

Snowflake Database Setup Documentation

Table of Contents

1. [Introduction](#)
2. [Database and Schema Creation](#)
3. [Table Creation](#)
 - [Dimension Tables](#)
 - [Fact Table](#)
4. [File Format and Stage Setup](#)
5. [Data Loading Process](#)
 - [Preparing Data Files](#)
 - [Loading Data into Tables](#)
6. [User Creation and Access Control](#)

Introduction

This document outlines the process for setting up a Snowflake data warehouse for retail analytics. The setup includes creating dimension and fact tables in a star schema, configuring data loading mechanisms, and establishing user access.

Database and Schema Creation

First, create the database and schema:

```
sql
-- Create database
CREATE DATABASE "testdb";

-- Create schema
CREATE SCHEMA "testschema";
```

Table Creation

Dimension Tables

Create the dimension tables that will store reference data:

DimLoyaltyInfo

sql

```
CREATE OR REPLACE TABLE DimLoyaltyInfo (
    LoyaltyInfoID INT PRIMARY KEY,
    ProgramName VARCHAR(100),
    ProgramTier VARCHAR(500),
    PointsSecured INT
);
```

DimCustomer

sql

```
CREATE OR REPLACE TABLE DimCustomer (
    CustomerID INT PRIMARY KEY AUTOINCREMENT START 1 INCREMENT 1,
    FirstName VARCHAR(1550),
    LastName VARCHAR(1550),
    Gender VARCHAR(1550),
    DOB DATE,
    Email VARCHAR(1500),
    Address VARCHAR(1555),
    City VARCHAR(1550),
    State VARCHAR(1550),
    ZipCode VARCHAR(1510),
    LoyaltyInfoID INT
);
```

DimProductData

sql

```
CREATE OR REPLACE TABLE DimProductData (
    ProductID INT PRIMARY KEY AUTOINCREMENT START 1 INCREMENT 1,
    ProductName VARCHAR(1500),
    Category VARCHAR(550),
    BrandName VARCHAR(550),
    UnitPrice DECIMAL(10, 2)
);
```

DimDate

sql

```
CREATE OR REPLACE TABLE DimDate (
    DateID INT PRIMARY KEY,
    Date DATE,
    DayofWeek VARCHAR(100),
    Month INT,
    Year INT,
    Quarter INT,
    IsWeekend BOOLEAN
);
```

DimStoreData

sql

```
CREATE OR REPLACE TABLE DimStoreData (
    StoreID INT PRIMARY KEY AUTOINCREMENT START 1 INCREMENT 1,
    StoreName VARCHAR(1500),
    StoreType VARCHAR(550),
    StoreOpeningDate DATE,
    Address VARCHAR(2555),
    City VARCHAR(550),
    State VARCHAR(550),
    Country VARCHAR(550),
    Region VARCHAR(550),
    ManagerName VARCHAR(1500)
);
```

Fact Table

Create the fact table that will store transactional data:

FactOrders

```
sql
```

```
CREATE OR REPLACE TABLE FactOrders (
    OrderID INT PRIMARY KEY AUTOINCREMENT START 1 INCREMENT 1,
    DateID INT,
    ProductID INT,
    StoreID INT,
    CustomerID INT,
    QuantityOrdered INT,
    OrderAmount DECIMAL(10, 2),
    DiscountAmount DECIMAL(10, 2),
    ShippingCost DECIMAL(10, 2),
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (DateID) REFERENCES DimDate(DateID),
    FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES DimProductData(ProductID),
    FOREIGN KEY (StoreID) REFERENCES DimStoreData(StoreID)
);
```

File Format and Stage Setup

Configure the file format and stage for data loading:

```
sql
```

```
-- Create file format for CSV files
CREATE OR REPLACE FILE FORMAT CSV_SOURCE_FILE_FORMAT
TYPE = 'CSV'
SKIP_HEADER = 1
FIELD_OPTIONALLY_ENCLOSED_BY = ''
DATE_FORMAT = 'YYYY-MM-DD';

-- Create stage for file uploads
CREATE OR REPLACE STAGE TESTSTAGE;
```

Data Loading Process

Preparing Data Files

The following PUT commands must be executed via the SnowSQL CLI, not inside a worksheet:

sql

```
-- These commands should be run in SnowSQL CLI
PUT file:///D:/dwbi/DimLoyaltyInfo.csv @TESTSTAGE/DimLoyaltyInfo/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/DimCustomerData.csv @TESTSTAGE/DimCustomerData/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/DimProductData.csv @TESTSTAGE/DimProductData/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/DimDate.csv @TESTSTAGE/DimDate/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/DimStoreData.csv @TESTSTAGE/DimStoreData/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/factorders.csv @TESTSTAGE/factorders/ AUTO_COMPRESS=FALSE;
PUT file:///D:/dwbi/Landing/*.csv @TESTSTAGE/Landing/ AUTO_COMPRESS=FALSE;
```

Loading Data into Tables

Load the data from the stage into the tables:

```

sql

-- Load data into DimLoyaltyInfo
COPY INTO DimLoyaltyInfo
FROM @TESTSTAGE/DimLoyaltyInfo/DimLoyaltyInfo.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load data into DimCustomer
COPY INTO DimCustomer (FirstName, LastName, Gender, DOB, Email, Address, City, State, ZipCode,
FROM @TESTSTAGE/DimCustomerData/DimCustomerData.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load data into DimProductData
COPY INTO DimProductData (ProductName, Category, BrandName, UnitPrice)
FROM @TESTSTAGE/DimProductData/DimProductData.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load data into DimDate
COPY INTO DimDate (DateID, Date, DayofWeek, Month, Year, Quarter, IsWeekend)
FROM @TESTSTAGE/DimDate/DimDate.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load data into DimStoreData
COPY INTO DimStoreData (StoreName, StoreType, StoreOpeningDate, Address, City, State, Country,
FROM @TESTSTAGE/DimStoreData/DimStoreData.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load data into FactOrders
COPY INTO FactOrders (DateID, ProductID, StoreID, CustomerID, QuantityOrdered, OrderAmount, Dis
FROM @TESTSTAGE/factorders/factorders.csv
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

-- Load additional data into FactOrders from Landing directory
COPY INTO FactOrders (DateID, ProductID, StoreID, CustomerID, QuantityOrdered, OrderAmount, Dis
FROM @TESTSTAGE/landing/
FILE_FORMAT = (FORMAT_NAME = 'CSV_SOURCE_FILE_FORMAT');

```

User Creation and Access Control

Create a new user for Power BI integration:

sql

```
-- Create a new user for Power BI integration
CREATE OR REPLACE USER testpbiuser
    . . . PASSWORD = 'testpbiuser',
    . . . LOGIN_NAME = 'pbiuser',
    . . . DEFAULT_ROLE = 'ACCOUNTADMIN',
    . . . DEFAULT_WAREHOUSE = 'COMPUTE_WH',
    . . . MUST_CHANGE_PASSWORD = TRUE;

-- Grant ACCOUNTADMIN role to the user
GRANT ROLE accountadmin TO USER testpbiuser;
```