

# 高可用性関連新機能

Oracle Database 23c新機能セミナー

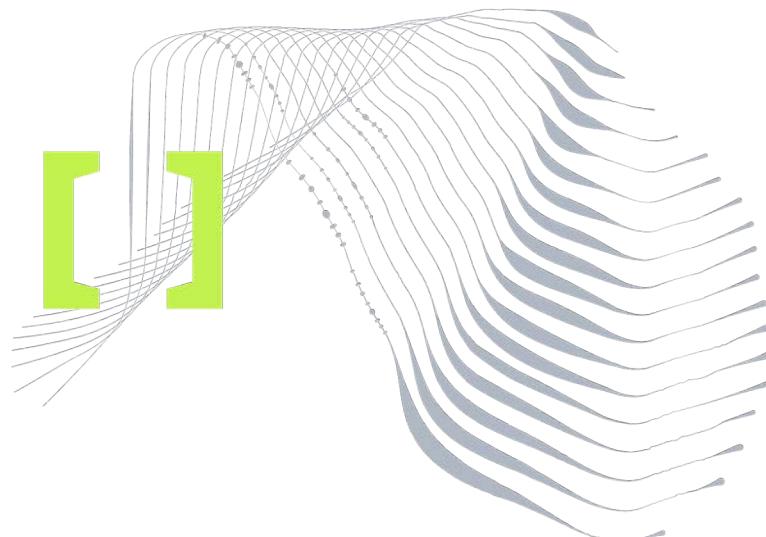
前半：宮本 拓弥 後半：百木 和美

日本オラクル株式会社  
2023年10月19日



# Agenda

1. Multitenant Architecture
2. Database Resident Connection Pool(DRCP)
3. 表領域管理
4. Backup & Recovery
5. Diagnosability



# Multitenant Architectureの新機能

- Hybrid read-only mode for pluggable databases
- Control PDB Open Order
- Read only users and sessions



# Multitenant Architectureの進化

ORACLE®  
Database 12c  
(Release 1)

- <12.1.0.1>**
  - Oracle Multitenantオプション
  - マルチテナント環境でのResource Managerサポート
- <12.1.0.2～>**
  - CDB再起動時のPDBのオープン・モードの保持
  - CREATE PLUGGABLE DATABASE文のUSER\_TABLESPACES句
  - PDBをクローニングする場合のデータの除外
  - PDBファイル用のデフォルトのOracle Managed Filesファイル・システム・ディレクトリまたはOracle ASMディスク・グループ
  - 非CDBのクローニングによるPDBの作成
  - PDBのロギング属性
  - CREATE PLUGGABLE DATABASE文のSTANDBYS句
  - すべてのPDB間でのユーザーが作成した表およびビューの問合せ

ORACLE®  
Database 12c  
(Release 2)

- アプリケーション・コンテナ
- 単一のCDBでの4096PDB(EXA)
- PDBでの異なる文字セットの使用
- 別のCDBへのPDBの再配置
- プロキシPDB
- ホットPDBクローニング
- PDB作成時のサービスの名前変更
- CDBのコンテナ特定サービスへ切替
- PDBのメモリー使用量の管理  
SGA\_MIN\_TARGET
- PDB単位でI/Oを制限
- PDBパフォーマンス・プロファイル
- Resource ManagerによるPDBの監視
- PDBアップグレード優先順位の設定
- CDBのUNDOモード
- PDBの作成の並列化
- アーカイブ・ファイルによるPDBのアンプラグとプラグ・イン
- PDBのリフレッシュ
- PDB作成時のデフォルト表領域の指定のサポートの改善
- PDB作成時のUSER\_TABLESPACES句拡張

ORACLE®  
Database 18c

- CDB フリート
- PDBスナップショット・カルーセル
- 論理パーティション化
- リフレッシュ可能なPDBスイッチャー
- ロックダウン・プロファイルの拡張機能
- DBCAの拡張機能
- 非CDBおよび再配置されたPDBの使用可能なバックアップ
- 別のCDBへのPDBのRMAN複製
- 計画メンテナンス中のセッションの再配置
- Oracle Data Guard環境でのPDBのコピー
- PDBレベルでのパラレル文のクエーリング
- 分割ミラーによるクローンPDB

19c ORACLE®  
Database

- PDBでのワークロードの取得およびリプレイ
  - PDBのADDM分析
  - インフラストラクチャ・データベース管理者向けのDatabase Vault操作の制御
  - 同じCDBでの複数のPDBシャードのサポート
  - PDBの自動再配置
  - DBCAを使用したリモートPDBのクローニング
  - リモートPDBの再配置
  - データ・ポンプ・インポートに対するクラウドのオブジェクト・ストアのサポート
  - CPU\_MIN\_COUNT初期化パラメータ
- **EE/SEとともにPDB3つまで無償**

21c ORACLE®  
Database

- PDBオープン時のリプレイ・アップグレード
- リプレイ・アップグレードを使用した、PDBとしての非CDBの採用
- Oracle Databaseとのネームスペース統合
- Oracle Cloudでの透過的アプリケーション・コンティニュイティ
- JOB\_QUEUE\_PROCESSES初期化パラメータの機能拡張
- MAX\_IDLE\_BLOCKER\_TIME初期化パラメータの機能拡張
- アプリケーションの同期のための拡張構文
- DBCA機能拡張(PDB単位のタイムゾーン設定)
- RAC構成におけるランク付け
- **非CDB構成がサポート対象外に**

統合・集約と一元管理の向上

可搬性とPDBの分離性向上への実装

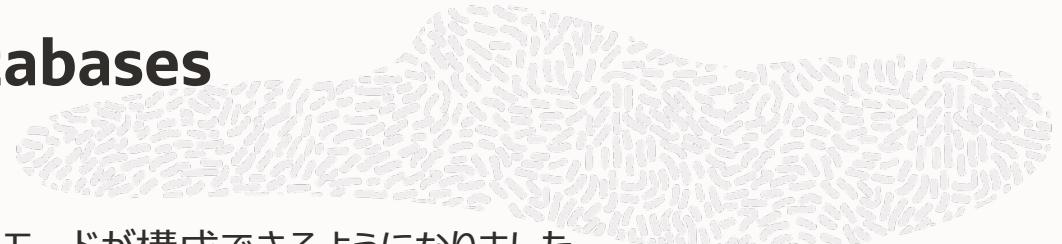
次世代基盤としての拡充

# Hybrid read-only mode for pluggable databases

- Hybrid read-only mode for pluggable databases
- Control PDB Open Order
- Read only users and sessions

# Hybrid read-only mode for pluggable databases

## 概要



プラガブル・データベース (PDB) をハイブリッド読み取り専用と呼ばれる新しいモードが構成できるようになりました。

### ハイブリッド読み取り専用モードについて

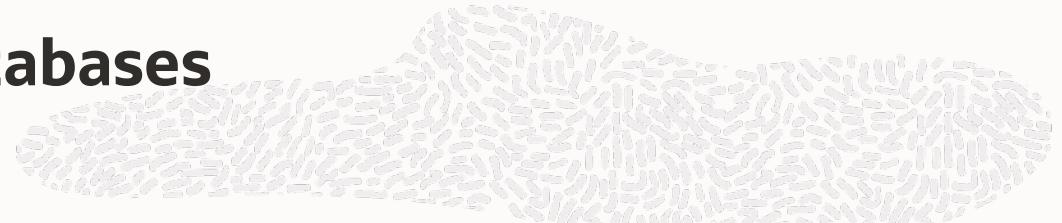
- PDBに接続しているユーザーに応じて、PDBを読み取り専用または読み取り/書き込みモードとして動作させることができます。
- CDB共通ユーザーの場合、PDBは読み取り/書き込みモードになります。
- PDBローカル・ユーザーの場合、PDBは読み取り専用モードに制限されます。
- `V$CONTAINER_TOPOLOGY`ダイナミック・ビューに新しい列`IS_HYBRID_READ_ONLY`が追加されました。
- `V$PDBS`ダイナミック・ビューの`OPEN_MODE`列の出力が変更されます。
  - ローカル・ユーザの場合は`READ ONLY`
  - 共通ユーザの場合は`READ WRITE`

### メリット

データベースやアプリケーションの管理者が、より高い権限を持つユーザを含むローカル・ユーザがPDBの継続的なメンテナンス作業を妨害するリスクなしに、オープンPDB用の安全なモードでアプリケーションにパッチを当てたり、メンテナンスできるようになります。

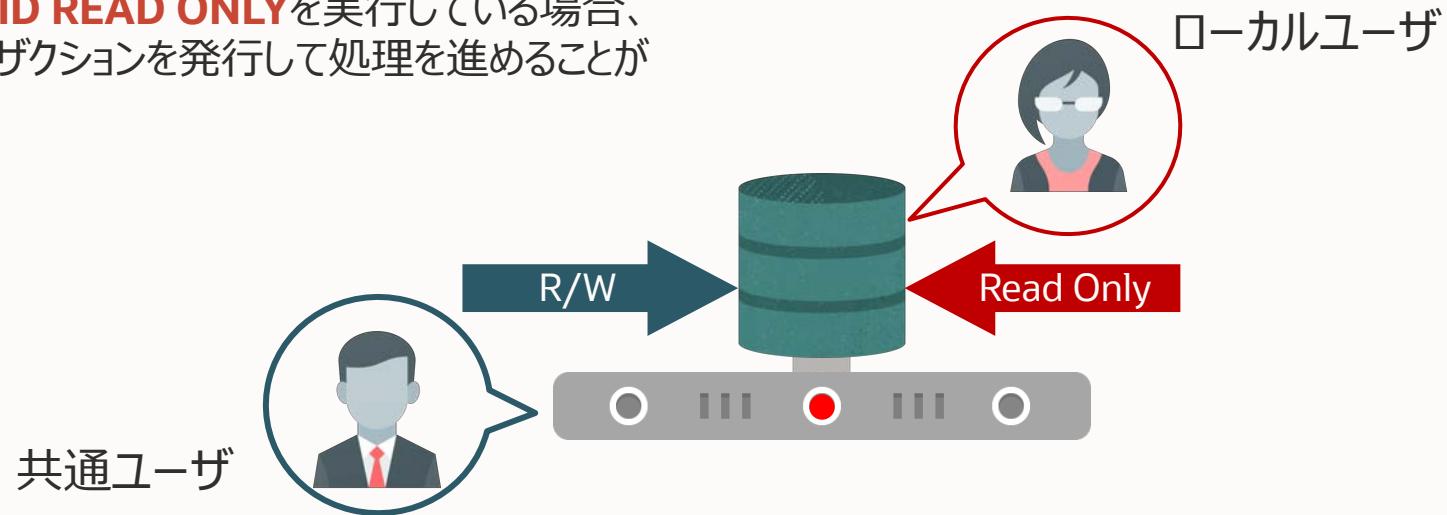
# Hybrid read-only mode for pluggable databases

各ユーザごとのオープン・モード



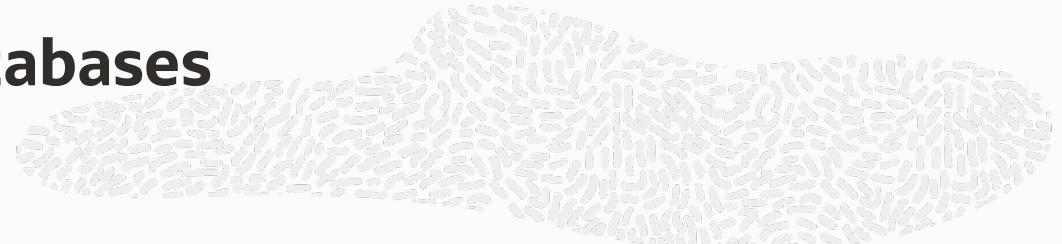
オープン・モード	CDB共通ユーザ	アプリケーション共通ユーザ	PDBローカル・ユーザ
Read Write	Read Write	Read Write	Read Write
Read Only	Read Only	Read Only	Read Only
Hybrid Read Only	Read Write	Read Only	Read Only

**ALTER PLUGGABLE DATABASE OPEN HYBRID READ ONLY**を実行している場合、PDBではCDB共通ユーザーがDMLとユーザー・トランザクションを発行して処理を進めることができ、これらのユーザーがREDOログを生成できます。



# Hybrid read-only mode for pluggable databases

## V\$CONTAINER\_TOPOLOGY ダイナミック・ビュー



```
desc V$CONTAINER_TOPOLOGY
```

名前	Nullかどうか	タイプ
----	----------	-----

```
-----
```

INSTANCE_NUMBER	NUMBER
-----------------	--------

CON_NAME	VARCHAR2(128)
----------	---------------

OPEN_MODE	VARCHAR2(10)
-----------	--------------

CPU_COUNT	NUMBER
-----------	--------

CON_ID	NUMBER
--------	--------

RESTRICTED	VARCHAR2(3)
------------	-------------

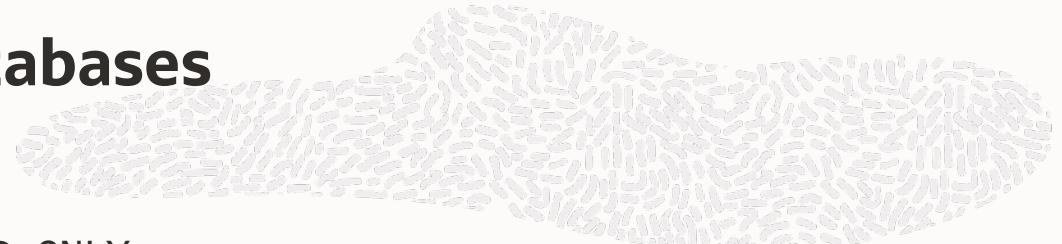
IS_HYBRID_READ_ONLY	VARCHAR2(3)
---------------------	-------------

```
SQL> select * from v$container_topology;
```

INSTANCE_NUMBER	CON_NAME	OPEN_MODE	CPU_COUNT	CON_ID	RES	IS_
2	CDB\$ROOT	READ WRITE	12	1	NO	NO
2	PDB\$SEED	READ ONLY	12	2	NO	NO
2	ORCL232BPDB1	READ WRITE	12	3	NO	NO
1	CDB\$ROOT	READ WRITE	12	1	NO	NO
1	PDB\$SEED	READ ONLY	12	2	NO	NO
1	ORCL232BPDB1	READ WRITE	12	3	NO	NO

# Hybrid read-only mode for pluggable databases

## Hybrid read-only modeの有効化



```
ALTER PLUGGABLE DATABASE ORCL232BPDB1 OPEN HYBRID READ ONLY;
```

```
SQL> show pdbs
```

CON_ID	CON_NAME	OPEN MODE	RESTRICTED
2	PDB\$SEED	READ ONLY	NO
3	ORCL232BPDB1	READ WRITE	NO

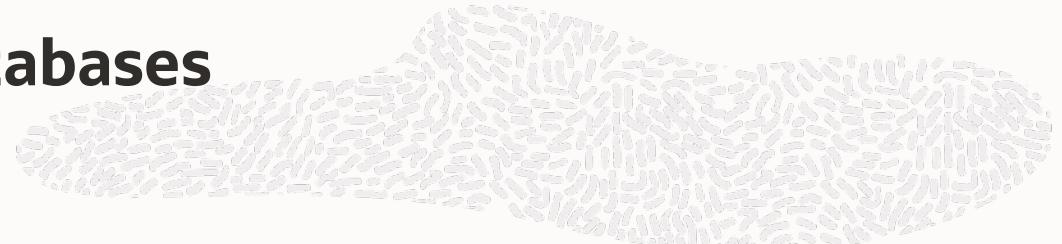
プラガブル・データベースが変更されました。

```
SQL> select * from v$container_topology;
```

INSTANCE_NUMBER	CON_NAME	OPEN_MODE	CPU_COUNT	CON_ID	RES	IS_
2	CDB\$ROOT	READ WRITE	12	1	NO	NO
2	PDB\$SEED	READ ONLY	12	2	NO	NO
2	ORCL232BPDB1	READ WRITE	12	3	NO	NO
1	CDB\$ROOT	READ WRITE	12	1	NO	NO
1	PDB\$SEED	READ ONLY	12	2	NO	NO
1	ORCL232BPDB1	READ WRITE	12	3	NO	YES

# Hybrid read-only mode for pluggable databases

ローカルユーザと共通ユーザで検証



## 1.ローカルユーザ(scott)で更新

インスタンス1のORCL232BPDB1へscottで接続

```
SQL> conn scott/tiger@ptvm07/ORCL232BPDB1
```

接続されました。

```
SQL> update emp set sal = 1200 where empno=7369;
```

```
update emp set sal = 1200 where empno=7369
```

\*

行1でエラーが発生しました。:

ORA-16000: データベースまたはプラガブル・データベースは読み取り専用アクセスでオープンされています

## 2.共通ユーザ(system)で更新

インスタンス1のORCL232BPDB1へsystemで接続

```
SQL> conn system/oracle@ptvm07/ORCL232BPDB1
```

接続されました。

```
SQL> update scott.emp set sal = 1200 where empno=7369;
```

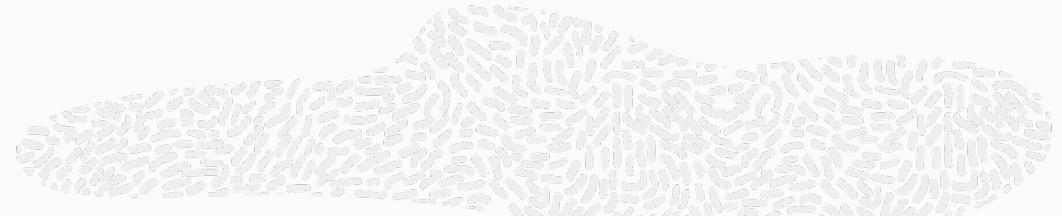
1行が更新されました。

# Control PDB Open Order

- Hybrid read-only mode for pluggable databases
- Control PDB Open Order
- Read only users and sessions

# Control PDB Open Order

## 概要



管理者は、各プラガブルデータベース（PDB）に対して、最も重要なPDBを最初に起動させる起動順序または優先順位を定義することができます。

以下のようなPDBのオープン順序とアップグレード順序で適用されます：

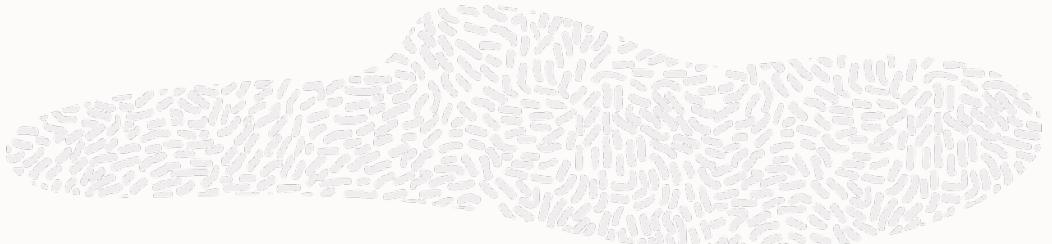
- CDBを開く際のPDB状態のリストア
- PDB OPEN ALLステートメントを使用する際のPDB状態の設定
- PDBデータベースのアップグレード操作の順序を設定する
- Active Data Guard（ADG）のスイッチオーバーまたはフェイルオーバーでPDBを起動する

## メリット

重要なPDBを重要度の低いPDBより先に起動・オープンすることができ、重要なアプリケーションが使用できるようになるまでの時間を短縮することができます。

# Control PDB Open Order

## コマンド例



ALTER PLUGGABLE DATABASE <PDBname> PRIORITY <value>

- <PDBname>にPDB名を入力する(必須)
- <value>に1から4096の整数を入力して優先度を設定する

### 注意：

PRIORITY句はOracle Database12cR2で導入され、PDBをアップグレードする際の優先度を指定できるようになった。

ALTER PLUGGABLE DATABASE xxx UPGRADE PRIORITY n;

### 実効例:

```
SQL> alter pluggable database ORCL232BPDB1 priority 4096;
```

```
SQL> select con_id,name,PRIORITY from v$pdbs;
```

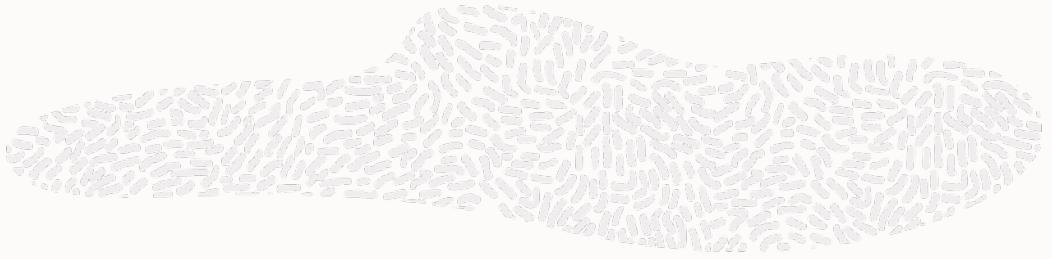
CON_ID	NAME	PRIORITY
2	PDB\$SEED	1
3	ORCL232BPDB1	<u>4096</u>

# Read only users and sessions

- Hybrid read-only mode for pluggable databases
- Control PDB Open Order
- Read only users and sessions

# Read only users and sessions

## 概要



### Read only users and sessionsについて

- 管理者は、データベースに接続しているユーザーの権限に関係なく、ユーザーまたはセッションの読み取り/書き込み操作を有効にするかどうかを制御できます。
- セッションに対する READ\_ONLY 設定は、あらゆる種類のコンテナのあらゆる種類のユーザーに適用されます。
- ユーザーに対する READ\_ONLY 設定は PDB のローカル ユーザーにのみ適用されます。
- ALTER USER または ALTER SESSION (READ\_ONLY) によって制御できます。

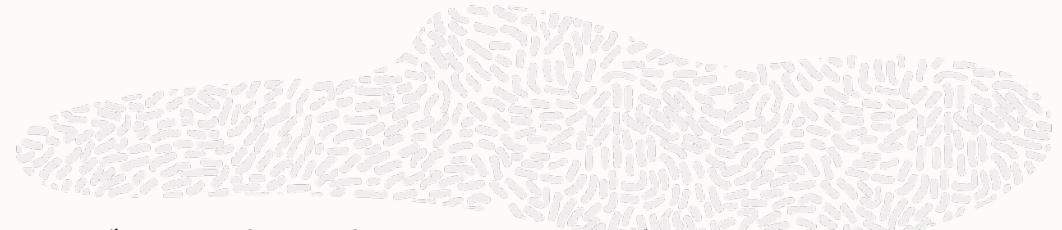
⇒ [https://docs.oracle.com/en/database/oracle/oracle-database/23/refrn/READ\\_ONLY.html#GUID-CDB2436A-7C36-4C94-95C6-C72A67F64C18](https://docs.oracle.com/en/database/oracle/oracle-database/23/refrn/READ_ONLY.html#GUID-CDB2436A-7C36-4C94-95C6-C72A67F64C18)

### メリット

- 権限を取り消したり再付与したりすることなく、任意のユーザーまたはセッションの読み取り/書き込み機能を無効化および再有効化できるため、テスト、管理、またはアプリケーション開発の目的でユーザーまたはセッションの権限を一時的に制御する柔軟性が向上します。
- 同じユーザーまたはセッションが使用するアプリケーションの様々な部分の読み取り/書き込み動作を簡単に制御できるようになります。

# Read only users and sessions

## 指定コマンドについて



ALTER USERまたはCREATE USER文のREAD ONLY句を使用して、PDBユーザーへの読み取り専用アクセスを設定できます。

PDBユーザーに対して読み取り専用アクセスが有効になった後、そのユーザーがPDBに接続するたびに、セッションはデータベースが読み取り専用モードでオープンされているかのように動作し、ユーザーは書き込み操作を実行できません。

この文は、ALTER USERまたはCREATE USER権限を持つすべてのユーザーが実行できます。

PDBローカル・ユーザに対して読み取り専用アクセスを有効にするコマンド例:

```
alter user user1 read only;
```

PDBローカル・ユーザに対して読み取り専用アクセスを取り消すコマンド例:

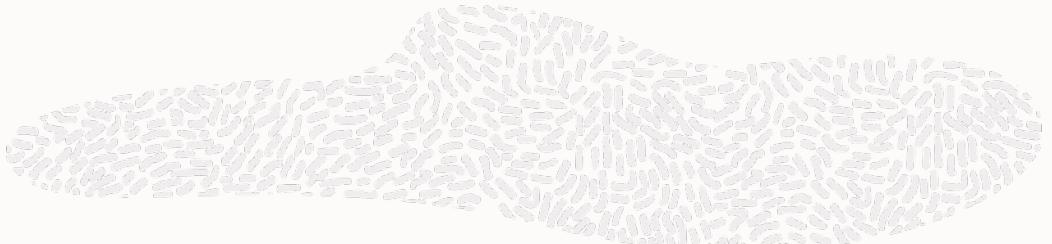
```
alter user user1 read write;
```

PDBローカル・ユーザの作成時に読み取り専用アクセスを有効にするコマンド例:

```
create user u1 identified by u1 read only;
```

# Read only users and sessions

## Read only usersの検証



実効例：

```
SQL> alter user local_scott read only;
ユーザーが変更されました。

SQL> select username,read_only from dba_users where username = 'LOCAL_SCOTT';

USERNAME          READ_ONLY
-----
LOCAL_SCOTT      YES

経過: 00:00:00.01
SQL> conn local_scott/tiger@ptvm07/orcl232bpdb1
接続されました。
SQL> CREATE TABLE EMP
  2  (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
  3  ENAME VARCHAR2(10),
  4  JOB VARCHAR2(9),
  5  MGR NUMBER(4),
  6  HIREDATE DATE,
  7  SAL NUMBER(7,2),
  8  COMM NUMBER(7,2),
  9  DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT);
CREATE TABLE EMP
*
行1でエラーが発生しました。:
ORA-16000: データベースまたはプラガブル・データベースは読み取り専用アクセスでオープンされています
```

# Database Resident Connection Pool新機能

- DRCPプールを複数作成可能
- DRCPプール停止時の挙動



# Oracle Database 23c Database Resident Connection Pool新機能



## 1. Database Resident Connection Pool(DRCP)とは

- 専用サーバー接続
- DRCP

## 2. Oracle Database 23c Database Resident Connection Pool新機能

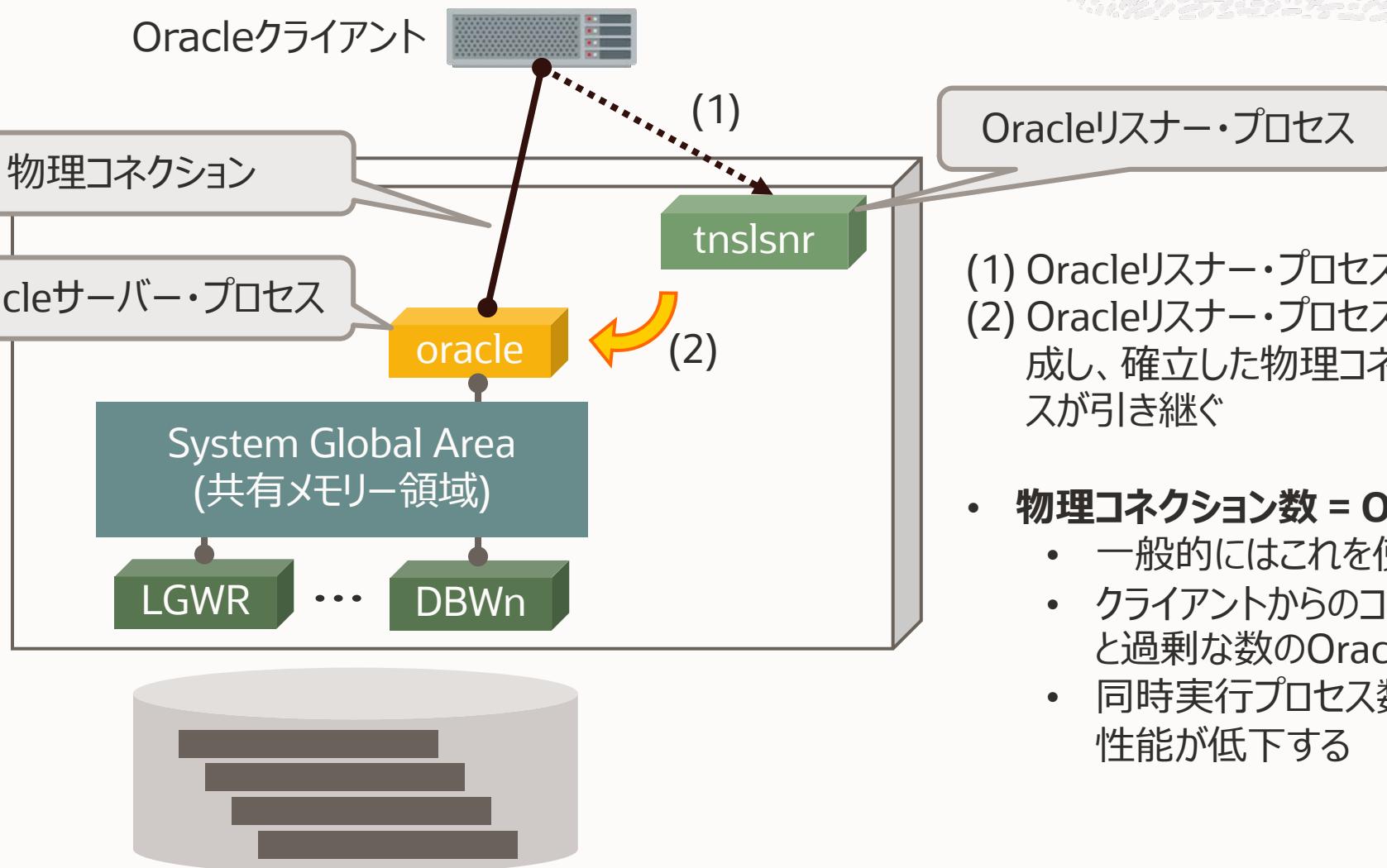
- DRCPプールを複数作成可能
- DRCPプール停止時の挙動

# Database Resident Connection Poolとは

データベース・コネクションとOracleサーバー・プロセスの関係

# 専用サーバー(Dedicated Server)接続 - 基本的な接続形態

1本の物理コネクションが1つのOracleサーバー・プロセスと直結

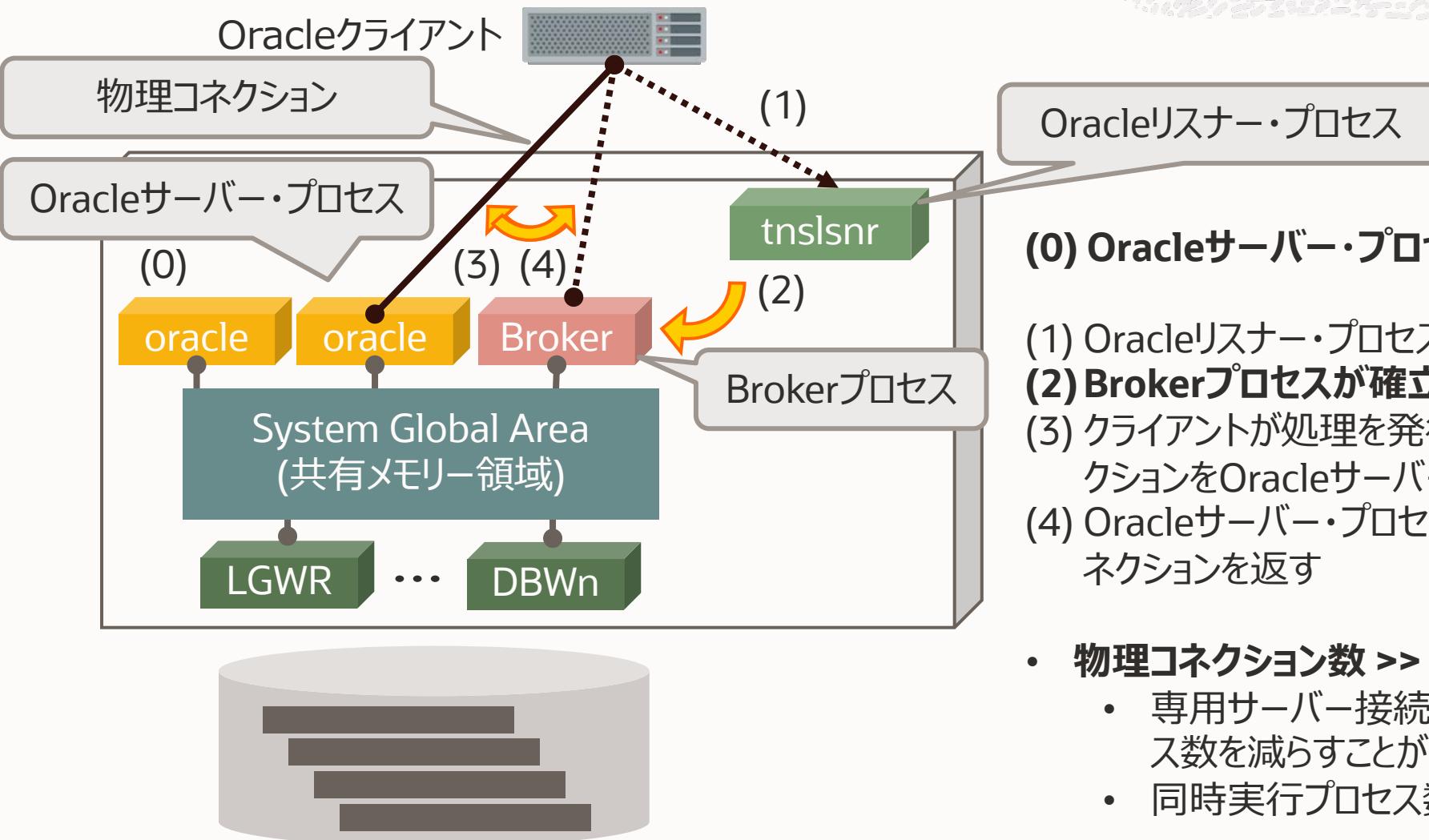


(1) Oracleリスナー・プロセスに接続リクエスト着信  
(2) Oracleリスナー・プロセスがOracleサーバー・プロセスを生成し、確立した物理コネクションはOracleサーバー・プロセスが引き継ぐ

- 物理コネクション数 = Oracleサーバー・プロセス数
  - 一般的にはこれを使用する
  - クライアントからのコネクション数を適切に管理しないと過剰な数のOracleサーバー・プロセスが生成される
  - 同時実行プロセス数>>CPUスレッド数 だとえって性能が低下する

# Database Resident Connection Pool

Brokerがコネクションを保持し、SQL処理時に1本の物理コネクションが1つのOracleサーバー・プロセスと直結



# DRCPプールを複数作成可能

- DRCPプールを複数作成可能
- DRCPプール停止時の挙動

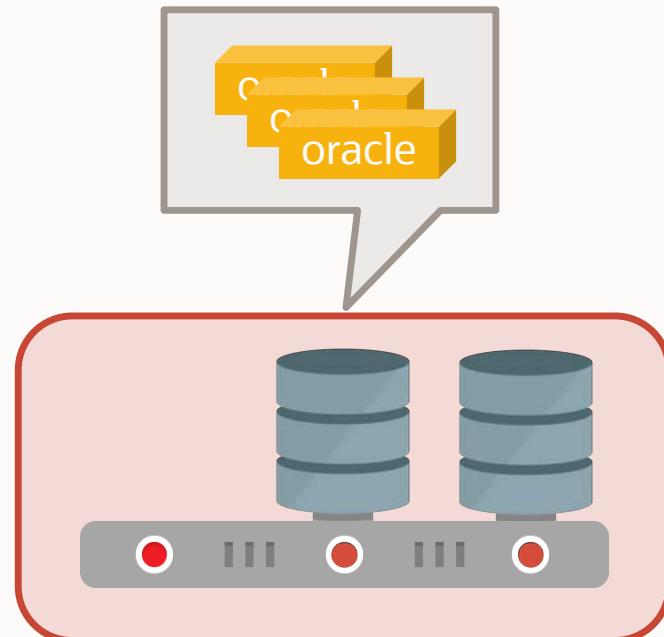
# Database Resident Connection Poolのプロセス・プールの区別変遷

どの単位でOracleサーバー・プロセスのプールを作成できるか

## 19cまで

- CDB全体で1つのプール
- PDBごとにはパラメータを制御できない

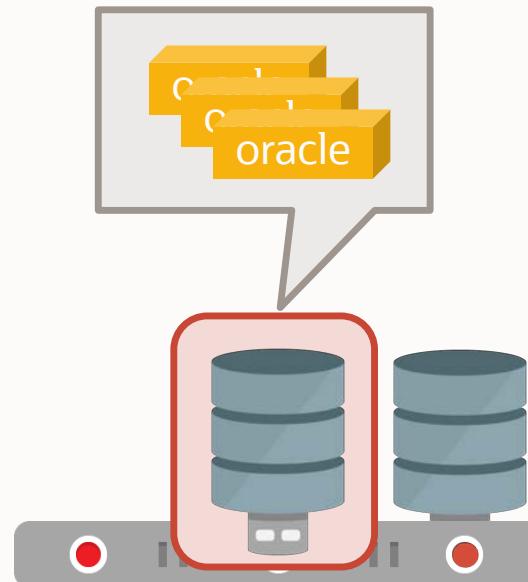
デフォルト・プール



## 21c

- PDBごとに1つのプール
- PDBごとにパラメータを制御できる

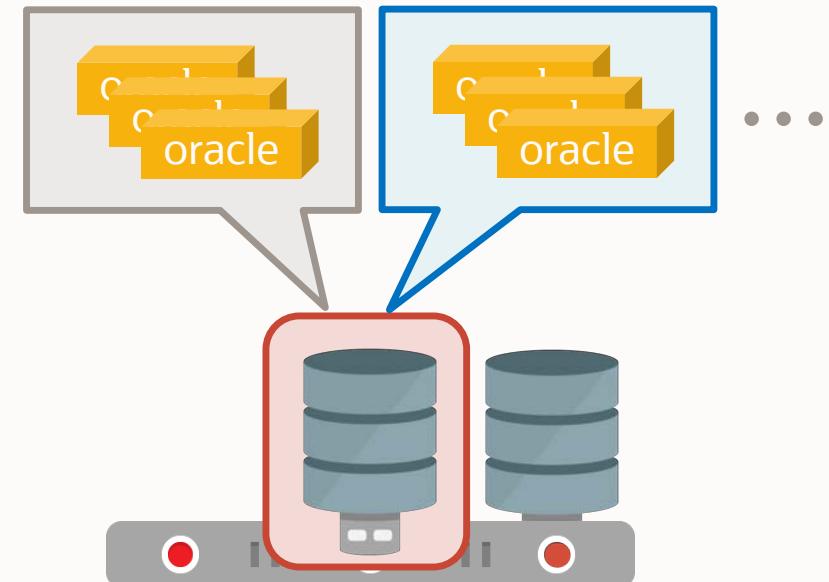
デフォルト・プール



## 23c

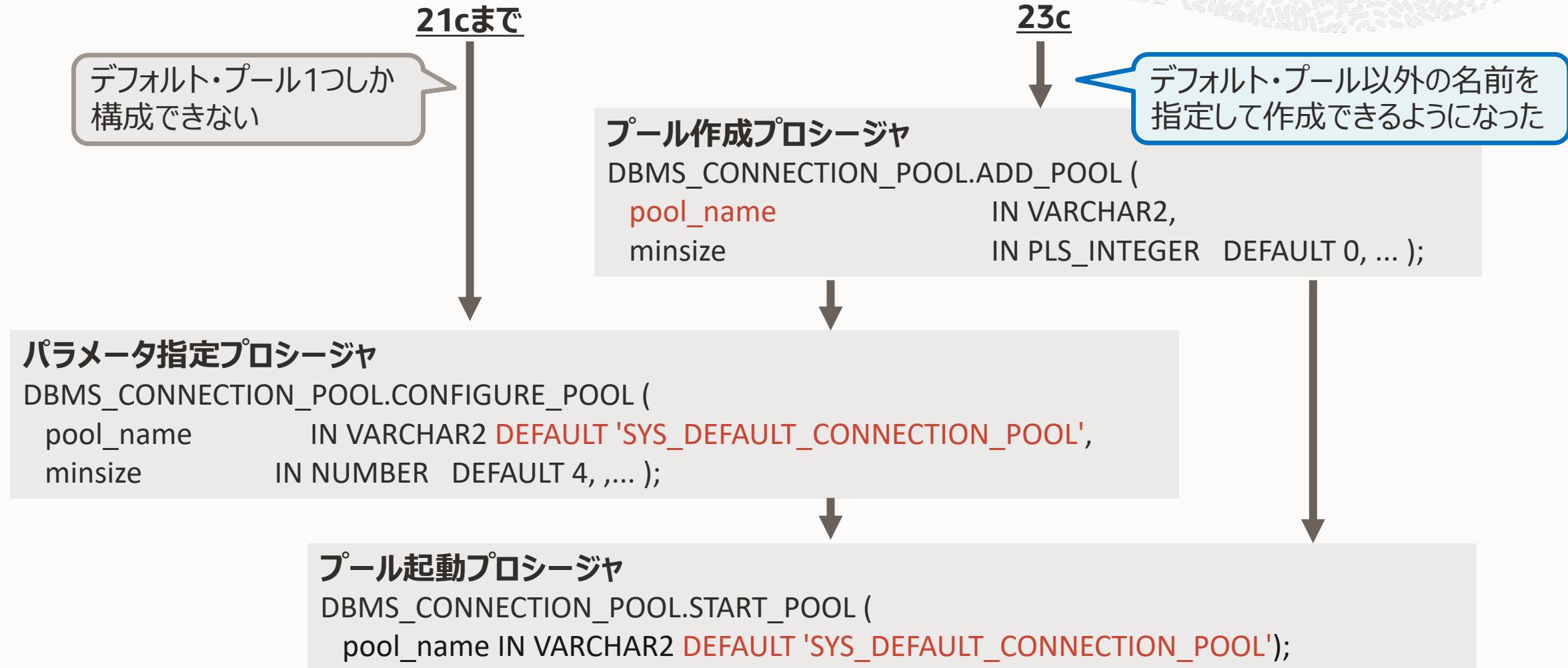
- PDBごとに複数プールを作成可能
- アプリケーションごとにプールを区別してパラメータを制御できる

デフォルト・プール ユーザー作成プール1



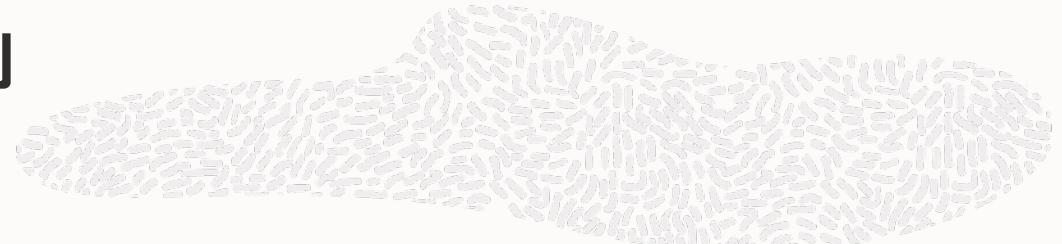
# Database Resident Connection Poolのプロセス・プールの区別変遷

DBMS\_CONNECTION\_POOL.ADD\_POOL() プロシージャが追加された



# DRCPに接続するためのデータベース接続文字列

SERVER=POOLED



21cまで: (プラガブル)データベースに1つのデフォルト・プールしかない

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST = ptvmscan-03.jp.oracle.com)(PORT = 1521))
  (CONNECT_DATA=(SERVICE_NAME=svcpdb1)
    (SERVER=POOLED))
)
```

デフォルト・プール(SYS\_DEFAULT\_CONNECTION\_POOL)に接続される

23cから: プラガブル・データベースに複数のプールが存在する

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=TCP)(HOST = ptvmscan-03.jp.oracle.com)(PORT = 1521))
  (CONNECT_DATA=(SERVICE_NAME=svcpdb1)
    (SERVER=POOLED)(POOL_NAME=pool1))
)
```

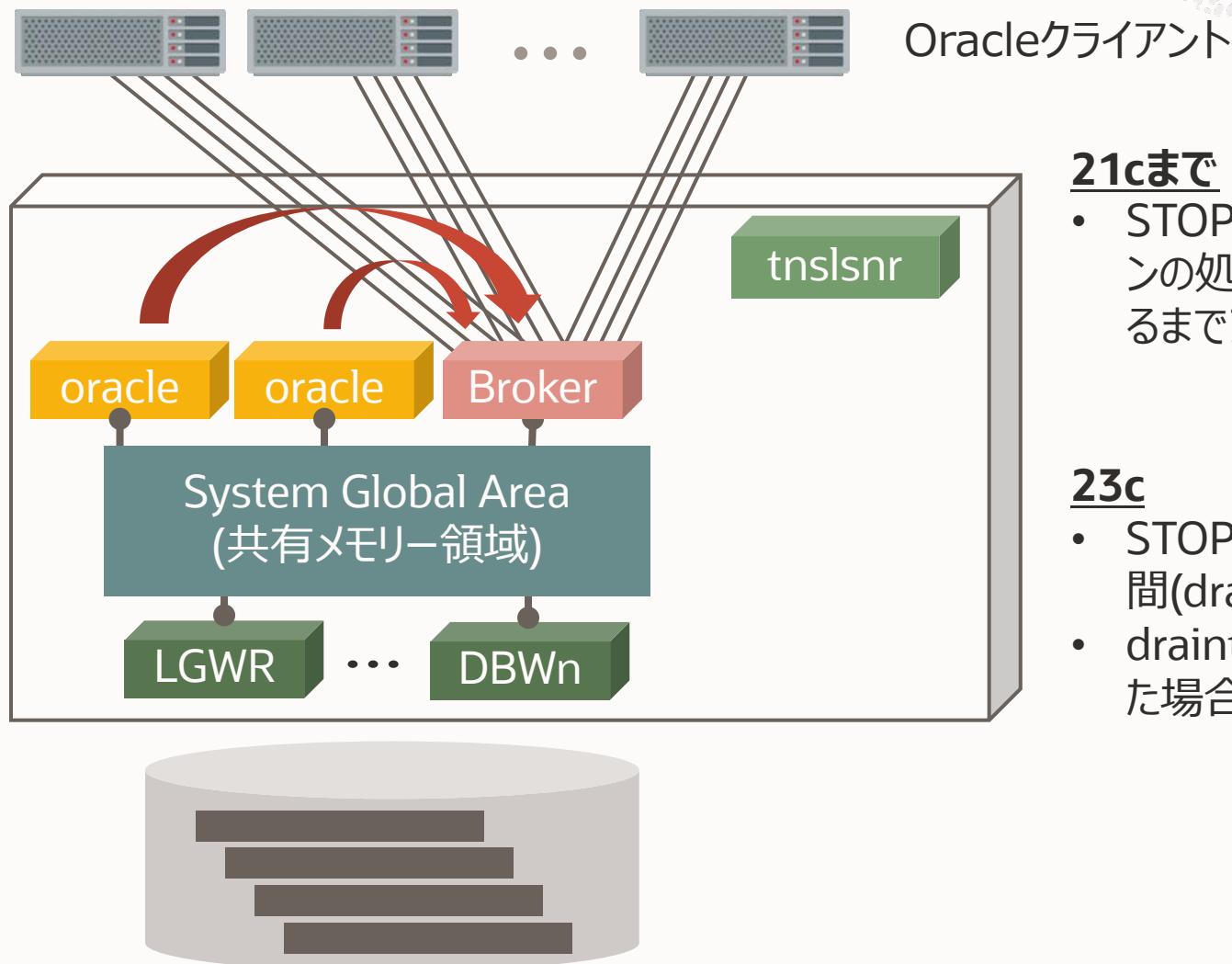
POOL\_NAMEで指定したプール(pool1)に接続される  
省略するとデフォルト・プールに接続される

# DRCPプール停止時の挙動

- DRCPプールを複数作成可能
- DRCPプール停止時の挙動

# DRCPプールの停止時の挙動

プール停止プロシージャSTOP\_POOL



## 21cまで

- STOP\_POOLプロシージャを実行してもすべてのコネクションの処理が完了しコネクションをBrokerプロセスに返却するまでプールは停止されない

## 23c

- STOP\_POOLプロシージャにプロセス解放までの猶予時間(draintime)を指定できるようになった
- draintimeを経過しても処理継続中のコネクションがあった場合は、プールは停止されコネクションは切断される

# プールを停止するときにプロセス解放の猶予時間(draintime)を指定できる プール停止プロシージャSTOP\_POOL



## 21cまで

```
DBMS_CONNECTION_POOL.STOP_POOL (
    pool_name      IN  VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL');
```

- すべてのコネクションの処理が完了しプロセスをプールに返却するまでプールは停止されない

## 23c

```
DBMS_CONNECTION_POOL.STOP_POOL (
    pool_name      IN  VARCHAR2 DEFAULT 'SYS_DEFAULT_CONNECTION_POOL',
    draintime      IN  PLS_INTEGER DEFAULT 2147483647);
```

- draintime (単位:秒)
  - プールに停止指示を出してから、コネクションの処理完了によりプロセスが解放されるのを待つ猶予時間
  - draintimeを経過しても処理継続中のコネクションがあった場合は、プールは停止されコネクションは切断される
  - 0を指定すると即時停止
  - draintimeを指定しないと21cまでと同じ動作

# 表領域管理

1. Read-Only Tablespace on Object Storage
2. LOBセグメントに関する新機能

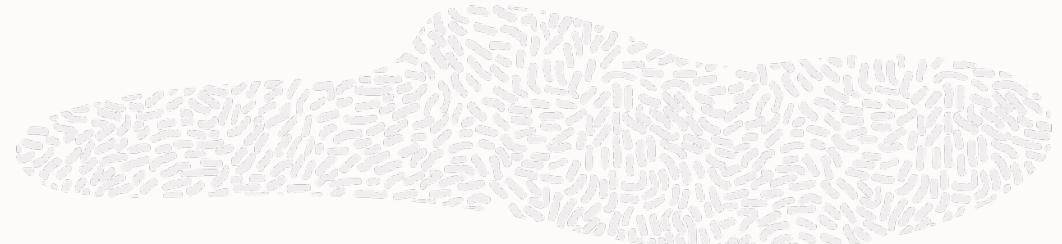
[ ]

O

# Read-Only Tablespace on Object Storage

1. **Read-Only Tablespace on Object Storage**
2. LOBセグメントに関する新機能

# Read-Only Tablespace on Object Storage



## 概要

すでにデータが格納されている表領域をオブジェクト・ストレージに移動させる

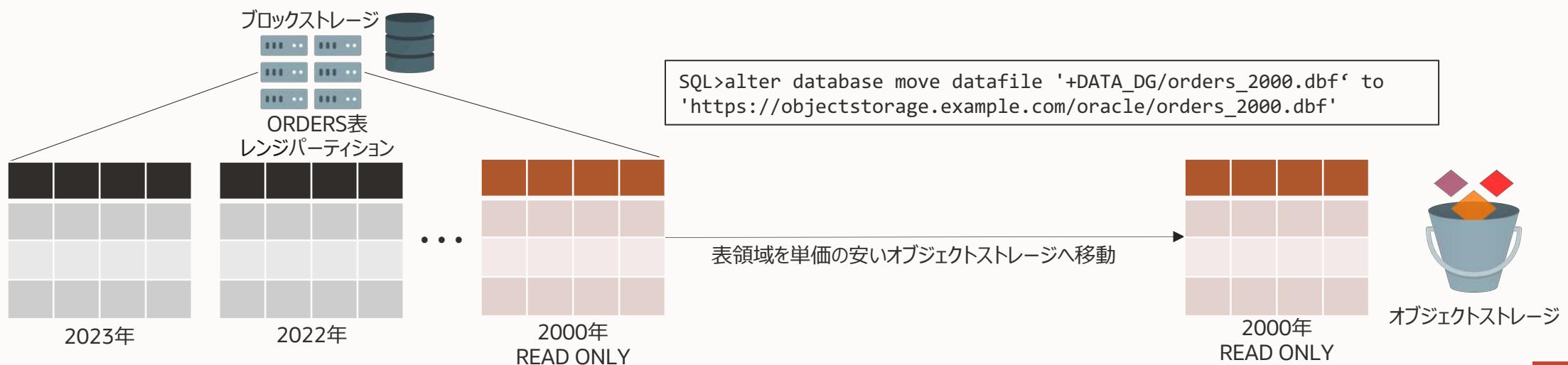
オブジェクト・ストレージに格納したファイルは更新できないため、その表領域はREAD ONLYとしてのみアクセス可能

## メリット

オブジェクトストレージを使うことで保存コストが低く抑えられる

## 用途例

パーティションを分けた更新しない古い表領域ファイルをブロックストレージから単価の安いオブジェクトストレージへ移行することで保存コストを下げる



# LOBセグメントに関する新機能

1. Read-Only Tablespace on Object Storage
2. LOBセグメントに関する新機能

# SecureFiles領域管理の理解

- LOBを持つ表を作成する場合には、自動あるいは手動でLOBセグメントを構成

Create Table 文でサンプル表T1を作成した場合に自動生成された場合

```
SQL>CREATE TABLE hr.t1(a CLOB) LOB(a) STORE AS SECUREFILE TABLESPACE users;
```

Large Object は、このセグメント内で管理

Owner	Object Name	Object Type	Size (MB)	Initial Extent (MB)	Next Extent (MB)	Num. Extents	Max. Extents
1 HR	T1 表セグメント	TABLE	0.063	0.063		1	1 2147483645
2 HR	SYS_IL00000762 LOB索引セグメント	LOBINDEX	0.063	0.063		1	1 2147483645
3 HR	SYS_LOB0000076 LOBセグメント	LOBSEGMENT	1.313	0.25		1	3 2147483645

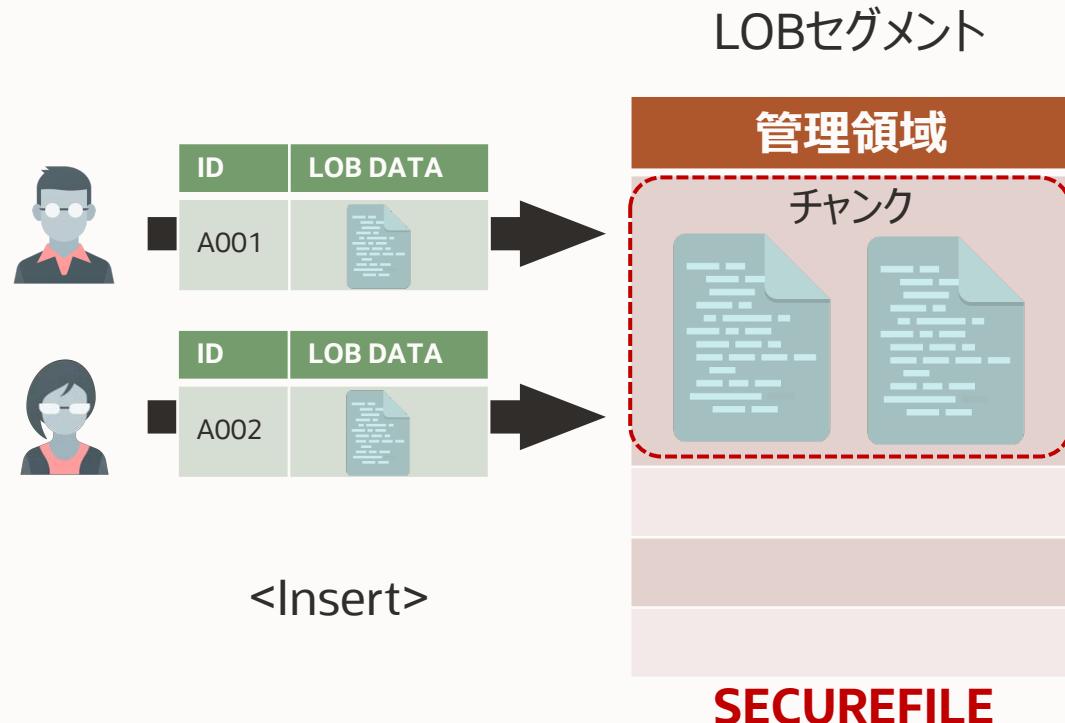
LOBセグメントに関する23cでの新機能は以下です。

- 重複除外設定にする場合の領域削減効果の見積もりが可能になった
- MOVEで実施していた名称変更がメタデータ変更のみで対応可能になった

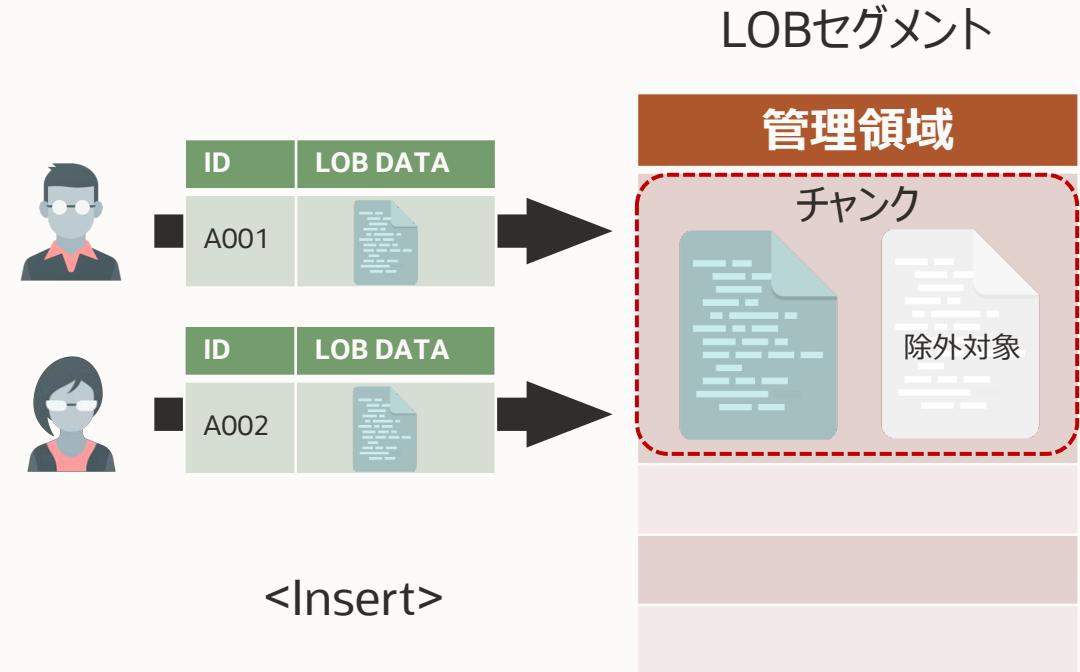
# SecureFiles領域管理の理解

LOBデータの重複除外のイメージ

重複除外なし



重複除外あり



重複は、LOBセグメント内で検出

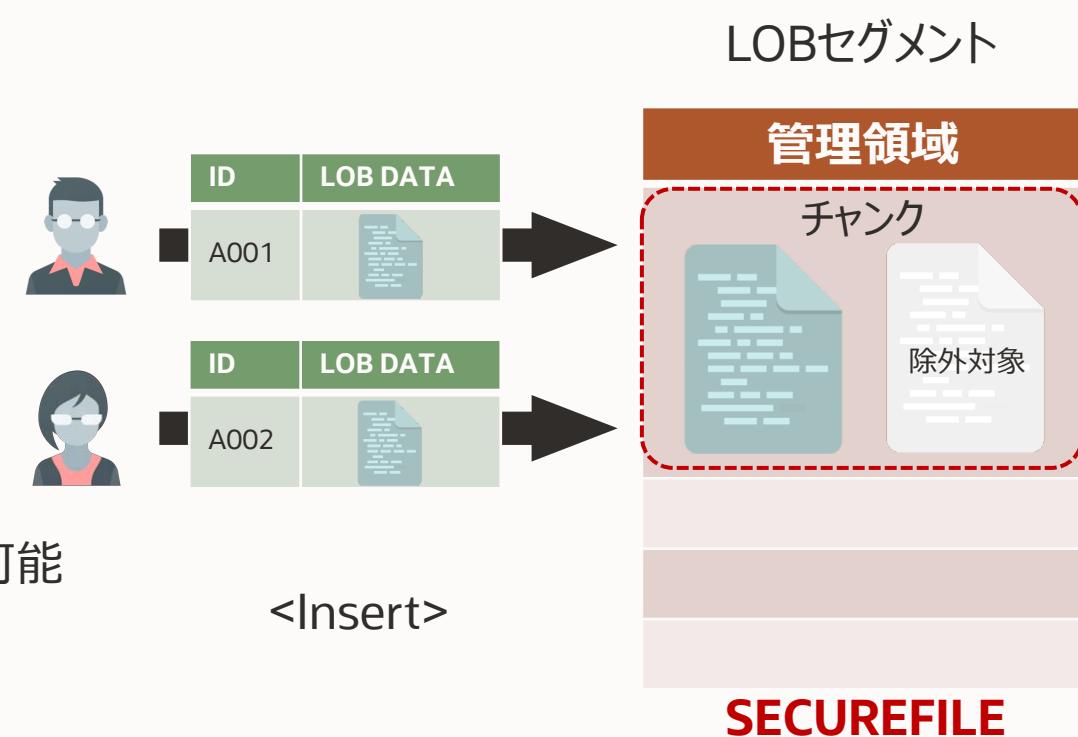
# LOB重複排除によって節約されるスペースの見積もり

## 重複除外

- 重複LOBデータを自動的に検出
- データのコピーを1つだけ格納することで領域を節約

## 23c

- 重複除外の有効前に、節約できるスペースの見積もりが可能
  - GET\_LOBDEDUPLICATION\_RATIO関数を提供



# LOBセグメントの名前変更

## 従来

- 既存の LOB セグメントの名前を変更するには、ユーザーは **ALTER TABLE ... MOVE** のような操作を実行  
→この操作は名前変更の一環として **LOB データを物理的に移動**するため、実行速度が遅くなる可能性がある

## 23c

- ALTER TABLE MODIFY** で LOB セグメントの名前変更は **メタデータ変更で対応可能**  
→LOB データの **物理的な移動が不要**になったので、LOB セグメントの名前変更の操作がシンプルに

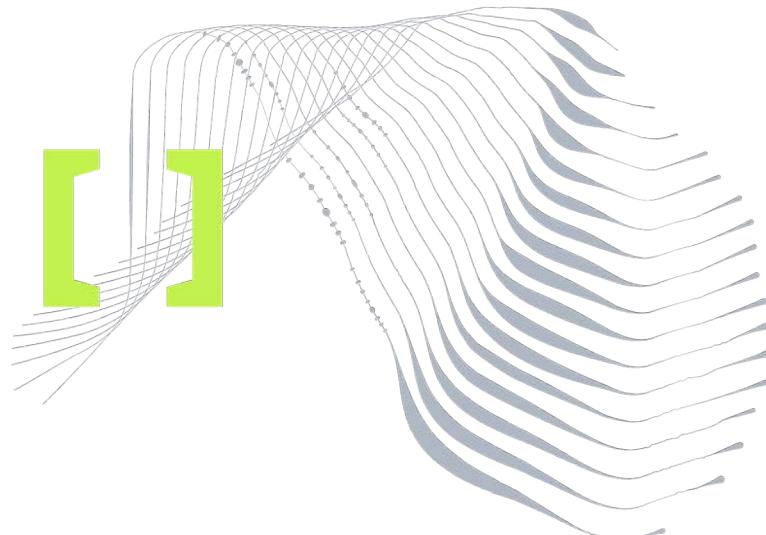
**ALTER TABLE MOVE LOB**



**ALTER TABLE MODIFY**

# Agenda

1. Multitenant Architecture
2. Database Resident Connection Pool(DRCP)
3. 表領域管理
4. Backup & Recovery
5. Diagnosability



# Backup & Recovery

1. デフォルト暗号化アルゴリズムの変更
2. リカバリ・カタログに関する機能拡張
3. RMANバックアップの取得先の拡張
4. Flashback Logを高速リカバリ領域(FRA)外の領域で管理



# デフォルト暗号化アルゴリズムの変更

1. デフォルト暗号化アルゴリズムの変更
2. リカバリ・カタログに関する機能拡張
3. RMANバックアップの取得先の拡張
4. Flashback Logを高速リカバリ領域(FRA)外の領域で管理

# デフォルト暗号化アルゴリズムの変更

RMANバックアップのデフォルト暗号化アルゴリズムがAES128-CFBからAES256-XTSに変更

## 従来

- 暗号化アルゴリズムを指定しない場合のデフォルトの暗号化アルゴリズムは Advanced Encryption Standard(AES)128-CFB
  - AES-CFBをバックアップ・セキュリティに使用
  - CFBモード (Cipher-FeedBack) : FIPS 140-2 で検証された動作モード
  - COMPATIBLE初期化パラメータ 10.2.0以上が必要



AES128-CFB → AES256-XTS

## 23c

- Advanced Encryption Standard (AES) 256-XTSがRMANバックアップのデフォルト暗号化アルゴリズムに
- XTSモード (XEX-based tweaked codebook mode with ciphertext stealing) : IEEE1619-2007で標準化された暗号化モード
- COMPATIBLE初期化パラメータ 21.0.0以下の場合、AES-CFBをバックアップ・セキュリティのアルゴリズムに使用
- COMPATIBLE初期化パラメータ 23.0.0以上の場合、AES-XTSをバックアップ・セキュリティのアルゴリズムに使用
- 今回の変更でより高速なアルゴリズムがデフォルト暗号化アルゴリズムに採用され、暗号強度も上げられました

# デフォルト暗号化アルゴリズムの変更

## AES-XTSモードのサポート

RMANがサポートするバックアップ暗号化アルゴリズム

COMPATIBLE初期化パラメータ	使用される暗号化アルゴリズム
21.0.0以下	<b>AES-CFB暗号化アルゴリズムを使用</b> <ul style="list-style-type: none"><li>• AES256</li><li>• AES128(デフォルト)</li><li>• AES192</li></ul>
23.0.0以上	<b>AES-XTS暗号化アルゴリズムを使用</b> <ul style="list-style-type: none"><li>• AES256(デフォルト)</li><li>• AES128</li></ul>

- COMPATIBLE初期化パラメータ  
<https://docs-uat.us.oracle.com/en/database/oracle/oracle-database/23/refrn/COMPATIBLE.html#GUID-6C57EE11-BD06-4BB8-A0F7-D6CDDD086FA9>
- V\$RMAN\_ENCRYPTION\_ALGORITHMS  
[https://docs.oracle.com/en/database/oracle/oracle-database/23/refrn/V-RMAN\\_ENCRYPTION\\_ALGORITHMS.html#GUID-9761C75A-59B9-4ED7-98B1-55FBBA1858F1](https://docs.oracle.com/en/database/oracle/oracle-database/23/refrn/V-RMAN_ENCRYPTION_ALGORITHMS.html#GUID-9761C75A-59B9-4ED7-98B1-55FBBA1858F1)

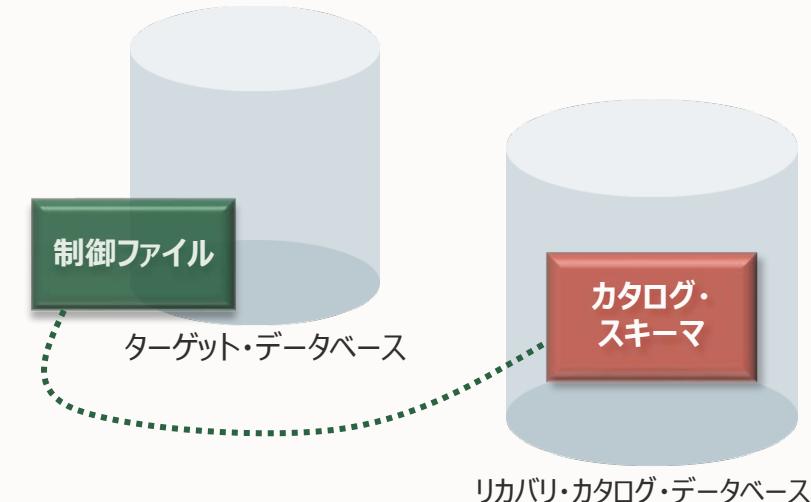
# リカバリ・カタログに関する機能拡張

1. デフォルト暗号化アルゴリズムの変更
- 2. リカバリ・カタログに関する機能拡張**
3. RMANバックアップの取得先の拡張
4. Flashback Logを高速リカバリ領域(FRA)外の領域で管理

# RMANリポジトリとリカバリカタログ

## RMAN リポジトリ

- RMANがバックアップ、リカバリおよびメンテナンスに使用するターゲット・データベースに関する管理情報の格納場所
- 格納場所は、制御ファイルとリカバリ・カタログ・データベース
  - 1. 制御ファイル  
データベースを構成するファイルのパスの情報(オンラインREDOログファイル、データファイル、アーカイブREDOログファイル)  
RMANで生成されるバックアップ・ファイルの情報
    - 領域に制限有り： 初期化パラメータ：CONTROL\_FILE\_RECORD\_KEEP\_TIME (デフォルト7日)
  - 2. リカバリ・カタログ・データベース
    - リポジトリを長期間保存可能
    - 複数のOracle Databaseのバックアップ状況を一元管理可能



# リカバリ・カタログに関する機能拡張

23c

1. リカバリ・カタログへのスタンバイ・データベースの登録がスタンバイから実行可能に
2. リカバリ・カタログの再同期プロセスにおいて複数のセーブポイントを導入
3. スタンバイ・データベースの制御ファイルを使用したリカバリ・カタログの完全同期化
4. UPGRADE CATALOGコマンドの実行動作を調整
5. メンテナンス・モードでリカバリ・カタログのアップグレードを監視

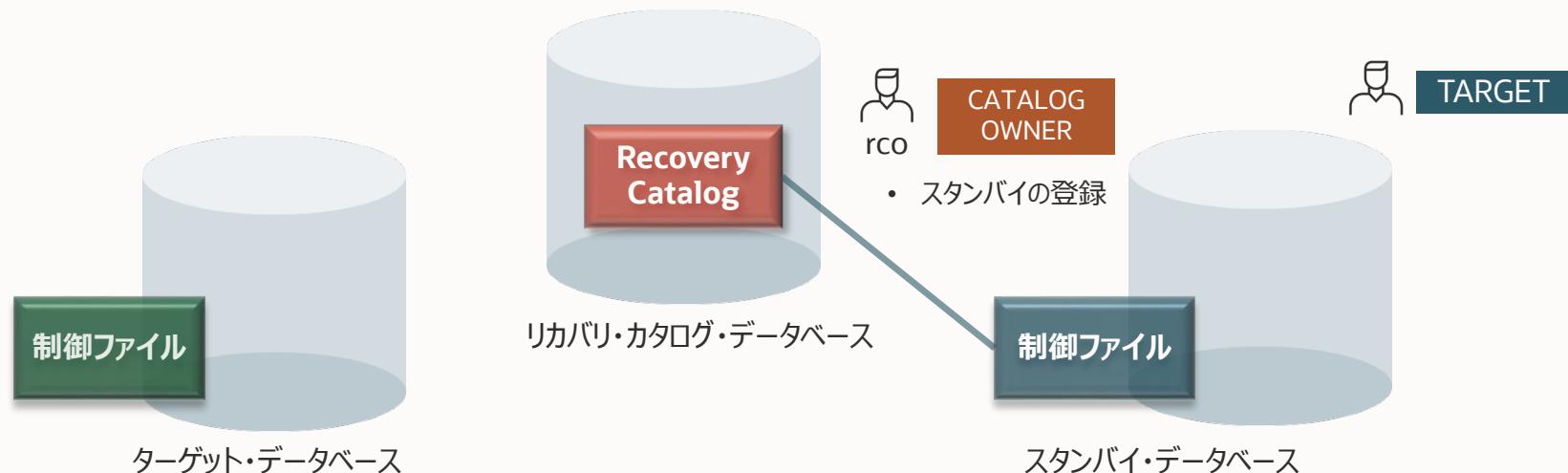
# リカバリ・カタログへのスタンバイ・データベースの登録がスタンバイから実行可能に

## 従来

- Data Guard環境でのデータベースの登録はプライマリ・データベース時のみ実行可能
  - プライマリ・データベースにカタログ・オーナーとして接続をし、REGISTER DATABASEコマンドを発行

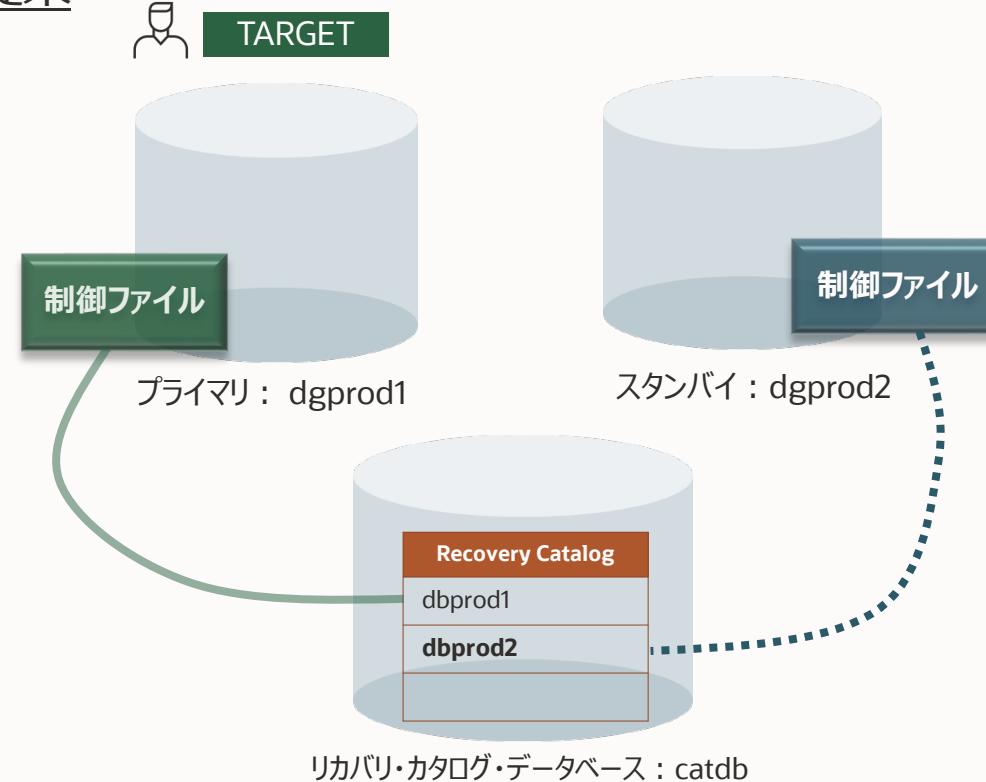
## 23c

- RMANがTARGETとしてスタンバイ・データベースに接続している場合、REGISTER DATABASEコマンドでスタンバイ・データベースを明示的にリカバリ・カタログに登録可能

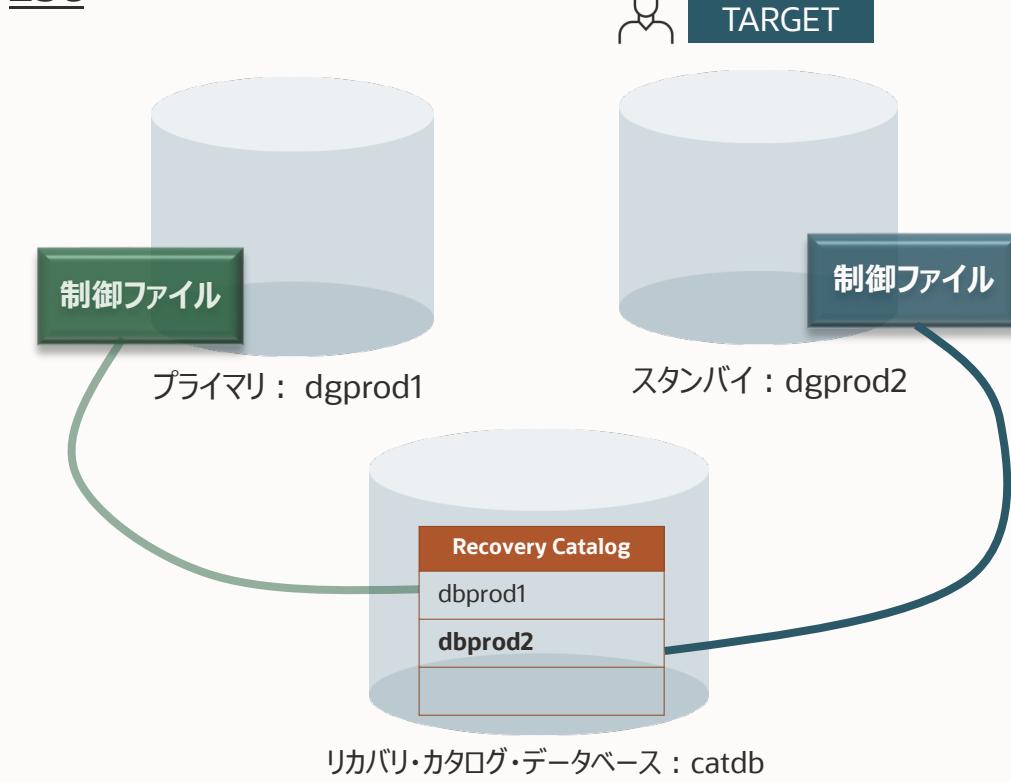


# リカバリ・カタログへのスタンバイ・データベースの登録がスタンバイから実行可能に

従来



23c



- プライマリにTARGET接続した場合は  
スタンバイの登録が可能  
RMAN> REGISTER DATABASE;

- スタンバイにTARGET接続した場合は  
スタンバイのカタログ登録が不可

NEW

- プライマリにTARGET接続した場合は  
スタンバイの登録が可能  
RMAN> REGISTER DATABASE;

- スタンバイにTARGET接続した場合も  
スタンバイのカタログ登録が可能  
RMAN> REGISTER DATABASE;

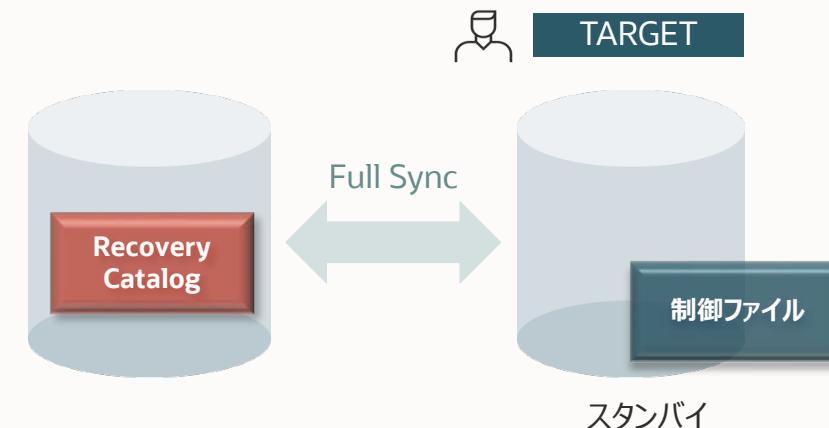
# スタンバイ・データベースの制御ファイルを使用したリカバリ・カタログの完全同期化

## 以前のリリース

- プライマリにTARGET接続している場合はプライマリの制御ファイルとリカバリ・カタログが自動で完全同期化
- スタンバイにTARGET接続している場合は、スタンバイの制御ファイルとリカバリ・カタログが自動で部分同期化
- スタンバイの完全同期化はプライマリでRESYNC CATALOG FROM DB\_UNIQUE\_NAMEを使用し、手動実行

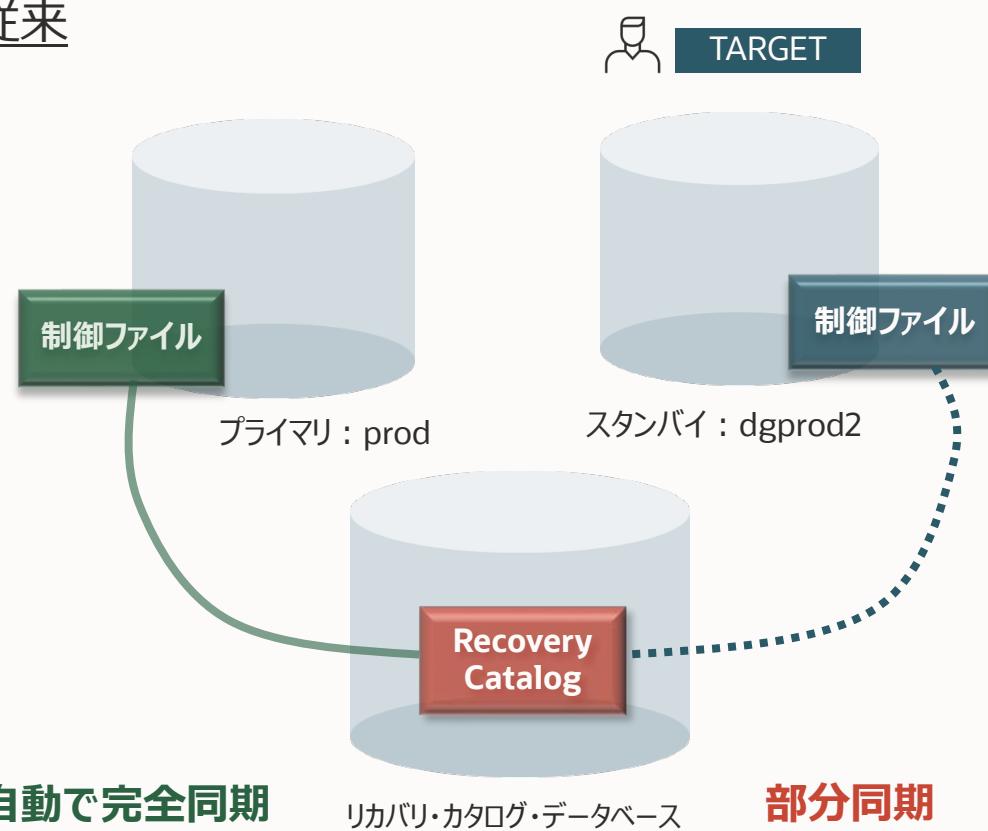
## 23c

- スタンバイにTARGET接続している場合は、スタンバイの制御ファイルとリカバリ・カタログが自動で完全同期化



# スタンバイ・データベースの制御ファイルを使用したリカバリ・カタログの完全同期化

従来

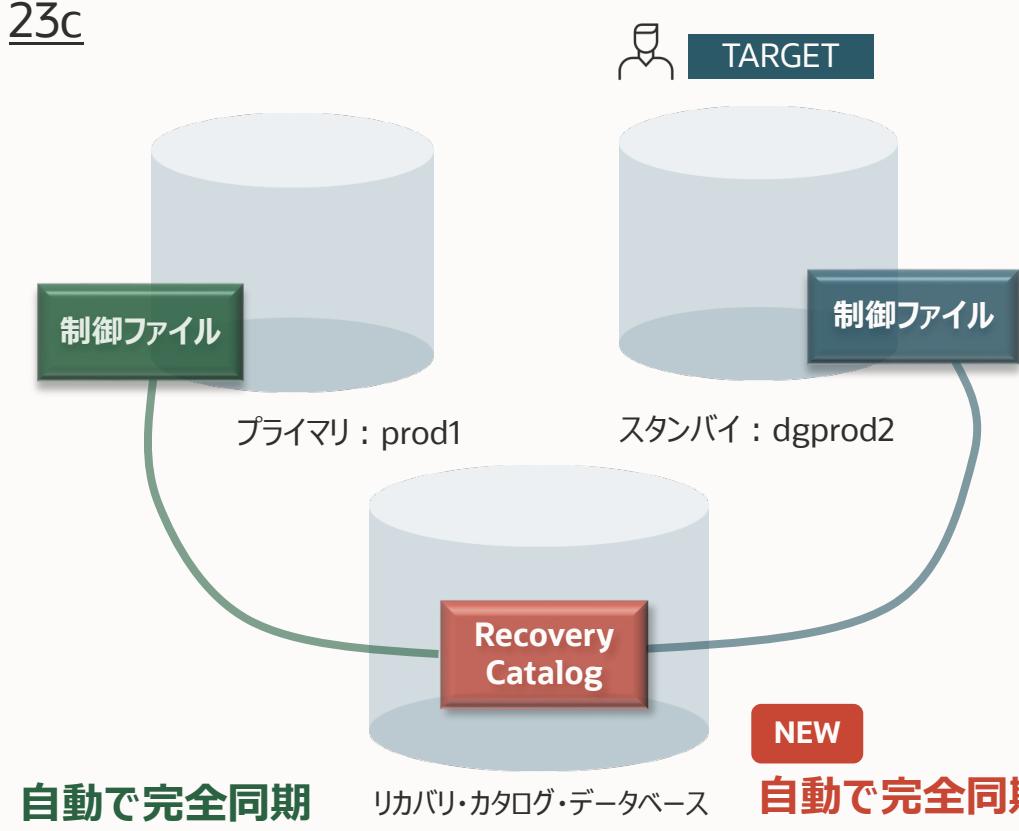


自動で完全同期

スタンバイの完全同期は  
プライマリで手動実行

```
RMAN> RESYNC CATALOG FROM DB_UNIQUE_NAME dgprod2;
```

23c



自動で完全同期

NEW

自動で完全同期

# リカバリ・カタログの再同期プロセスにおいて複数のセーブポイントを導入

## リカバリ・カタログの再同期

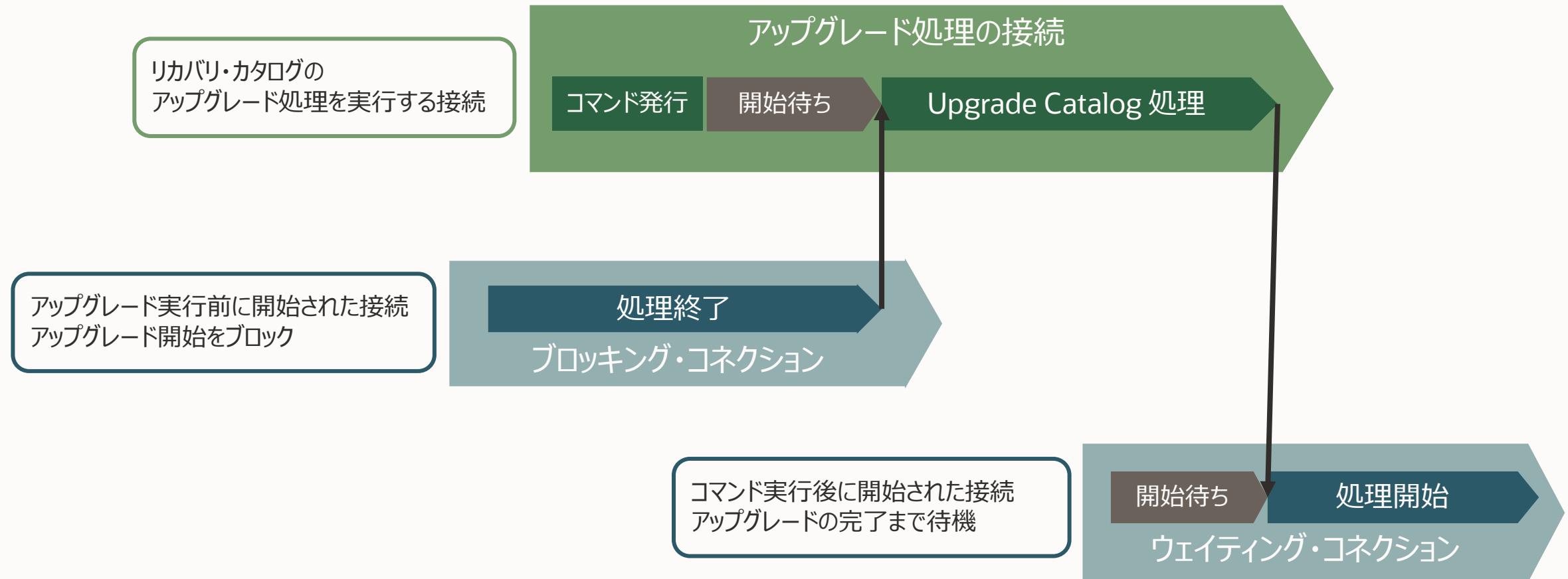
- リカバリ・カタログのデータを制御ファイル内の全ての変更済みメタデータにアップデートする処理

23c

- リカバリ・カタログの再同期プロセスにおいて**複数のセーブポイントを導入**
- RMANが制御ファイルからリカバリ・カタログへの同期処理を実行中に、定義済みの**セーブポイントが周期的に作成**
- 再同期中に予期せぬ事態が起こってもRMANが特定のセーブポイントまでロールバックすることが可能
  - トランザクション全体をロールバックする必要なし



# リカバリ・カタログのアップグレード処理とカタログ接続



# メンテナンス・モードでリカバリ・カタログのアップグレードを監視

23c

リカバリ・カタログにメンテナンス・モードで接続し、**カタログ・スキーマに対する接続を管理**

アップグレード処理実行中にブロッキング/ウェイティング・コネクションを管理し、アップグレード処理の実行をスムーズに

## ブロッキング・コネクション

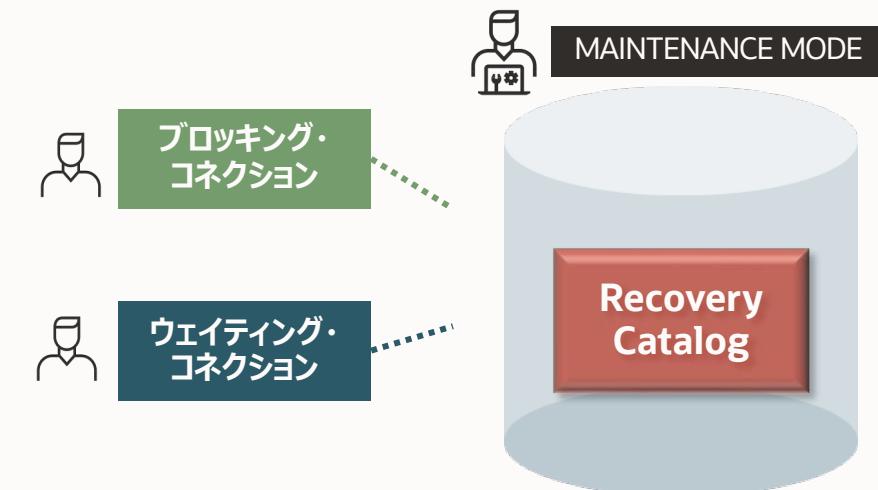
- リカバリ・カタログ・スキーマ関連のRMANジョブで、UPGRADE CATALOGコマンド実行前に開始された接続
- リカバリ・カタログのアップグレード開始を妨げているもの

## ウェイティング・コネクション

- UPGRADE CATALOGコマンド実行後に開始されたRMAN接続
- リカバリ・カタログのアップグレード完了を待っているもの

メンテナンス・モードでは、以下の接続のリスト表示・切断が可能

- リカバリ・カタログに接続されている全ての接続
- ブロッキング・コネクションとウェイティング・コネクション



## カタログに接続しているコネクションの表示

```
RMAN> SET CATALOG MAINTENANCE ON;  
RMAN> CONNECT CATALOG rco@catdb;  
LIST CONNECTED USERS;  
RMAN-03023: executing command: SET catalog maintenance ON
```

全ての接続済みコネクションを表示

```
RMAN-06008: connected to recovery catalog database
```

```
RMAN-07556: Following sessions are blocking the catalog schema upgrade
```

```
RMAN-07559: =====
```

```
RMAN-07560: Logon_time           SID  Serial User
```

```
RMAN-07559: =====
```

既存の接続は'RMAN'と'RMAN3'

```
RMAN-07561: 2023-08-01 10:54:14    191    35956 RMAN
```

```
RMAN-07561: 2023-08-01 10:54:20    192    42240 RMAN3
```

```
RMAN-07559: =====
```

# アップグレード処理をブロックしているコネクションの表示とターミネート

```
RMAN> SET CATALOG MAINTENANCE ON;  
CONNECT CATALOG rco@catdb;  
LIST BLOCKING CONNECTED USERS;  
TERMINATE BLOCKING CONNECTED USERS;
```

ブロッキング・コネクションを表示

ブロッキング・コネクションをターミネート

```
RMAN-03023: executing command: SET catalog maintenance ON
```

```
RMAN-06008: connected to recovery catalog database  
RMAN-07556: Following sessions are blocking the catalog schema upgrade
```

```
RMAN-07559: =====
```

```
RMAN-07560: Logon_time           SID  Serial User
```

ブロッキング・コネクションは‘RMAN’

```
RMAN-07559: =====
```

```
RMAN-07561: 2023-08-01 11:17:13    53    34909 RMAN
```

```
RMAN-07559: =====
```

```
RMAN-07556: Following sessions are blocking the catalog schema upgrade
```

```
RMAN-07559: =====
```

```
RMAN-07560: Logon_time           SID  Serial User
```

‘RMAN’をターミネート

```
RMAN-07559: =====
```

```
RMAN-07561: 2023-08-01 11:17:13    53    34909 RMAN
```

```
RMAN-06958: Executing: alter system kill session '53, 34909'
```

```
RMAN-07559: =====
```

# UPGRADE CATALOGコマンドの実行動作を調整

## 従来

- ブロッキング・コネクションがあれば、アップグレード処理は実行されずエラーを出力  
→長時間実行するカタログ・スキーマ関連のRMANジョブなどがある場合、アップグレード時間の確保が困難

## 23c

- UPGRADE CATALOGコマンドを以下のオプションと共に実行
- リカバリ・カタログ・アップグレード開始のタイミングを調整可能に

オプション	詳細	例
WAIT <i>n</i>	アップグレード処理実行前の待機時間を指定	UPGRADE CATALOG WAIT 600; ※アップグレード実行前に600秒(10分)待機させる場合
TERMINATE CONNECTED USERS	即時に全てのブロッキング・コネクションを強制終了し、アップグレード処理を実行	UPGRADE CATALOG TERMINATE CONNECTED USERS;
WAIT <i>n</i> TERMINATE CONNECTED USERS	待機時間終了後、全ブロッキング・コネクションを強制終了し、アップグレード処理を実行	UPGRADE CATALOG WAIT 600 TERMINATE CONNECTED USERS;

# RMANバックアップの取得先の拡張

1. デフォルト暗号化アルゴリズムの変更
2. リカバリ・カタログに関する機能拡張
- 3. RMANバックアップの取得先の拡張**
4. Flashback Logを高速リカバリ領域(FRA)外の領域で管理

# Oracle Databaseバージョン固有のRMAN SBTライブラリの同梱

## 従来

- メディア・マネージャとしてSBTライブラリを使用する場合、手動でダウンロードが必要
  - Oracle Secure Backupもしくはサード・パーティ・ソフトウェアは利用可能
- ターゲット・データベース上の指定されたディレクトリに格納されている SBTライブラリを使用するように絶対パスを指定
- ターゲット・データベースのバージョンとの互換性を確保するために、最新のSBTライブラリを定期的にダウンロード

サード・パーティ・ソフトウェアを使用する場合

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS 'SBT_LIBRARY=pathname';
```

## 23c

Oracle Homeディレクトリにネイティブ SBT ライブラリ(SBT\_LIBRARY)が同梱

- Alias(別名)を使用し、Oracle Homeディレクトリから直接ライブラリにアクセスできるようにRMANチャネルをユーザーが設定可能

以下3つをメディア・マネージャとして使用可能に

- Oracle提供のネイティブSBTライブラリ(OCI、RAおよびAmazon S3)
- Oracle Secure Backup
- サード・パーティ・ソフトウェア

Recovery Applianceを使用する場合

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS 'SBT_LIBRARY=oracle.zdlra';
```

# Oracle Databaseバージョン固有のRMAN SBTライブラリのサポート

RMANのバックアップおよびリストアを実行するためのネイティブSBTライブラリ

各ネイティブSBTライブラリには、事前定義されたAlias(別名)を使用

バックアップ先	Alias(別名)	目的
Oracle Cloud Infrastructure(OCI)	<b>oracle.oci</b>	Oracle Cloud Infrastructureでのクラウド・バックアップおよびリストアを可能にするOS固有のSBTライブラリ
Zero Data Loss Recovery Appliance(RA)	<b>oracle.zdlra</b>	Zero Data Loss Recovery Applianceを使用したバックアップおよびリストアを可能にするOS固有のSBTライブラリ。
Amazon S3 Cloud	<b>oracle.osbws</b>	Amazon S3 Cloudを使用したバックアップおよびリストアを可能にするOS固有のSBTライブラリ

## 例：OCIをバックアップ先として指定する場合

oracle.ociは、RMANがOCIへのバックアップおよびリストアを実行をできるようにするSBTライブラリの事前定義済のAlias(別名)

```
RMAN> CONFIGURE CHANNEL DEVICE TYPE sbt
PARMS 'SBT_LIBRARY=oracle.oci SBT_PARMs=(OPC_PFILE=/orclhome/dbs/opc<ORACLE_SID>.ora)' ;
```

## SBT-to-SBT Copy

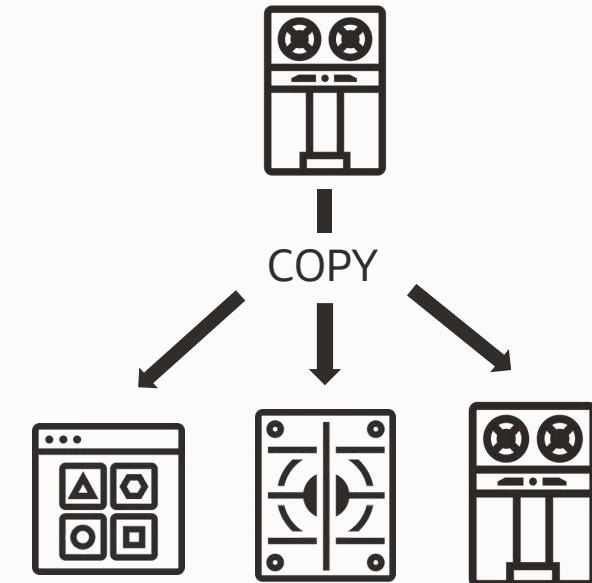
### 従来

- RMANのBACKUP BACKUPSETコマンドを使用すると、ディスク上に作成したバックアップ・セットをバックアップ
  - 複数のメディア間でバックアップを実行する場合に有効
  - バックアップ・セットのバックアップはテープのみに対して可能

### 23c

- 既存のSBTバックアップ(テープやクラウドなど)を他のSBT宛先にコピー可能に
- チャネル、入力デバイス、使用されたライブラリを設定し、バックアップコマンドを発行
- セキュリティを強化するために暗号化スキームを変更可能に

```
RUN {  
  ALLOCATE CHANNEL t1 DEVICE TYPE sbt PARMS = 'SBT_LIBRARY=oracle.oci';  
  INPUT DEVICE PARMS = 'SBT_LIBRARY=oracle.osbws';  
  BACKUP BACKUPSET ALL;  
}
```



# Flashback Logを 高速リカバリ領域(FRA)外の領域で管理

1. デフォルト暗号化アルゴリズムの変更
2. リカバリ・カタログに関する機能拡張
3. RMANバックアップの取得先の拡張
4. **Flashback Logを高速リカバリ領域(FRA)外の領域で管理**

# Flashback Database

指定した時刻までデータベースの状態を戻す

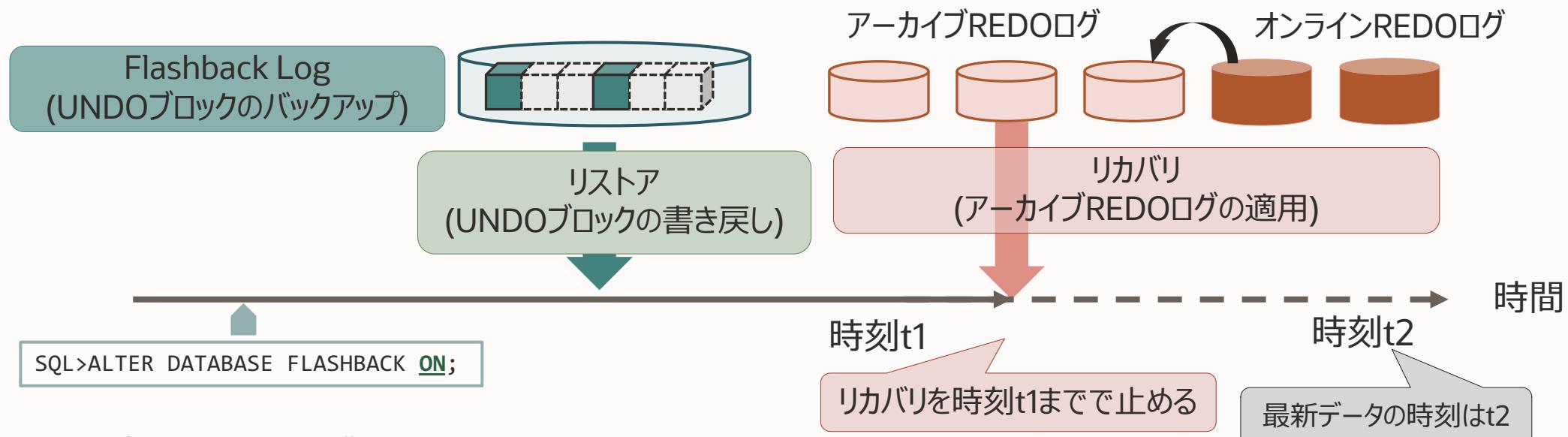
データファイル全体をリストアするポイント・イン・タイムリカバリより迅速で簡単なリカバリ方法

## 事前設定

Flashback Databaseを有効にし、UNDOブロックのバックアップ(Flashback Log)を高速リカバリ領域(FRA)に取得  
リカバリ方法(時刻t1まで戻す場合)

- 戻したい指定地点より前のUNDOブロック(Flashback Log)をリストア
- REDOを適用して指定時刻(時刻t1)までロールフォワード

⇒Flashback Databaseの実行には**Flashback Log**と**アーカイブREDOログ**が必須



# Flashback Logを高速リカバリ領域(FRA)外の領域で管理

## 従来

- 高速リカバリ領域(FRA)内には複数の大容量ファイルが存在
  - 各ファイルが容量を拡大するため、限られたFRA領域の奪い合いが発生
- Flashback LogとアーカイブREDOログはその他のファイルと共に高速リカバリ領域(FRA)内に格納
  - FRAの容量が逼迫すると、容量確保のためにFlashback Logが自動削除される
  - Flashback Log が大量削除されると、Flashback Databaseが実行不可になるので領域管理が必要

## 23c

- Flashback Logの格納先として、**高速リカバリ領域(FRA)以外の領域**を指定可能に
  - FRA領域の奪い合いを回避
  - Flashback Logの領域を確保・管理が容易に**
- 設定には**DB\_FLASHBACK\_LOG\_DEST**と**DB\_FLASHBACK\_LOG\_DEST\_SIZE**を使用
  - Flashback Logの格納先とその領域サイズを明示的に指定



# Oracle Managed Files(OMF)

従来



23c



**Database Area**  
DB\_CREATE\_FILE\_DEST



**Fast Recovery Area(FRA)**  
DB\_RECOVERY\_FILE\_DEST

**NEW** **Flashback Log用領域**  
DB\_FLASHBACK\_LOG\_DEST

# Flashback Logを高速リカバリ領域(FRA)外の領域で管理

## 高速リカバリ領域関連パラメータ

初期化パラメータ	必須	説明
DB_RECOVERY_FILE_DEST	Yes	<ul style="list-style-type: none"><li>リカバリ領域の場所を指定</li><li>ファイル・システム上のディレクトリまたはASMディスク・グループを指定</li></ul>
DB_RECOVERY_FILE_DEST_SIZE	Yes	<ul style="list-style-type: none"><li>ディスク割当て制限を指定</li><li>このデータベースのリカバリ領域で使用されるデータの最大格納量をバイト単位で示したもの</li></ul>
DB_FLASHBACK_RETENTION_TARGET	No	<ul style="list-style-type: none"><li>データベースをFlashbackできる時点の上限(分)を指定</li><li>このパラメータは、Flashback Databaseでのみ必要</li></ul>
DB_FLASHBACK_LOG_DEST	No	<ul style="list-style-type: none"><li>Flashback Logを格納する場所を指定</li><li>Flashback Logを高速リカバリ領域以外のディスクに書き込む場合に指定</li></ul>
DB_FLASHBACK_LOG_DEST_SIZE	No	<ul style="list-style-type: none"><li>Flashback Logの格納先の容量制限を指定</li><li>DB_FLASHBACK_LOG_DESTで指定した領域で使用されるデータの最大格納量をバイト単位で示したもの</li></ul>

# Diagnosability

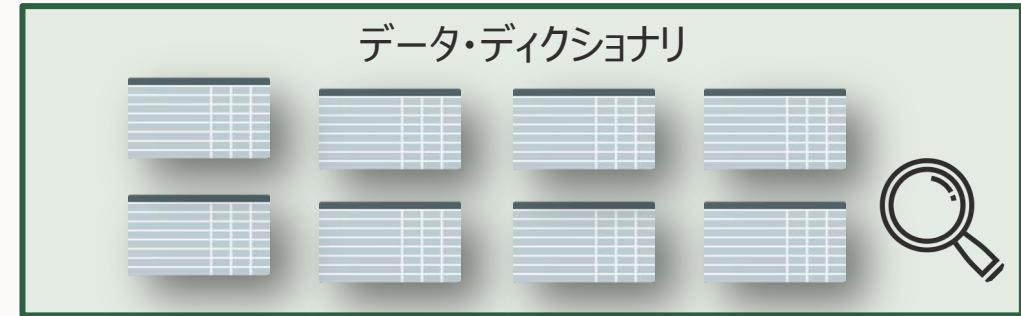
1. PL/SQL DBMS\_DICTIONARY\_CHECK パッケージ

[ ]

O

# PL/SQL DBMS\_DICTIONARY\_CHECK パッケージ

- データ・ディクショナリの不整合により、インスタンスがクラッシュする可能性がある
  - 予期しないエントリ
  - ディクショナリ表間の無効な参照



## DBMS\_DICTIONARY\_CHECK

- データ・ディクショナリの不整合を特定するPL/SQLパッケージ・プロシージャ
- 不整合の検知
- 問題の修復方法
- データベース障害を回避するためのガイド

# DBMS\_DICTIONARY\_CHECK パッケージ

```
DBMS_DEBUG_JDWP
DBMS_DEBUG_JDWP_CUSTOM
DBMS_DEFER_IMPORT_INTERNAL
DBMS_DESCRIBE
DBMS_DG
DBMS_DGPDB
DBMS_DICTIONARY_CHECK
DBMS_DIMENSION
DBMS_DISRUPT
DBMS_DISTRIBUTED_TRUST_ADMIN
DBMS_DLM
DBMS_DM_EXP_INTERNAL
DBMS_DM_MODEL_EXP
DBMS_DM_MODEL_IMP
```

## ディクショナリ表間の無効な参照の例

OBJ\$にないLOBセグメント

OBJ\$にないSOURCE\$のエントリ

OBJ\$-PARTOBJ\$とTABPART\$の間の無効なデータ

所有者のいないセグメント

seg\$にエントリがないマテリアライズド・セグメント

オブジェクト・エントリのないセグメント

recyclebin\$にないごみ箱オブジェクト

Control Seqが制限に近づいているかどうかのチェック

```
1  create or replace NONEDITABLE PACKAGE dbms_dictionary_check IS
2
3      PRAGMA SUPPLEMENTAL_LOG_DATA(default, READ_ONLY);
4
5      Procedure LobNotInObj;
6          -- Check if a lob segment is not in OBJ$
7          -- Note 1360208.1
8
9      Procedure OIDOnObjCol;
10         -- Check if an object type column is not in OID$
11         -- Note 1360268.1
12
13      Procedure SourceNotInObj;
14          -- Check if an entry in SOURCE$ is not in OBJ$
15          -- Note 1360233.1
16
17      Procedure IndIndparMismatch;
18          -- Check for index name mismatch between partitions
19          -- Note 1360285.1
20
21      Procedure InvCorrAudit;
22          -- Check for invalid AUDIT$ entries
23          -- Note 1360489.1
24
25      Procedure OversizedFiles;
26          -- Check for oversized database files
27          -- Note 1360490.1
```

MOS情報連携

# DBMS\_DICTIONARY\_CHECK パッケージの実行例

```
SQL> set serveroutput on size unlimited
SQL> execute dbms_dictionary_check.full
execute dbms_dictionary_check.critical
```

```
dbms_dictionary_check on 01-9月 -2023 10:30:06
```

```
-----
```

```
Catalog Version 23.0.0.0.0 (2300000000)
```

```
db_name: ORCL23C
```

```
Is CDB?: YES CON_ID: 3 Container: ORCL23CPDB1
```

```
Trace File:
```

```
/u01/app/oracle/diag/rdbms/orcl23c/orcl23c/trace/orcl23c_ora_1129373_DICTCHECK.trc
```

Procedure Name	Catalog	Fixed	
Result	Version	Vs Release	Timestamp

```
-----
```

...	...	...	...
..- OIDOnObjCol	...	2300000000 <= *All Rel*	09/01 10:30:06 PASS
..- LobNotInObj	...	2300000000 <= *All Rel*	09/01 10:30:07 PASS
..- SourceNotInObj	...	2300000000 <= *All Rel*	09/01 10:30:07 PASS

プロシージャの実行

```
..- IndexSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- IndexPartitionSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- IndexSubPartitionSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- TableSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- TablePartitionSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- TableSubPartitionSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- PartCol ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- ValidSeg ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- IndPartObj ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
..- DuplicateBlockUse ... 2300000000 <= *All Rel* 09/01 10:30:08 PASS
...
..- ObjNotTrigger ... 2300000000 <= *All Rel* 09/01 10:30:09 PASS
..- InvalidTSMaxSCN ... 2300000000 > 1202000000 09/01 10:30:09 PASS
..- OBJRecycleBin ... 2300000000 <= *All Rel* 09/01 10:30:10 PASS
..- LobSeg ... 2300000000 <= *All Rel* 09/01 10:30:10 PASS
```

```
01-9月 -2023 10:30:10 Elapsed: 4 secs
```

全部 'PASS'

```
Found 0 potential problem(s) and 0 warning(s)
```

異常なし

```
Trace File:
/u01/app/oracle/diag/rdbms/orcl23c/orcl23c/trace/orcl23c_ora_1129373_DICTCHECK.trc
```

```
PL/SQLプロシージャが正常に完了しました。
```



ありがとうございました

