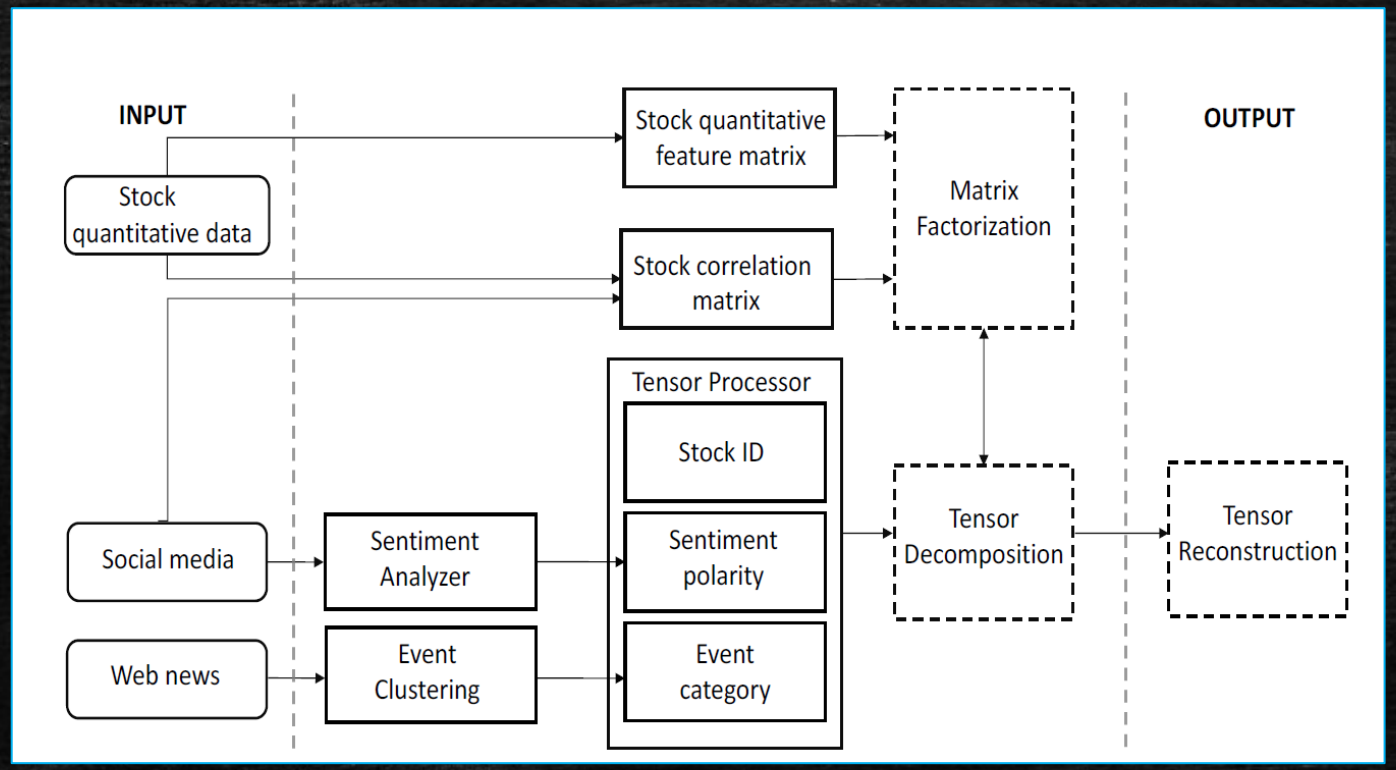
# DSE 203 (Fall 2020)

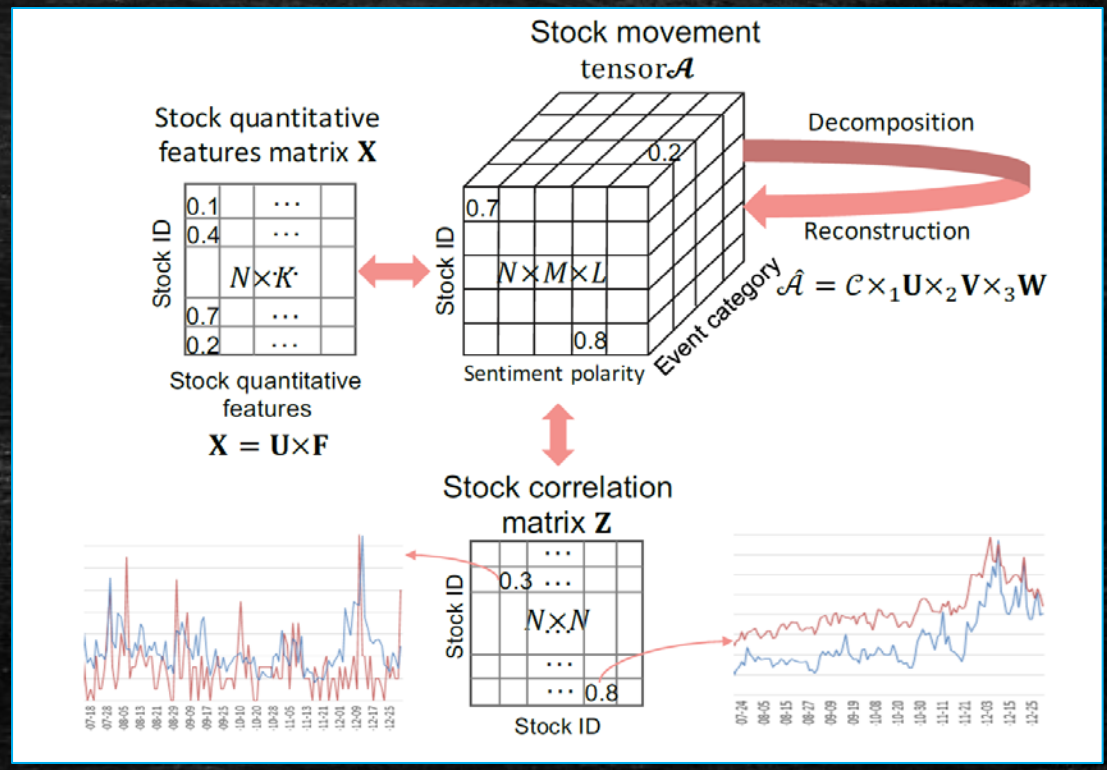
What is Data Integration?

# A Statistical/ML View of Data Integration

- Task: Stock Prediction

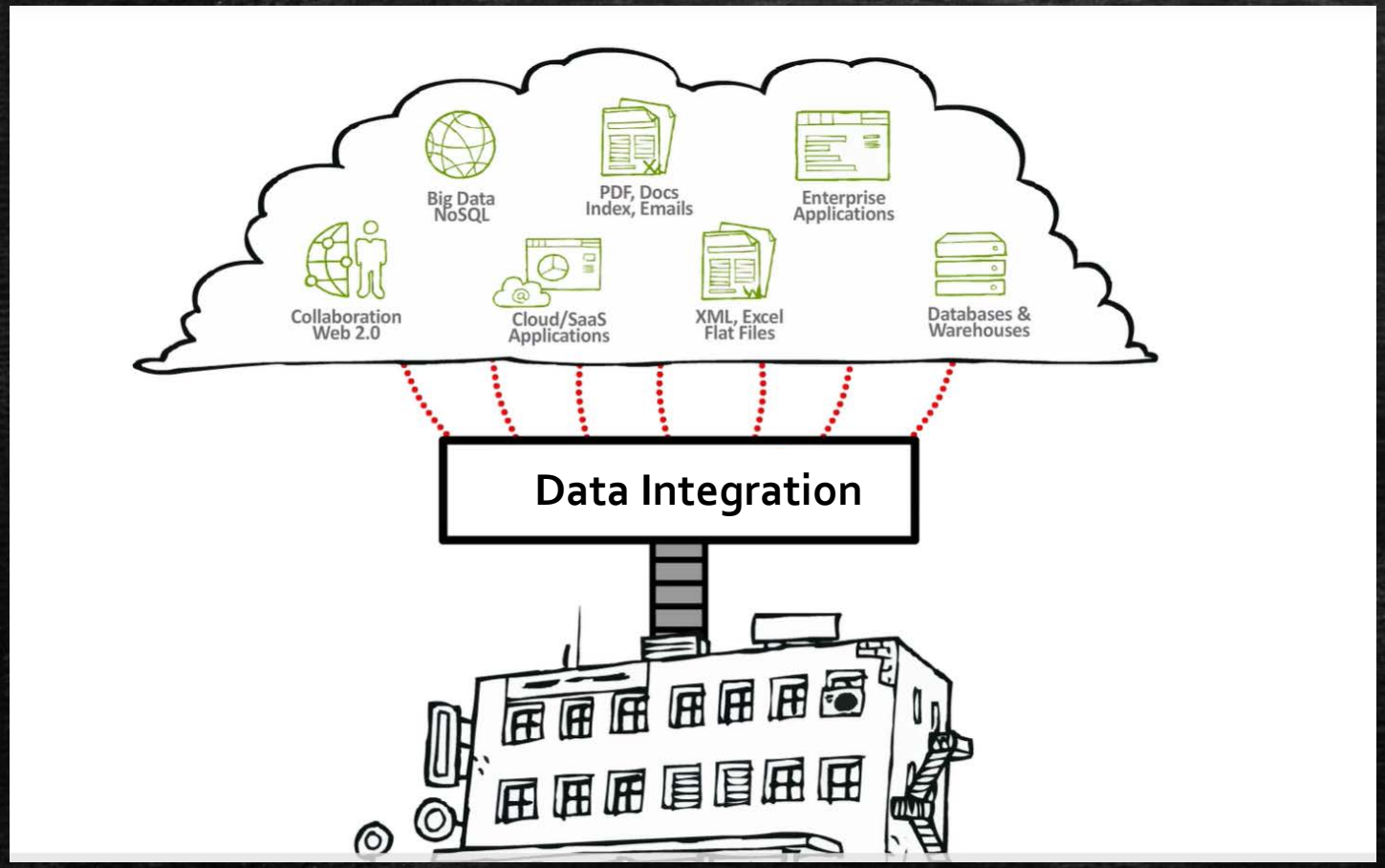
- Data sources are feature suppliers
- Features feed a mathematical model
- Elements of the data are not queried




In this course, we are not discussing this view of data integration

# An Information Systems View of Data Integration

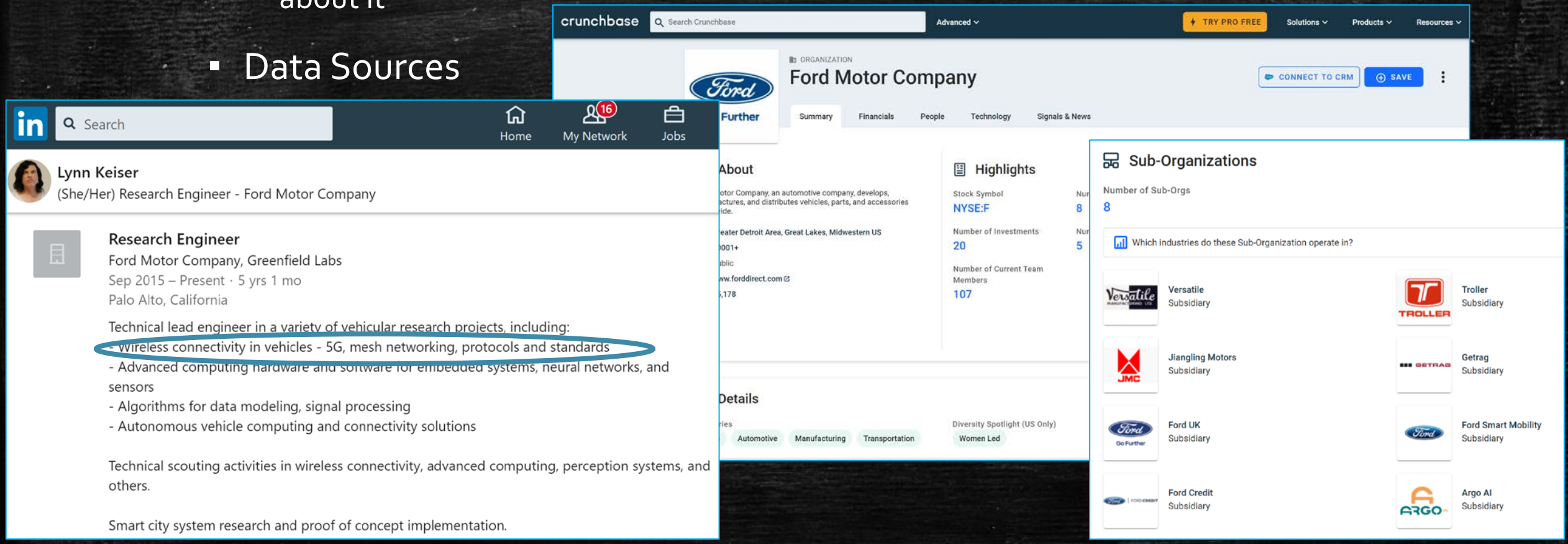



- Scenario
  - Identify data sources that have information on the application domain
  - Create a set of new data schema/data object for your problem
  - The elements of the data schema/object (e.g., relations, attributes, …) will be supplied by some combination of data sources
  - The resulting combined data schema should be used for querying and analysis as if it is an new independent data source

Adapted from Denodo.com

# Example – Market Intelligence

- Goal:
  - Create a profile for all companies in an industrial sector by gathering information about it
- Data Sources



# A Simple Example Use Case

- Two book seller companies, one US-based, and the other a world-wide company, are merging
- These companies had developed their data systems completely independently
- When they merge, it would be important that their online customers and employees can see a single view of all their data
  - Online customers primarily search and query for books and look at reviews
  - Employees, in addition, perform analytical operations on the data
    - OLAP style operations
    - Mining style operations
- We assume that data in both data sources are relational

# Schema of USBooks

- Books(ISBN, Title, Genre, [Authors], Price*, Publisher, PubDate, ReprintDate, [Awards]) * All prices in US Dollars
- Authors(AuthID, Name, [Recognitions])
- Stores(ID, Name, Address, Size) /*Size = {Small, Medium, Large} */
- DailySales(ISBN, StoreID, Date, Quantity, TotalSales)
- Reviews(ReviewID, ReviewerID, BookID, ReviewSource, URL, Date, 5StarRating, Review)
- Constraints
  - Books.Authors = [Authors.AuthID]
  - Stores.ID = DailySales.StoreID
  - Reviews.BookID = Books.ISBN

# Schema of WorldBooks

- Fictions(ISBN, Title, Authors, Language, Price*, Publisher, PubDate, ReprintDate, [Awards])
  - Similar Tables – NonFictions, Poetry, Drama

*Prices are specified as Currency, Value
e.g., GBP 50.20, USD 35.49

- Authors(AID, Salutation, FName, MInit, LName, Suffix)
- Stores(ID, StreetAddress, City, State, Country, PostalCode)
- USSales(ISBN, StoreID, Date, Qty, DailyTotal)
  - Similar Tables – EUSales, AfricaSales, AsiaSales, AuNZSales
- OnlineReviews(ID, ReviewerID, Date, 10-Star, ReviewText, Amazon, BNobles, PubSite, OtherInternetURL)
- Constraints – similar to USBooks

# Structural Heterogeneity
## A Data Integration Problem

- One table vs. Multiple tables → Book vs. Fiction, Non-Fiction, Poetry, Drama
  - Splitting Attribute → Book.genre
    - Worst case → incomplete coverage (partial domain mapping)
- One attribute vs. Multiple attributes → address vs. StreetAddress, City, State, Country, PostalCode
- Value vs. attribute → Reviews.ReviewSource vs. Amazon
  - Note Type mismatch
- New/Missing attributes → stores.Country in Worldbooks, stores.Size in USBooks
- New/Missing tables → EUSales

# Semantic Heterogeneity – I
## A Data Integration Problem

- Numerical Values and Units
  - USBooks.Books.Price → a float and the currency is in $
  - Worldbooks.Fiction.Price → a string with both the currency and the value
- IDs
  - The identifiers in the two data stores are *incompatible*
  - ISBN is globally compatible
- Value Mapping
  - Address is ***possibly*** a concatenation of StreetAddress, City, State, Country, PostalCode but the order of concatenation is still a question
- Semantic Differences
  - River bank vs. money bank vs billiards bank shot
- Hidden Semantics
  - AfricaSales – Table name implicitly introduces the value of a "new" entity called Continent

# Semantic Heterogeneity – II
## A Data Integration Problem

- Name mismatch for Schema Elements
  - Reviews vs. OnlineReviews
- Intra-aggregation
  - When comparable population is divided differently
    - Census Tracts vs. Counties for states
- Constraint Mismatch
  - When attributes referring to the same thing have different cardinalities or disjointedness assertions
    - Gender → Male and Female vs. Male, Female, LGBTQ, Undeclared
- Internal Element ordering
  - One array is ordered, and the corresponding array is not or is ordered differently

# Semantic Heterogeneity – III
## A Data Integration Problem

- Value Mismatches
  - Harry Potter and the Philosopher's Stone vs. Harry Potter and the Sorcerer's Stone
- Granularity and Non-uniformity of Content
  - Addresses

*Street :* 30 Commercial Road
*City area/District :* Fratton
*City/Town/Village:* PORTSMOUTH
*County :* Hampshire
*Postal code :* PO1 1AA
*Country :* UNITED KINGDOM

*House :* Ramu Dhobi Household
*Sub-locality :* Bukkapatnam Village
*Locality :* BUKKAPATNAM
*sub-prv 3* : Pennukonda Taluk
*District* : Anantpur
*Postcode :* 515144
*Country* : INDIA

How do we handle/reconcile this kind of heterogeneity?

# Target Schema

- Books(ISBN, Title, Genre, [Authors], Price*, Publisher, PubDate, ReprintDate)
- Authors(AuthorID, Salutation, FName, MInit, LName, Suffix)
- Stores(GlobalID, StreetAddress, City, State, Country, PostalCode)
- Sales(ISBN, CountryID, StoreID, Day, Month, Year, Quantity, TotalDailySales)
- Reviews(ReviewID, ReviewerID, BookID, ReviewSource, URL, Date, 5StarRating, Review)
- Countries(CountryID, Continent, CountryName)
- Awards(ISBN, AuthorID, AwardName, AwardDate)

* All prices in US Dollars

Anything Missing?

# Target Schema

- Books(ISBN, Title, Genre, [Authors], Price*, Publisher, PubDate, ReprintDate)
- Authors(AuthorID, Salutation, FName, MInit, LName, Suffix)
- Stores(GlobalID, StreetAddress, City, State, Country, PostalCode)
- Sales(ISBN, CountryID, StoreID, Day, Month, Year, Quantity, TotalDailySales)
- Reviews(ReviewID, ReviewerID, BookID, ReviewSource, URL, Date, 5StarRating, Review)
- Countries(CountryID, Continent, CountryName)
- Awards(ISBN, AuthorID, AwardName, AwardDate)
- StoreIDMap(DataSource,OriginalID, GlobalID) /* ID Mapping for bookkeeping */

* All prices in US Dollars

# Constraints for the Target Schema

- A touch of Dependency Theory
  - Functional dependency → Every book in the target database must have a single price, authors, title, …
    - ISBN → price, authors, title, …
    - What will happen if this constraint does not hold?
  - Inclusion dependency → Every store in the WorldBooks database must be included in the target database
    - WorldBooks.store(StreetAddress, City, State, Country, PostalCode) ⊂ Target.Stores(StreetAddress, City, State, Country, PostalCode)
- Constraints determine
  - Whether a tuple in the target schema is expected
  - Whether a tuple in the target schema is invalid

# Logic of Transformation – I

- Start with the tables of the target schema and determine how to populate it using data from the sources
- Target.Books
  - Get all books from USBooks (SQL query)
  - Make a union query for Worldbooks to get fiction, non-fiction, poetry, drama
    - Add a genre column to the result of the union query and fill it based on the name of the table it came from
    - Using a function split the currency and value for the price attribute
    - Using another function, convert the value from the native currency to $
  - Merge results
  - Export results to destination (CSV file, table in a database …)

# Logic of Transformation – II

Countries(CountryID, Continent, CountryName)

- Get all country names from WorldBooks.stores
- Add two (empty) columns to the result for Continent and CountruID
- Use a lookup table (created externaly) to find the continent of each country and fill in the "continent" column
- Autogenerate the CountryID value by using an autoincrement function
- Export to a destination

Stores(GlobalID, StreetAddress, City, State, Country, PostalCode)

- Remember to replace the Country with the CountryID by joining with the Countries table

You will use this kind of Schema/Data Mapping Logic in the Project