
# Moving an Oracle Database into an EDB Ark™ Cluster

**September 13, 2016**

**Moving an Oracle Database into an EDB Ark Cluster**
**by EnterpriseDB® Corporation**


EnterpriseDB Corporation, 34 Crosby Drive, Suite 100, Bedford, MA 01730, USA
**T** +1 781 357 3390 **F** +1 978 589 5701 **E** info@enterprisedb.com **www**.enterprisedb.com

# Table of Contents

# 1 Introduction

You can use the EDB Ark Administrative console to host an EDB Postgres Advanced Server (Advanced Server) cluster that resides on an OpenStack cloud. This tutorial walks through the steps required to use Migration Toolkit to migrate an 'on-premise' Oracle database into an Advanced Server database hosted in an EDB Ark hosted cluster.

Migration Toolkit facilitates migration of database objects and data into an Advanced Server or PostgreSQL database. Migration Toolkit can assist you in migrating from:

- Oracle
- MySQL
- Sybase
- SQL Server

You can also use Migration Toolkit to migrate between PostgreSQL and Advanced Server.

This tutorial describes using either an online or offline migration process to migrate from an Oracle database to an instance that resides in an EDB Ark cluster. The tutorial assumes:

- You have Migration Toolkit running on a local workstation.
- Your existing database is compatible with features supported by Advanced Server.
- You have created a cluster into which you would like to move the database.

> By default, only port `9999` on the master server node of an EDB Ark cluster is open for client connections. Before connecting with `ssh` or `scp`, an OpenStack Administrative user must first modify the security group for the EDB Ark cluster, opening port `22` for connections. Before performing a migration, an Administrator must also open port `5444` for connections from the Migration Toolkit host.

This tutorial uses the term Postgres when referring to either PostgreSQL or EDB Postgres Advanced Server.

## 1.1 Pre-Requisites

**Installing Migration Toolkit**

You can use the following installers to add Migration Toolkit to an Advanced Server installation:

- Advanced Server graphical installer
- StackBuilder Plus or Application StackBuilder
- the `ppas-migrationtoolkit` RPM package

Before installing Migration Toolkit, you must install Java (1.7.0 or higher). For detailed installation instructions for Migration Toolkit, please refer to the *EDB Postgres Migration Guide*, available at:

http://www.enterprisedb.com/documentation/english

**Installing a Source-Specific Driver**

Before invoking Migration Toolkit to perform a migration, you must download and install a source-specific JDBC driver on your workstation; JDBC drivers are available from the EnterpriseDB website at:

http://www.enterprisedb.com/downloads/third-party-jdbc-drivers

After downloading the source-specific driver, move the driver file into the `JAVA_HOME/jre/lib/ext` directory.

**Specifying Connection Properties in the toolkit.properties File**

Migration Toolkit uses connection information specified in the `toolkit.properties` file (see Figure 1.1) to connect to the source and target databases. The `toolkit.properties` file resides in the `etc` directory, under the `edbmtk` directory on the workstation from which you will invoke Migration Toolkit.

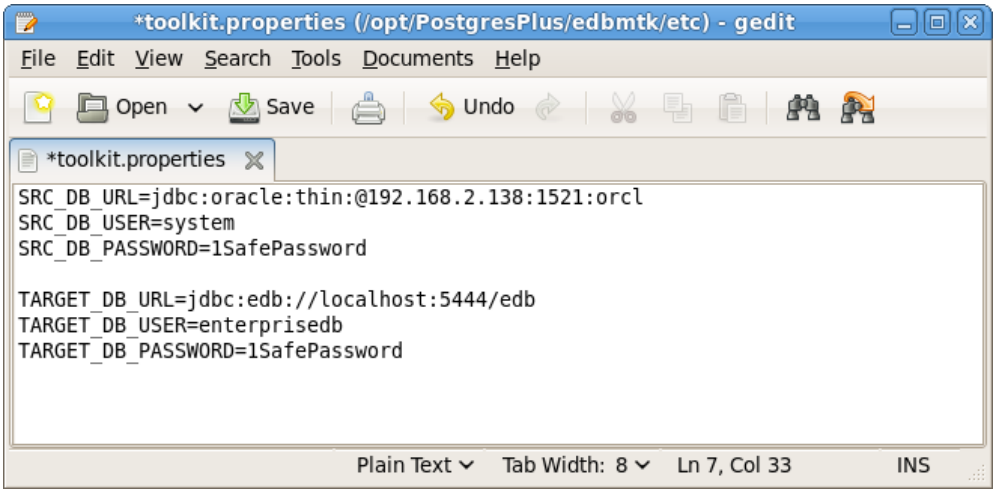


*Figure 1.1 - The toolkit.properties file.*

The first three parameters listed in the `toolkit.properties` file provide connection information for the source database:

- `SRC_DB_URL` contains a JDBC URL that specifies the type and location of the source database.
- `SRC_DB_USER` specifies a privileged user of the source database.
- `SRC_DB_PASSWORD` specifies the password of the source database user.

The last three parameters listed in the `toolkit.properties` file provide connection information for the target (your EDB Ark cluster):

- `TARGET_DB_URL` contains a JDBC URL of the target database. If you are performing an online migration through an `ssh` tunnel, specify that the port number of the tunnel in the URL. If you are performing an offline migration, specify the address and port number of your EDB Ark master node when forming the URL.
- `TARGET_DB_USER` specifies the name of a privileged target database user.
- `TARGET_DB_PASSWORD` specifies the password of the target database user.

**Migrating from an Oracle Source**

When migrating from an Oracle database, `SRC_DB_URL` takes the form of a JDBC URL. An Oracle URL can take one of two forms:

```
jdbc:oracle:thin:@host_name:port:database_id
```

or

```
jdbc:oracle:thin:@//host_name:port/database_id|service_name
```

Where:

`jdbc` identifies the connection protocol.

The sub-protocol identifies the migration target as an `oracle` database.

When migrating from an Oracle database, the URL should include a driver type of `thin`.

The *`host_name`* parameter specifies the name or IP address of the host where the Oracle server is running.

*`port`* specifies the port number that the Oracle database listener is monitoring.

*`database_id`* specifies the database `SID` of the Oracle database.

*`service_name`* is the name of the Oracle service.

For example, the following parameters describe an Oracle source database (named `mgmt`) that resides on host `192.168.2.88`, and is listening on port `1521`. Migration Toolkit will attempt to connect using a user name of `system`, and password of `password`.

```
SRC_DB_URL=jdbc:oracle:thin:@//192.168.2.88:1521/mgmt
SRC_DB_USER=system
SRC_DB_PASSWORD=password
```

**Using an ssh Tunnel to the EDB Ark Cluster**

Forwarding an unused port on your local workstation through an `ssh` tunnel to the target cluster allows you to take advantage of `ssh` compression and encryption during the migration process. To forward a port through an `ssh` tunnel, use the command:

```
ssh -i ssh_key -C -L local_port:127.0.0.1:target_port
user@host_name
```

Where:

*`ssh_key`* specifies the complete path and name of the cluster's `ssh` key file.

`-C` instructs `ssh` to use compression.

`-L` instructs `ssh` to forward the *`local_port`* to the remote host and *`target_port`*.

*`local_port`* specifies an otherwise unused port on your local workstation.

*`target_port`* specifies the port on which the target Postgres server is listening on the EDB Ark cluster. You can find the *`target_port`* in the `DBPORT` column of the `Details` panel of the `Clusters` tab in the EDB Ark console (see Figure 1.2).

*`user`* specifies the name of the connecting user.

*`host_name`* specifies the name on the EDB Ark cluster with which you wish to connect. You can find the *`host_name`* in the `DNSNAME` column of the `Details` panel of the `Clusters` tab in the EDB Ark console (see Figure 1.2).

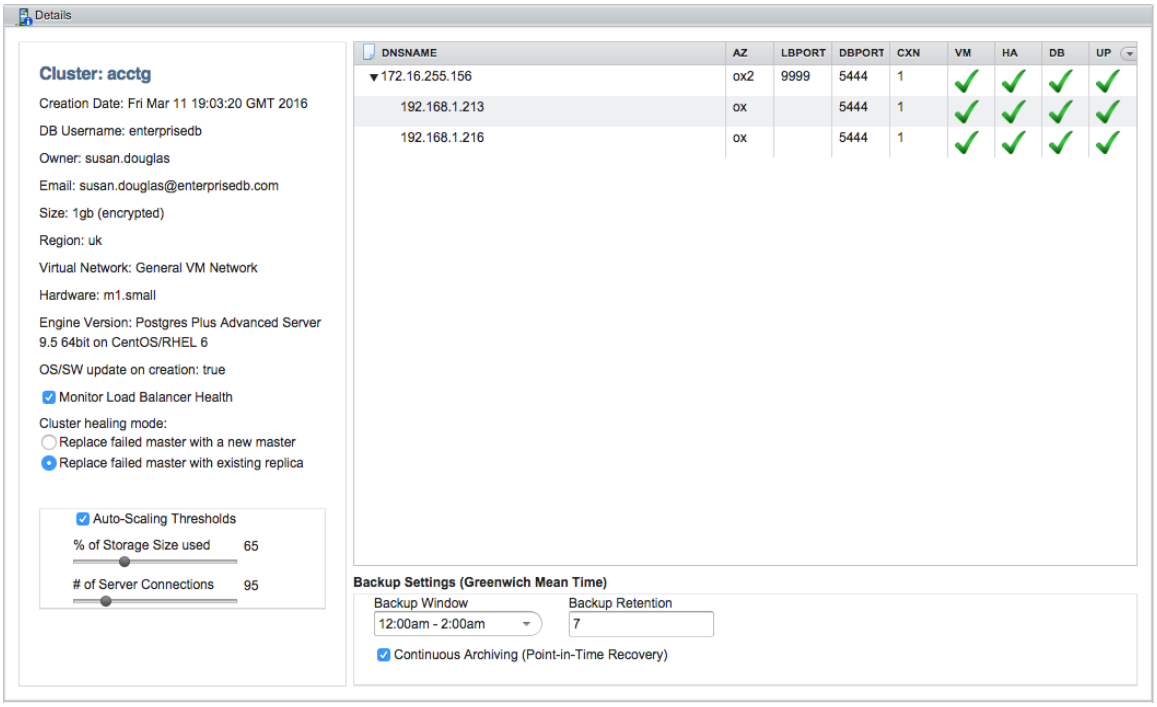


*Figure 1.2 - The Details panel of the EDB Ark Console.*

Please note that there is no performance benefit gained by using the load balancing ports during a migration; specify the port number shown in the `DBPORT` column.

# 2 Using Migration Toolkit

Migration Toolkit is a powerful command-line tool that offers granular control of the migration process. Migration Toolkit can migrate immediately and directly into a Postgres database (*online migration*), or you can also choose to generate scripts to use at a later time to recreate object definitions in a Postgres database (*offline migration*).

By default, Migration Toolkit performs an online migration, creating data and database objects directly into a Postgres database. If you include the `-offlineMigration` option when invoking Migration Toolkit, Migration Toolkit will generate SQL scripts to reproduce the migrated objects or data in a new database. You can alter the migrated objects by customizing the migration scripts generated by Migration Toolkit before you execute them, or schedule the actual migration for a time that best suits your work load.

If you are only migrating schema objects to a cluster, and use an `ssh` tunnel (with compression enabled), online migration of database object definitions may be a viable option. If you are migrating large amounts of data, network overhead may make an online migration prohibitively slow; Migration Toolkit's `-offlineMigration` option might provide a better migration path.

## *2.1 Performing an Online Migration*

You can use Migration Toolkit to perform an online migration. During an online migration, objects are migrated immediately and directly into a Postgres database on a target host. Before performing an online migration, you must:

- Install Migration Toolkit and your source-specific driver.
- Specify connection properties for the source and target databases in the `toolkit.properties` file.

The workstation that is hosting Migration Toolkit must be able to connect to the specified source and target databases during an online migration.

**Invoking Migration Toolkit to Perform an Online Migration.**

To migrate a complete schema, navigate to the directory that contains the Migration Toolkit executable, and invoke Migration Toolkit with the command:

```
$ ./runMTK.sh schema_name
```

Where:

> *schema_name* is the name of the schema within the source database that you wish to migrate. You must include at least one *schema_name*.

As the migration progresses, Migration Toolkit lists the objects migrated, and any error messages that result from the migration. When the migration completes, the `Migration Summary` will provide a count of the objects migrated (as shown in Figure 1.3).

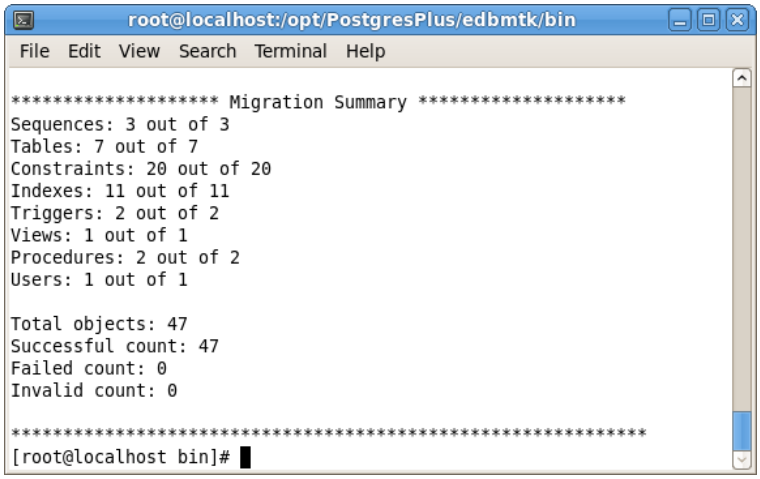


*Figure 1.3 - The migration summary.*

You can use a client application (such as Postgres Enterprise Manager) to confirm that Migration Toolkit has moved the specified objects to the EDB Ark cluster (see Figure 1.4).

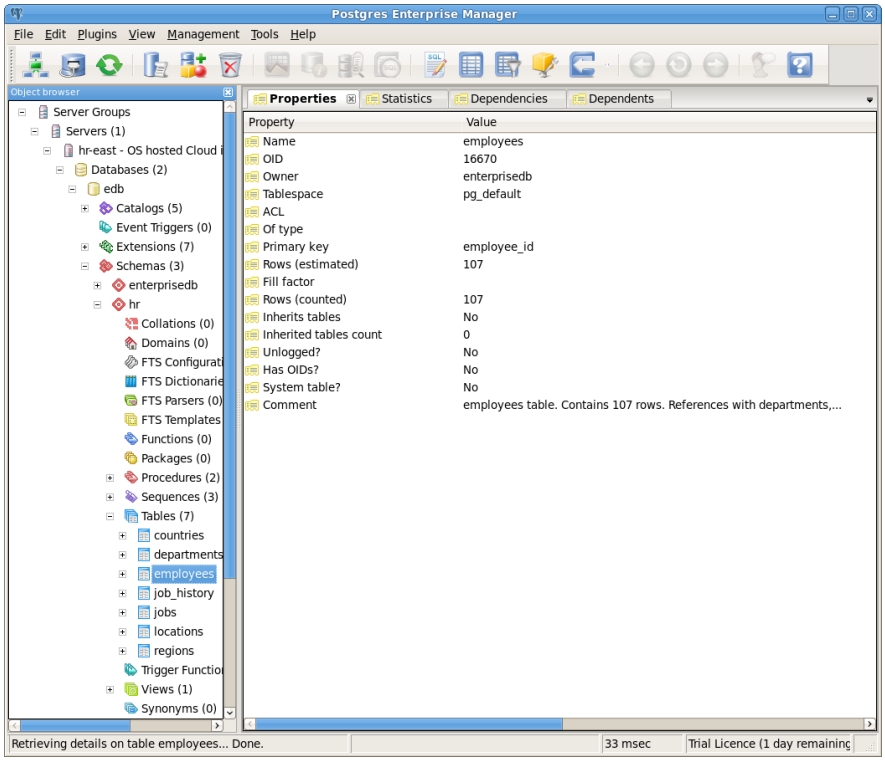


*Figure 1.4 - The migrated schema now resides in an EDB Ark Cluster.*

Unless specified in the command line, Migration Toolkit expects the source database to be Oracle and the target database to be Advanced Server. Include the `-sourcedbtype` and `-targetdbtype` keywords to specify an alternate source type or target type. For more information about migrating non-Oracle databases, please see the *EDB Postgres Migration Guide* at:

http://www.enterprisedb.com/documentation/english

## 2.2 Performing an Offline Migration

When you perform an offline migration, Migration Toolkit generates scripts that you can use to recreate object definitions and/or data in an EDB Ark cluster. Before performing a migration, you must:

- Install Migration Toolkit and your source-specific driver.
- Specify connection properties for the source and target databases in the `toolkit.properties` file.

Migration Toolkit will connect to the target database before performing an offline migration to confirm the type of the target database; your workstation must be able to connect to the specified target database when you invoke Migration Toolkit.

**Invoking Migration Toolkit to Perform an Offline Migration**

Navigate to the directory that contains the Migration Toolkit binary, and invoke Migration Toolkit, specifying the `-offlineMigration` keyword and the name of the source schema.

```
$ ./runMTK.sh -offlineMigration schema_name
```

By default, when you perform an offline migration that contains data, a separate file is created for each table. To create a single file that contains the data from multiple tables, specify the `-singleDataFile` and `-safemode` keywords:

```
$ ./runMTK.sh -offlineMigration -singleDataFile
-safeMode schema_name
```

You can also specify a file destination when invoking migration toolkit. In the example that follows, the migration files for the objects in the `hr` schema are written to the `/tmp/migration` directory:

```
$ ./runMTK.sh -offlineMigration /tmp/migration
-singledatafile -safemode hr
```

After executing the command shown in the example, the `/tmp/migration` directory contains:

```
mtk_hr_constraint_ddl.sql
mtk_hr_procedure_ddl.sql
mtk_hr_trigger_ddl.sql
mtk_hr_data.sql
mtk_hr_schema_ddl.sql
mtk_hr_view_ddl.sql
mtk_hr_ddl.sql
```

```
mtk_hr_sequence_ddl.sql
mtk_hr_index_ddl.sql
mtk_hr_table_ddl.sql
```

If you inspect the file contents, you will find that the files contain the SQL commands required to recreate the migrated objects on the EDB Ark cluster. The `mtk_hr_ddl.sql` file contains all of the ddl commands required to recreate all of the database objects, while the `mtk_hr_data.sql` file contains the commands that restore the data into the database.

**Copying the Migration Files to the Cloud**

After generating the migration scripts, move the scripts to the master node of the EDB Ark cluster (see Figure 1.5).

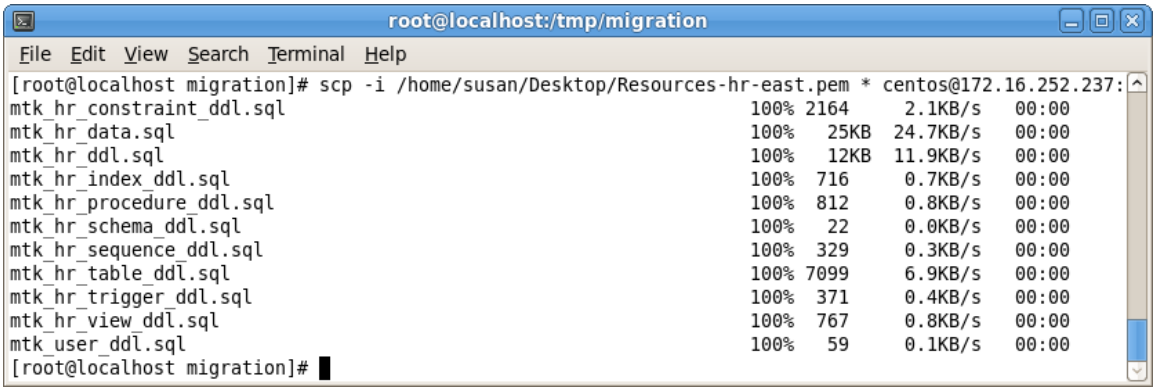


*Figure 1.5 - Moving the migration files to the EDB Ark host.*

You can the `scp` command to copy the files; before using `scp`, you must download the SSH key for the cluster, and modify the permissions associated with the key.


To download your private key, navigate to the `Clusters` tab of the EDB Ark console, and click the `Download SSH Key` icon located to the left of the window.

As the key downloads, the `Accessing Your Cluster Instance` popup opens. The popup displays information that will help you connect to your cluster, including the key name, the key file permissions required, and the user name and IP address you can use when connecting.

Before using your `ssh` key, open a terminal window, navigate to the location of the `ssh` key file, and modify the key permissions with the command:

```
$ chmod 0600 ssh_key
```

Where:

*ssh_key* specifies the complete path and name of the EDB Ark ssh private key file.

After setting the permissions for the key file, you can use the key and scp to copy files to the cloud. The syntax of the scp command is:

```
scp -i ssh_key file_name user@host_name:
```

Where:

*ssh_key* specifies the complete path and name of the cluster's ssh key file.

*file_name* specifies the name of the file you are copying to the Cloud. By default, each database object definition is saved in a separate file with a name derived from the schema name and object type in your home folder; use an * to copy all of the files in the directory with a single command.

*user* specifies the name of the connecting user. Use the name displayed on the Accessing Your Cluster Instance popup.

*host_name* specifies the IP address of the master node of the EDB Ark cluster; the host name is located in the DNSNAME column, on the Details panel of the Clusters tab in the EDB Ark console.

The colon (:) at the end of this command specifies that the file will be copied to the root directory on the cluster's primary node; please note that you can specify a file destination by adding a destination path after the colon.

**Executing the Migration Scripts**

Use ssh to connect to the master node of your EDB Ark cluster, specifying the location of the ssh key (on your workstation) in the command:

```
ssh -i /path/ssh_key user@host_name
```

Where:

*path* specifies the location of your EDB Ark ssh certificate on the system from which you are connecting.

*ssh_key* specifies the path and name of the EDB Ark ssh private key file.

*user* specifies the name of the connecting user.

*host_name* specifies the host name of the node to which you wish to connect.

Once connected, assume the identity of the database superuser:

```
sudo su enterprisedb
```

Then, use the `psql` client to connect to the server:

```
/opt/PostgresPlus/CloudDB/bin/psql -d db_name
```

Where:

*db_name* specifies the name of the database into which you wish to migrate.

Use the `\i` meta-command to invoke the migration script that creates the object you wish to migrate. For example, to create all of the database objects in the example schema, first invoke the `mtk_hr_ddl.sql` script:

```
hr=# \i ./mtk_hr_ddl.sql
```

Then, restore the data into the database with the command:

```
hr=# \i ./mtk_hr_data.sql
```

Please note: during the restoration process, you may encounter migration conflicts. If scripts are not invoked in an order that first creates the database objects, and then adds data, the server will be unable to restore data. Configuration differences on the source and target systems may also result in difficulties when recreating database objects that must be resolved manually. For more information about the migration process and Migration Toolkit options, please consult the *EDB Postgres Migration Guide*, available at:

http://www.enterprisedb.com/documentation/english

## 2.3 Confirming the Migration Results

After creating database objects and restoring data, you can connect to EDB Ark with a client application (such as EDB Postgres Enterprise Manager), and query the migrated database (as shown in Figure 1.6).

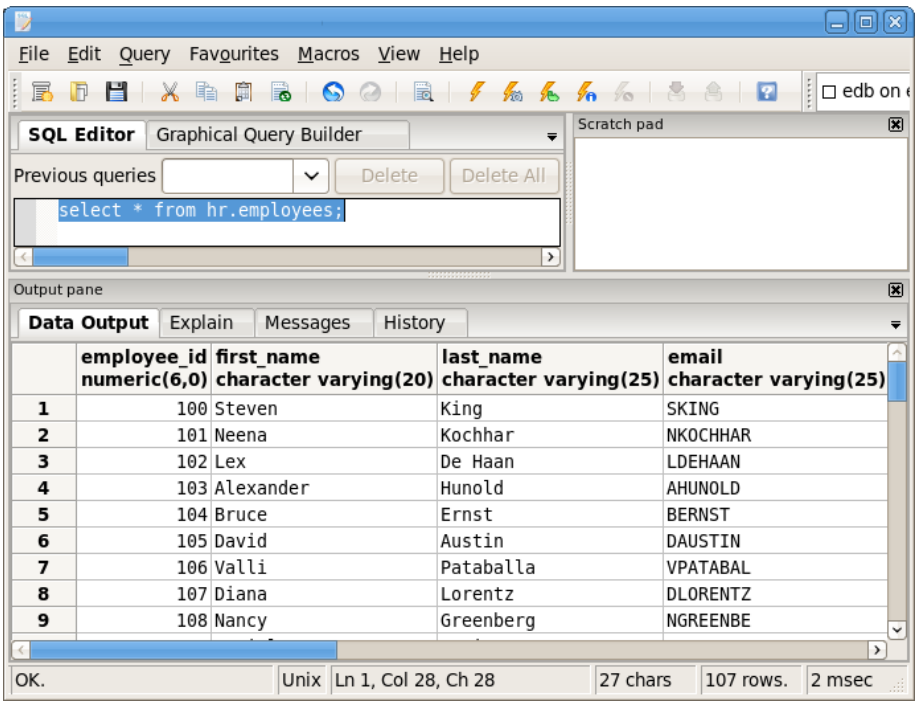


*Figure 1.6 - The migrated data now resides on an EDB Ark cluster*

EnterpriseDB also offers consulting services that will help you through the migration process; for more information, please visit the EnterpriseDB website at:

http://www.enterprisedb.com/solutions/oracle-migration-assessment