澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Understand the Data

CISC7204: DATA SCIENCE & VISUALIZATION

Derek F. Wong

NLP^{2}CT – *Natural Language Processing & Portuguese-Chinese Machine Translation Research Group*

derekfw@um.edu.mo

E11-4010 (Ext: 4478)

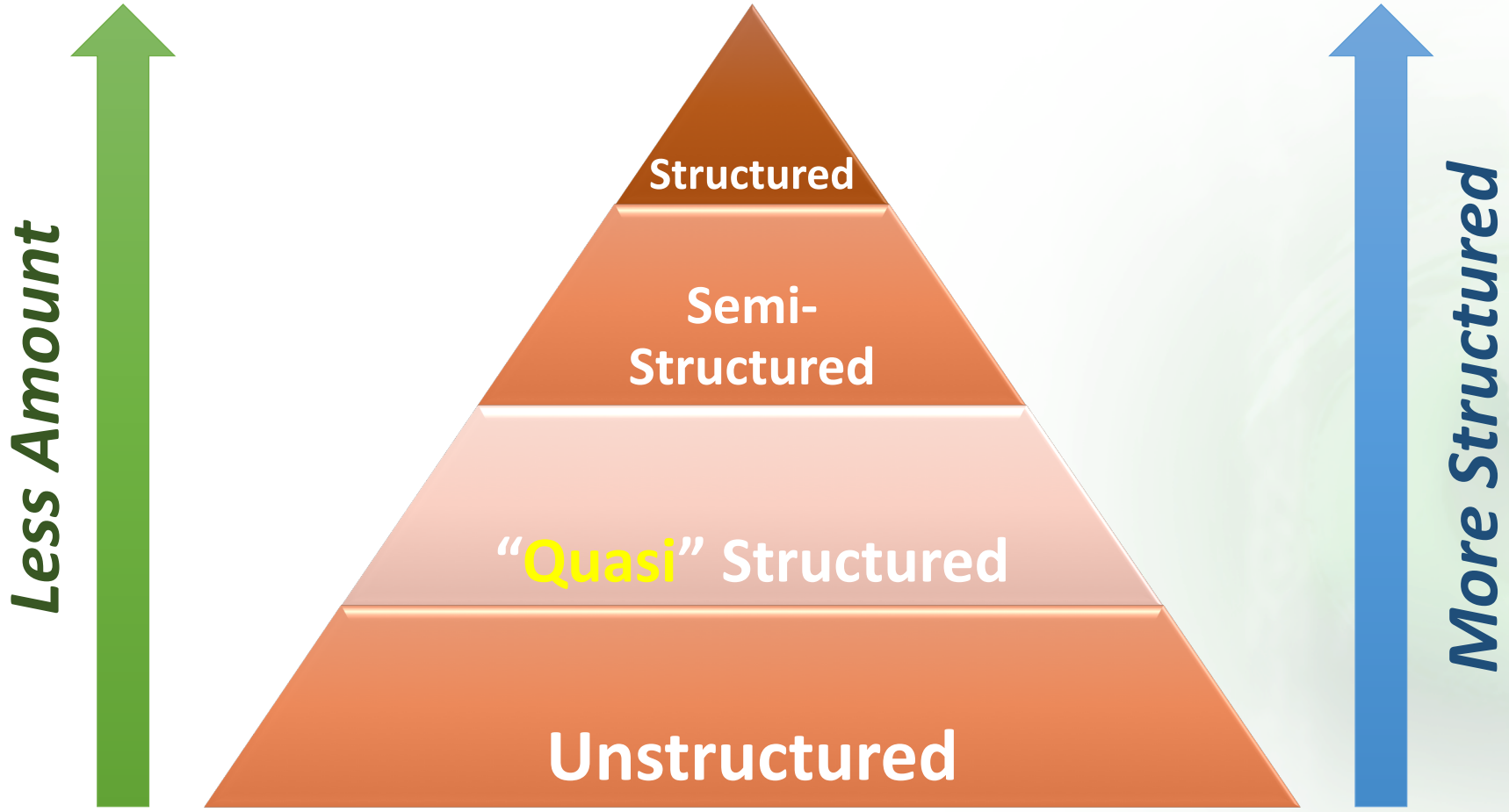
Office Hours: Thu – 16:00~17:30, Fri 11:00~12:30

Natural Language Processing & Portuguese – Chinese Machine Translation Laboratory
自然語言處理與中葡機器翻譯實驗室

Content

- Data Sources & Structures
- Relational Data
  - Entity Relationships
  - Relational Operations
- Data Types
  - Analysis & Representation

Structures of Data



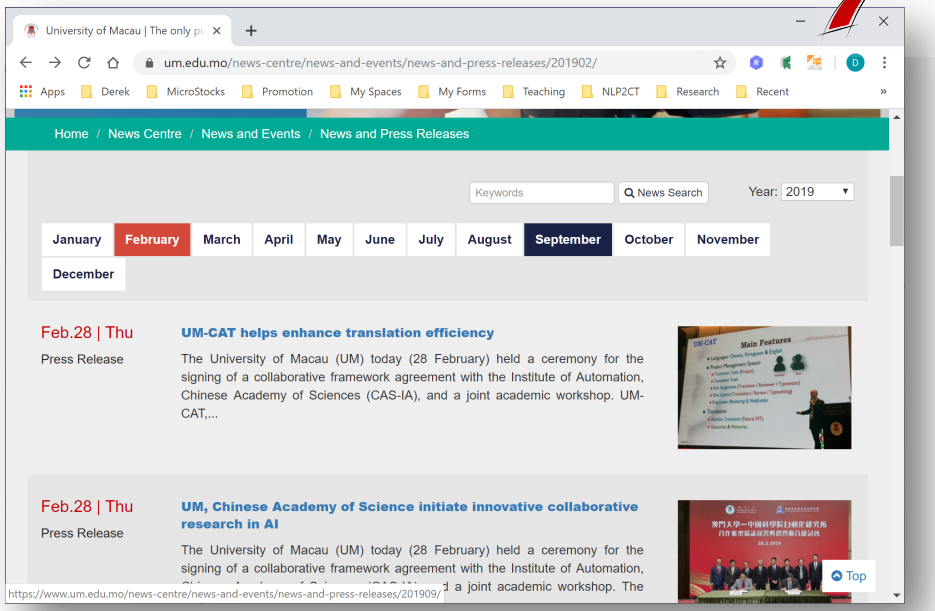
Structured Data

Data containing a *defined data type*, *format*, and *structure*

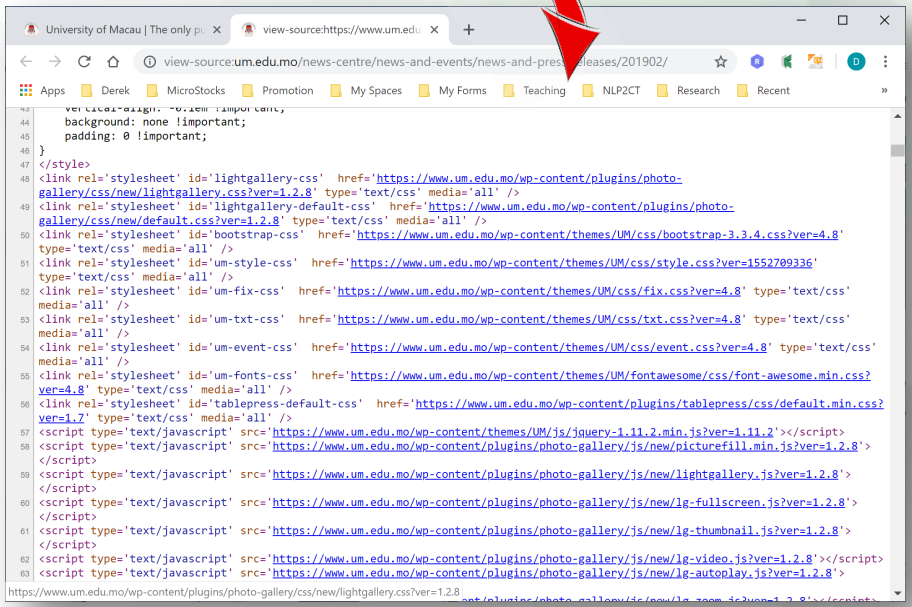
ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Semi-Structured Data

Textual data files with a *discernible pattern* that *enables parsing* (such as *eXtensible Markup Language* [XML] data files that are self-describing and defined by an XML schema)

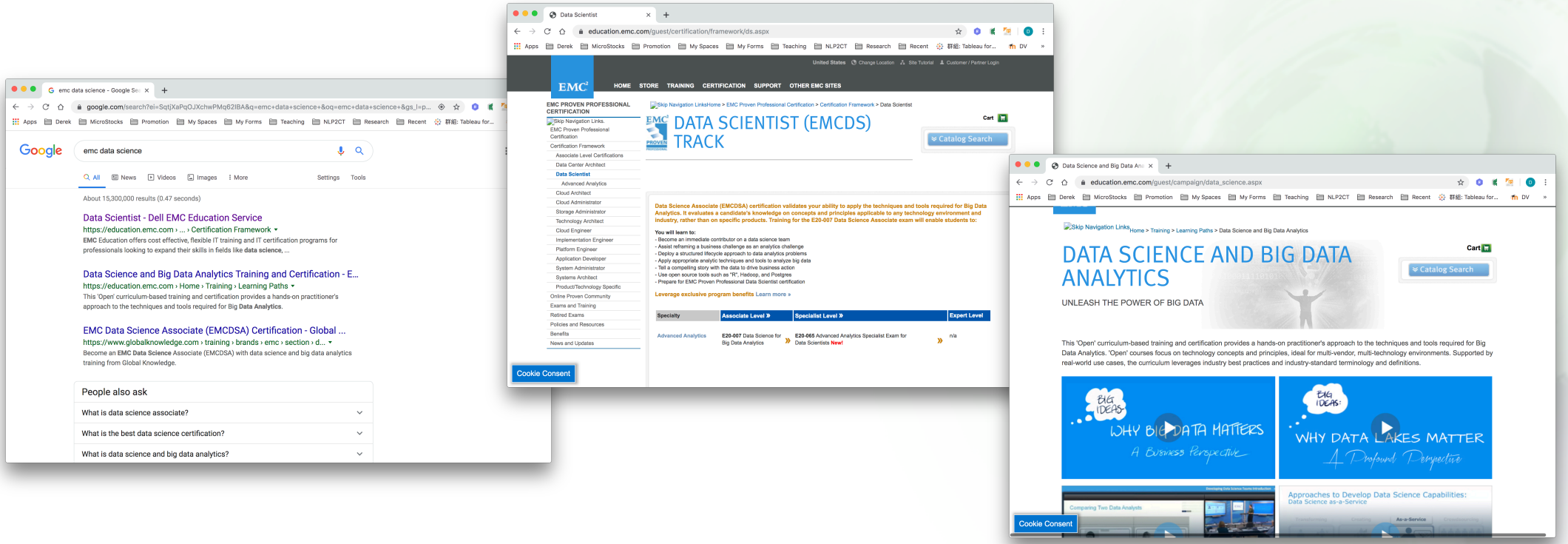


View Page Source



Quasi Structured Data

Textual data with *erratic data formats* that can be formatted with *effort*, *tools*, and *time* (for instance, web clickstream data that may contain *inconsistencies* in data values and formats)



Unstructured Data

Data that has *no inherent structure*, which may include *text* documents, *PDFs*, *images*, and *video*

Subject: **Data Vis Class**
Date: 28 August 2019
To: Peter Chow

Event: DV Class
Date: Aug-28-2019
Start: 19:00
End: 22:00
Where: E4-G003

Hi Peter, we've now scheduled the DV Class.

It will be in Room E4-G003 Wed from 19:00-22:00.

Create New Calendar Entry

- General Office

Relational Data

Entity Description

The Basic Relation

The Table

The term technical term "*relation*" can be interchanged with the standard notion we have of "*tabular data*," say an instance of a "*Person*" relation

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

The Basic Relation

The Table

The term technical term "*relation*" can be interchanged with the standard notion we have of "*tabular data*," say an instance of a "*Person*" relation

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Rows are called *tuples* (or *records*), represent a single instance of this relation, and *must be unique*

The Basic Relation

The Table

The term technical term "*relation*" can be interchanged with the standard notion we have of "*tabular data*," say an instance of a "*Person*" relation

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Columns are called *attributes* (or *fields*), specify some element contained by each of the tuples

Multiple Tables or Relations

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Role

ID	Role
1	Instructor
2	TA
3	Student

Primary Keys

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Role

ID	Role
1	Instructor
2	TA
3	Student

Primary key: *unique ID* for every tuple in a relation (i.e. every row in the table), each relation must *have exactly one* primary key

Foreign Keys

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Role

ID	Role
1	Instructor
2	TA
3	Student

A **foreign key** is *an attribute* that *points to* the *primary key* of another relation

If you *delete* a *primary key*, need to *delete all foreign keys* pointing to it

Indexes

Indexes are created as ways to "*quickly*" access elements of a table

For example, consider finding people with last name "Ao": no option but just scan through the whole dataset: $O(n)$ operations

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Indexes

Person

Location	ID	First Name	Last Name	Role ID
0	1	Derek	Wong	1
100	2	Erin	Lai	2
200	3	Man	Lee	3
300	4	Judy	Leong	3
400	5	Harris	Guo	3
500	6	Steven	Hong	3

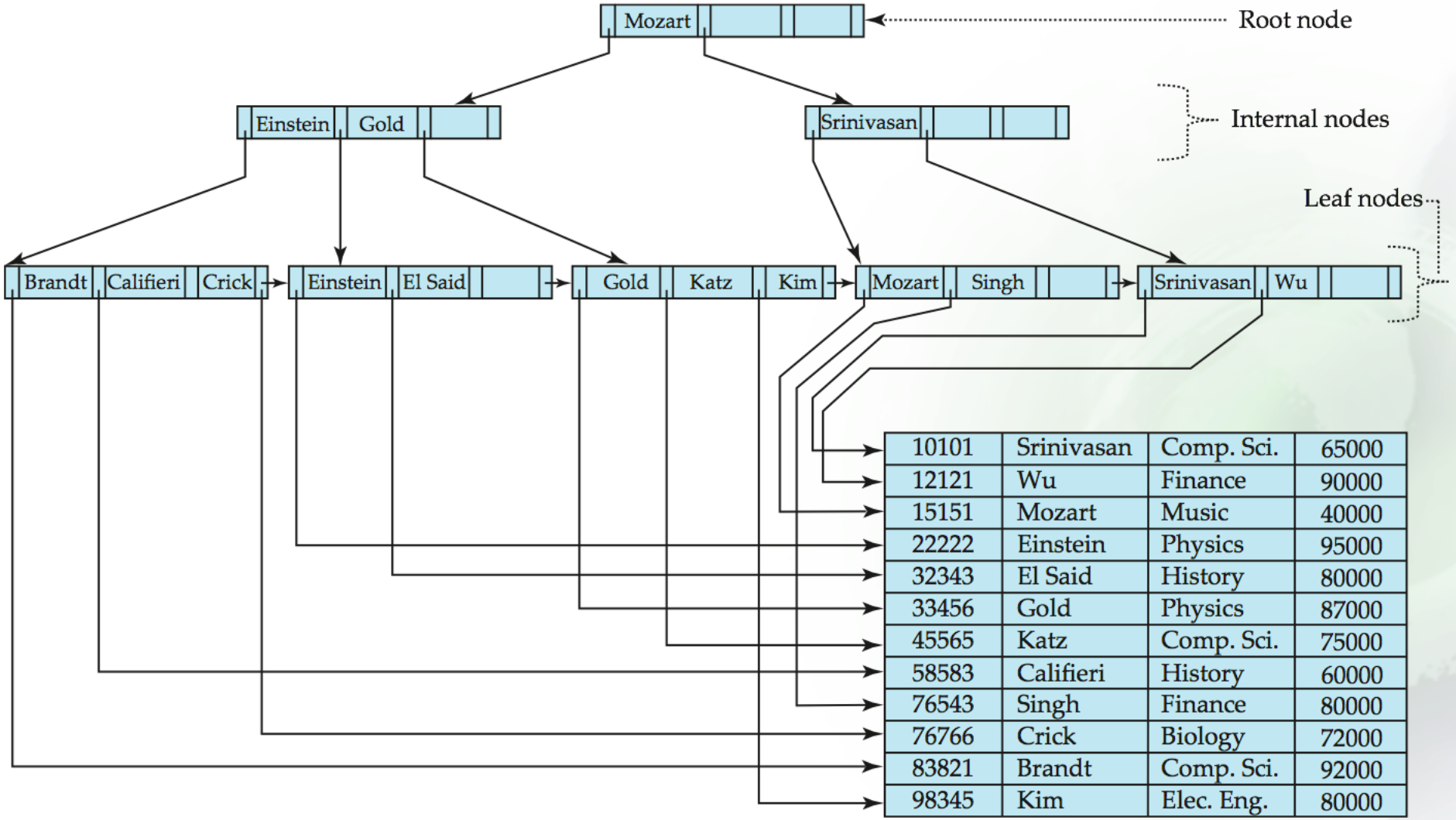
First Name Index

First Name	Location
Derek	0
Erin	100
Harris	400
Judy	300
Man	200
Steven	500

Think of *an index as a separate sorted table* containing the *indexed column* and the *tuple location*: searching for value takes $O(\log n)$ time

In practice, use data structure like a ***B+ tree*** or several others

Example B+ Tree



10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	80000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	60000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Indexes

- The *primary key always* has an *index* associated with it (so you can think of primary keys themselves as *always being a fast way to access data*)
- Indexes *don't have to be on a single column*, can have an index *over multiple columns* (with some ordering)

Entity Relationships

Entity Relationships

Several types of *inter-table* relationships

- One-to-one
- One-to-zero/one
- One-to-many (and many-to-one)
- Many-to-many

- These *relate one (or more) rows* in a table with *one (or more) rows* in another table, *via a foreign key*

Note that these relationships are really *between* the "*entities*" that the tables represent, but we won't formalize this beyond the basic intuition

One-to-Many Relationships

We have already seen a one-to-many relationship: one *role* can be shared by *many people*, denoted as follows

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Role

ID	Role
1	Instructor
2	TA
3	Student

One-to-One Relationships

In a *true one-to-one relationship* spanning multiple tables, *each row in a table has exactly one row in another table*

Not very common to break these *across multiple tables*, as you may as well just add another attribute to an existing table, but it is possible

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3

⋮

Account ID

Person ID	Account ID
1	derekwong
2	erinlai
3	manlee

⋮

One-to-Zero/One Relationships

More common in databases is to find "one-to-zero/one" relationships broken across multiple tables

Consider adding a "Grades" table to our database: *each person can have at most one tuple in the grades table*

Grades

Person ID	HW1	HW2
4	90	85
5	82	90
6	75	85

Bars and *circles* denote "*mandatory*" versus "*option*" relationships (we won't worry about these, just know that there is notation for them)

Many-to-Many Relationships

Creating a grades table as done before *is a bit cumbersome*, because we need to *keep adding columns to the table*, *null* entries if someone doesn't do the homework

Alternatively, consider *adding two tables*, a "*homework*" table that represents information about each homework, and an *associative table* that links *homeworks* to people

Homework

ID	Name	Q1	Q2
1	HW1	60	40
2	HW2	70	30

Person Homework

Person ID	HW ID	Score
4	1	90
4	2	85
5	1	82
5	2	90

Associative Tables

Person Homework

Person ID	HW ID	Score
4	1	90
4	2	85
5	1	82
5	2	90

What is the primary key of this table? What are foreign keys?

Which indexes would you want to create on this table?

Many-to-Many Relationships

Setups like this encode *many-to-many relationships*: each person can have multiple homeworks, and each homework can be done by multiple people

We could also write this in terms of relationships specified by the *associative table*, *but this is not really correct*, as it is mixing up the underlying relationships with how they are stored in a database

Relational Operations

Filtering

Filtering or “Query” operations retrieve a *part of data* from a table to form a *new table* containing the data

Person

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Filtering

Two basic of query operators: 1) *Selection*

Person

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Selection

Students

ID	First Name	Last Name	Role
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Non-Students

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Chen	TA

Selection of "*tuples*"

Filtering

Two basic of query operators: 1) Selection; and 2) *Projection*

Person

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Projection

Person Name

ID	First Name	Last Name
1	Derek	Wong
2	Erin	Lai
3	Man	Lee
4	Judy	Leong
5	Harris	Guo
6	Steven	Hong

Selection of "*Columns*"

Filtering

The *two operations can be used together* to extract necessary data

ID	First Name	Last Name	Role
1	Derek	Wong	Instructor
2	Erin	Lai	TA
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Selection

ID	First Name	Last Name	Role
3	Man	Lee	Student
4	Judy	Leong	Student
5	Harris	Guo	Student
6	Steven	Hong	Student

Projection

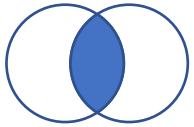
ID	First Name	Last Name
3	Man	Lee
4	Judy	Leong
5	Harris	Guo
6	Steven	Hong

Selection + Projection

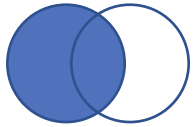
Joins

Join operations *merge multiple tables* into *a single relation* (can be then saved as *a new table* or just *directly used*)

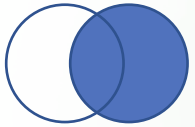
Four typical types of join operators:



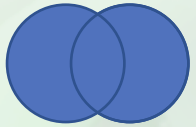
Inner



Left



Right



Outer

Join two tables on *columns* from each table, where these columns *specify* which *rows are kept*

Ex: Joining Person and Grades

Consider *joining* two tables, *Person* and *Grades*, on *ID / Person ID*

Person

ID	First Name	Last Name	Role ID
1	Derek	Wong	1
2	Erin	Lai	2
3	Man	Lee	3
4	Judy	Leong	3
5	Harris	Guo	3
6	Steven	Hong	3

Grades

Person ID	HW1	HW2
4	90	85
5	82	90
6	75	85
7	90	95

Ex: Joining Person and Grades

Inner joins

Join two tables where we only return the rows where the *two joined columns* contain the *same value*

ID	First Name	Last Name	Role ID	HW1	HW2
4	Judy	Leong	3	90	85
5	Harris	Guo	3	82	90
6	Steven	Hong	3	75	85

Only these *three rows* have an entry in "Person" and an entry in "Grades"

Left Joins

Keep all rows of the *left table*, add *entries from right table* that match the corresponding columns

ID	First Name	Last Name	Role ID	HW1	HW2
1	Derek	Wong	1	NULL	NULL
2	Erin	Lai	2	NULL	NULL
3	Man	Lee	3	NULL	NULL
4	Judy	Leong	3	90	85
5	Harris	Guo	3	82	90
6	Steven	Hong	3	75	85

Right Joins

Like a left join but with the *roles* of the tables *reversed*

ID	First Name	Last Name	Role ID	HW1	HW2
4	Judy	Leong	3	90	85
5	Harris	Guo	3	82	90
6	Steven	Hong	3	75	85
7	NULL	NULL	NULL	90	95

Outer Joins

Return *all rows* from both *left* and *right* join

ID	First Name	Last Name	Role ID	HW1	HW2
1	Derek	Wong	1	NULL	NULL
2	Erin	Lai	2	NULL	NULL
3	Man	Lee	3	NULL	NULL
4	Judy	Leong	3	90	85
5	Harris	Guo	3	82	90
6	Steven	Hong	3	75	85
7	NULL	NULL	NULL	90	95

Data Types

Data Analysis vs. Underlying Representation of Data

Types of Data

Understanding the *different types* of data makes it possible to

- Identify the type of data *test* for the *analysis*
- Identify the *level of complexity* and *performance* of the *mathematical tools* to be applied

Quantitative or Qualitative?

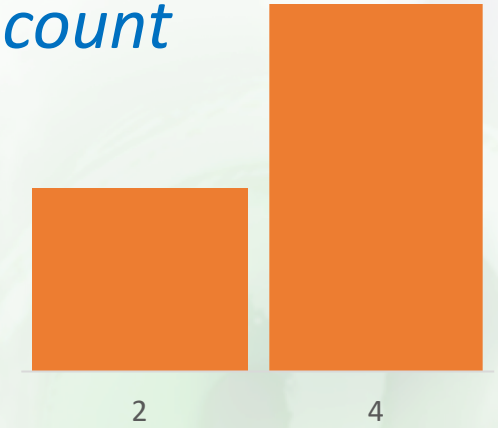
- *Quantitative data* consist of values representing *counts* or *measurements*
  - *e.g. Year in school*

- *Qualitative* (or non-numeric) data consist of values that can be placed into *nonnumeric categories*
  - *e.g. Color (red, yellow, green)*

Types of Data

Quantitative

- *Numerical* values representing counts or measures
- Something we can *measure* with a *tool* or a *scale* or *count*
- We can *compare these values* on a number line
  - *2 pounds is less than 4 pounds*



- You can take a *mathematical average* of these values, i.e. can be used in computations
  - *e.g. weight*
  - *e.g. number of students in a class*

Types of Data

Qualitative

- *Non-numerical* in nature (but could be *coded* as a number, so be careful)
  - *e.g. low=1, med=2, high=3 (still qualitative)*
- Could be considered a *label* in some cases
  - *e.g. Candy color (red, yellow, green)*
  - *e.g. Numbers on a baseball uniform*
- #90 *is not* "*larger than*" #45 in the *mathematical* sense, they are *just a label*
  - *e.g. ID (34B, 67AA, 19G, …)*
  - *e.g. Education level (HS, 2-yr, 4-yr, MS, PhD) representing counts or measures*

Types of Data

Qualitative

- *Cannot use* meaningfully in a *computation*…
  - Can you take the *average* of the *observed color of candies*? No, it is non-numerical!
    - *Yellow, yellow, red, green, yellow, red…*
  - e.g. ID #s 56, 213, 788,… *Average ID*? No!
  - If *variable* is represented by *numbers* (as with IDs), ask yourself *if an average makes sense*… if not, then it is *qualitative not quantitative*

Types of Data

Olympic Medals

Quantitative

- *Number* of medals won by China in all years

Qualitative

- Medal *Type*:
  - *Gold/Silver/Bronze*

Summer Olympic China medalists 1984 ~ 2016

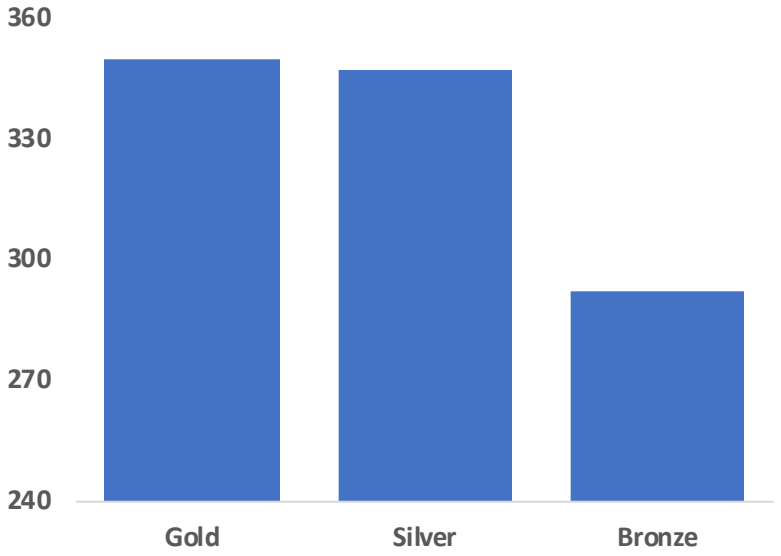
Name	Sex	NOC	Games	Year	City	Sport	Event	Medal
Ye Qiaobo	F	CHN	1992 Winter	1992	Albertville	Speed Skating	Speed Skating Women's	Silver
Chen Jing	F	CHN	2004 Summer	2004	Athina	Volleyball	Volleyball Women's Volleyball	Gold
Du Li	F	CHN	2004 Summer	2004	Athina	Shooting	Shooting Women's Air Rifle	Gold
Gao E	F	CHN	2004 Summer	2004	Athina	Shooting	Shooting Women's Double Trap	Bronze
Gao Feng	F	CHN	2004 Summer	2004	Athina	Judo	Judo Women's Extra-Lightweight	Bronze

Types of Data

Olympic Medals

Quantitative

- *Number* of medals won by China in all years



Qualitative

- Medal *Type*:
  - *Gold/Silver/Bronze*
  - *Summarized with a table or chart*

Gold	Silver	Bronze
350	347	292

Types of Data

Olympic Medals

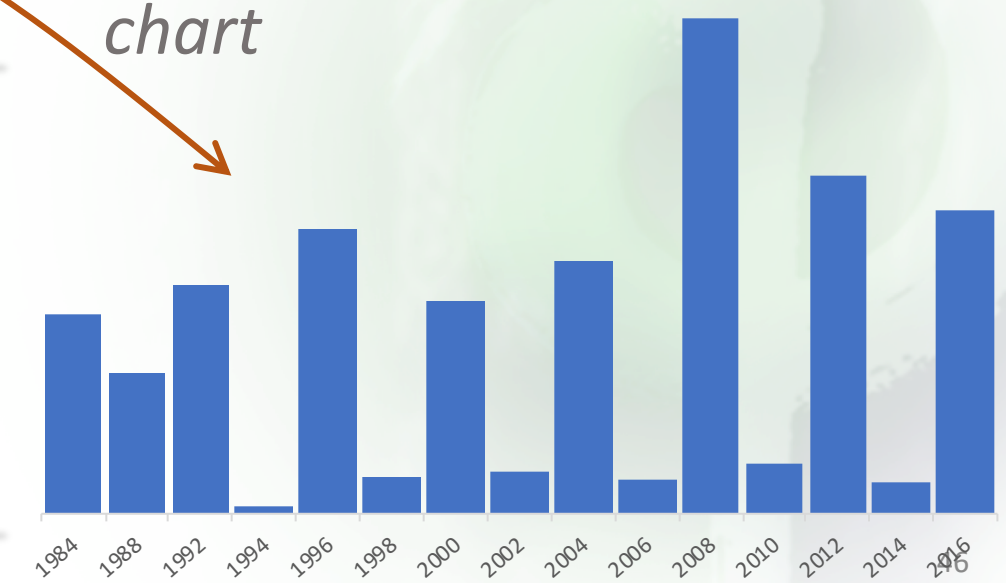
Quantitative

- *Number* of medals won by China in all years
- Can be shown with a *distribution*, or *summarized* with a sum, etc.

Year	#Medal
1984	74
1988	52
1992	85
1996	106
1998	14
2000	79

Qualitative

- Medal *Type*:
  - *Gold/Silver/Bronze*
  - *Summarized with a table or chart*

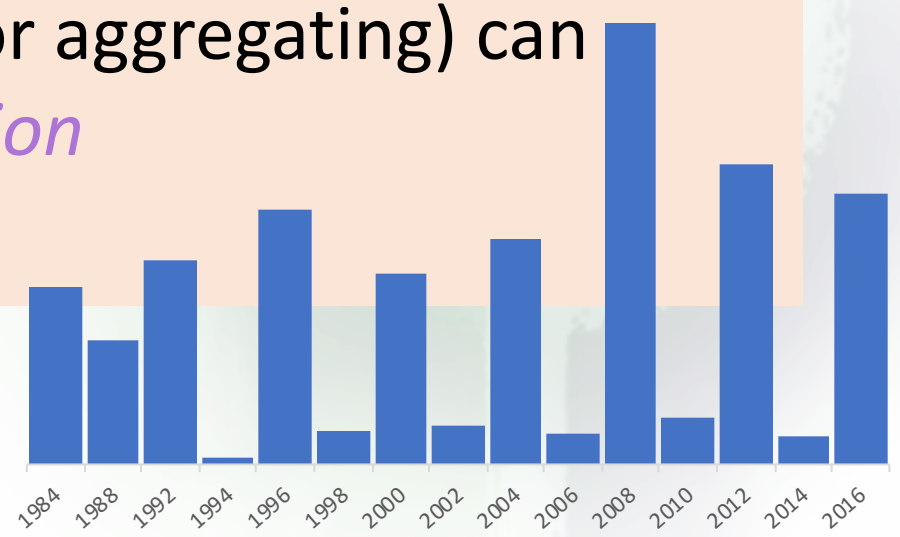


Types of Data

Olympic Medals

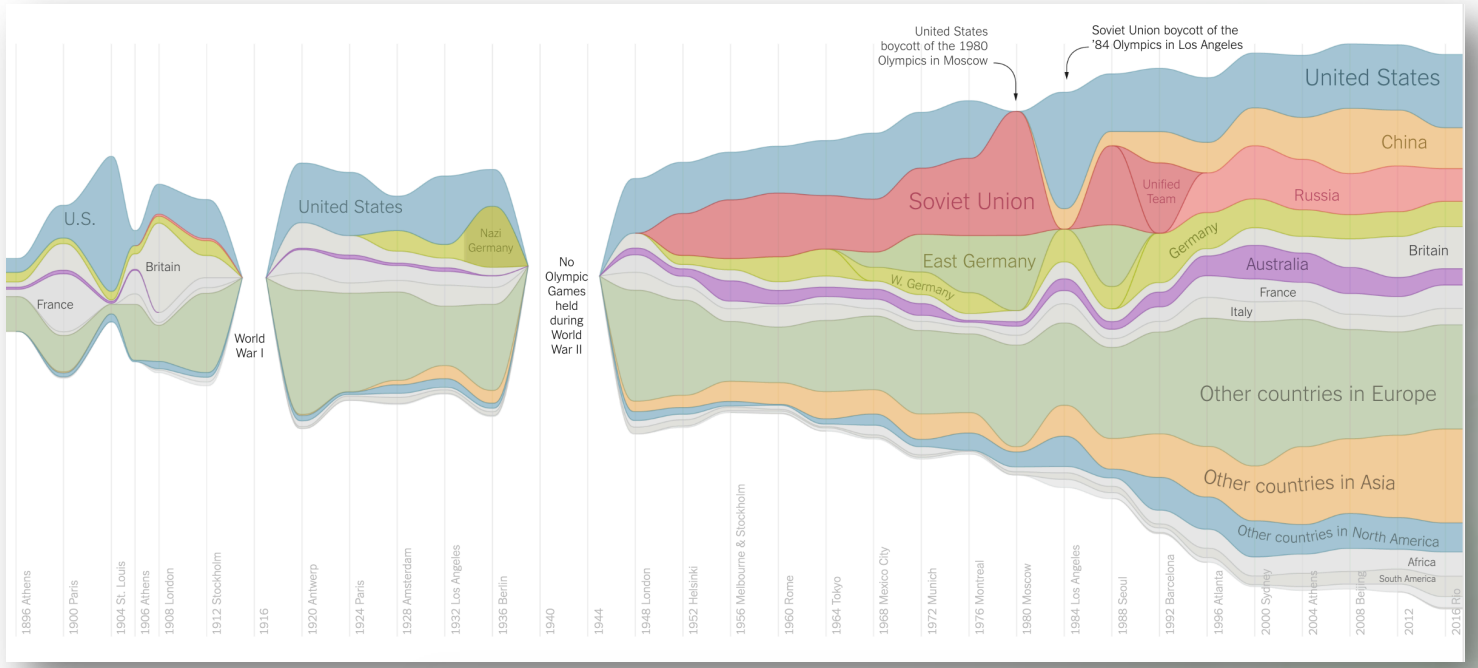
Quantitative

- *Commonly* used summaries:
  - *Average, Maximum or Minimum value, Standard deviation, etc.*
- Summarizing a distribution with a *single value* can be very useful
- But be *aware* that *averaging* (or pooling, or aggregating) can potentially *hide some interesting information*

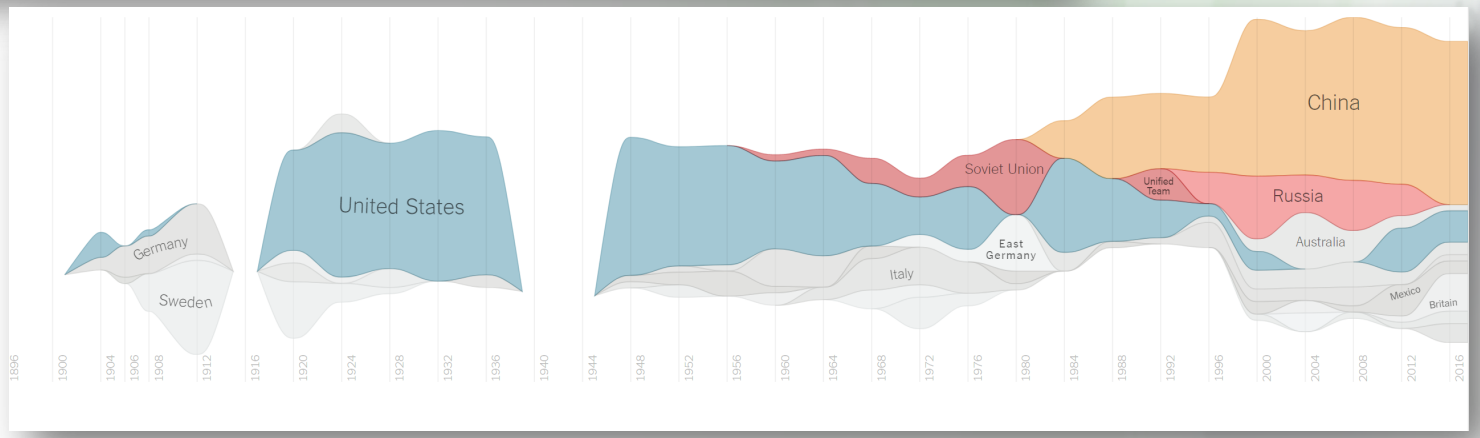


Pooling (or Aggregating) Data

All sports:



Diving:



Levels of Measurement

Levels of Measurement

Qualitative Data

Nominal level (by name)

- *No* natural *ranking* or *ordering* of the data exists
  - *e.g. Color (yellow, red, green)*

Ordinal level (by order)

- Provides an *order*, but *no precise mathematical difference* between levels
  - *e.g. heat (low, medium, high)*
  - *e.g. movie ratings (1-star, 2-star, etc.)*
    - *Watching two 2-star** movies isn't the same as watching one 4-star**** movie (the math not relevant here)*
  - Could be *coded numerically*, so again, be careful

Levels of Measurement

Qualitative Data

Political affiliation (*dem, rep, ind*)	Nominal
Level of pain (*low, med, high*)	Ordinal
Answer to survey: (*strongly disagree, disagree, agree, strongly agree*)	Ordinal
Eye color (*blue, green, brown, etc.*)	Nominal

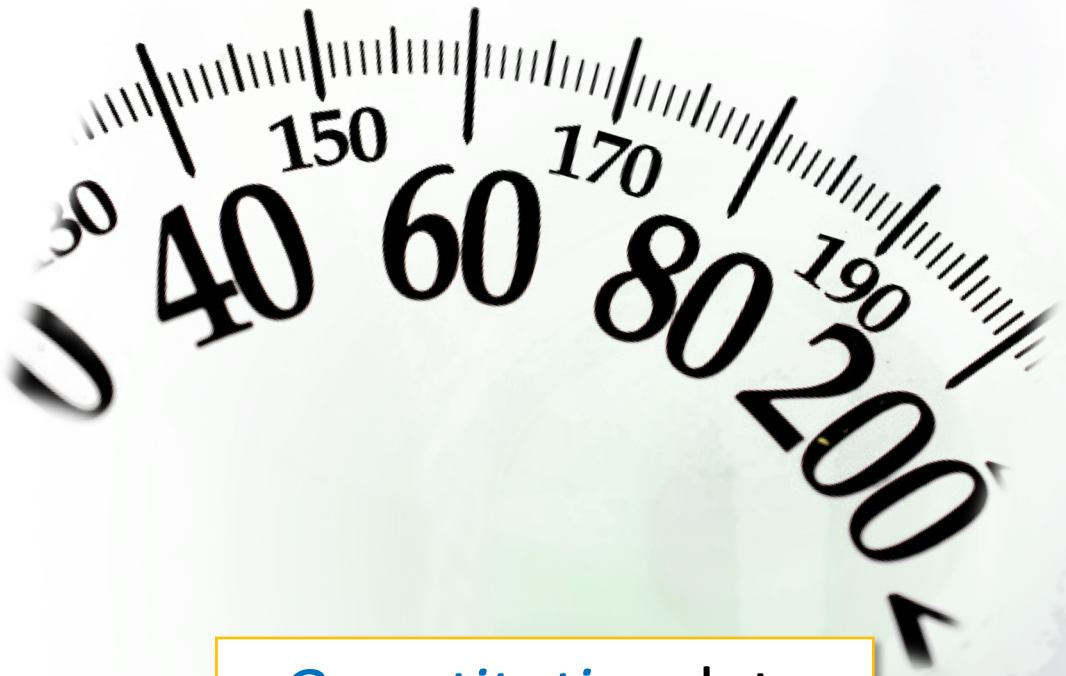
Two Kinds of Quantitative Data

Continuous

- Can take on any value in an interval
- Could have any number of decimals
  - *e.g. weight, home value, height*
  - *2.45, 7.63454, 4.0, π, etc.*

Discrete

- Can take on *only particular values*
  - *e.g. number of prerequisite courses (0, 1, 2, …)*
  - *e.g. number of students in a course*
  - *e.g. shoe sizes (7, 7-1/2, 8, 8-1/2,…)*



Quantitative data can either be *Interval* or *Ratio*

Levels of Measurement

Quantitative Data

Interval level (aka *differences* or *subtraction* level)

- Intervals of *equal length signify equal differences* in the characteristic
  - *The difference in 20° and 30° degree Celsius is the same as the difference between 25° and 35° degree Celsius*
- Differences make sense, but *ratios* do not
  - *20 degrees C is not twice as hot as 10 degrees C*
  - *E.g. 10C=50F and 20C=68F*
- Occurs when a numerical scale *does not* have a *true zero* start point
  - *Zero does not signify an absence of the characteristic*
  - *Does 0° degrees C represent an absence of heat?*

Levels of Measurement

Quantitative Data

Interval level (aka *differences* or *subtraction* level)

- IQ tests (interval scale)
  - *We do not have meaning for a 0 IQ*
  - *A 120 IQ is not twice as intelligent as a 60 IQ*
- Calendar years (interval scale)
  - *An interval of one calendar year (2005 to 2006, 2014 to 2015) always has the same meaning*
  - *But ratios of calendar years do not make sense because the choice of the year 0 is arbitrary and does not mean "the beginning of time"*
  - *Calendar years are therefore at the interval level of measurement*

Levels of Measurement

Quantitative Data

Ratio level (even *more meaning* than interval level)

- At this level, both *differences* and *ratios* are meaningful
  - *Two 2 oz glasses of water IS equal to one 4 oz glass of water*
  - *4 oz of water is twice as much as 2 oz of water*
- Occurs when scale does have a '*true zero*' start point
  - *0 oz of water is a 'true zero' as it is empty, absence of water*
- Ratios involve *division* (or *multiplication*) rather than *addition* or *subtraction*

Levels of Measurement

Quantitative Data

Example: ***Interval level***

- **Temperature used to cook food**
  - A brownie recipe calls for the brownies to be cooked at *400 degrees for 30 minutes*
  - Would the results be the same if you cooked them at *200 degrees for 60 minutes*? How about at *800 degrees for 15 minutes*?
  - *200 degrees is not half as hot as 400 degrees*. The *ratio of temperatures* does *not make sense* here

Levels of Measurement

Quantitative Data

Example: ***Ratio level***

- **Centimeters**
  - Difference of 40 cm (an interval) *makes sense* and has the *same meaning* anywhere along the scale
  - *10 cm is twice as long as 5 cm* (put two 5 cm items together and they are equivalent to 10 cm). Ratios make sense
  - *0 cm* truly represents *no length* or *absence of length*
- **Mass, Length, Time**

Possible Data Types & Level of Measure

Provides	Nominal	Ordinal	Interval	Ratio
Order of values		✓	✓	✓
Frequency of Distribution (Counts)	✓	✓	✓	✓
Mode	✓	✓	✓	✓
Median		✓	✓	✓
Mean			✓	✓
Can quantify the difference between value			✓	✓
Can add or subtract			✓	✓
Can multiple and divide				✓
Has “true zero”				✓

Central tendency can be measured by mode, median, or mean; Dispersion can be measured by standard deviation and coefficient of variation

Possible Data Types & Level of Measure

Nominal:
Attributes are only *named* (weakest)

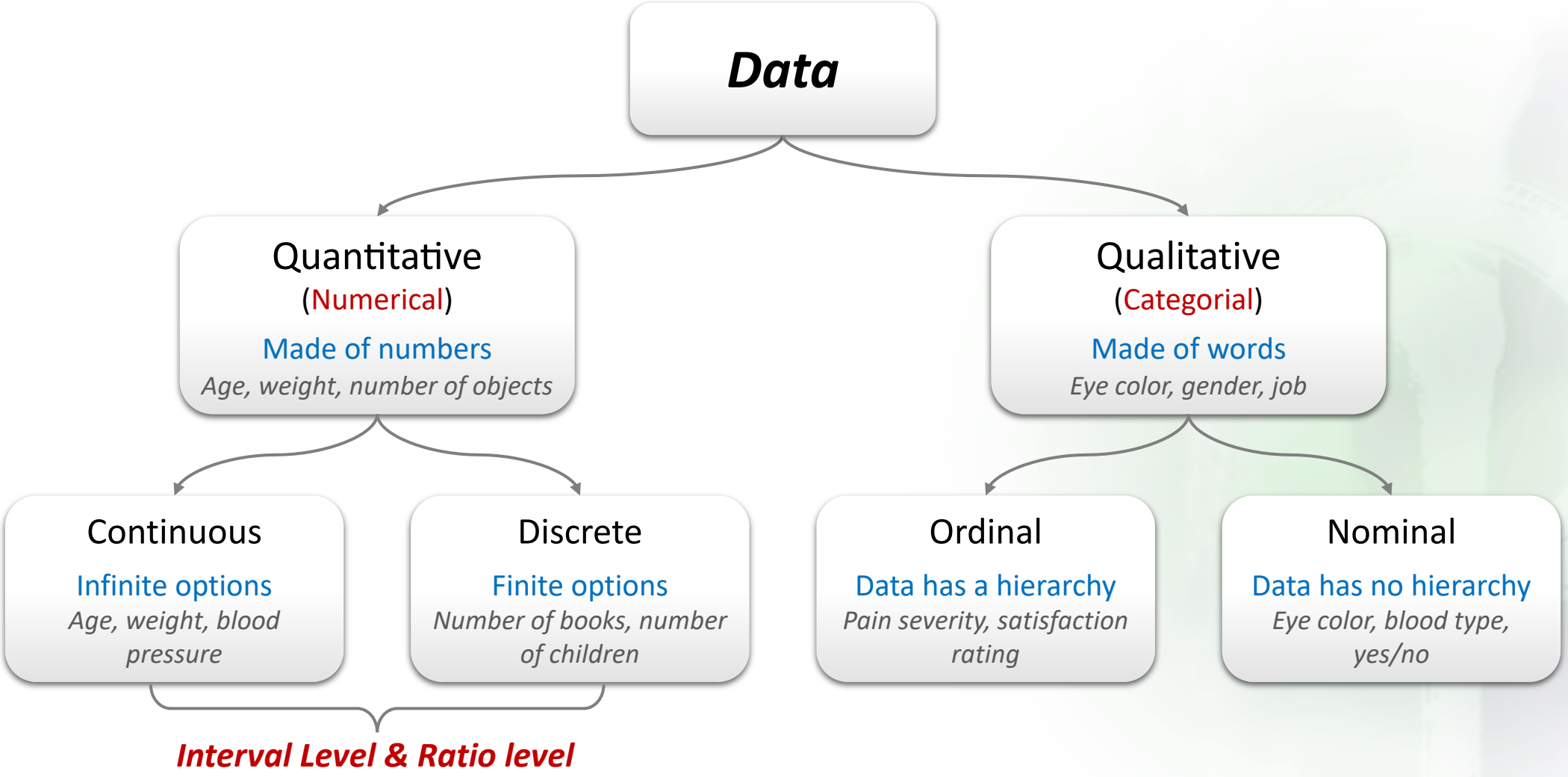
Ordinal:
Attributes can be *ordered*

Interval:
Distance is meaningful

Raito:
Absolute *zero*

Data Types

Data Analytics



Data Types

Programming Language & DBMS

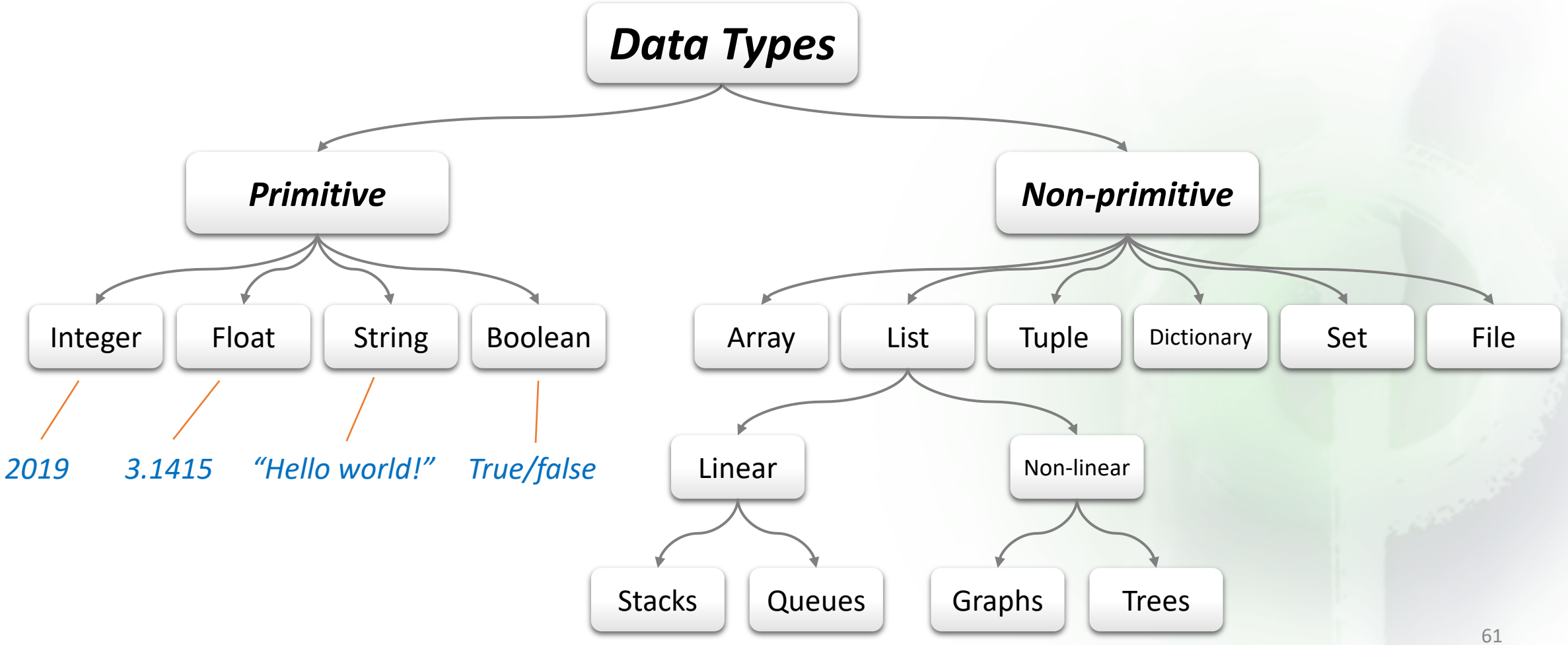
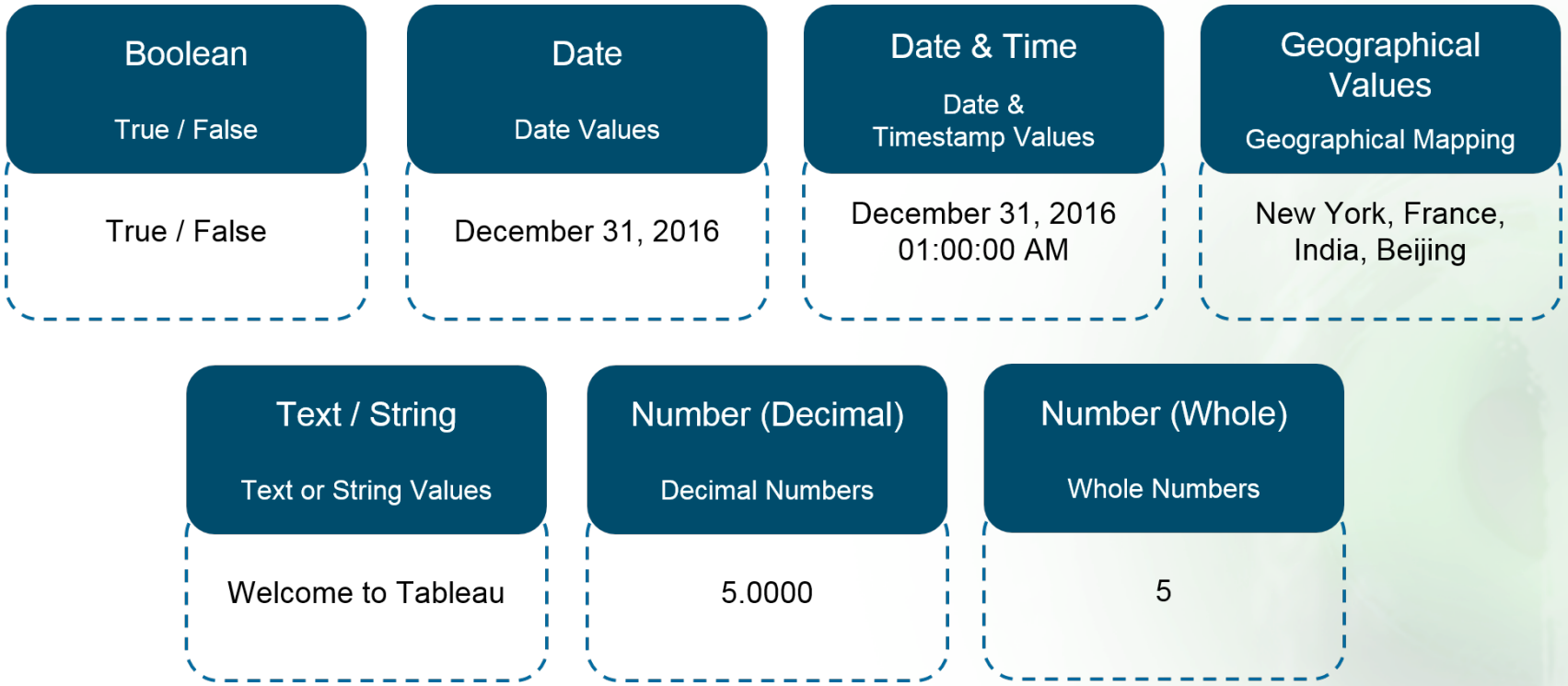


Tableau Desktop

Data Type



References

- Types of Data & Measurement Scales: Nominal, Ordinal, Interval and Ratio

Acknowledgements

Some of the materials are adapted from:

- Rhonda DeCook, 2019
- Zico Kolter, 2016