



Migration Portal

Release 2.7

User's Guide

May 17, 2020

Contents

1	What's New	2
2	Supported Versions	7
3	Using the EDB Migration Portal	8
3.1	Overview of the Migration Portal Home Page	10
3.2	Overview of the Migration Portal Projects Page	11
3.3	Overview of the Migration Portal Wiki Page	13
4	Migrating a Database	14
4.1	Performing a Schema Extraction	15
4.1.1	Supported Object Types	18
4.1.2	Unsupported Object Types	18
4.1.3	Oracle System Schemas	19
4.2	Performing a Schema Assessment	20
4.2.1	Generating an Assessment Report	25
4.3	Schema Migration	28
4.3.1	Migrating a Schema to a CDS Cluster	29
4.4	Data Migration	32
5	Advanced Data Migration	33
6	Conclusion	34
	Index	35

EDB Postgres Migration Portal (Migration Portal) is a web-based service 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 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 of 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/>

CHAPTER 1

What's New

The following enhancements are added to the EDB Postgres Migration Portal for this release:

- You can view the reason an object failed to migrate for the selected schema in the `Common Failures` tab on the `Workspace` page:

The screenshot shows the EDB Migration Portal interface. On the left, the 'COMMON FAILURES' tab is selected, displaying a list of failures for the 'HR' schema. The main panel shows the 'HR Schema' with a table of failures.

Count	Reason for failure	Occurrences
2	only superuser can change options of a file_fdw foreign table	Tables 2
1	column "empno" does not exist	Views 1
1	syntax error at or near "loc"	Views 1
1	relation "hr.emp" does not exist	Materialized View 1

- You can download a CSV file for the common failures for the project:

The screenshot shows the EDB Migration Portal interface, similar to the previous one, but with a red box highlighting the 'CSV' download link in the top right corner of the 'COMMON FAILURES' tab.

- A warning message is displayed if a project or a schema is less than 70% compatible or if any DDL doesn't succeed after multiple attempts. An online link provides quick access to EDB experts at <mailto:migration-services@enterprisedb.com> for migration assistance.

The screenshot displays the EDB Migration Portal interface. On the left, a sidebar shows a list of projects (284) with search and 'New' buttons. The main content area shows the 'Overview' for a project named 'case_no_v2'. It indicates the source DB is ORACLE 11G and the target DB is ENTERPRISEDB 10. A red box highlights the 'Compatibility 68%' status with a warning icon. Below the overview, the 'Schemas' section shows migration progress for 'HR' (58%), 'MIGRATIONTEST' (77%), and 'schema_1900' (100%). Each schema has a bar chart showing failed, repaired, and passed objects. A warning message is displayed: 'Seems your schema has migration challenges. Contact us at migration-services@enterprisedb.com for help.'

- The [FAQ page](#) is updated regularly with the latest frequently asked questions.
- Changes were made to the new UI for better user experience.

Repair Handlers

New Repair Handler

The following repair handler is added to improve the Advance Server compatibility ratio:

- ERH 1011 - Block Comment Terminator:

Adds block comment terminator `*/` for unterminated block comments inside Procedures, Functions, Packages, Package Body, Triggers, Views, and Type Body.

For example:

```
CREATE OR REPLACE FUNCTION example_terminated (
    input_str IN VARCHAR2
)
/*****
Function      : example_terminated
Author       : EnterpriseDB
Description   : Returns substring
*****/
Change
Date         : sysdate
Changes      : new changes
/*****
RETURN VARCHAR2
AS
    var1      VARCHAR2(50);
```

(continues on next page)

(continued from previous page)

```
BEGIN
  RETURN( SUBSTR(input_str) ) ;
END example_terminated;
```

would become;

```
CREATE OR REPLACE FUNCTION example_terminated (
input_str  IN VARCHAR2
)
/*****
  Function      : example_terminated
  Author       : EnterpriseDB
  Description   : Returns substring
*/
/*****
  Change
  Date          : sysdate
  Changes       : new changes
*/
/*****/
RETURN VARCHAR2
AS
  var1          VARCHAR2(50);
BEGIN
  RETURN( SUBSTR(input_str) ) ;
END example_terminated;
```

Updated Repair Handler

The following repair handler is updated:

- ERH 2055 - Unsupported using Index Clause

Removes USING INDEX LOCAL/GLOBAL clause from the partitioned table DDL statement.

For example: Example 1 :

```
CREATE TABLE tab(
  ID INT,
  OPEN_STATUS INT,
  CLOSED_DATE DATE,
  CONSTRAINT tab_pk PRIMARY KEY (ID, OPEN_STATUS,
  CLOSED_DATE) USING INDEX LOCAL (PARTITION CLOSED, PARTITION
  OPEN ) ENABLE
) PARTITION BY RANGE (OPEN_STATUS,CLOSED_DATE)
(PARTITION CLOSED VALUES LESS THAN (0, MAXVALUE),
PARTITION OPEN VALUES LESS THAN (1, MAXVALUE) );
```

would become;

```
CREATE TABLE tab(
  ID INT,
  OPEN_STATUS INT,
  CLOSED_DATE DATE,
  CONSTRAINT tab_pk PRIMARY KEY (ID, OPEN_STATUS, CLOSED_DATE)
) PARTITION BY RANGE (OPEN_STATUS,CLOSED_DATE)
(PARTITION CLOSED VALUES LESS THAN (0, MAXVALUE),
PARTITION OPEN VALUES LESS THAN (1, MAXVALUE) );
```

Example 2:

```
CREATE TABLE tab2(
  ID NUMBER(19,0) NOT NULL ENABLE,
  P_DATE DATE,
  CONSTRAINT tab2_pk PRIMARY KEY (ID)
  USING INDEX GLOBAL PARTITION BY HASH (ID)
  (PARTITION SYS_P255 ,
  PARTITION SYS_P256 )
)
PARTITION BY RANGE (P_DATE) INTERVAL (NUMTODSINTERVAL(7,'DAY'))
(PARTITION P_FIRST VALUES LESS THAN (TO_DATE('2017-01-04
00:00:00', 'YYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')) ) ;
```

would become;

```
CREATE TABLE tab2(
  ID NUMBER(19,0) NOT NULL ENABLE,
  P_DATE DATE,
  CONSTRAINT tab2_pk PRIMARY KEY (ID)
)
PARTITION BY RANGE (P_DATE) INTERVAL (NUMTODSINTERVAL(7,'DAY'))
(PARTITION P_FIRST VALUES LESS THAN (TO_DATE('2017-01-04
00:00:00', 'YYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')) )
```

Knowledge Base**New Knowledge Base entries**

The following new Knowledge Base entries have been added; refer to the Knowledge Base section on the Migration Portal for workaround details.

- **Remainder function:**

In Oracle, the REMAINDER function behaves differently than the MOD function. However, Advanced Server does not support the REMAINDER function.

- **BFILENAME function:**

In Oracle, the BFILENAME function returns the BFILE locator object from a specified directory and filename. This function is used to access the data within the BFILE.

- **Median function:**

Oracle supports the MEDIAN function to calculate the median of the values of an expression. Advanced Server Version 12 supports the MEDIAN function. However, the earlier versions of Advanced Server do not support this function.

- **SOUNDEX function:**

Oracle supports the SOUNDEX function to return a string that contains the phonetic representation of a string. In Advanced server, the SOUNDEX function can be used after creating a fuzzystmatch extension.

- **DBMS_UTILITY package:**

Oracle uses the COMPILE_SCHEMA procedure in the DBMS_UTILITY package to compile all procedures, functions, packages, and triggers in the specified schema. However, the COMPILE_SCHEMA procedure does not exist in the EDB Advanced Server DBMS_UTILITY package. Hence while deploying the migrated schema on EDB Advanced Server, a runtime error occurs.

- **DBMS_DDL.WRAP and CREATE_WRAPPED functions:**

In Oracle, `DBMS_DDL.WRAP` or `DBMS_DDL.CREATE_WRAPPED` function is used to obfuscate code. However, in Advanced Server, `EDBWRAP` function is used for obfuscation. Before migrating schema, you must unwrap the PL/SQL object from Oracle and then wrap the PL/SQL object in Advanced Server using `EDBWRAP` utility.

- **Oracle TZ_OFFSET functions:**

In Oracle, the `TZ_OFFSET()` is used to get the time zone offset corresponding to the value you specify. However, in Advanced Server, you can write a function around Advanced Server catalog `pg_timezone_names` or you can directly query `pg_timezone_names` to get the time zone offset.

Updated Knowledge Base entry

The following knowledge base entry has been updated:

- **Merge statement:**

In Oracle, `MERGE` statement is used to conditionally insert or update a table. In Advanced Server, you can use `ON CONFLICT DO UPDATE` or `UPSERT` statement.

Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Advanced Server 10, 11, or 12. 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 also 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	OS X Sierra
Windows	10
Linux	CentOS 7

CHAPTER 3

Using the EDB Migration Portal

The Migration Portal allows you to easily migrate your database schema 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.

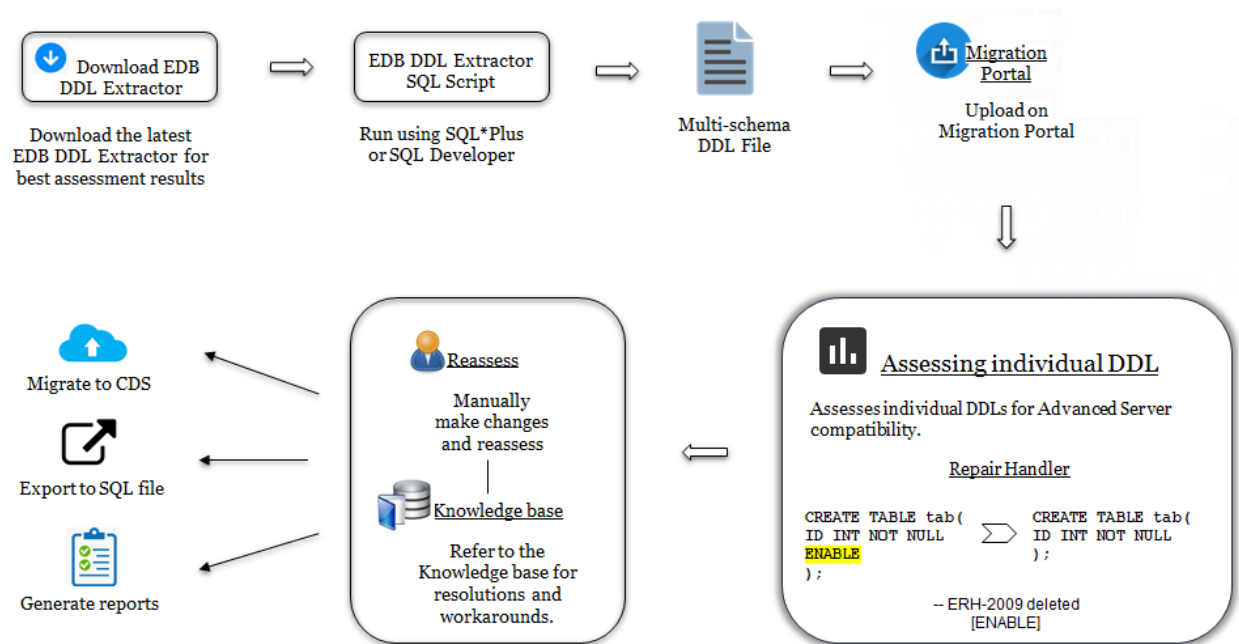


Fig. 1: The assessment and migration process

To access the migration portal:

1. Open a browser and navigate to <https://www.enterprisedb.com/>
2. On the EnterpriseDB home page, click Enterprise Postgres → EDB Postgres Migration Portal.

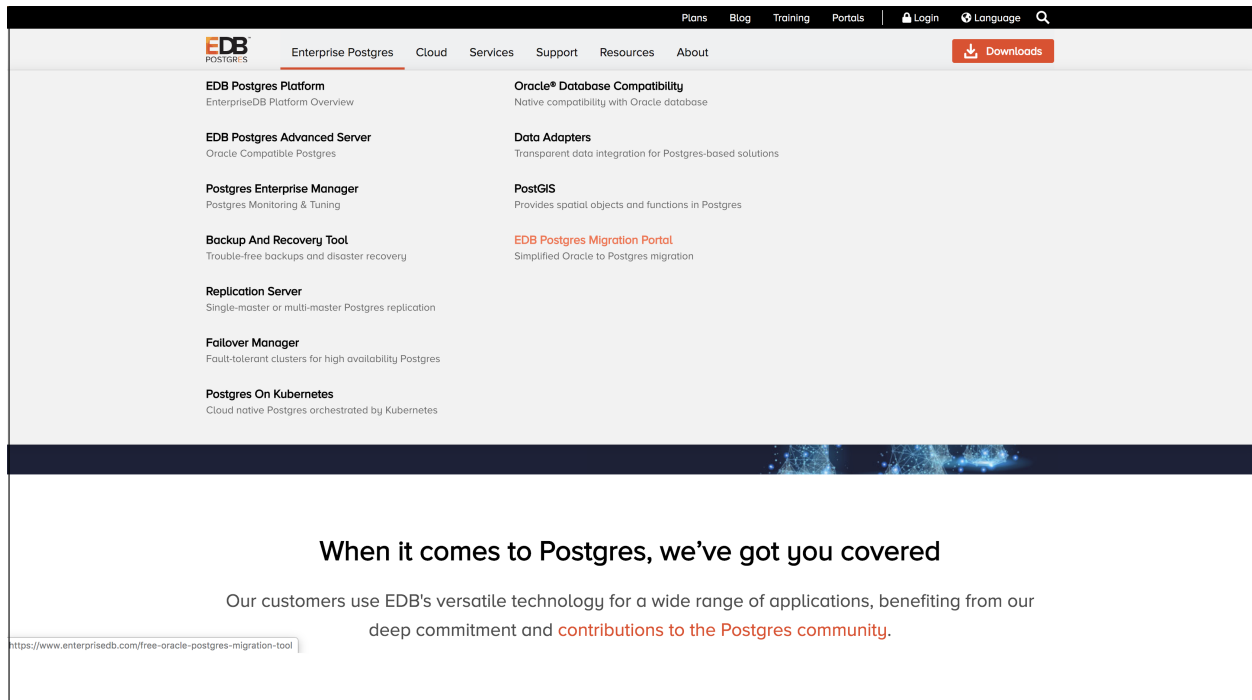


Fig. 2: Accessing the Migration Portal.

3.1 Overview of the Migration Portal Home Page

The Migration Portal home page provides quick access to migration tools.

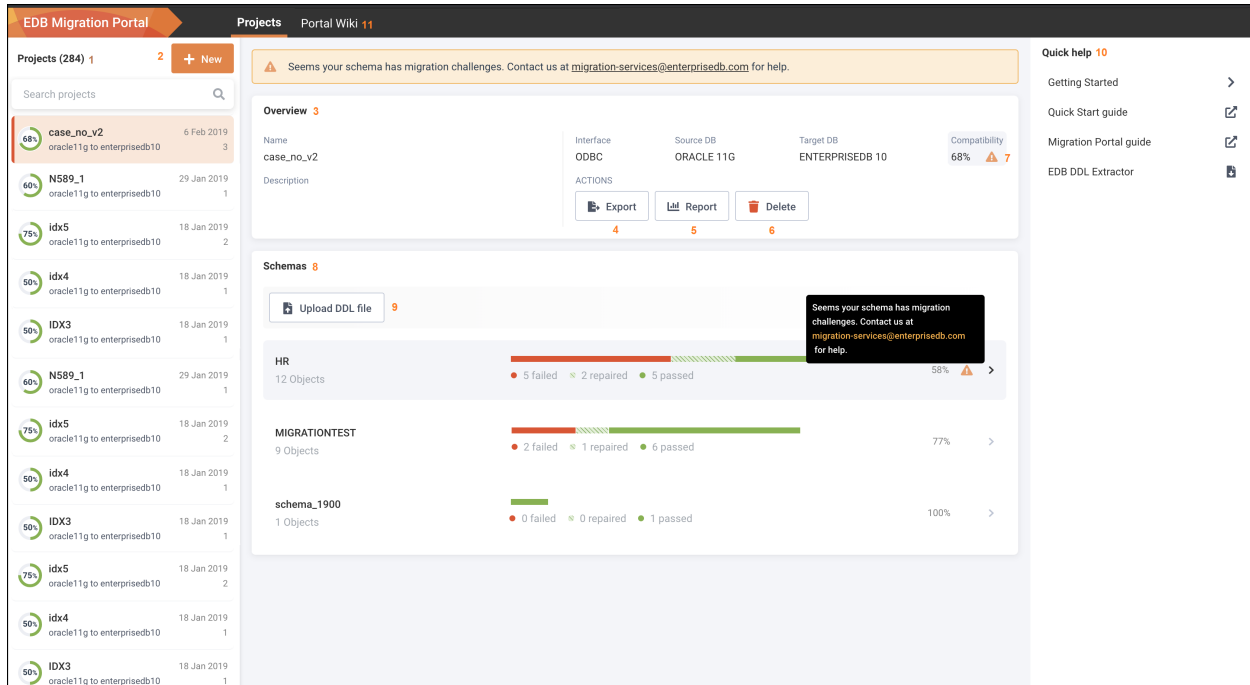


Fig. 3: The Migration Portal home page.

The Migration Portal home page allows access to the following Migration Portal features:

1. **Projects:** The `Projects` panel displays a list of assessed projects.
2. **Create project:** Click `+ New` (the button located to the right of the `Projects` label) to create a new project.
3. **Overview:** The `Overview` panel provides details about the selected project and displays the compatibility percentage after schema assessment.
4. **Export:** Use the `Export` button to either download an Advanced Server compatible `.sql` file or to migrate a schema to an EDB Cloud Database Services cluster.
5. **Report:** Use the `Report` button to view and download the schema assessment report.
6. **Delete:** Use the `Delete` button to delete a selected project.
7. **Warning sign:** A warning message is displayed if a project or a schema is less than 70% compatible or any DDL doesn't succeed after multiple attempts.
8. **Upload DDL file:** Use the `Upload new schema` button to upload a new or additional DDL file.
9. **Schemas:** The `Schemas` panel displays the assessment result from an uploaded DDL file.
10. **Quick help:** The `Quick help` panel contains all the help guides.
11. **Portal Wiki:** The `Portal Wiki` has links to product information and different help guides.

3.2 Overview of the Migration Portal Projects Page

The Migration Portal Projects page provides detailed information about your migration.

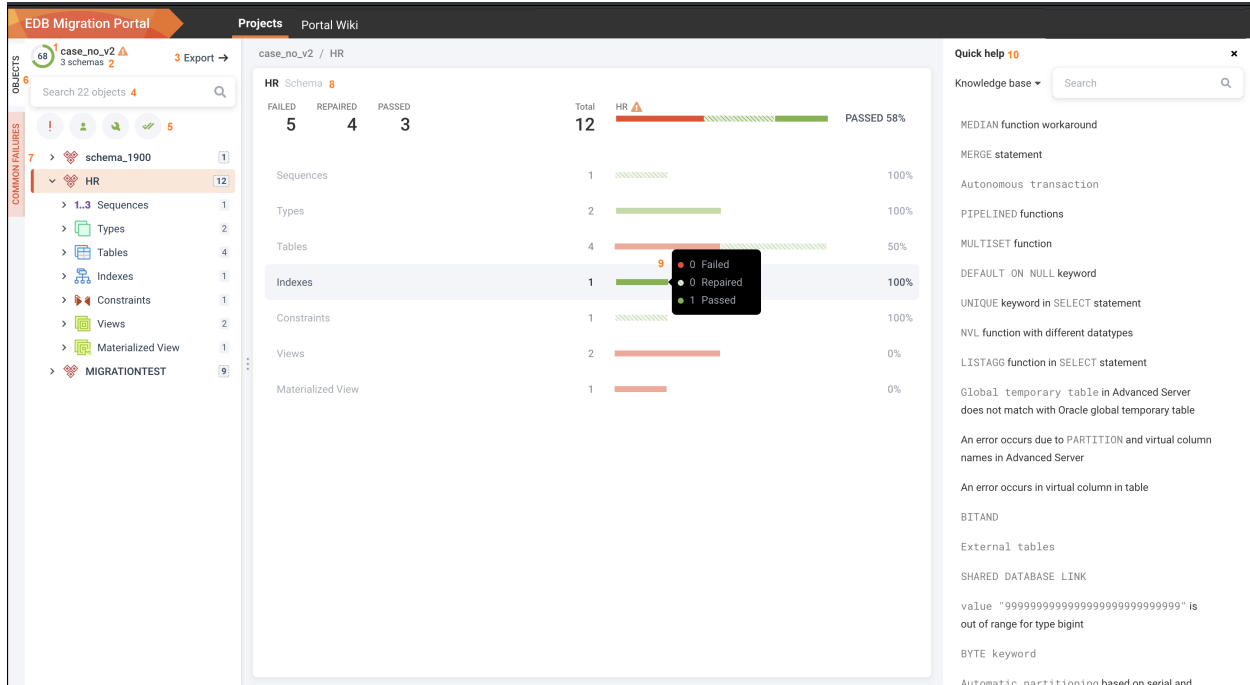


Fig. 4: The Migration Portal Projects Page overview.

Use the following resources to gather information about your migration projects:

1. **Compatible:** The Compatible gauge displays the compatibility percentage of the assessed schema.
2. **Schema Count:** Displays the number of schemas in a project.
3. **Export:** Use the Export icon to either download the .sql file or to migrate the schema to an *EDB Cloud Database Service* cluster.
4. **Search objects:** Use the Search box to search for objects.
5. **Filters:** You can filter the system repaired and manual repaired objects from the left panel of the Project page. In addition, you can select one or more filter combinations to refine the information.
6. **Objects:** Displays the objects for the selected schemas.
7. **Common Failures:** Displays the reason for the failed objects for the selected schemas.

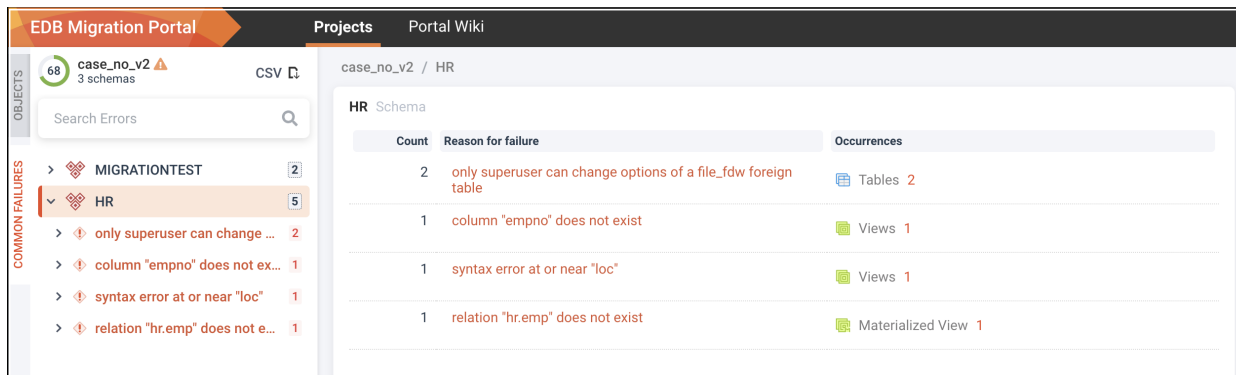


Fig. 5: Common Failures tab

Note: You can download a CSV file for the common failures for the project.

- Schema:** The Schema panel displays the result of the assessment.
- Tooltip:** Hover over a result set to display a tooltip with the number of passed, failed, and repaired objects.
- Quick help:** The Quick help panel displays links to Knowledge base articles and repair handler documentation.
- Search:** Use the Search box to search the Knowledge base entries or repair handler documentation for specific information.

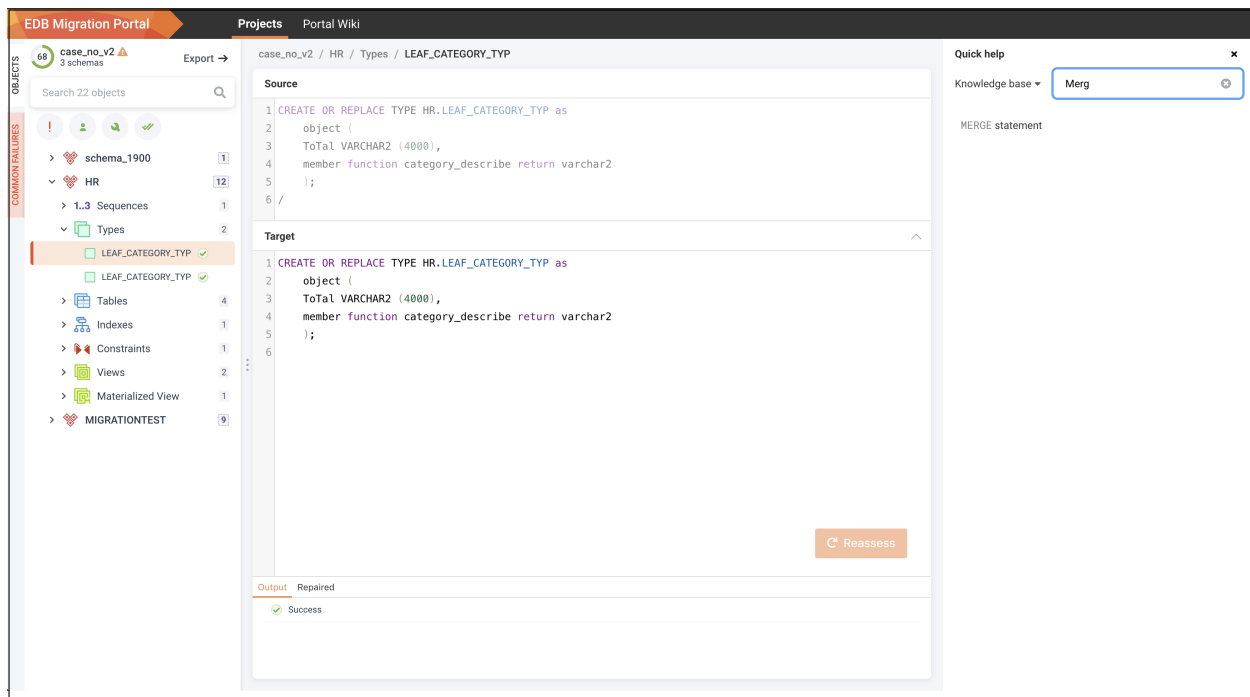


Fig. 6: Searching the Knowledge Base entry.

3.3 Overview of the Migration Portal Wiki Page

The Portal Wiki page provides quick access to information:

- What's New information
- Quick Start guide
- Migration Data guide
- DDL Extractor guide
- Knowledge Base
- Repair handlers
- Migration Portal User's Guide
- FAQs

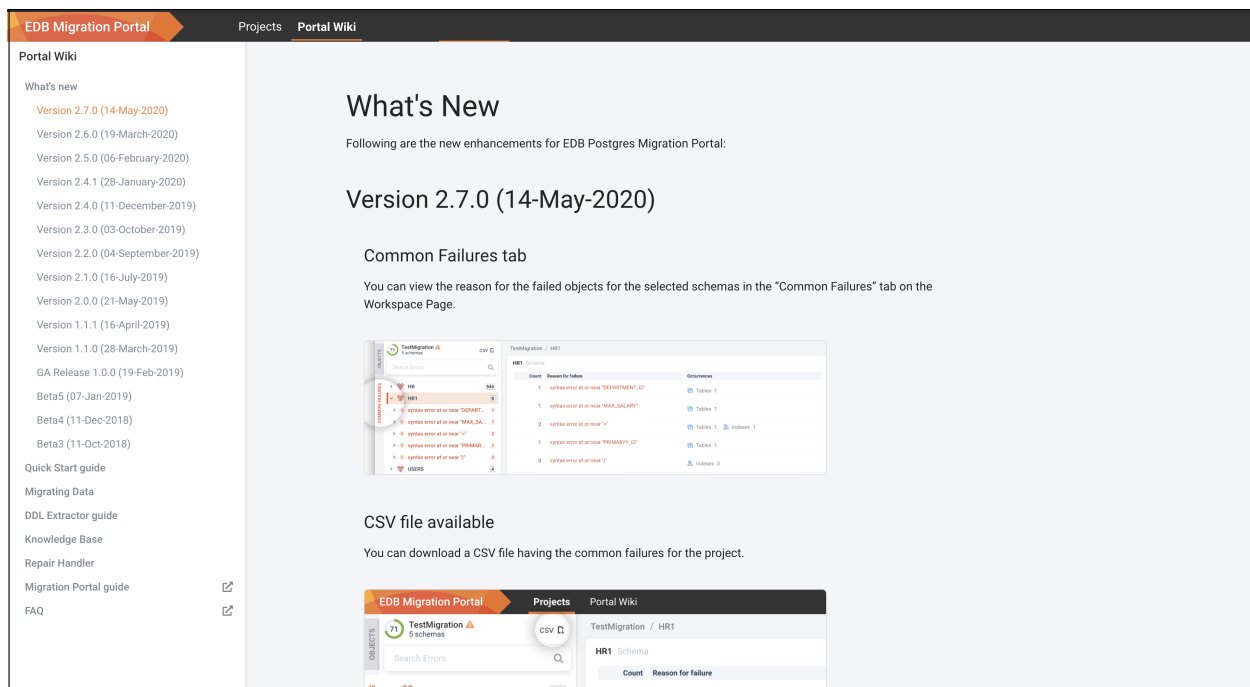


Fig. 7: The Migration Portal Wiki page.

Migrating a Database

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

1. Perform a *Schema Extraction*.
2. Perform a *Schema Assessment*.
3. Perform a *Schema Migration*.
4. Perform a *Data Migration*.

The following sections provide detailed information about each step in the migration process.

4.1 Performing a Schema Extraction

Prerequisites

Before extracting a schema, you must download the latest EDB DDL Extractor script from the Migration Portal [Projects](#) page or from the link provided in the DDL Extractor guide in the Portal Wiki. The script can be run in SQL Developer or SQL*Plus. It uses Oracle's DBMS_METADATA built-in package to extract DDLs for different objects under schemas (specified while running the script). The EDB DDL extractor creates the DDL file that will be uploaded to the portal and analyzed for EDB Postgres compatibility.

Note: You must have `SELECT CATALOG ROLE` and `SELECT ANY DICTIONARY` privileges in the Oracle database.

For SQL*Plus

1. Connect to SQL*Plus and run the command:

```
SQL>@edb_ddl_extractor.sql
```

2. Provide the schema name and the path/directory in which the extractor will store the extracted DDL. When extracting multiple schemas, use a comma (',') as a delimiter.

Note: If you want to extract all the user schemas from the current database, do not mention any schema names while extracting. However, it is recommended to mention the schema names that you would like to extract.

3. If you want to extract dependent objects from other schemas, enter `yes` or `no`.

For example, on Linux:

Enter SCHEMA NAME[S] to extract DDLs:

HR, SCOTT, FINANCE

Enter the PATH to store DDL file:

/home/oracle/extracted_ddls/

Extract dependent objects from other schemas? (yes/no): yes

On Windows:

Enter SCHEMA NAME[S] to extract DDLs:

HR, SCOTT, FINANCE

Enter the PATH to store DDL file:

C:\Users\Example\Desktop\

Extract dependent objects from other schemas? (yes/no): yes

For SQL Developer

1. Connect to the SQL server and run the following command:

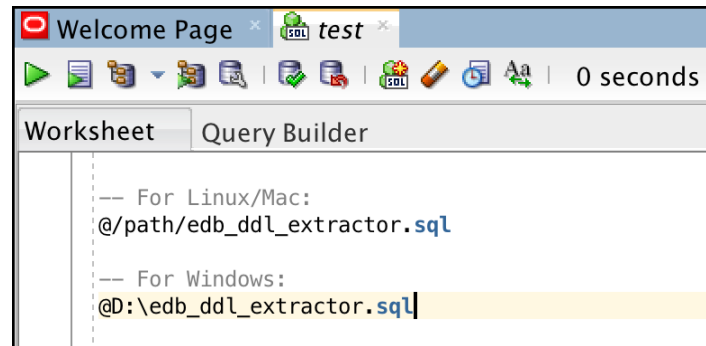


Fig. 1: Enter the path for Linux or Windows.

2. Enter a comma-separated list of schemas:

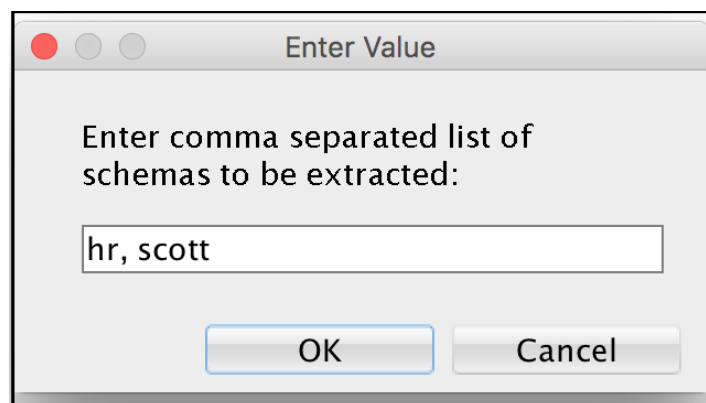


Fig. 2: Provide a list of schemas.

3. Enter file path for the output file:

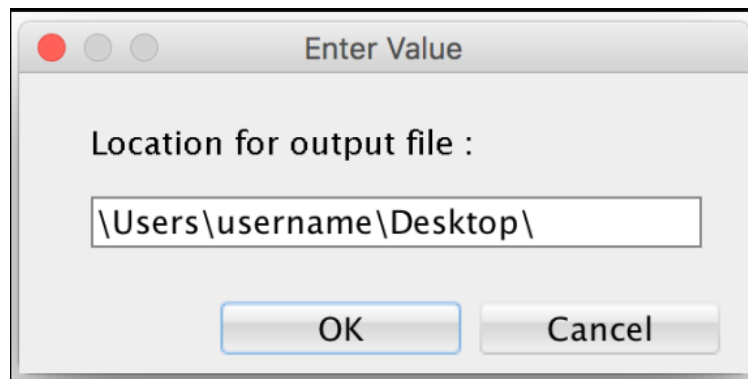


Fig. 3: Specify the output file path.

4. Extract dependent objects from other schemas?(yes/no): yes

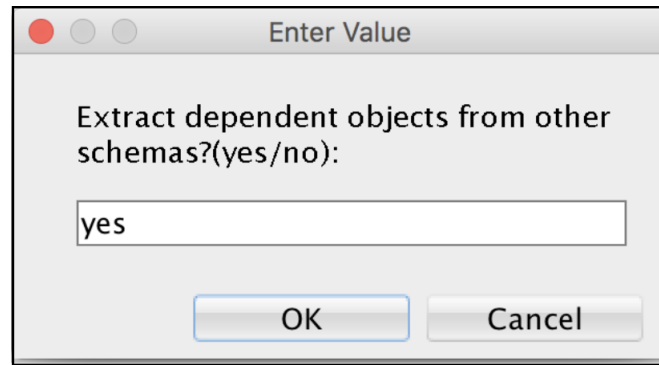


Fig. 4: *Extracting dependent objects.*

Note: You can also enter a single schema name in both the SQL*Plus and SQL Developer tools.

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

Additional Notes

- The EDB DDL Extractor does not extract objects that have names like:

```
BIN$b54+4XIEYwPgUAB/AQBWwA= =$0
```

To extract these objects, you must change the name of the objects and re-run the extraction process.

- DDL Extractor extracts `nologging` tables as normal tables. Once these tables are migrated to Advanced Server, WAL log files will be created.

4.1.1 Supported Object Types

The migration portal supports the 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

Note: COMMENTS on Columns, Tables, and Materialized Views are supported.

4.1.2 Unsupported 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

4.1.3 Oracle System Schemas

EDB DDL Extractor script will ignore the following system schemas while extracting from Oracle:

ANONYMOUS	APEX_PUBLIC_USER	APEX_030200
APEX_040000	APEX_040000	APPQOSSYS
AUDSYS	BI	CTXSYS
DMSYS	DBSNMP	DIP
DVF	DVSYS	EXFSYS
FLows_FILES	FLows_020100	GSMADMIN_INTERNAL
GSMCATUSER	GSMUSER	IX
LBACSYS	MDDATA	MDSYS
MGMT_VIEW	OE	OJMSYS
OLAPSYS	ORDPLUGINS	ORDSYS
ORDDATA	OUTLN	ORACLE_OCM
OWBSYS	OWBYSS_AUDIT	PM
RMAN	SH	SI_INFORMTN_SCHEMA
SPATIAL_CSW_ADMIN_USR	SPATIAL_WFS_ADMIN_USR	SYS
SYSBACKUP	SYSDG	SYSKM
SYSTEM SYSMAN	TSMSYS WKPROXY	WKSYS
WK_TEST	WMSYS	XDB
XS\$NULL		

4.2 Performing a Schema Assessment

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

1. Navigate to the [Migration Portal](#).
2. Enter your EDB credentials.
3. Click the `CREATE PROJECT` icon to create a new project.

Fig. 5: The Migration Portal New project dialog.

4. On the New project dialog, enter the project name in the Project name field.
5. Specify project details:
 - Select the radio button next to the Application interface.
 - Select a Source DB and Version for the source.
 - Select the Target DB and Version for the target.
6. Click Choose file to upload the .SQL file generated by the latest EDB DDL Extractor for Oracle Database.

Note:

- You should not modify the .SQL file.
- Only the .SQL file generated by the latest EDB DDL Extractor can be uploaded.

For more information, refer to the [Schema Extraction](#) section.

7. Check the box next to Add Index Prefix to specify an index prefix (idx) when creating a project to ensure better assessment results, as Advanced Server does not support the same name for tables and indexes.

8. Click **Create & assess** to create a new project and to assess the schemas.

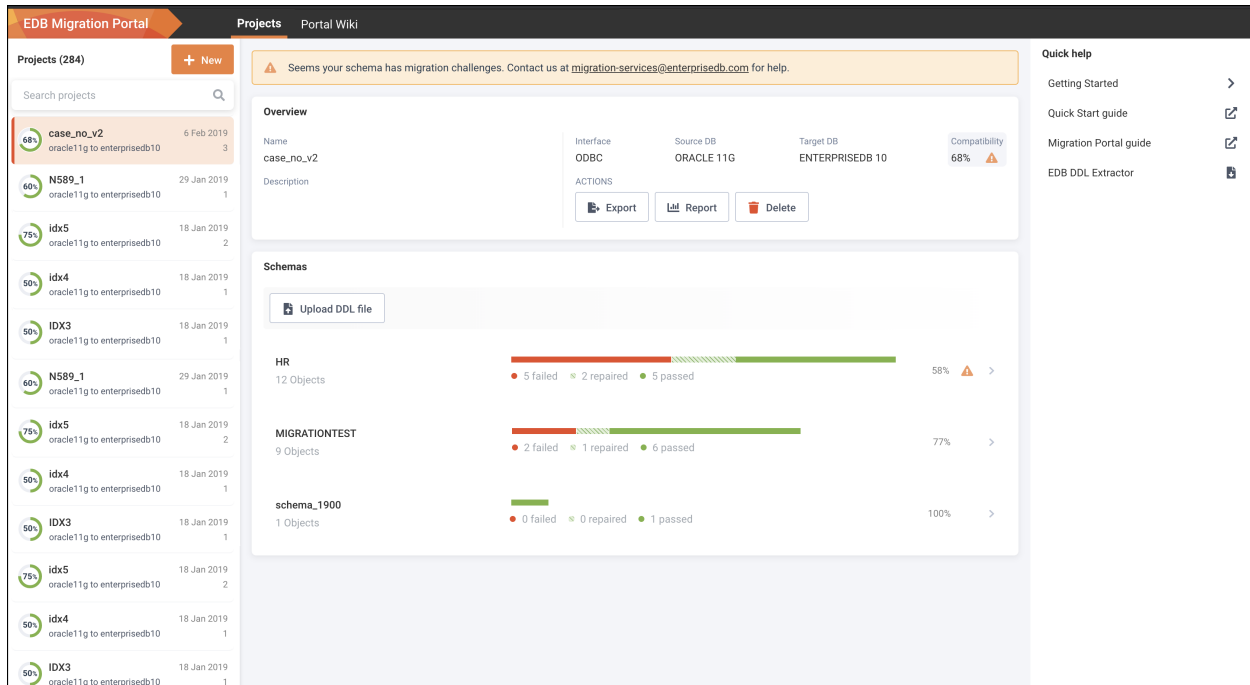


Fig. 6: The Schema analysis result.

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

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

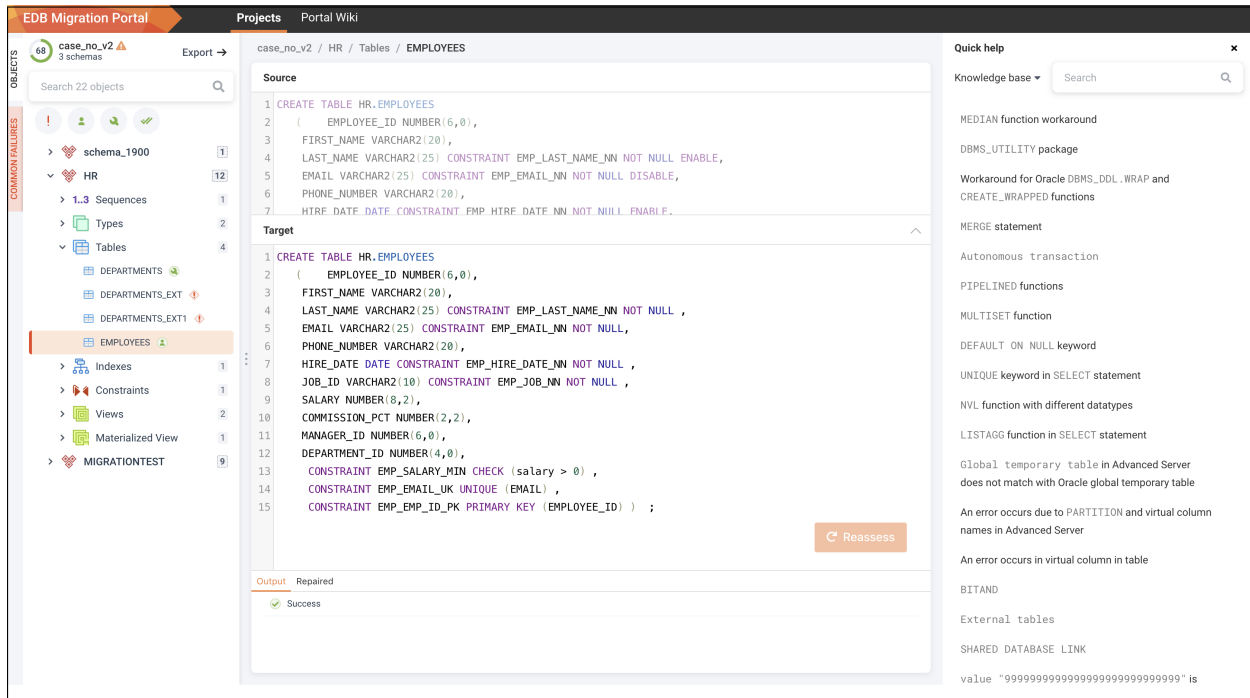


Fig. 10: *Workaround or resolution for incompatible objects.*

Similarly, you can make all the incompatible objects compatible.

Note: If the information or workaround for incompatible objects are not available in the Knowledge Base, please contact the support team for assistance.

When you have finished working with the DDL, you can either download the modified EDB compatible DDL as a .sql file or migrate the schemas *to a CDS cluster*.

4.2.1 Generating an Assessment Report

Migration Portal's report functionality provides a high-level assessment summary of the schemas assessed for your project. In addition, the report provides details about the failed objects and the cause of failure.

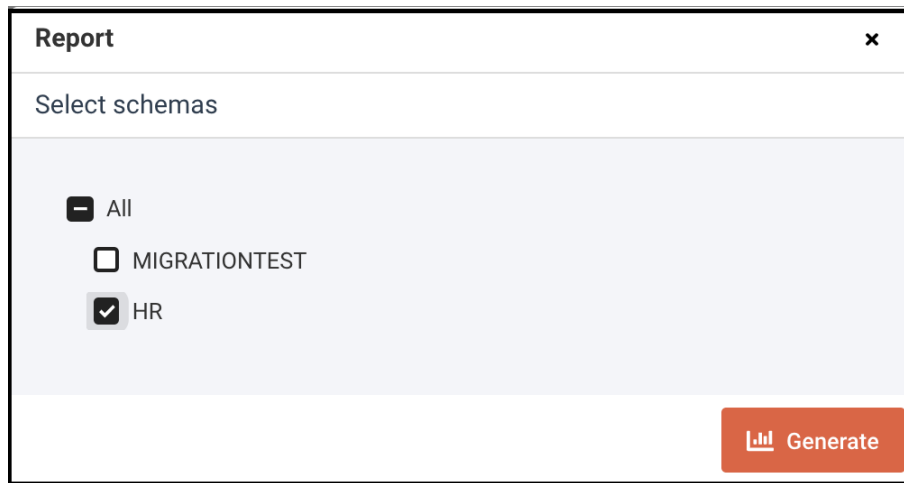


Fig. 11: *Select schemas for reports.*

To generate a report:

1. Click the `Report` button to access the schema selection dialog.
2. Select the schemas that you wish to include in the report.
3. Click `Generate` to generate the onscreen report.

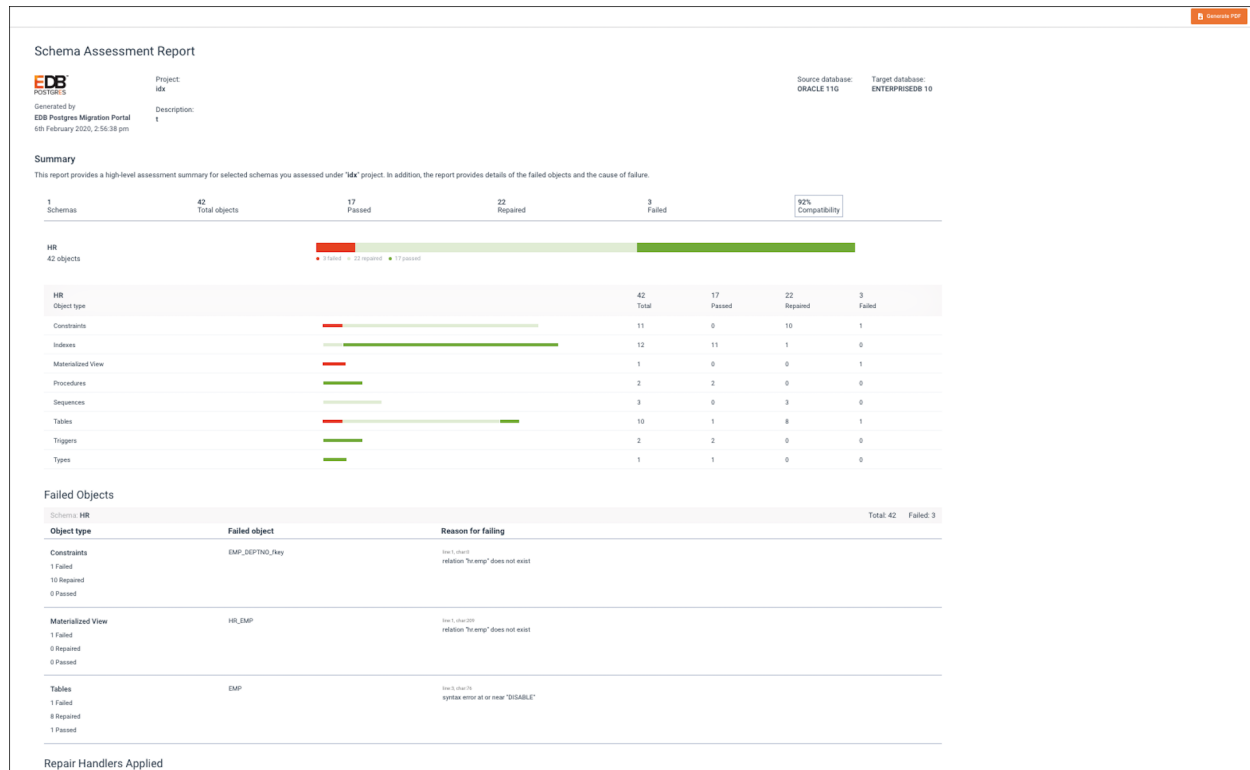


Fig. 12: The Schema Assessment Report.

You can optionally select **Generate PDF** to save the report in .pdf format. You can also view the count of distinct repair handlers applied to the DDLs under the selected schemas.

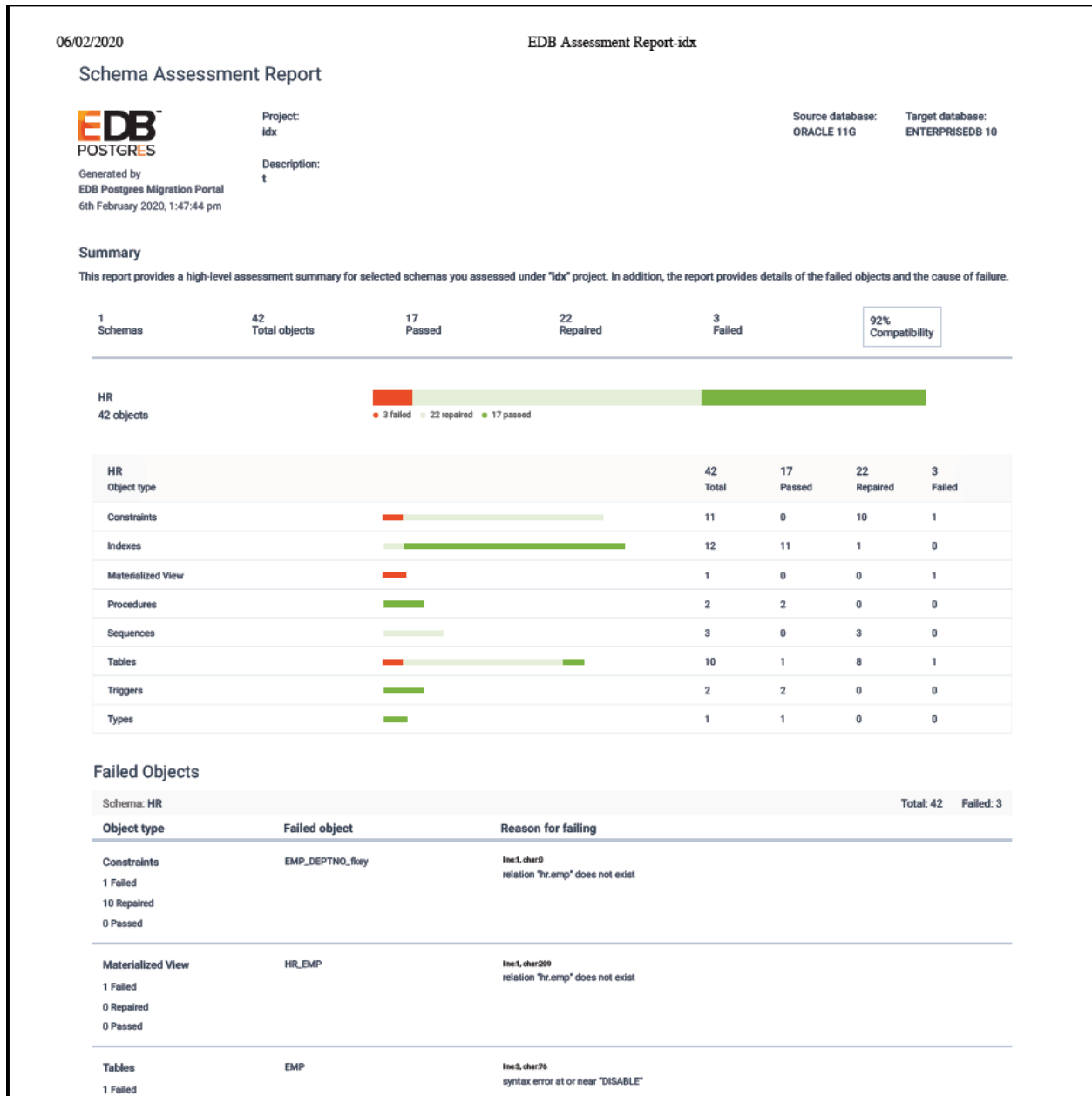


Fig. 13: The saved pdf report.

4.3 Schema Migration

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

Please Note: For more information about using Toad Edge with Advanced Server, see [Toad Edge® for Postgres](#).

Using PSQL or pgAdmin to Deploy a schema

Ensure that the assessed schema is 100% compatible on Migration Portal.

Perform the following steps to deploy a schema using PSQL or pgAdmin:

1. On the **Project Page**, select the required project.
2. Click **Export** to download the assessed file.

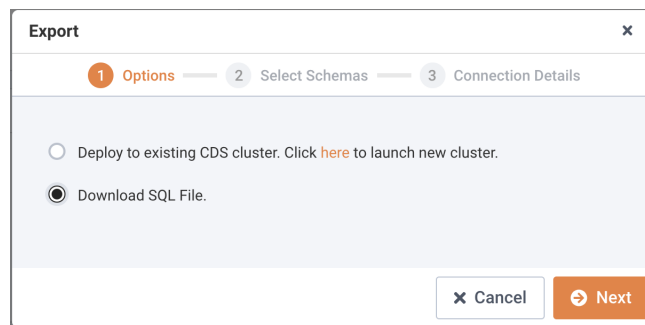


Fig. 14: Selecting download file option

3. Click **Download**.

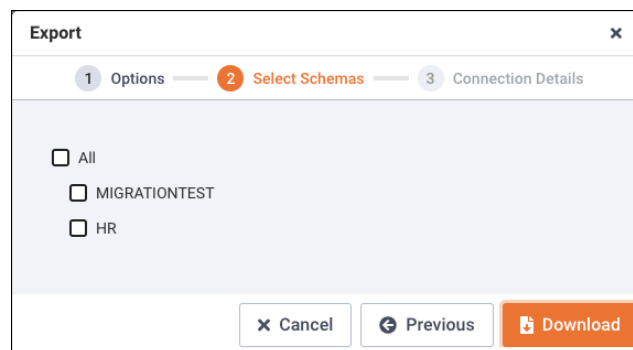


Fig. 15: Downloading assessed file

4. (For PSQL) Connect to desired Advanced Server using the psql/edb-psql client and run the following command:

```
edb=# \i /exported_ddls_folder/hr_schema/ProjHR_hr.sql
```
5. (For pgAdmin) Connect to the desired database and click **Open**.
6. Upload the schema and click **Run**.

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

4.3.1 Migrating a Schema to a CDS Cluster

Perform the following steps to migrate your database to a CDS cluster:

1. On the **Project** page, click **Export**.
2. Select the **Deploy to existing CDS cluster**. Click [here](#) to launch new cluster option.

For information about creating a new cluster, see [Creating a Server Cluster](#).

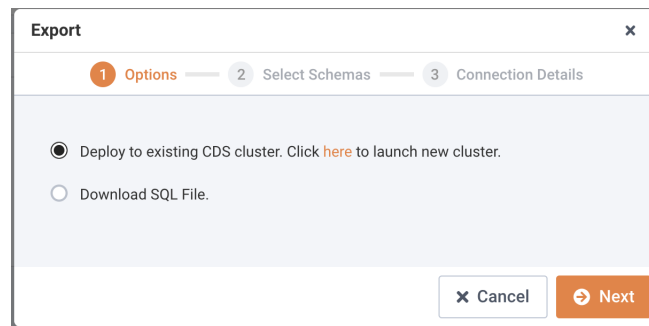


Fig. 16: Migrating database to CDS cluster.

3. Click **Next**.
4. Select the schemas you wish to migrate.

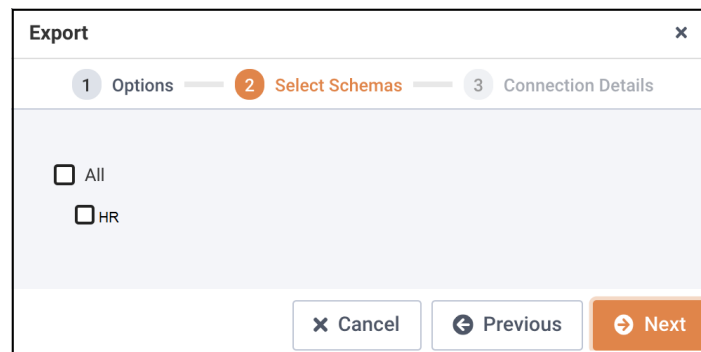
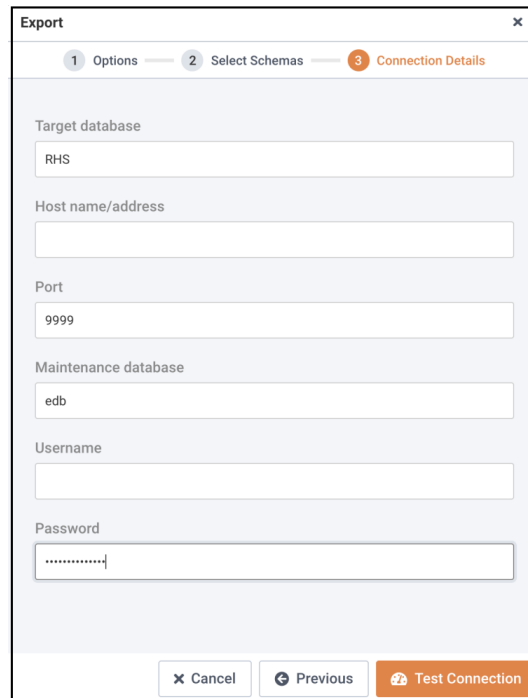


Fig. 17: Select schemas for migration.

5. Click **Next** to continue.



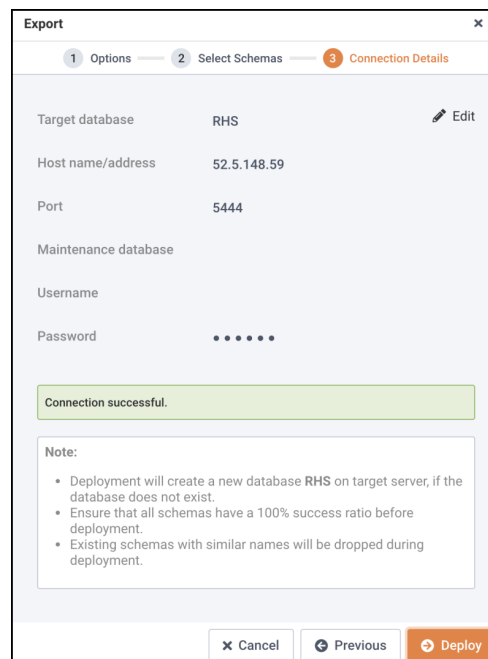
The image shows the 'Export' dialog box with the 'Connection Details' tab selected. The dialog has three tabs: 'Options', 'Select Schemas', and 'Connection Details'. The 'Connection Details' tab contains the following fields:

- Target database: RHS
- Host name/address: (empty)
- Port: 9999
- Maintenance database: edb
- Username: (empty)
- Password: (masked with dots)

At the bottom, there are three buttons: 'Cancel', 'Previous', and 'Test Connection'.

Fig. 18: Provide connection details.

6. Enter the required details in the Connection Details dialog box.



The image shows the 'Export' dialog box with the 'Connection Details' tab selected, after a successful connection test. The fields are now populated with the following values:

- Target database: RHS (with an 'Edit' icon)
- Host name/address: 52.5.148.59
- Port: 5444
- Maintenance database: (empty)
- Username: (empty)
- Password: (masked with dots)

A green message box displays 'Connection successful.' Below this, a 'Note' box contains the following text:

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

At the bottom, there are three buttons: 'Cancel', 'Previous', and 'Deploy'.

Fig. 19: Testing a successful connection.

7. Click Test Connection to verify the connection details.

Note: You can click Edit to make changes to the connection details and retest the connection details.

8. Once the connection is successful, click `Deploy`.
9. You can view the deployment details on the `Deploy` dialog; click `Download Summary` to download the deployment log.

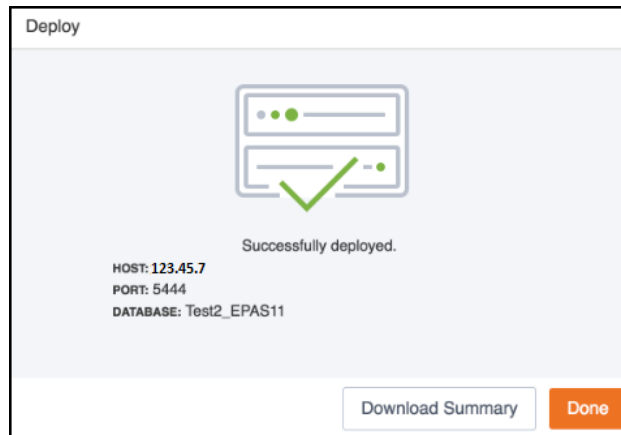


Fig. 20: A successful deployment.

10. Click `Done` to close the window.

4.4 Data Migration

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

1. Use EDB Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see the [EDB Postgres Migration Toolkit Guide](#).
2. Configure the Migration Toolkit `toolkit.properties` file, ensuring 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](#).

3. 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 of `password`.

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

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

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

For more information, see the [EDB Postgres Migration Toolkit Guide](#).

CHAPTER 6

Conclusion

EDB Postgres Migration Portal Copyright © 2019 - 2020 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.
- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

A

Advanced Data Migration, [33](#)

C

Conclusion, [34](#)

D

Data Migration, [32](#)

K

knowledge base, [12](#), [13](#)

M

Migrating a Database, [14](#)

Migration Portal, [1](#)

Migration Toolkit, [32](#)

O

Overview of the Migration Portal Home
Page, [10](#)

Overview of the Migration Portal
Projects Page, [11](#)

Overview of the Migration Portal Wiki
Page, [13](#)

P

Performing a Schema Assessment, [20](#)

Performing a Schema Extraction, [15](#)

pgAdmin, [28](#)

psql, [28](#)

R

repair handlers, [13](#)

resources, [13](#)

S

schema assessment report, [24](#), [25](#)

schema connection details, [30](#)

Schema Migration, [28](#)

supported browsers, [7](#)

supported databases, [7](#)

supported object types, [17](#), [18](#)

supported operating systems, [7](#)

Supported Versions, [7](#)

T

test schema connection, [30](#)

Toad Edge, [28](#)

U

Using the EDB Migration Portal, [8](#)

W

What's New, [2](#)