



EDB Postgres Migration Portal

Version 3.0.1

1	What's New	3
2	Supported Versions	3
3	Using Migration Portal	3
3.1	Overview of the Migration Portal Home Page	5
3.2	Overview of the Migration Portal Projects Page	7
3.3	Overview of the Migration Portal Wiki Page	9
4	Migrating a Database	10
4.1	Performing a Schema Extraction	10
4.2	Performing a Schema Assessment	15
4.3	Schema Migration	23
4.4	Data Migration	54
5	Advanced Data Migration	55

1 What's New

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

- Made minor UI enhancements for better user experience.

2 Supported Versions

The Migration Portal supports assessment and migration from Oracle 11g and 12c to EDB Postgres Advanced Server 10, 11, 12, Or 13. 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.



The assessment and migration process

Note: Migration Portal currently does not have team collaboration features. We recommend collaborative meetings, screen sharing, and using the downloadable assessment report to share results.

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**.



Accessing the Migration Portal

3. Click [Open Migration Portal](#).
4. 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 EDB migration portal interface. On the left, a sidebar lists five projects: ProjectManagement, HumanResources, Finance, Marketing, and Administration, all last updated on 22 Sep 2020. The main area has tabs for 'Projects' and 'Portal Wiki 12'. The 'Overview' tab is active, displaying details for the selected project, ProjectManagement, which is 71% compatible. It shows the interface as JDBC, source DB as Oracle 11g, and target DB as EDB Postgres Advanced Server 12. Below this, the 'Schemas' tab is shown, listing three schemas: HR, HRM, and schema_6595, each with its compatibility status (100%, 53%, and 100% respectively) and repair counts. A 'Quick help' sidebar on the right provides links to various guides and a 'Contact Us' button.

The Migration Portal home

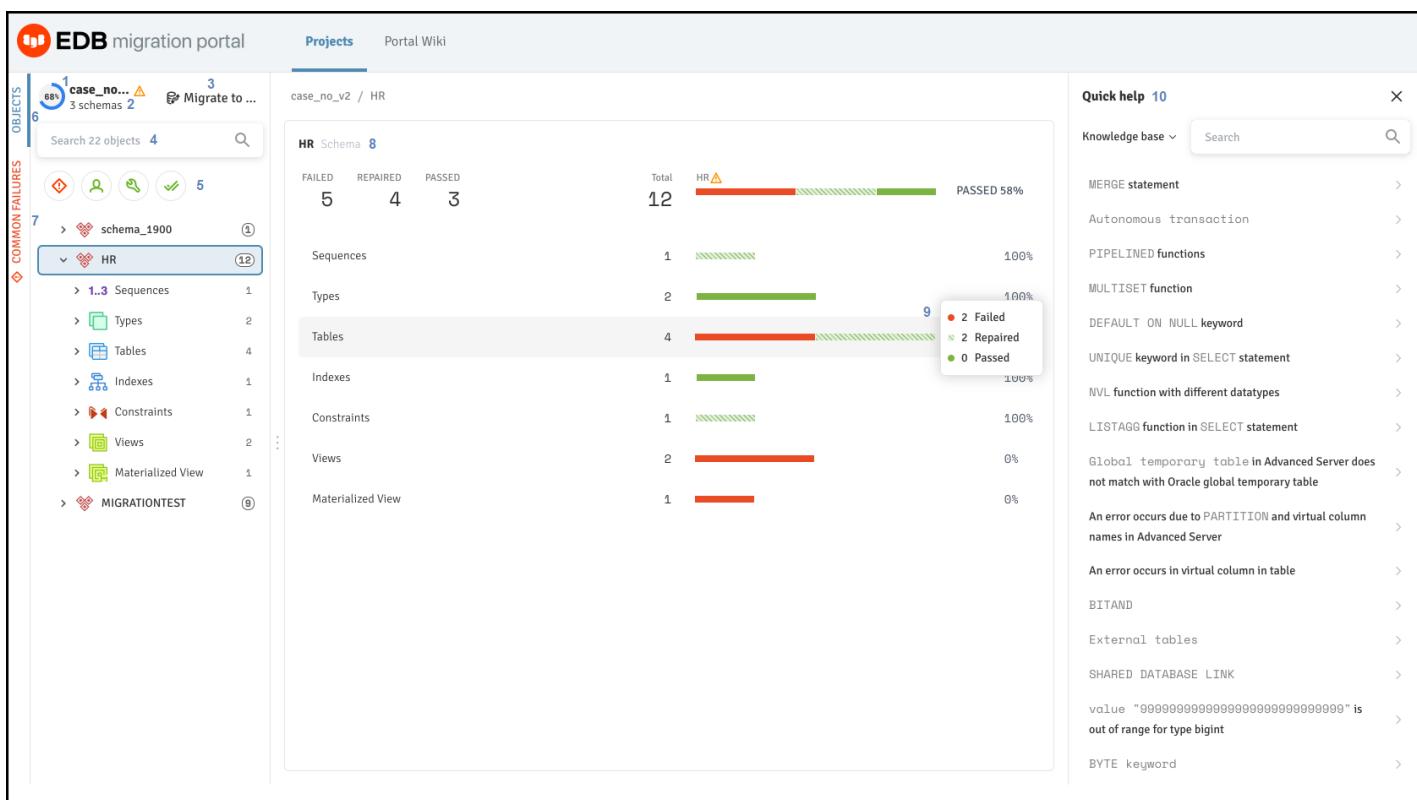
The Migration Portal home page allows access to the following Migration Portal features:

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. **Search Projects:** Use the **Search** box to search for projects.
4. **Overview:** The **Overview** panel provides details about the selected project and displays the compatibility percentage after schema assessment.
5. **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](#).
6. **Report:** Use the **Report** button to view and download the schema assessment report.
7. **Delete:** Use the **Delete** button to delete a selected project.
8. **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.
9. **Upload DDL file:** Use the **Upload new schema** button to upload a new or additional DDL file.
10. **Schemas:** The **Schemas** panel displays the assessment result from an uploaded DDL file.

1. **Quick help:** The **Quick help** panel contains all the help guides.
2. **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.



The Migration Portal Projects Page overview.

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

1. **Compatible:** The **Compatible** gauge displays the color on the basis of the compatibility percentage of the assessed schema.
2. **Schema Count:** Displays the number of schemas in a project.
3. **Migrate to:** Use the **Migrate to** to migrate the schema to an *EDB Postgres Advanced Server* on-premise or on cloud.
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 sidebar.

panel of the Projects 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.

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 "FROM"	Views 1
1	relation "hr.emp" does not exist	Materialized View 1

 A tooltip over the 'HR' schema row indicates 5 objects."/>

Common Failures tab

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.

The screenshot shows the EDB migration portal interface. On the left, there's a sidebar with 'OBJECTS' and 'COMMON FAILURES' sections. The main area has tabs for 'Projects' and 'Portal Wiki'. A search bar at the top says 'Search 22 objects'. Below it, a tree view shows a project named 'case_no_v2' containing an 'HR' schema with various tables like 'Sequences', 'Types', 'Tables', and 'Indexes'. An 'Indexes' node is selected, showing a sub-tree with 'DEPT_LOCATION_IX'. The central part of the screen displays two panes: 'Source' and 'Target', both showing the same DDL command:

```

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

```

Below these panes is a 'Quick help' section with a search bar containing 'Merg'. At the bottom, there are tabs for 'OUTPUT' (which is 'Success') and 'REPAIRED'.

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

The screenshot shows the EDB migration portal Wiki page. At the top, there are navigation links for 'Projects' and 'Portal Wiki'. On the right, there is a user profile for 'Nancy Drew' with the initials 'ND'. The left sidebar has a 'Portal Wiki' heading and a 'What's new' button, which is highlighted with a blue border. Below this are links to various guides: Quick Start guide, Migrating Schema, Migrating Data, DDL Extractor guide, Knowledge Base, Repair Handler, Migration Portal guide, and FAQ. The main content area starts with a 'What's New' section, followed by 'Version 3.0.1 (08-Jan-2021)', 'UI Enhancements', 'Version 3.0.0 (09-Dec-2020)', 'Migration Portal now supports', 'Added New Cloud Platforms', and a list of supported cloud platforms: Amazon AWS EC2 instance, Google cloud, and Microsoft Azure. To the right of the main content, there is a vertical list of 'Jump to version' links, starting from Version 3.0.1 down to Beta 3.

Jump to version

- Version 3.0.1 (08-Jan-2021)
- Version 3.0.0 (09-Dec-2020)
- Version 2.10.0 (16-Sep-2020)
- Version 2.9.0 (19-Aug-2020)
- Version 2.8.0 (11-Jun-2020)
- Version 2.7.0 (14-May-2020)
- Version 2.6.0 (19-March-2020)
- Version 2.5.0 (06-February-2020)
- Version 2.4.1 (28-January-2020)
- Version 2.4.0 (11-December-2019)
- Version 2.3.0 (03-October-2019)
- Version 2.2.0 (04-September-2019)
- Version 2.1.0 (16-July-2019)
- Version 2.0.0 (21-May-2019)
- Version 1.1.1 (16-April-2019)
- Version 1.1.0 (28-March-2019)
- Version 1.0.0 (19-February-2019)
- Beta 5 (07-Jan-2019)
- Beta 4 (11-Dec-2018)
- Beta 3 (11-Oct-2018)

The Migration Portal Wiki page

4 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 CONNECT and SELECT_CATALOG_ROLE roles and CREATE TABLE privilege.

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.

3. 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.

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

For SQL Developer

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



Enter the path for Linux or Windows

2. Enter a comma-separated list of schemas:



Provide a list of schemas

3. Enter the file path for the output file:



Specify the output file path

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



Extracting dependent 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.
- DDL Extractor creates Global Temporary tables to store the schema names and their dependency information. These tables are dropped at the end of successful extraction.
- DDL Extractor script does not extract schemas whose name starts with **PG_** because PostgreSQL does not support it. If you want to extract these schemas, you must change name of schema before extraction.

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 X

Project name

Application interface
 JDBC ODBC .NET OCI ProC Other

Source DB Version
▼ ▼

Target DB Version
▼ ▼

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

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

Description(Optional)
200 characters left

X Cancel ✓ Create & assess

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 EDB Postgres 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.

The screenshot shows the EDB migration portal's 'Projects' page. A project named 'case_no_v2' is selected. The 'Overview' section shows the project's name, description, interface (ODBC), source DB (Oracle 11g), target DB (EDB Postgres Advanced Server 10), and compatibility (68%). The 'Schemas' section lists three schemas: HR, MIGRATIONTEST, and schema_1900, each with a progress bar indicating migration status. A message box on the right says 'This project has migration challenges. Contact us for help.' and includes a 'Contact Us' button.

Name	Description	Interface	Source DB	Target DB	Compatibility
case_no_v2	ttt	ODBC	Oracle 11g	EDB Postgres Advanced Server 10	68%

Schemas

Schema	Objects	Status	Progress (%)
HR	12 Objects	5 failed, 2 repaired, 5 passed	58%
MIGRATIONTEST	9 Objects	2 failed, 1 repaired, 6 passed	77%
schema_1900	1 Objects	0 failed, 0 repaired, 1 passed	100%

The Schema analysis result

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.

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

The screenshot shows the EDB migration portal interface. On the left, there's a sidebar with 'COMMON FAILURES' and 'OBJECTS' sections. The main panel displays the 'case_no_v2 / HR' project. It shows a summary of objects by type: FAILED (5), REPAIRED (4), PASSED (3). Below this, detailed statistics are provided for Sequences, Types, Tables, Indexes, Constraints, Views, and Materialized View. For example, Tables have 4 objects, all of which are Failed (red). A legend on the right indicates: Failed (red), Repaired (green), Passed (blue). The overall success rate is shown as 58%.

Verifying the DDL objects

10. 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.

This screenshot shows the EDB migration portal focusing on the 'DEPARTMENTS_EXT' table under the 'Tables' section. The 'Source' tab displays the SQL code for creating the table:

```

1 CREATE TABLE HR.DEPARTMENTS_EXT
2   ( DEPTNO VARCHAR2(4),
3     DNAME VARCHAR2(20),
4     LOCATION VARCHAR2(20)
5   )
6   ORGANIZATION EXTERNAL

```

The 'Target' tab shows the corresponding foreign table definition:

```

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');

```

At the bottom, the 'REPAIRED' tab shows a note: "only superuser can change options of a file_fdw foreign table" with a line number reference.

Incompatible objects are identified

- I1. 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.

EDB migration portal

Projects Portal Wiki

case_no... 3 schemas Migrate to ...

Search 22 objects

HR Schema

FAILED	REPAIRED	PASSED	Total	Status	Percentage
5	4	3	12	HR	PASSED 58%

Sequences 1 100%
Types 2 100%
Tables 4 100%
Indexes 1 100%
Constraints 1 100%
Views 2 0%
Materialized View 1 0%

Common Failures

- > **schema_1900** (1)
- > **HR** (12)
 - > **1..3 Sequences** 1
 - > **Types** 2
 - > **Tables** 4
 - > **Indexes** 1
 - > **Constraints** 1
 - > **Views** 2
 - > **Materialized View** 1
- > **MIGRATIONTEST** (9)

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

Assessment result with errors

12. 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.

The screenshot shows the EDB migration portal interface. The top navigation bar includes the EDB logo, the title "migration portal", and tabs for "Projects" and "Portal Wiki". On the left, there's a sidebar titled "COMMON FAILURES" with sections for "OBJECTS" and "MIGRATION TESTS". The main content area displays a migration task for "case_no_v2 / HR / Tables / EMPLOYEES". It shows the "Source" code for creating the HR.EMPLOYEES table:

```
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),
```

The "Target" section shows the same create table statement with minor differences in constraint names and order:

```
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) );
```

The "OUTPUT" tab indicates the task was "REPAIRED" with a "Success" status.

The right side of the screen features a "Quick help" panel with a search bar and a list of links to various Oracle documentation topics, such as "MERGE statement", "Autonomous transaction", and "BYTE keyword".

Workaround or resolution for incompatible objects

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.



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.

Schema Assessment Report

edb POWER TO POSTGRES

Generated by:
Migration Portal v2.9.0
14th September 2020, 2:50:49 pm

Project: case_no_v2 **Source database:** Oracle 11g **Target database:** EDB Postgres Advanced Server 10

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.

1 Schemas	12 Total objects	5 Passed	2 Repaired	5 Failed	58% Compatibility	
HR	12 objects		5 failed 2 repaired 5 passed			
HR Object type			12 Total	5 Passed	2 Repaired	5 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

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.

14/09/2020

EDB Assessment Report-case_no_v2



Project:
case_no_v2

Source database:
Oracle 11g

Target database:
EDB Postgres Advanced Server 10

Generated by:

Migration Portal v2.9.0
14th September 2020, 2:51:34 pm

Description:

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.

1	12	5	2	5	58%
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

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 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



Schema migration home page

Migrating to an Existing On-Premise EDB Postgres Advanced Server Host

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 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](#)

Existing on-premise EDB Postgres Advanced Server home page

2. 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

⚠ 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% ⚠
----------------------------	-------------------	-------------------------	--	--

Select one or more schemas

All

schema_1900

HR

MIGRATIONTEST

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

Selecting schemas for migration

Note: If your schemas are not 100% compatible, a banner will be displayed as shown; complete 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

The Contact Us form

3. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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 12 (your assessed version).

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

Download the assessed schemas

4. Click on **Windows**:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | 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](#)

Selecting Windows operating system

5. 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
```

Note: You can also use **pgAdmin** instead.

The converted schemas are migrated to the target server.

Migrate schemas to existing on-premise EDB Postgres Advanced Server



Select Schemas — Download Schema — Import — Finish

Project Name
ID2019_1

Interface
JDBC

Source DB
Oracle 11g

Target DB
EDB Postgres Advanced Server 12

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

A successful migration

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](#)

Existing on-premise EDB Postgres Advanced Server home page

2. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

All
 HR

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

Selecting schemas for migration

3. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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 12 (your assessed version).

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

Download the assessed schemas

- Click on **Linux**:

Migrate schemas to existing on-premise EDB Postgres Advanced Server

Project Name: ID2019_1 Interface: JDBC Source DB: Oracle 11g Target DB: EDB Postgres Advanced Server 12 Compatibility: 100%

Import the selected schemas

Run the following command:

Windows **Linux**

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](#)

Selecting Linux operating system

- To import the schemas, invoke the following **edb-psql** client commands:

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

Note: You can optionally use the **pgAdmin** client for the import.

The converted schemas are migrated to the target server.

Migrate schemas to existing on-premise EDB Postgres Advanced Server

?

Select Schemas Download Schema Import 4 Finish

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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

A successful schema migration

Migrating to a New On-Premise EDB Postgres Advanced Server Installation

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.

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

New on-premise EDB Postgres Advanced Server home page

2. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

All
 HR

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

Selecting schemas for migration

3. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select the Operating System/Platform

Windows 64-bit 

Linux 64-bit 

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

Selecting Windows operating system

4. Download [Windows Installer](#).

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

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | Compatibility: 100%

Install EDB Postgres Advanced Server for Windows

 Download Windows installer 

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

Downloading Windows installer

- 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   Install EDB Postgres Advanced Server  Download Schema  Import >

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | Compatibility: 100%

Install EDB Postgres Advanced Server on Windows

 EDB Postgres™ Advanced Server Installation Guide for Windows 

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

6. Download the assessed schemas.

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

Project Name: ID2019_1 Interface: JDBC Source DB: Oracle 11g Target DB: EDB Postgres Advanced Server 12 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 12 (your assessed version).

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

Downloading the assessed file

7. 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

?

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

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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](#)

Importing schemas into EDB Postgres Advanced Server

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

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

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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](#)

Migrating schemas home page

2. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

All

HR

[Cancel](#)

[← Previous](#)

[→ Next](#)

Selecting schemas for migration

3. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select the Operating System/Platform

Windows 64-bit



Linux 64-bit



[Cancel](#)

[← Previous](#)

[→ Next](#)

Selecting Linux operating system

4. You can select one of the following options to install the EDB Postgres Advanced Server:

- Repository
- More options

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

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | Compatibility: 100%

Install EDB Postgres Advanced Server for Linux

Repository

More options

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

Selecting Linux repository

5. 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 Import >

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Install EDB Postgres Advanced Server on Linux

EDB Postgres™ Advanced Server Installation Guide for Linux

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

Selecting Linux installation guide

6. Download the assessed schemas:

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 Import → 7 Finish

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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 12 (your assessed version).

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

Downloading the assessed schemas

7. 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

Project Name: ID2019_1 Interface: JDBC Source DB: Oracle 11g Target DB: EDB Postgres Advanced Server 12 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](#)

Importing schemas into EDB Postgres Advanced Server

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

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | 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](#)

A *successful schema migration*

Migrating to the Cloud

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

2. 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

EDB Postgres Advanced Server on Cloud

3. Select one or more schemas to migrate to 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Select one or more schemas

All

HR

[Cancel](#)

[← Previous](#)

[→ Next](#)

Selecting schemas for migration

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

Migrate your schemas to the cloud

?

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

Project Name HumanResources	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------------	-------------------	-------------------------	--	-----------------------

Select the Cloud Platform

 IBM Cloud

AWS

Google Cloud

Microsoft Azure

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

Selecting cloud option for migration

5. 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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	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

Launching a cloud cluster

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

6. Enter the required connection details on the **Connect** page:

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_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Connect to IBM™ Cloud cluster

Target Database
ID2019_1

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
Password

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

Connecting to the cloud cluster

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

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

Migrate schemas to EDB Postgres Advanced Server on Cloud



✓ Select Schemas — ✓ Select Platform — ✓ Launch Cluster — 4 Connect — 5 Deploy — 6 Finish

Project Name ID2019_1	Interface JDBC	Source DB Oracle 11g	Target DB EDB Postgres Advanced Server 12	Compatibility 100%
--------------------------	-------------------	-------------------------	--	-----------------------

Target Database

ID2019_1

Host Name/Address Port

52.5.148.59 9999

Maintenance Database

dmp_exec

Note: Deployment will create a new database **ID2019_1** 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](#)

Verify the connection details

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

Migrate schemas to EDB Postgres Advanced Server on Cloud

Project Name: ID2019_1 | Interface: JDBC | Source DB: Oracle 11g | Target DB: EDB Postgres Advanced Server 12 | 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](#)

A successful migration

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](#).

3. 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](#).