



EDB

EDB Migration Portal

リリース 4.5.0

EnterpriseDB Corporation

2023 年 05 月 25 日

目次

第 1 章	Release notes	3
1.1	Version 4.5	3
1.1.1	サポート終了のお知らせ	4
1.2	Version 4.4	5
1.2.1	サポート終了のお知らせ	6
1.3	Version 4.3	6
1.4	Version 4.2	7
1.5	Version 4.1	8
1.6	Version 4.0	9
第 2 章	Supported platforms	11
2.1	対応ブラウザ	11
第 3 章	Using Migration Portal	13
3.1	Overview of the Migration Portal home page	14
3.2	Overview of the Migration Portal Projects page	15
3.3	Migration Portal quick start	16
第 4 章	Migrating a database	19
4.1	Performing a schema extraction	19
4.1.1	EDB DDL エクストラクタを使用したスキーマの抽出	19
4.1.2	スキーマとオブジェクトのサポート	22
4.2	Performing a schema assessment	27
4.2.1	スキーマ評価を実行する	28
4.2.2	評価レポートを生成する	31
4.2.3	評価レポートを評価する	32
4.3	Performing a schema migration	33
4.3.1	オフライン移行	34
4.3.2	オンライン移行	36
4.4	Performing a data migration	38
4.4.1	高度なデータ移行	38
第 5 章	Known issues, limitations, and notes	39
5.1	Migration Portal	39
5.1.1	ラップされたオブジェクト	39
5.1.2	ファイルエンコーディング	39
5.1.3	ALTER ステートメント	40
5.1.4	ユーザー、ロール、プロファイル、および付与	40
5.1.5	付与	40
5.1.6	特権	40
5.1.7	パスワード	40

5.1.8	プロファイル	41
5.1.9	Oracle のデフォルトケースを使用	41
5.2	EDB DDL エクストラクタ	44
5.3	Oracle Data Pump ユーティリティ	45

Migration Portal は、Oracle データベーススキーマを EDB Postgres Advanced Server プラットフォームに移行するための Web ベースのサービスです。Migration Portal は、Oracle データベーススキーマを評価および分析し、以下を変換します。

- テーブル
- シーケンス
- 制約
- インデックス（LOB インデックスとマテリアライズドビューのインデックスを除く）
- 同義語
- ビュー
- マテリアライズドビュー
- DB リンク
- タイプとタイプボディ
- トリガー
- 機能
- 手続き
- パッケージ
- ユーザー
- ロール
- プロフィール
- ロールとオブジェクトの付与

Migration Portal は、EDB Postgres Advanced Server と互換性のあるデータ定義言語（DDL）ステートメントを生成します。

使いやすいポータルインターフェイスにより、評価と移行のプロセスが簡素化されます。ポータルにログインし、migration process を起動します。

EnterpriseDB は、企業が既存のデータベースシステムを長年 Postgres に移行するのを支援してきました。詳細については、

[EnterpriseDB website](#) を参照してください。

第1章 Release notes

Migration Portal のドキュメントには、Migration Portal 4.0 の最新バージョンについて説明されています。

Version	Release date
<i>Version 4.5</i>	18 May 2023
<i>Version 4.4</i>	14 Feb 2023
<i>Version 4.3</i>	08 Dec 2022
<i>Version 4.2</i>	04 Oct 2022
<i>Version 4.1</i>	04 Aug 2022
<i>Version 4.0</i>	22 Apr 2022

1.1 Version 4.5

Migration Portal 4.5 の新機能、機能強化、バグ修正、およびその他の変更には以下が含まれます。

Type	Description
Enhancement	移行ポータルに移行の複雑さと機能の分析機能は、以前はキャプチャされなかった次の Oracle 機能の存在をキャプチャしてレポートするようになりました。
Enhancement	EDB DDL エクストラクタユーティリティは、スキーマを抽出するときに必要な場合のみ `GRANT` ステートメントを抽出するオプションを提供するようになりました。以前は、`GRANT` ステートメントは常に抽出されました。これで、それらを抽出することを選択できます。デフォルトでは、`GRANT` ステートメントは抽出されません。
Enhancement	Migration Portal は、プロジェクトが通常のサポートの終わりに近づいていることを示し、ターゲットデータベースのサポートが Migration Portal から削除されるときにプロジェクトへのさらなる更新を制限することを示します。
Enhancement	Migration Portal には、プロジェクトの作成日と、プロジェクトが作成されたときに使用された Migration Portal バージョンが、Migration Portal ユーザーインターフェイスの Overview セクションに表示されるようになりました。
Enhancement	Migration Portal は、ターゲットデータベースバージョン 11~14 の互換性のないファンクションとして `SQLERRM()` ファンクションを検出するようになりました。

次のページに続く

表 2 – 前のページからの続き

Type	Description
Enhancement	Migration Portal は、新しい修復ハンドラー（ERH-2103）で機能強化されました。これにより、 <code>=ANY`オペレーター`の`IN`および!= ANY`オペレーター`の`NOT IN`</code> の使用が置き換えられます。
Enhancement	修復ハンドラー ERH-2100 は、EDB Postgres Advanced Server でサポートされていない <code>`WRITE ON DIRECTORY`</code> 付与特権を検出および削除するようになりました。
Bug fix	<code>`IMPORT FULL`</code> や <code>`EXPORT FULL`</code> などのサポートされていないシステム許可が Migration Portal によって削除されなかった問題を修正しました。
Bug fix	<code>`DBMS_UTILITY.FORMAT_CALL_STACK`</code> および <code>`DBMS_UTILITY.LNAME_ARRAY`</code> 関数は、EDB Postgres Advanced Server でサポートされているにもかかわらず、互換性がないとして誤ってフラグが付けられていた問題を修正しました。
Bug fix	別のロールに割り当てられているロールの <code>`CREATE ROLE`</code> ステートメントが抽出されない問題を修正しました。
Bug fix	プロジェクトのエクスポート <code>sql</code> ファイルから不要な <code>`SET SEARCH_PATH=GLOBAL_OBJECTS`</code> ステートメントを削除しました。
Bug fix	特定のユースケースのファイル解析フェーズ中に Migration Portal の評価プロセスでエラーが発生する問題を修正しました。
Bug fix	Oracle Data Pump ユーティリティによって抽出され、アップロードされたソース DDL に含まれるネストされたテーブルが解析エラーを引き起こしていた問題を修正しました。EDB Postgres Advanced Server は、ネストされたテーブルをサポートしていません。その結果、Migration Portal による処理と評価の前に、ソース DDL から削除されます。
Security fix	Migration Portal が使用する <code>ejs</code> ライブラリ（Node.js 用）が更新され、 CVE-2022-29078 および CVE-2021-23566 で特定されたセキュリティの脆弱性に対処します。
Security fix	Migration Portal で使用される <code>terser</code> （NPM）ライブラリが更新され、 CVE-2022-25858 で特定されたセキュリティの脆弱性に対処します。

1.1.1 サポート終了のお知らせ

EDB Postgres Advanced Server（EPAS）バージョン 11 の標準サポートの終了は 2023 年 11 月 20 日に予定されているため、2023 年以降にサポートされる EPAS のより新しいバージョンを新しい Migration Portal プロジェクトに選択する必要があります。EPAS 11 のサポートは、2023 年 5 月 20 日に Migration Portal から削除され、この日以降に新しい EPAS 11 プロジェクトが作成されないようにする予定です。

利用可能なサポート期間を最大化し、新機能、特に Oracle との互換性を強化する機能を活用するために、EDB は最新の EPAS バージョンを Migration Portal プロジェクトのターゲットとして選択することをお勧めします。EDB Postgres Advanced Server の各メジャーバージョンのサポート期間の詳細については、EDB

[Platform Compatibility](#) ページを参照してください。

1.2 Version 4.4

Migration Portal 4.4 の新機能、機能強化、バグ修正、およびその他の変更には次が含まれます。

Type	Description
Enhancement	Migration Portal は、EDB Postgres Advanced Server の最新メジャーバージョンである EDB Postgres Advanced Server EDB Postgres Advanced Server バージョン 15 をサポートするようになりました。
Enhancement	Migration Portal が拡張され、移行プロジェクト内のオブジェクト名の Oracle のデフォルトの大文字と小文字を保持するオプションが提供されるようになりました。Postgres では、オブジェクト名のデフォルトの大文字小文字は小文字です。Oracle では、大文字です。ほとんどの場合、データベースを Postgres に移行するときは Postgres のデフォルトケースを使用する必要があります。ただし、この新しいオプションは、データベースクエリで引用符で囲まれた大文字の識別子を特に使用するアプリケーションのサポートを向上させるために提供されています。新しい機能により、すべてのオブジェクトを引用符で囲まれた大文字の名前で作成できますが、初期実装では手続きコードで指定されたオブジェクト名参照は更新されないことに注意してください。
Enhancement	Migration Portal の移行の複雑さと機能分析機能が拡張され、以前にキャプチャされなかった特定の Oracle 機能の存在をキャプチャおよびレポートするようになりました。これには、外部結合の条件としての `BETWEEN` 句の使用に対応する `OUTER JOIN WITH BETWEEN` クエリで。
Enhancement	Migration Portal は、新しい ERH-3008 修復ハンドラーで拡張されました。これは、`CREATE VIEW` ステートメントから `CONSTRAINT` 句を削除します。この機能は EDB Postgres Advanced Server でサポートされていないためです。この機能強化の一環として、スキーマ評価レポートの機能分析セクション関連の機能が更新され、「既存の利用可能な回避策のある機能」のカテゴリで `VIEW COLUMN CONSTRAINT` の発生を特定および報告するようになりました。
Enhancement	ERH-2057 リペアハンドラーが拡張され、`LONG` データ型から精度部分が削除されました。
Bug fix	EDB DDL エクストラクタが更新され、以前にフェッチされていなかった `ROLE` から `ROLE` の付与が抽出されました。
Bug fix	日付リテラルを指定するために使用される場合、`VIEW` の DDL が `DATE` キーワードの後に切り捨てられていた問題を修正しました。
Bug fix	列名として `EXTENDED` で指定されたテーブル列の `DEFAULT` 句が削除される問題を修正しました。
Bug fix	ユーザーがオブジェクトを手動で修復した後、評価レポートに誤ったオブジェクト数が表示される問題を修正しました。
Bug fix	Oracle `PIPE ROW` 構文から Postgres `RETURN NEXT` 構文に変換した後、スキーマ名プレフィックスで指定された行型オブジェクトからスキーマ名が誤って削除される ERH-2059 修復ハンドラーの問題を修正しました。
Other	EDB Postgres Advanced Server でサポートされていない、単一の引数を持つ `TO_TIMESTAMP_TZ()` ファンクションコールを移行するための提案された回避策を含む Migration Portal ナレッジベースエントリを追加しました。

次のページに続く

表 3 – 前のページからの続き

Type	Description
Other	EDB Postgres Advanced Server バージョン 10 のサポートが削除されました。ユーザーは、新しい移行プロジェクトのターゲットデータベースとして EDB Postgres Advanced Server 10 を選択できなくなりました。ユーザーは、新しいスキーマをアップロードしたり、既存の EDB Postgres Advanced Server 10 プロジェクトのオブジェクトを編集および再評価したりすることもできなくなりました。

1.2.1 サポート終了のお知らせ

EDB Postgres Advanced Server (EPAS) バージョン 11 の標準サポートの終了は 2023 年 11 月 20 日に予定されているため、2023 年以降にサポートされる EPAS のより新しいバージョンを新しい Migration Portal プロジェクトに選択する必要があります。EPAS 11 のサポートは、2023 年 5 月 20 日に Migration Portal から削除され、この日以降に新しい EPAS 11 プロジェクトが作成されないようにする予定です。

利用可能なサポート期間を最大化し、新機能、特に Oracle との互換性を強化する機能を活用するために、EDB は最新の EPAS バージョンを Migration Portal プロジェクトのターゲットとして選択することをお勧めします。EDB Postgres Advanced Server の各メジャーバージョンのサポート期間の詳細については、EDB

[Platform Compatibility](#) ページを参照してください。

1.3 Version 4.3

Migration Portal 4.3 の新機能、機能強化、バグ修正、およびその他の変更には次が含まれます。

Type	Description
Enhancement	Migration Portal は、ソース Oracle データベースからのユーザー、ロール、プロファイル、および権限の評価と移行をサポートするように拡張されました。
Enhancement	Migration Portal が更新され、ユーザー、ロール、プロファイル、および許可の抽出のサポートを提供する EDB DDL エクストラクタの新しいバージョンへのリンクを提供します。Migration Portal は、古いバージョンの EDB DDL エクストラクタで生成された DDL を引き続き処理および評価できます。ただし、新しくサポートされたユーザー関連の情報は、それらの古いバージョンでは抽出できません。
Enhancement	Migration Portal スキーマ移行の複雑さのスコアリングと機能分析機能が強化され、以前のリリースの場合のようにバージョン 14 だけでなく、選択したターゲット EDB Postgres Advanced Server バージョンに基づいたスコアと機能分析を提供します。
Enhancement	スキーマをターゲットデータベースに移行するための Migration Portal ユーザーインターフェイスが簡素化され、手順数が削減され、冗長オプションでオフラインおよびオンライン移行モードが有効になりました。

次のページに続く

表 4 – 前のページからの続き

Type	Description
Enhancement	Migration Portal の移行の複雑さと機能分析機能が拡張され、次の機能を含む、以前はキャプチャされなかった Oracle 機能の存在をキャプチャおよびレポートするようになりました。 <i>CONNECT BY CONDITION</i> 、 <i>CONNECT BY NOCYCLE</i> 、およびビルトイン/SYS 所有のファンクション <code>GROUP_ID</code> 。
Security fix	Migration Portal で使用される jackson-databind ライブラリが更新され、 CVE-2022-42003 および CVE-2022-42004 で特定されたセキュリティの脆弱性に対処します。
Bug fix	シーケンスの <code>MAXVALUE</code> の処理を修正して、警告を回避し、ターゲット EDB Postgres Advanced Server データベースでサポートされている <code>MAXVALUE</code> に許可される最大値に関する問題を解決しました。
Bug fix	Oracle Data Pump ユーティリティを使用して生成された抽出 DDL で見つかった仮想テーブル列に関連する <code>ALTER TABLE MODIFY</code> ステートメントの処理の問題を修正しました。
Bug fix	<code>PIPELINED</code> 句がパッケージ関数定義内で使用されている場合、 <code>PIPELINED</code> 句機能が報告されない問題を修正しました。
Bug fix	特定の場合に互換性のある機能でカタログビューが検出および報告されないことに関連する問題を修正しました。

1.4 Version 4.2

Migration Portal 4.2 の新機能、機能強化、バグ修正、およびその他の変更には以下が含まれます。

Type	Description
Enhancement	4.1.0 リリースより前に作成された既存のプロジェクトのスキーマ移行の複雑さスコアを計算し、機能分析を生成するためのサポートが追加されました。
Enhancement	オブジェクトを手動で修復した後のスコアの更新に対応するために、スキーマ移行の複雑さのスコアリング機能が強化されました。
Enhancement	スキーマ評価レポートの機能分析セクションに、未解決のエラーがあるオブジェクトと、それらのオブジェクトで使用されている互換性のない Oracle 機能のリストをリストする新しい「未解決のオブジェクト」表を追加しました。
Enhancement	多数の Oracle 機能を含むプロジェクトのスキーマ評価レポート生成のパフォーマンスが最大 3 倍向上しました。
Bug Fix	互換性のないビルトインパッケージと SYS スキーマ TYPE オブジェクトの使用をキャプチャし、より正確にレポートするように Migration Portal を修正しました。
Bug Fix	次の Oracle 機能の存在をより正確にキャプチャし、スキーマ移行の複雑さへの影響を評価するように Migration Portal を修正しました。USER DEFINED EXCEPTION RISING、USER DEFINED EXCEPTION HANDLING、XMLAGG、XMLQUERY、XMLTABLE、XMLPI、XMLPARSE、および XMLROOT。

次のページに続く

表 5 – 前のページからの続き

Type	Description
Bug Fix	DDL エクストラクタの新しいバージョンを作成して、CACHE キーワードで定義された Oracle テーブルのエントリが DDL ファイルに出力される問題を修正しました。Migration Portal が更新され、新しいバージョンの DDL エクストラクタへのリンクが提供されました。Migration Portal は、古いバージョンの DDL エクストラクタによって生成された DDL を引き続き処理および評価できます。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、PRAGMA EXCEPTION_INIT 宣言が互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	「機能分析」セクションで LONG RAW が互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、ユーザー定義の例外宣言が互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、BULK COLLECT INTO 句が互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、NULLIF 関数が互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、REGEXP_LIKE 句の回避策のステータスを「None / Needs Investigation」から「Exists」に更新しました。修復ハンドラーは現在利用可能であり、REGEXP_LIKE 句に自動的に適用されます。
Bug Fix	EDB Postgres Advanced Server でサポートされ、サポートされていない Oracle の事前定義 PL / SQL 例外のリストを更新しました。
Bug Fix	ユーザーがオブジェクト（スキーマ、テーブル、ビューなど）のサブセットを選択し、ウィザードページ間を移動すると、ターゲット DDL オブジェクト選択ページでのユーザー選択状態が失われる問題を修正しました。

1.5 Version 4.1

Migration Portal 4.1 の新機能、機能強化、バグ修正、およびその他の変更には以下が含まれます。

Type	Description
Enhancement	Migration Portal スキーマ評価レポートには、アップロードされた DDL で見つかった Oracle コンストラクト（つまり、機能）に関連するエフォートタイプと複雑さレベルに基づいて移行の複雑さスコアを提供する新しい機能分析セクションが含まれるようになりました。この新しいセクションでは、DDL で見つかった互換性のない機能の概要も提供します。Feature Analysis セクションは現在、新しい移行プロジェクトに対してのみ生成され、このリリースより前に作成された既存のプロジェクトに対しては生成されないことに注意してください。

次のページに続く

表 6 – 前のページからの続き

Type	Description
Enhancement	ユーザーは、プロジェクト内の使用可能なスキーマと基になるオブジェクトにフィルタリングを適用して、選択したオブジェクトタイプのサブセットのターゲットデータベース DDL を生成できるようになりました。さらに、ダウンロード用に単一の DDL SQL ファイルが作成される代わりに、zip アーカイブが作成されます。これには、選択したオブジェクトタイプごとに個別のファイルと、プロジェクトと選択したスキーマの統合スクリプトが含まれます。
Enhancement	Q 引用符で囲まれた文字列をドル引用符または E 引用符で囲まれた文字列に変換するための新しい修復ハンドラー ERH-3005 が追加されました。
Enhancement	ERH-2078 リペアハンドラーを強化して、パッケージヘッダーレベルでパイプライン化された特定の変換を適用します。
Enhancement	ERH-2034 リペアハンドラーを更新して、パッケージ内で提供される TYPE 定義から NULL を削除します。
Security fix	セキュリティ問題（ CVE-2021-43138 、 CVE-2022-1650 、 CVE-2022-23221 、 CVE-2021-42392 、 CVE-2021-20066 、および CVE-2022-31129 ）に対処するために、 <code>async</code> 、 <code>eventsourcing</code> 、 <code>h2</code> 、 <code>jsdom</code> 、および <code>moment</code> ライブラリを最新バージョンにアップグレードしました。
Bug fix	Oracle Data Pump ユーティリティを使用して生成されたソース Oracle DDL ファイルを処理するときに、一部のインデックスがスキップされるコーナーケースを修正しました。
Bug fix	ALTER TABLE ステートメントの ENABLE NOVALIDATE 句から ENABLE キーワードのみが削除される動作を修正し、完全な句が削除されるようになりました。また、関連するリペアハンドラー ERH-2003 および ERH-2005 を更新しました。
Bug fix	長時間実行されるスキーマ評価と並行して実行される他の評価のパフォーマンスに影響を与えていた同期スレッドの問題を修正しました。
Bug fix	Oracle Data Pump ユーティリティによって生成される CREATE TABLE ステートメントの DEDUPLICATE LOB 句により、DDL 解析関連のエラーが発生する問題を修正しました。

1.6 Version 4.0

Migration Portal 4.0 の新機能、機能強化、バグ修正、およびその他の変更には以下が含まれます。

Type	Description
Enhancement	Migration Portal は、EDB DDL エクストラクタの使用を再度サポートするようになりました。DDL エクストラクタに関連する問題が解決され、DDL エクストラクタファイル形式の使用に対して導入された一時的な制限が削除されました。Migration Portal は、 documentation の指示に従って、Oracle Data Pump ユーティリティの使用を引き続きサポートします。ただし、EDB DDL エクストラクタの使用をお勧めします。
Enhancement	スキーマ評価アルゴリズムは、パフォーマンス関連のいくつかの問題に対処し、全体的なスキーマ評価時間を短縮するように最適化されています。
Enhancement	EDB Postgres Advanced Server はサポートしていないため、TABLE 定義から CACHE / NOCACHE キーワードを削除するための修復ハンドラーを追加しました。
Security fix	jackson-databind および h2 ライブラリは、セキュリティ問題 (CVE-2020-36518、CVE-2022-23221、CVE-2021-42392、および CVE-2021-23463) に対処するために最新バージョンにアップグレードされました。
Bug fix	文法規則を修正して、修復ハンドラーの実行をスキップして EDITIONABLE キーワードを削除する解析エラーを解決しました。
Bug fix	Oracle PIPELINED および PIPE ROW ファンクションコールの適切な変換を実行するように修復ハンドラーを修正しました。
Bug fix	オブジェクト名が Unicode 文字に基づいている場合、大文字と小文字の区別が失われる問題を修正しました。
Bug fix	新しいユーザーに対して Migration Portal のランディングページを初めて開いたときに、はじめにウィザードが表示される問題を修正しました。
Bug fix	ターゲット EPAS DDL スクリプトでの変換中に VIEW 定義が切り捨てられる問題を修正しました。
Bug fix	以前は、UTF-8 と互換性のない文字を含むアップロードされた DDL ファイルは、評価フェーズの開始時に失敗していました。Migration Portal は、SQL ファイルを UTF-8 として読み取るようになりました。UTF-8 以外のファイルをアップロードした場合、UTF-8 非対応文字は全て置換文字「 □ 」に変換され、問題なく評価が継続されます。
Bug fix	ソース Oracle DDL の名前のない制約の制約名の割り当てを修正しました。
Bug fix	修復ハンドラーを更新して、Oracle Data Pump で生成されたソース DDL ファイルの OIDINDEX ストレージ句を削除しました。
Bug fix	COLLATE および COMPUTATION キーワードを識別子として使用できるようになりました。
Bug fix	単一行コメントの一部として表示される複数行コメントの処理を修正しました。
Bug fix	ラップされたオブジェクトを適切に識別してスキップするように文法を修正しました。
Bug fix	複数のプロジェクトを同時に実行中に ConcurrentModificationException が発生する問題を修正しました。
Bug fix	複数のスキーマにわたる一意のオブジェクト名の処理を修正しました。オブジェクト名は、異なるスキーマ間で一意である必要はありません。
Bug fix	特定の順序が必要ないため、一部の句の順序を無視するように create table 文法をいくつか修正しました。

第2章 Supported platforms

Migration Portal は、Oracle 11g、12c、18c、および 19c から EDB Postgres Advanced Server 11、12、13、または 14 への評価と移行をサポートしています。Migration Portal は、次のブラウザーでサポートされています。

2.1 対応ブラウザ

最高のユーザーエクスペリエンスを実現するには、Google Chrome ブラウザーの使用をお勧めします。Migration Portal は、次のブラウザーでもサポートされています。

Browser	Supported version
Apple Safari on Macintosh OS	14 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

第3章 Using Migration Portal

Migration Portal を使用すると、データベーススキーマを Oracle から EDB Postgres Advanced Server に簡単に移行できます。評価のためにスキーマをアップロードし、すぐにフィードバックと提案を得ることができます。このポータルを使用すると、すべてのオブジェクトの評価済み DDL をダウンロードし、オンプレミスまたはクラウドで EDB Postgres データベースを作成できます。

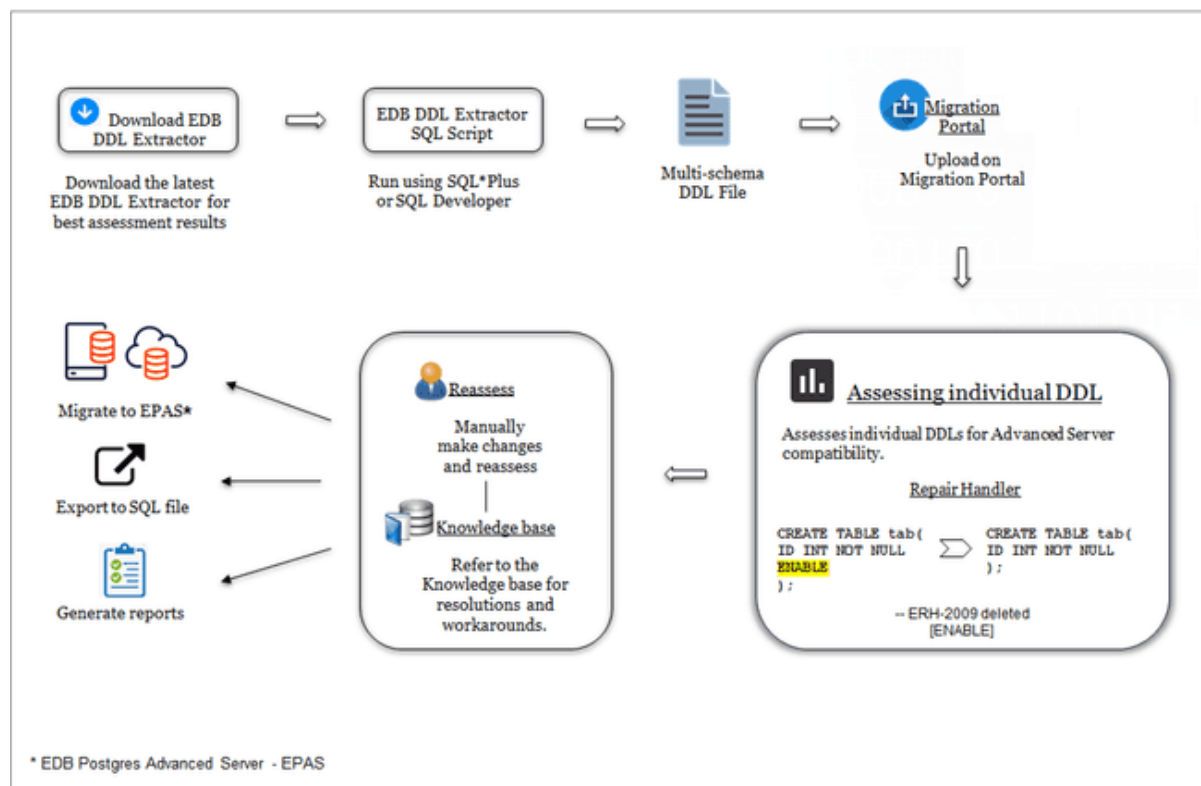


図 1: The assessment and migration process

注釈: Migration Portal には現在、チームコラボレーション機能がありません。共同会議、画面共有、およびダウンロード可能な評価レポートを使用して結果を共有することをお勧めします。

Migration Portal にアクセスするには：

1. ブラウザーを開き、<https://www.enterprisedb.com/> に移動します。
2. EnterpriseDB ホームページで、**PostgreSQLSoftware>Migration** を選択します。
3. Migration Portal ページで、**Findoutmore>OpenMigrationPortal** を選択します。
4. 資格情報を使用してログインします。Migration Portal を使用して Oracle スキーマを移行する概要に

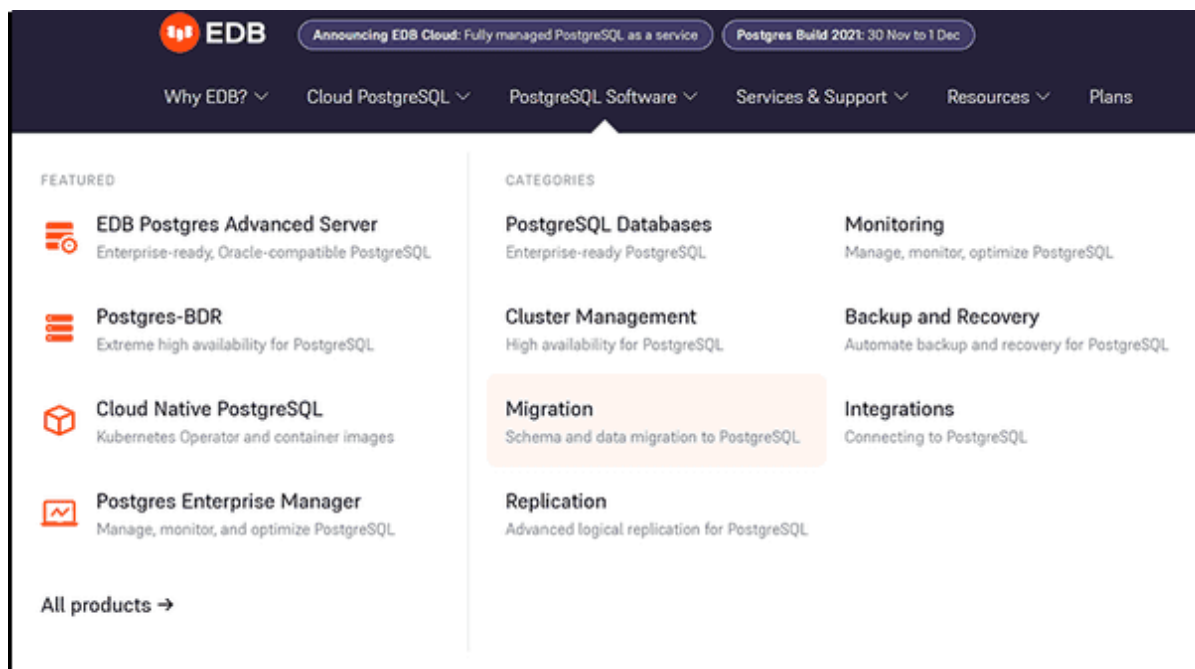


図 2: Migration Portal on the EnterpriseDB home page

ついては、

Migration Portal quick start を参照してください。

3.1 Overview of the Migration Portal home page

Migration Portal のホームページから、移行ツールにすばやくアクセスできます。

Migration Portal ホームページでは、次の Migration Portal 機能にアクセスできます。

- **プロジェクト**：プロジェクトパネルには、評価されたプロジェクトのリストが表示されます。
- **プロジェクトの作成**：+New を選択してプロジェクトを作成します。
- **プロジェクトを検索する**：検索ボックスを使用して、プロジェクトを検索します。
- **概要**：概要パネルには、選択したプロジェクトに関する詳細が表示され、スキーマ評価後に互換性の割合が表示されます。
- **Migrateto**：Migrateto を選択して、EDB Postgres Advanced Server と互換性のある .sql ファイルをダウンロードするか、スキーマをオンプレミスまたはクラウド内の EDB Postgres Advanced Server に移行します。詳細については、*Performing a schema migration* を参照してください。
- **レポート**：レポートを選択して、スキーマ評価レポートを表示およびダウンロードします。
- **Delete**：Delete を選択して、選択したプロジェクトを削除します。
- **警告記号**：プロジェクトまたはスキーマの互換性が 70 %未満である場合、または複数回試行しても DDL が成功しない場合、警告メッセージが表示されます。

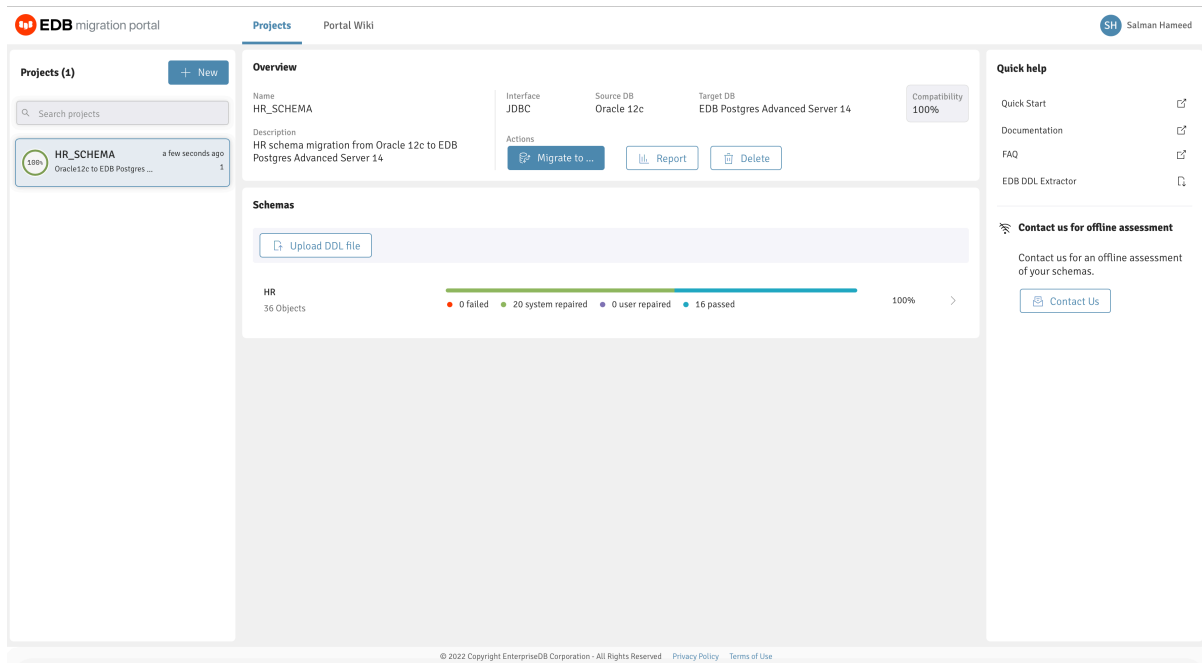


図 3: The Migration Portal home page.

- **DDL ファイルのアップロード**：新しいスキーマのアップロードを選択して、新しい DDL ファイルをアップロードします。
- **スキーマ**：スキーマパネルには、アップロードされた DDL ファイルからの評価結果が表示されます。
- **クイックヘルプ**：クイックヘルプパネルには、いくつかの必要なリソースへのクイックリンクが含まれています。
- **PortalWiki**：PortalWiki を選択して、製品情報やその他のヘルプガイドへのリンクにアクセスします。

3.2 Overview of the Migration Portal Projects page

Migration Portal Projects ページには、移行に関する詳細情報が表示されます。

次のリソースを使用して、移行プロジェクトに関する情報を収集します。

- **互換性**：互換性ゲージは、評価されたスキーマの互換性の割合に基づいて色を表示します。
- **SchemaCount**：プロジェクト内のスキーマの数を表示します。
- **Migrateto**：Migrateto を選択して、スキーマをオンプレミスまたはクラウド内の EDB Postgres Advanced Server に移行します。
- **オブジェクトの検索**：検索ボックスを使用して、オブジェクトを検索します。
- **フィルター**：プロジェクトページの左側のパネルから、システム修復されたオブジェクトと手動修復されたオブジェクトをフィルタリングできます。1 つ以上のフィルターの組み合わせを選択して、情報を絞り込むことができます。
- **Objects**：選択したスキーマのオブジェクトを表示します。

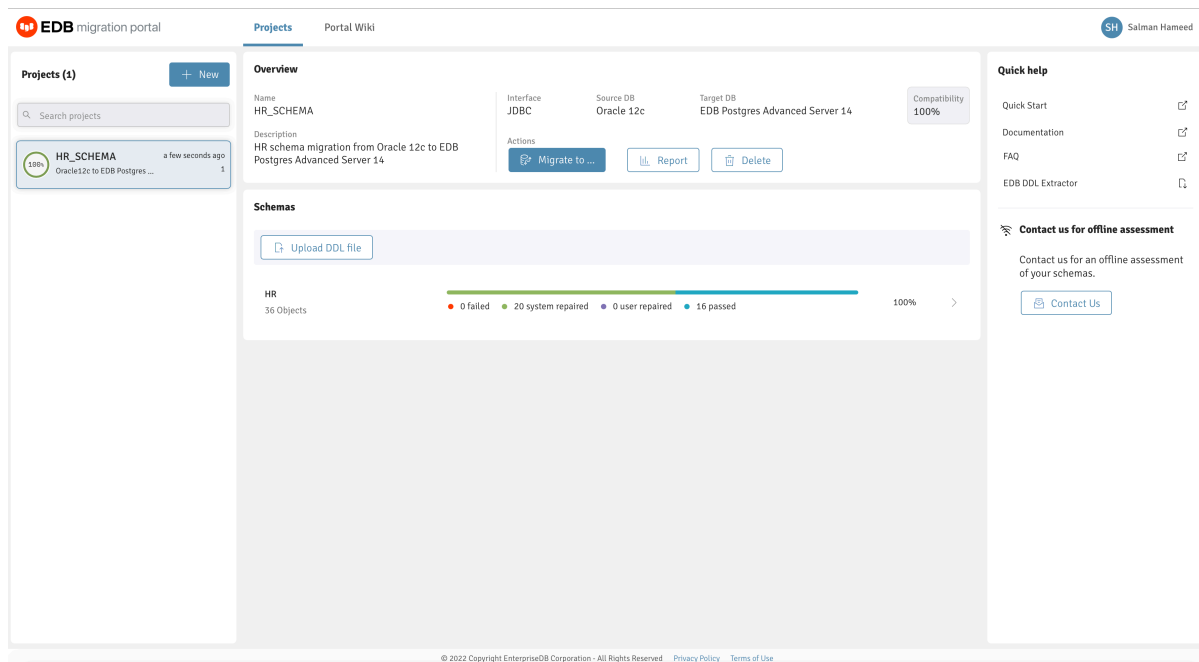


図 4: The Migration Portal Projects Page overview.

- **一般的な障害**：選択したスキーマのオブジェクトが失敗した理由を表示します。

注釈: プロジェクトの一般的な障害の`.csv` ファイルをダウンロードできます。

- **スキーマ**：評価の結果を表示します。
- **ツールチップ**：結果セットにカーソルを合わせると、成功、失敗、および修復されたオブジェクトの数を示すツールチップが表示されます。
- **クイックヘルプ**：クイックヘルプパネルには、ナレッジベースの記事と修復ハンドラーのドキュメントへのリンクが表示されます。
- **検索**：検索ボックスを使用して、特定の情報についてナレッジベースのエントリまたは修復ハンドラーのドキュメントを検索します。

3.3 Migration Portal quick start

Migration Portal を使用して Oracle スキーマを移行するには：

1. スキーマのメタデータ情報を含む`.sql` ファイルを生成します。詳細については、[Performing a schema extraction](#) を参照してください。
1. EDB 資格情報で [Using Migration Portal](#) にログインします。
1. 新しいプロジェクトを作成します。
1. スキーマメタデータを含む`.sql` ファイルをアップロードします。

注釈: 大きな DDL ファイルをアップロードすると、ブラウザーがフリーズまたはクラッシュする場合があります（ブラウザーと、使用中のシステムリソースによって異なります）。このシナリオでは、別のブラウザーで Migration Portal を実行するか、抽出するスキーマの数を減らして、そのファイルを Migration Portal にアップロードしてみてください。

1. **作成と評価** を選択します。Migration Portal は、すべてのコンストラクトを確認し、修復アクションを実行して EDB Postgres Advanced Server との互換性を向上させ、手動介入が必要な残りのエラーにフラグを付けます。
 2. DDL オブジェクト、たとえば、100 %の成功率を示さない TABLES を確認します。
 3. EDB Postgres Advanced Server と互換性のないオブジェクトを選択します。
 4. EDB Postgres Advanced Server と互換性のないオブジェクトの考えられる回避策を確認するには、ナレッジベースを参照してください。
-
1. 必要な変更を加え、**Reassess** を選択して分析を再実行します。
 1. スキーマのエラーを解決した後、変更した EDB 互換 DDL ファイルをダウンロードし、pgAdmin や PSQL クライアントなどのクライアントアプリケーションを使用して、スキーマを EDB Postgres Advanced Server に移行できます。クラウド上の EDB Postgres Advanced Server に移行することもできます。詳細については、[Performing a schema migration](#) を参照してください。

第4章 Migrating a database

データベースを移行するには、次の手順を完了する必要があります。

1. *Performing a schema extraction* を実行します。
2. *Performing a schema assessment* を実行します。
3. *Performing a schema migration* を実行します。
4. *Performing a data migration* を実行します。

4.1 Performing a schema extraction

次のいずれかの方法を使用して、スキーマ抽出を実行できます。EDB は、EDB DDL エクストラクターを使用してスキーマを抽出することをお勧めします。 -

EDB DDL エクストラクターを使用したスキーマの抽出（推奨方法） - Extracting schemas using Oracle Data Pump utilities

詳細については、*Known issues, limitations, and notes* を参照してください。

4.1.1 EDB DDL エクストラクターを使用したスキーマの抽出

Migration Portal Projects ページから最新の EDB DDL エクストラクタスクリプトをダウンロードします。

注釈: Migration Portal は、システムとブラウザーのリソースの可用性によっては、非常に大きな SQL ファイルに対して応答しなくなる場合があります。これを解決するには、一度に抽出するスキーマの数を減らして、SQL ファイルのサイズを縮小してみてください。

前提条件

SQL Developer または SQL *Plus で EDB DDL エクストラクタスクリプトを実行できます。Oracle の DBMS_METADATA ビルトインパッケージを使用して、スキーマ（スクリプトの実行中に指定）の下のさまざまなオブジェクトの DDL を抽出します。EDB DDL エクストラクターは、ポータルにアップロードされ、EDB Postgres Advanced Server の互換性について分析された DDL ファイルを作成します。

注釈: `CONNECT` および `'SELECT_CATALOG_ROLE'` ロールと `'CREATE TABLE'` 特権が必要です。

SQL*Plus の場合

1. SQL*Plus に接続し、次のコマンドを実行します。

```
@edb_ddl_extractor.sql
```

2. プロンプトが表示されたら、スクリプトを実行している Oracle ユーザーに必要な権限がある場合は、**Enter** を押してスクリプトの実行を続行します。

3. プロンプトが表示されたら、スキーマ名を入力します。複数のスキーマを抽出する場合は、コンマ（`,`）を区切り文字として使用します。

注釈: 現在のデータベースからすべてのユーザースキーマを抽出する場合は、抽出中にスキーマ名を言及しないでください。ただし、抽出するスキーマ名に言及することをお勧めします。

4. プロンプトが表示されたら、エクストラクターが抽出した DDL を保存するパスまたはディレクトリを入力します。パスを指定しない場合、抽出された DDL は SQL*Plus コマンドを実行したディレクトリに出力されます。

5. プロンプトが表示されたら、他のスキーマから依存オブジェクトを抽出するかどうかに応じて、**yes** または **no** と入力します。

6. プロンプトが表示されたら、他のスキーマから `grant` ステートメントを抽出するかどうかに応じて、**yes** または **no** と入力します。

例：

1. 抽出スクリプトを実行します。

```
@edb_ddl_extractor.sql
```

2. **Enter** を押します

3. コンマで区切って 3 つのスキーマを指定します。

```
Enter a comma separated list of schemas to be extracted (Default all schemas): HR,
→SCOTT, FINANCE
```

4. 出力ファイルのパスを入力します。

Linux では、次を使用できます。

```
Location for output file (Default current location) : /home/oracle/extracted_ddls/
```

同様に Windows の場合：

Location for output file (Default current location) : c:\Users\Example\Desktop\

5. yes と入力して、他のスキーマの依存オブジェクトを抽出します。

WARNING:

Given schema(s) list may contain objects which are dependent on objects from other schema(s), not mentioned in the list. `Assessment may fail for such objects. It is suggested to extract all dependent objects together.

Extract dependent object from other schemas?(yes/no) (Default no / Ignored for all schemas option): yes

6. yes と入力して、grant ステートメントを抽出します。

Extract GRANT statements?(yes/no) (Default no): yes

SQL Developer の場合

edb_ddl_extractor.sql スクリプトを SQL Developer にロードし、ソース Oracle データベースに接続した後、スクリプトを実行します。スクリプトが実行されたら、プロンプトに応答します。

1. データベース接続のユーザーが、[**スクリプト出力**] タブにリストされている必要な特権を持っている場合、**はい**を選択してスクリプトの実行を続行します。
2. スキーマのコンマ区切りリストを入力し、**OK** を選択します。
3. 出力ファイルのパスを入力し、**OK** を選択します。デフォルトは、DDL エクストラクタスクリプトの場所です。
4. **yes** または **no** と入力して依存オブジェクトを抽出し、**OK** を選択します。デフォルトでは、依存オブジェクトを抽出しません。
5. **yes** または **no** と入力して grant ステートメントを抽出し、**OK** を選択します。デフォルト設定では、grant ステートメントを抽出しません。

DDL エクストラクタの実行の出力が Script Output タブに表示されます。出力ファイルの名前は、スクリプト出力の Extraction Completed メッセージの後に表示されます。

注釈: SQL*Plus と SQL Developer の両方に単一のスキーマ名を入力することもできます。

次に、スクリプトはソースデータベース内のオブジェクトタイプを反復処理します。タスクが完了すると、.SQL 出力は入力した場所（たとえば、c:\Users\Example\Desktop\）に保存されます。

Migration Portal が想定するファイルエンコーディング形式については、[ファイルエンコーディング](#) を参照してください。

4.1.2 スキーマとオブジェクトのサポート

以下のリストと表は、Migration Portal でサポートされているおよびサポートされていないスキーマとオブジェクトを示しています。

サポートされていないスキーマ

SQL ダンプファイルの生成中に、これらの Oracle システムスキーマを除外します。

ANONYMOUS	APEX_PUBLIC_USER	APEX_030200	APEX_040000	APEX_040200
APPQOSSYS	AUDSYS	CTXSYS	DMSYS	DBSNMP
DBSFWRUSER	DEMO	DIP	DMSYS	DVF
DV\$SYS	EXFSYS	FLows_FILES	FLows_020100	FRANCK
GGSYS	GSMADMIN_INTERNAL	GSMCATUSER	GSMROOTUSER	GSMUSER
LBACSYS	MDDATA	MDSYS	MGMT_VIEW	OJVMSYS
OLAPSYS	ORDPLUGINS	ORDSYS	ORDDATA	OUTLN
ORACLE_OCM	OWBSYS	OWBYSS_AUDIT	PDBADMIN	RMAN
REMOTE_SCHEDULER_AGENT	SI_INFORMTN_SCHEMA	SPATIAL_CSW_ADMIN_USR	SPATIAL_WFS_ADMIN_USR	SQLTXADMIN
SQLTXPLAIN	SYSSUMF	SYS	SYSBACKUP	SYSDG
SYSKM	SYSRAC	SYSTEM	SYSMAN	TSM\$SYS
WKPROXY	WKSYS	WK_TEST	WMSYS	XDB
XS\$NULL				

注釈: EDB Postgres Advanced Server は、`pg_` で始まるスキーマ名をサポートしていません。

サポートされているオブジェクトタイプ

- テーブル
- シーケンス
- 制約
- インデックス (LOB インデックスとマテリアライズドビューのインデックスを除く)
- 同義語
- ビュー
- マテリアライズドビュー
- DB リンク
- タイプとタイプボディ
- トリガー
- 機能
- 手続き
- パッケージ
- ユーザー
- ロール
- プロフィール
- ロールとオブジェクトの付与

注釈: 列、テーブル、およびマテリアライズドビューの COMMENTS もサポートされています。

サポートされていないオブジェクトタイプ

- エディション
- オペレーター
- スケジューラー
- LOB インデックスとマテリアライズドビューのインデックス
- XML スキーマ

- 表領域
- ディレクトリ
- RLS ポリシー
- キュー *
- ライブラリー
- インデックスタイプ

* EDB Postgres Advanced Server はキューテーブルのサポートを提供していますが、Migration Portal は現在サポートしていません。ソース DDL のキューテーブルは、ソースおよびターゲット DDL オブジェクトとしてアップロードされません。

Extracting schemas using Oracle Data Pump utilities

Migration Portal は、抽出されたスキーマを評価するために SQL ダンプファイル内のメタデータのみを必要とします。

前提条件

- 自分自身ではないスキーマをエクスポートする予定がある場合は、DATAPUMP_IMP_FULL_DATABASE ロールが割り当てられていることを確認してください。それ以外の場合は、自分のスキーマのみをエクスポートできます。
- 表領域にオブジェクトを作成するための十分な表領域クォータがあることを確認します。
- SQL*Plus で、サーバーファイルシステム上のディレクトリを指すディレクトリオブジェクトを作成します。例：**Linux の場合**：

```
create directory DMPDIR as /sample/file/path;
```
Windows の場合：


```
```sql
create directory DMPDIR as c:\sample\file\path\;
```


```

- SQL*Plus で、Data Pump ユーティリティを実行しているユーザーに読み書き権限を付与します。

```
grant read,write on directory DMPDIR to <Username>;
```

次のいずれかの手順を実行します。

- データベース内の 1 つ以上のスキーマを抽出する
- データベース内のすべてのスキーマを抽出する

データベース内の 1 つ以上のスキーマを抽出する

1. `expdp` コマンドを実行する前に、サーバー上に `.par` 拡張子（たとえば、`export.par`）を持つファイルを作成します。属性と値をファイルに追加します。

`edb_notranlate_3` このコマンドの属性と値は、次のオプションを指定します。- `CONTENT=metadata_only` は、スキーマのメタデータのみを抽出します。- `INCLUDE=` は、抽出に含めるデータベースオブジェクトタイプを指定します。Migration Portal でサポートされているデータベースオブジェクトタイプのみを抽出します。

[サポートされているオブジェクトタイプ](#) を参照してください。

2. コマンドラインから `export` コマンドを実行して `schemas_metadata.dump` ファイルを生成します。たとえば、`<Schema_1>`、`<Schema_2>`、`<Schema_3>` などのメタデータ情報を抽出するには、次を実行します。

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR SCHEMAS=<Schema_1>,<Schema_2>,<Schema_3> DUMPFILE=schemas_metadata.dump parfile=export.par
```

3. ダンプファイルから SQL ファイルを生成するには、`import` コマンドを実行します。

たとえば、`schemas_metadata.dump` ファイルから `YourSchemas.SQL` ファイルを生成するには、次のように入力します。

```
impdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR TRANSFORM=OID:n,SEGMENT_ATTRIBUTES:n EXCLUDE=STATISTICS SQLFILE=YourSchemas.sql DUMPFILE=schemas_metadata.dump
```

Migration Portal が想定するファイルエンコーディング形式については、[ファイルエンコーディング](#) を参照してください。

データベース内のすべてのスキーマを抽出する

注釈: 除外スキーマリストに属するユーザーアカウントからこの手順を実行しないでください（[サポートされていないスキーマ](#) を参照）。コマンドを実行しているユーザーアカウントがスキーマの除外リストにある場合、`impdp` コマンドは失敗します。

1. `expdp` コマンドを実行する前に、サーバー上に `.par` 拡張子（たとえば、`export.par`）を持つファイルを作成します。属性と値をファイルに追加します。

`edb_notranlate_6` このコマンドの属性と値は、次のオプションを指定します。- `CONTENT=metadata_only` は、スキーマのメタデータのみを抽出します。- `INCLUDE=` は、抽出に含めるデータベースオブジェクトタイプを指定します。Migration Portal でサポートされているデータベースオブジェクトタイプのみを抽出します。

[サポートされているオブジェクトタイプ](#) を参照してください。- `FULL=Y`

は、データベース内のすべてのスキーマを抽出します。

2. コマンドラインから `export` コマンドを実行して `schemas_metadata.dump` ファイルを生成します。

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR DUMPFILE=schemas_metadata.dump_
↳parfile=export.par
```

3. `impdp` コマンドを実行する前に、サーバー上に `.par` 拡張子（たとえば、`import.par`）を持つパラメーターファイルを作成します。属性と値をファイルに追加します。

```
EXCLUDE=SCHEMA:"IN(select username as schema_name from sys.dba_users where_
↳LOWER(username) LIKE pg_% OR username IN
(ANONYMOUS,APEX_PUBLIC_USER,APEX_030200,APEX_040000,APEX_040200,APPQOSSYS,AUDSYS,
↳CTXSYS,DMSYS,DBSNMP,
DBSFUSER,DEMO,DIP,DMSYS,DVF,DVSY,EXFSYS,FLWS_FILES,FLWS_020100,FRANCK,GGSYS,
↳GSMADMIN_INTERNAL,
GSMCATUSER,GSMROOTUSER,GSMUSER,LBACSYS,MDDATA,MDSYS,MGMT_VIEW,OJVMSYS,OLAPSYS,
↳ORDPLUGINS,ORDSYS,
ORDDATA,OUTLN,ORACLE_OCM,OWBSYS,OWBYSS_AUDIT,PDBADMIN,RMAN,REMOTE_SCHEDULER_AGENT,SI_
↳INFORMTN_SCHEMA,
SPATIAL_CSW_ADMIN_USR,SPATIAL_WFS_ADMIN_USR,SQLTXADMIN,SQLTXPLAIN,SYSSUMF,SYS,
↳SYSBACKUP,SYSDG,SYSKM,
SYSRAC,SYSTEM,SYSMAN,TSMSYS,WKPROXY,WKSYS,WK_TEST,WMSYS,XDB,XS$NULL))"
EXCLUDE=STATISTICS
```

このコマンドの属性と値は、次のオプションを指定します。 - `EXCLUDE` は、除外するスキーマを指定します。 **サポートされていないスキーマ** のリストを追加して、Data Pump ユーティリティがシステムスキーマ（`pg_` で始まるスキーマを含む）を抽出しないようにします。 • `EXCLUDE=STATISTICS` は、SQL 出力ファイルに統計関連情報を含めないことを指定します。それらを SQL ファイルに含めないことで、移行ポータルの DDL を評価する機能に影響を与えることなく、ファイルのサイズが大幅に削減されます。

4. ダンプファイルから SQL ファイルを生成するには、`import` コマンドを実行します。

たとえば、`schemas_metadata.dump` ファイルから `YourSchemas.SQL` ファイルを生成するには、次のように入力します。

`edb_notranlate_9` Migration Portal が想定するファイルエンコーディング形式については、

[ファイルエンコーディング](#) を参照してください。

4.2 Performing a schema assessment

EDB Postgres Advanced Server との互換性について Oracle データベーススキーマを評価できます。評価に関するレポートを生成することもできます。

4.2.1 スキーマ評価を実行する

1. *Using Migration Portal* に移動します。
2. EDB 資格情報を入力します。
3. **New** を選択してプロジェクトを作成します。
4. ダイアログボックスで、**プロジェクト名**フィールドにプロジェクト名を入力します。
5. プロジェクトの詳細を指定します。
 - アプリケーションインターフェイスを選択します。 - ソース DB とソースのバージョンを選択します。 - ターゲットのターゲット DB とバージョンを選択します。
6. スキーマメタデータで生成された .SQL ファイルをアップロードするには、**Choosefile** を選択します。

注釈: .SQL ファイルを変更しないでください。

7. プロジェクトの作成時にインデックス (_IDX) と制約 (_CON) にサフィックスを追加することを指定するには、**サフィックスを追加してオブジェクト名を一意にする**チェックボックスをオンにします。Oracle がサポートしているのに対し、EDB Postgres Advanced Server はテーブルとインデックスに同じ名前をサポートしていないため、この選択により、より良い評価結果が保証されます。

8. プロジェクトの作成時に Oracle のデフォルトの大文字と小文字の使用を指定するには、**スキーマの移行に Oracle のデフォルトの大文字と小文字を使用**チェックボックスをオンにします。Oracle のデフォルトのケース機能を使用して、Oracle データベースから EDB Postgres Advanced Server データベースにスキーマを移行するときに、すべてのデータベースオブジェクトの Oracle のデフォルトの大文字の命名規則を保持します。この機能の詳細については、

Known issues, limitations, and notes を参照してください。

9. **作成と評価**を選択して、新しいプロジェクトを作成し、スキーマを評価します。

分析ツールはすべてのコンストラクトをレビューし、修復アクションを実行して、EDB Postgres Advanced Server との互換性を向上させます。手動介入が必要な残りのエラーにフラグを立てます。

10. 100 %の成功率を示さない DDL オブジェクト (TABLES など)を確認します。
11. EDB Postgres Advanced Server と互換性のないオブジェクトを選択し、詳細を表示します。中央パネルの下部で、修復アクションの詳細を表示できます。
12. 右側のパネルのナレッジベース情報を参照して、EDB Postgres Advanced Server とすぐに互換性がないオブジェクトの考えられる回避策を見つけます。Portal wiki ページでナレッジベース情報を表示することもできます。
13. **ナレッジベース** タブで、EDB Postgres Advanced Server と互換性のないオブジェクトのエラーメッセージを入力し、**検索**を選択できます。

オブジェクトの詳細パネルには、失敗したオブジェクトの回避策または解決策が表示されます。そのオブジェクトの**評価**タブで手動で変更を加え、**再評価**を選択できます。

New project ×

Project name

Application interface

☒ JDBC ☐ ODBC ☐ .NET ☐ OCI ☐ ProC ☐ Other

Source DB

Oracle ▼

Version

11g ▼

Target DB

EDB Postgres Advanced Server ▼

Version

15 ▼

DDL file [\(How to get a DDL file?\)](#)

No file chosen 📁 Choose file

☐ Add suffix to make object names unique ⓘ

☐ Use Oracle's default case for schema migration. ⓘ

Description(Optional)

Say something about the project...

200 characters left

✕ Cancel ✓ Create & assess

図 1: The Migration Portal New project dialog

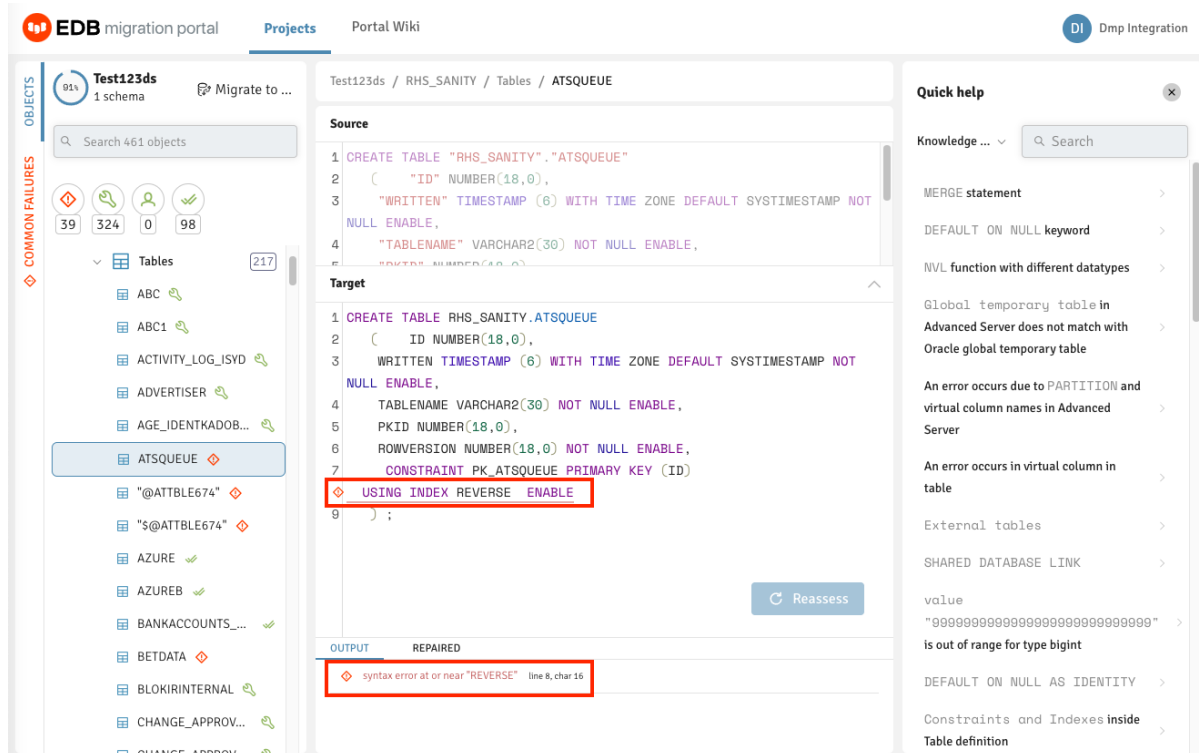
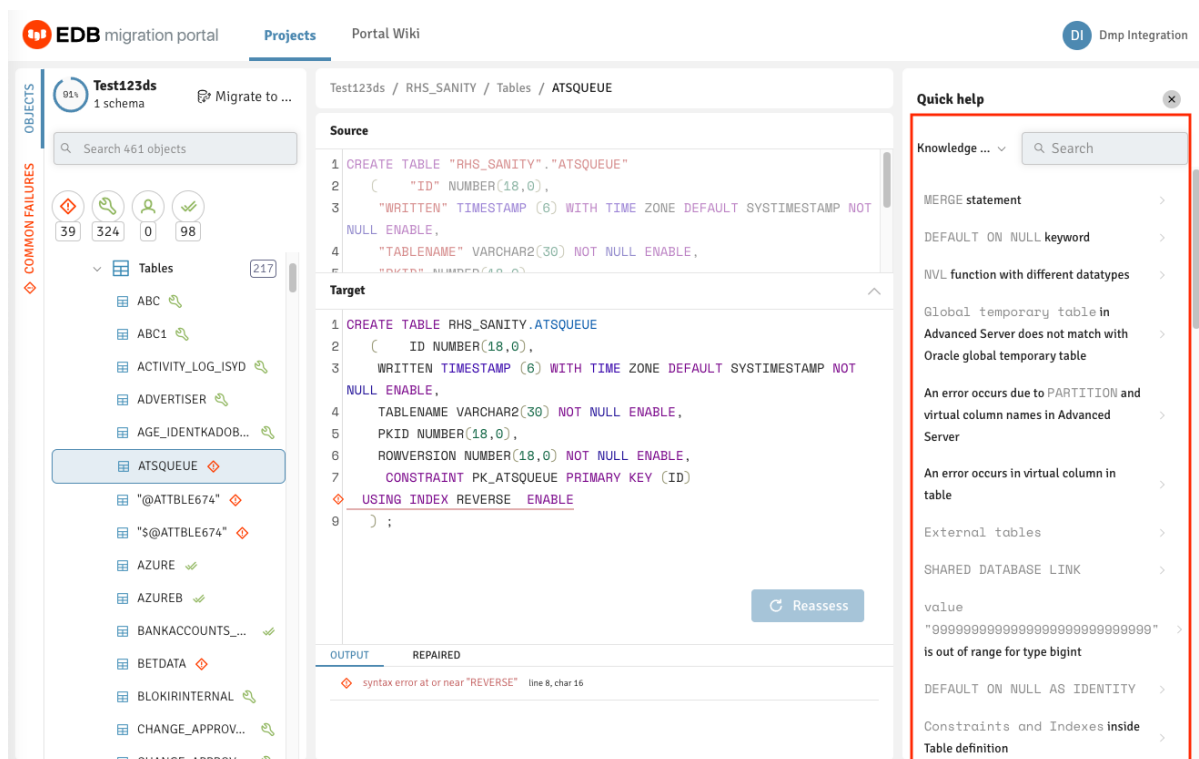


Figure 2: Incompatible objects are identified



⊗ 3: Incompatible objects are identified

注釈: 再評価中に失敗したオブジェクトが合格した場合、そのオブジェクトの依存オブジェクトも再評価されます。

同様に、互換性のないすべてのオブジェクトを互換性のあるものにすることができます。

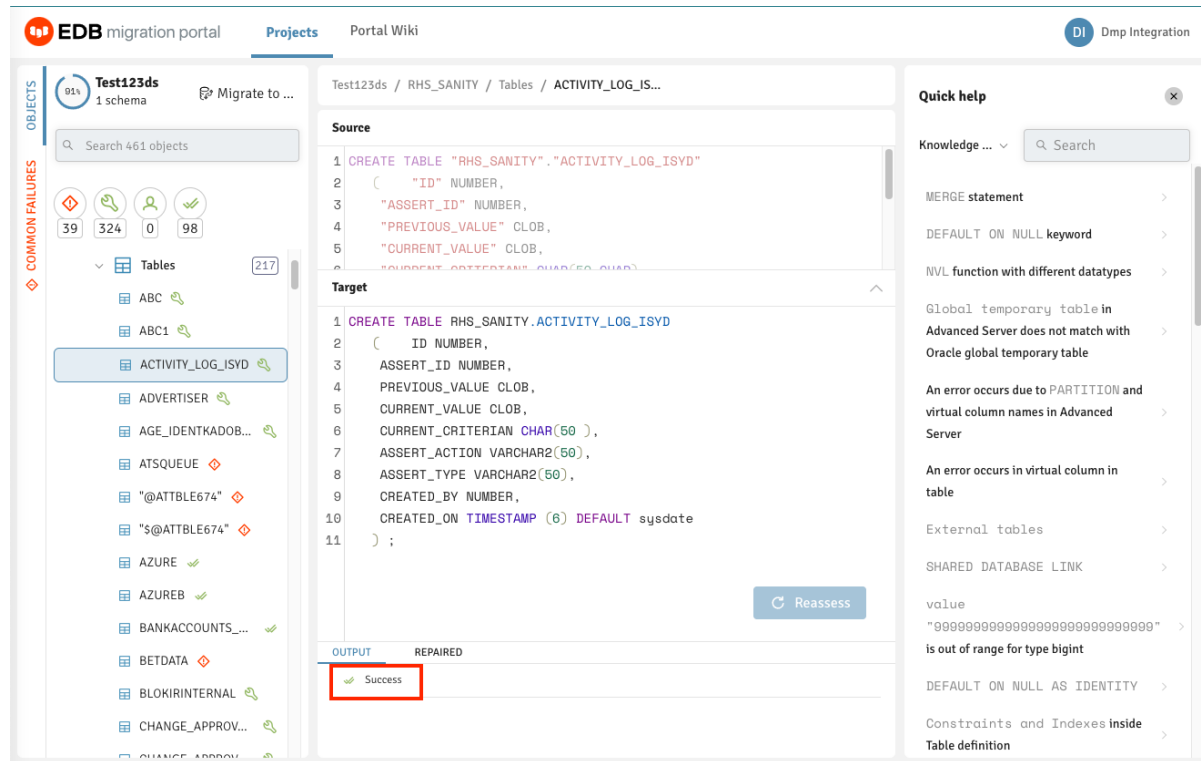


図 4: Workaround or resolution for incompatible objects

注釈: 互換性のないオブジェクトの情報または回避策がナレッジベースで利用できない場合は、サポートチームに連絡してください。

DDL での作業が終了したら、変更した EDB 互換 DDL を .sql ファイルとしてダウンロードするか、スキーマを移行できます。

Performing a schema migration を参照してください。

4.2.2 評価レポートを生成する

Migration Portal レポートは、プロジェクト用に評価されたスキーマの高レベルの評価概要を提供します。さらに、レポートは、失敗したオブジェクトと失敗の原因に関する詳細を提供します。

1. レポートを選択します。
2. レポートに含めるスキーマを選択します。
3. 画面上のレポートを生成するには、**Generate** を選択します。

オプションで **GeneratePDF** を選択して、レポートを PDF 形式で保存できます。選択したスキーマの DDL に適用された個別の修復ハンドラーの数を表示することもできます。

注釈: Migration Portal は、Oracle ラップされたオブジェクトに隠されている機密性の高い PL / SQL ソースコードを評価しません。これらのラップされたオブジェクトは評価されたオブジェクトの数に含まれないため、互換性割合の実際の値は、評価レポートで計算された値と異なる場合があります。ラップされたオブジェクトの詳細については、Known issues and notes の注を参照してください。

4.2.3 評価レポートを評価する

評価レポートは、スキーマおよびプロジェクト全体のオブジェクトと機能の互換性に関するデータと情報を提供します。

- **オブジェクト** : Migration Portal は、処理するすべての CREATE ステートメントのオブジェクトを生成します。さらに、アップロードされた DDL に制約を追加する ALTER TABLE ステートメントごとにオブジェクトを生成します。
- **特徴**: オブジェクトの作成に使用される Oracle 句または構造。機能は、1 つ以上のオブジェクトで複数回使用できます。

評価レポートには、**互換性のない機能**および**未解決のオブジェクト**のテーブルがリストされています。

オブジェクトの互換性

オブジェクトは、ターゲットデータベースで正常に実行された場合、互換性があります。評価レポートには、次のカテゴリに基づいたオブジェクト互換性の概要が表示されます。

| Object level | compatibility | Description |
|-----------------|---------------|--|
| Passed | | 修復ハンドラーを適用せずにターゲットデータベースで実行します。 |
| System repaired | | 1 つ以上の修復ハンドラーを適用した後、ターゲットデータベースで正常に実行されます。修復ハンドラーは、変換ロジックを使用して、DDL を自動的に EDB Postgres Advanced Server と互換性のあるものにします。オブジェクトの動作に予期しない影響が発生した場合に備えて、レポートの 修復ハンドラーが適用されました セクションの説明を注意深く確認してください。 |
| User repaired | | ユーザーが手動で修復および再評価した後、ターゲットデータベースで正常に実行されます。 |
| Failed objects | | ターゲットデータベースで正常に実行されません。 |

注釈: 提示されたオブジェクト分析は、ターゲット DDL 内のオブジェクトの既存の状態に基づいています。オブジェクトが更新および再評価された場合、これは変わる可能性があります。たとえば、修正されたオブジェクトは、再評価後に*未解決のオブジェクト*テーブルから削除されます。

機能の互換性

EDB Postgres Advanced Server では、一部の機能をネイティブに利用できない場合があります。評価レポートには、機能の発生と互換性レベルの概要が示されます。

| Feature level | compatibility | Workaround status |
|--|---------------|----------------------------|
| Compatible features | | |
| • | | |
| Features with available workaround | existing | Exists |
| Features with reasonable effort new workaround | | Possible |
| Features with no workaround or requiring further investigation | no | None / Needs Investigation |

注釈: 回避策のない機能の場合、アプリケーションレベルの変更により機能の非互換性に対処できる場合があります。####制限事項

評価レポートは、ユーザーがターゲット DDL を手動で変更し、**再評価**を選択した後に更新されます。ただし、レポートの**互換性のない機能表**は更新されず、ソース DDL の互換性のない機能のリストが引き続き表示されます。スキーマ移行の複雑さ評価スコアは、互換性のない機能を持つ互換性のないオブジェクトが解決されると更新されます。また、オブジェクトが修正されて再評価された後、オブジェクトは**未解決のオブジェクト**テーブルから削除され、それらのオブジェクトで元々見つかった互換性のない機能のリストとともに削除されます。

4.3 Performing a schema migration

スキーマのエラーを解決した後、スキーマを EDB Postgres Advanced Server データベースに移行できます。

Migration Portal には、次のターゲットデータベースオプションのいずれかを使用してスキーマを移行するためのガイド付き手順が用意されています。

- オフライン移行-データ定義言語 (DDL) SQL ファイルをダウンロードして、psql クライアントや pgAdmin などのクライアントアプリケーションを使用して、ターゲット EDB Postgres Advanced Server データベースにスキーマを手動で作成します。このオプションは、オンプレミスとクラウドベースの移行の両方に適しています。
- オンライン移行-ターゲットの EDB Postgres Advanced Server データベースに直接接続し、スキーマを自動的に作成します。一般に、このオプションは、クラウドで実行されているデータベース (BigAnimal クラスターまたはセルフマネージドインスタンス) への移行に最適です。

スキーマを移行する前に、EDB は、 *Known issues, limitations, and notes* ターゲットデータベースの構成要件を特定することをお勧めします。

4.3.1 オフライン移行

クライアントを使用して DDL SQL ファイルを実行し、Windows または Linux プラットフォーム上の EDB Postgres Advanced Server にスキーマを手動で移行します。

1. **オフライン移行** を選択します。
2. EDB Postgres Advanced Server に移行する 1 つ以上のスキーマを選択します。スキーマを展開し、各スキーマの下で特定のオブジェクトタイプを選択することもできます。

Generate and download DDL SQL scripts for offline migration of schemas ?

1 Select Schemas
2 Download Schema
3 Finish

| Project Name | Interface | Source DB | Target DB | Compatibility |
|---------------|-----------|------------|---------------------------------|---------------|
| TestMigration | JDBC | Oracle 18c | EDB Postgres Advanced Server 14 | 100% |

Select one or more schemas

Note: Click the arrows next to the schema name to see and select specific object types to be migrated. Also note that the numbers in parenthesis next to the object type name indicate the number of compatible objects, total number of objects, and % of compatible objects currently available in the schema for the object type.

☒ All

☒ GLOBAL_OBJECTS

☒ HRPLUS

☒ CONSTRAINT (10/10 - 100%)
 ☒ FUNCTION (2/2 - 100%)
 ☒ INDEX (11/11 - 100%)
 ☒ PACKAGE (1/1 - 100%)
 ☒ PACKAGE BODY (1/1 - 100%)
 ☒ PROCEDURE (9/9 - 100%)
 ☒ SEQUENCE (3/3 - 100%)
 ☒ TABLE (8/8 - 100%)
 ☒ TRIGGER (3/3 - 100%)
 ☒ VIEW (8/8 - 100%)

☒ SUPER

Cancel

← Previous
Next →

注釈: スキーマの互換性が 100 %でない場合は、バナーが表示されます。必要に応じて、お問い合わせフォームに記入してください。

1. **DownloadSQLfile** を選択して、評価されたスキーマをダウンロードします。zip パッケージは、選択したスキーマとオブジェクトの DDL を含む SQL ファイルで構成されています。DDL は、プロジェクト、スキーマ、またはオブジェクトごとに SQL ファイルにグループ化されます。

34

第 4 章 Migrating a database

| File name | Description | Example |
|--------------------------------------|---|--|
| <ProjectName>*
_ALL_OBJECTS.sql | 選択したすべてのスキーマとオブジェクトの DDL
で構成される単一の統合ファイル。 | AllTestMigration_ALL_OBJECTS.sql
consists of DDLs
for all objects
of schemas OE,
SH, IX, and
PM, and only
selected objects
of schema HR. |
| <SchemaName>* .sql | 特定のスキーマ内の選択されたオブジェクトの
DDL で構成される 1 つ以上のファイル。 | HR.sql consists of DDLs for
all the selected objects of the
HR schema. |
| <SchemaName>*_
<ObjectType>* .sql | それぞれが選択した単一のオブジェクトの DDL
で構成される 1 つ以上のファイル。 | HR_TABLE.sql consists of
DDLs only for TABLE object
of the HR schema. |

1. zip ファイルのダウンロードが完了したら、**完了**を選択して、Migration Portal のメインページに戻ります。
1. スキーマをターゲットデータベースにインポートします。
 1. edb psql コマンドラインユーティリティを使用して、変換されたスキーマの移行元の EDB Postgres Advanced Server データベースに接続します。
1. 新しいデータベースを作成します。

```
CREATE DATABASE <database_name>;
```

1. 次の psql コマンドを使用して、新しいデータベースに接続します。

```
\connect <database_name>
```

1. 次の psql コマンドを使用して、必要な SQL ファイルを新しいデータベースにインポートします。

```
\i <path_to_exported_schema_file>.sql
```

注釈: または、インポートに `pgAdmin` クライアントを使用できます。スキーマをターゲットデータベースにインポートします。

4.3.2 オンライン移行

ターゲット EDB Postgres Advanced Server データベースに直接接続してスキーマを作成します。

1. **オンライン移行** を選択します。
2. EDB Postgres Advanced Server に移行する 1 つ以上のスキーマを選択します。スキーマを展開し、各スキーマの下で特定のオブジェクトタイプを選択することもできます。

Connect to database and perform online migration of schemas ?

1 Select Schemas
2 Connect
3 Deploy
4 Finish

| Project Name | Interface | Source DB | Target DB | Compatibility |
|---------------|-----------|------------|---------------------------------|---------------|
| TestMigration | JDBC | Oracle 18c | EDB Postgres Advanced Server 14 | 100% |

Select one or more schemas

Note: Click the arrows next to the schema name to see and select specific object types to be migrated. Also note that the numbers in parenthesis next to the object type name indicate the number of compatible objects, total number of objects, and % of compatible objects currently available in the schema for the object type.

- ☒ All
 - > ☒ GLOBAL_OBJECTS
 - > ☒ HRPLUS
 - ☒ CONSTRAINT (10/10 - 100%)
 - ☒ FUNCTION (2/2 - 100%)
 - ☒ INDEX (11/11 - 100%)
 - ☒ PACKAGE (1/1 - 100%)
 - ☒ PACKAGE BODY (1/1 - 100%)
 - ☒ PROCEDURE (9/9 - 100%)
 - ☒ SEQUENCE (3/3 - 100%)
 - ☒ TABLE (8/8 - 100%)
 - ☒ TRIGGER (3/3 - 100%)
 - ☒ VIEW (8/8 - 100%)
 - > ☒ SUPER

Cancel
← Previous
→ Next

3. **Connection** ページで必要な接続の詳細を入力します。

1. 接続の詳細を確認するには、**TestConnection** を選択します。

注釈: ****接続の編集****を選択して、接続の詳細を変更し、接続の詳細を再テストできます。

1. テスト接続が成功した場合、**次へ**を選択してターゲットデータベースに接続し、選択したスキーマをターゲットサーバーに作成します。

1. 移行が完了すると、移行が正常に完了したか、エラーが発生したかを示すメッセージがページに表示されます。**DownloadSummary** ボタンを選択して、移行の試行に対して生成されたログをダウンロードして確認します。

Connect to database and perform online migration of schemas



✓ Select Schemas

2 Connect

3 Deploy

4 Finish

| | | | | |
|---------------|-----------|------------|---------------------------------|---------------|
| Project Name | Interface | Source DB | Target DB | Compatibility |
| TestMigration | JDBC | Oracle 18c | EDB Postgres Advanced Server 14 | 100% |

Connect to your cluster

Target Database

TestMigration

Host Name/Address

Host Name/Address

Port

5432

Note: Ensure that your database server is accessible from Migration Portal.

Maintenance Database

edb_admin

Username

edb_admin

Password

Password

Test Connection

Cancel

← Previous

→ Next

図 5: Connecting to the cloud cluster

1. **Done** を選択して、Migration Portal のメインページに戻ります。

4.4 Performing a data migration

スキーマの移行を実行した後、データを移行します。

1. Migration Toolkit を使用して、データを移行します。 Migration Toolkit の使用の詳細については、Migration Toolkit を参照してください。
2. Migration Toolkit `toolkit.properties` ファイルを構成し、ソースデータベースとターゲットデータベースの接続情報がプロパティファイルで使用できることを確認します。

```
SRC_DB_URL = jdbc: oracle:thin:@localhost:1521:ORCL
SRC_DB_USER = user_name
SRC_DB_PASSWORD = password
TARGET_DB_URL= jdbc:edb://localhost:5444/migration
TARGET_DB_USER = enterprisedb
TARGET_DB-PASSWORD = password
```

詳細については、Building the toolkit.properties File を参照してください。

3. Migration Toolkit を `-dataOnly` モードで起動します。 `-truncLoad` キーワードを含めて、テーブル間の外部キーの依存関係を解決します。

たとえば、次のコマンドは、データを `hr` という名前のスキーマに移行します。

```
runMTK.sh -dataOnly -targetSchema hr -truncLoad HR
```

このコマンドは、指定された `source_schema` を `target_schema` に移行します。データは、`enterprisedb` という名前のデータベーススーパーユーザーと `password` のパスワードを使用して、ローカルにインストールされた EDB Postgres Advanced Server インスタンスにロードされます。

注釈: テーブルは、データのロードを試みる前に切り捨てられます。

4.4.1 高度なデータ移行

並列データのロードが必要な大規模なデータベースの場合、次のいずれかの方法を使用できます。

- EDB Postgres Advanced Server データベースリンク機能（Oracle データベースとの互換性のため）
- Dblink またはデータベースリンクスタイルの移行（データに CLOB データが含まれる場合）

詳細については、Migration Toolkit を参照してください。

第5章 Known issues, limitations, and notes

このページには、次の既知の問題、制限事項、および注意事項がリストされています。

- *Using Migration Portal*
- *EDB DDL エクストラクタを使用したスキーマの抽出*
- *Extracting schemas using Oracle Data Pump utilities*

5.1 Migration Portal

5.1.1 ラップされたオブジェクト

Migration Portal はラップされたオブジェクトを評価できません。抽出された DDL にそれらを含める場合、それらは Migration Portal にロードされず、評価されるオブジェクトの数に含まれません。ラップされたオブジェクトを評価して EDB Postgres Advanced Server に移行する場合は、Migration Portal にアップロードする DDL ファイルにオブジェクトのラップされていないバージョンを含めます。これを行う推奨される方法は、スキーマ抽出を実行する前に、Oracle データベース内のラップされたバージョンをクリアテキストバージョンに置き換えることです。スキーマ抽出を実行した後、オブジェクトをラップされたバージョンに置き換えることができます。

5.1.2 ファイルエンコーディング

Migration Portal は、.SQL 出力ファイルを UTF-8 エンコーディング形式にすることをお勧めします。非 UTF-8 エンコーディングの .SQL ファイルをアップロードすると、UTF-8 と互換性のないすべての文字は、出力 DDL で置換文字「☒」に変換されます。

Tip: Linux の iconv ユーティリティまたは Windows の LibIconv ユーティリティを使用して、抽出したファイルを UTF-8 形式に手動で変換できます。たとえば、データベースのキャラクタセットが Latin-1 (ISO-8859-1) の場合、次のように、抽出したファイルを UTF-8 形式に変換できます。

5.1.3 ALTER ステートメント

ALTER TABLE と ALTER TRIGGER を除き、Migration Portal は DDL 内の他の ALTER ステートメントを処理しません。

5.1.4 ユーザー、ロール、プロファイル、および付与

5.1.5 付与

Migration Portal は、抽出およびアップロードされたスキーマの Oracle ユーザーの評価と移行をサポートするようになりました。プロジェクトに関連付けられたスキーマのすべてのユーザー、ロール、およびプロファイルは、Migration Portal ビューの GLOBAL_OBJECTS 擬似スキーマの下にグループ化されます。現在、システム権限、オブジェクト権限、およびロールの付与は、ユーザーオブジェクトの DDL に含まれています。同様に、ロールへのシステムおよびオブジェクト権限の付与は、ロールオブジェクトの DDL に含まれます。

付与は現在ユーザーオブジェクトとロールオブジェクトにアタッチされているため、ユーザーに権限が付与されている 1 つ以上のオブジェクトに障害が発生すると、ユーザーオブジェクト自体が評価に失敗します。この問題を解決するには、個々のステートメントに SQL 単一行コメント (--) を使用するか、連続した grant ステートメントに SQL 複数行コメント (/ * */) を使用して、失敗したオブジェクトの grant ステートメントをコメントアウトし、オブジェクトを再評価できます。実行されます。失敗した付与ステートメントに対応する失敗したオブジェクトが修復された後、以前に失敗した付与ステートメントのコメントを解除し、ユーザーオブジェクトを再評価できます。

5.1.6 特権

スキーマをターゲット EDB Postgres Advanced Server データベースに移行するには、移行の実行に使用されているターゲットデータベースユーザーに次の権限が必要です。

- プロファイルを移行する SUPERUSER
- ユーザーとロールを移行する SUPERUSER または CREATE ROLE

5.1.7 パスワード

ユーザーを移行する場合、Migration Portal はユーザーのパスワードを移行しません。ユーザーは、パスワードなしでターゲット EDB Postgres Advanced Server データベースに作成されます。ユーザーにパスワードを構成することが必要または必要な場合は、ユーザーの移行後にターゲットデータベースで手動で行うことができます。

5.1.8 プロファイル

Migration Portal は、スキーマユーザーに割り当てられた non-DEFAULT プロファイルを移行します。Oracle プロファイルには、パスワードとリソースの両方の制限が含まれます。EDB Postgres Advanced Server は、Oracle パスワード関連の制限のみをサポートします。リソース制限はプロファイルとともに抽出されますが、パスワード制限のみが Migration Portal によってアップロード、評価、移行されます。また、Oracle データベースでは、DEFAULT プロファイルが新しい制限値でオーバーライドされている可能性があります。DEFAULT プロファイルは、Migration Portal によって移行されません。EDB Postgres Advanced Server の DEFAULT プロファイルを、Oracle DEFAULT プロファイルに適用されている制限オーバーライドで構成することが望ましい場合は、ALTER PROFILE ステートメントを使用して、ターゲット EDB Postgres Advanced Server データベースに手動で適用する必要があります。

注釈: バージョン 4.3.0 以降、Migration Portal はユーザー、ロール、プロファイル、および付与の評価をサポートしています。ただし、Migration Portal の以前のバージョン (4.3.0 より前) を使用して作成されたプロジェクトの場合、この機能はサポートされていません。ユーザー、ロール、プロファイル、および権限を評価するには、最新バージョンの EDB DDL エクストラクタを使用して、新しいプロジェクトを作成します。

5.1.9 Oracle のデフォルトケースを使用

Oracle のデフォルトのケース機能を使用するオプションは、Oracle データベースから EDB Postgres Advanced Server データベースにスキーマを移行するときに、すべてのデータベースオブジェクトの Oracle のデフォルトの大文字の命名規則を保持します。

EDB DDL エクストラクタおよび Oracle Data Pump ユーティリティは、PL/SQL の本体内で参照されるオブジェクト名を除き、Oracle のデフォルトのケース (つまり、オブジェクト名の大文字、または大文字と小文字を区別するオブジェクト名のそれぞれのケース) ですべての DDL を抽出します DDL の作成時にユーザーが明示的に二重引用符で囲まない限り、オブジェクト。

そのため、Migration Portal はこの機能のソーススキーマファイルからのすべての二重引用符を保持し、さらに、Migration Portal は二重引用符を次のものに適用します。

- ビューおよびマテリアライズドビューの SELECT ステートメント内の列名。
- チェック制約内の列名。
- トリガーヘッダー内のテーブル名と列名。

Oracle の大文字と小文字を使用するオプションが選択されていない場合、Migration Portal のデフォルトの動作は、すべてのオブジェクトが小文字 (EDB Postgres Advanced Server のデフォルトのケース) で作成されるか、オブジェクト名が引用符で囲まれた大文字と小文字を混合して指定されている場合は大文字と小文字が混合されます。ソース DDL 内。

useOraCase 機能を使用して、以下のソースとターゲットの DDL を見つけます。

```
-- Source DDL
CREATE TABLE "RHS_SANITY"."DEPT"
```

(次のページに続く)

```

(   "DEPTNO" NUMBER(2,0),
    "DNAME" VARCHAR2(14),
    "LOC" VARCHAR2(14),
    CONSTRAINT "DEPTDEPTNO_PK" PRIMARY KEY ("DEPTNO")
USING INDEX ENABLE,
    CONSTRAINT "DEPTDNAME_UK" UNIQUE ("DNAME")
USING INDEX ENABLE
) ;

-- Target DDL
CREATE TABLE "RHS_SANITY"."DEPT"
(   "DEPTNO" NUMBER(2,0),
    "DNAME" VARCHAR2(14),
    "LOC" VARCHAR2(14),
    CONSTRAINT "DEPTDEPTNO_PK" PRIMARY KEY ("DEPTNO")
USING INDEX ENABLE,
    CONSTRAINT "DEPTDNAME_UK" UNIQUE ("DNAME")
USING INDEX ENABLE
) ;

```

Oracle のデフォルトのケースを使用している間、他のプロジェクトと比較して互換性率が低下する場合があります。この互換性比の低下に寄与する可能性のある理由は次のとおりです。

- 最も一般的な理由は、オブジェクト/リレーションが存在しないことです。このシナリオは、テーブル/オブジェクトが Oracle のケースを保持するために二重引用符で囲まれているのに対し、失敗したオブジェクトでは二重引用符なしで参照されていることが原因です。例：

```

CREATE TABLE "HR"."EMP"
(   "EMPNO" NUMBER(4,0),
    "ENAME" VARCHAR2(10),
    "JOB" CLOB,
    "MGR" NUMBER(4,0),
    "HIREDATE" DATE,
    "SAL" NUMBER(7,2),
    "COMM" NUMBER(7,2),
    "DEPTNO" NUMBER(2,0),
    CONSTRAINT "PK_EMP" PRIMARY KEY ("EMPNO")
USING INDEX ENABLE
) ;

CREATE OR REPLACE PROCEDURE "HR"."PROC"
IS
CURSOR emp_cur IS SELECT hiredate, CAST(hiredate AS timestamp(6)) AS timestamped_
→FROM emp;
REC emp_cur%rowtype;

```

(前のページからの続き)

```

BEGIN
  OPEN emp_cur;
  LOOP
    FETCH emp_cur INTO rec;
    EXIT WHEN emp_cur%NOTFOUND ;
    DBMS_OUTPUT.PUT_LINE(rec.hiredate || ||rec.timestamped);
  END LOOP;
  CLOSE EMP_CUR;
END ;

```

エラーは次のように表示されます。

edb_notranlate_2 上記の例では、テーブル名とその列名は、CREATE TABLE ステートメントで引用符で囲まれた大文字の名前で指定されています。ただし、CREATE OR REPLACE PROCEDURE ステートメントでは、テーブルとその列は、引用符で囲まれていない小文字の識別子を使用して参照されます。

- %type を使用して定義された変数は、コードオブジェクトで失敗します。先行する table.column-name は employees.employee_id%type のように小文字ですが、実際のテーブルのテーブル名と列名は二重引用符で囲まれた大文字です。例：

```

CREATE TABLE "HR"."EMPLOYEES"
(
  "EMPLOYEE_ID" NUMBER(6,0),
  "FIRST_NAME" VARCHAR2(20),
  "LAST_NAME" VARCHAR2(25) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE,
  "EMAIL" VARCHAR2(25) CONSTRAINT "EMP_EMAIL_NN" NOT NULL ENABLE,
  "PHONE_NUMBER" VARCHAR2(20),
  "HIRE_DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL ENABLE,
  "JOB_ID" VARCHAR2(10) CONSTRAINT "EMP_JOB_NN" NOT NULL ENABLE,
  "SALARY" NUMBER(8,2),
  "COMMISSION_PCT" NUMBER(2,2),
  "MANAGER_ID" NUMBER(6,0),
  "DEPARTMENT_ID" NUMBER(4,0),
  CONSTRAINT "EMP_SALARY_MIN" CHECK ("SALARY" > 0) ENABLE,
  CONSTRAINT "EMP_EMP_ID_PK" PRIMARY KEY ("EMPLOYEE_ID")
  USING INDEX ENABLE,
  CONSTRAINT "EMP_EMAIL_UK" UNIQUE ("EMAIL")
  USING INDEX ENABLE
);

CREATE OR REPLACE PROCEDURE "HR"."PKG" AS
-- declare variables for data fetched from cursor
empid      employees.employee_id%TYPE; -- variable for employee_id
hiredate   employees.hire_date%TYPE;   -- variable for hire_date
firstname  employees.first_name%TYPE;  -- variable for first_name
lastname   employees.last_name%TYPE;   -- variable for last_name
rowcount   NUMBER;
bonusamount NUMBER;

```

(次のページに続く)

(前のページからの続き)

```

yearsworked NUMBER;
-- declare the cursor with a parameter
CURSOR cursor1 (thismonth NUMBER)IS
    SELECT employee_id, first_name, last_name, hire_date FROM employees
        WHERE EXTRACT(MONTH FROM hire_date) = thismonth;
BEGIN
    null;
END;

```

エラーは次のように表示されます。

```
syntax error at or near "%" (line 3, char 37)
```

- テーブルの列のデータ型に TYPE 参照がある場合、大文字と小文字の違いにより失敗します。例：

```

CREATE OR REPLACE TYPE "HR"."USER_TYPE" IS OBJECT(
    "ID" NUMBER,
    "NAME" VARCHAR2(256)
);

CREATE TABLE "HR"."MY_TABLE"
(
    "ID" NUMBER,
    "MY_NAME" user_type
);

```

エラーは次のように表示されます。

```

type "user_type" does not exist
LINE 4:      "MY_NAME" user_type
              ^
SQL state: 42704
Character: 58

```

5.2 EDB DDL エクストラクタ

- EDB DDL エクストラクタスクリプトは、Flashback を使用して復元されたオブジェクトで、まだ BIN\$b54+4XlEYwPgUAB/AQBWwA==\$0 のような名前が付いているオブジェクトを抽出しません。これらのオブジェクトを抽出する場合は、オブジェクトの名前を変更し、抽出プロセスを再実行する必要があります。
- EDB DDL エクストラクタは、nologging テーブルを通常のテーブルとして抽出します。これらのテーブルが EDB Postgres Advanced Server に移行されると、WAL ログファイルが作成されます。

- EDB DDL エクストラクタは、VALID ステータスのオブジェクトのみを抽出します。Migration Portal で評価する INVALID ステータスを持つオブジェクトについては、最初にそれらを VALID に更新します。
- EDB DDL エクストラクタは、Oracle ラップ機能を使用して難読化されたオブジェクトを抽出しません。そのため、これらのオブジェクトは、Migration Portal によって評価される DDL のセットには含まれません。これらのオブジェクトを評価して EDB Postgres Advanced Server に移行する場合、オブジェクトのラップバージョンをラップされていないバージョンに置き換えます。詳細については、[ラップされたオブジェクト](#) を参照してください。
- EDB DDL エクストラクタは、スキーマ名とその依存関係情報を格納するグローバル一時テーブルを作成します。これらのテーブルは、抽出が正常に終了すると削除されます。
- EDB DDL エクストラクタスクリプトは、PostgreSQL がサポートしていないため、名前が PG_ で始まるスキーマを抽出しません。これらのスキーマを抽出する場合は、抽出前にスキーマの名前を変更する必要があります。
- EDB DDL エクストラクタは、プロファイル、ロール、および付与の情報を自動的に抽出します。
- EDB DDL エクストラクタは現在、Oracle 11g からの ROLES、SYSTEM GRANTS ON ROLES、OBJECT GRANTS ON ROLES、および ROLE GRANTS の抽出をサポートしていません。これにより、抽出されたファイルのこれらのオブジェクトタイプに対応するセクションにエラーメッセージが書き込まれます。これらのエラーは、Migration Portal によるこれらのファイルの評価に問題を引き起こしません。

5.3 Oracle Data Pump ユーティリティ

- Oracle の引用演算子で IDENTIFIED BY 句を使用してデータベースリンクを作成する場合、Migration Portal は SQL ファイルの解析に失敗する場合があります。たとえば、IDENTIFIED BY VALUES q'[:1] '。ファイルを正常に解析するには、実際のパスワードを使用してみてください。たとえば、IDENTIFIED BY my_password。
- Oracle Data Pump ユーティリティによって生成された DDL には、Migration Portal によって処理されない ALTER FUNCTION、ALTER PACKAGE、ALTER TYPE などの ALTER STATEMENTS が含まれる場合があります。
- Oracle Data Pump がスキーマモードで実行されている場合、つまり、expdp コマンドの実行時に SCHEMAS パラメーターが使用されている場合、プロファイルとロールが抽出されません。エクスポートされるスキーマユーザーにプロファイルが割り当てられている場合、プロファイルはエクスポートされる impdp ファイルの一部でないため、Migration Portal でのスキーマユーザーオブジェクトの評価は失敗します。この問題を解決するには、ユーザーオブジェクトターゲット DDL のプロファイル割り当てを削除し、ユーザー作成オブジェクトを再評価します。
- スキーマモードで impdp または expdp を使用して、Oracle 11g から ROLES、SYSTEM GRANTS ON ROLES、OBJECT GRANTS ON ROLES、および ROLE GRANTS を抽出しようとすると、エラーが表示される場合があります。回避策は、full=y オプションを使用することです。これにより、すべてのスキーマと前述のオブジェクトタイプがエラーなしで抽出されます。