Designing a data pipeline to integrate and transform worker compensation data from multiple sources involves several steps, including data extraction, transformation, and loading.

Step 1 : Data Extraction

1. CSV Files

Use Pandas to read data from CSV files

Eg :

import pandas as pd

csv\_data = pd.read\_csv('path\_to\_csv\_file.csv')

1. Relational Database

Use SQLAlchemy to extract data

Eg :

from sqlalchemy import create\_engine

engine = create\_engine('database\_connection\_string')

query = 'SELECT \* FROM workers\_compensation'

db\_data = pd.read\_sql(query, engine)

1. RESTful API

Use the Requests library in Python to fetch data from APIs

Eg :

import requests

response = requests.get('api\_endpoint\_url')

api\_data = response.json()

Step 2 : Data Transformation

1. Data Cleaning

Handle missing values, remove duplicates, and standardize data formats

1. Data Normalization

Data from different sources may have varying formats and conventions.

For example, date formats, numerical precision, and categorical values might differ.

Normalizing these ensures that all data adheres to a single standard, making it easier to merge and compare

Step 3 : Data Loading

Using data warehouse such as Amazon Redshift, Google BigQuery, or Snowflake