MONASH University

FIT5196 DATA WRANGLING

Week 10

Data Integration & Enrichment

By Jackie Rong

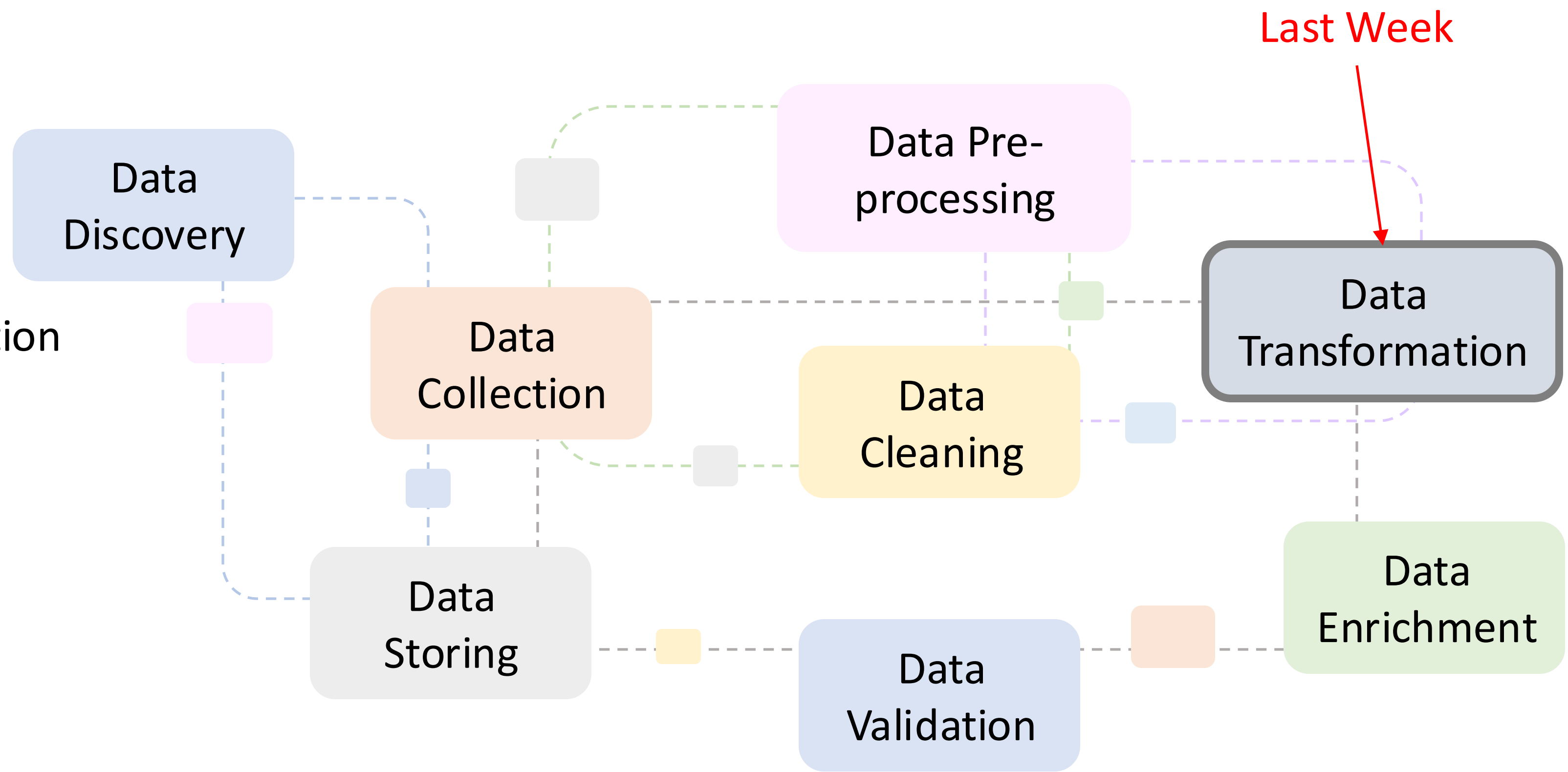
Faculty of Information Technology

Monash University

Data Wrangling Tasks (Recap)

In the **Data Pre-processing** stage, preliminary data preparation tasks are performed to make raw data more suitable for analysis.

- Overview of Data Transformation
- Data Normalisation
- Data Discretisation
- Data Construction
  - Feature Engineering
  - Data Sampling



Data Transformation

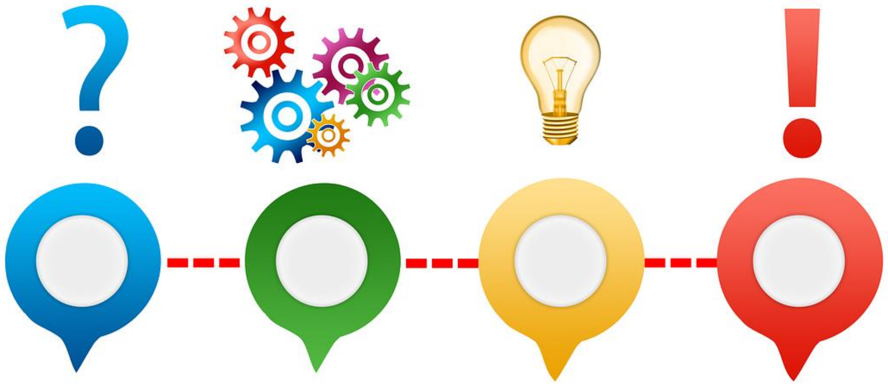
- **Data transformation** involves cleaning and converting raw data into a format that is more suitable for analysis.
- The **goal** of data transformation is to ensure the data is in usable and efficient format that makes analysis straightforward and reliable.
- Reasons for data transformation
  - Fix skewness in data
  - Enhance data visualisation
  - Better interpretability
  - Improve the compatibility of data with assumptions underlying a modelling process

Data Transformation

- Data transformation involves
  - Data Normalisation
  - Linear Transformation
  - Power Transformation
  - Data Discretisation
  - Data Construction
  - Data Reduction

Data Enrichment

- Overview of Data Enrichment
- Schema Integration
- Data-level Integration



Data Enrichment

- **Data enrichment** refers to the process of enhancing existing data by appending additional context or information from external sources.
- This process enhances the quality, depth, and value of the data, making it more useful for detailed analysis and informed decision-making.
  - **Contextual Addition**: Adding data that provides more insight into the existing data, such as demographic information, geographic details, or industry-specific metrics.
  - **Quality Improvement**: Enhancing the quality of data by adding more accurate or timely information, which can improve the granularity or accuracy of analysis.
  - **Value Enhancement**: Directly increasing the utility of the data for analytical or operational purposes, making it more comprehensive for decision-making processes.

Data Integration

- **Data integration** is a crucial component of the data wrangling process, which involves combining data from different sources to create a unified view.
- This process is essential for data analysis and decision-making, particularly in environments where data is collected from multiple sources or systems.
  - **Source Diversity**: Data comes from multiple sources, such as different databases, spreadsheets, or external APIs.
  - **Schema Merging**: Integrating various data schemas into a single, unified schema that represents all data consistently.
  - **Entity Resolution**: Identifying and consolidating records that refer to the same entities across datasets.
  - **Centralization**: Often results in a centralized data repository that facilitates easier access and analysis.

Data Enrichment vs. Data Integration

	Data Enrichment	Data Integration
Purpose	Enrichment is about enhancing the data's value by adding more detailed information to the existing dataset.	Integration is primarily about combining data to create a unified database or dataset, focusing on consistency and accessibility.
Output	The result of data enrichment is an enhanced dataset with additional layers of information.	The result of data integration is a consolidated dataset from multiple sources.
Process	Enrichment involves appending relevant data to existing records to provide deeper insights.	Integration involves merging and reconciling data from various sources into one coherent set, addressing conflicts in data structure or format.

Data Integration is Challenging

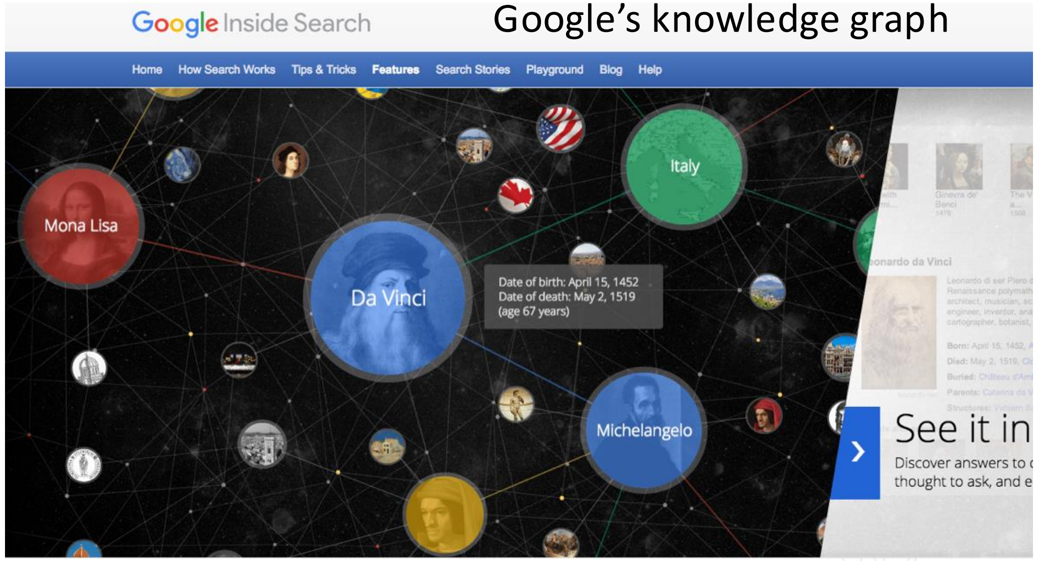
- **Heterogeneous data**
  - Data coming from different sources is often developed independently (e.g., different schema, different objectives)
- **Various formats**
  - Text, web logs, social networks, sensors, astronomy, genomics, medical records, surveillance, etc.
- **Incompatible Taxonomies**
  - Different object identity and separate schema
    - Different definitions of a customer, an account, etc.
- **Time synchronisation**
  - Each source might have a time window that is different from each other.
  - Synchronisation of data collected in different time windows

Data Integration is Challenging

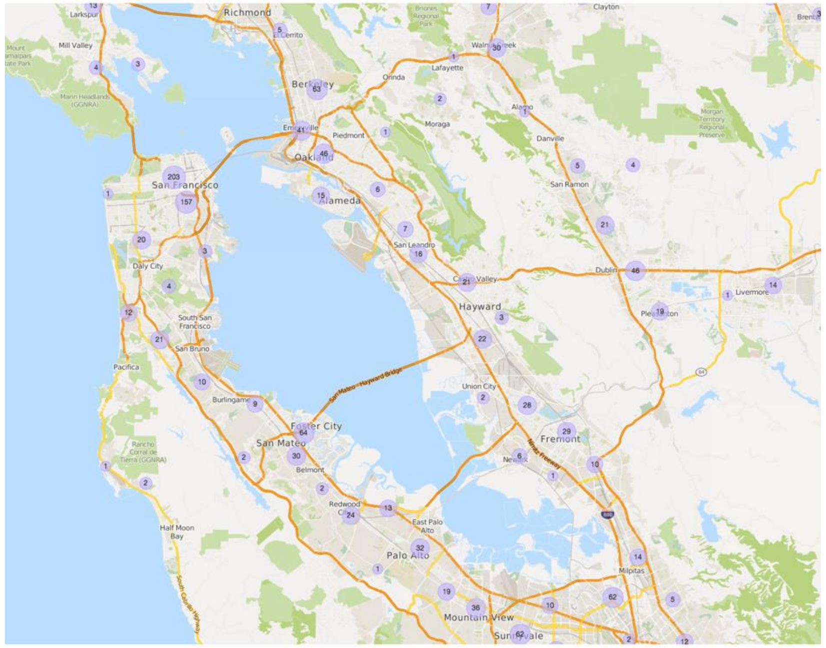
- **Dealing with legacy data**
  - Historical data stored in legacy form, such as IMS, spreadsheets, and other ad-hoc structure
  - Combine historical data with modern data
- **Abstraction levels**
  - Different data sources might provide data at different level of abstraction,
  - e.g.,
    - suburb level vs. state level
    - annual vs. weekly
- **Data Quality**
  - Data is often erroneous, and combining data often aggravates the problems. Erroneous data has potentially devastating impact on the overall quality of the integrated data.
- **The number of sources**
  - e.g., web-scale integration.

Applications of Data Enrichment

Google's knowledge graph

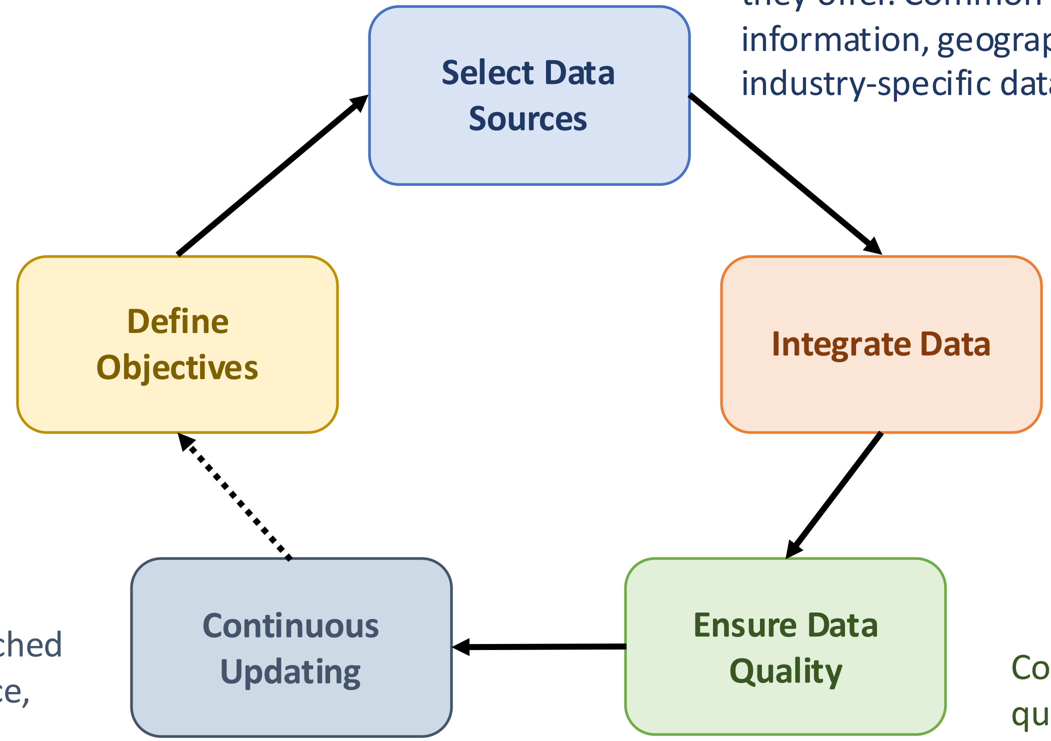


Map mashup: HousingMaps



Map mashup: TrendMaps
shows the latest trend in twitter

Steps of Data Enrichment



Determine what specific information is missing from your current dataset and what you need to enhance its value for particular uses, such as targeted marketing, customer relationship management, or advanced analytics.

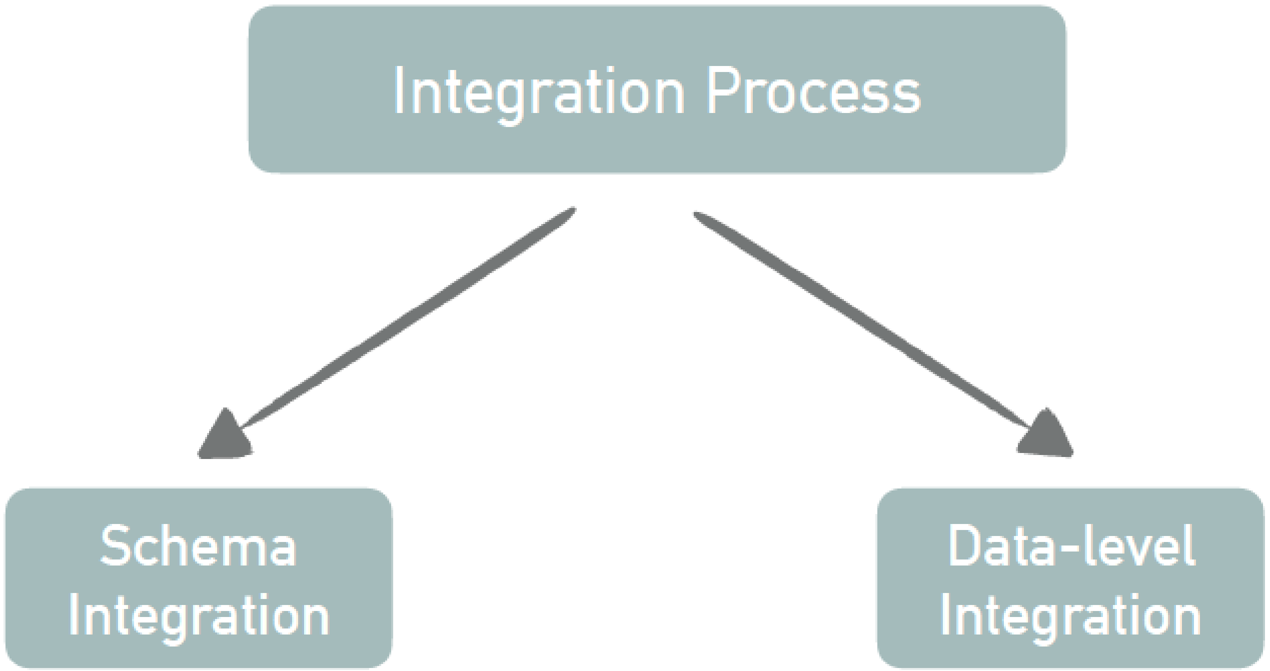
Choose appropriate external sources based on the reliability, accuracy, and relevance of the data they offer. Common sources include demographic information, geographic data, social media data, industry-specific databases, and more.

Merge the external data with your existing dataset. This might involve complex ETL (Extract, Transform, Load) processes, especially if the data structures differ significantly.

Conduct thorough checks to verify the quality of the enriched data. This includes validating the accuracy, completeness, and timeliness of the data.

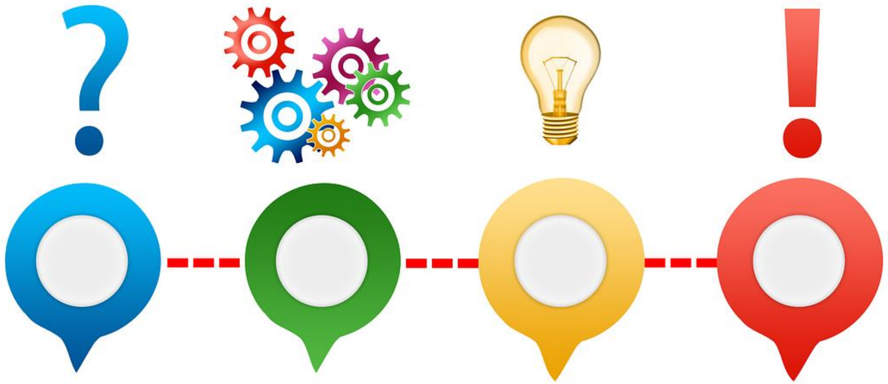
Periodically update the enriched data to maintain its relevance, especially for dynamically changing datasets like consumer behaviour or market trends.

Data Integration Category



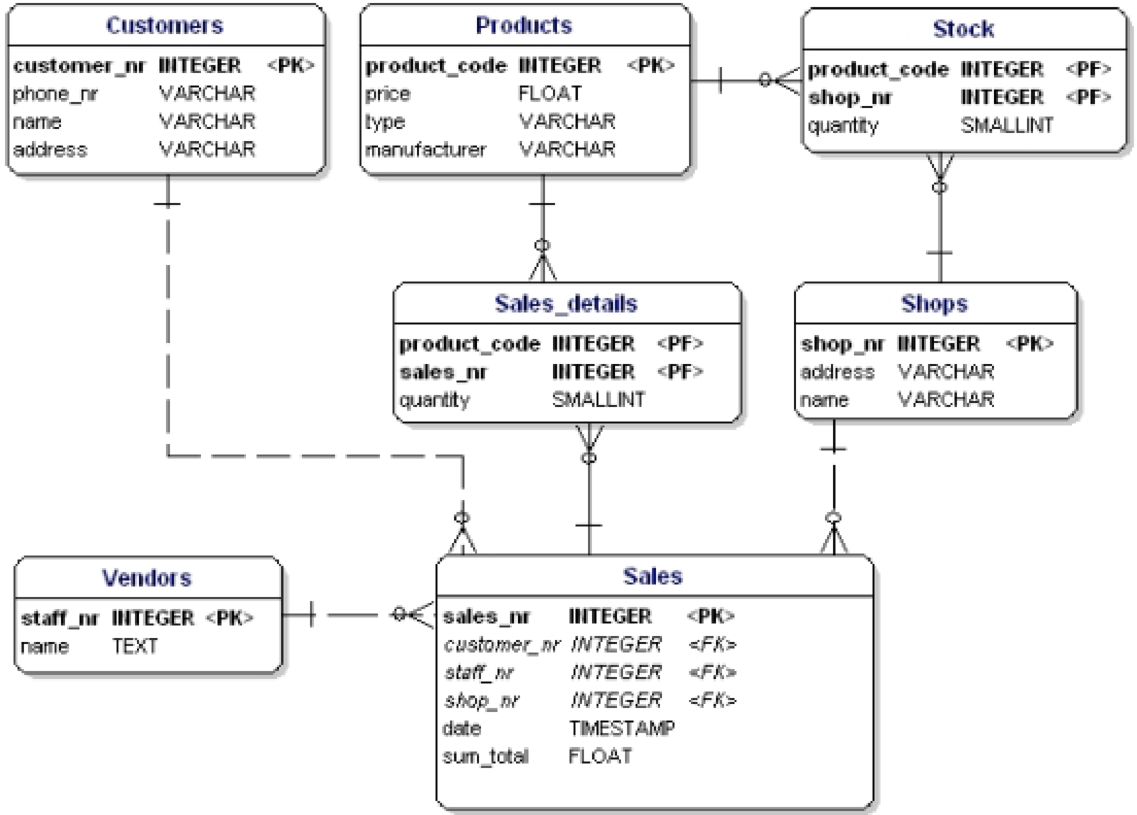
Data Enrichment

- Overview of Data Enrichment
- Schema Integration
- Data-level Integration



Schema

- **Relational databases**
  - A **schema** specifies a set of tables.
  - A table contains a set of attributes associated with their data types.



This figure is from http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html

Schema

- Relational databases
  - A schema specifies a set of tables.
  - A table contains a set of attributes associated with their data types.
- **Data models like XML and JSON**
  - A **schema** is defined as a set of tags, classes and properties.


```xml
<us-patent-grant lang="EN" dtd-version="v4.2 2006-08-23" file="US07893369-20110222.XML"
    produced="20110208" date-publ="20110222">
<us-bibliographic-data-grant>
<publication-reference>
<document-id>
<country>US</country>
<doc-number>07893369</doc-number>
<kind>B2</kind>
<date>20110222</date>
</document-id>
</publication-reference>
<application-reference appl-type="utility">
<document-id>
<country>US</country>
<doc-number>12540086</doc-number>
<date>20090812</date>
</document-id>
</application-reference>
<us-application-series-code>12</us-application-series-code>
<priority-claims>
<priority-claim sequence="01" kind="national">
<country>CN</country>
<doc-number>2008 1 0210099</doc-number>
<date>20080822</date>
</priority-claim>
</priority-claims>
<us-term-of-grant>
<us-term-extension>35</us-term-extension>
</us-term-of-grant>
<classifications-ipcr>
<classification-ipcr>
<ipc-version-indicator><date>20060101</date></ipc-version-indicator>
<classification-level>A</classification-level>
<section>H</section>
<class>01</class>
<subclass>H</subclass>
<main-group>19</main-group>
<subgroup>11</subgroup>
<symbol-position>F</symbol-position>
<classification-value>I</classification-value>
<action-date><date>20110222</date></action-date>
<generating-office><country>US</country></generating-office>
<classification-status>B</classification-status>
<classification-data-source>H</classification-data-source>
</classification-ipcr>
</classifications-ipcr>
<classification-national>
<country>US</country>
<main-classification>200 11R</main-classification>
</classification-national>
```

Schema

- Relational databases
  - A schema specifies a set of tables.
  - A table contains a set of attributes associated with their data types.
- Data models like XML and JSON
  - A schema is defined as a set of tags, classes and properties.
- **Data science**
  - A data **schema** is defined as the representation of the data arrangement, relationships and contents.

Why Need Schema Integration?

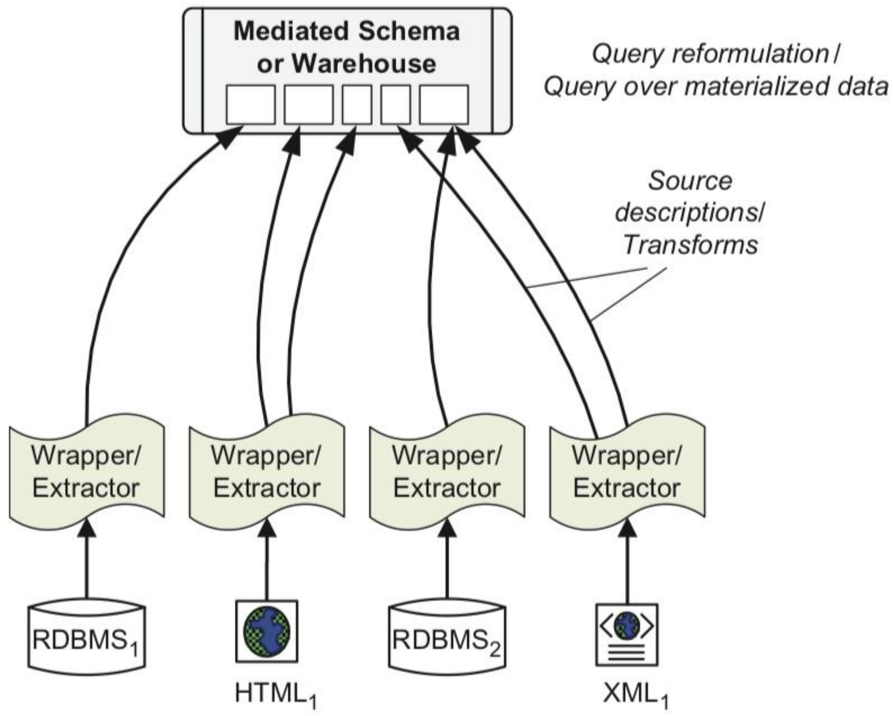


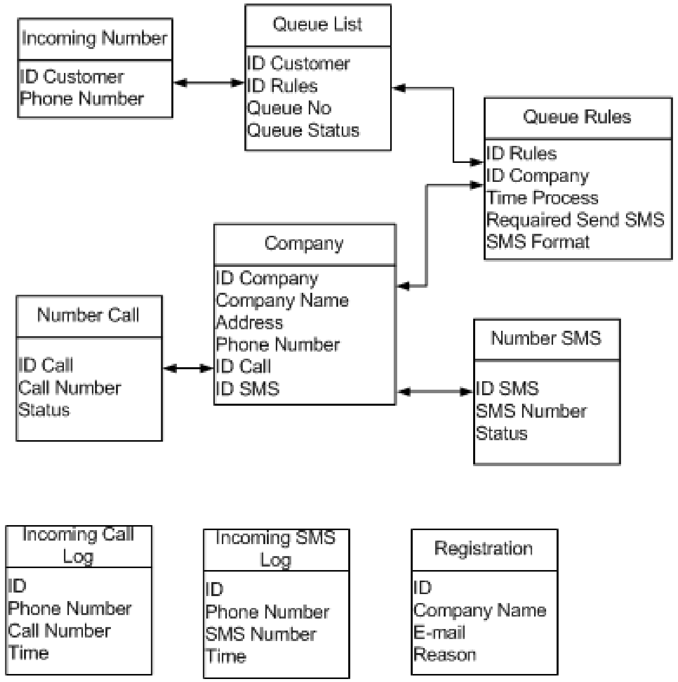
FIGURE 1.4 The basic architecture of a general-purpose data integration system. Data sources can be relational, XML, or any store that contains structured data. The *wrappers* or *loaders* request and parse data from the sources. The *mediated schema* or central *data warehouse* abstracts all source data, and the user poses queries over this. Between the sources and the mediated schema, *source descriptions* and their associated *schema mappings*, or a set of *transformations*, are used to convert the data from the source schemas and values into the global representation.

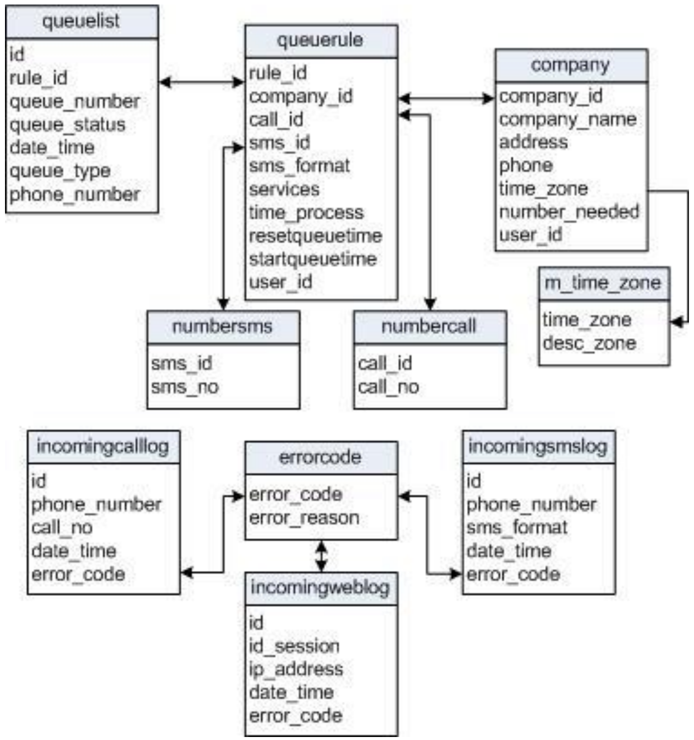
Schema Mapping

- The linkage between each data source and the mediate schema is done through semantic mapping
  - Specifies how attributes in the sources correspond to attributes in the mediated schema (when such correspondences exist)
  - Specifies how the different groupings of attributes into tables are resolved.
  - Specifies how to resolve schema conflict from different sources

Problems with Schema integration

- **Structure conflicts**
  - Inconsistencies in the data structure among schemas, which include
    - Different data source origins: Data can be represented in a structure form (e.g., XML, HMTL, JSON, semi-structured, or completely unstructured data.
  - Inconsistencies among the set of elements inside the different schemas





Figures are from http://www.urremote.com/untethering-the-queue-2

Problems with Schema integration

- **Naming conflicts**
  - homonyms vs synonyms
    - The same name is used for different objects.
    - Different names are used for the same object.
  - Examples
    - Homonyms: ID can refer to customer ID, product ID, store ID, etc.
    - Synonyms: Customer ID and Client ID can refer to the same real-world object, i.e., customer/client.

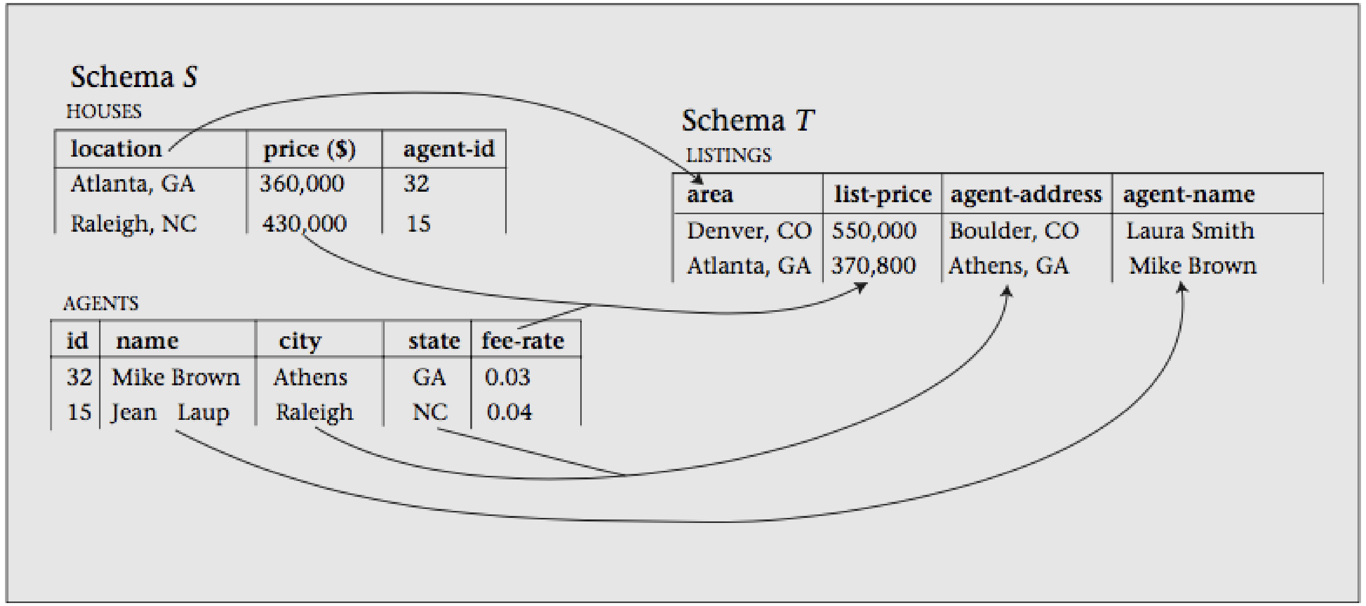


Figure 2. The Schemas of Two Relational Databases S and T on House Listing, and the Semantic Correspondences between Them.

Figure is from "Semantic-Integration Research in the Database community" by AnHai Doan and Alon Y. Halevy

Problems with Schema integration

- **Entity resolution/conflict resolution**
  - Different units:
    - Temperature units: Celsius and Fahrenheit
    - Currencies
  - Data type heterogeneity
    - Same kind of attributes with different data types
      - phone number can be stored as string in one database and integer in another database
  - Value heterogeneity
    - The use of Abbreviations: Professor vs. Prof, Street vs. St, Road vs. Rd
  - Level of abstraction: different aggregation levels for an attributes
    - Address can be split into multiple fields, street number, street name, suburb, city, post-code, etc.

Problems with Schema integration

- **Entity resolution/conflict resolution**
  - Semantic heterogeneity: differences in meaning and interpretation of data values[1]
    - Naming
      - Case sensitivity
      - Synonyms/Homonyms
      - Acronyms
    - Generalisation/Specialisation: one schema may refer to "phone", but the other schema has multiple elements such as "home phone", "work phone" and "cell phone"
  - Different points of time
    - Fortnight and monthly payment

Schema Matching

- **Semantic matching**: relates a set of elements in schema S to a set of elements in schema T.

DVD-VENDOR
Movies(id, title, year)
Products(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
Locations(lid, name, taxRate)

AGGREGATOR
Items(name, releaseInfo, classification, price)

FIGURE 5.1 Example of two database schemas. Schema DVD-VENDOR belongs to a DVD vendor, while AGGREGATOR belongs to a shopping site that aggregates products from multiple vendors.

  - One-to-One matching
    - Movies.title ≈ Items.name
    - Movies.year ≈ Items.year
    - Product.rating ≈ Items.classification
  - One-to-Many matching
    - Items.price ≈ Products.basePrices $\times$(1 + Locations.taxRate)

Name-Based Matching

- **Name-Based Matcher**: compares the names of attributes (or column headers) in the hope that the names convey the true semantics of the elements.
  - Split names according to certain delimiters, such as capitalization, numbers, or special symbols.
    - ClientName ⇒ Client Name
    - saleLocID ⇒ Sale Loc ID
  - Expand known abbreviations or acronyms
    - loc ⇒ location
    - cust ⇒ customer
    - St ⇒ Street
    - DOB ⇒ Date of Birth
  - Expand a string with its synonyms
    - Location ⇒ Address
    - Cost ⇒ Price

Name-Based Matching

- Expand a string with its hypernyms
  - product ⇒book, DVD, etc.
- Remove articles, propositions, and conjunctions
  - Exclude words like “in”, “at”

DVD-VENDOR
Movies(id, title, year)
Products(mid, releaseDate, releaseCompany, basePrice, rating, saleLocID)
Locations(lid, name, taxRate)

AGGREGATOR
Items(name, releaseInfo, classification, price)

(a)

name-based matcher: name $\approx$ ⟨name: 1, title: 0.2⟩
releaseInfo $\approx$ ⟨releaseDate: 0.5, releaseCompany: 0.5⟩
price $\approx$ ⟨basePrice: 0.8⟩
(b)

data-based matcher: name $\approx$ ⟨name: 0.2, title: $0{,}8$⟩
releaseInfo $\approx$ ⟨releaseDate: 0.7⟩
classification $\approx$ ⟨rating: 0.6⟩
price $\approx$ ⟨basePrice: 0.2⟩
(c)

average combiner: name $\approx$ ⟨name: 0.6, title: 0.5⟩
releaseInfo $\approx$ ⟨releaseDate: 0.6, releaseCompany: 0.25⟩
classification $\approx$ ⟨rating: 0.3⟩
price $\approx$ ⟨basePrice: 0.5⟩
(d)

FIGURE 5.3 (a) Two schemas (reproduced from Figure 5.1); (b)-(c) the similarity matrices produced by two matchers for the above two schemas; and (d) the combined similarity matrix.

Instance-based Matching

- Data-Based Matcher makes use of the data values.
  - **Rule-based matching method**
    - Handcrafted rules exploit schema information such as element names, data types, structures, number of sub-elements, and integrity constraints.
    - For DVD-vendor database:
      - All possible classification: G, PG, PG-13, R, etc
      - Given a new attribute, if most of its values appear in the list above.
  - Advantages
    - Relatively inexpensive, do not require training
  - Disadvantages:
    - Cannot exploit data instances effectively (e.g., value format, frequently occurring values, etc.)

Instance-based Matching

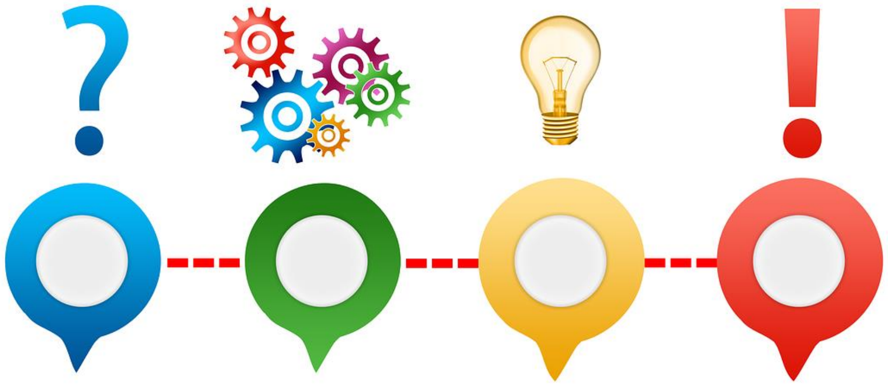
- Data-Based Matcher makes use of the data values.
  - **Learning-based matching method**: learning techniques that can exploit both schema and data information.
    - Classification-based methods
    - (semi-)automated but Needs training

Example 5.6

If s_i is address, then positive examples may include "Madison WI" and "Mountain View CA," and negative examples may include "(608) 695 9813" and "Lord of the Rings." Now suppose that element t_j is location and that we have access to three data instances of this element: "Milwaukee WI," "Palo Alto CA," and "Philadelphia PA." Then the classifier C_i may predict confidence scores 0.9, 0.7, and 0.5, respectively. In this case we may return the average confidence score of 0.7 as the similarity score between s_i = address and t_j = location.

Data Enrichment

- Overview of Data Enrichment
- Schema Integration
- Data-level Integration



MONASH University

Data-Level Integration

- Data-Level Integration: related to the integrated contents/values of data not the schema
- Categories
  - Attribute-level (columns)
    - Redundancy
    - Correlation
  - Tuple-level (rows)
    - Duplication
    - Inconsistency

Attribute-Level Integration

- Problems: combining different data sources might result in a redundant representation
- Examples
  - When any of the attributes can be calculated from others
    - e.g., annual salary from fortnight payment
  - When different values represent the same attribute but with different units
    - e.g., weight in kg and lb
- Techniques to find correlation between attributes
  - Chi-square Test for categorial variable
  - Correlation Coefficient for numerical attributes

Chi-square test

- **Chi-square test** for categorial variables
  - Test for independence compares two variables in a contingency table to see if they are related.
  - Hypothesis statements:
    - Null Hypothesis: The two categorical variables are independent.
    - Alternative Hypothesis: The two categorical variables are dependent.
  - The chi-square test statistic

$$x^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

where O represents the observed frequency, and E is the expected frequency under the null hypothesis:

$$E = \frac{Row\ Total\ \times\ Column\ Total}{Sample\ Size}$$

Chi-square test

- **Chi-square test** for categorial variables: Is gender independent of education level?

	High School	Bachelors	Masters	Ph.d.	Total
Female	60	54	46	41	201
Male	40	44	53	57	194
Total	100	98	99	98	395

	High School	Bachelors	Masters	Ph.d.	Total
Female	50.886	49.868	50.377	49.868	201
Male	49.114	48.132	48.623	48.132	194
Total	100	98	99	98	395

- **Null Hypothesis**: Gender and Education Level are independent.
- **Alternative Hypothesis**: Gender and Education Level are dependent

$$50.886 = \frac{100 \times 201}{395}$$

$$x^2 = \frac{(60 - 50.886)^2}{50.886} + \frac{(54 - 49.868)^2}{49.868} + \cdots$$
$$= 8.006$$

Chi-square test

- **Chi-square test** for categorial variables: Is gender independent of education level?

Percentage Points of the Chi-Square Distribution

Degrees of Freedom	Probability of a larger value of x [2]								
	0.99	0.95	0.90	0.75	0.50	0.25	0.10	0.05	0.01
1	0.000	0.004	0.016	0.102	0.455	1.32	2.71	3.84	6.63
2	0.020	0.103	0.211	0.575	1.386	2.77	4.61	5.99	9.21
3	0.115	0.352	0.584	1.212	2.366	4.11	6.25	7.81	11.34
4	0.297	0.711	1.064	1.923	3.357	5.39	7.78	9.49	13.28
5	0.554	1.145	1.610	2.675	4.351	6.63	9.24	11.07	15.09
6	0.872	1.635	2.204	3.455	5.348	7.84	10.64	12.59	16.81
7	1.239	2.167	2.833	4.255	6.346	9.04	12.02	14.07	18.48
8	1.647	2.733	3.490	5.071	7.344	10.22	13.36	15.51	20.09
9	2.088	3.325	4.168	5.899	8.343	11.39	14.68	16.92	21.67
10	2.558	3.940	4.865	6.737	9.342	12.55	15.99	18.31	23.21
11	3.053	4.575	5.578	7.584	10.341	13.70	17.28	19.68	24.72
12	3.571	5.226	6.304	8.438	11.340	14.85	18.55	21.03	26.22
13	4.107	5.892	7.042	9.299	12.340	15.98	19.81	22.36	27.69
14	4.660	6.571	7.790	10.165	13.339	17.12	21.06	23.68	29.14
15	5.229	7.261	8.547	11.037	14.339	18.25	22.31	25.00	30.58
16	5.812	7.962	9.312	11.912	15.338	19.37	23.54	26.30	32.00
17	6.408	8.672	10.085	12.792	16.338	20.49	24.77	27.59	33.41
18	7.015	9.390	10.865	13.675	17.338	21.60	25.99	28.87	34.80
19	7.633	10.117	11.651	14.562	18.338	22.72	27.20	30.14	36.19
20	8.260	10.851	12.443	15.452	19.337	23.83	28.41	31.41	37.57
22	9.542	12.338	14.041	17.240	21.337	26.04	30.81	33.92	40.29
24	10.856	13.848	15.659	19.037	23.337	28.24	33.20	36.42	42.98
26	12.198	15.379	17.292	20.843	25.336	30.43	35.56	38.89	45.64
28	13.565	16.928	18.939	22.657	27.336	32.62	37.92	41.34	48.28
30	14.953	18.493	20.599	24.478	29.336	34.80	40.26	43.77	50.89
40	22.164	26.509	29.051	33.660	39.335	45.62	51.80	55.76	63.69
50	27.707	34.764	37.689	42.942	49.335	56.33	63.17	67.50	76.15
60	37.485	43.188	46.459	52.294	59.335	66.98	74.40	79.08	88.38

$$x^2 = 8.006$$

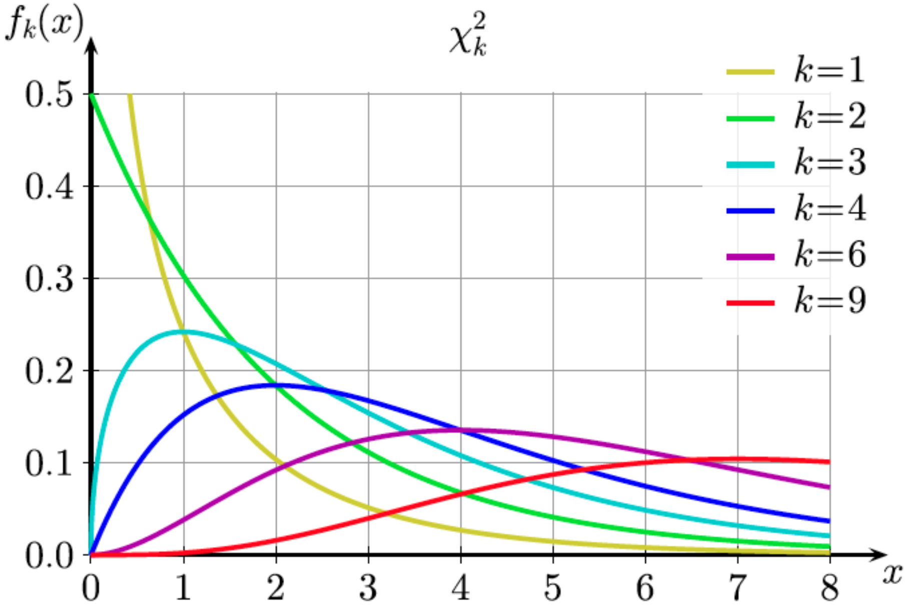
The degree of freedom:

$$(r-1)(c-1) = 3$$

The critical value of x^2 at a 5% level of significance: 7.815

Chi-square test

- **Chi-square test** for categorial variables: Is gender independent of education level?



- $x^2 = 8.006 > 7.815$ (The critical value of 2 with 3 degree of freedom)
- Reject the null hypothesis and conclude that the education level depends on gender at a 5% level of significance

Correlation Coefficient

- **Correlation Coefficient**, r , also called Pearson correlation coefficient
  - Measures the strength and the direction of a linear relationship between two variables

$$r = \frac{n\sum(xy) - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}}$$

  - The value of r is such that $-1 \ < r < +1$
    - **Positive correlation**: If x and y have a strong positive linear correlation, r is close to +1
    - **Negative correlation**: If x and y have a strong negative linear correlation, r is close to -1.
    - **No correlation**: If there is no linear correlation or a weak linear correlation, r is close to 0.

Coefficient of Determination

- **Coefficient of determination**
  - The proportion of the variance (fluctuation) of one variable that is predictable from the other variable.
  - $0 < r^2 < 1$ denotes the strength of the linear association between x and y.
  - The coefficient of determination is a measure of how well the regression line represents the data. If the regression line passes exactly through every point on the scatter plot, it would be able to explain all of the variation. The further the line is away from the points, the less it is able to explain.

Coefficient of Determination

	x	y	xy	x^2	y^2
	313000	1340	419420000	97969000000	1795600
	2384000	3650	8701600000	5.68346E+12	13322500
	342000	1930	660060000	1.16964E+11	3724900
	420000	2000	840000000	1.764E+11	4000000
	550000	1940	1067000000	3.025E+11	3763600
	490000	880	431200000	2.401E+11	774400
	335000	1350	452250000	1.12225E+11	1822500
	482000	2710	1306220000	2.32324E+11	7344100
	452500	2430	1099575000	2.04756E+11	5904900
	640000	1520	972800000	4.096E+11	2310400
	463000	1710	791730000	2.14369E+11	2924100
	1400000	2920	4088000000	1.96E+12	8526400
	588500	2330	1371205000	3.46332E+11	5428900
	365000	1090	397850000	1.33225E+11	1188100
	1200000	2910	3492000000	1.44E+12	8468100
	242500	1200	291000000	58806250000	1440000
	419000	1570	657830000	1.75561E+11	2464900
	285000	2200	627000000	81225000000	4840000
	367500	3110	1142925000	1.35056E+11	9672100
Sum	11739000	38790	28809665000	1.21209E+13	89715500

$$r = \frac{n\sum(xy) - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}}$$
$$= 0.676747624$$

$$r^2 = (0.676747624)^2 = 0.457987347$$

Coefficient of Determination

- Regression Sum of Squares (SSR) (or explained sum of squares)

$$SSR = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

- Residual Sum of Squares (RSS)

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} e_i^2$$

$TSS = RSS + SSR?$

- Total Sum of Squares (TSS)

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

- R^2 is defined as

$$R^2 = 1 - \frac{RSS}{TSS}$$

Tuple-level Integration

- Duplicates
  - Two or more rows (i.e., tuples) refer to the same object.
- Inconsistent update
  - Duplicated records are not updated simultaneously.
- Issues with tuple-level integration
  - Formatting convertors
  - Different naming conventions
  - ...
- Tuple Matching methods
  - String Matching
  - Data Matching

String Matching

- Problems:
  - Given two sets of strings X and Y , find all pairs of strings (x, y), where $x \in X$ and $y \in Y$, such that x and y refer to the same entity.

Set X
x_1=Dave Smith
x_2=Joe Wilson
x_3=Dan Smith

(a)

Set Y
y_1=David D. Smith
y_2=Daniel W. Smith

(b)

Matches
(x_1, y_1)
(x_3, y_2)

(c)

Figure is from chapter 4 of "Principles of Data Integration"

String Matching

- Methods: Similarity Measures
  - **Sequence-based Similarity Measures**: View strings as sequences of characters, compute a cost of transforming one string into the other.
    - Edit Distance
    - The Needleman-Wunch measure
    - The Affine Gap measure
    - The Smith-Waterman measure
  - **Set-based Similarity Measures**: View strings as sets or multi-sets of tokens and use set-related properties to compute similarity scores.
    - The Overlap measure
    - The TF/IDF measure
  - **Hybrid Similarity Measures**: combines sequence-based and set-based measures
    - The Generalised Jaccard measure
    - The Soft TF/IDF measure
  - **Phonetic Similarity Measure**: matches strings based on their sound.

Edit Distance

- The minimum edit distance between two strings
- Is minimum number of editing operations
  - Insertion
  - Deletion
  - Substitution
- Needed to transform one to another.

Edit Distance

$$d(i,j)=\min\begin{cases} d(i-1,j-1) & \text{if } x_i=y_j \quad \text{// copy}\\ d(i-1,j-1)+1 & \text{if } x_i<>y_j \quad \text{// substitute}\\ d(i-1,j)+1 & \text{// delete } x_i\\ d(i,j-1)+1 & \text{// insert } y_j \end{cases}$$

(a)

$$d(i,j)=\min\begin{cases} d(i-1,j-1)+c(x_i,y_j) & \text{// copy or substitute}\\ d(i-1,j)+1 & \text{// delete } x_i\\ d(i,j-1)+1 & \text{// insert } y_j \end{cases}$$

$$c(x_i,y_j)=\begin{cases} 0 & \text{if } x_i=y_j\\ 1 & \text{otherwise} \end{cases}$$

(b)

- Transform string $x_1, \dots, x_i, \dots, x_n$ to $y_1, \dots, y_j, \dots, y_m$
  - Transform $x_1, \dots, x_{i-1}$ into $y_1, \dots, y_{j-1}$ if $x_i = y_j$
  - Transform $x_1, \dots, x_{i-1}$ into $y_1, \dots, y_{j-1}$, then substituting x_i with y_i if $x_i \neq y_i$
  - Deleting x_i, then transform $x_1, \dots, x_{i-1}$ into $y_1, \dots, y_j$
  - Transform $x_1, \dots, x_i$ into $y_1, \dots, y_{j-1}$, then insert y_j

Edit Distance

		y_0	y_1	y_2	y_3	y_4
			d	**a**	**v**	**e**
x_0		0	1	2	3	4
x_1	**d**	1	0	1		
x_2	**v**	2				
x_3	**a**	3				

(a)

		y_0	y_1	y_2	y_3	y_4
			d	**a**	**v**	**e**
x_0		0	1	2	3	4
x_1	**d**	1	0	1	2	3
x_2	**v**	2	1	1	1	2
x_3	**a**	3	2	1	2	2

(b)

x=d – v a
| | | |
y=d a v e

Substitute a with e
Insert a (after d)

(c)

$$d(i,j) = \min \begin{cases} d(i-1, j-1) + c(x_i, y_j) & \text{// copy or substitute} \\ d(i-1, j) + 1 & \text{// delete } x_i \\ d(i, j-1) + 1 & \text{// insert } y_j \end{cases}$$

$$c(x_i, y_j) = 0 \text{ if } x_i = y_j$$
$$\quad 1 \text{ otherwise}$$

Figure is from chapter 4 of "Principles of Data Integration"

Needleman-Wunch Measure

```
d--va
|   ||
deeve
```

(a)

	d	a	v	e
d	2	–1	–1	–1
a	–1	2	–1	–1
v	–1	–1	2	–1
e	–1	–1	–1	2

(b)

$c_g = 1$

$$s(i, j) = \max \begin{cases} s(i\text{–}1, j\text{–}1) + c(x_i, y_j) \\ s(i\text{–}1, j) - c_g \\ s(i, j\text{–}1) - c_g \end{cases}$$

$$s(0, j) = -jc_g$$
$$s(i, 0) = -ic_g$$

(a)

		d	e	e	v	e
	0	–1	–2	–3	–4	–5
d	–1	2	1	0	–1	–2
v	–2	1	1	0	2	1
a	–3	0	0	0	1	1

(b)

```
d--va
|   ||
deeve
```

(c)

Figure is from chapter 4 of “Principles of Data Integration”

TF/IDF Measure

$x = aab \Rightarrow B_x = \{a, a, b\}$
$y = ac \Rightarrow B_y = \{a, c\}$
$z = a \Rightarrow B_z = \{a\}$

(a)

$tf(a, x) = 2$
$tf(b, x) = 1$
...
$tf(c, z) = 0$

$idf(a) = 3/3 = 1$
$idf(b) = 3/1 = 3$
$idf(c) = 3/1 = 3$

(b)

	a	b	c
v_x	2	3	0
v_y	3	0	3
v_z	3	0	0

(c)

$$s(p, q) = \frac{\sum_{t \in T} v_p(t) \cdot v_q(t)}{\sqrt{\sum_{t \in T} v_p(t)^2} \cdot \sqrt{\sum_{t \in T} v_q(t)^2}}$$

$$s(x, y) = \frac{2 \cdot 3}{\sqrt{2^2 + 3^2}\sqrt{3^2 + 3^2}}$$

Figure is from chapter 4 of “Principles of Data Integration”

Data Matching

Table X

	Name	Phone	City	State
x_1	Dave Smith	(608) 395 9462	Madison	WI
x_2	Joe Wilson	(408) 123 4265	San Jose	CA
x_3	Dan Smith	(608) 256 1212	Middleton	WI

(a)

Table Y

	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

(b)

Matches

(x_1, y_1)
(x_3, y_2)

(c)

- Data Matching is challenging due to variations in
  - formatting conventions
  - use of abbreviations, shortening
  - different naming conventions,
  - omissions
  - errors

Figure is from chapter 7 of "Principles of Data Integration"

Data Matching

Table X

	Name	Phone	City	State
x_1	Dave Smith	(608) 395 9462	Madison	WI
x_2	Joe Wilson	(408) 123 4265	San Jose	CA
x_3	Dan Smith	(608) 256 1212	Middleton	WI

(a)

Table Y

	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

(b)

Matches
(x_1, y_1)
(x_3, y_2)

(c)

- Methods
  - Rules-based method
  - Learning-based methods
    - Supervised learning
    - Clustering
    - probabilistic approach

Figure is from chapter 7 of “Principles of Data Integration”

Rule-based Data Matching

Table X

	Name	Phone	City	State
x_1	Dave Smith	(608) 395 9462	Madison	WI
x_2	Joe Wilson	(408) 123 4265	San Jose	CA
x_3	Dan Smith	(608) 256 1212	Middleton	WI

(a)

Table Y

	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

(b)

Matches

(x_1, y_1)
(x_3, y_2)

(c)

- A linearly weighted combination of the individual similarity scores between x and y:

$$sim(x,y) = \sum_{i=1}^{n} \alpha_i sim_i(x,y)$$

- A rule for the example in the figure

$$sim(x,y) = 0.3s_{name}(x,y) + 0.3s_{phone}(x,y) + 0.1s_{city}(x,y) + 0.3s_{state}(x,y)$$

Figure is from chapter 7 of “Principles of Data Integration”

Rule-based Data Matching

Table X

	Name	Phone	City	State
x_1	Dave Smith	(608) 395 9462	Madison	WI
x_2	Joe Wilson	(408) 123 4265	San Jose	CA
x_3	Dan Smith	(608) 256 1212	Middleton	WI

(a)

Table Y

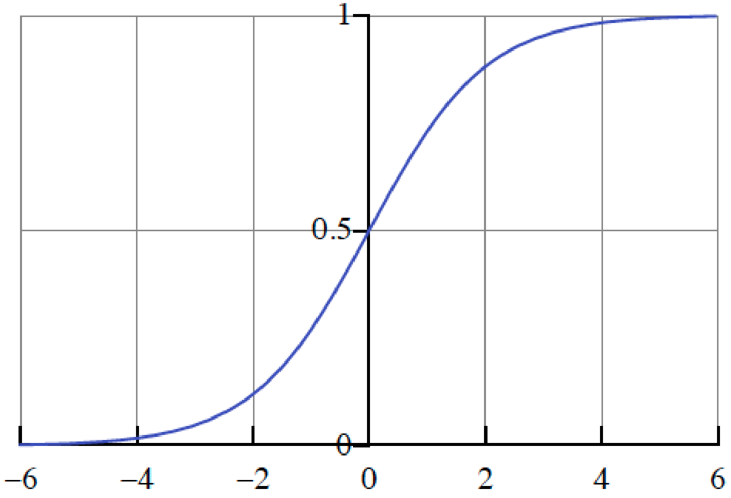
	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

(b)

Matches

(x_1, y_1)
(x_3, y_2)

(c)



$$sim(x, y) = \frac{1}{1 + e^{-z}}$$

where

$$z = \sum_{i}^{n} \alpha_i sim_i(x, y)$$

Figure is from chapter 7 of "Principles of Data Integration"

Learning-based Data Matching

- **Supervised learning**: learn a matching model with training data

$$T = \{(x_1, y_1, l_1), (x_2, y_2, l_2), \dots, (x_n, y_n, l_n)\}$$

where (x_i, y_i) indicates a tuple pair, and l_i indicates the Boolean label.

  - Define a set of features $f_1, f_2, \dots, f_m$
  - Convert each training sample (x_i, y_i, l_i) into a feature vector

$$(< f_1(x_i, y_i), f_2(x_i, y_i), \dots, f_m(x_i, y_i) >, c_i)$$

  - Apply supervised learning algorithms

Learning-based Data Matching

- **Supervised learning**: learn a matching model with training data

<a_1 = (Mike Williams, (425) 247 4893, Seattle, WA), b_1 = (M. Williams, 247 4893, Redmond, WA), yes>
<a_2 = (Richard Pike, (414) 256 1257, Milwaukee, WI), b_2 = (R. Pike, 256 1237, Milwaukee, WI), yes>
<a_3 = (Jane McCain, (206) 111 4215, Renton, WA), b_3 = (J. M. McCain, 112 5200, Renton, WA), no>

(a)

match names   match phones   match cities   match states   check area code against city

$v_1 = <[s_1(a_1,b_1), s_2(a_1,b_1), s_3(a_1,b_1), s_4(a_1,b_1), s_5(a_1,b_1), s_6(a_1,b_1)], 1>$

$v_2 = <[s_1(a_2,b_2), s_2(a_2,b_2), s_3(a_2,b_2), s_4(a_2,b_2), s_5(a_2,b_2), s_6(a_2,b_2)], 1>$

$v_3 = <[s_1(a_3,b_3), s_2(a_3,b_3), s_3(a_3,b_3), s_4(a_3,b_3), s_5(a_3,b_3), s_6(a_3,b_3)], 0>$

(b)

Figure is from chapter 7 of “Principles of Data Integration”

Learning-based Data Matching

- **Supervised learning**: learn a matching model with training data

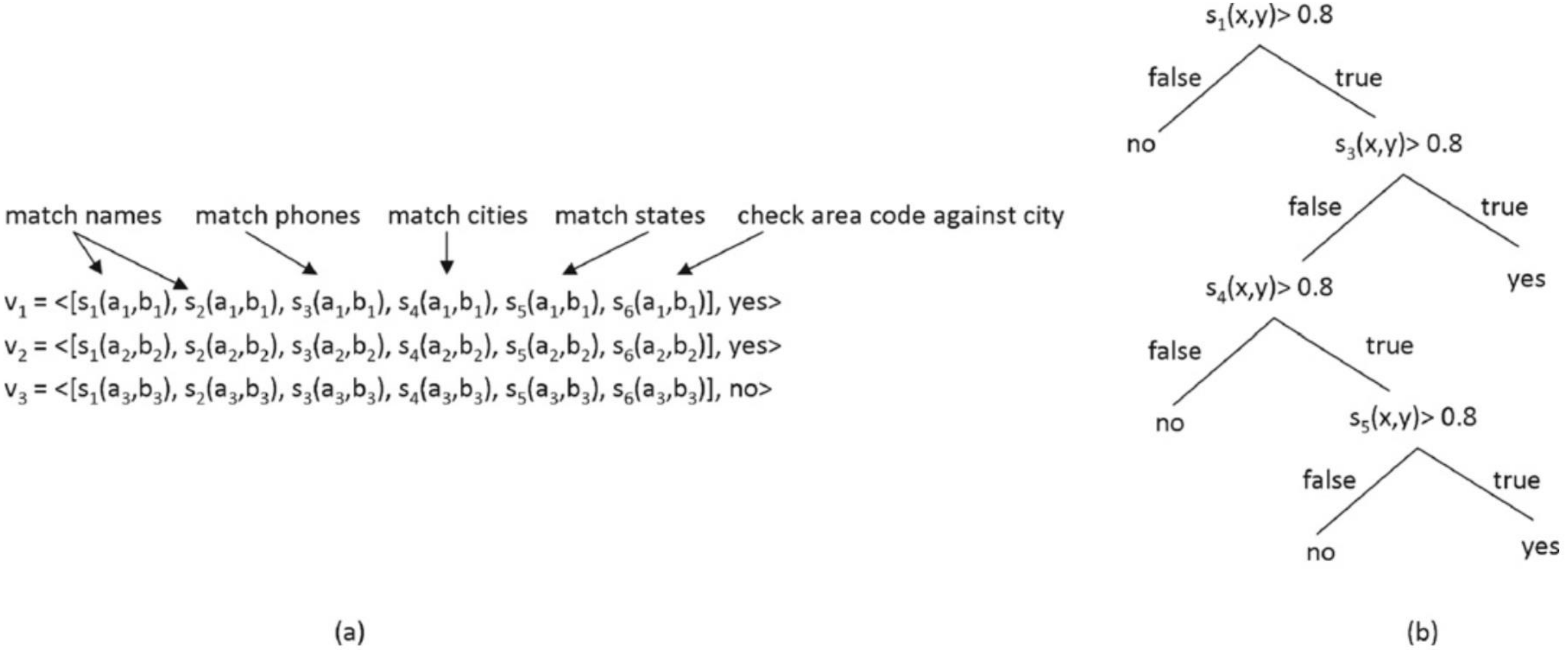


Figure is from chapter 7 of "Principles of Data Integration"

Learning-based Data Matching

- **Clustering approach**: tuples in the same cluster match
  - The problem of constructing entities(that is, clusters): only tuples within a cluster match.
  - An iterative process: leverage what we have known so far (in the previous iterations) to build “better” entities.
  - Generating a canonical tuple: “merge” all matching tuples within each cluster to construct an “entity profile”

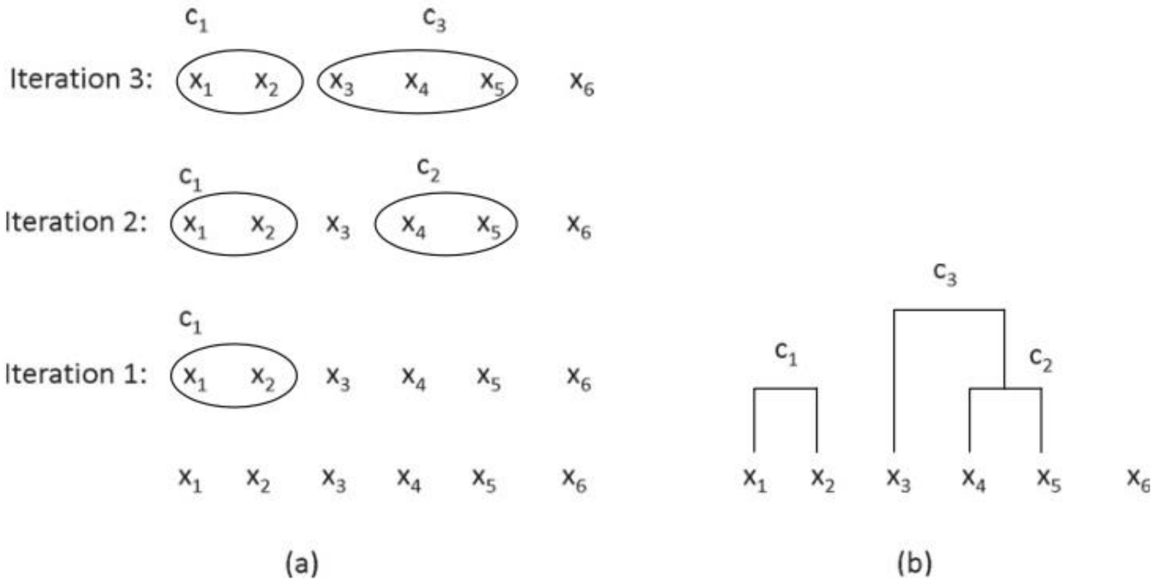


Figure is from chapter 7 of “Principles of Data Integration”

Summary & To-do List

- Please download and read materials provided on Moodle.
- Review content learnt from Week10.

- Assessments
  - Read the tasks in Assessment 2 and continue to work on it.

- Next week: Data Validation