

Data Pipelines with Redis

Background Information

You have been hired by a telecommunications company that wants to optimize their business processes. They have a Neo4j graph database that contains information about their customers, their subscriptions, and the services they are using. However, they also want to store this data in a more traditional relational database to allow for easier querying and analysis. They have asked you to create a data pipeline that extracts data from their Neo4j database, transforms it using pandas, and loads it into a Postgres database.

Guidelines

1. Extracting data from Neo4j:

To extract data from the Neo4j database, you will need to use the Neo4j Python driver. You must authenticate with the database and write a Cypher query to extract the necessary data. The fields that you should extract from the database include the following:

- Customer ID
- Subscription ID
- Service ID
- Start date of subscription
- End date of subscription
- Price of subscription

2. Transforming data using Pandas:

Once you have extracted the data from Neo4j, you must transform it using Pandas. You should create a Pandas DataFrame from the extracted data and perform any necessary data cleaning and manipulation. For example, you may need to convert date fields from strings to datetime objects and remove null values.

3. Loading data into Postgres:

Finally, you must load the transformed data into a Postgres database. You should create a new table in the Postgres database with the following fields:

- Customer ID (integer)
- Subscription ID (integer)
- Service ID (integer)
- Start date of subscription (date)
- End date of subscription (date)

- Price of subscription (float)

You should then use the psycopg2 Python library to connect to the Postgres database and write a function to insert the transformed data into the new table.

You can find the file to start working on this project here [\[link\]](#).

Deliverables

We will be expected to deliver a GitHub repository with the following:

- A Python file containing extract, transform, and load functions.
- A README file explaining how to set up and run the data pipeline, explaining the data schema and the transformations performed on the data.