



# Migration Portal

## Version 2.9

1	What's New	3
2	Supported Versions	5
3	Using Migration Portal	6
3.1	Overview of the Migration Portal Home Page	7
3.2	Overview of the Migration Portal Projects Page	8
3.3	Overview of the Migration Portal Wiki Page	11
4	Migrating a Database	11
4.1	Performing a Schema Extraction	12
4.2	Performing a Schema Assessment	16
4.3	Schema Migration	24
4.4	Data Migration	50
5	Advanced Data Migration	51

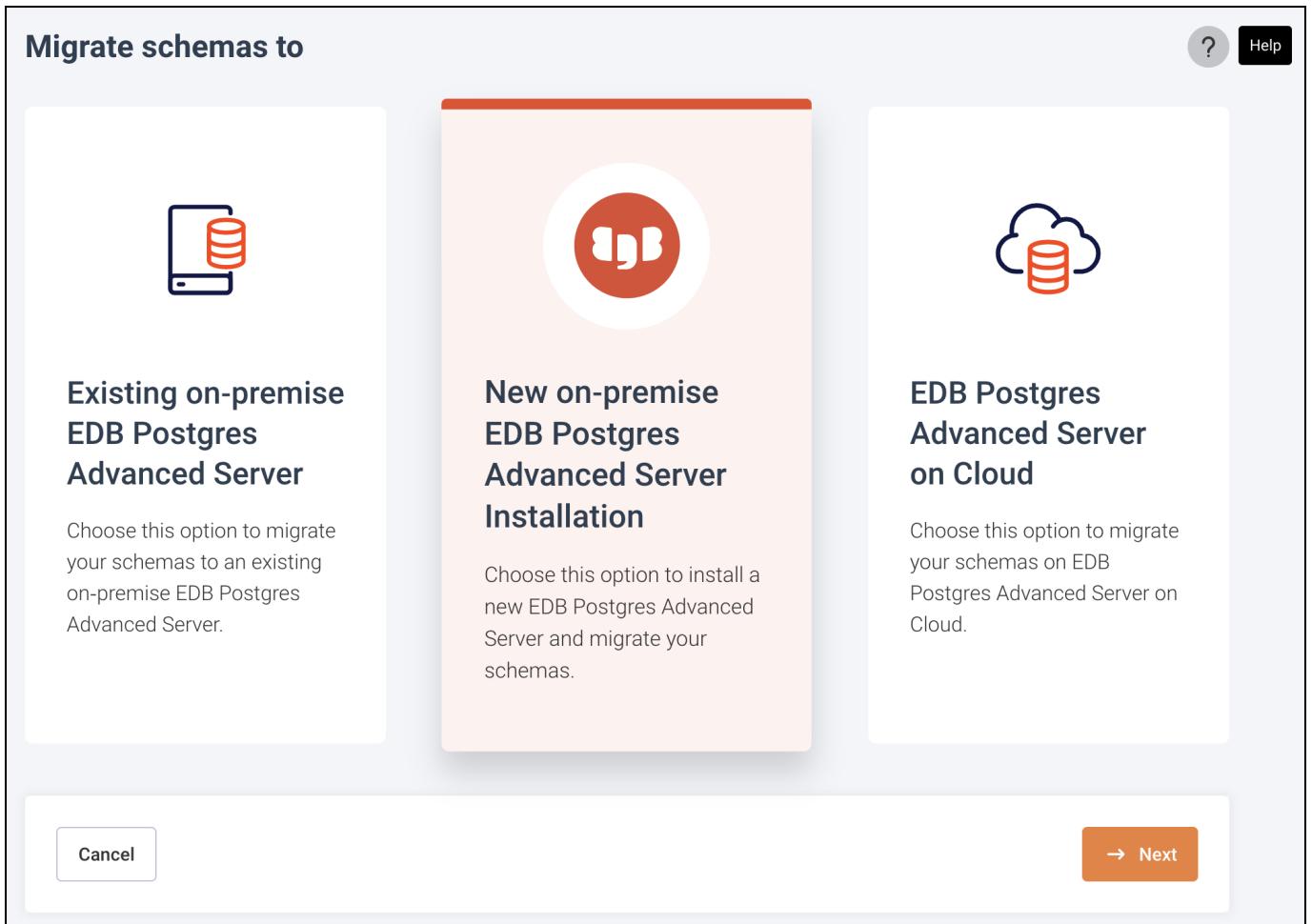
# 1 What's New

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

- Starting this release, you can install EDB Postgres Advanced Server or use EDB Postgres Advanced Server on the cloud directly from the Migration Portal.

Following are the options for migrating schemas:

- Existing on-premise EDB Postgres Advanced Server
- New on-premise EDB Postgres Advanced Server
- EDB Postgres Advanced Server on Cloud



For more information, see [Schema Migration <mp\\_schema\\_migration>](#).

- You can now fill up a [Contact Us](#) form for any help required for migration, installation, or other related queries.

## Contact Us

Your message...

Required project details will be sent along with your message.

By submitting this form, I agree to EDB's [Terms & Conditions](#) and [Privacy Policy](#).

 [Send](#)

- Many changes are made to the new UI for better user experience.
- Report generating duration for larger projects is optimized drastically. For example, previously, the time required to generate a report was 5 to 7 minutes. However, now it takes less than 5 seconds to generate a report.

## New Repair Handler

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

- ERH 2090 - Convert Function:

Replaces Oracle's **CONVERT** function with EDB Postgres Advanced Server compatible **convert** or **convert\_to** function inside any PL/SQL block.

For example:

```
CREATE VIEW MY_VIEW1 AS SELECT CONVERT('Ã‰ÍÑØABCD', 'AL32UTF8', 'WE8ISO8859P1') as CONVERT_TEST FROM dual;
CREATE VIEW MY_VIEW2 AS SELECT CONVERT('Ã‰ÍÑØABCD', 'AL32UTF8') as
```

```
CONVERT_TEST FROM dual;
```

would become:

```
CREATE VIEW MY_VIEW1 AS SELECT convert('Ã Ä Í Õ Ø A B C D E','LATIN1','UTF8')
as CONVERT_TEST FROM dual;
CREATE VIEW MY_VIEW2 AS SELECT convert_to('Ã Ä Í Õ Ø A B C D E','UTF8')
as CONVERT_TEST FROM dual;
```

## 2 Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Postgres 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:

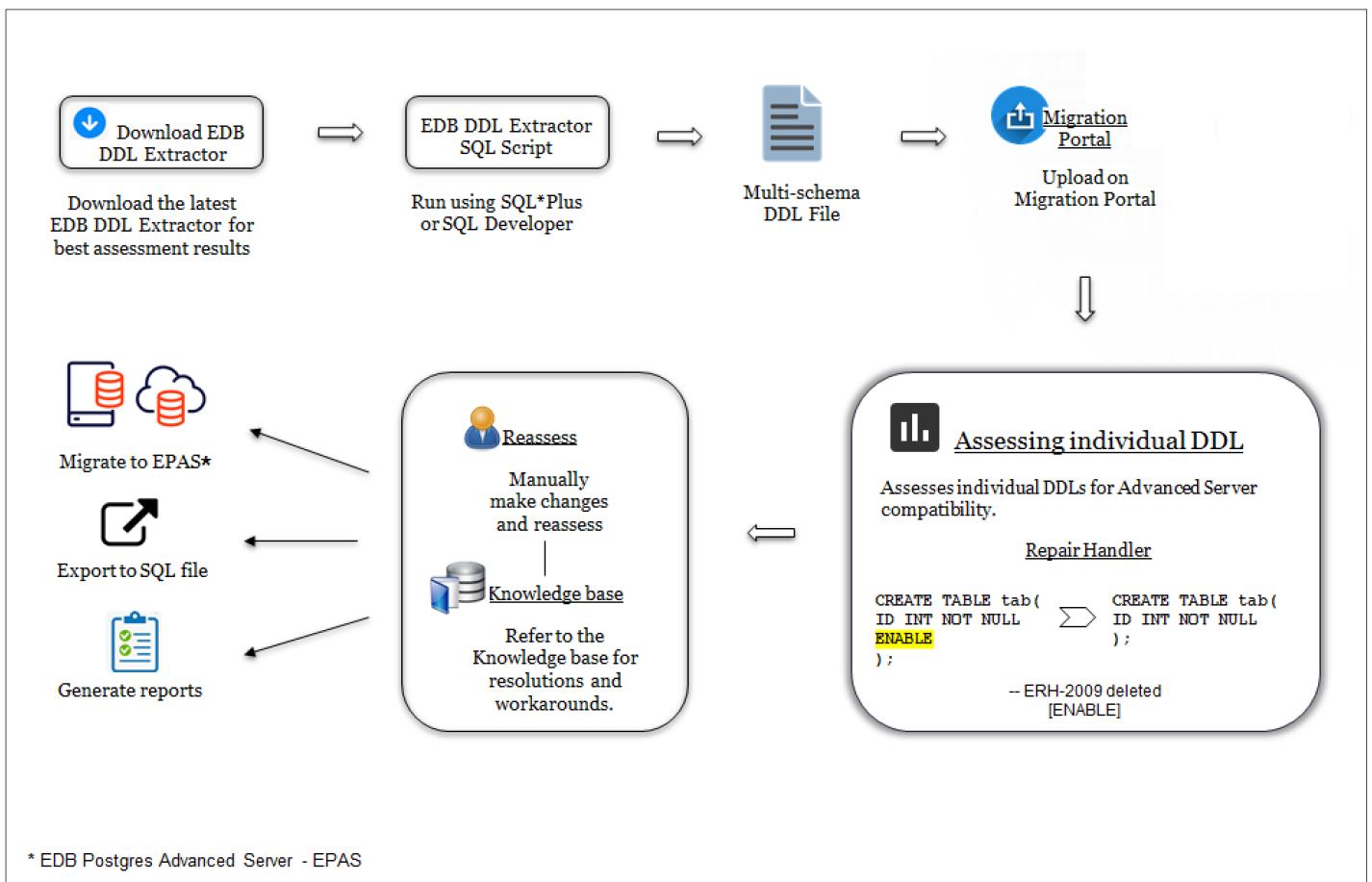
Browser	Supported Version
Apple Safari on Macintosh OS	11 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

### Supported Operating Systems

Operating Systems	Supported Version
Macintosh	OS X Sierra
Windows	10
Linux	CentOS 7

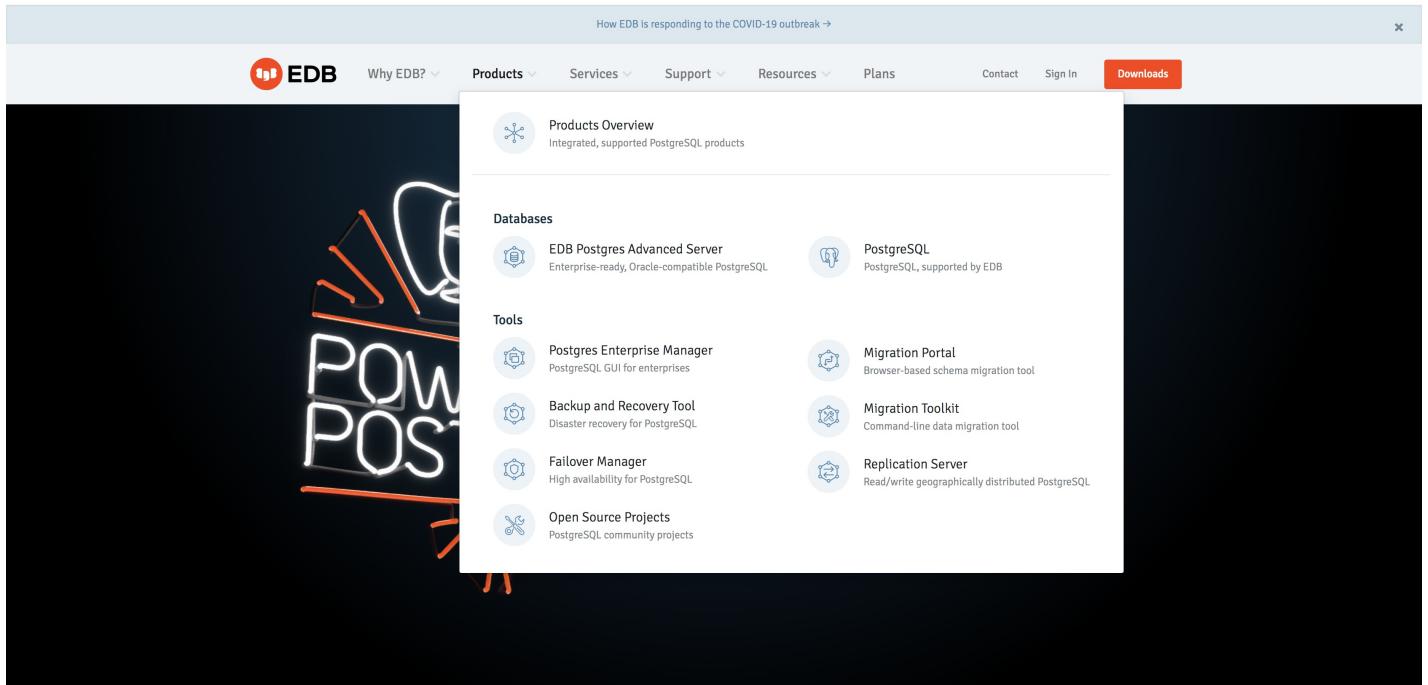
### 3 Using Migration Portal

Migration Portal allows you to easily migrate your database schema from Oracle to EDB Postgres 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.



To access the Migration Portal:

1. Open a browser and navigate to <https://www.edbpostgres.com/>
2. On the EnterpriseDB home page, click **Products > Migration Portal**.



1. Click [Open Migration Portal](#).
2. Log in using your credentials.

### 3.1 Overview of the Migration Portal Home Page

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

The screenshot shows the Migration Portal interface. The top navigation bar includes 'Projects' (highlighted in green), 'Portal Wiki 12', and 'Quick help 11'. The left sidebar lists 'Projects (295) 1' with a '+ New' button. The main content area is divided into several panels:

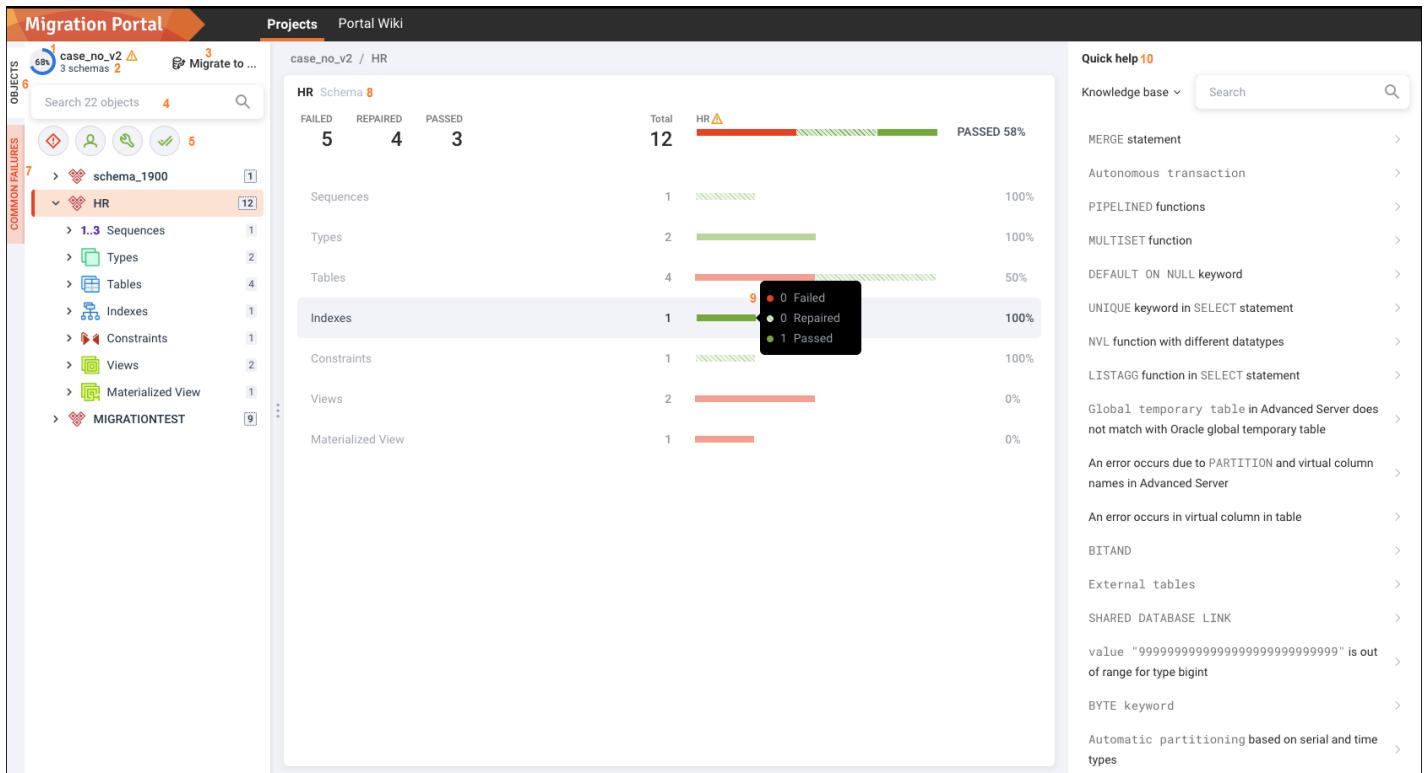
- Overview**: Displays project details: Name (case\_no\_v2), Description (ttt), Interface (ODBC), Source DB (Oracle 11g), Target DB (EDB Postgres Advanced Server 10), and Compatibility (68%). Actions include 'Migrate to ...' (button 5), 'Report' (button 6), and 'Delete' (button 7). A warning message at the top states: 'This project has migration challenges. Contact us for help.' and a 'Contact Us' link.
- Schemas**: Shows schema assessment results for 'HR' (12 Objects), 'MIGRATIONTEST' (9 Objects), and 'schema\_1900' (1 Objects). Each schema has a progress bar indicating failed, repaired, and passed attempts. For example, 'HR' has 5 failed, 2 repaired, and 5 passed attempts, resulting in 58% compatibility.
- Quick help**: Contains links to 'Getting Started', 'Quick Start guide', 'Migration Portal guide', and 'EDB DDL Extractor'.

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

- Projects**: The **Projects** panel displays a list of assessed projects.
- Create project**: Click **+ New** (the button located to the right of the **Projects** label) to create a new project.
- Overview**: The **Overview** panel provides details about the selected project and displays the compatibility percentage after schema assessment.
- Migrate**: Use the **Migrate** button to either download an EDB Postgres Advanced Server compatible **.sql** file or to migrate a schema to an EDB Postgres Advanced Server on-premise or on cloud. For more information, see **Schema Migration <mp\_schema\_migration>**.
- Report**: Use the **Report** button to view and download the schema assessment report.
- Delete**: Use the **Delete** button to delete a selected project.
- 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.
- Upload DDL file**: Use the **Upload new schema** button to upload a new or additional DDL file.
- Schemas**: The **Schemas** panel displays the assessment result from an uploaded DDL file.
- Quick help**: The **Quick help** panel contains all the help guides.
- 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.



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

- Compatible:** The **Compatible** gauge displays the color on the basis of the compatibility percentage of the assessed schema.
- Schema Count:** Displays the number of schemas in a project.
- Migrate to:** Use the **Migrate to** to migrate the schema to an *EDB Postgres Advanced Server* on-premise or on cloud.
- Search objects:** Use the **Search** box to search for objects.
- Filters:** You can filter the system repaired and manual repaired objects from the left panel of the Projects page. In addition, you can select one or more filter combinations to refine the information.
- Objects:** Displays the objects for the selected schemas.
- Common Failures:** Displays the reason for the failed objects for the selected schemas.

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

## Note

You can download a [CSV](#) file for the common failures for the project.

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

```

Source
1 CREATE INDEX HR.DEPARTMENT_LOCATION_IX ON HR.DEPARTMENTS (LOCATION_ID)
2 ;

Target
1 CREATE INDEX DEPT_LOCATION_IX ON HR.DEPARTMENTS (LOCATION_ID)
2 ;

```

## 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
- Migrating Schema
- Migrating Data
- DDL Extractor guide
- Knowledge Base
- Repair handlers
- Migration Portal User's Guide
- FAQs

The screenshot shows the Migration Portal Wiki page. On the left, there is a sidebar with links to various sections like 'Portal Wiki', 'What's new', 'Quick Start guide', etc. The main content area is titled 'What's New' and lists 'Version 2.9.0 (19-Aug-2020)' as the latest enhancement. Below it, there is a modal window titled 'Migrate schemas to' with three options: 'Existing on-premise EDB Postgres Advanced Server', 'New on-premise EDB Postgres Advanced Server Installation', and 'EDB Postgres Advanced Server on Cloud'.

## 4 Migrating a Database

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

1. Perform a **Schema Extraction <mp\_schema\_extraction>**.
2. Perform a **Schema Assessment <mp\_schema\_assessment>**.

3. Perform a Schema Migration <mp\_schema\_migration>.
4. Perform a Data Migration <mp\_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

The *script user* must have [CONNECT](#), [RESOURCE](#) and [SELECT\\_CATALOG\\_ROLE](#) roles.

### 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 or 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.

1. If you want to extract dependent objects from other schemas, enter [yes](#) or [no](#).

### For example, on Linux:

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

SCOTT, FINANCE Location for output file (Default current location) :  
 /home/oracle/extracted\_ddls/

### 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

### On Windows:

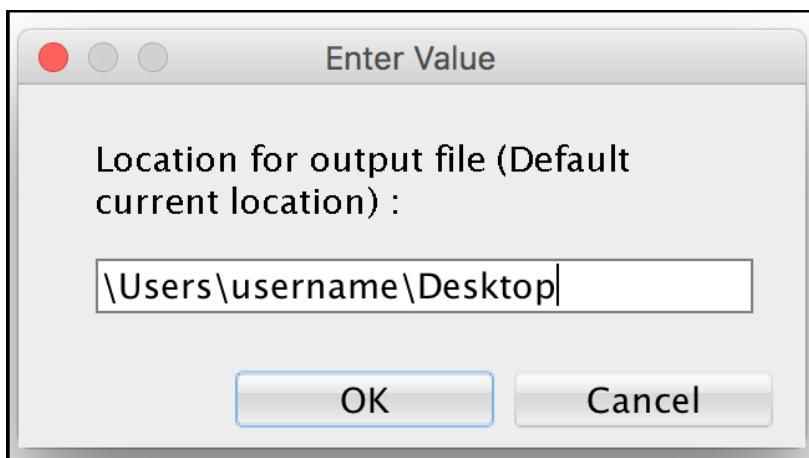
Enter comma separated list of schemas to be extracted (Default all schemas): HR,  
 SCOTT, FINANCE Location for output file (Default current location) :  
 c:\Users\Example\Desktop\

### WARNING:

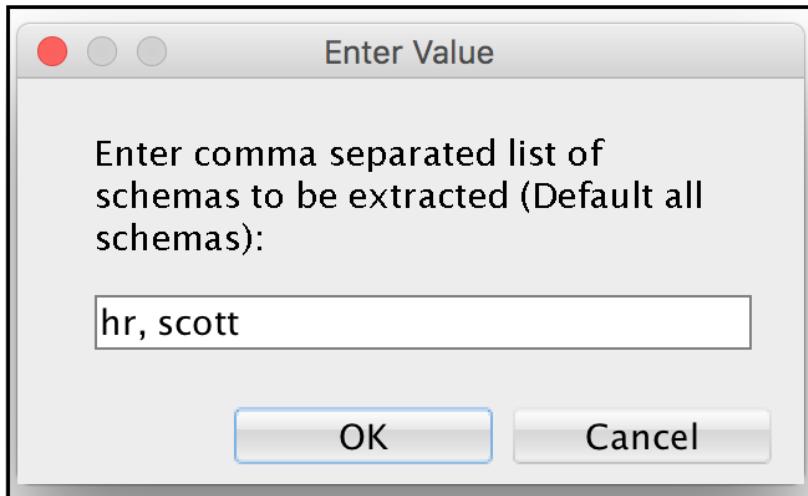
Given schema(s) list may contain objects which are dependent on objects from other schema(s), not mentioned in the list. Assessment may fail for such objects. It is suggested to extract all dependent objects together.

### For SQL Developer

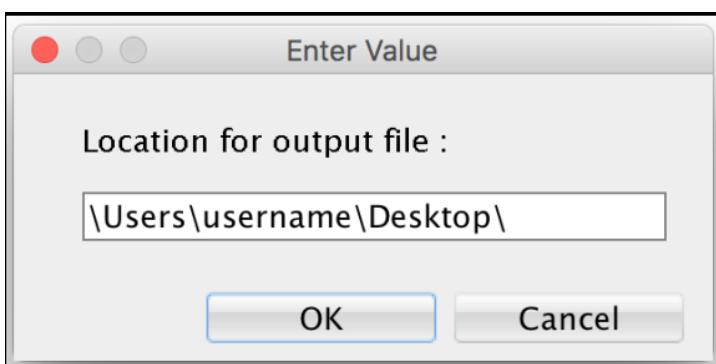
1. Connect to the SQL server and run the following command:



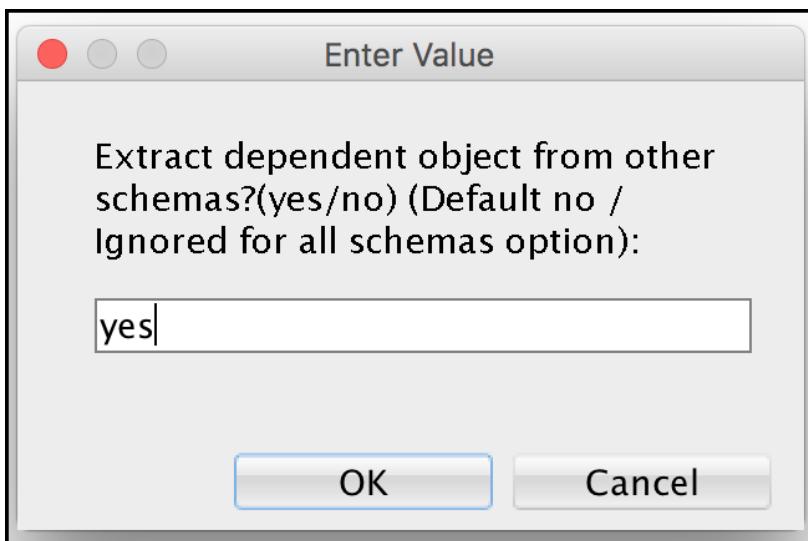
1. Enter a comma-separated list of schemas:



1. Enter the file path for the output file:



1. Enter (yes/no) to extract dependant objects:



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

The script then 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 script does not extract objects that were restored using `Flashback` and still 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.
- DDL Extractor extracts `nologging` tables as normal tables. Once these tables are migrated to EDB Postgres Advanced Server, WAL log files will be created.

## 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 also supported.

## 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

## 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	OJVMSYS
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 XS\$NULL	WMSYS	XDB

## 4.2 Performing a Schema Assessment

To assess an Oracle database schema for compatibility with EDB Postgres 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.

## New project

Project name

Application interface  
 JDBC     ODBC     .NET     OCI     ProC     Other

Source DB

Version

Target DB

Version

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

Add Index Prefix  
i Index objects will be prefixed with "idx" as unique names are required for tables and indexes.

Description(Optional)  
  
200 characters left

1. On the **New project** dialog, enter the project name in the **Project name** field.

## 2. 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.

## 3. Click **Choose file** to upload the .SQL file generated by the latest EDB DDL Extractor for Oracle Database.

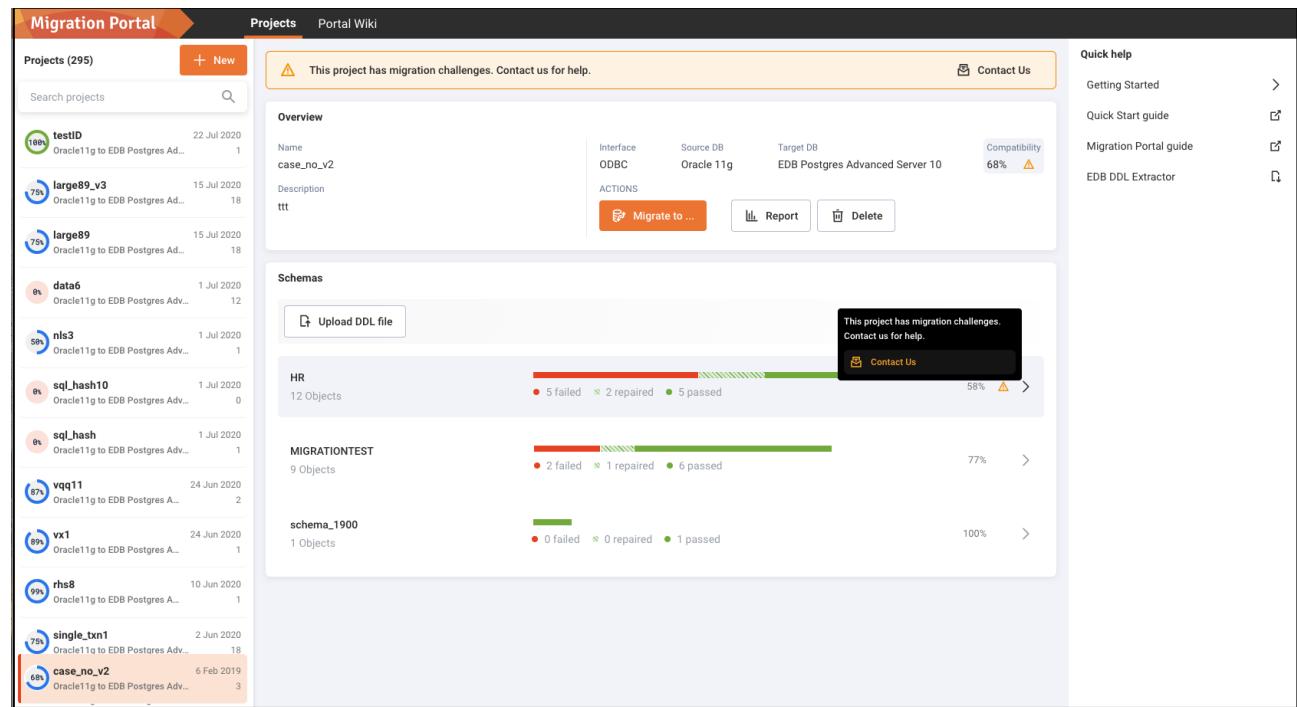
### Note:

- 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 <mp\_schema\_extraction>** section.

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

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



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

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

**case\_no\_v2 / HR**

**HR Schema**

	FAILED	REPAIRED	PASSED	Total	HR	PASSED 58%
Sequences	1			12	HR	100%
Types	2					100%
Tables	4					50%
Indexes	1				0 Failed 0 Repaired 1 Passed	100%
Constraints	1					100%
Views	2					0%
Materialized View	1					0%

**Common Failures**

- case\_no\_v2 (3 schemas)
- Migrate to ...
- Search 22 objects

**Quick help**

- Knowledge base Search
- MERGE statement
- Autonomous transaction
- PIPELINED functions
- MULTISET function
- DEFAULT ON NULL keyword
- UNIQUE keyword in SELECT statement
- NVL function with different datatypes
- LISTAGG function in SELECT statement
- Global temporary table in Advanced Server does not match with Oracle global temporary table
- An error occurs due to PARTITION and virtual column names in Advanced Server
- An error occurs in virtual column in table
- BITAND
- External tables
- SHARED DATABASE LINK
- value "99999999999999999999999999999999" is out of range for type bigint
- BYTE keyword
- Automatic partitioning based on serial and time types

- Click the objects that are not compatible with EDB Postgres Advanced Server and view the details. At the bottom of the middle panel, you can view repair action details.

**case\_no\_v2 / HR / Tables / DEPARTMENTS\_EXT**

**Source**

```

1 CREATE TABLE HR.DEPARTMENTS_EXT
2   (
3     DEPTNO VARCHAR2(4),
4     DNAME VARCHAR2(20),
5     LOCATION VARCHAR2(20)
6   )
7   ORGANIZATION EXTERNAL
8     ( TYPE ORACLE_LOADER
9       (
10          DEPTNO VARCHAR2(4),
11          DNAME VARCHAR2(20),
12          LOCATION VARCHAR2(20)
13        )
14      server file_fdw_server
15      options (filename '/tmp/data.csv', format 'csv');

```

**Target**

```

CREATE FOREIGN TABLE HR.DEPARTMENTS_EXT
  (
    DEPTNO VARCHAR2(4),
    DNAME VARCHAR2(20),
    LOCATION VARCHAR2(20)
  )
server file_fdw_server
options (filename '/tmp/data.csv', format 'csv');

```

**Output**

Repaired

only superuser can change options of a file\_fdw foreign table line 1, char 0

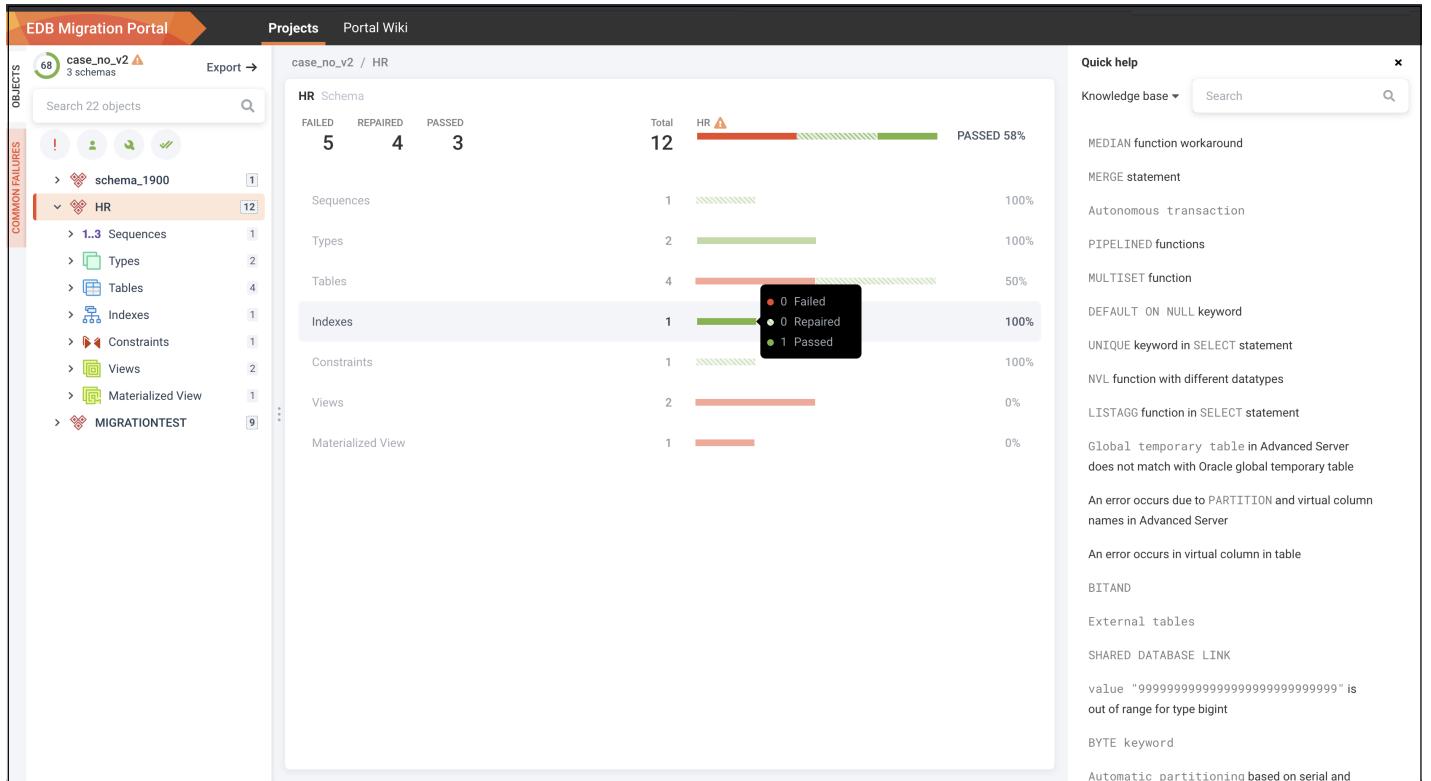
**Common Failures**

- case\_no\_v2 (3 schemas)
- Migrate to ...
- Search 22 objects

**Quick help**

- Knowledge base Search
- MERGE statement
- Autonomous transaction
- PIPELINED functions
- MULTISET function
- DEFAULT ON NULL keyword
- UNIQUE keyword in SELECT statement
- NVL function with different datatypes
- LISTAGG function in SELECT statement
- Global temporary table in Advanced Server does not match with Oracle global temporary table
- An error occurs due to PARTITION and virtual column names in Advanced Server
- An error occurs in virtual column in table
- BITAND
- External tables
- SHARED DATABASE LINK
- value "99999999999999999999999999999999" is out of range for type bigint
- BYTE keyword
- Automatic partitioning based on serial and time types

- Refer to the Knowledge Base information in the right panel to locate the possible workarounds for the objects that are not immediately compatible with EDB Postgres Advanced Server. You can also view the Knowledge Base information on the Portal Wiki page.



1. On the **Knowledge Base** tab, you can enter the error message for the incompatible objects with EDB Postgres Advanced Server and click **Search**.

The object detail panel displays the workaround or the resolution for the failed object. You can manually make the changes on the **Assessment** tab for that object, and click **Reassess**.

**Note:** If any failed object passes while reassessing, the dependent objects for that object are also reassessed.

```

Source
1 CREATE TABLE HR.EMPLOYEES
2   ( EMPLOYEE_ID NUMBER(6,0),
3     FIRST_NAME VARCHAR2(20),
4     LAST_NAME VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL ENABLE,
5     EMAIL VARCHAR2(25) CONSTRAINT EMP_EMAIL_NN NOT NULL DISABLE,
6     PHONE_NUMBER VARCHAR2(20),
7     HIRE_DATE DATE CONSTRAINT EMP_HIRE_DATE_NN NOT NULL ENABLE,
8     JOB_ID VARCHAR2(10) CONSTRAINT EMP_JOB_NN NOT NULL,
9     SALARY NUMBER(8,2),
10    COMMISSION_PCT NUMBER(2,2),
11    MANAGER_ID NUMBER(6,0),
12    DEPARTMENT_ID NUMBER(4,0),
13    CONSTRAINT EMP_SALARY_MIN CHECK (salary > 0) ,
14    CONSTRAINT EMP_EMAIL_UK UNIQUE (EMAIL) ,
15    CONSTRAINT EMP_EMP_ID_PK PRIMARY KEY (EMPLOYEE_ID) ) ;

```

```

Target
1 CREATE TABLE HR.EMPLOYEES
2   ( EMPLOYEE_ID NUMBER(6,0),
3     FIRST_NAME VARCHAR2(20),
4     LAST_NAME VARCHAR2(25) CONSTRAINT EMP_LAST_NAME_NN NOT NULL ,
5     EMAIL VARCHAR2(25) CONSTRAINT EMP_EMAIL_NN NOT NULL ,
6     PHONE_NUMBER VARCHAR2(20),
7     HIRE_DATE DATE CONSTRAINT EMP_HIRE_DATE_NN NOT NULL ,
8     JOB_ID VARCHAR2(10) CONSTRAINT EMP_JOB_NN NOT NULL ,
9     SALARY NUMBER(8,2),
10    COMMISSION_PCT NUMBER(2,2),
11    MANAGER_ID NUMBER(6,0),
12    DEPARTMENT_ID NUMBER(4,0),
13    CONSTRAINT EMP_SALARY_MIN CHECK (salary > 0) ,
14    CONSTRAINT EMP_EMAIL_UK UNIQUE (EMAIL) ,
15    CONSTRAINT EMP_EMP_ID_PK PRIMARY KEY (EMPLOYEE_ID) ) ;

```

**Output** Repaired  
Success

**Quick help**

- Knowledge base Search
- MERGE statement
- Autonomous transaction
- PIPELINED functions
- MULTISET function
- DEFAULT ON NULL keyword
- UNIQUE keyword in SELECT statement
- NVL function with different datatypes
- LISTAGG function in SELECT statement
- Global temporary table in Advanced Server does not match with Oracle global temporary table
- An error occurs due to PARTITION and virtual column names in Advanced Server
- An error occurs in virtual column in table
- BITAND
- External tables
- SHARED DATABASE LINK
- value "99999999999999999999999999999999" is out of range for type bigint
- BYTE keyword
- Automatic partitioning based on serial and time types

Similarly, you can make all the incompatible objects compatible.

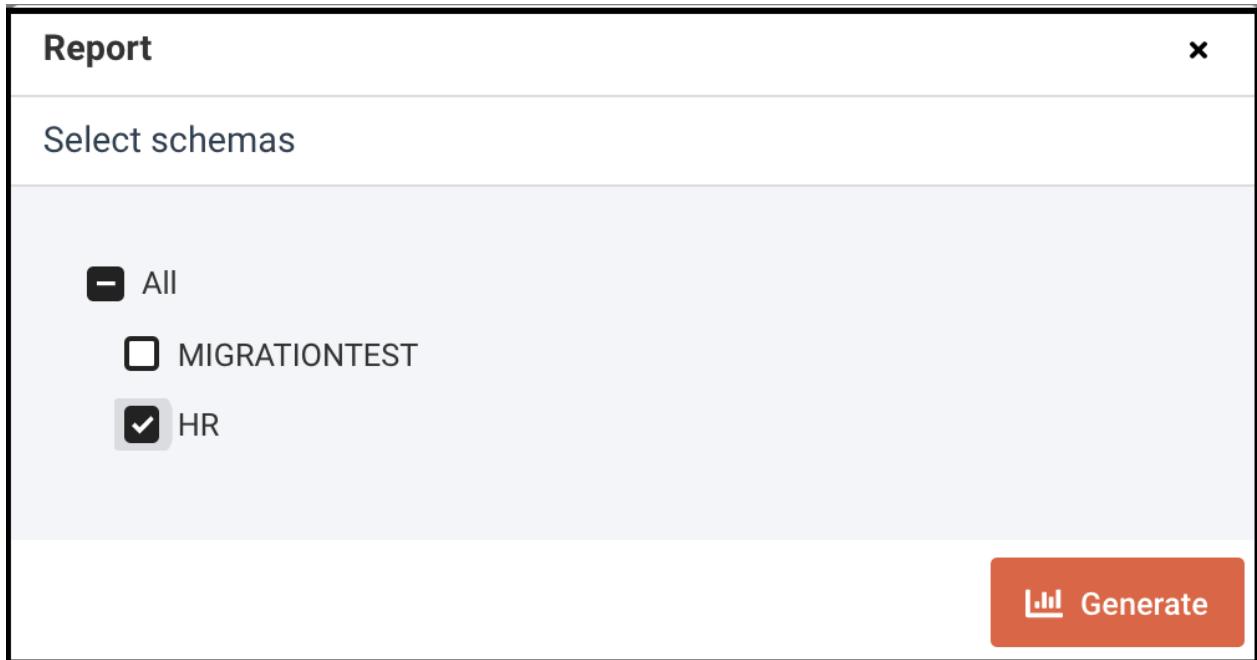
## Note

If the information or workaround for incompatible objects is 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 see, [Schema Migration](#).

## 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.



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.

The screenshot shows the 'Schema Assessment Report' page. At the top left is the EDB logo and project information: 'Project: case\_no\_v2', 'Source database: Oracle 11g', 'Target database: EDB Postgres Advanced Server 10', 'Description: ttt', and 'Generated by Migration Portal v2.8.0 on 11th August 2020, 2:27:41 pm'. On the far right is a 'Generate PDF' button. The main content includes a 'Summary' section with a progress bar showing 58% compatibility, and detailed tables for failed objects categorized by object type (Constraints, Indexes, Materialized View, Sequences, Tables, Types, Views) and common failures.

Object type	Total	Passed	Repaired	Failed	Compatibility
HR	12 objects	5	2	5	58%

Object type	Total	Passed	Repaired	Failed
Constraints	1	1	0	0
Indexes	1	1	0	0
Materialized View	1	0	0	1
Sequences	1	0	1	0
Tables	4	1	1	2
Types	2	2	0	0
Views	2	0	0	2

Failed Objects		
Schema	Total	Failed
HR	12	5

Common Failures		
Count	Reason	Occurrences
2	only superuser can change options of a file, f/w 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)

All failed objects		
Object type	Failed object	Reason for failing
Materialized View	HR_EMP	line 1, char 297 relation "hr.emp" does not exist
1 Failed		
0 Repaired		
0 Passed		

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.

11/08/2020

EDB Assessment Report-case\_no\_v2

# Schema Assessment Report



**Project:**  
case\_no\_v2

**Source database:**  
Oracle 11g

**Target database:**  
EDB Postgres Advanced Server 10

Generated by  
**Migration Portal v2.8.0**  
11th August 2020, 12:24:05 pm

**Description:**  
ttt

## Summary

This report provides a high-level assessment summary for selected schemas you assessed under "**case\_no\_v2**" project. In addition, the report provides details of the failed objects and the cause of failure.

<b>1</b>	<b>12</b>	<b>5</b>	<b>2</b>	<b>5</b>	<b>58%</b>
Schemas	Total objects	Passed	Repaired	Failed	Compatibility



Object type	Total	Passed	Repaired	Failed
Constraints	1	1	0	0
Indexes	1	1	0	0
Materialized View	1	0	0	1
Sequences	1	0	1	0
Tables	4	1	1	2
Types	2	2	0	0
Views	2	0	0	2

## Failed Objects

Schema	Total	Failed
HR	12	5

### Common Failures

Count	Reason	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)

<https://stage-migration.enterprisedb.com>

1/2

## 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 Postgres Advanced Server.

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

You can choose one of the following options for migrating schemas; migrate to an:

- Existing on-premise EDB Postgres Advanced Server
- New on-premise EDB Postgres Advanced Server
- EDB Postgres Advanced Server on Cloud

Migrate schemas to

?

Help



Existing on-premise EDB Postgres Advanced Server

Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.

[Cancel](#)



New on-premise EDB Postgres Advanced Server Installation

Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.



EDB Postgres Advanced Server on Cloud

Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

[→ Next](#)

### Existing on-Premise EDB Postgres Advanced Server

You can migrate schemas to an existing on-premise EDB Postgres Advanced Server on Windows or Linux platforms.

## Migrating schemas on Windows

1. Select the **Existing on-premise EDB Postgres Postgres Advanced Server** option:

**Migrate schemas to**

?



**Existing on-premise EDB Postgres Advanced Server**

Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.



**New on-premise EDB Postgres Advanced Server Installation**

Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.



**EDB Postgres Advanced Server on Cloud**

Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

**Cancel**

→ **Next**

1. Select one or more schemas to migrate on EDB Postgres Advanced Server:

## Migrate schemas to existing on-premise EDB Postgres Advanced Server

?

1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

**⚠ This project has migration challenges. Contact us for help.**

Contact Us X

Project Name case_no_v2	Interface ODBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 10	Compatibility 68% <b>⚠</b>
----------------------------	-------------------	-------------------------	--	-------------------------------

Select one or more schemas

All  
 schema\_1900  
 HR  
 MIGRATIONTEST

Cancel ← Previous → Next

### Note

In case your schemas are not 100% compatible, a banner will be displayed as shown, and you can fill the Contact Us form for any assistance required.

**Contact Us**

Your message...

Required project details will be sent along with your message.

By submitting this form, I agree to EDB's [Terms & Conditions](#) and [Privacy Policy](#).

Send

1. Download the assessed schemas:

## Migrate schemas to existing on-premise EDB Postgres Advanced Server

?

1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Download the assessed schemas**

 [Download SQL file](#)

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 11 (your assessed version).

[Cancel](#) [← Previous](#) [→ Next](#)

1. Click on **Windows**:

## Migrate schemas to existing on-premise EDB Postgres Advanced Server

?

1 Select Schemas 2 Download Schema 3 Import 4 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Import the selected schemas**

Run the following command:

[Windows](#) [Linux](#)

On CLI

```
edb=# \i c:\users\...\<project_name>.sql
```

On cmd/shell

```
edb-psql -f <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use  pgAdmin instead.

[Cancel](#) [← Previous](#) [→ Next](#)

### 1. To import the schemas, run the following command:

- On CLI

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell

```
edb-psql -f <project_name>.sql
```

## Migrating schemas on Linux

To migrate schemas to an existing on-premise EDB Postgres Advanced Server on Linux, complete the following steps:

1. Click the **Existing on-premise EDB Postgres Advanced Server** option:

### Migrate schemas to



**Existing on-premise EDB Postgres Advanced Server**



Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.

**New on-premise EDB Postgres Advanced Server Installation**



Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.

**EDB Postgres Advanced Server on Cloud**



Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

**Cancel**

**→ Next**

1. Select one or more schemas to migrate to EDB Postgres Advanced Server:

## Migrate schemas to existing on-premise EDB Postgres Advanced Server



1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

Select one or more schemas

- All
- HR
- DEPEND\_SECOND
- DEPEND\_FIRST

Cancel

← Previous

→ Next

### 1. Download the assessed schemas:

## Migrate schemas to existing on-premise EDB Postgres Advanced Server



1 Select Schemas — 2 Download Schema — 3 Import — 4 Finish

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

Download the assessed schemas

Download SQL file

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 11 (your assessed version).

Cancel

← Previous

→ Next

1. Click on **Linux**:

**Migrate schemas to existing on-premise EDB Postgres Advanced Server**

Project Name: ID2019    Interface: JDBC    Source DB: Oracle 11g    Target DB: EDB Postgres Advanced Server 11    Compatibility: 100%

**Import the selected schemas**

Run the following command:

Windows    **Linux**

Run the following command on Terminal

```
shell=$ sudo su - enterprisedb
shell=$ edb-psql edb
edb=# create database <database_name>;
edb=# \connect <database_name>
database_name=# \i <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use pgAdmin instead.

[Cancel](#)    [← Previous](#)    [→ Next](#)

1. To import the schemas, run the following command:

```
sudo su - enterprisedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

## New On-Premise EDB Postgres Advanced Server

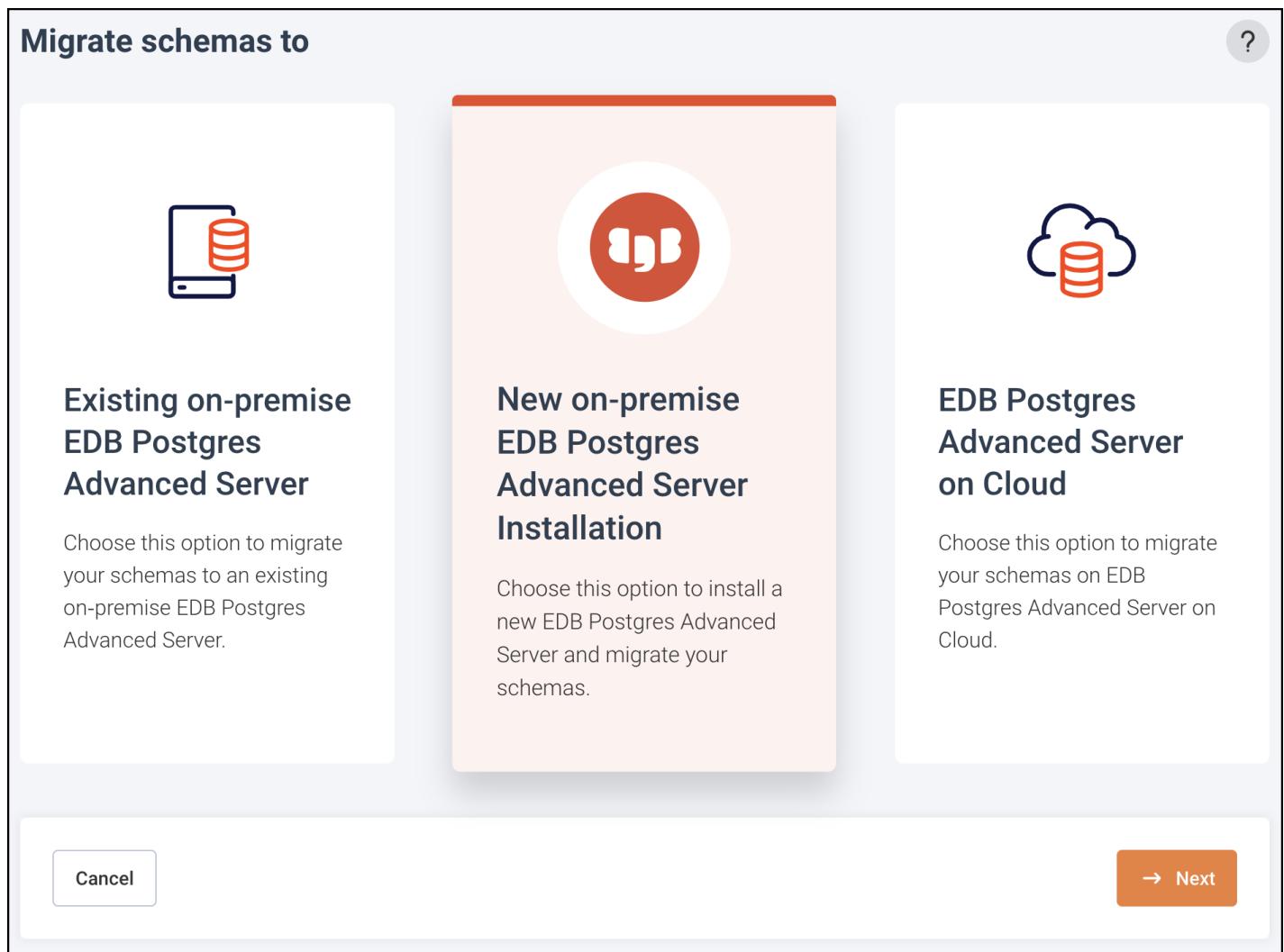
You can install new EDB Postgres Advanced Server on-premise on Windows or Linux

platforms and migrate the schemas.

## Migrating schemas on Windows

To migrate schemas to a new on-premise EDB Postgres Advanced Server on Windows, complete the following steps:

1. Click **New on-premise EDB Postgres Postgres Advanced Server** option.



1. Select one or more schemas to migrate on EDB Postgres Advanced Server.

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

- All
- HR
- DEPEND\_SECOND
- DEPEND\_FIRST

[Cancel](#)

[← Previous](#)

[→ Next](#)

1. Select the Windows operating system.

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 **Select Platform** — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Select the Operating System/Platform

Windows 64-bit 

Linux 64-bit 

[Cancel](#) [← Previous](#) [→ Next](#)

### 1. Download Windows Installer.

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 Select Platform — 3 **Get EDB Postgres Advanced Server** — 4 Install EDB Postgres Advanced Server

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Install EDB Postgres Advanced Server for Windows

 Download Windows installer 

[Cancel](#) [← Previous](#) [→ Next](#)

- For installation steps, click EDB Postgres Advanced Server Installation Guide for Windows.

### Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

Get EDB Postgres Advanced Server **4 Install EDB Postgres Advanced Server** Download Schema **6**

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Install EDB Postgres Advanced Server on Windows**

EDB Postgres™ Advanced Server Installation Guide for Windows

**Cancel** Previous Next

- Download the assessed schemas.

### Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

Advanced Server **5 Download Schema** Import **7** Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Download the assessed schemas**

Download SQL file

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 11 (your assessed version).

**Cancel** Previous Next

1. You can import schemas by running the following command:

- On CLI

```
\i c:\users\...\<project_name>.sql
```

- On cmd/shell

```
edb-psql -f <project_name>.sql
```

**Migrate schemas to new on-premise EDB Postgres Advanced Server Installation** ?

Advanced Server — ✓ Install EDB Postgres Advanced Server — ✓ Download Schema — 6 Import — 7 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Import schemas into EDB Postgres Advanced Server**

On CLI

```
edb=# \i c:\users\...\<project_name>.sql
```

On cmd/shell

```
edb-psql -f <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use  pgAdmin instead.

[Cancel](#) [← Previous](#) [→ Next](#)

### Note

You can also use pgAdmin instead.

The schemas are migrated to the target server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

< Advanced Server ✓ Install EDB Postgres Advanced Server ✓ Download Schema ✓ Import 7 Finish

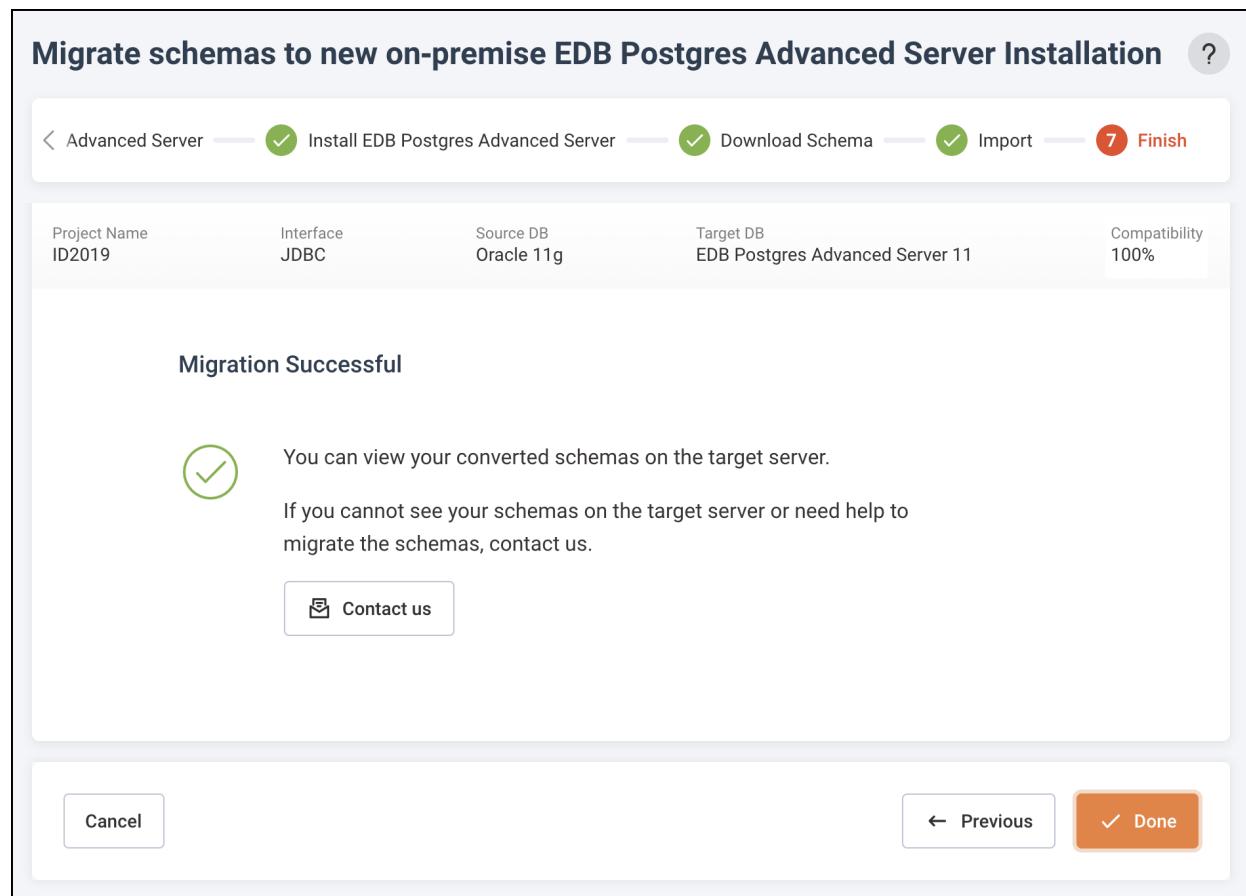
Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Migration Successful**

You can view your converted schemas on the target server.  
If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

Contact us

[Cancel](#) [← Previous](#) [✓ Done](#)



## Migrating schemas on Linux

To migrate schemas to an on-premise EDB Postgres Advanced Server on Linux, complete the following steps:

1. Click **New On-premise EDB Postgres Advanced Server** option.

## Migrate schemas to



### Existing on-premise EDB Postgres Advanced Server

Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.



### New on-premise EDB Postgres Advanced Server Installation

Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.



### EDB Postgres Advanced Server on Cloud

Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

Cancel

→ Next

1. Select one or more schemas to migrate on EDB Postgres Advanced Server.

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 Select Platform — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

All

HR

DEPEND\_SECOND

DEPEND\_FIRST

[Cancel](#) [← Previous](#) [→ Next](#)

1. Select the **Linux** operating system.

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

1 Select Schemas — 2 **Select Platform** — 3 Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Select the Operating System/Platform

Windows 64-bit 

Linux 64-bit 

[Cancel](#) [← Previous](#) [→ Next](#)

1. You can select one of the following options to install the EDB Postgres Advanced Server:
  - Repository
  - More options

The screenshot shows the 'Migrate schemas to new on-premise EDB Postgres Advanced Server Installation' interface. At the top, a progress bar indicates steps 1 through 4: 'Select Schemas' (green checkmark), 'Select Platform' (green checkmark), 'Get EDB Postgres Advanced Server' (red circle with '3'), and 'Install EDB Postgres Advanced Server' (grey circle with '4'). Below the progress bar, project details are listed: Project Name (ID2019), Interface (JDBC), Source DB (Oracle 11g), Target DB (EDB Postgres Advanced Server 11), and Compatibility (100%). The main section is titled 'Install EDB Postgres Advanced Server for Linux' and contains two options: 'Repository' and 'More options', each with a corresponding icon and a right-pointing arrow. At the bottom, there are 'Cancel', 'Previous', and 'Next' buttons.

2. For information on the installation procedure, click [EDB Postgres Advanced Server Installation Guide](#) for Linux:

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< — ✓ Get EDB Postgres Advanced Server — 4 Install EDB Postgres Advanced Server — 5 Download Schema — 6 >

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

### Install EDB Postgres Advanced Server on Linux



EDB Postgres™ Advanced Server Installation Guide for Linux



[Cancel](#)

[← Previous](#)

[→ Next](#)

### 1. Download the assessed schemas:

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

< Advanced Server — ✓ Install EDB Postgres Advanced Server — 5 Download Schema — 6 Import — 7 Finish

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

### Download the assessed schemas

✚ [Download SQL file](#)

Note: Ensure the version number for the EDB Postgres Advanced Server database that you are migrating is 11 (your assessed version).

[Cancel](#)

[← Previous](#)

[→ Next](#)

### 1. To import the schemas, run the following command:

```
sudo su - enterpriseedb
edb-psql edb
create database <database_name>;
\connect <database_name>
\i <project_name>.sql
```

## Migrate schemas to new on-premise EDB Postgres Advanced Server Installation

Advanced Server → ✓ Install EDB Postgres Advanced Server → ✓ Download Schema → 6 Import → 7 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

### Import schemas into EDB Postgres Advanced Server

Run the following command on Terminal

```
shell=$ sudo su - enterpriseedb
shell=$ edb-psql edb
edb=# create database <database_name>;
edb=# \connect <database_name>
database_name=# \i <project_name>.sql
```

[PSQL Documentation](#) | [Advanced Server Documentation](#)

Use  pgAdmin instead.

[Cancel](#) [← Previous](#) [→ Next](#)

### Note

You can also use pgAdmin instead.

The converted schemas are migrated to the target server.

Migrate schemas to new on-premise EDB Postgres Advanced Server Installation ?

Advanced Server ✓ Install EDB Postgres Advanced Server ✓ Download Schema ✓ Import 7 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

Migration Successful

You can view your converted schemas on the target server.  
If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

Contact us

Cancel Previous Done

## EDB Postgres Advanced Server on Cloud

To migrate schemas on EDB Postgres Advanced Server on Cloud, complete the following steps:

1. Click **EDB Postgres Advanced Server on Cloud** option:

## Migrate schemas to



### Existing on-premise EDB Postgres Advanced Server

Choose this option to migrate your schemas to an existing on-premise EDB Postgres Advanced Server.



### New on-premise EDB Postgres Advanced Server Installation

Choose this option to install a new EDB Postgres Advanced Server and migrate your schemas.



### EDB Postgres Advanced Server on Cloud

Choose this option to migrate your schemas on EDB Postgres Advanced Server on Cloud.

Cancel

→ Next

1. Select one or more schemas to migrate on EDB Postgres Advanced Server:

## Migrate schemas to EDB Postgres Advanced Server on Cloud



1 Select Schemas — 2 Select Platform — 3 Launch Cluster — 4 Connect — 5 Deploy — 6 Finish

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

Select one or more schemas

All

HR

DEPEND\_SECOND

DEPEND\_FIRST

[Cancel](#)

[← Previous](#)

[→ Next](#)

1. Select the cloud platform. For example, **IBM Cloud**:

## Migrate schemas to EDB Postgres Advanced Server on Cloud



1 Select Schemas — 2 Select Platform — 3 Launch Cluster — 4 Connect — 5 Deploy — 6 Finish

Project Name  
ID2019

Interface  
JDBC

Source DB  
Oracle 11g

Target DB  
EDB Postgres Advanced Server 11

Compatibility  
100%

Select the Cloud Platform



IBM Cloud

[Cancel](#)

[← Previous](#)

[→ Next](#)

1. To launch a new cluster, click **Go to Cloud**:

## Migrate schemas to EDB Postgres Advanced Server on Cloud

?

1 Select Schemas   2 Select Platform   3 Launch Cluster   4 Connect   5 Deploy   6 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Launch new IBM™ Cloud cluster**

Navigate to IBM™ Cloud to launch a new cluster.

→ Go to IBM™ Cloud

Note: If you have an existing cluster running, click Next.

[Cancel](#)   [← Previous](#)   **→ Next**

Or, if you have an existing cluster running, click **Next**.

1. Enter the required connection details on the Connect page:

## Migrate schemas to EDB Postgres Advanced Server on Cloud

Project Name: ID2019    Interface: JDBC    Source DB: Oracle 11g    Target DB: EDB Postgres Advanced Server 11    Compatibility: 100%

Connect to IBM™ Cloud cluster

Target Database: ID2019

Host Name/Address: Host Name/Address    Port: 9999

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

Maintenance Database: edb

Username: enterprisedb

Password:

[Cancel](#) [← Previous](#) [Next →](#)

The screenshot shows the 'Connect' step of a six-step migration process. The top navigation bar indicates steps 1 through 6: Select Schemas, Select Platform, Launch Cluster, Connect (highlighted in red), Deploy, and Finish. Below the navigation, project details are listed: Project Name (ID2019), Interface (JDBC), Source DB (Oracle 11g), Target DB (EDB Postgres Advanced Server 11), and Compatibility (100%). The main section is titled 'Connect to IBM™ Cloud cluster' and contains fields for 'Target Database' (ID2019), 'Host Name/Address' (Host Name/Address), 'Port' (9999), 'Maintenance Database' (edb), 'Username' (enterprisedb), and 'Password'. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

### Note

You can click [Edit Connection](#) to make changes to the connection details and retest the connection details.

1. Click [Test Connection](#) to verify the connection details:

## Migrate schemas to EDB Postgres Advanced Server on Cloud

?

1 Select Schemas 2 Select Platform 3 Launch Cluster 4 Connect 5 Deploy 6 Finish

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Target Database**  
ID2019

**Host Name/Address** Port  
52.5.148.59 5444

**Maintenance Database**  
dmp\_exec

Note: Deployment will create a new database **ID2019** 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.

 Edit Connection

[Cancel](#) [← Previous](#) [Next →](#)

- Once the connection is successful, click **Next**:

**Migrate schemas to EDB Postgres Advanced Server on Cloud**

Project Name ID2019	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 11	Compatibility 100%
------------------------	-------------------	-------------------------	--	-----------------------

**Migration Successful**

You can view your converted schemas on the target server.

[Download Summary](#)

If you cannot see your schemas on the target server or need help to migrate the schemas, contact us.

[Contact us](#)

[Cancel](#) [← Previous](#) [✓ Done](#)

The converted schemas are migrated to the target server.

## 4.4 Data Migration

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

1. Use Migration Toolkit to migrate the data. For detailed information about using Migration Toolkit, see the [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](#).

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

For example, the following command migrates data to a schema named `hr`:

```
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.

---

## 5 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 [Migration Toolkit Guide](#).