Hands-on Lab: Working with Facts and Dimension Tables

Estimated time needed: 30 minutes

Purpose of the lab:

The lab is designed to guide you through the process of designing a data warehouse for a cloud service provider. It focuses on using billing data provided in a CSV file to create a star schema, including the design of fact and dimension tables. This schema will support complex queries related to billing, such as average billing per customer, billing by country, industry, and category, as well as trends over time.

Benefits of Learning the Lab:

By completing this lab, you will acquire practical skills in organizing and analyzing large datasets using data warehousing techniques. These skills are essential for making informed business decisions, optimizing data retrieval, and enhancing the understanding of data relationships. This knowledge is particularly beneficial in real-world scenarios, such as analyzing cloud billing data, where it can lead to more efficient data management and insightful analyses.

Objectives

In this lab you will:

- Study the schema of the given csv file
- Design the fact tables
- Design the dimension tables
- Create a star schema using the fact and dimension tables

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 - Study the schema of the given csv file

In this lab, we will design a data warehouse for a cloud service provider.

The cloud service provider has given us their billing data in the csv file `cloud-billing-dataset.csv`. This file contains the billing data for the past decade.

Here are the field wise details of the billing data.

Field Name	Details
customerid	Id of the customer
category	Category of the customer. Example: Individual or Company
country	Country of the customer
industry	Which domain/industry the customer belongs to. Example: Legal, Engineering
month	The billed month, stored as YYYY-MM. Example: 2009-01 refers to the month January in the year 2009
billedamount	Amount charged by the cloud services provided for that month in USD

We need to design a data warehouse that can support the queries listed below:

- average billing per customer
- billing by country
- top 10 customers
- top 10 countries
- billing by industry
- billing by category
- billing by year
- billing by month
- billing by quarter
- average billing per industry per month
- average billing per industry per quarter
- average billing per country per quarter
- average billing per country per industry per quarter

Here are five rows picked at random from the csv file.

customerid	category	country	industry	month	billedamount
1	Individual	Indonesia	Engineering	2009-1	5060
614	Individual	United States	Product Management	2009-1	9638
615	Individual	China	Services	2009-1	11573
616	Individual	Russia	Accounting	2009-1	18697
617	Individual	Chile	Business Development	2009-1	944

Exercise 2 - Design the fact tables

The fact in this data is the bill which is generated monthly.

The fields `customerid` and `billedamount` are the important fields in the fact table.

We also need a way to identify the additional customer information, other than the id, and date information. So we need fields that refer to the customer and date information in other tables.

The final fact table for the bill would look like this:

Field Name	Details
billid	Primary key - Unique identifier for every bill
customerid	Foreign Key - Id of the customer
monthid	Foreign Key - Id of the month. We can resolve the billed month info using this
billedamount	Amount charged by the cloud services provided for that month in USD

Exercise 3 - Design the dimension tables

There are two dimensions to our fact(monthly bill).

1. Customer information
2. Date information

Let us organize all the fields that give information about the customer into a dimension table.

Field Name	Details
customerid	Primary Key - Id of the customer
category	Category of the customer. Example: Individual or Company
country	Country of the customer
industry	Which domain/industry the customer belongs to. Example: Legal, Engineering

Let us organize or derive all the fields that give information about the date of the bill.

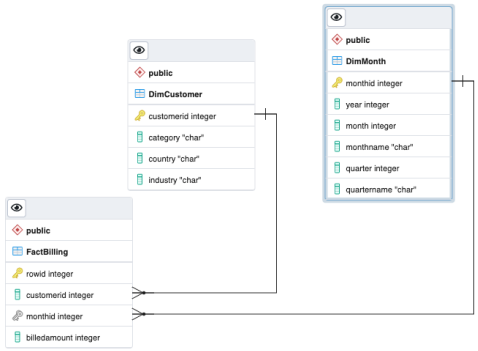
Field Name	Details
monthid	Primary Key - Id of the month
year	Year derived from the month field of the original data. Example: 2010
month	Month number derived from the month field of the original data. Example: 1, 2, 3
monthname	Month name derived from the month field of the original data. Example: March
quarter	Quarter number derived from the month field of the original data. Example: 1, 2, 3, 4
quartername	Quarter name derived from the month field of the original data. Example: Q1, Q2, Q3, Q4

Exercise 4 - Create a star schema using the fact and dimension tables

Based on the previous two exercises, we have now arrived at 3 tables, we can name them as in the table below.

Table Name	Type	Details
FactBilling	Fact	This table contains the billing amount, and the foreign keys to customer and month data
DimCustomer	Dimension	This table contains all the information related the customer
DimMonth	Dimension	This table contains all the information related the month of billing

When we arrange the above tables in Star Schema style, we get a table strucutre that looks likes the one in the image below.

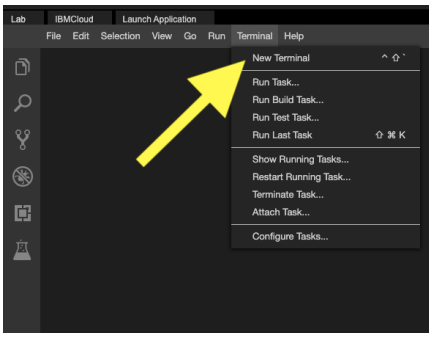


The image shows the fact and dimension tables along with the relaionships between them.

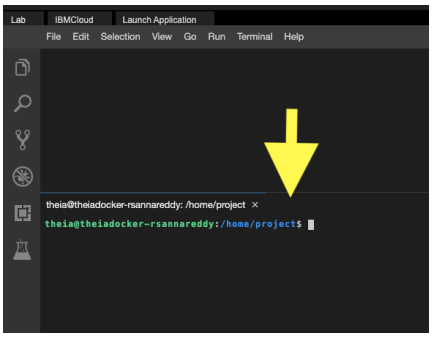
Exercise 5 - Create the schema on the data warehouse

Step 1: Start the postgresql server.

Open a new terminal, by clicking on the menu bar and selecting **Terminal->New Terminal**, as shown in the image below.



This will open a new terminal at the bottom of the screen.



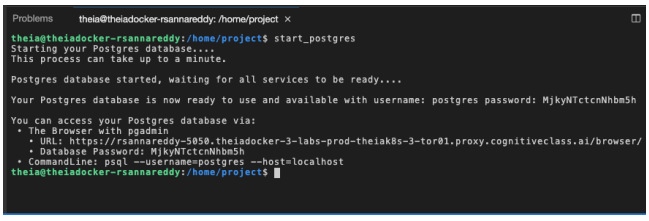
Run the commands below on the newly opened terminal. (You can copy the code by clicking on the little copy button on the bottom right of the codeblock below and then paste it, wherever you wish.)

Start the PostgreSQL server, by running the command below:

```
1
start_postgres
```

Copied! Executed!

You should see an output similar to the one below.



Step 2: Create the database on the data warehouse.

Using the createdb command of the PostgreSQL server, we can directly create the database from the terminal.

Run the command below to create a database named billingDW.

```
1. 1
```

```
1. createdb -h localhost -U postgres -p 5432 billingDW
```

Copied! Executed!

In the above command

- -h mentions that the database server is running on the localhost
- -U mentions that we are using the user name postgres to log into the database
- -p mentions that the database server is running on port number 5432

You should see an output like this.

```
theia@theiadocker-rsannareddy:/home/project$ createdb -h localhost -U postgres -p 5432 billingDW
theia@theiadocker-rsannareddy:/home/project$
```

Step 3: Download the schema .sql file.

The commands to create the schema are available in the file below.

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Working%20with%20Facts%20and%20Dimension%20Tables/star-schema.sql

Download the file by running the command below.

```
1. 1
```

```
1. wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Working%20with%20Facts%20and%20Dimension%20Tables/star-schema.sql
```

Copied! Executed!

Step 4: Create the schema

Run the command below to create the schema in the under `billingDW` database.

```
1. 1
```

```
1. psql  -h localhost -U postgres -p 5432 billingDW < star-schema.sql
```

Copied! Executed!

You should see an ouput similar to the one below.

```
theia@theiadocker-rsannareddy:/home/project$ psql  -h localhost -U postgres -p 5432 billingDW < star-schema.sql
BEGIN
CREATE TABLE
CREATE TABLE
CREATE TABLE
ALTER TABLE
ALTER TABLE
COMMIT
theia@theiadocker-rsannareddy:/home/project$
```

Practice exercises

In this practice exercise, you will analyze the below csv file, which contains data about the daily sales at different stores of an international fashion retailer.

storeid	country	city	date	totalsales
1	Japan	Tokyo	01 February 2020	20300.50
2	UK	London	01 February 2020	34000.20
3	USA	New York	01 February 2020	28900.00
4	USA	Chicago	01 February 2020	27690.00
5	France	Paris	01 February 2020	12090.00

1. Problem:

   Design the schema for the dimension table DimStore.

▾ Click here for Hint

Make sure that this table contains the country and city of the store.

▾ Click here for Solution

Field Name	Details
storeid	Primary key - Unique identifier for every store
city	City where the store is located.
country	Country where the store is located.

2. Problem:

   Design the schema for the dimension table DimDate.

▾ Click here for Hint

Here the customer needs reports to the granularity of a day. Make sure that you include the day, weekday and weekdayname also in this table.

▾ Click here for Solution

Field Name	Details
dateid	Primary Key - Id of the date
day	Day derived from the date field of the original data. Example: 13, 19
weekday	Weekday derived from the date field of the original data. Example: 1, 2, 3, 4, 5, 6, 7. 1 for sunday, 7 for saturday
weekdayname	Weekday name derived from the date field of the original data. Example: Sunday, Monday
year	Year derived from the date field of the original data. Example: 2010
month	Month number derived from the date field of the original data. Example: 1, 2, 3
monthname	Month name derived from the date field of the original data. Example: March
quarter	Quarter number derived from the date field of the original data. Example: 1, 2, 3, 4
quartername	Quarter name derived from the date field of the original data. Example: Q1, Q2, Q3, Q4

3. Problem:

   Design the schema for the fact table FactSales.

▾ Click here for Hint

Make sure that the totalsales field is captured and there is a way to refer to the store and the date. Also add a rowid to uniquely identify every row.

▾ Click here for Solution

Field Name	Details
rowid	Primary key - Unique identifier for every row
storeid	Foreign Key - Id of the store
dateid	Foreign Key - Id of the date
totalsales	Total sales

Congratulations! You have successfully finished this lab.

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-09-22	0.1	Ramesh Sannareddy	Created initial version of the lab