DSSA Data Gathering & Warehousing

Instructor: Carl Chatterton **Term**: Fall 2022 **Module**: 2 **Week**: 6

Building A Workflow App to model data as a Star Schema

Introduction

Extract, Transform & Load (ETL) is a process that extracts, transforms, and loads data from one or multiple sources to a **data warehouse** or other unified data repository.

The following repository contains instructions for connecting to a PostgreSQL database called `dvdrental` and writing your own ETL to create a **star-schema** in the Data Warehouse

To complete this project successfully you will need to use a few python libraries,

- **SQLAlchemy** is a massive python library with many modules that you will have to explore and read documentation to use in your final project.
- **psycopg2 or psrycopg3** Provide light weight cursor objects you can use for connecting and querying. You will also have to explore and read documentation to use in your final project
- **Pandas** - A is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool. It makes it easy to manipulate data using Dataframes.
- **NetworkX** - Is a Graph and Network Analysis library. We will be using this for constructing a DAG. There are limited number of modules we will need from networkx or you can try to code a DAG from scratch.

Reading software documentation often feels like reading the owners manual of a car, but is a necessary part of good software development practices.

About the DVD Rental Database

The DVD rental database represents the business processes of a DVD rental store as an OLTP PostgreSQL DB.

> The DVD rental database has many objects including: 15 tables 1 trigger 7 views 8 functions 1 domain 13 sequences

15 tables in the DVD Rental database:

> - actor – stores actors data including first name and last name.
> - film – stores film data such as title, release year, length, rating, etc.
> - film_actor – stores the relationships between films and actors.
> - category – stores film's categories data.
> - film_category- stores the relationships between films and categories.
> - store – contains the store data including manager staff and address.
> - inventory – stores inventory data.
> - rental – stores rental data.
> - payment – stores customer's payments.

> - staff – stores staff data.
> - customer – stores customer data.
> - address – stores address data for staff and customers
> - city – stores city names.
> - country – stores country names.

Entity Relationship Diagrams - An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object , a component of data.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

The **DVD Rental Database ERD** can be found in the `docs` folder of this repository as a PDF

Objectives

The main objective of this lab is to implement an ETL process in python to create a **Star-Schema** in a Data Warehouse that looks like the following:

img

Put simply, we need to:

1. *extract* data from a OLTP database called `dvdrental`
2. *transform* it by creating an aggregation of the count of rentals
3. *load* the data into the `dw` Data Warehouse

A walk-through of each table

Fact Table: FACT_RENTAL

- `sk_customer` is the `customer_id` from customer table
- `sk_date` is `rental_date` from the rental table
- `sk_store` the `store_id` from the store table
- `sk_film` is the `film_id` from the film table
- `sk_staff` is the `id` from the staff table
- `count_rentals` A count of the total rentals grouped by all other fields in the table

Dimension Table: STAFF

- `sk_staff` is the `id` field from the staff table
- `name` a concatenation of `first_name` and `last_name` from the staff table
- `email` is the `email` field from the staff table

Dimension Table: CUSTOMER

- `sk_customer` is the `customer_id` from customer table
- `name` is the concatenation of `first_name` & `last_name` from the customer table
- `email` is the customer's email

Dimension Table: DATE

- `sk_date` is unique `rental_date` used as a primary key
- `quarter` is a column formatted from `rental_date` for quarter of the year
- `year` is a column formatted from `rental_date` for year
- `month` is a column formatted from `rental_date` for month of the year
- `day` is a column formatted from `rental_date` for day of the month

Dimension Table: STORE

- `sk_store` the `store_id` from the store table
- `name` is a concatenation of `first_name` & `last_name` from the staff table
- `address` is the `address` field from the address table
- `city` is the `city` field from the city table
- `state` is the `district` field from the address table
- `country` is the `country field from the country table

Dimension Table: FILM

- `sk_film` is the `film_id` from the film table
- `rating_code` is the `rating` field from the film table
- `film_duration` is the `length` field from the film table
- `rental_duration` is the `rental_duration` from the film table
- `language` is the `name` field from the language table
- `release_year` is the `release_year` from the film table
- `title` is the `title` field from the film table
