Final Assignment: Data Warehouse Fundamentals

Estimated time needed: 90 minutes

This comprehensive lab is designed to provide hands-on experience in designing, implementing, and querying a data warehouse using PostgreSQL. It simulates a real-world scenario where you, as a data engineer, assist a waste management company in Brazil in managing and analyzing their solid waste collection data. The lab involves multiple stages, from designing and creating a star schema for the data warehouse, loading data, writing complex SQL queries for data aggregation, and creating materialized view for query optimization.

What you'll learn:

The lab offers a multitude of learning benefits, particularly for those seeking to enhance their data engineering and business intelligence skills:

- **Practical experience in data warehouse design**: It provides hands-on experience in designing and implementing a star schema, which is crucial for any data warehousing project.
- **SQL Query writing skills**: Enhances your ability to write complex SQL queries, including grouping sets, rollups, and cubes, essential for data analysis and reporting.
- **Data loading and transformation**: Offers practice in data loading and transformation, an essential skill for managing data warehouses.
- **Real-world scenario applications**: The scenario-based approach of the lab ensures that the skills acquired are relevant and applicable in real-world data warehousing and business intelligence projects.
- **Career advancement**: These skills are in high demand in the fields of data engineering, business intelligence, and analytics, contributing significantly to professional growth and opportunities.

This lab serves as a comprehensive guide for anyone aiming to strengthen their expertise in data warehousing and business intelligence, providing practical skills that are directly applicable in professional environments.

About the SN Labs Cloud IDE

This Skills Network Labs Cloud IDE provides a hands-on environment for course and project-related labs. It utilizes Theia, an open-source IDE (Integrated Development Environment) platform that can run on a desktop or the cloud. To complete this lab, you will be using the Cloud IDE based on Theia and PostgreSQL.

Single Session Exercise

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.

Software used in the lab

In this lab, you will use PostgreSQL Database. PostgreSQL is a Relational Database Management System (RDBMS) designed to store, manipulate, and retrieve data efficiently.

Scenario

You are a data engineer hired by a solid waste management company. It collects and recycles solid waste across major cities in the country of Brazil. They operate hundreds of trucks of different types to collect and transport solid waste. The company would like to create a data warehouse so that it can create reports like:

- Total waste collected per year per city
- Total waste collected per month per city
- Total waste collected per quarter per city
- Total waste collected per year per trucktype
- Total waste collected per trucktype per city
- Total waste collected per trucktype per station per city

You will use your data warehousing skills to design and implement a data warehouse for the company.

Learning objectives

After completing this lab, you will be able to:

- Design a data warehouse.
- Load data into the data warehouse.
- Create a materialized view.

Note: Screenshots

Throughout this lab, you will be prompted to take screenshots and save them on your own device. These screenshots will be uploaded for peer review in the next section of the course. You can use various free screengrabbing tools or your operating system's shortcut keys (**Alt+PrintScreen in Windows, Command+Shift+5 on Mac, Shift+Ctrl+Show windows on Chromebook**) to capture the required screenshots. The screenshots can be either jpg or png.

About the data set

The data set you would be using in this assignment is not a real-life data set. It was programmatically created for this assignment purpose.

Prerequisites

You need to use PostgreSQL Database to proceed with the assignment.

This lab will guide you to understand how to create tables and load data in PostgreSQL using pgAdmin.

Exercise 1: Design a data warehouse

The solid waste management company has provided you the sample data they want to collect.

Trip number	Waste Type	Waste Collected in tons	Collection Zone	City	Date
1	Dry	45.23	South	Sao Paulo	23-Jan-20
2	Wet	43.12	Central	Rio de Janeiro	24-Jan-20
3	Electronic	40.19	South	Sao Paulo	23-Jan-20
4	Plastic	34.87	West	Rio de Janeiro	24-Jan-20
5	Wet	45.34	West	Rio de Janeiro	23-Jan-20

You will start your project by designing a Star Schema warehouse by identifying the columns for the various dimensions and fact tables in the schema.

Task 1: Design the dimension table MyDimDate

Write down the fields in the MyDimDate table in any text editor, one field per line. The company is looking at a granularity of day, which means they would like to have the ability to generate the report on a yearly, monthly, daily, and weekday basis.

Here is a partial list of fields to serve as an example:

dateid

month

monthname

…

…

Take a screenshot of the fieldnames for the table MyDimDate.

Name the screenshot `1-MyDimDate.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 2: Design the dimension table MyDimWaste

Write down the fields in the MyDimWaste table in any text editor, one field per line.

Take a screenshot of the fieldnames for the table MyDimWaste.

Name the screenshot `2-MyDimWaste.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 3: Design the dimension table MyDimZone

Write down the fields in the MyDimZone table in any text editor, one field per line.

Take a screenshot of the fieldnames for the table MyDimZone.

Name the screenshot `3-MyDimZone.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 4: Design the fact table MyFactTrips

Write down the fields in the MyFactTrips table in any text editor, one field per line.

Take a screenshot of the fieldnames for the table MyFactTrips.

Name the screenshot `4-MyFactTrips.jpg`. (Images can be saved with either the .jpg or .png extension.)

Exercise 2 - Create schema for data warehouse on PostgreSQL

In this exercise, you will create the tables you have designed in the previous exercise. Open pgAdmin and create a database named **Project**, then create the following tables.

Task 5: Create the dimension table MyDimDate

Create the MyDimDate table.

Take a screenshot of the SQL statement you used to create the table MyDimDate.

Name the screenshot `5-MyDimDate.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 6: Create the dimension table MyDimWaste

Create the MyDimWaste table.

Take a screenshot of the SQL statement you used to create the table MyDimWaste.

Name the screenshot `6-MyDimWaste.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 7: Create the dimension table MyDimZone

Create the MyDimZone table.

Take a screenshot of the SQL statement you used to create the table MyDimZone.

Name the screenshot `7-MyDimZone.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 8: Create the fact table MyFactTrips

Create the MyFactTrips table.

Take a screenshot of the SQL statement you used to create the table MyFactTrips.

Name the screenshot `8-MyFactTrips.jpg`. (Images can be saved with either the .jpg or .png extension.)

Exercise 3: Load data into the data warehouse

In this exercise, you will load the data into the tables.

After the initial schema design, you were told that due to operational issues, data could not be collected in the format initially planned. This implies that the previous tables (MyDimDate, MyDimWaste, MyDimZone, MyFactTrips) in the *Project* database and their associated attributes are no longer applicable to the current design. The company has now provided data in CSV files with new tables DimTruck and DimStation as per the new design.

You will need to load the data provided by the company in CSV format. First, create a new database named **FinalProject**. Then, create the tables DimDate, DimTruck, DimStation, and FactTrips by defining the structure of the columns as per the CSV files. Next, load the data from the CSV files into the appropriate tables.

> Note: Ensure that you upload the files to this path: /var/lib/pgadmin/

Task 9: Load data into the dimension table DimDate

Download the data from https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimDate.csv

Load this data into DimDate table.

Take a screenshot of the first 5 rows in the table DimDate.

Name the screenshot `9-DimDate.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 10: Load data into the dimension table DimTruck

Download the data from https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimTruck.csv

Load this data into DimTruck table.

Take a screenshot of the first 5 rows in the table DimTruck.

Name the screenshot `10-DimTruck.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 11: Load data into the dimension table DimStation

Download the data from https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/DimStation.csv

Load this data into DimStation table.

Take a screenshot of the first 5 rows in the table DimStation.

Name the screenshot `11-DimStation.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 12: Load data into the fact table FactTrips

Download the data from https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0260EN-SkillsNetwork/labs/Final%20Assignment/FactTrips.csv

Load this data into FactTrips table.

Take a screenshot of the first 5 rows in the table FactTrips.

Name the screenshot `12-FactTrips.jpg`. (Images can be saved with either the .jpg or .png extension.)

Exercise 4 - Write aggregation queries and create materialized views

In this exercise, you will query the data you have loaded in the previous exercise.

Task 13: Create a grouping sets query

Create a grouping sets query using the columns stationid, trucktype, total waste collected.

Take a screenshot of the SQL and the output rows.

Name the screenshot `13-groupingsets.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 14: Create a rollup query

Create a rollup query using the columns year, city, stationid, and total waste collected.

Take a screenshot of the SQL and the output rows.

Name the screenshot `14-rollup.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 15: Create a cube query

Create a cube query using the columns year, city, stationid, and average waste collected.

Take a screenshot of the SQL and the output rows.

Name the screenshot `15-cube.jpg`. (Images can be saved with either the .jpg or .png extension.)

Task 16: Create a materialized view

Create a materialized view named max_waste_stats using the columns city, stationid, trucktype, and max waste collected.

Take a screenshot of the SQL.

Name the screenshot `16-mv.jpg`. (Images can be saved with either the .jpg or .png extension.)

© IBM Corporation. All rights reserved.