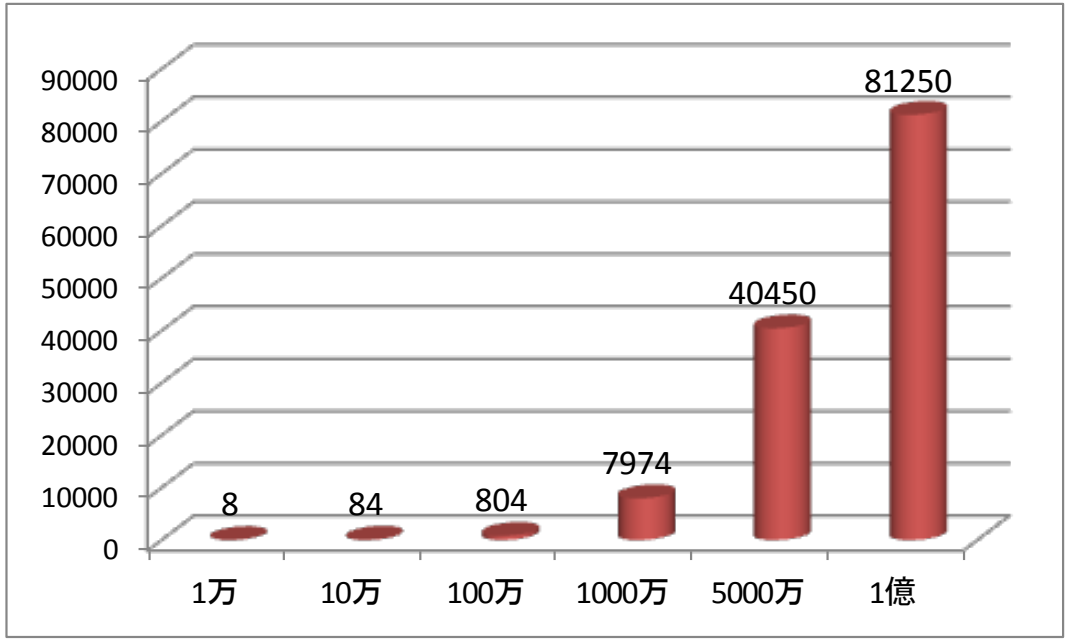
Target

Target Type	SESSION
Target Object	EMP
Target Owner	SCOTT

- 収集した監査ログから、条件に応じて(イベント時間 クライアント情報、SQLコマンド等)自由検索し抽出
- フィルタやソート、ハイライト、チャートなど書式の変更
- HTML、CSV形式での出力、レポートの自動配信

Unified Auditingに生成されるログのサイズ

SYSAUX表領域のサイズ



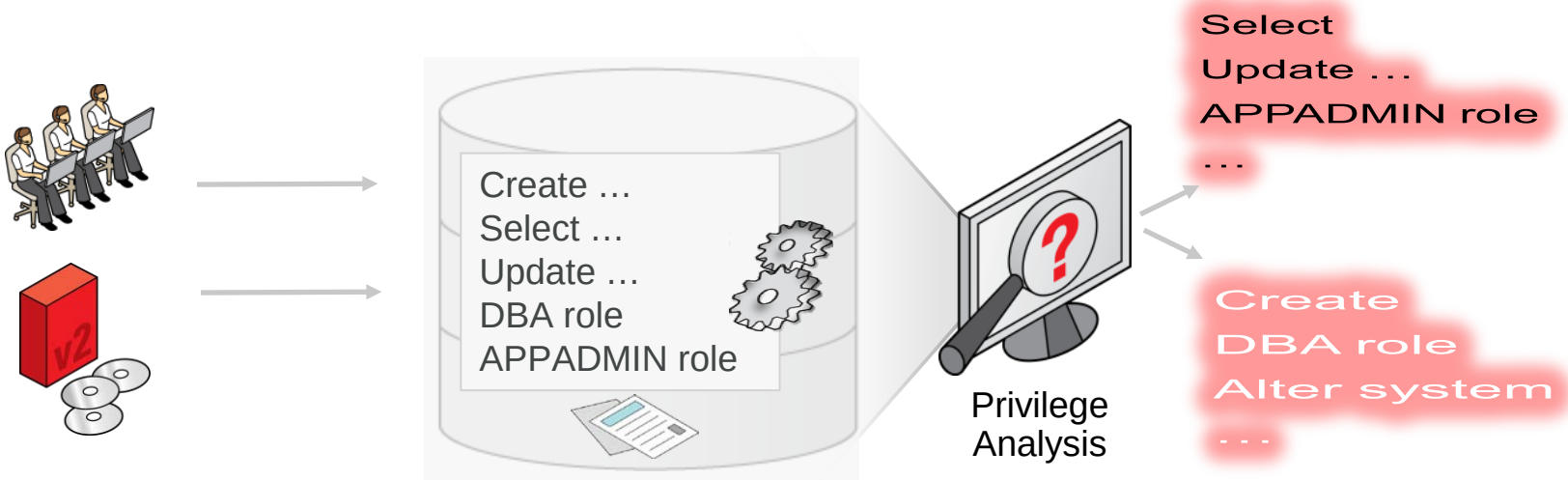
レコード数	AVSYS (MB)
1万	8
10万	84
100万	804
1000万	7974
5000万	40450
1億	81250

監査ログが1万レコードごとに、
SYSAUX表領域がおよそ8MB増加
1億レコードで約80GBを消費

- **権限管理 新機能**
  - **Privilege Analysis**

Privilege Analysis

不正アクセスの原因となる過度な権限付与を検出



- ユーザーやロールに付与されたシステム権限、オブジェクト権限の使用・未使用を洗い出してレポーティング
- アプリケーションや開発者・管理者に本当に必要する権限のみを付与
- 最小権限の原則を実現し、不正アクセスの未然防止に

権限分析の対象

実行されたシステム/オブジェクト権限の洗い出し

- ロール
  - 指定したロールの権限の使用状況を分析する (複数指定可)
- 条件指定
  - 指定した条件に適合する場合、権限の使用状況を分析する
    (特定のユーザーやアプリケーション等)
- ロール＋条件指定
  - 指定したロールかつ条件に適合する場合、権限の使用状況分析する
- データベース
  - データベース内のすべての権限の使用状況を分析する
    (SYSユーザーを除く)

権限分析の手順

DBMS_PRIVILEGE_CAPTUREによるキャプチャの開始

- 分析ポリシーの作成

DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE	
name	ポリシー名
description	説明(任意)
type	いずれか一つ選択 DBMS_PRIVILEGE_CAPTURE.G_DATABASE DBMS_PRIVILEGE_CAPTURE.G_ROLE DBMS_PRIVILEGE_CAPTURE.G_CONTEXT DBMS_PRIVILEGE_CAPTURE.G_ROLE_AND_CONTEXT
roles	Ex) role_name_list('role1', 'role2')
condition	Ex) SYS_CONTEXT("USERENV", "SESSION_USER")='SCOTT'

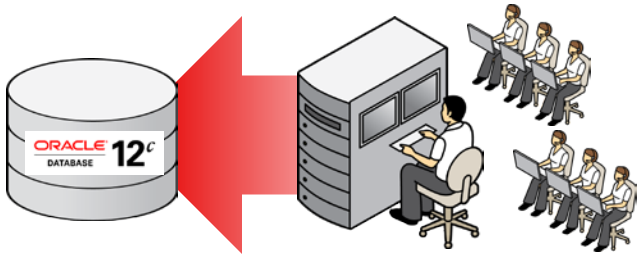
- 分析ポリシーの有効化

DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ('ポリシー名')

権限分析の手順

キャプチャの停止～レポーティング

キャプチャ実行中



- 分析ポリシーの無効化

```
EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE (‘ポリシー名)
```

- 分析レポートの作成 (実行後、専用のビューで分析結果を参照可能)

```
EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT (‘ポリシー名')
```

- 分析レポートの削除 (作成したレポートの情報を含めて削除される)

```
EXEC DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE   (‘ポリシー名‘)
```

Privilege Analysisの専用ビュー

結果テーブル	説明
DBA_USED_PRIVS DBA_UNUSED_PRIVS	すべての使用/未使用の権限。 （システム権限、ユーザー権限、オブジェクト権限とPUBLIC権限を含める）
DBA_USED_OBJPRIVS DBA_UNUSED_OBJPRIVS DBA_USED_OBJPRIVS_PATH DBA_UNUSED_OBJPRIVS_PATH	すべての使用/未使用のオブジェクト権限。 「PATH」を付けてのテーブルは権限付与もリストする。
DBA_USED_SYSPRIVS DBA_UNUSED_SYSPRIVS DBA_USED_SYSPRIVS_PATH DBA_UNUSED_SYSPRIVS_PATH	すべての使用/未使用のシステム権限。 「PATH」を付けてのテーブルは権限付与もリストする
DBA_USED_PUBPRIVS	すべての使用したPUBLIC権限。
DBA_USED_USERPRIVS DBA_UNUSED_USERPRIVS DBA_USED_USERPRIVS_PATH DBA_UNUSED_USERPRIVS_PATH	すべての使用/未使用のユーザー権限。 「PATH」を付けてのテーブルは権限付与もリストする

例) ユーザーがAny権限を使用していないか調査

```
BEGIN
 DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
  name          => 'ANY_priv_analysis_pol',
  type          => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
  condition     => 'SYS_CONTEXT("USERENV", "SESSION_USER")="APP_USER"');
END;/

EXEC DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ('ANY_priv_analysis_pol');

    -----処理実行-----

EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ('ANY_priv_analysis_pol');

EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ('ANY_priv_analysis_pol');
```

調査対象は、APP_USER

```
SELECT USERNAME, SYS_PRIV, OBJECT_OWNER, OBJECT_NAME FROM DBA_USED_PRIVS;

USERNAME   SYS_PRIV                  OBJECT_OWNER  OBJECT_NAME
---------- -------------------------- ------------ ------------
APP_USER   SELECT ANY TABLE           HR           EMPLOYEES
APP_USER   CREATE SESSION
APP_USER                              SYS          ORA$BASE
APP_USER                              SYS          DUAL
```

APP_USERが
SELECT ANY TABLE権限で
HRユーザーのEMPLYEES表にで
アクセスした履歴が判明

-> SELECT ANY TABLEは必要ないのでは？

例) DBAロールの使用状況を調査する

```
BEGIN
 DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
  name       => 'dba_role_analysis',
  type       => DBMS_PRIVILEGE_CAPTURE.G_ROLE,
  roles      => role_name_list('dba'));
END;/
EXEC DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ('dba_role_analysis');
```

調査対象は、DBAロール

-----処理実行-----

```
EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ('dba_role_analysis');
EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ('dba_role_analysis');
```

SCOTTユーザーが、
DROP ANY TABLE権限を使用して
(DBA-EM - >OLAP_DBAロール配下)
HRユーザーのTEST表を削除している

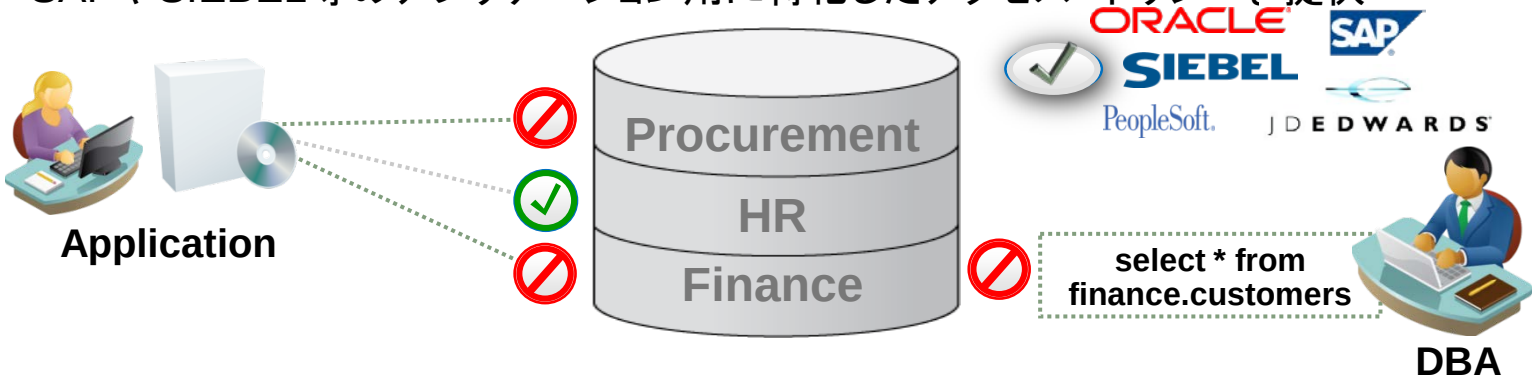
```
SELECT USERNAME, USED_ROLE,SYS_PRIV,PATH FROM DBA_USED_SYSPRIVS_PATH;

USER     USED_ROLE                 SYS_PRIV              PATH
---------- -------------------------  ----------------------  ----------------------
SCOTT    OLAP_DBA                  DROP ANY TABLE     SYS.GRANT_PATH(SCOTT,DBA,OLAP_DBA)

SELECT USERNAME, SYS_PRIV, OBJECT_OWNER, OBJECT_NAME FROM DBA_USED_PRIVS

USERNAME    SYS_PRIV                  OBJECT_OWNER  OBJECT_NAME
---------------  ---------------------------  ----------------------  ----------------------
SCOTT        SELECT ANY TABLE     HR                 TEST
```

Database Vault 新機能

Oracle Database Vault

特権ユーザー管理

- データベース内の特権ユーザー(SYSユーザー、DBAロール等)を強制的にアクセス制御し、
  特権ユーザーといえどもアプリケーションのデータにアクセスできない
- アプリケーションをバイパスしたアクセスでもデータベース側でデータを保護
- ユーザーやIPアドレスなどのクライアント情報や曜日・時間を組み合わせた細かなアクセスポリシーの作成が可能
- SAPやSIEBEL等のアプリケーション用に特化したアクセス・ポリシーを提供



簡単になった起動設定

必要なコンポーネントはデフォルトでインストール済み

- DVの管理者ユーザーとアカウント管理者を作成 (SYSユーザーで実行)

```
GRANT CREATE SESSION TO dbv_owner IDENTIFIED BY password;
GRANT CREATE SESSION TO dbv_acctmgr IDENTIFIED BY password;
BEGIN
 DVSYS.CONFIGURE_DV (
   dvowner_uname         => 'dbv_owner',
   dvacctmgr_uname       => 'dbv_acctmgr');
 END;
```

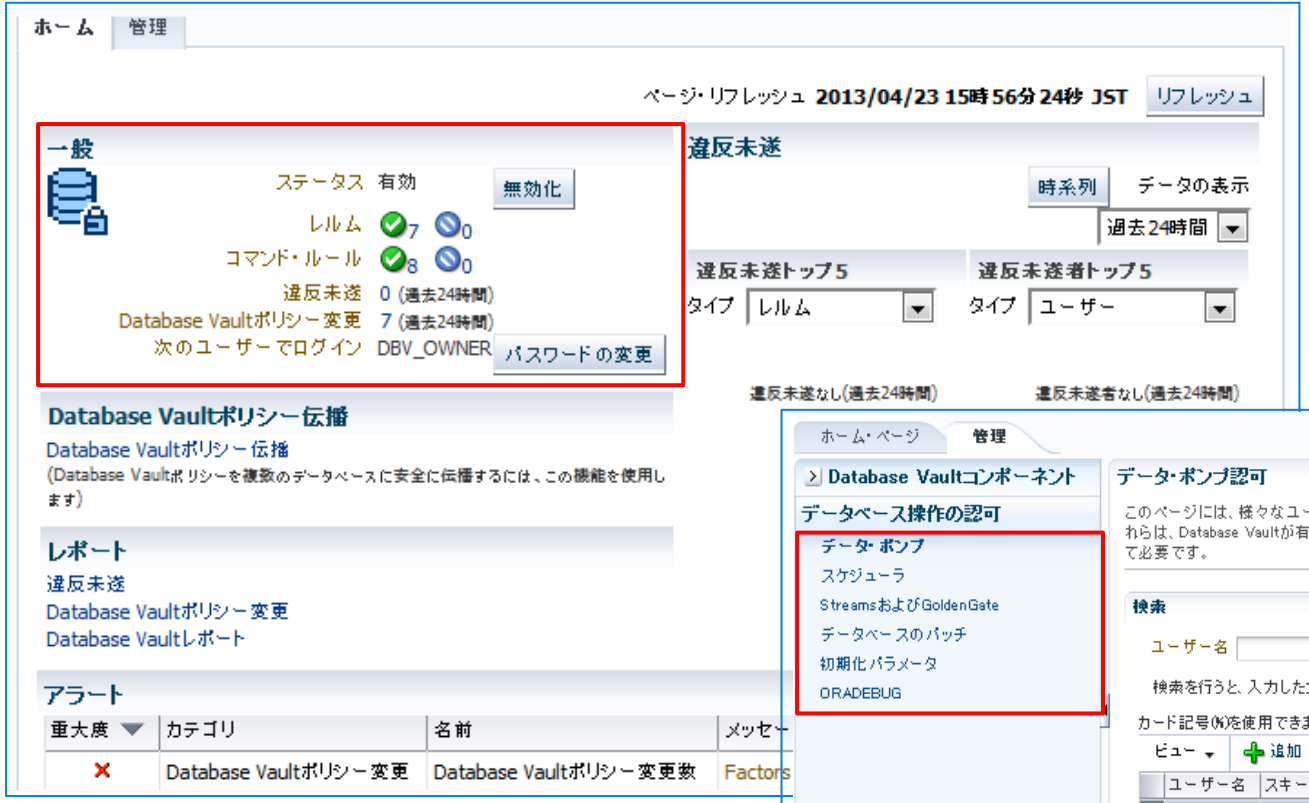
- DVの有効化 (DV管理者で実行後、再起動必要)

```
EXEC DVSYS.DBMS_MACADM.ENABLE_DV;
```

- DVの無効化 (DV管理者で実行後、再起動必要)

```
EXEC DVSYS.DBMS_MACADM.DISABLE_DV;
```

必要な設定はすべてEnterprise Managerに統合



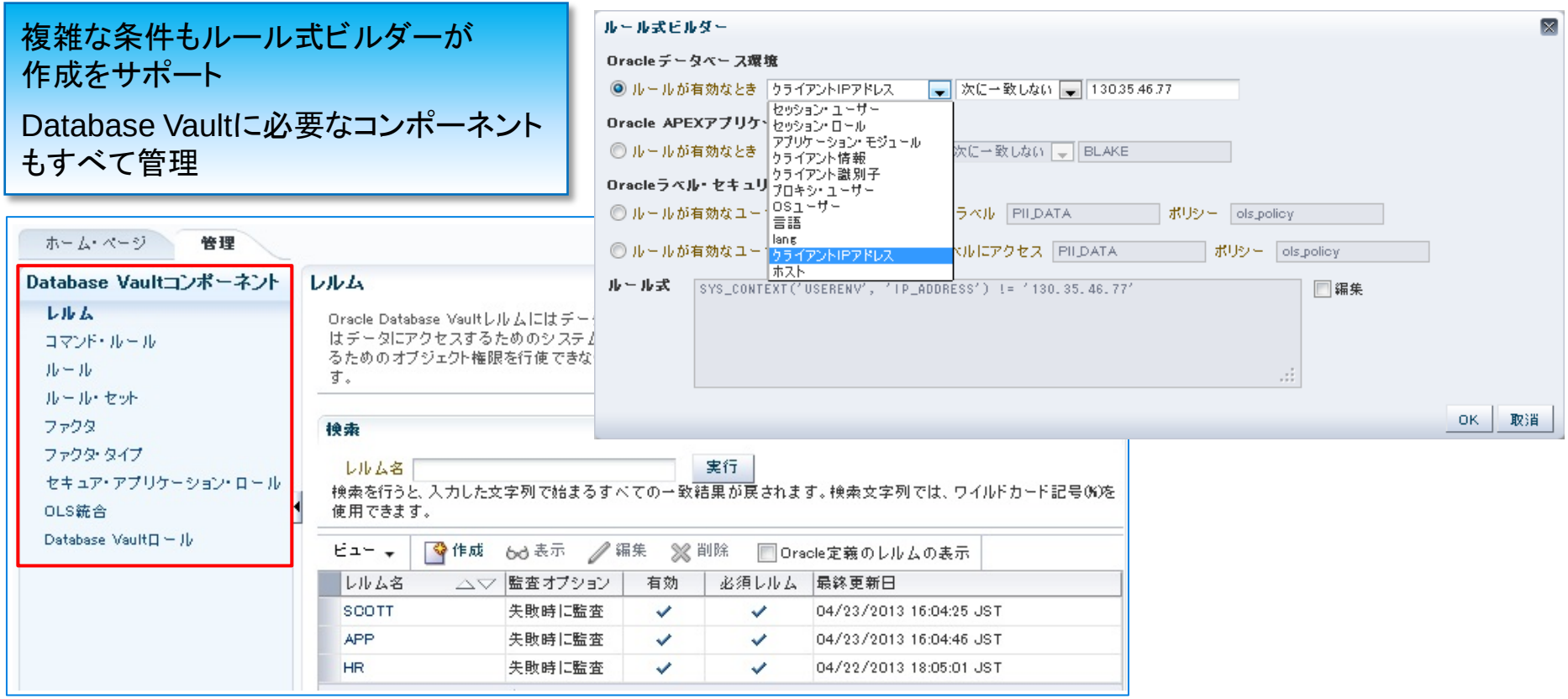
Database Vaultの起動状態
レルム・コマンドルール等の
ポリシーの反映状況

Datapumpやパッチ適用時に
必要なロールはプリセット済み

直感的&シンプルなインターフェースに進化

複雑な条件もルール式ビルダーが作成をサポート

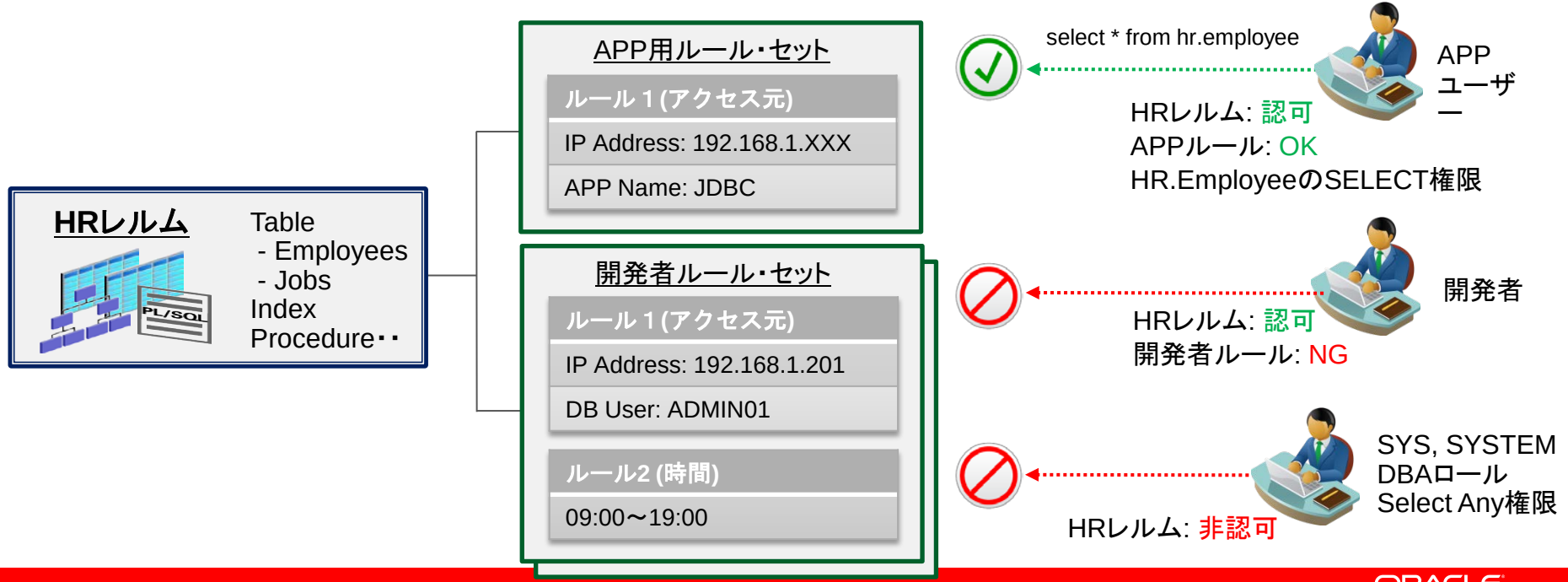
Database Vaultに必要なコンポーネントもすべて管理



特権ユーザーからのアクセスを遮断

レルムによって論理的にオブジェクトの守備範囲を定義

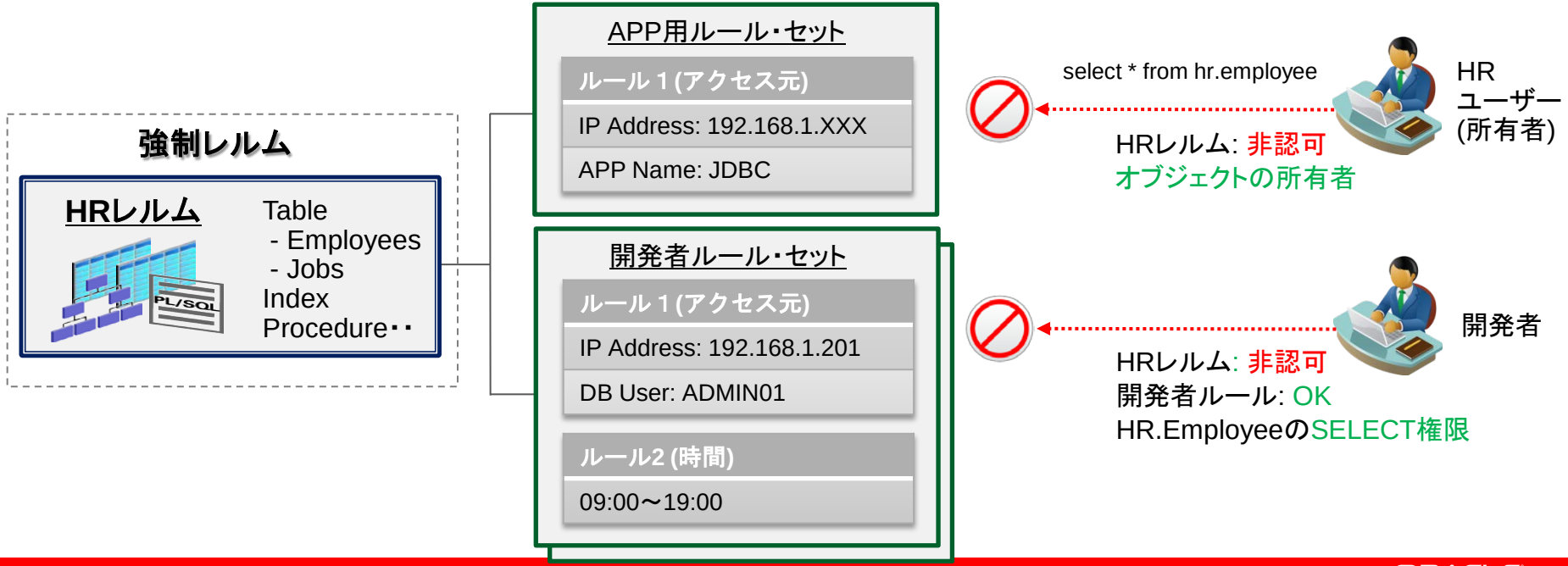
- レルム内のオブジェクトにアクセスするためには、
  レルムの認可、ルールの許可、オブジェクトへのアクセス権をすべて満たさなければならない



強制レルム

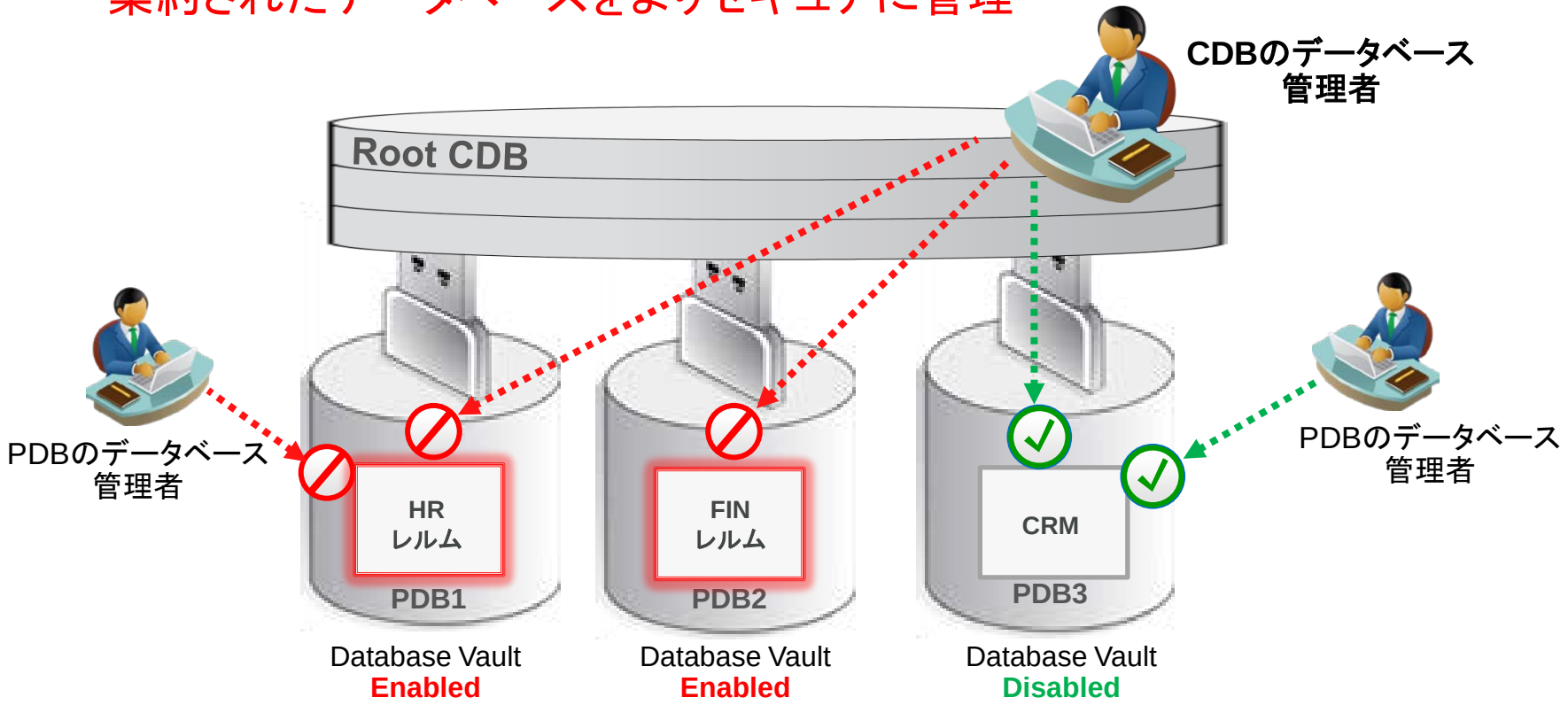
より厳格に、抜け漏れのないデフォルト遮断

- オブジェクトの所有者(作成者)、オブジェクト権限を与えられているユーザーでも、レルム認可されていない限りはアクセスできない



マルチテナント・アーキテクチャに対応

集約されたデータベースをよりセキュアに管理




SECURITY
INSIDE
OUT
Oracle Security Solutions
ORACLE


ORACLE®