EDB Postgres™ Migration Portal Guide

Version 1.0.0

February 19, 2019

EDB Postgres™ Migration Portal Guide
by EnterpriseDB® Corporation

EnterpriseDB Corporation, 34 Crosby Drive, Suite 100, Bedford, MA 01730, USA
T +1 781 357 3390   F +1 978 467 1307   E info@enterprisedb.com www.enterprisedb.com

Table of Contents

1 Introduction

EDB Postgres™ Migration Portal (Migration Portal) is a web-based tool for migrating Oracle database schemas to the EDB Postgres platform. The Migration Portal assesses and analyzes Oracle database schemas and converts types, tables, sequences, constraints, triggers, views, stored procedures, packages, dblinks, materialized views, and indexes, producing DDLs that are compatible with EDB Postgres Advanced Server.

The user-friendly portal interface simplifies migration; log on to the portal with your browser of choice and start the migration process.

The EDB Postgres™ Migration Portal guide provides a high-level description of the steps involved in the migration process. The guide also includes solutions to common migration problems, and details unsupported features and their potential workarounds.

EnterpriseDB has helped companies migrate their existing database systems to Postgres for years. For more information, visit the EnterpriseDB website at:

https://www.enterprisedb.com/

1.1 Typographical Conventions Used in this Guide

Certain typographical conventions are used in this manual to clarify the meaning and usage of various commands, statements, programs, examples, etc. This section provides a summary of these conventions.

In the following descriptions a *term* refers to any word or group of words that are language keywords, user-supplied values, literals, etc. A term's exact meaning depends upon the context in which it is used.

- *Italic font* introduces a new term, typically, in the sentence that defines it for the first time.
- `Fixed-width (mono-spaced) font` is used for terms that must be given literally such as SQL commands, specific table and column names used in the examples, programming language keywords, etc. For example, `SELECT * FROM emp;`
- *`Italic fixed-width font`* is used for terms for which the user must substitute values in actual usage. For example, `DELETE FROM` *`table_name`*;
- A vertical pipe | denotes a choice between the terms on either side of the pipe. A vertical pipe is used to separate two or more alternative terms within square brackets (optional choices) or braces (one mandatory choice).
- Square brackets [ ] denote that one or none of the enclosed term(s) may be substituted. For example, `[ a | b ]`, means choose one of "`a`" or "`b`" or neither of the two.
- Braces {} denote that exactly one of the enclosed alternatives must be specified. For example, `{ a | b }`, means exactly one of "`a`" or "`b`" must be specified.
- Ellipses ... denote that the proceeding term may be repeated. For example, `[ a | b ] ...` means that you may have the sequence, "`b a a b a`".

1.2 What's New

The following enhancements are added to the EDB Postgres Migration Portal v1.0.0:

- Starting this release, Migration Portal will be generally available.
- Migration Portal is now integrated with Cloud Database Service (CDS). So now you can deploy the assessed schema directly to the CDS cluster through Migration Portal. For more information see Migrating schema to a cluster.
- EDB DDL Extractor will not extract wrapped objects.
- Following are the new repair handlers added to improve Advanced Server compatibility ratio:
    - ERH 1006: Converts `RETURN` statement in a trigger to `RETURN :new`.
    - ERH 1007: Removes trigger name from the `END` clause, if the trigger name is different in the `CREATE TRIGGER` clause.
    - ERH 1008: Removes label name form the `END` clause in `SPL` objects.
    - ERH 2052: Removes unsupported compress or nocompress options from `CREATE INDEX` statement.
    - ERH 2053: Removes unsupported parallel or noparallel options from `CREATE INDEX` statement.
    - ERH 2054: Removes `SCALE` from the create sequence DDL statement.
    - ERH 2055: Removes `USING INDEX LOCAL` clause from the partitioned table DDL statement.
    - ERH 2056: Removes `REVERSE` clause from index DDL statement.
    - ERH 2057: Removes size specification from integer types in SPL objects. For example `p_var INTEGER (1)`; is modified to `p_var INTEGER` ;
    - ERH 2058 - Converts `PIPELINED` clause in pipelined functions to `SETOF RECORD.`
      For example `RETURN` number_ntt `PIPELINED` is converted to `RETURN SETOF RECORD.`
    - ERH 2059 - Converts `PIPE ROW` statement in pipelined functions to `RETURN NEXT.`

For example `PIPE ROW(csvString);` is converted to `RETURN NEXT(csvString);`

- 2060 - Removes `SHARING=METADATA` clause from the source DDL.

1.3 Supported Browsers, Operating Systems, and Database

The Migration Portal supports migration from Oracle 11 and 12c to EDB Advanced Server 10 or 11. Migration Portal is supported on the following browsers and operating systems:

Supported Browsers

For the best user experience, we recommend using the Google Chrome browser. Migration Portal is supported on the following browsers:

Browser	Supported Version
Apple Safari on Macintosh OS	11 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above
Internet Explorer	11 and above

Supported Operating Systems

Operating Systems	Supported Version
Macintosh	OSX Sierra
Windows	10
Linux	CentOs 7

2 Using the EDB Migration Portal

The Migration Portal allows you to easily migrate your database from Oracle to Advanced Server. You can upload schemas for assessment and get immediate feedback and suggestions. The portal allows you to download assessed DDLs for all objects and create your EDB Postgres database on-premises or in the cloud.

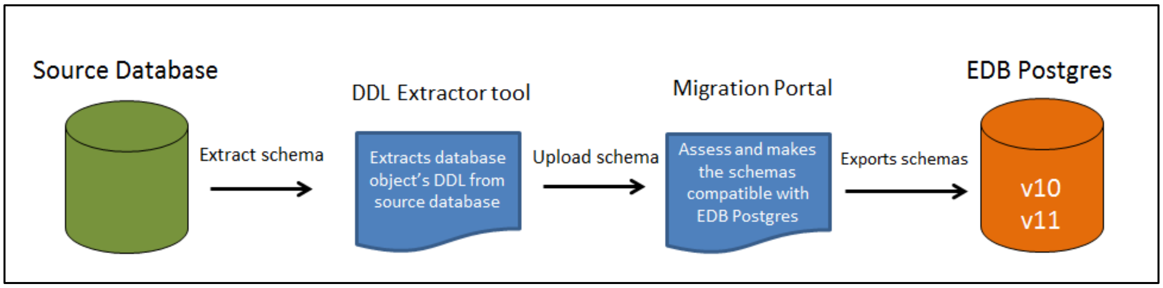
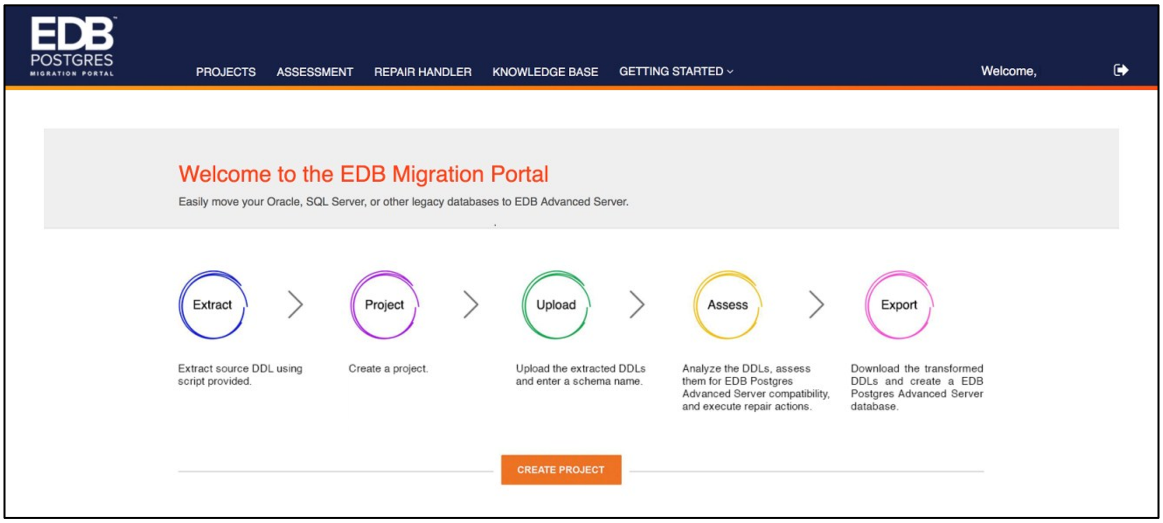


Figure 2 – The migration process.

To access the migration portal, open any of the browsers and navigate to:

https://migration.enterprisedb.com/

2.1 Overview of the Migration Portal



Welcome to the EDB Migration Portal

Easily move your Oracle, SQL Server, or other legacy databases to EDB Advanced Server.

Extract source DDL using script provided.

Create a project.

Upload the extracted DDLs and enter a schema name.

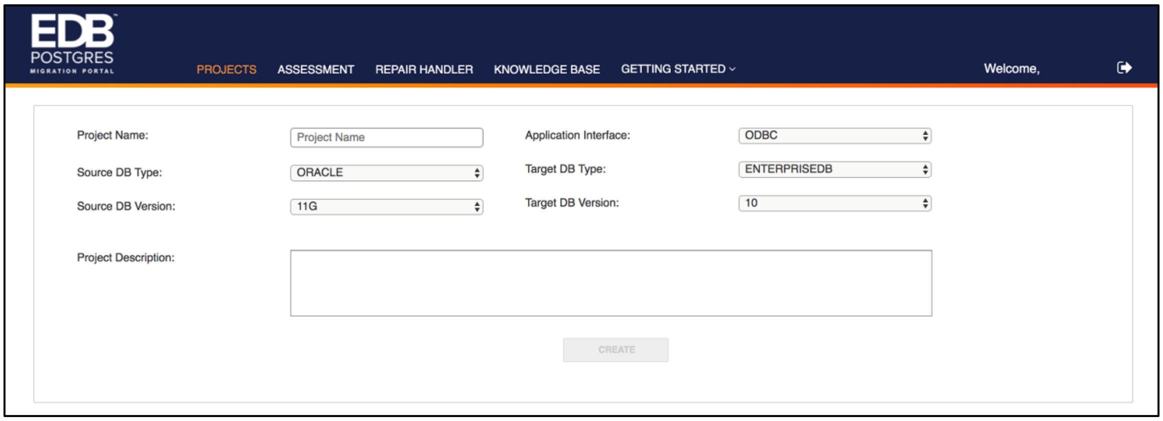
Analyze the DDLs, assess them for EDB Postgres Advanced Server compatibility, and execute repair actions.

Download the transformed DDLs and create a EDB Postgres Advanced Server database.

Figure 2.1 – Migration Portal

The following list provides information about the tabs displayed on the Migration Portal.

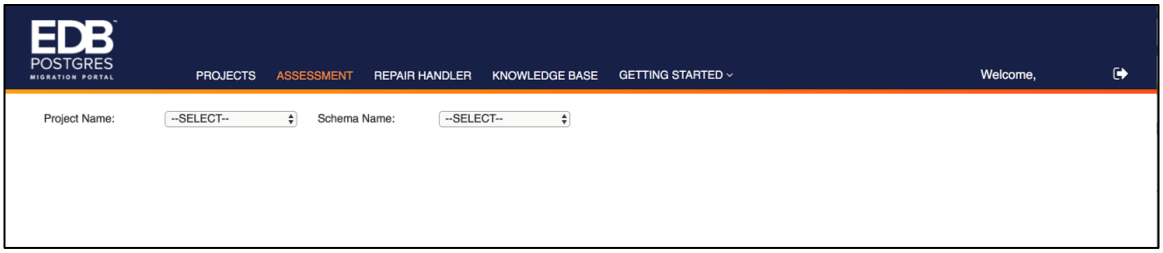
- **Projects**: The `Project` tab is used to create a project for assessing the existing schema.



Project Name:	Project Name	Application Interface:	ODBC
Source DB Type:	ORACLE	Target DB Type:	ENTERPRISEDB
Source DB Version:	11G	Target DB Version:	10
Project Description:			

Figure 2.2 – Project page

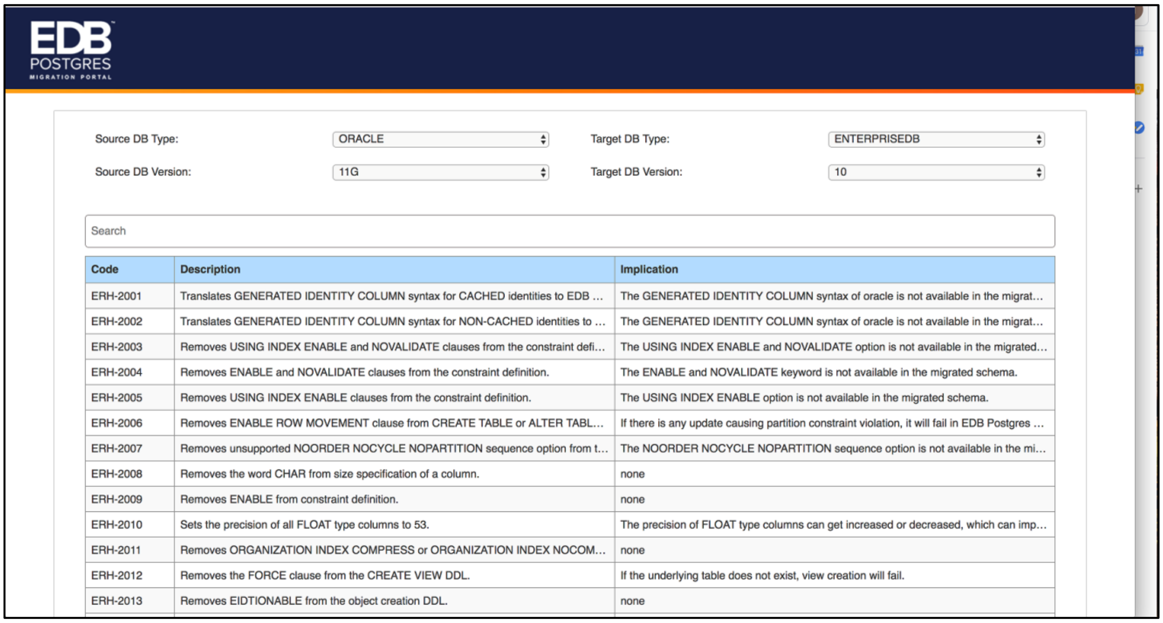
- **Assessment**: The `Assessment` tab is used to assess the migrated schema. You can check the details for already assessed projects or schemas, and reassess the individual DDLs.



Project Name: --SELECT--   Schema Name: --SELECT--

Figure 2.3 –Assessment page

- **Repair Handler**: The `Repair Handler` tab is used to review the transformations that the migration portal performs to make the source DDL compatible with the target database.

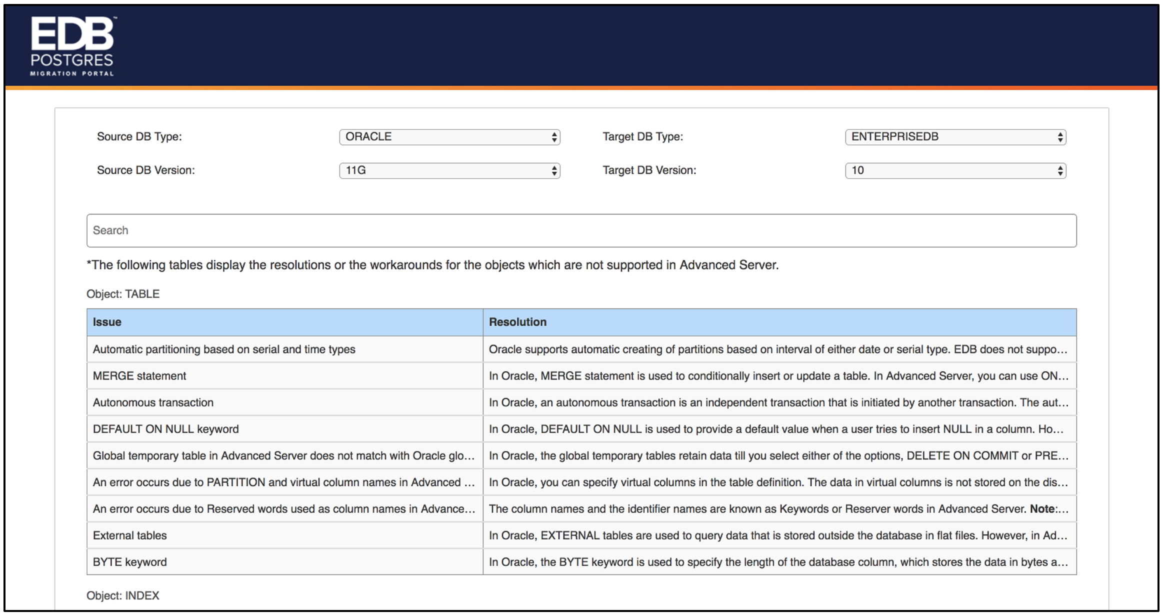


Source DB Type: ORACLE
Target DB Type: ENTERPRISEDB
Source DB Version: 11G
Target DB Version: 10

Code	Description	Implication
ERH-2001	Translates GENERATED IDENTITY COLUMN syntax for CACHED identities to EDB ...	The GENERATED IDENTITY COLUMN syntax of oracle is not available in the migrat...
ERH-2002	Translates GENERATED IDENTITY COLUMN syntax for NON-CACHED identities to ...	The GENERATED IDENTITY COLUMN syntax of oracle is not available in the migrat...
ERH-2003	Removes USING INDEX ENABLE and NOVALIDATE clauses from the constraint defi...	The USING INDEX ENABLE and NOVALIDATE option is not available in the migrated...
ERH-2004	Removes ENABLE and NOVALIDATE clauses from the constraint definition.	The ENABLE and NOVALIDATE keyword is not available in the migrated schema.
ERH-2005	Removes USING INDEX ENABLE clauses from the constraint definition.	The USING INDEX ENABLE option is not available in the migrated schema.
ERH-2006	Removes ENABLE ROW MOVEMENT clause from CREATE TABLE or ALTER TABL...	If there is any update causing partition constraint violation, it will fail in EDB Postgres ...
ERH-2007	Removes unsupported NOORDER NOCYCLE NOPARTITION sequence option from t...	The NOORDER NOCYCLE NOPARTITION sequence option is not available in the mi...
ERH-2008	Removes the word CHAR from size specification of a column.	none
ERH-2009	Removes ENABLE from constraint definition.	none
ERH-2010	Sets the precision of all FLOAT type columns to 53.	The precision of FLOAT type columns can get increased or decreased, which can imp...
ERH-2011	Removes ORGANIZATION INDEX COMPRESS or ORGANIZATION INDEX NOCOM...	none
ERH-2012	Removes the FORCE clause from the CREATE VIEW DDL.	If the underlying table does not exist, view creation will fail.
ERH-2013	Removes EIDTIONABLE from the object creation DDL.	none

Figure 2.4 – Repair Handler page

- **Knowledge Base**: The `Knowledge Base` tab is used to find workarounds or solutions for the objects that failed in the assessment process.



Source DB Type: ORACLE

Target DB Type: ENTERPRISEDB

Source DB Version: 11G

Target DB Version: 10

Search

*The following tables display the resolutions or the workarounds for the objects which are not supported in Advanced Server.

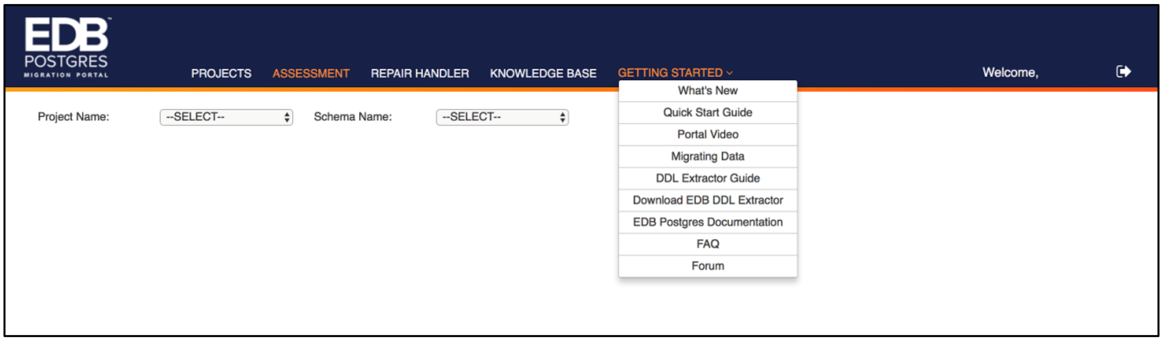
Object: TABLE

Issue	Resolution
Automatic partitioning based on serial and time types	Oracle supports automatic creating of partitions based on interval of either date or serial type. EDB does not suppo…
MERGE statement	In Oracle, MERGE statement is used to conditionally insert or update a table. In Advanced Server, you can use ON…
Autonomous transaction	In Oracle, an autonomous transaction is an independent transaction that is initiated by another transaction. The aut…
DEFAULT ON NULL keyword	In Oracle, DEFAULT ON NULL is used to provide a default value when a user tries to insert NULL in a column. Ho…
Global temporary table in Advanced Server does not match with Oracle glo…	In Oracle, the global temporary tables retain data till you select either of the options, DELETE ON COMMIT or PRE…
An error occurs due to PARTITION and virtual column names in Advanced …	In Oracle, you can specify virtual columns in the table definition. The data in virtual columns is not stored on the dis…
An error occurs due to Reserved words used as column names in Advance…	The column names and the identifier names are known as Keywords or Reserver words in Advanced Server. **Note**:…
External tables	In Oracle, EXTERNAL tables are used to query data that is stored outside the database in flat files. However, in Ad…
BYTE keyword	In Oracle, the BYTE keyword is used to specify the length of the database column, which stores the data in bytes a…

Object: INDEX

Figure 2.5 – Knowledge Base page

- **Getting Started**: The options in the `Getting Started` tab provide access to documentation and other information links.



Project Name: --SELECT--

Schema Name: --SELECT--

Figure 2.6 – Getting Started tab

3 Migrating a Database

To migrate a database, you must complete the following steps:

- Schema Extraction
- Schema Assessment
- Schema Migration
- Data Migration

3.1 Schema Extraction

Prerequisites

For schema extraction, you must download the latest EDB DDL Extractor tool from the Getting Started menu on the EnterpriseDB website:

www.migration.enterprisedb.com

The SQL script will extract data definitions, stored procedures, views, etc., from an Oracle database into text file.

The DDL Extractor for Oracle database is used as a part of EDB Migration Portal. The EDB DDL extractor creates the DDL file that will be uploaded to the portal and analyzed for EDB Postgres compatibility.

The EDB DDL Extractor for Oracle database uses Oracle's `DBMS_METADATA` built-in package when extracting DDLs.

Please note: You must have `SELECT CATALOG ROLE` or `SELECT ANY DICITIONARY` privileges in the Oracle database.

To extract the schema, complete the following steps:

1. Download the EDB DDL Extractor tool for Oracle database as: `edb_ddl_extractor.sql`
2. Connect to SQL*Plus with user having `SELECT_CATALOG_ROLE` and `SELECT ANY DICTIONARY` privileges and run the command: `SQL>@edb_ddl_extractor.sql`

3. Provide the schema name and the path\directory in which the extractor will store the extracted DDL. For multiple schema extraction, you must use (‘,’) delimiter.

For example, on Linux:

Enter `SCHEMA NAME[S]` (use ‘,’ delimiter for multiple schemas) to extract DDLs:
`HR, SCOTT, FINANCE`
Enter the `PATH` to store DDL file: `/home/oracle/extracted_ddls/`

On Windows:

Enter `SCHEMA NAME[S]` (use ‘,’ delimiter for multiple schemas) to extract DDLs:
`HR, SCOTT, FINANCE`
Enter the `PATH` to store DDL file: `c\Users\Example\Desktop\`

Please Note: You can also enter a single schema name.

The script iterates through the object types in the database and once the task is completed, the .SQL output is stored at the entered location, (i.e., `c:\Users\Example\Desktop\`).

EDB DDL Extractor does not extract objects that have names like `BIN$b54+4XIEYwPgUAB/AQBWwA= =$0`. If you want to extract these objects, you must change the name of the objects and re-run the extraction process.

3.1.1 Supported and Unsupported Object Types

The migration portal supports migration of the following object types:

- Synonyms
- DB Links
- Types and Type Body
- Sequences
- Tables
- Constraints
- Indexes (Except LOB indexes and indexes on materialized views)
- Views
- Materialized Views
- Triggers
- Functions
- Procedures
- Packages

The portal does not support migration of the following object types:

- Editions
- Operators
- Schedulers
- LOB indexes and indexes on materialized views
- XML Schemas
- Profiles
- Role and Object Grants
- Tablespaces
- Directories
- Users
- RLS Policy
- Queues

3.2 Schema Assessment

To assess an Oracle database schema for compatibility with Advanced Server, you must:

1. Connect to your Oracle database using SQL*Plus.

   Please Note: You must connect with a database role that has `SELECT_CATALOG_ROLE or SELECT ANY DICTIONARY` privileges.

2. Run the DDL Extractor with the command: `SQL>@edb_ddl_extractor.sql.`

   The EDB DDL Extractor uses the SQL*Plus `ACCEPT` command to get user input of multiple schema names using ',' delimiter. Maximum 240 bytes can be accepted. So, the maximum length of schema name can be of 30 characters and you can input 7 to 8 schema names.

3. Enter the path where the extraction file will be created.

   On Linux:
   Enter the schema name to extract the DDLs : `schema_name`
   Enter the path to store DDL file: `/home/oracle/extracted_ddls/`

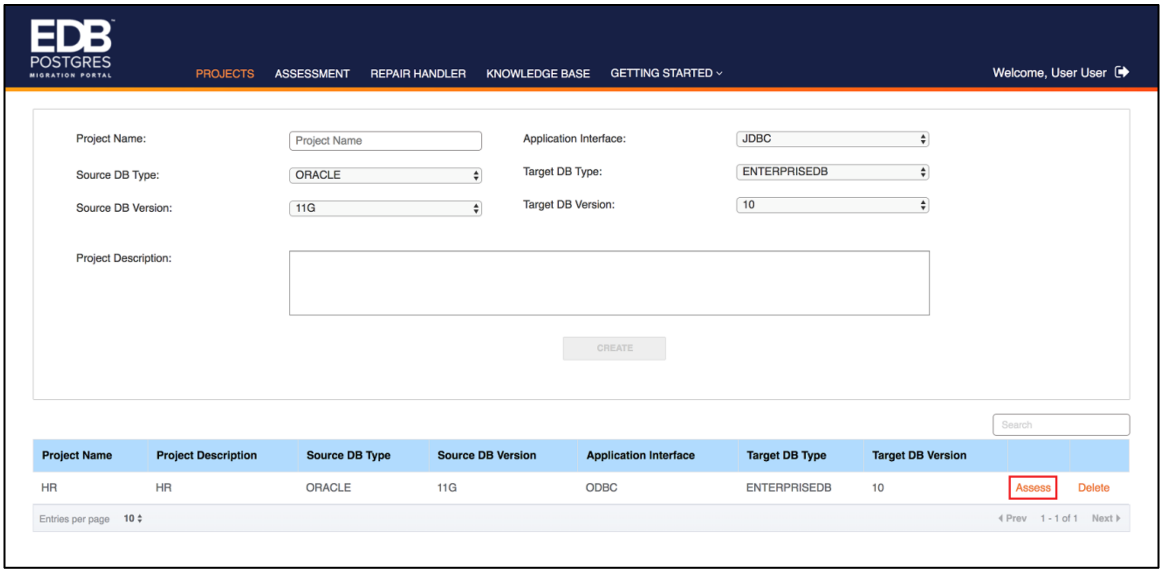
   On Windows:
   Enter schema name to extract the DDLs : HR
   Enter the path to store DDL file: `C:/Extracted_DDL/`

4. Go to https://migration.enterprisedb.com.

5. Enter your EDB credentials.

6. Click `CREATE PROJECT` to create a new project.

7. Enter the project name, project description, and click `Assess`.



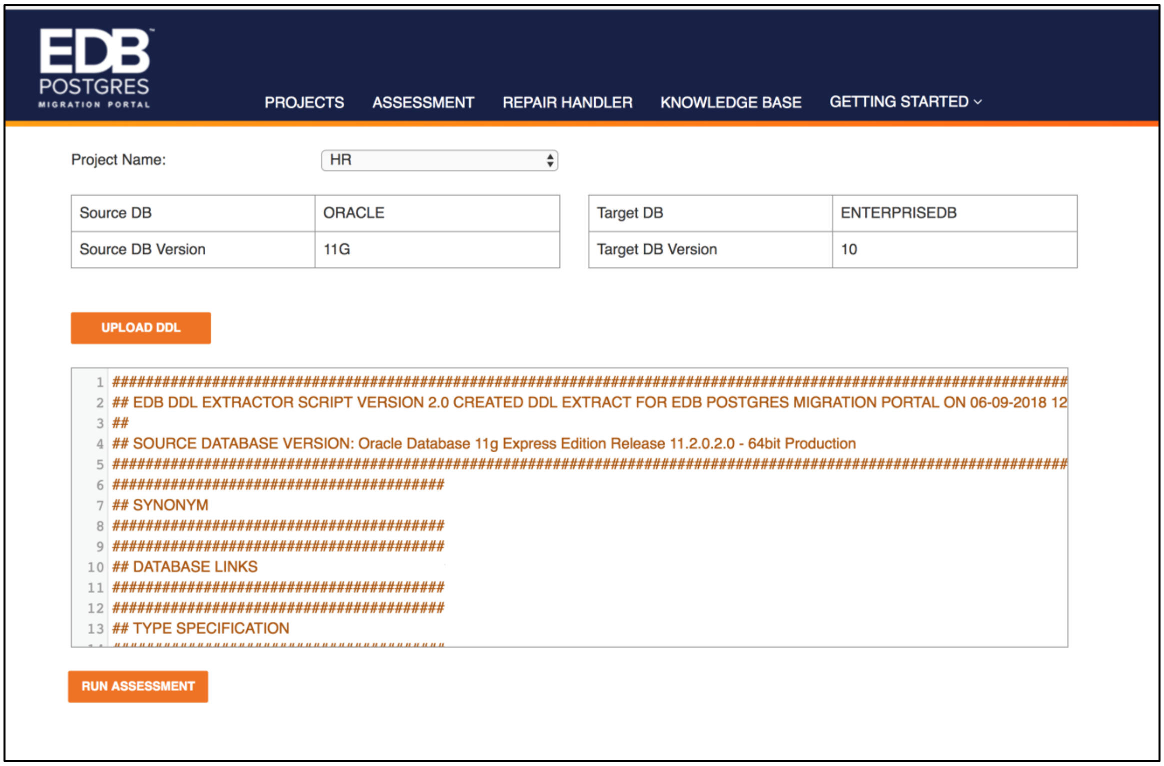
Project Name:	Project Name	Application Interface:	JDBC
Source DB Type:	ORACLE	Target DB Type:	ENTERPRISEDB
Source DB Version:	11G	Target DB Version:	10
Project Description:			

Project Name	Project Description	Source DB Type	Source DB Version	Application Interface	Target DB Type	Target DB Version		
HR	HR	ORACLE	11G	ODBC	ENTERPRISEDB	10	Assess	Delete

Figure 3.1 – Assessing the extracted schema.

8. Upload the `.SQL` file generated by the `EDB DDL Extractor` for Oracle Database. **Please Note**: You should not modify the `.SQL` file.

9. Click `RUN ASSESSMENT`.



Project Name: HR

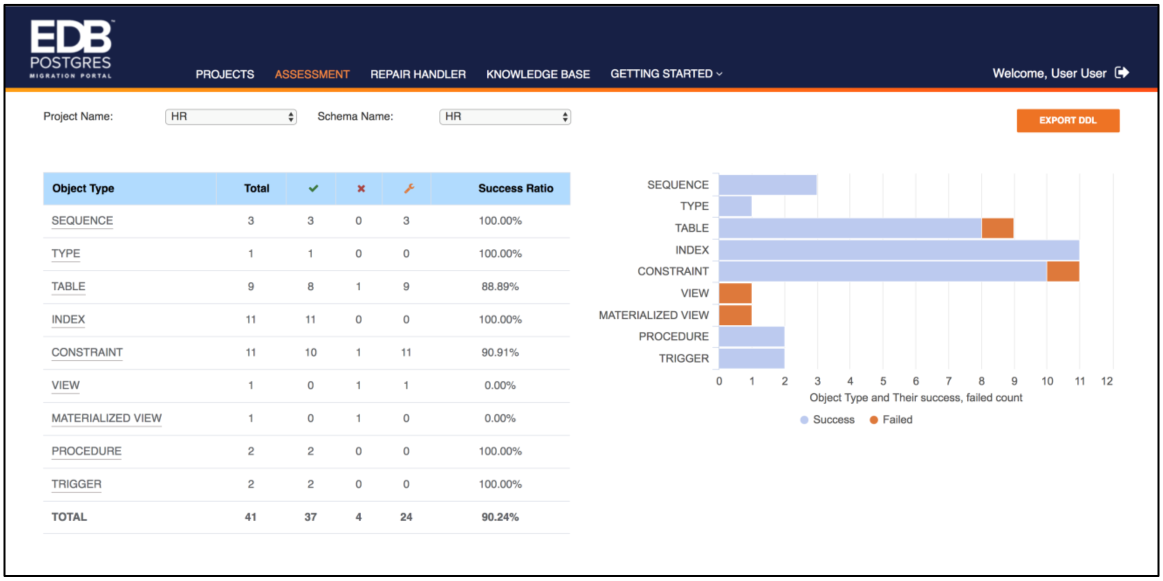
Source DB	ORACLE	Target DB	ENTERPRISEDB
Source DB Version	11G	Target DB Version	10

```
##########################################################################################################
## EDB DDL EXTRACTOR SCRIPT VERSION 2.0 CREATED DDL EXTRACT FOR EDB POSTGRES MIGRATION PORTAL ON 06-09-2018 12
##
## SOURCE DATABASE VERSION: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
##########################################################################################################
########################################
## SYNONYM
########################################
########################################
## DATABASE LINKS
########################################
########################################
## TYPE SPECIFICATION
########################################
```

Figure 3.2 – Uploading the extracted DDL.

The analysis tool will review every construct, execute repair actions to improve compatibility with Advanced Server, and flag any remaining errors that require manual intervention.

10. Verify the DDL objects (e.g., TABLES) that do not show a 100% success ratio.

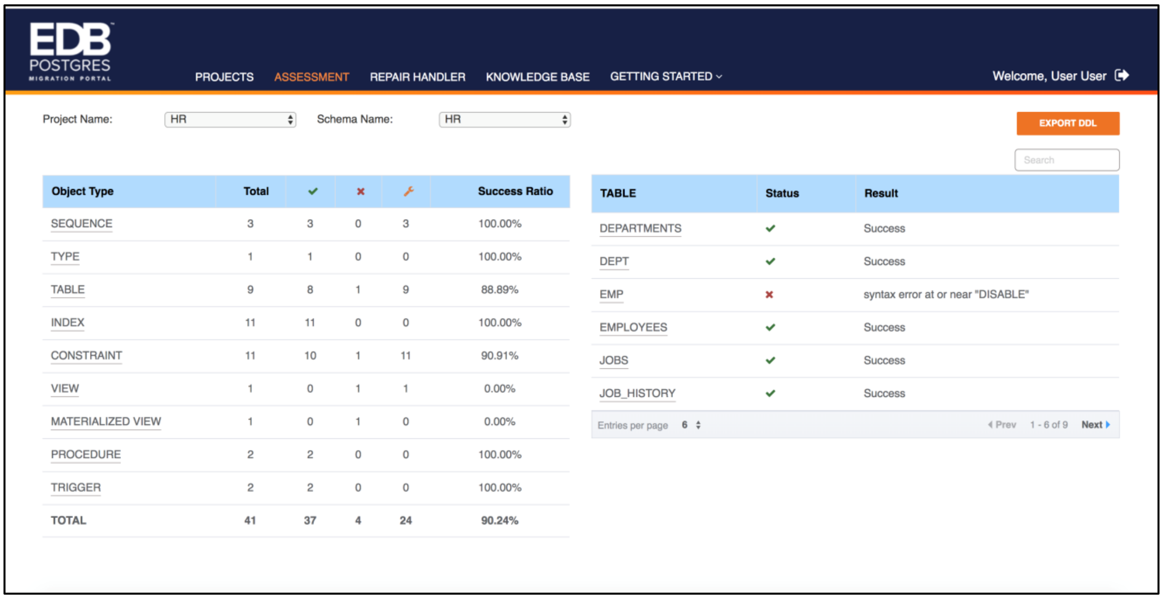


Project Name: HR
Schema Name: HR

Object Type	Total	✓	✗	🔧	Success Ratio
SEQUENCE	3	3	0	3	100.00%
TYPE	1	1	0	0	100.00%
TABLE	9	8	1	9	88.89%
INDEX	11	11	0	0	100.00%
CONSTRAINT	11	10	1	11	90.91%
VIEW	1	0	1	1	0.00%
MATERIALIZED VIEW	1	0	1	0	0.00%
PROCEDURE	2	2	0	0	100.00%
TRIGGER	2	2	0	0	100.00%
TOTAL	41	37	4	24	90.24%

Figure 3.3 – Verifying the DDL objects.

11. Click the objects that are not compatible with EDB Postgres and view the details.



Project Name: HR
Schema Name: HR

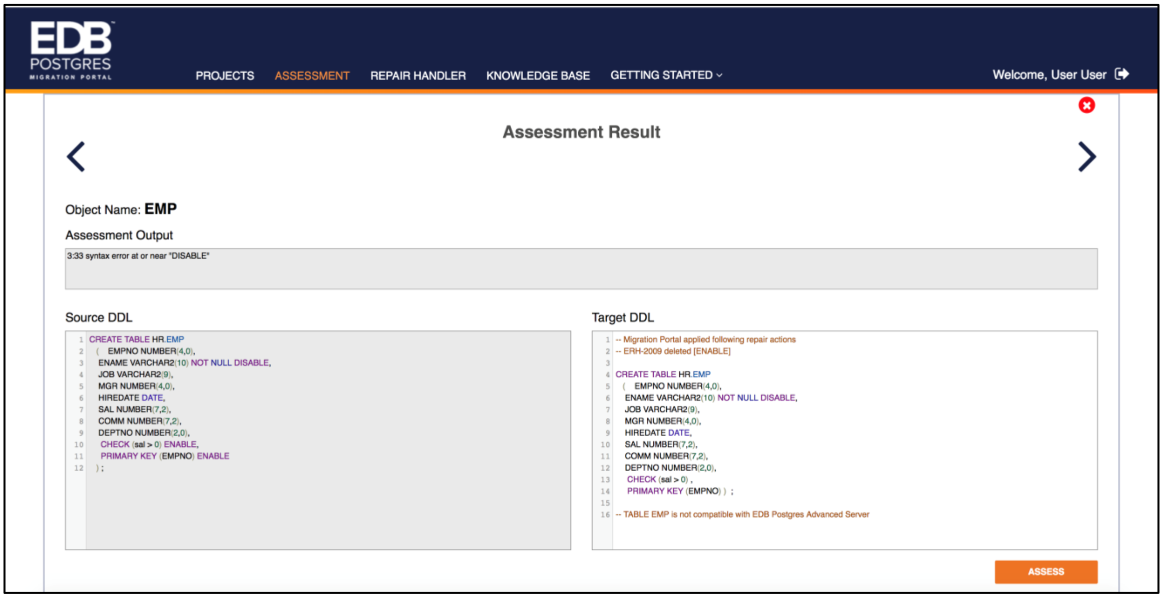
Object Type	Total	✓	✗	🔧	Success Ratio
SEQUENCE	3	3	0	3	100.00%
TYPE	1	1	0	0	100.00%
TABLE	9	8	1	9	88.89%
INDEX	11	11	0	0	100.00%
CONSTRAINT	11	10	1	11	90.91%
VIEW	1	0	1	1	0.00%
MATERIALIZED VIEW	1	0	1	0	0.00%
PROCEDURE	2	2	0	0	100.00%
TRIGGER	2	2	0	0	100.00%
TOTAL	41	37	4	24	90.24%

TABLE	Status	Result
DEPARTMENTS	✓	Success
DEPT	✓	Success
EMP	✗	syntax error at or near "DISABLE"
EMPLOYEES	✓	Success
JOBS	✓	Success
JOB_HISTORY	✓	Success

Entries per page 6 ◂ Prev 1 - 6 of 9 Next ▸

Figure 3.4 – Non-compatible objects.

13. Refer to the Knowledge Base information to check the possible workarounds for the objects that are not immediately compatible with Advanced Server.



Assessment Result

Object Name: **EMP**

Assessment Output

```
3:33 syntax error at or near "DISABLE"
```

Source DDL

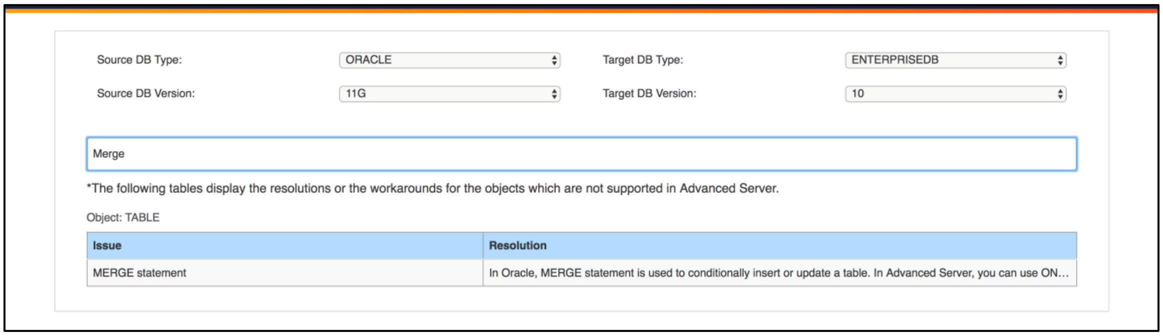
```
CREATE TABLE HR.EMP
  (   EMPNO NUMBER(4,0),
   ENAME VARCHAR2(10) NOT NULL DISABLE,
   JOB VARCHAR2(9),
   MGR NUMBER(4,0),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2,0),
    CHECK (sal > 0) ENABLE,
    PRIMARY KEY (EMPNO) ENABLE
  ) ;
```

Target DDL

```
-- Migration Portal applied following repair actions
-- ERH-2009 deleted [ENABLE]

CREATE TABLE HR.EMP
  (   EMPNO NUMBER(4,0),
   ENAME VARCHAR2(10) NOT NULL DISABLE,
   JOB VARCHAR2(9),
   MGR NUMBER(4,0),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2,0),
    CHECK (sal > 0) ,
    PRIMARY KEY (EMPNO) )  ;

-- TABLE EMP is not compatible with EDB Postgres Advanced Server
```

Figure 3.5 – Assessment result with errors.

14. On the `Knowledge Base` tab, you can enter the object name which is not compatible with Advanced Server and click `Search`.



Source DB Type:	ORACLE	Target DB Type:	ENTERPRISEDB
Source DB Version:	11G	Target DB Version:	10

Merge

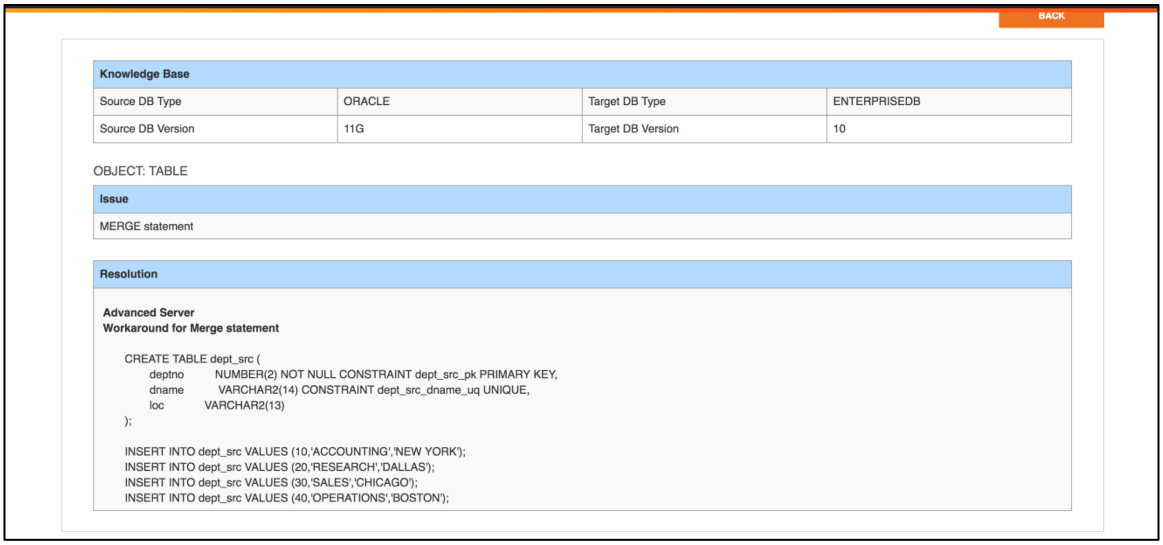
*The following tables display the resolutions or the workarounds for the objects which are not supported in Advanced Server.

Object: TABLE

Issue	Resolution
MERGE statement	In Oracle, MERGE statement is used to conditionally insert or update a table. In Advanced Server, you can use ON…

Figure 3.6 – Searching for object name

15. The object detailed panel displays the workaround or the resolution for the mentioned object. You can manually make the changes on the `Assessment` tab for that object, and click `Assess`.



Knowledge Base			
Source DB Type	ORACLE	Target DB Type	ENTERPRISEDB
Source DB Version	11G	Target DB Version	10

OBJECT: TABLE

Issue
MERGE statement

Resolution

Advanced Server
Workaround for Merge statement

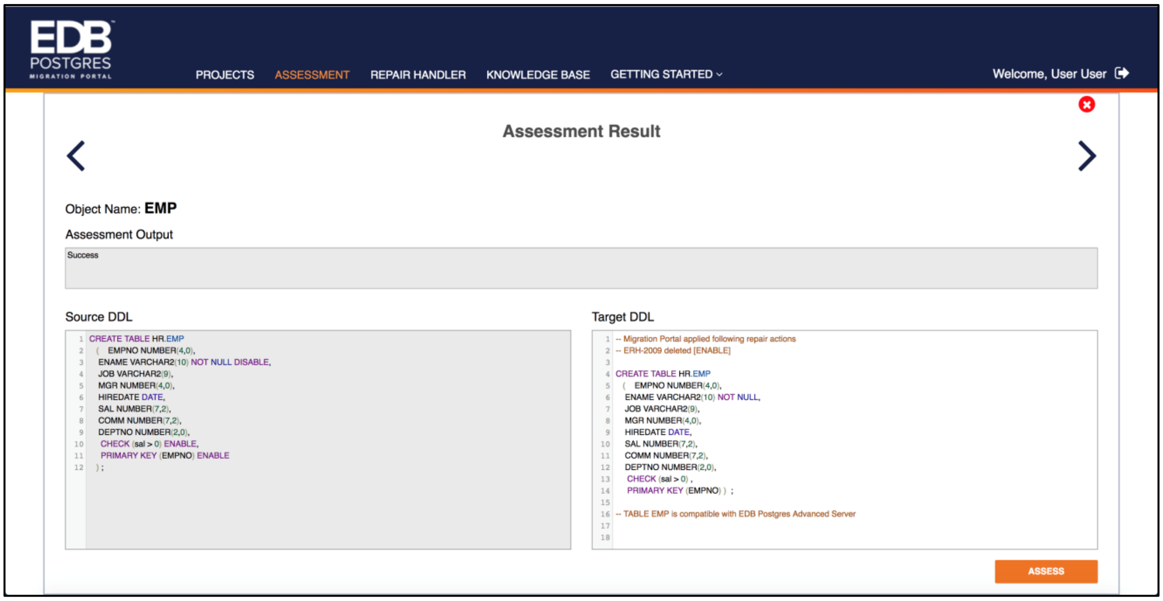
```
CREATE TABLE dept_src (
    deptno      NUMBER(2) NOT NULL CONSTRAINT dept_src_pk PRIMARY KEY,
    dname       VARCHAR2(14) CONSTRAINT dept_src_dname_uq UNIQUE,
    loc         VARCHAR2(13)
);

INSERT INTO dept_src VALUES (10,'ACCOUNTING','NEW YORK');
INSERT INTO dept_src VALUES (20,'RESEARCH','DALLAS');
INSERT INTO dept_src VALUES (30,'SALES','CHICAGO');
INSERT INTO dept_src VALUES (40,'OPERATIONS','BOSTON');
```

Figure 3.7—Workaround or resolution for non-compatible object

Similarly, you can make all the non-compatible objects compatible.

Please Note: In case the object is not available in the Knowledge Base, please contact the support team for assistance.

When you have finished working with the DDL, you can download the modified EDB compatible DDL as a text file and apply it to an existing Advanced Server instance.



Assessment Result

Object Name: **EMP**

Assessment Output

Success

Source DDL

```
CREATE TABLE HR.EMP
  (   EMPNO NUMBER(4,0),
   ENAME VARCHAR2(10) NOT NULL DISABLE,
   JOB VARCHAR2(9),
   MGR NUMBER(4,0),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2,0),
    CHECK (sal > 0) ENABLE,
    PRIMARY KEY (EMPNO) ENABLE
  ) ;
```

Target DDL

```
-- Migration Portal applied following repair actions
-- ERH-2009 deleted [ENABLE]

CREATE TABLE HR.EMP
  (   EMPNO NUMBER(4,0),
   ENAME VARCHAR2(10) NOT NULL,
   JOB VARCHAR2(9),
   MGR NUMBER(4,0),
   HIREDATE DATE,
   SAL NUMBER(7,2),
   COMM NUMBER(7,2),
   DEPTNO NUMBER(2,0),
    CHECK (sal > 0) ,
    PRIMARY KEY (EMPNO) )  ;

-- TABLE EMP is compatible with EDB Postgres Advanced Server
```

Figure 3.8 – Assessment result after resolving the errors.

3.3 Schema Migration

After resolving errors in your schemas, you can use the schemas with a client application such as pgAdmin, ToadEdge, or PSQL client. Or you can migrate the schema to EDB CDS cluster.

Please Note: Advanced Server is also supported by Toad Edge, for more information, see Toad Edge® for Postgres.

The Advanced Server instance must be installed in Redwood Mode to enable native compatibility with key Oracle capabilities.

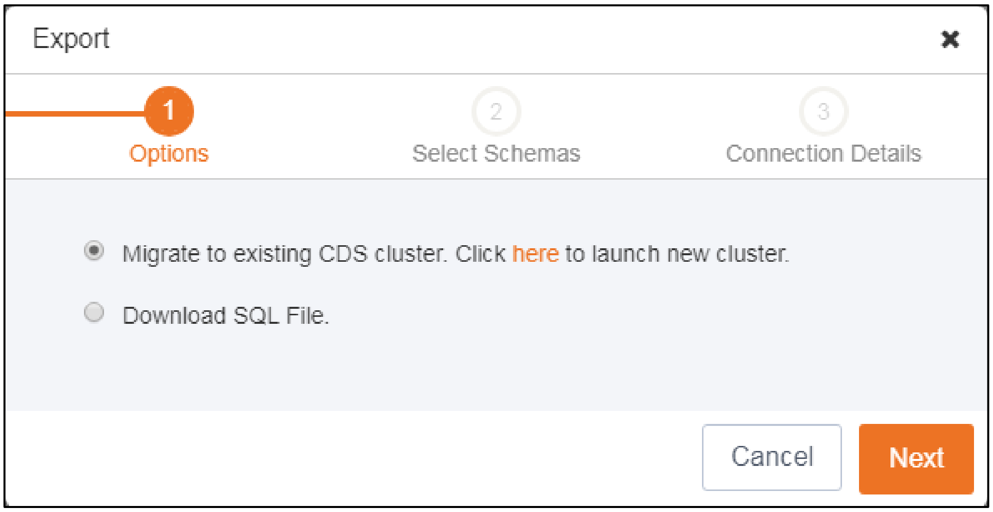
3.3.1 Migrating schema to CDS cluster

To migrate database to CDS cluster, complete the following steps:

1. On the Assessment page, click `Export/Deploy`.

2. Select `Migrate to existing CDS cluster` option.

   To create a new cluster, see Creating a Server Cluster.



Export

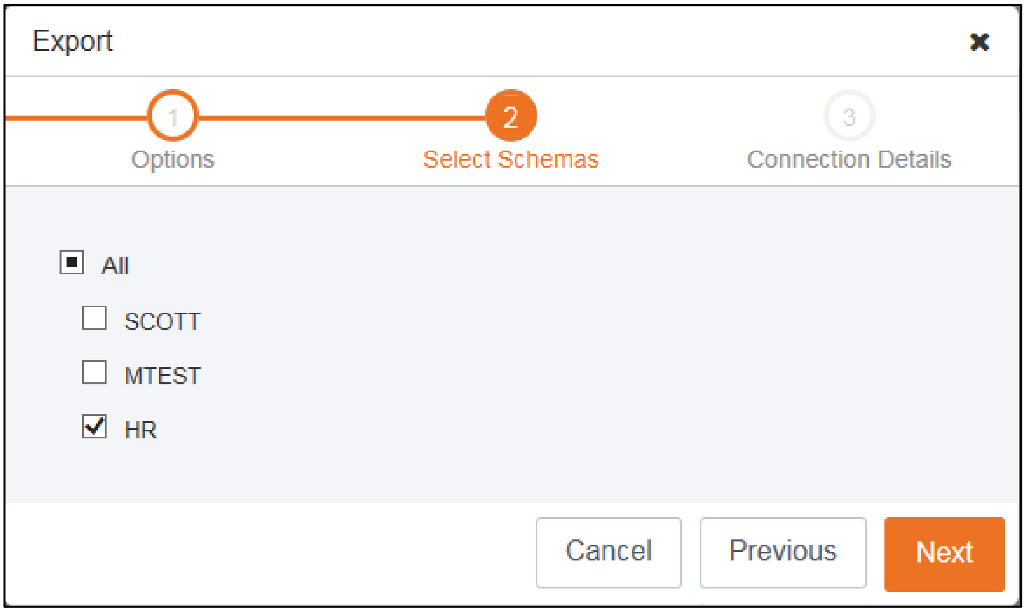
● Migrate to existing CDS cluster. Click here to launch new cluster.

○ Download SQL File.

Figure 4.1 Migrating database to CDS cluster

3. Click `Next`.

4. Select the required schemas.



■ All

□ SCOTT

□ MTEST

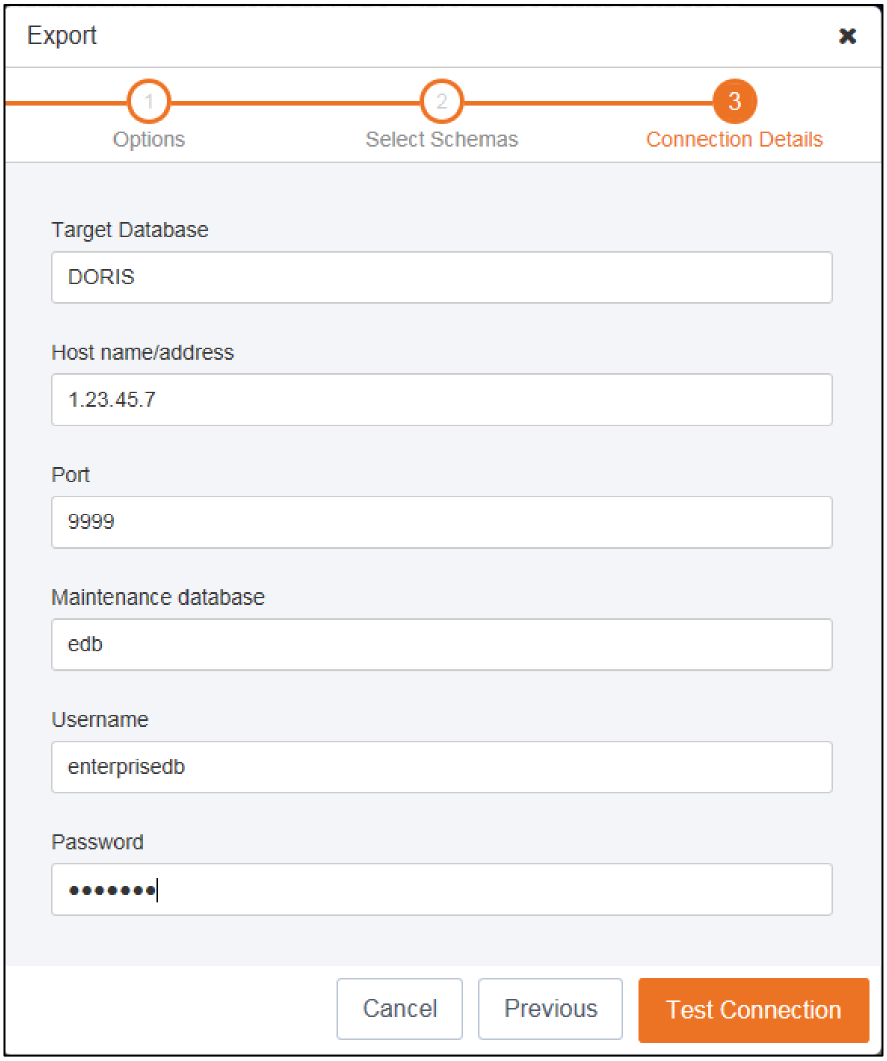
☑ HR

Figure 4.2: Select schemas

5. Click `Next`.

6. Enter the following details in the Connection Details:

- Enter the Host name or IP address in the *`Host name/address`* field.
- Enter port number in the *`Port`* field.
- Enter the database name in the *`Maintenance database`* field.
- Enter username in the *`Username`* field.
- Enter password in the *`Password`* field.



Target Database: DORIS

Host name/address: 1.23.45.7

Port: 9999

Maintenance database: edb

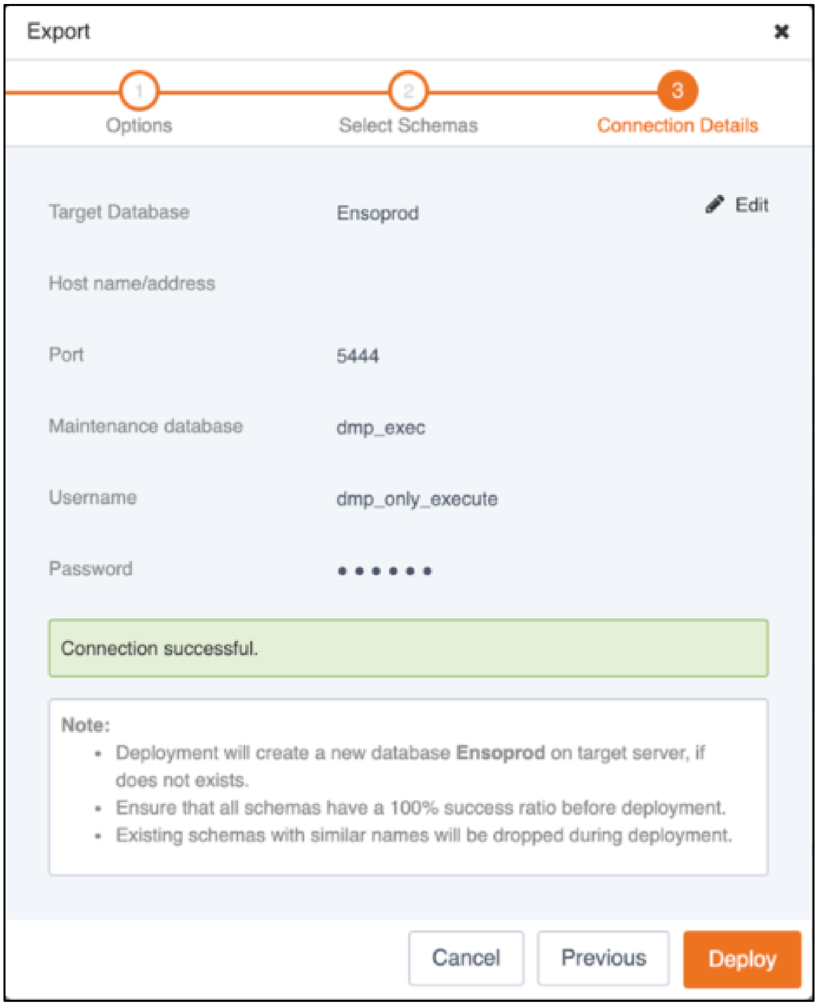
Username: enterprisedb

Password: •••••••

Figure 4.3 Connection details

7. Click `Test Connection`, to verify the connection details.
   Note: You can click `Edit` to make changes in the connection details and retest the connection details.

8. Once the connection is successful, click `Deploy`.



Target Database	Ensoprod
Host name/address	
Port	5444
Maintenance database	dmp_exec
Username	dmp_only_execute
Password	••••••

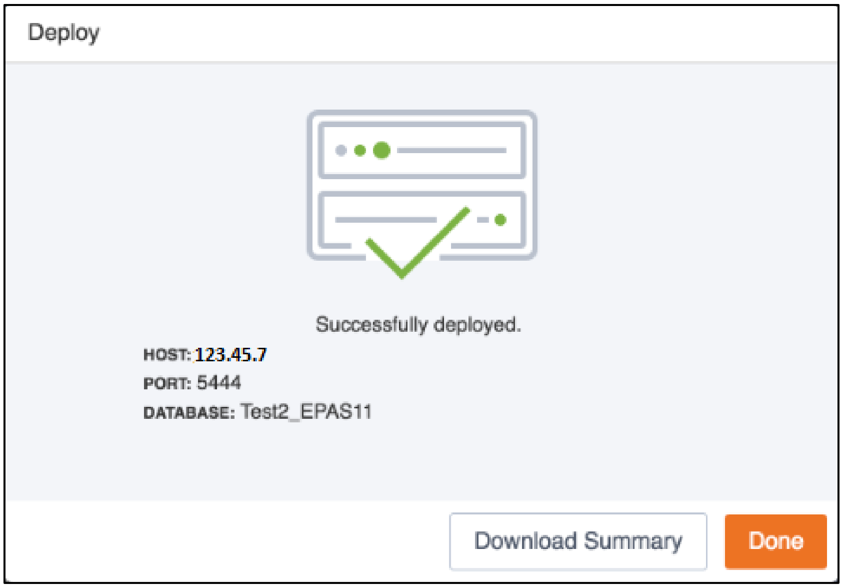
Connection successful.

Note:

- Deployment will create a new database **Ensoprod** on target server, if does not exists.
- Ensure that all schemas have a 100% success ratio before deployment.
- Existing schemas with similar names will be dropped during deployment.

Figure 4.4: Connection successful

9. On the deployment window, you can view the deployment details. You can click `Download Summary` to download the log.



Successfully deployed.

HOST: 123.45.7

PORT: 5444

DATABASE: Test2_EPAS11

Figure 4.5: Deployment details

9. Click `Done`, to close the window

3.4 Data Migration

After performing the schema migration, complete the following steps to migrate data:

1. Use EDB Migration Toolkit to migrate the table data. For detailed information about using Migration Toolkit, see EDB Postgres Migration Guide.

2. Configure the `toolkit.properties` file; ensure that connection information for the source and target databases is available in the property file:

   ```
   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
   ```

   For more information, see Building the toolkit.properties File.

10. Invoke Migration Toolkit in `–dataOnly` mode; include the `–truncLoad` keyword to resolve foreign key dependencies across tables.

    For example, the following command:

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

    The command migrates the specified source_schema to the target_schema. The data is loaded into the locally installed EDB Postgres instance with a database superuser named `enterprisedb` and the password as `password`.

    Please Note: The tables are truncated before attempting the data load.

4 Advanced Data Migration

For larger databases that require a parallel data load, you can use one of the following methods:

- Use the EDB Postgres Advanced Server database link feature (for compatibility with Oracle databases).

Or

- Dblink or a database link style migration (if your data contains CLOB data).

For more information, see the EDB Postgres Migration Guide.