ORACLE DATABASE 12c

Oracle Multitenant：Oracle Database 12c Release 2（12.2）における独立性

Oracle ホワイト・ペーパー | 2017 年 3 月

ORACLE

免責事項

下記事項は、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。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないで下さい。オラクルの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 Database 12c Release 2が利用可能に。

世界でもっとも多く使用されているデータベースの最新世代である Oracle Database 12c Release 2（12.2）が Oracle Cloud で利用可能になり、Oracle Technology Network（OTN）からダウンロードできるようになりました。

目次

独立性

独立性を犠牲にした統合であれば、その統合によるスケールメリットは限定的です。Oracle Database 12c Release 2（12.2）には、独立性を支援する非常に高度な機能がいくつか導入されているため、PDB同士を厳密に分離することができます。これにより、いわゆる"noisy neighbor"（うるさい隣人）と呼ばれる問題を回避できます。また、これは構成可能な機能であるため、ユースケースに合わせて独立性のレベルを適切に設定できる点が優れています。

12.2 では、既存の強力な分離機能をベースにして包括的なモデルを構築しましたが、簡単な構成だけで特定のユースケースに適した独立性のレベルを正確に実現できます。

これらの機能についてこれから詳しく説明しますが、"万能型"アプローチで独立性に対処するのはデータベース・クラウドでは適切でない理由と、本当に必要なのは構成可能な分離機能である理由を理解しておく必要があります。このトピックについて考える場合は、住宅のセキュリティという現実世界でよく聞くたとえを検討すると理解しやすくなります。最初は、セキュリティは厳しいほど良いと考えるかもしれませんが、実際にはセキュリティを優先すると利便性が損なわれます。"厳重なセキュリティ"という言葉を聞くと、住宅よりも刑務所が連想されます。窓に格子を付け、天辺に有刺鉄線を張り巡らせた高い塀で周りを囲み、鋼鉄の扉に錠前を 3 つ付け、武装した見張り番を扉の前に配置すれば、住宅のセキュリティは強化されるでしょう。しかし、これでは決して暮らしやすいとは言えません。逆に、子供たちが出入りしやすいように、扉にも窓にもまったく鍵をかけなかったら、所有物がなくなってしまう可能性があります。人は適切なところでバランスを取ろうとしますが、そのバランスは状況によって異なります。近所に知り合いが多い郊外よりも人口密度の高い都市環境にいる方が警戒心は強くなるはずです。小さな町ではわざわざ鍵をかけたりしないこともあります。誰が何をしているのか、全員が知っているからです。興味深い例として、ビジネス・ホテルのセキュリティについて検討してみましょう。通常、セキュリティは 24 時間体制で、すべての共有スペースに防犯カメラが設置され、警備員が配備され、客室に入るには精巧な鍵が必要です。人は、別の客室の利用者が自分の部屋に入れる可能性があることには非常に不安を覚えますが、ホテルの従業員が日中自分の知らない間に文字どおり何度でも客室に入ることができ（ベッド・メーキングとバスルームの清掃ができている場合は入らないでしょうが）、たいていは夜間に自分がいるときでも客室に入れることはほとんど心配しようとしません。これは興味深いことではありませんか。どういうわけかこのような状況では、ホテルの経営者にセキュリティを任せることにまったく問題を感じません。

似たようなことが、別のユースケースのデータベース・クラウドについても言えます。

パブリック・クラウド上の Database as a Service（DBaaS）では、"隣接する"テナントが競合他社である可能性は十分にあります。各テナントは、自社の PDB 内を強い権限で管理できる必要があります。しかし、隣接する PDB のすべての PDBA からその PDB を完全に分離することも必要です。この 2 つの要件を両立させる必要があるため、これは特に難しいユースケースです。これに適した住居のたとえとして、マンションの所有権の話があります。住民は、自分が所有する空間は完全に自分で管理したいと思います。玄関より内側にあるものはすべてその人が責任を持ちます。

プライベート・クラウド上の DBaaS は開発チームにとって非常に生産性の高い構成になっています。それぞれの開発者は、ある開発者のテストが別の開発者のテストを妨害しない程度に他の開発者から分離されている必要がありますが、通常は共同作業に適した環境で、そこでは誰もが他の開発者の環境を尊重するものと考えられています。これに適したたとえとして、大きな家を友人同士でシェアする例があります。全員が同じ正面玄関の鍵を持っていて、普段は各自の寝室のドアに鍵をかけません。共有スペースと共有設備はいくつかありますが、各自の寝室内のプライバシーはほどほどに守られています。

Software as a Service（SaaS）はホテルでの滞在にたとえることができるでしょう。客室料金を支払っているため維持管理とセキュリティはすべてホテルの経営者に任せます。また、客室係は実質的にいつでも客室に入れるでしょうが、客室内の私物にはそれなりの敬意を払ってもらいます（この場合は通常、客室内の金庫を使用して貴重品を客室係から守ります）。ホテル内に他の宿泊客がいることは誰もが知っていますが、他の客室の利用者は自分の部屋に入ることができないと、十分な根拠を持って見込んでいます。

PDB の独立性はデータベース・クラウドなどの統合度の高い環境では特に重要なトピックですが、一般に、このトピックを検討する場合は、共有することにより発生しうるあらゆるリスクを検討する必要があります。リスクは次のようにいくつかのカテゴリに分類されます。

- 共有コンピューティング・リソースの競合
- システム・アクセス
- ファイル・システム・アクセス
- ネットワーク・アクセス
- 共通ユーザーまたは共通オブジェクト・アクセス
- 管理機能

システムとファイル・システムへのアクセスは PDB レベルの特定のパラメータで管理できますが、その他はロックダウン・プロファイルで保護できます。

システム・アクセスの管理

Oracle OS ユーザーは、通常は特権ユーザーですが、このユーザーをオペレーティング・システムの相互作用に使用することは推奨しません。別の PDB から同じ OS ユーザーをオペレーティング・システムの相互作用に使用すると、特定の PDB に属するデータが侵害される可能性があります。マルチテナント環境では、Oracle OS ユーザーを使用する代わりに、`PDB_OS_CREDENTIAL` 初期化パラメータを使用して、PDB の特定のユーザー・アカウントを OS の相互作用のために指定できます。データベースが extproc エージェントを使って外部プロシージャにアクセスすると、`PDB_OS_CREDENTIAL` によって、PDB からのオペレーティング・システムとのやり取りで使用される OS ユーザーの ID が特定されます。セキュリティを強化するには、マルチテナント環境内の PDB ごとに、一意のオペレーティング・システム・ユーザーを設定する必要があります。`PDB_OS_CREDENTIAL` で識別される OS ユーザーを使用すると、権限の弱いユーザーが OS の相互作用を実行することになり、ある PDB に属するデータに対して、別の PDB に接続しているユーザーがアクセスできなくなります。

PDB のオペレーティング・システム・ユーザーとなる特定のユーザーを設定していない場合、PDB ではデフォルトで `oracle` オペレーティング・システム・ユーザーが使用されます。ルートの場合は、オペレーティング・システムとのやり取りに oracle OS ユーザーを使用できます。

次のようにして、`DBMS_CREDENTIAL.CREATE_CREDENTIAL` プロシージャを使用して、PDB のオペレーティング・システム・ユーザーを設定できます。

1. `DBMS_CREDENTIAL` PL/SQLパッケージの`EXECUTE`権限と、`ALTER SYSTEM`権限を持つユーザーとして、`CDB$ROOT`に接続します。

   次に例を示します。

   ```
   sqlplus c##sec_admin
   Enter password: password
   ```

2. `DBMS_CREDENTIAL.CREATE_CREDENTIAL`プロシージャを実行して、オペレーティング・システム・ユーザー用のOracle資格証明を作成します。

   たとえば、`os_admin`という名前のユーザーに対して資格証明を設定する場合は次のようになります。

   ```
   BEGIN
   DBMS_CREDENTIAL.CREATE_CREDENTIAL (
         credential_name => 'PDB1_OS_USER',
         username => 'os_admin',
         password => 'password');
   END;
   /
   ```

3. オペレーティング・システム・ユーザーを使用する PDB に接続します。
   次に例を示します。

```
CONNECT c##sec_admin@hrpdb
Enter password: password
```

使用可能な PDB を見つけるには、DBA_PDBS データ・ディクショナリ・ビューに問い合わせます。現在の PDB を確認するには、show con_name コマンドを実行します。

4. 手順 2 で資格証明を設定したユーザーの PDB_OS_CREDENTIAL 初期化パラメータを設定します。次に例を示します。

```
ALTER SYSTEM SET PDB_OS_CREDENTIAL = PDB1_OS_USER SCOPE = SPFILE;
```

PDB_OS_CREDENTIAL パラメータは静的パラメータであるため、SCOPE = SPFILE 句を使用して設定する必要があります。

5. PDB を再起動します。

```
ALTER PLUGGABLE DATABASE PDB1 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE PDB1 OPEN;
```

ファイル・アクセスの管理

ここまでは、テナントが他のテナントのデータを保護しながら OS とやり取りできる方法について説明してきました。テナントの独立性における別の重要な原則は、PDB の作成以降に、（たとえばパブリック・クラウド環境の DBaaS などで）ファイル・アクセスを保護することです。その際の主要な目標の 1 つは、PDB の作成時に各テナントのデータ・ファイルとディレクトリ・オブジェクトの一意のパスを設定し、PDB 間の共有ディレクトリをなくして相互のファイルにアクセスできないようにすることです。

CREATE_FILE_DEST句

Oracle Database 12c Release 1（12.1.0.2）以降、CREATE PLUGGABLE DATABASE 文の CREATE_FILE_DEST 句で、PDB のファイルの、デフォルトの Oracle Managed Files（OMF）ファイル・システム・ディレクトリや Oracle Automatic Storage Management（Oracle ASM）ディスク・グループを指定するようになりました。この句を使用すると、CDB のルートで指定する OMF デフォルト・パスとは関係なく、新しい PDB で OMF を有効にできます。PDB のデータ・ファイルと一時ファイルは、指定したディレクトリとそのサブディレクトリに制限されます。ただし、この句を問題なく使用するには、いくつかの事前チェックが必要です。この句でファイル・システム・ディレクトリがデフォルトの場所として指定されている場合は、そのディレクトリが存在することが必要です。また、CREATE PLUGGABLE DATABASE 文を実行するユーザーは、指定したディレクトリにファイルを作成できる適切な権限を持っている必要があります。あるいは、CDB のルート（CDB$ROOT）に存在するディレクトリ・オブジェクトの名前を指定できます。このディレクトリ・オブジェクトは、CREATE_FILE_DEST で使用されるファイル・システム・ディレクトリを指します。OMF のデフォルトの場所が CDB のルートで設定されている場合は、CREATE_FILE_DEST の値がルートの設定より優先されます。また、CREATE_FILE_DEST=NONE と指定すると、PDB で

OMF が無効になります。ルートで OMF を使用している場合、この句を省略すると、PDB ではルートから OMF のデフォルト・パスが継承されます。

たとえば、デフォルトの OMF ディレクトリとして/u01/app/oracle/pdb1/を使用する場合は、次のような構文になります。

```
CREATE PLUGGABLE DATABASE PDB1 ADMIN USER ADMIN IDENTIFIED BY PASSWORD
CREATE_FILE_DEST = '/u01/app/oracle/pdb1/';
```

もう 1 つの例として、CDB$ROOT の既存のディレクトリ・オブジェクトも使用できます。/u02/oracle/pdb/を指す pdb_dir というディレクトリ・オブジェクトがルートに存在すると仮定した場合、PDB を作成してその OMF ディレクトリを/u02/oracle/pdb/に設定する構文は次のようになります。

```
CREATE PLUGGABLE DATABASE PDB2 ADMIN USER ADMIN IDENTIFIED BY
CREATE_FILE_DEST = pdb_dir;
```

PATH_PREFIX句

CREATE PLUGGABLE DATABASE 文で PATH_PREFIX 句を使用すると、PDB に関連付けられるすべてのディレクトリ・オブジェクト・パスを、指定するディレクトリまたはそのサブディレクトリに限定できます。PATH_PREFIX 句は、OMF によって作成されるデータ・ファイル、一時ファイル、またはファイルには適用されません。ユーザーが作成するディレクトリ・オブジェクトにのみ適用されます。また、PDB の Oracle XML リポジトリ、CREATE_PFILE 文で作成するファイル、Oracle ウォレットのエクスポート・ディレクトリはすべて、対応する PDB の指定した PATH_PREFIX ディレクトリに限定されます。この句では、PDB に関連付けられるすべてのファイル・パスの接頭辞として使用される絶対パスか、ルートに存在するディレクトリ・オブジェクトの名前を指定できます（このディレクトリ・オブジェクトは、PATH_PREFIX で使用される絶対パスを指します）。ファイル・パスを制限しない場合は NONE と指定できます。これは、句全体を省略するのと同じです。CREATE_FILE_DEST 句と同様に、PATH_PREFIX 句にも固有の制限事項があります。たとえば、PDB の作成後に PATH_PREFIX の値を変更することはできません。また、PDB 内のすべてのローカル・ディレクトリ・オブジェクトの接頭辞として、PATH_PREFIX の値が必ず追加されます。そのため、これらの制限事項を考慮してローカル・ディレクトリ・オブジェクトを更新し、接頭辞がそれらのオブジェクトの機能に影響しないようにすることが重要です。

たとえば、PDB2 に関連付けられるすべてのファイル・パスの接頭辞として/u01/app/oracle/pdb2/を使用する場合は、次のような構文になります。

```
CREATE PLUGGABLE DATABASE PDB2 ADMIN USER ADMIN IDENTIFIED BY PASSWORD
PATH_PREFIX='/u01/app/oracle/pdb2/';
```

パス名を慎重に指定して、ファイル名がパス名に追加されたときに問題が起こらないようにしてください。たとえば、UNIX システムでは、パス名の末尾にスラッシュ（/）を付ける必要があります。

ロックダウン・プロファイルについて

Multitenant アーキテクチャでは、主要なインフラストラクチャおよびメモリ・コンポーネントの共有によって、スケールメリットを実現できます。ただし、テナントが共有するリソースはこれだけではありません。PDB ではホスト環境だけでなく、OS、ネットワーク、および共通オブジェクトも共有します。特定の権限によってデータベース・ユーザーが PDB をまたいだ操作を実行する可能性があることを考えると、PDB が脆弱性の影響を受ける可能性があります。プライベートまたはパブリックのクラウド環境では特に、テナントの独立性がセキュリティの主要要件です。オラクルでは最近、Multitenant による OS とファイル・システムの相互作用の管理について調査しました。その他の領域（ネットワーク・アクセス、共通オブジェクトへのアクセス、管理機能へのアクセスなど）は、ロックダウン・プロファイルで制御できます。ロックダウン・プロファイルは、12.2 で導入された新しい機能です。

ロックダウン・プロファイルは、PDB での特定の操作や機能を制限するメカニズムです。この新しい Multitenant 機能は CDB 管理者が管理し、特定の PDB でのユーザー・アクセスの制限に使用できます。ロックダウン・プロファイルによって、PDB ユーザーによる次の操作を制限できます。

- `ALTER SYSTEM` や `ALTER SESSION` などの特定の SQL 文の実行
- ネットワークにアクセスするプロシージャ（`UTL_SMTP`、`UTL_HTTP` など）の実行
- 共通ユーザーのオブジェクトへのアクセス
- OS とのやり取り（`PDB_OS_CREDENTIAL` による機能以外）
- CDB 内での無制限の PDB 間接続
- AWR スナップショットの作成
- Java の一部または全体の使用
- Oracle Advanced Queueing や Oracle Partitioning などの特定のデータベース・オプションの使用

ロックダウン・プロファイルの作成

ロックダウン・プロファイルを作成するには、`CREATE LOCKDOWN PROFILE` システム権限を持ち、CDB のルートに接続している必要があります。ロックダウン・プロファイルを作成すると、そのプロファイルに適用する制限事項を追加できます。ロックダウン・プロファイルでは、複数の制限を同時に適用できます。たとえば、ロックダウン・プロファイルを使用すると、PDB でのネットワーク・アクセスと `ALTER SYSTEM` 文の使用を同時に無効にできます。

次の例では、`ALTER SYSTEM` 文の `SET` 句に関連付けられているすべての権限（`CURSOR_SHARING` パラメータ値の変更を除く）を制限する、`sec_profile` というロックダウン・プロファイルの作成方法を示しています。また、このロックダウン・プロファイルによって、PDB では XDB プロトコル（FTP、HTTP、HTTPS）を使用できなくなります。

1. CREATE LOCKDOWN PROFILE システム権限を持つユーザーとして、CDB のルートに接続します。

```
CONNECT c##cdb_admin

Enter password: PASSWORD
```

2. sec_profile というロックダウン・プロファイルを作成します。

```
CREATE LOCKDOWN PROFILE sec_profile;
```

3. ALTER LOCKDOWN PROFILE 文を使用して、プロファイルに制限を追加します。

```
ALTER LOCKDOWN PROFILE sec_profile DISABLE STATEMENT = ('ALTER
SYSTEM') CLAUSE = ('SET') OPTION ALL EXCEPT =
('CURSOR_SHARING');
ALTER LOCKDOWN PROFILE sec_profile DISABLE FEATURE =
('XDB_PROTOCOLS');
```

これは、権限の付与によって有効になる管理機能を制限できる、ロックダウン・プロファイルの一般的なユースケースの 1 つです。権限付与だけでは、すべて可能になるか何もできないかの二者択一です。権限が付与されると、その権限に付随するすべての機能を実行できます。たとえば、ALTER SYSTEM 権限を持つ自律型の PDB 管理者は、権限付与の目的が特定のパラメータの管理だけであっても、その権限で可能なすべての機能を実行できます。上記の例では、ロックダウン・プロファイルによって ALTER SYSTEM 権限の範囲を制限して、特定の操作だけを実行できるようにする方法を示しています。ロックダウン・プロファイルは権限付与を補完するものであり、権限に付随する機能を除外できます。

ロックダウン・プロファイルの有効化

ロックダウン・プロファイルを使用すると、ユーザーが特定の機能にアクセスできなくなったり、プロファイルによって無効化された操作を実行できなくなったりします。ただし、ロックダウン・プロファイルを作成して制限を追加するだけでは、PDB にその制限を適用するには不十分です。ロックダウン・プロファイルを有効にするには、PDB に割り当てる必要があります。そのためには、PDB_LOCKDOWN 初期化パラメータの値をプロファイル名に設定します。パラメータを初めて設定するか別のプロファイルに変更すると、新しいロックダウン・プロファイルがただちに有効になります。ロックダウン・プロファイルは次のように、個々の PDB、または CDB やアプリケーション・コンテナ内のすべての PDB に割り当てることができます。

» CDB のルートとの接続中に PDB_LOCKDOWN パラメータを設定すると、ロックダウン・プロファイルは CDB 内のすべての PDB に適用されます。これは CDB のルートには適用されません。たとえば CDB1 という CDB がある場合は、次のような構文になります。

```
CONNECT sys/password@localhost/CDB1 AS SYSDBA
ALTER SYSTEM SET PDB_LOCKDOWN = sec_profile;
```

» アプリケーションのルートとの接続中に PDB_LOCKDOWN パラメータを設定すると、アプリケーションのルートと、アプリケーション・コンテナ内のすべてのアプリケーション PDB に、ロックダウン・プロファイルが適用されます。たとえば、APP_ROOT という、アプリケーションのルートがある場合は、次のような構文になります。

```
CONNECT sys/password@localhost/APP_ROOT AS SYSDBA
ALTER SYSTEM SET PDB_LOCKDOWN = sec_profile;
```

» PDB との接続中に PDB_LOCKDOWN パラメータを設定すると、ロックダウン・プロファイルはその PDB のみに適用され、CDB またはアプリケーション・コンテナによって適用されるロックダウン・プロファイル（存在する場合）より優先されます。たとえば、PDB3 という PDB がある場合は、次のような構文になります。

CONNECT sys/***password***@localhost/PDB3 AS SYSDBA

ALTER SYSTEM SET PDB_LOCKDOWN = sec_profile;

3 番目の箇条書きが示すように、個々の PDB で設定されるロックダウン・プロファイルの方が優先順位が高く、CDB またはアプリケーションのルートで設定されるその他のロックダウン・プロファイルより優先されます。この機能によって、個々の PDB に対する制限を、必要に応じて柔軟に追加または削除できます。その一方で、CDB レベルまたはアプリケーションのルートレベルのロックダウン・プロファイルによって、多くの PDB に対する制限をまとめて簡単に管理できます。

CDB 内の PDB すべてのロックダウン・プロファイルは、次の方法で有効化できます。

1. 共通のALTER SYSTEMまたはSYSDBA権限を持つユーザーとして、CDBのルートに接続します。

   ```
   CONNECT c##cdb_admin Enter password: password
   ```

2. ALTER SYSTEM SET PDB_LOCKDOWN文を実行します。

   ```
   ALTER SYSTEM SET PDB_LOCKDOWN = sec_profile;
   ```

使用可能なロックダウン・プロファイルの詳細（プロファイル名、ルール、ルールの種類など）は、DBA_LOCKDOWN_PROFILES データ・ディクショナリ・ビューで確認できます。

ロックダウン・プロファイルの削除

ロックダウン・プロファイルの削除は、作成および有効化と同様に、1 つのコマンド操作で実行できます。PDB への制限が動的に変わる可能性があるクラウド環境では特に、ロックダウン・プロファイルの作成、変更、削除を簡単に実行できる必要があります。ロックダウン・プロファイルを削除するには、CDB のルートに接続しており、DROP LOCKDOWN PROFILE システム権限を持っている必要があります。この権限は共通で、または CDB のルートでローカルに付与されます。削除するロックダウン・プロファイルが PDB_LOCKDOWN 初期化パラメータに割り当てられている（つまり使用中である）場合は、ロックダウン・プロファイルを削除すると、ただちにその影響が無効になります。ただし PDB_LOCKDOWN パラメータの値は、削除済みプロファイルの名前のままとなります。CDB のルートで DBA_LOCKDOWN_PROFILES に問い合わせると、既存のロックダウン・プロファイルのリストが表示されます。

ロックダウン・プロファイルは次の方法で削除できます。

1. `DROP LOCKDOWN PROFILE`システム権限を持つユーザーとして、CDBのルートに接続します。

   ```
   CONNECT c##cdb_admin

   Enter password: password
   ```

2. `DROP LOCKDOWN PROFILE`文を実行します。

   ```
   DROP LOCKDOWN PROFILE sec_profile;
   ```

ロックダウン・プロファイルを使用した、PDBでの操作の制限

Oracle Multitenant 12.2 のオンプレミスおよびクラウドのデプロイメントの両方において、ロックダウン・プロファイルは重要な役割を果たすことができます。データベース統合は、テナント間の運用の独立性と適切なリソース割当てを必要とする分野です。ただし個々のテナントにおける独立性の維持とリソースの割当てだけでは、必ずしも十分ではありません。多数のテナントが含まれるプライベートまたはパブリックのクラウド環境の場合は特にそうです。初期に独立性の維持とリソース割当てを適切に行うだけでなく、CDB の使用期間全般にわたり、これら 2 つの条件を満たすことも重要です。つまり、テナントがリソース使用率の制限に違反したり、必要以上の権限を持ったりしないようにする必要があります。先に説明した SaaS のたとえを使うならば、数千部屋の大規模ホテルの管理と似ています。新規の宿泊客は到着時に、自分の部屋と特定の共有エリアに入れるルーム・キーを受け取ります。各部屋は、互いに完全に独立しています。つまり、宿泊客に他の部屋のキーを渡すことはありません。また、VIP や特別会員である宿泊客は、VIP ラウンジや無料のルーム・サービスを利用できる場合もあります。つまり、宿泊客によって利用できるリソースは異なりますが、ポリシーは厳密に順守されます。これは、クラウドにおける CDB の管理に非常に似ています。ホテルの例で説明すると、ホテルの経営者が決定した、宿泊客のアクセス権が、ルーム・キーに組み込まれています。CDB では、テナント PDB に適用されるロックダウン・プロファイルによってこれを実現します。つまり、Oracle Exadata Express Cloud Service はロックダウン・プロファイルのメリットを大いに活用できるクラウド・プラットフォームです。Exadata Express Service のサービス・レベルに基づくさまざまな値を含むロックダウン・プロファイルによって、いくつかのリソース・マネージャ・パラメータ（`CPU_COUNT`、`SESSIONS`、`SGA_TARGET` など）が設定および制限されます。

ロックダウン・プロファイルを使用して、データベースの機能、オプション、SQL 文、および SQL 文の句を制限できます。

ロックダウン・プロファイルによる機能の無効化

`ALTER LOCKDOWN PROFILE` 文の `FEATURE` 句を使用すると、特定のデータベース機能に関連付けられた操作を無効または有効にできます。`FEATURE` 句では、1 つまたは複数の機能名を指定できます。または機能のバンドル名のみを指定して、そのバンドルに含まれるすべての機能のユーザー操作を無効または有効にできます。たとえば、`COMMON_USER_LOCAL_SCHEMA_ACCESS` と `LOCAL_USER_COMMON_SCHEMA_ACCESS` は、機能バンドル `COMMON_SCHEMA_ACCESS` に属する 2 つの機能名です。サポートされるすべての機能、および対応する機能バンドルと操作の説明を、表 1 に示します。

表1 - ロックダウン・プロファイルの機能

機能バンドル	機能	操作
AWR_ACCESS	AWR_ACCESS	PDB で、自動ワークロード・リポジトリ（AWR）スナップショットを自動および手動で作成する
COMMON_SCHEMA_ACCESS	COMMON_USER_LOCAL_SCHEMA_ACCESS	共通ユーザーが、起動者の権限コード・ユニットを起動するか、PDB 内の任意のローカル・ユーザーが所有する BEQUEATH CURRENT_USER ビューにアクセスする
COMMON_SCHEMA_ACCESS	LOCAL_USER_COMMON_SCHEMA_ACCESS	• ANYシステム権限（CREATE ANY TABLEなど）を持つローカル・ユーザーが、共通ユーザーのスキーマ内で、その権限が適用されるオブジェクトを作成するか、オブジェクトにアクセスする。注：LOCAL_USER_COMMON_SCHEMA_ACCESS機能を無効にしても、SYSDBA権限や特定のオブジェクト権限を持つユーザーは、共通ユーザーのスキーマ内でオブジェクトを作成したり、オブジェクトにアクセスしたりすることができます。 • BECOME USER システム権限を持つローカル・ユーザーが共通ユーザーになる • ローカル・ユーザーがALTER USER文を発行して共通ユーザーを変更する • ローカル・ユーザーがプロキシ接続に共通ユーザーを使用する
COMMON_SCHEMA_ACCESS	SECURITY_POLICIES	ローカル・ユーザーが共通オブジェクトに次のような特定のセキュリティ・ポリシーを作成する • データ改訂 • ファイングレイン監査（FGA） • Real Application Security（RAS） • 仮想プライベート・データベース（VPD）
CONNECTIONS	COMMON_USER_CONNECT	共通ユーザーが PDB に直接接続する。この機能が無効な場合に共通ユーザーが PDB に接続するには、まず CDB のルートに接続してから、ALTER SESSION SET CONTAINER 文を使用して接続先の PDB に切り替える必要があります。
CONNECTIONS	LOCAL_SYSOPER_RESTRICTED_MODE_CONNECT	SYSOPER 権限を持つローカル・ユーザーが、RESTRICTED モードで開いている PDB に接続する
CTX_LOGGING	CTX_LOGGING	CTX_OUTPUT.START_LOG や CTX_OUTPUT.START_QUERY_LOG などの Oracle Text PL/SQL プロシージャでロギングを行う
JAVA	JAVA	Java 全体。この機能が無効な場合、Java に依存するデータベースのオプションと機能がすべて無効になります。
JAVA_RUNTIME	JAVA_RUNTIME	java.lang.RuntimePermission を必要とする、Java を介した操作
NETWORK_ACCESS	AQ_PROTOCOLS	Oracle Streams Advanced Queuing（AQ）を介して、HTTP、SMTP、および OCI の通知機能を使用する
NETWORK_ACCESS	CTX_PROTOCOLS	Oracle Text のデータストア・タイプである FILE_DATASTORE と URL_DATASTORE にアクセスする操作 イベント EVENT_INDEX_PRINT_TOKEN および EVENT_OPT_PRINT_TOKEN を使用し、CTX ロギングの一部としてトークンを出力する
NETWORK_ACCESS	DBMS_DEBUG_JDWP	DBMS_DEBUG_JDWP PL/SQL パッケージを使用する
NETWORK_ACCESS	UTL_HTTP	UTL_HTTP PL/SQL パッケージを使用する
NETWORK_ACCESS	UTL_INADDR	UTL_INADDR PL/SQL パッケージを使用する
NETWORK_ACCESS	UTL_SMTP	UTL_SMTP PL/SQL パッケージを使用する
NETWORK_ACCESS	UTL_TCP	UTL_TCP PL/SQL パッケージを使用する
NETWORK_ACCESS	XDB_PROTOCOLS	XDB を介して HTTP、FTP、およびその他のネットワーク・プロトコルを使用する
OS_ACCESS	DROP_TABLESPACE_KEEP_DATAFILES	DROP TABLESPACE 文で INCLUDING CONTENTS AND DATAFILES 句を指定せずに PDB の表領域を削除する
OS_ACCESS	EXTERNAL_GLOBAL_AUTHENTICATION	PDB に外部ユーザーとグローバル・ユーザーを作成する PDB に外部ロールとグローバル・ロールを作成する
OS_ACCESS	EXTERNAL_FILE_ACCESS	PATH_PREFIX が設定されていない場合に、PDB で外部のファイルまたはディレクトリ・オブジェクトを使用する
OS_ACCESS	EXTERNAL_PROCEDURES	PDB で外部プロシージャ・エージェント extproc を使用する
OS_ACCESS	FILE_TRANSFER	DBMS_FILE_TRANSFER パッケージを使用する
OS_ACCESS	JAVA_OS_ACCESS	Java から java.io.FilePermission を使用する
OS_ACCESS	LOB_FILE_ACCESS	データ型 BFILE と CFILE を使用する

OS_ACCESS	TRACE_VIEW_ACCESS	次のトレース・ビューを使用する • [G]V$DIAG_OPT_TRACE_RECORDS • [G]V$DIAG_SQL_TRACE_RECORDS • [G]V$DIAG_TRACE_FILE_CONTENTS • V$DIAG_SESS_OPT_TRACE_RECORDS • V$DIAG_SESS_SQL_TRACE_RECORDS
OS_ACCESS	UTL_FILE	UTL_FILE を使用する。この機能が無効な場合、データベースで UTL_FILE.FOPEN ファンクションの使用がブロックされます。

次の例では、機能バンドル NETWORK_ACCESS のすべての機能を無効にしています。

```
ALTER LOCKDOWN PROFILE sec_profile DISABLE FEATURE = ('NETWORK_ACCESS');
```

次の例では、COMMON_USER_LOCAL_SCHEMA_ACCESS と LOCAL_USER_COMMON_SCHEMA_ACCESS 以外のすべての機能を無効にしています。

```
ALTER LOCKDOWN PROFILE sec_profile DISABLE FEATURE ALL EXCEPT =
('COMMON_USER_LOCAL_SCHEMA_ACCESS', 'LOCAL_USER_COMMON_SCHEMA_ACCESS');
```

ロックダウン・プロファイルによるデータベース・オプションの無効化

ALTER LOCKDOWN PROFILE 文の OPTION 句を使用すると、特定のデータベース・オプションを無効または有効にできます。データベース機能の制限と同様に、サポートされているデータベース・オプションを、ALL 句を指定してまとめて、または ALL EXCEPT 句を使用して部分的に無効にできます。ALL EXCEPT 句を使用すると、SQL 文で指定したデータベース・オプションを除く、すべてのサポート対象データベース・オプションが無効になります。現在、Oracle Database Advanced Queueing と Oracle Partitioning は、ロックダウン・プロファイルでオンまたはオフにできるオプションです。

次の例では、Oracle Partitioning オプションに関連付けられているユーザー操作を無効にしています。

```
ALTER LOCKDOWN PROFILE sec_profile DISABLE OPTION = ('PARTITIONING');
```

次の例では、Oracle Database Advanced Queueing オプションに関連付けられているユーザー操作を有効にしています。

```
ALTER LOCKDOWN PROFILE sec_profile ENABLE OPTION = ('DATABASE QUEUEING');
```

ロックダウン・プロファイルによるSQL文および文の句の無効化

ロックダウン・プロファイルを使用して、特定の SQL 文の範囲を制限することもできます。この機能を使用すると、適切な権限を持つユーザーがミッション・クリティカルな操作を実行することを制限できるので便利です。そのためには、ロックダウン・プロファイルによって、ALTER DATABASE、ALTER PLUGGABLE DATABASE、ALTER SESSION、および ALTER SYSTEM の文を任意の組み合わせで無効にします。ただし、これらの文を完全に無効にはしたくない場合は、これらの文に含まれる特定の句のみを無効にすることもできます。そのためには、1 つの句を明確に特定するのに十分なキーワードを指定する必要があります。たとえば、ALTER SYSTEM 文の ARCHIVE LOG 句を無効にするには ARCHIVE を指定すれば十分ですが、ALTER SYSTEM 文の FLUSH SHARED_POOL 句を無効にするために FLUSH を指定するのは不適切です。このキーワードで始まる複数の ALTER SYSTEM 文（ALTER SYSTEM FLUSH GLOBAL CONTEXT など）が存在するためです。

また、ALTER SYSTEM 文または ALTER SESSION 文の SET 句に限り、指定するオプションのデフォルトの最小値または最大値を設定できます。VALUE 句を使用すると、句のオプションのデフォルト値を設定して、プロファイルの適用先の PDB を閉じてもう一度開いた後に、PDB でそのデフォルト値を有効にできます。ただし、有効にしているロックダウン・プロファイルのルールに VALUE 句が含まれていない場合は、PDB を閉じてもう一度開く必要はありません。前の項で説明したとおり、プロファイルはただちに有効になるためです。この句の目的は、オプションのデフォルト値を同時に設定して、ユーザーによるこの値の設定または変更を制限することです。一方、MINVALUE 句と MAXVALUE 句を使用すると、これより小さい、または大きいオプション句の値を設定することがそれぞれ制限されます。MINVALUE と MAXVALUE の設定は、ロックダウン・プロファイルが PDB に割り当てられるとただちに有効になります。PDB を閉じて開き直す必要はありません。

構文とこれらの機能の動作については、次の例を参照してください。

» ALTER DATABASE 文を無効にする

```
ALTER LOCKDOWN PROFILE sec_profile DISABLE STATEMENT = ('ALTER DATABASE');
```

» ALTER SYSTEM SUSPEND 文と ALTER SYSTEM RESUME 文を無効にする

```
ALTER LOCKDOWN PROFILE sec_profile
  DISABLE STATEMENT = ('ALTER SYSTEM')
  CLAUSE = ('SUSPEND', 'RESUME');
```

» ALTER SESSION 文の COMMIT_WAIT パラメータと CURSOR_SHARING パラメータを無効にする

```
ALTER LOCKDOWN PROFILE sec_profile
  DISABLE STATEMENT = ('ALTER SESSION')
  CLAUSE = ('SET')
  OPTION = ('COMMIT_WAIT', 'CURSOR_SHARING');
```

» ALTER SESSION 文の PDB_FILE_NAME_CONVERT パラメータを無効にする。また、PDB_FILE_NAME_CONVERT のデフォルト値を'cdb1_pdb0'、'cdb1_pdb1'に設定する。このデフォルト値は、次に PDB を閉じてもう一度開くと有効になります。

```
ALTER LOCKDOWN PROFILE sec_profile
  DISABLE STATEMENT = ('ALTER SYSTEM')
  CLAUSE = ('SET')
  OPTION = ('PDB_FILE_NAME_CONVERT')
  VALUE = ('cdb1_pdb0', 'cdb1_pdb1');
```

» ALTER SESSION 文の CPU_COUNT パラメータを、2 未満または 6 より大きい値の場合に無効にする。

```
ALTER LOCKDOWN PROFILE sec_profile
  DISABLE STATEMENT = ('ALTER SYSTEM')
  CLAUSE = ('SET')
  OPTION = ('CPU_COUNT')
  MINVALUE = '2'
  MAXVALUE = '6';
```

結論

Oracle Multitenant 12.2 では、統合によるスケールメリットと、テナント間の優れた独立性を同時に実現できます。PDB ごとにリソースや運用の要件が異なる可能性があることを考慮すると、PDB ごとに独立性レベルをカスタマイズできることが、12.2 での主要な原則の 1 つになります。12.1 は非常に高機能な製品ですが、12.2 ではさらに、テナントの独立性が大幅に機能拡張されています。

第 1 世代のクラウド・アーキテクチャでは個々のデータベースが専用の VM でホストされるため、データベースの数とコストの関係は一次関数的ですが、Oracle Multitenant はこれとは異なり、Database Cloud に真のスケールメリットをもたらします。規模が大きくなるほどスケールメリットも大きくなるため、この可能性を現実のものにするには、統合の障害をすべて排除することが不可欠です。その主要な要素の 1 つがテナントの独立性です。高度に統合された環境では、テナントは主要なインフラストラクチャやメモリ・コンポーネントだけでなく、ネットワーク、OS、ファイル・システム、およびデータベースの共通オブジェクトも共有します。`PDB_OS_CREDENTIAL` パラメータを使用して、権限の弱い OS ユーザーを指定し、OS の相互作用の潜在的なリスクを軽減します。また、CDB 管理者は `CREATE_FILE_DEST` 句と `PATH_PREFIX` 句を使用して、PDB のデータ・ファイルとディレクトリ・オブジェクトの PDB 固有のパスをそれぞれ設定できます。これら 2 つの機能の他に、12.2 以降ではロックダウン・プロファイルという優れた機能も追加されています。ロックダウン・プロファイルは基本的に、PDB での特定の操作を制限するセキュリティ・メカニズムです。ロックダウン・プロファイルでは、`ALTER SYSTEM` や `ALTER SESSION` などの強力な権限の範囲を制限できます。また、特定のリソースや管理機能へのアクセスを無効にできるため、CDB の管理が非常に簡単になります。テナントの独立性に関するこれらすべてのソリューションを組み合わせれば、世界規模のデータベース・クラウド・アーキテクチャの展開に役立ちます。

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

海外からのお問い合わせ窓口
電話：**+1.650.506.7000**
ファクシミリ：**+1.650.506.7200**

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Oracle Multitenant：Oracle Database 12c Release 2（12.2）における独立性 2017 年 3 月
著者：Can Tuzla
共著者：John P. McHugh、Prashanth Shanthaveerappa、Patrick Wheeler