#### **1. Understand the Data Sources**

The data sources include information about entities such as sources, companies, and their relationships. Here’s a detailed look at the data columns you provided:

* **source\_id**: Unique identifier for the source.
* **source\_name**: Name of the source.
* **iso3\_country**: ISO3 code for the country.
* **original\_inventory\_sector**: The sector to which the source belongs.
* **lat**: Latitude of the source.
* **lon**: Longitude of the source.
* **geometry\_ref**: Reference to the geometric location.
* **relationship**: Type of relationship the source has.
* **ultimate\_parent\_name**: Name of the ultimate parent company.
* **ultimate\_parent\_id**: Identifier for the ultimate parent company.
* **percent\_interest\_parent**: Percentage interest of the parent.
* **company\_name**: Name of the company.
* **company\_id**: Identifier for the company.
* **percent\_interest\_company**: Percentage interest of the company.
* **interest\_units**: Units of interest.
* **start\_date**: Start date of the record.
* **end\_date**: End date of the record.
* **created\_date**: Date when the record was created.
* **modified\_date**: Date when the record was last modified.
* **percent\_company\_datasource**: Percentage source data for the company.
* **percent\_parent\_datasource**: Percentage source data for the parent.

#### **2. Define the Data Model**

Based on the columns, we can define a relational data model. This model will help structure the data in a way that captures relationships and allows for efficient querying.

##### **Entity-Relationship (ER) Model**

**Entities**:

1. **Source**
   * Attributes: source\_id, source\_name, iso3\_country, original\_inventory\_sector, lat, lon, geometry\_ref, created\_date, modified\_date.
2. **Company**
   * Attributes: company\_id, company\_name, ultimate\_parent\_id, percent\_interest\_company, interest\_units, start\_date, end\_date, created\_date, modified\_date, percent\_company\_datasource.
3. **Parent**
   * Attributes: ultimate\_parent\_id, ultimate\_parent\_name, percent\_interest\_parent, percent\_parent\_datasource.
4. **Relationship**
   * Attributes: source\_id (foreign key), company\_id (foreign key), relationship, created\_date, modified\_date.

##### **Relational Schema**

**Table 1: Sources**

* source\_id (Primary Key)
* source\_name
* iso3\_country
* original\_inventory\_sector
* lat
* lon
* geometry\_ref
* created\_date
* modified\_date

**Table 2: Companies**

* company\_id (Primary Key)
* company\_name
* ultimate\_parent\_id (Foreign Key referencing Parents)
* percent\_interest\_company
* interest\_units
* start\_date
* end\_date
* created\_date
* modified\_date
* percent\_company\_datasource

**Table 3: Parents**

* ultimate\_parent\_id (Primary Key)
* ultimate\_parent\_name
* percent\_interest\_parent
* percent\_parent\_datasource

**Table 4: Relationships**

* source\_id (Foreign Key referencing Sources)
* company\_id (Foreign Key referencing Companies)
* relationship
* created\_date
* modified\_date

#### **3. Data Integration Strategy**

##### **ETL Pipeline Stages**

1. **Data Ingestion**:
   * Extract data from the provided CSV and XLSX files.
   * Use tools like PySpark or Pandas for initial data loading and processing.
2. **Data Cleaning**:
   * Normalize and clean data to ensure consistency.
   * Standardize columns such as source\_name, company\_name, and dates.
3. **Data Transformation**:
   * Use fuzzy matching to reconcile variations in company\_name.
   * Generate unique IDs where missing.
   * Map relationships between source\_id and company\_id.
4. **Data Loading**:
   * Load the transformed data into the respective tables in the PostgreSQL data warehouse.
   * Ensure referential integrity between tables using foreign keys.
5. **Metadata Management**:
   * Track metadata such as created\_date and modified\_date for each record.
   * Store information about data sources and data lineage.

#### **4. Data Quality and Traceability**

* **Data Quality**:
  + Implement validation rules to ensure data accuracy and completeness.
  + Use PySpark for data validation and error handling.
* **Data Traceability**:
  + Maintain detailed metadata in PostgreSQL to track the source and changes of each data point.
  + Implement logging to record ETL process steps and transformations.
* **Error Correction**:
  + Use automated error detection mechanisms.
  + Provide tools for manual intervention and data correction.