Hands-on Lab: Populating a Data Warehouse

Estimated time needed: 15 minutes

Objectives

In this lab you will:

- Create an instance of IBM DB2 on cloud
- Create credentials for external accessibility
- Create a db2cli dsn
- Verify a db2cli dsn
- Create the schema on production data warehouse
- Populate the production data warehouse
- Work with db2cli interactive command line

About Skills Network Cloud IDE

Skills Network Cloud IDE (based on Theia and Docker) provides an environment for hands on labs for course and project related labs. Theia is an open source IDE (Integrated Development Environment), that can be run on desktop or on the cloud. to complete this lab, we will be using the Cloud IDE based on Theia running in a Docker container.

Important Notice about this lab environment

Please be aware that sessions for this lab environment are not persistent. A new environment is created for you every time you connect to this lab. Any data you may have saved in an earlier session will get lost. To avoid losing your data, please plan to complete these labs in a single session.

Exercise 1 - Create an instance of IBM DB2 on cloud

We will be using the cloud instance of IBM DB2 as a production data warehouse in this lab.

If you do not have an instance of IBM DB2 on cloud, follow the instructions in this lab to create one.

Exercise 2 - Create service credentials

To access your IBM DB2 cloud instance from external programs, you need service credentials.

If you do not have service credentials, follow the instructions in this lab to create your service credentials.

Make a note of the following details:

- user
- password
- host
- port
- database name

You will need them later in this lab.

Exercise 3 - Create a db2cli dsn

You can access the IBM DB2 cloud instance using the web browser user interface.

Using the `db2cli` you can access your cloud IBM DB2 instance from the command line.

db2cli can be very helpful in automating data load tasks.

In this exercise we will be creating a dsn (data source name). A dsn in short is a simple name that refers to a data source.

Creating a dsn is two step process.

Step 1: We add the database, host, port and the security mode details. A sample commmand is given for your reference below:

```
db2cli writecfg add -database dbname -host hostname -port 50001 -parameter "SecurityTransportMode=SSL"
```

Step 2: We give a name to the data source we just created. This dsn name helps us to easily point to the IBM DB2 instance. A sample commmand is given for your reference below.

```
db2cli writecfg add -dsn dsn_name -database dbname -host hostname -port 50001
```

Run the commands below on the terminal to create a dsn named `production`. Make sure you use the database name, host and port details you noted in exercise 2.

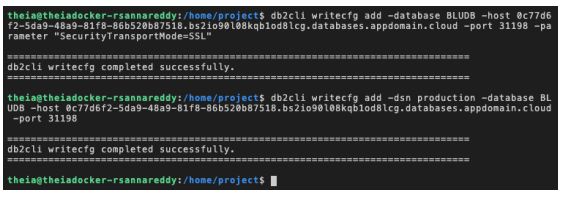
```
1
2
3
```

```
db2cli writecfg add -database BLUDB -host 0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud -port 31198 -parameter "SecurityTransportMode=SSL"

db2cli writecfg add -dsn production -database BLUDB -host 0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud -port 31198
```

Copied! Executed!

You should see an output similar to the one below.



Exercise 4 - Verify a db2cli dsn

Now that the dsn is created, we need to verify if it is working, before we go ahead and start using it.

The generic syntax for the command to verify the dsn is given below:

```
db2cli validate -dsn alias -connect -user userid -passwd password
```

Run the command below to verify the `production` dsn. Make sure you use your username and password that you noted in Exercise 2.

```
1. 1
```

```
1. db2cli validate -dsn production -connect -user jrg38634 -passwd SuWySBe5Y4MsYnh9
```

Copied! Executed!

You should see an output similar to the one below.

```
================================================================================
db2dsdriver.cfg validation for data source name "production":
================================================================================

[ Parameters used for the connection ]

Keywords                  Valid For     Value
--------------------------------------------------------------------------------
DATABASE                  CLI,.NET,ESQL BLUDB
HOSTNAME                  CLI,.NET,ESQL 0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.data
bases.appdomain.cloud
PORT                      CLI,.NET,ESQL 31198
SECURITYTRANSPORTMODE     CLI,.NET      SSL

================================================================================
Connection attempt for data source name "production":
================================================================================

[SUCCESS]

================================================================================
The validation is completed.
================================================================================
```

Your dsn is validated. You can now use it to access the IBM DB2 cloud instance.

Exercise 5 - Create the schema on production data warehouse

Step 1: Download the schema file.

Run the command below to download the schema file.

```
1. 1
```

```
1. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Populating%20a%20Data%20Warehouse/star-schema.sql
```

Copied! Executed!

Step 2: Create the schema.

Run the command below to create the schema on the production data warehouse. Make sure you use your username and password that you noted down in Exercise 2.

```
1. 1
```

```
1. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9 -inputsql star-schema.sql
```

Copied! Executed!

The command above tells db2cli to run the commands in the file `star-schema.sql` on the production data warehouse.

Exercise 6 - Populate the production data warehouse

Step 1: Download the data files.

Run the commands below to download the data files.

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
```

```
1. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Populating%20a%20Data%20Warehouse/DimCustomer.sql
2.
3. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Populating%20a%20Data%20Warehouse/DimMonth.sql
4.
5. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Populating%20a%20Data%20Warehouse/FactBilling.sql
6.
7. ls *.sql
```

Copied! Executed!

Step 2: Load the data in the data warehouse.

Run the commands below to load the data on to the production data warehouse. Make sure you use your username and password that you noted in Exercise 2.

```
1. 1
2. 2
3. 3
```

```
1. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9 -inputsql DimCustomer.sql
2. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9 -inputsql DimMonth.sql
3. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9 -inputsql FactBilling.sql
```

Copied! Executed!

Exercise 7 - Verify the data on the production data warehouse

Step 1: Download the verification sql file.

Run the command below to download the sql file to verify the data.

```
1. 1
```

```
1. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Populating%20a%20Data%20Warehouse/verify.sql
```

Copied! Executed!

Step 2: Verify the data in the data warehouse.

Run the command below to verify the data on the production data warehouse. Make sure you use your username and password that you noted down in Exercise 2.

```
1. 1
```

```
1. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9 -inputsql verify.sql
```

Copied! Executed!

You have successfully loaded the data, if you see an output similar to the one below.

```
theia@theiadocker-rsannareddy:/home/project$ db2cli execsql -dsn production -user jrg38634 -passwd SuW
ySBe5Y4MsYnh9 -inputsql verify.sql
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
> select count(*) as rowcount from DimMonth;
FetchAll:  Columns: 1
  ROWCOUNT
  132
FetchAll: 1 rows fetched.
> select count(*) as rowcount  from DimCustomer;
FetchAll:  Columns: 1
  ROWCOUNT
  1000
FetchAll: 1 rows fetched.
> select count(*) as rowcount from FactBilling;
FetchAll:  Columns: 1
  ROWCOUNT
  132000
FetchAll: 1 rows fetched.
>
```

Exercise 8 - Work with db2cli interactive command line

db2cli can also be used interactively.

Run the command below to open an interactive sql command shell to your production data warehouse. Make sure you use your username and password that you noted in Exercise 2.

```
1. 1
```

```
1. db2cli execsql -dsn production -user jrg38634 -passwd SuWySBe5Y4MsYnh9
```

Copied! Executed!

```
theia@theiadocker-rsannareddy:/home/project$ db2cli execsql -dsn production -user jrg38634 -passwd SuW
ySBe5Y4MsYnh9
IBM DATABASE 2 Interactive CLI Sample Program
(C) COPYRIGHT International Business Machines Corp. 1993,1996
All Rights Reserved
Licensed Materials - Property of IBM
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
>
```

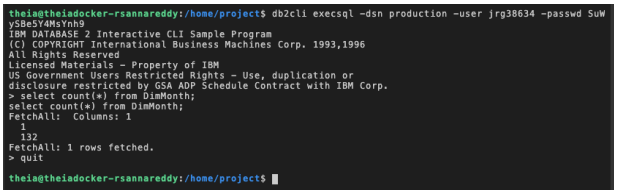
Run the command below on the db2cli.

```
1. 1
```

```
1. select count(*) from DimMonth;
```

Copied! Executed!

You should see an output as seen in the image below.



You are encouraged to run more sql queries. When done type `quit` to exit db2cli.

Practice exercises

1. Problem:

   Using the db2cli interactive shell, find the count of rows in the table FactBilling

▸ Click here for Hint
▾ Click here for Solution

At the db2cli prompt, run the following sql statement:

```
1. 1
```

```
1. select count(*) from FactBilling;
```

Copied!

2. Problem:

   Using the Cloud UI (not db2cli), create a simple MQT named avg_customer_bill with fields customerid and averagebillamount.

▸ Click here for Hint
▾ Click here for Solution

Access the UI for DB2, go to the Run SQL screen, in the editor, copy the following command:

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
```

```
1. CREATE TABLE avg_customer_bill (customerid, averagebillamount) AS
2. (select customerid, avg(billedamount)
3. from factbilling
4. group by customerid
5. )
6.      DATA INITIALLY DEFERRED
7.      REFRESH DEFERRED
8.      MAINTAINED BY SYSTEM;
```

Copied!

Clck the ‘Run All’ Button to run the statement. You should see status as ‘Success’ on the Result section.

3. Problem:

   Refresh the newly created MQT

▸ Click here for Hint
▾ Click here for Solution

At the db2cli prompt, run the following command:

```
1. 1
```

```
1. refresh table avg_customer_bill;
```

Copied!

4. Problem:

   Using the newly created MQT find the customers whose average billing is more than 11000.

▸ Click here for Hint
▾ Click here for Solution

At the db2cli prompt, run the following command:

```
1. 1
```

```
1. select * from avg_customer_bill where averagebillamount > 11000;
```

Copied!

Congratulations! You have successfully finished the Populating a Data Warehouse lab.

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-05-10	0.3	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-04	0.2	Benny Li	Republished
2021-09-29	0.1	Ramesh Sannareddy	Created initial version of the lab