!

THE UNIVERSITY OF WINNIPEG

PROFESSIONAL, APPLIED AND CONTINUING EDUCATION

Assignment 7 - Data Warehousing

by

Ayotunde Oyewole

Executive Summary

A remote repository was provided for this project. The repository contained sql query files in a zip folder. This project explored the process of building a data warehouse by running sql query files using a bash script. PostgreSQL was used as the database for the project. A star schema was used to create the tables and then they were loaded using multiple sql files. Finally, the whole process was verified using a verify.sql file.

1.0 Database Creation

A database was created on the postgresql server from terminal. The database was not created in the bash script because unlike other sql systems, postgresql does not have the "create database if not exist option.

```
ay@ay-virtual-machine:~$ createdb -h localhost -p 5432 -U postgres bill_DWH
Password:
```

2.0 File and Folder Creation

The project was completed in the assignment_7 directory. The screenshot below shows the creation and migration to that directory as well as the creation of the bash script used to run the sql commands. The file was modified from the beginning using chmod and then opened for editing with nano.

```
ay@ay-virtual-machine:~/Documents$ mkdir ./assignment_7
ay@ay-virtual-machine:~/Documents$ ls
'Assignment 3'   assignment_5       assignment_6       assignment_7
'assignment 4'   assignment_5.zip   assignment_6.zip   kafka_2.13-3.6.1
ay@ay-virtual-machine:~/Documents$ cd ./assignment_7
ay@ay-virtual-machine:~/Documents/assignment_7$ touch data_warehousing.sh
ay@ay-virtual-machine:~/Documents/assignment_7$ chmod +x data_warehousing.sh
ay@ay-virtual-machine:~/Documents/assignment_7$ nano data_warehousing.sh
ay@ay-virtual-machine:~/Documents/assignment_7$
```

3.0 Data_warehousing.sh

The bash script started with the shebang to indicate that it is a bash executable file. The remaining folders needed for the project were created from the script. That way if the script were executed on another system, it will create the folders it needs. A zip folder, log folder and unzip folder was created. A log file was also created using touch.

Wget was used to download the zip file from the provided URL (saved as a variable). and then 'tar' was used to unzip the file into a new folder.

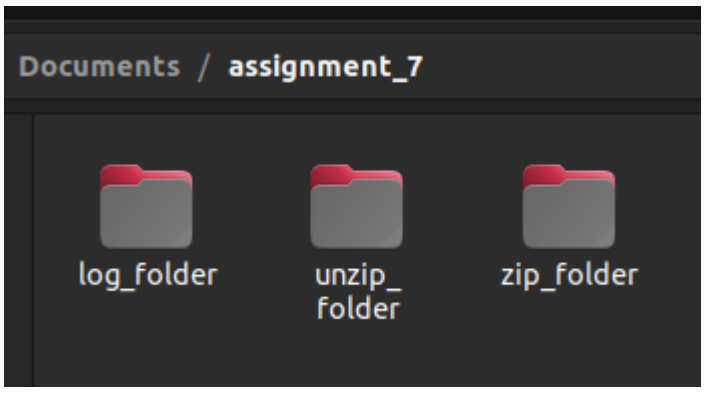
```
  GNU nano 6.2                          /home/ay/Documents/assignment_7/data_warehousing.sh *
#!/bin/bash

# Create log file in its own directory
mkdir -p ~/Documents/assignment_7/zip_folder ~/Documents/assignment_7/unzip_folder ~/Documents/assignment_7/log_folder

touch ~/Documents/assignment_7/log_folder/data_warehouse_log.log
date +"%c log file for assignment 7 - data warehouse created" >> ~/Documents/assignment_7/log_folder/data_warehouse_log.l>

# Download zip file from the url
url=https://elasticbeanstalk-us-east-2-340729127361.s3.us-east-2.amazonaws.com/billing-datawarehouse.tgz
wget $url -P ~/Documents/assignment_7/zip_folder
date +"%c downloaded billing-datawarehouse.tgz successfully." >> ~/Documents/assignment_7/log_folder/data_warehouse_log.l>

# Unzip file into a seperate folder
tar -xvzf ~/Documents/assignment_7/zip_folder/billing-datawarehouse.tgz -C ~/Documents/assignment_7/unzip_folder >> ~/Doc>
date +"%c Unzipped billing-datawarehouse.tgz successfully. Files saved to unzip_folder" >> ~/Documents/assignment_7/log_f>
```



4.0 Sql Scripts

The five sql scripts in the unzip_folder were executed one after the other. The output of all the commands were passed directly into the log file. That way, even when the terminal is closed, the log file could be accessed and the output of the script reviewed.

```
# Execute star-schema.sql
psql "host=localhost port=5432 dbname=bill_DWH user=postgres password=postgres" -a -f ~/Documents/assignment_7/unzip_fold
date +"%c Star Schema executed. All tables created successfully " >> ~/Documents/assignment_7/log_folder/data_warehouse_l


# Execute DimCustomer.sql
psql "host=localhost port=5432 dbname=bill_DWH user=postgres password=postgres" -a -f ~/Documents/assignment_7/unzip_fold
date +"%c DimCustomer.sql executed successfully" >> ~/Documents/assignment_7/log_folder/data_warehouse_log.log


# Execute DimMonth.sql
psql "host=localhost port=5432 dbname=bill_DWH user=postgres password=postgres" -a -f ~/Documents/assignment_7/unzip_fold
date +"%c DimMonth.sql executed successfully" >> ~/Documents/assignment_7/log_folder/data_warehouse_log.log

# Execute FactBilling.sql
psql "host=localhost port=5432 dbname=bill_DWH user=postgres password=postgres" -a -f ~/Documents/assignment_7/unzip_fold
date +"%c FactBilling.sql executed successfully" >>~/Documents/assignment_7/log_folder/data_warehouse_log.log


# Execute verify.sql
psql "host=localhost port=5432 dbname=bill_DWH user=postgres password=postgres" -a -f ~/Documents/assignment_7/unzip_fold
date +"%c Verify.sql executed successfully" >> ~/Documents/assignment_7/log_folder/data_warehouse_log.log
```

5.0 Executing the Script

The image below shows the script execution. The output from the wget command is displayed. All other logs are in the log file.

```
ay@ay-virtual-machine:~/Documents/assignment_7$ ./data_warehousing.sh
--2024-02-09 14:00:04--  https://elasticbeanstalk-us-east-2-340729127361.s3.us-east-2.amazonaws.com/billing-datawarehouse
tgz
Resolving elasticbeanstalk-us-east-2-340729127361.s3.us-east-2.amazonaws.com (elasticbeanstalk-us-east-2-340729127361.s3.
s-east-2.amazonaws.com)... 52.219.108.122, 52.219.97.202, 3.5.130.118, ...
Connecting to elasticbeanstalk-us-east-2-340729127361.s3.us-east-2.amazonaws.com (elasticbeanstalk-us-east-2-340729127361
s3.us-east-2.amazonaws.com)|52.219.108.122|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 944578 (922K) [application/x-compressed-tar]
Saving to: ‘/home/ay/Documents/assignment_7/zip_folder/billing-datawarehouse.tgz.1’

billing-datawarehouse.tgz.1                         100%[=================================================================
===================================================================================>] 922.44K  4.40MB/s    in 0.2s


2024-02-09 14:00:05 (4.40 MB/s) - ‘/home/ay/Documents/assignment_7/zip_folder/billing-datawarehouse.tgz.1’ saved [944578/9
44578]

ay@ay-virtual-machine:~/Documents/assignment_7$
```

6.0 Output of Verify.sql

From the log file, the output of the verify.sql is as shown below.

```
134423 \echo "Checking row in DimMonth Table"
134424 "Checking row in DimMonth Table"
134425 select count(*) from "DimMonth";
134426  count
134427 -------
134428    132
134429 (1 row)
134430
134431 \echo "Checking row in DimCustomer Table"
134432 "Checking row in DimCustomer Table"
134433 select count(*) from "DimCustomer";
134434  count
134435 -------
134436   1000
134437 (1 row)
134438
134439 \echo "Checking row in FactBilling Table"
134440 "Checking row in FactBilling Table"
134441 select count(*) from "FactBilling";
134442  count
134443 --------
134444  132000
134445 (1 row)
134446
134447 Fri 09 Feb 2024 02:00:10 PM Verify.sql executed successfully
```

Additionally, some sample logs from the log file are as shown below

```
 1 Fri 09 Feb 2024 01:54:49 PM log file for assignment 7 - data warehouse created
 2 Fri 09 Feb 2024 01:54:49 PM downloaded billing-datawarehouse.tgz successfully.
 3 Fri 09 Feb 2024 01:54:49 PM Unzipped billing-datawarehouse.tgz successfully. Files saved to unzip_folder
 4 Fri 09 Feb 2024 01:54:49 PM Star Schema executed. All tables created successfully
 5 Fri 09 Feb 2024 01:54:49 PM DimCustomer.sql executed successfully
 6 Fri 09 Feb 2024 01:54:49 PM DimMonth.sql executed successfully
 7 Fri 09 Feb 2024 01:54:49 PM FactBilling.sql executed successfully
 8 Fri 09 Feb 2024 01:54:49 PM Verify.sql executed successfully
 9 Fri 09 Feb 2024 01:55:53 PM log file for assignment 7 - data warehouse created
10 Fri 09 Feb 2024 01:55:54 PM downloaded billing-datawarehouse.tgz successfully.
11 DimCustomer.sql
12 DimMonth.sql
13 FactBilling.sql
14 star-schema.sql
15 verify.sql
16 Fri 09 Feb 2024 01:55:54 PM Unzipped billing-datawarehouse.tgz successfully. Files saved to unzip_folder
17 -- This script was generated by a beta version of the ERD tool in pgAdmin 4.
18 -- Please log an issue at https://redmine.postgresql.org/projects/pgadmin4/issues/new if you find any bugs, includi
   reproduction steps.
19 BEGIN;
20 BEGIN
21 CREATE TABLE public."FactBilling"
22 (
23     billid serial,
24     customerid integer NOT NULL,
25     monthid integer NOT NULL,
26     billedamount integer NOT NULL,
27     PRIMARY KEY (billid)
28 );
29 CREATE TABLE public."DimMonth"
30 (
```

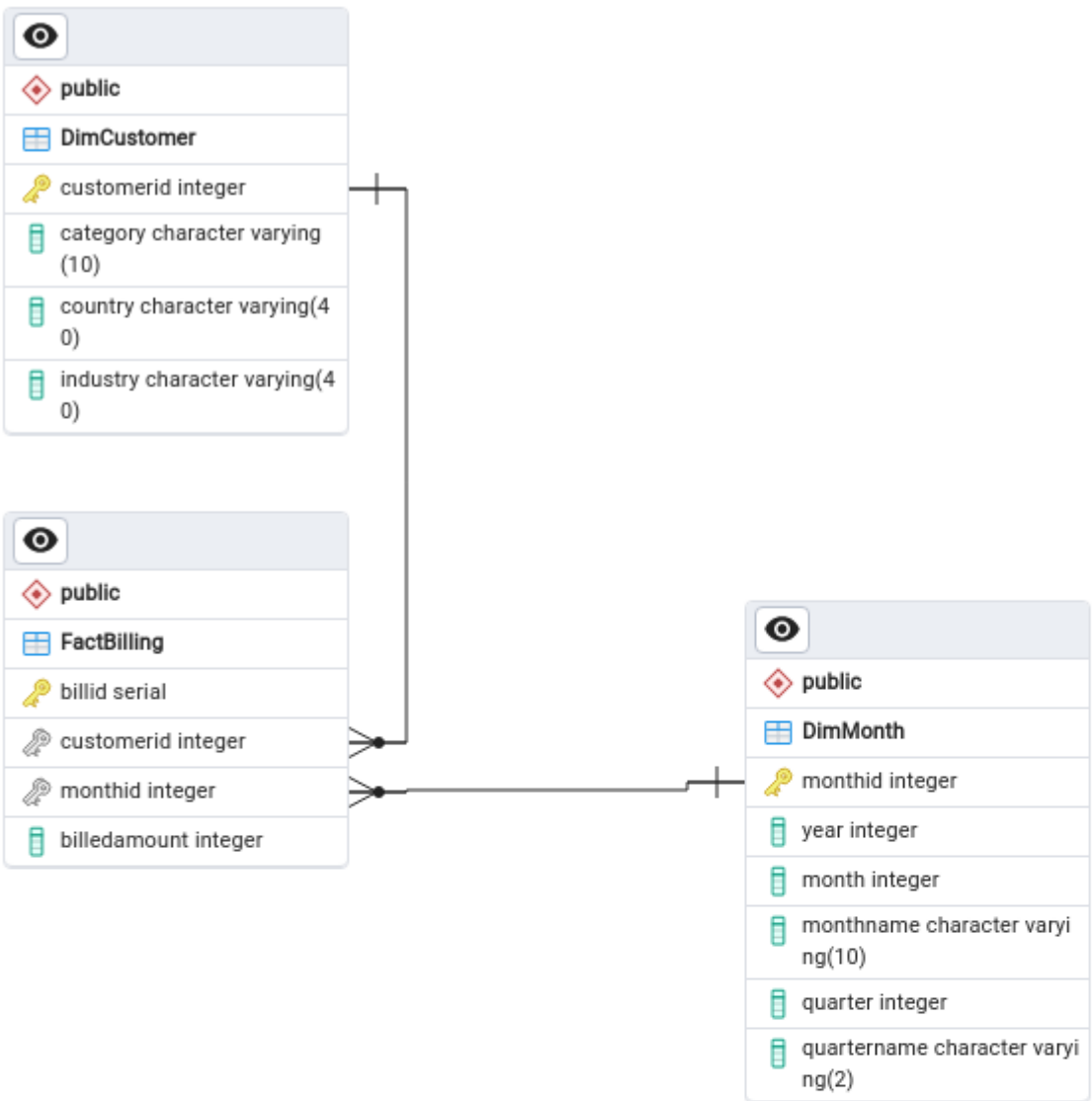
```
1057 (8,'Individual','Brazil','Engineering'),
1058 (919,'Company','Indonesia','Business Development'),
1059 (890,'Company','South Africa','Training');
1060 Fri 09 Feb 2024 01:55:55 PM DimCustomer.sql executed successfully
1061 INSERT INTO "DimMonth"
1062 (monthid,year,month,monthname,quarter,quartername)
1063 VALUES
1064 (20091, 2009, 1, 'Janauary', 1, 'Q1'),
1065 (200910, 2009, 10, 'October', 4, 'Q4'),
1066 (200911, 2009, 11, 'November', 4, 'Q4'),
1067 (200912, 2009, 12, 'December', 4, 'Q4'),
1068 (20092, 2009, 2, 'February', 1, 'Q1'),
1069 (20093, 2009, 3, 'March', 1, 'Q1'),
1070 (20094, 2009, 4, 'April', 2, 'Q2'),
1071 (20095, 2009, 5, 'May', 2, 'Q2'),
1072 (20096, 2009, 6, 'June', 2, 'Q2'),
1073 (20097, 2009, 7, 'July', 3, 'Q3'),
1074 (20098, 2009, 8, 'August', 3, 'Q3'),
1075 (20099, 2009, 9, 'September', 3, 'Q3'),
1076 (20101, 2010, 1, 'Janauary', 1, 'Q1'),
1077 (201010, 2010, 10, 'October', 4, 'Q4'),
1078 (201011, 2010, 11, 'November', 4, 'Q4'),
1079 (201012, 2010, 12, 'December', 4, 'Q4'),
1080 (20102, 2010, 2, 'February', 1, 'Q1'),
1081 (20103, 2010, 3, 'March', 1, 'Q1'),
1082 (20104, 2010, 4, 'April', 2, 'Q2'),
1083 (20105, 2010, 5, 'May', 2, 'Q2'),
1084 (20106, 2010, 6, 'June', 2, 'Q2'),
1085 (20107, 2010, 7, 'July', 3, 'Q3'),
1086 (20108, 2010, 8, 'August', 3, 'Q3'),
1087 (20109, 2010, 9, 'September', 3, 'Q3'),
```

The full log file is attached to this assignment submission.

7.0 Verification from Database

The database was reviewed to be sure all tables were created and loaded successfully. The ERD for the database as also retrieved and shown below.

ERD for Database



DimCustomer Table

```
27  select * from "DimCustomer" limit 10;
```

Data Output | Messages | Notifications

	customerid [PK] integer	category character varying (10)	country character varying (40)	industry character varying (40)
1	1	Individual	Indonesia	Engineering
2	614	Individual	United States	Product Management
3	615	Individual	China	Services
4	616	Individual	Russia	Accounting
5	617	Individual	Chile	Business Development
6	618	Individual	Nicaragua	Human Resources
7	41	Company	Brazil	Marketing
8	619	Individual	Russia	Business Development
9	620	Individual	China	Business Development
10	956	Individual	Peru	Research and Development

DimMonth Table

```
27  select * from "DimMonth" limit 10;
```

Data Output | Messages | Notifications

	monthid [PK] integer	year integer	month integer	monthname character varying (10)	quarter integer	quartername character varying (2)
1	20091	2009	1	Janauary	1	Q1
2	200910	2009	10	October	4	Q4
3	200911	2009	11	November	4	Q4
4	200912	2009	12	December	4	Q4
5	20092	2009	2	February	1	Q1
6	20093	2009	3	March	1	Q1
7	20094	2009	4	April	2	Q2
8	20095	2009	5	May	2	Q2
9	20096	2009	6	June	2	Q2
10	20097	2009	7	July	3	Q3

FactBilling Table

```
27  select * from "FactBilling" limit 10;
```

Data Output | Messages | Notifications

	billid [PK] integer	customerid integer	monthid integer	billedamount integer
1	1	1	20091	5060
2	2	614	20091	9638
3	3	615	20091	11573
4	4	616	20091	18697
5	5	617	20091	944
6	6	618	20091	3539
7	7	41	20091	6591
8	8	619	20091	16061
9	9	620	20091	1250
10	10	956	20091	15105

Conclusion

A data warehouse was created from sql query files. The files were run with bash script and then tested from pgadmin to verify their correct execution. All the files executed successfully and all the tables were created and loaded as appropriate. The logs for the whole process is provided along with this submission.