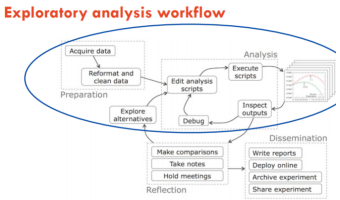
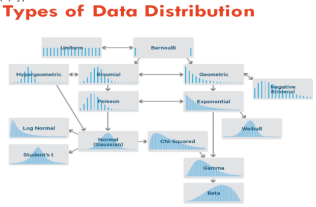
Week 2:
-structured Data[structured rows and columns to store], such as CSV or RDBMS
-Semi-structured Data[irregular, self-describing data, more flexible], such as JSON or XML(sql sever)
-Unstructured Data[text heavy], ie. text, e-mails, images, video
Nominal: 分类 应用于分类的数据，如性别，皮色（Dischotomous Data 二分类数据）；countable, mode, counts/distribution.
Ordinal: 定序变量 代表数据按某种特性的排序; countable, order defined, mode, medium, counts/distribution, minimum, maximum, range, percentiles.
Interval: 定距变量 变量之间比较有意义 0 意为 0; countable, order defined, difference defined, mode, medium, mean, counts/distribution, minimum, maximum, range, percentiles, standard deviation, variance.
Ratio: 定比变量 有绝对零点，取 0 时意为没有; countable, order defined, difference defined, zero-defined, mode, medium, mean, counts/distribution, minimum, maximum, range, percentiles, standard deviation, variance.
--Text(Not defined as traditional data type in statistics, Requires interpretation, coding or conversion)
--Images, Video
(2)Calculating descriptive statistics:(mean, median, variance $\frac{\sum(X_i - mean)^2}{N-1}$, stddev)
Variance&SD: how spread out a distribution is, measures of variability
(3)Exploratory analysis workflow

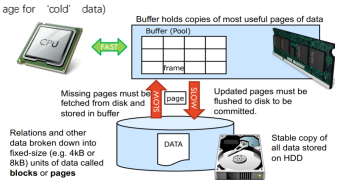


Python:
Interpreted: direct execution without compilation
-Dynamically-typed: don't have to declare a static type
-Readable: easy-to-understand syntax
-Deployable: easy to incorporate in applications
-The Dropbox desktop client is written entirely in Python (>40 million users)
-Productivity: facilitates rapid, interactive prototyping
(4)Various visualisation:(bar, chart, boxplots)
(5)Types of distribution and correlation



Typical Exam Questions:
1.What is normal distribution?
represents the distribution of many random variables as a symmetrical bell-shaped graph.
the proportion of the area that falls under the curve between two points on a probability distribution plot indicates the probability that a value will fall within that interval.
2.How would you do correlation analysis?
(Correlation analysis is a method of statistical evaluation used to study the strength of a relationship(association) between two, numerically measured, continuous variables)
The t-test is used to establish if the correlation coefficient is significantly different from zero, and, hence that there is evidence of an association between the two variables.
There is then the underlying assumption that the data is from a normal distribution sampled randomly.
P-value indicates how meaning the strength of association is.(assignment significance level is 0.01, p-values are all less than 0.01.
3.Expression understanding of different types of data. Which descriptive statistics or measures are supported on what types of data?
-Measures of central tendency describe the center position of a distribution for a data set using the mean, median, or mode, which measures the most common patterns of the analyzed data set.
-Measures of variability aid in analyzing how spread-out the distribution is for a set of data
Range, quartiles, absolute deviation, and variance are all examples of measures of variability.
4.Data cleansing(some example operations? Why is it needed?)
Typical steps involved(type and name conversion: filtering of missing or inconsistent data: unifying semantic data representations: matching of entries from different sources)

Week 3&4 (Accessing Data in Relational Databases)
(1)Database concepts
The database represents the entities (real-world things),the attributes (their relevant properties), and the logical relationships between the entities.
DBMS(database management system)
RDBMS (relational database management system)
--Database advantages:
Improved Data Sharing(Different users get different views of the data, Efficient concurrent access)
Enforcement of Standards(All data access is done in the same way)
Improved Data Quality(Integrity constraints, data validation rules)
Better Data Accessibility/ Responsiveness(Use of standard data query language (SQL))
Security, Backup/Recovery, Concurrecy(Disaster recovery is easier)
--Relational data model:
-Tables-an arrangement of related information stored in columns and rows.
-Field/Attribute-column in a table, contains homogenous set of data.
-record-a row in table
-data types-kind of data that can be stored in a field.
-primary key-a field in a table whose value is uniquely identifies each record in the table. A PK cannot be null and only ever be issued once
-foreign keys- When we need to refer to a record in a separate table we reference its ID as a foreign key.
-relational keys(single attribute or composite multiple attributes; primary key and foreign keys are included)
-relational database systems:
A relational database is a collection of such tables(rows with multiple attributes, rows of the same format form a 'table', Every relation has a schema, which describes the columns)
--Databases vs file system
(2)Transforming, Cleaning & Loading Data
type and name conversion
Filtering or fixing of missing or inconsistent data
Specifically with PostgreSQL: psql vs manual INSERTs using Python's psycopg2
(3)Querying a database with SQL
1)Handling NULL values in SQL: (want to see null value as 0)
SELECT station, obsdate, level, COALESE(discharge, 0) AS discharge, COALESE(temperature, 0) AS temperature,
FROM <table>
2)Combining data from multiple tables: Joins
3)Summarising data with SQL: aggregate functions and grouping
4)More SQL, subqueries, where clauses, joins
Typical Exam Questions:
(1)Explain problems that you could come across when importing data to a relational database also propose their solutions.
-how to deal with missing values. After we reset missing value as "NaN". Those tuples with that could not be uploaded to the database.
(2)Describe a data cleaning pipeline.
– type and name conversion – filtering of missing or inconsistent data. – unifying semantic data representations. – matching of entries from different sources. – Rescaling and optional dimensionality reduction
(3)What is the role of NULL in databases?
It represents missing or inconsistent data
(4)Difference between a natural join and other join type? Caveats of natural join?
Natural join applied if and only if there is a common column variables with the same attribute name between tables required to join together.
warning: If there is no common attribute in two tables; the query produces a cross join, also called the cross product of both tables
Week 5 (Scalable Analytics –The Role of Indexes and Data Partitioning)
Primary storage: expensive,Fasetest, but volatile(cache,RAM)- for currently used data
Secondary: cheap,big,nonvo latile, slower(online storage) (hard disks, solid-state drives)- for the main database
Tertiary: slowest(off-line storage), cheapest.(magnetic tape, optical storage)- for keeping the old version data.
Logical schema versus physical data design
1.Access to Data on Secondary Storage FAST:
Block-wise transfer(transfer data in fixed-size chunks (blocks or pages) between storage layers)
2.Caching or Buffering(Keep 'hot' data in memory, use secondary storage for 'cold' data)



3.Optimised File Organisation(Heap Files vs. Sorted Files; Row Stores vs Column Stores)
4.Indexing
5.Partitioning
(2)Alternative File Organizations: heap, sorted files, indexes(see (3)), coumn stores
--Heap Files
1.Simplest file structure contains records in no particular order.(Access method is a linear scan)
2.Rows appended to end of file as they are inserted(Hence the file is unordered)
3.Deleted rows create gaps in file(File must be periodically compacted to recover space)
--Sorted File
1.Rows are sorted based on some attribute(s)
2.Successive rows are stored in same (or successive) pages
3.Access method could be a binary search
4.Problem: Maintaining sorted order
After the correct position for an insert has been determined, shifting of subsequent tuples necessary to make space (very expensive)(typically are not used per-se by DBMS, but rather in
form of index-organised (clustered) files)
--Index file
1.Ordered index: search keys are stored in sorted order
2.Hash index: search keys are distributed uniformly across buckets using a hash function.
--Column Store
1.Advantages
-Non-needed attributes do not have to be read
-Better compression possibility
-Better parallel processing(different columns can be stored at different disks, allowing for parallel IO)
-also possibility to use SIMD instructions rather than tuple-at-a-time
2.disadvantage
-updates; -full-row access; -small-table data
(Popular among analytical databases: Vectorwise, Vertica. Etc.)
(3)Indexing
[CREATE INDEX name ON relation-name (<attributelist>); Example: CREATE INDEX StationNameIdx ON Stations(name); Drop: DROP INDEX index-name]
Pro&Con:
1.as efficient for searches (especially on ranges) as an ordered file; As flexible as a heap file for inserts and updates?
2. Additional I/O to access index pages; Index must be updated when table is modified. Indexes can make queries go faster, updates slower. Require disk space, too.

--Clustered index(Especially good for "range searches" where search key is between two limits):
1.Index entries and rows are ordered in the same way
2.There can be at most one clustering index on a table.(eg. the student information list in alphabetical order of students' name)
3.CREATE TABLE statement generally creates an clustered index on primary key.(For secondary indexes: CLUSTER TABLE name ON Index)
--Unclustered index(be necessary for attributes other than the primary key):
1.Index entries and rows are not ordered in the same way.(it actually creates another table with the index attribute and a row address column in it. After searching the target in the second table by index, it goes back to the first table to grab inofrmation by the row address attribute)
2.There can be many secondary indices on a table.
3.Index created by CREATE INDEX is generally an unclustered, secondary index.
--Covering index
an index that contains all attributes required
to answer a given SQL query:
1.all attributes from the WHERE filter condition
2.if it is a grouping query, also all attributes from GROUP BY & HAVING
3.all attributes mentioned in the SELECT clause
4.Typically a multi-attribute index
--Choosing an index
1.Attributes in WHERE clause are candidates for index keys.
-Exact match condition: hash index(hash index can only contain one attribute, and can only be used in the "=" condition), B+tree index
-equality queries: clustering.
-Range query: tree index types, Clustering, B+tree index
-WHERE clause contains several conditions(or attributes) or For index-only strategies:
Multi-attribute(covering index. Etc.), B+tree index
-Bitmap Indexes for OLAP or R-Tree for spatial databases
(4)Distributed Data Management:
Advantanges:
Easier to manage than a large table
Better availability
Queries faster on smaller partitions
1.horizontal(水平分)
2.vertical data partitioning(垂直分)
Methods for the two:

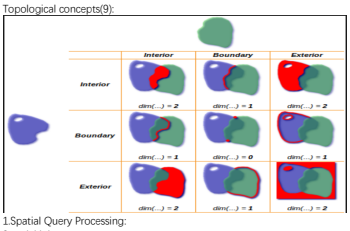
-round robin:
Placement of partitions is going through all nodes in rounds
-hash partitioning
Target node is determined by a hash function on the tuple id or key
-range partitioning
Each node stores a partitioned defined by a range predicate
Week 8 (Text Data Processing)
(1)Text data as unstructured data
E.g: Images, Video, Email, Social media
*(2)Spam detection as supervised classification
(may not in exam)About machine learning:
Supervised learning
Prediction, Classification (discrete labels), Regression (real values)
Unsupervised learning
Clustering, Probability distribution estimation, Finding association (in features), Dimension reduction
1.Split a string (document) into
pieces called tokens
2.Possibly remove some
characters, e.g., punctuation
3.Remove "stop words" such as "a", "the", "and" which are considered irrelevant
--normalisation
1.Map similar words to the same token
2.Stemming/lemmatization(Avoid grammatical and derivational sparseness – E.g., "was" => "be")
3.Lower casing, encoding(E.g., "Naïve" => "naive")
--Indicator Features
1.Binary indicator feature for each word in a document
2.Ignore frequencies
--Term Frequency Weighting(TF)
1.Term frequency(give more weight to terms
that are common in document)
TF = |occurrences of term in doc|
2.Damping(Sometimes want to reduce
impact of high counts)
TF = log(|occurrences of term in doc|)
--Inverse document frequency (IDF)
1.Give less weight to terms that are
common across documents(deals with the problems of the Zipf distribution(the, of, and, a , to···))
IDF = log(|total docs|/|docs containing term|)
2.TFIDF(TFIDF = TF * IDF)
Typical Exam Questions:
(1)How do you classify a piece of text?
(2)Explain TF and IDF?
*(3) Similarities between text and image processing?
In some of their methods, they all need to highlight interest parts by separate them from the whole.
Thresholding in image data process(do the separation of light and dark regions) is similar to the Term frequency weighting(give more weight to terms that are common in document) in text processing.

Week 9: (Geo-)Spatial Data
The difference between geometry and geography type of data is:
Geometry: 2-D spatial data, basis is cartesian plane 直角坐标系
Geography 球形 is made up of points on the earth's surface.
(1).Definition:
1.Spatial data is about objects and entities which have a location and/or a geometry
2.Geospatial data which refers to data or information that identifies the geographic location of features and boundaries on Earth (such as localities, cities, suburbs etc)
3.Example:
Non-spatial: List the names of all bookstore with more than ten thousand titles.
Spatial: List all customers who live in Victoria and its adjoining states
4.Spatial Database Management System (SDBMS)
-Handle large amount of spatial data stored in secondary storage.
-Spatial semantics built into query language
-Specialized index structure to access spatial data
5.Geographic Information System (GIS)
-SDBMS Client
-Characterized by a rich set of geographic analysis functions
-SDBMS allows GIS to scale to large databases, which are now becoming the norm
-Information in a GIS is typically organized in "layers" (example: "roads", "train stations", "suburbs", "water bodies" ···)
-GIS allows data exploration and integration across layers
(2)Types of Spatial Data:
1.Point Data – Points in a multidimensional space
Example:
-geo-location of some data entities from location-aware apps(As time-series: trajectory data of moving objects (such as cars))
-raster data such as satellite imagery, where each pixel stores a measured value
-Feature vectors extracted from text
2.Region Data
-Objects have spatial extent with location and boundary
-DB typically uses geometric approximations constructed using line segments, polygons, etc., called !!vector data!!
(2)Field-based versus Object-based data models
[Field based for modelling smoothly varying entities, like rainfall – Object based for modelling discrete entities, like country]
1.Field-based method:
-Spatial Framework is a partitioning of space (e.g., grid imposed by Latitude and Longitude)
-Field Functions: (f: Spatial Framework -> Attribute Domain)
-Field Operations (Examples, addition(+) and composition(o).)
-Types:
Local: value of the new field at a given location in the spatial frame-work depends only on the value of the input field at that location [e.g., Thresholding, Identify mountain ranges elevation over 2000 feet]
Focal: value of the resulting field at a given location depends on the values that the input field assumes in a small neighborhood of the location [e.g., Gradient), Identify peaks (points higher than its neighbors)]
Zonal: Zonal operations are naturally associated with aggregate operators or the integration function. An operation that calculates the average height of the trees for each species is a zonal operation.[eg: Determine average elevation of a set of river basins]

2.Objecte-based data method:
-Objects: distinct identifiable things relevant to an application
-Objects have attributes and operations
Attribute: a simple (e.g. numeric, string) property of an object
Operations: function maps object attributes to other objects
-Example:
A roadmap
Objects: roads, landmarks, ···
Attributes of road objects: spatial: location, e.g. polygon boundary of land-parcel; non-spatial: name (e.g. Route 66), type (e.g. motorway, residential street)
Operations on road objects: determine center line, determine length, determine intersection with other roads, ...
3.spatial objects(Geometry types):
Simple:
0 dimensional (points, city..), 1 dimensional (curves, river..), 2 dimensional (surfaces, country)
Collections:
Polygon collection (e.g. boundary of Japan or Hawaii), ···
-OpenGIS (OGC) Data Model
-Topological Relationships, 9-Intersection Model
4.SRID:
-The SRID is an integer that identifies the coordinate system in which the type is described
-Different SRIDs are incompatible.
-Different SRID means different results for some operations, such as area
5.Types of coordinates:
-Cartesian Coordinates(point positions from a defined origin along axes on a plane)
-Geodetic or Geocentric Coordinates (Geographic Coordinates) angular coordinates (longitude and latitude)
-!Projected Coordinates (planar Cartesian coordinates that result from performing a mathematical mapping from a point on the Earth surface to a plane)
6.Spatial operations:

Set theory based	Union, Intersection, Containment
Toplogical	Touches, Disjoint, Overlap, etc.
Directional	East,North-West, etc.
Metric	Distance

Topological concepts(9):



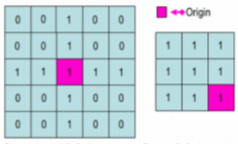
1.Spatial Query Processing:
Sptatial join:
District(ploygen)[多边形地区] –spatial join– Location(Point)[经纬度地区]
Filter-Refine Strategy
-Filter Step: Objects with minimum bounding box (MBR) intersecting query regions
-Refine Step: Query region really intersecting only with our targets

(4)Database support for (geo-)spatial data; PostGIS
Exam question:
(4)Also revisit what you did regarding spatial data in your assignment:
1. Import shapefile to create a SA2 schema with 'SA2_MAIN16', 'SA2_NAME16' and 'geom' columns.
2. Update 'geom' column by SQL ST_SetSRID(ST_MakePoint(longitude, latitude), 4326)
3. connected "BikeSharingPods.csv" or "BikeParking-Spaces.csv" to "Neighbourhoods.csv" through "SA2.csv" Connect "BikeSharingPods.csv" to "SA2.csv" by (ST_Contains(SA2.geom, BikeSharingPods.geom)
SA2 right outer JOIN Neighbourhoods ON (SA2.SA2_NAME16 = Neighbourhoods.area_name)
4. GROUP BY area_id to add the number of the same area and calculate z-score
Week 10: (Time Series Data)
(1)Temporal Database Concepts
1.Data types [SQL supports time instants and intervals]
-Instant data types(queries: DATE, TIMESTAMP, TIME)
-Interval data types(eg. Year-Month Intervals: INTERVAL YEAR TO MONTH [year and month fileds could be used])
-period
2.types of time:
-valid time: [records the time when a fact is true in the real world. – Can move forward and backward]
-transaction time: [records the history of database activity. – Only moves forward]
-user-defined time [Eg. a birthdate or a publication time]
3.Time support in relational databases
(2)Time Series Data
1.Point-based [in flat table]
-each tuple is timestamped with a time point/instant
-Most basic and simple data model
-Timestamps are atomic values and can be easily compared with =,/=, , ⩾, ⩽
-Syntactically different relations store different information
-If a fact is valid at several time points, multiple tuples are needed
-Good for time domain analysis
-Allows for indexing
2.sequence-based representation [in nested table]
-Single row with array of time point data
-Requires array datatype => outside 1NF idea of relational model
-To represent discrete time points now, we need to make sure that different arrays represent same data and time points
-Special functions needed to – Create arrays from set of data [array_agg()] – Pivot data back into sets [UNNEST]
Week 11 (Image Data Processing)
[Images can be described as vector graphics or raster data]
Raster images
1.Matrix-like with fixed number of rows and columns
2.Digital images consist of fixed number of picture elements, called pixels
3.Each pixel represents brightness of a given color
(1)Types of images:
1.RGB[Each pixel has a particular color; that color is described by the value of pixels in RGB channel (red, green and blue)]
2.binary[Each pixel is just black (0) or white (1); Single channel; Referred as 'mask' in the image processing domain]
3.grey scale(Single channel; Each pixel is a shade of gray.)

(2)typical image data analysis workflow
1.image analysis process:
Image Acquisition---Preprocessing---Feature Extraction---Classification / Image Similarity/Object Detection etc.
-Aspects of Image Processing
Image Enhancement: [Processing an image so that the result is more suitable for a particular application. Eg: sharpening or deblurring an out of focus image]
Image Restoration: [reversing the damage done to an image by a known cause. Eg: removing of blur caused by linear motion, removal of optical distortions]
Image Segmentation: [subdividing an image into constituent parts, or isolating certain aspects of an image. Eg: identifying cars,]
Method-Digital filters【preprocessing operations】:
used to suppress either the high frequencies in the image[smoothing the

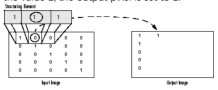
image]or the low frequencies,[enhancing or detecting edges in the image.]
(Median filter; Average filter; Weiner filter; Spatial filter)
Method-Dynamic Range[used to adjust contrast]

Image will have high contrast, if the dynamic range is high
2. Data Analysis Using Images
Removal of noise---
Extraction of region of interest (ROI) using binary mask(Image enhancement – Okay to extract ROI or shape features – Not-Okay for intensities-based analysis)---
Measurements[Binary image; Greyscale intensities]
(3)Image metadata
1.Typically includes[when captured, how, which camera/instrument, which settings; GPS location; author, copyright···]
2. Especially digital photography; standard by iptc.org; Several standard format, such as EXIF or XMP
(4)Morphological operations: dilation and erosion
1.Morphological Image Processing
-Morphological techniques probe an image with a small shape or template called a structuring element.
-The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one

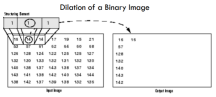


Cross-shaped 5x5 element Square 3x3 element

2. Morphological Dilation
The value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

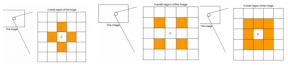


Dilation of a Binary Image

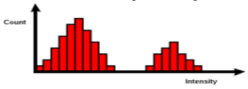


Dilation of a Grayscale Image

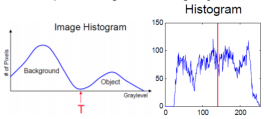
3. Morphological Erosion
The value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.
(5)Object connectivity


4-neighbourhood d-neighbourhood 8-neighbourhood
Example:
All the coloured pixels are "connected" to 'p'... or, they are 8-connected to p. However, only the green ones are '4- connected to p. And the orange ones are d-connected to p.

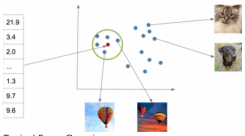

(6)ROI Histograms and thresholding (global, local, adaptive)
1.Histogram of the Pixel Intensity Values
-Histograms plots how many times (frequency) each intensity value in image occurs
-Different images can have same histogram[eg: Half of pixels are gray, half are white-same histogram = same statistics]
-We can' t reconstruct image from histogram




2.Thresholding[creates binary images from grey-level ones by turning all pixels below some threshold to zero and all pixels about that threshold to one; the separation of light and dark regions]



Issues:
Many objects at different gray levels
Variations in background gray level
Noise in image.
3. Global vs Adaptive/local thresholding
Local threshold T(x, y) is calculated for each pixel, based on some local statistics such as range, variance, or surface-fitting parameters of the neighborhood pixels within a local block of size wxw.
(7)Feature extraction from image data
1. Image feature:
[Simple pattern within image; Color information; Metadata]
-Core idea of using features: transform visual information into the vector space[gives possibility to perform mathematical operations on them, for example finding similar vectors]
-Image Similarity Search
[Extract feature vector from images and build index of them]

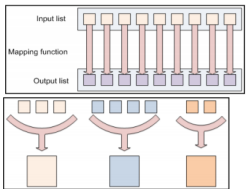


Typical Exam Questions:
(1)Given an image and its histogram choose appropriate thresholding to segment ROI
(2)Define and explain morphological operations, 4 & 8 object connectivity
(3)Give an example of an image data pre-processing operation.
Image Enhancement
Morphological dilation and erosion, Convolution and deconvolution, removing image noise
- Image Restoration
- Image Segmentation
Tresholding to identify region of interest
Week 12 (Big Data Processing)
Big data: [too large (volume),-too fast (velocity), or-needs to be combined from diverse sources (variety)]
Challenges:
[Data Privacy, Data Security, Data Discrimination]

(1) Distributed Data Processing for analyzing "Big Data"
1.Scale-up:[To scale with increasing load, buy more powerful, larger hardware]
from single workstation; to dedicated db server; to large massive-parallel database appliance
2.Scale-out: [Buy computing by the slice / brick] Rack of servers + disks; Grow by adding slices; Spread data and computation to new slices
3.Motivation: The three V' s; main goal: scalability
(2)Scale-agnostic processing: MapReduce paradigm
MapReduce[higher-order functions]:

```
SELECT out_key, reduce(out_value)
  FROM map(InputData)
GROUP BY out_key
```

1.Scan large volumes of data
2.Map: Extract some interesting information(by a given function f to all elements of a collection)
3.Shuffle and sort intermediate results
4.Reduce: aggregate intermediate results(All of the values with the same key are presented to a single reducer together)
5.Generate final output
Paradigm:
map()&reduce()



Pros:
-flexible(user-defined functions)
-great scalability(FP approach)
-easy parallelism(stateless functions)
-fault-tolerance
Cons:
-programming skills and functional thinking
-complex frameworks
-batch-processing oriented(can be carried out at any time)
(3)Dataflow-Oriented Data Analytics Platforms:
(need more high-level tools and interfaces than MapReduce.like:Apache Spark or Apache Flink)
1.Apache Hadoop:
-Open-source implementation of original MapReduce from Google[Apache top-level project]
-Java framework, but also provides a Python interface nowadays
-Parts: own distributed file system (HDFS), job scheduler (YARN), MR framework (Hadoop)2. Apache Spark
-Distributed cluster computing framework on top of HDFS/YARN
-Concentrates on main-memory caching and more high-level data flow control
-Originates from UC Berkeley AmpLab
3. Apache Flink
-Efficient data flow runtime on top of HDFS/YARN
-Similar to Spark, but more emphasize on build-in dataflow optimiser and pipelined processing
-Strong for data stream processing
-Origin: Stratosphere research project by TU Berlin, Humboldt University Berlin and HPI Potsdam

Spark and Flink

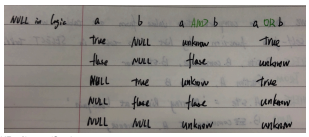
W6
General Approach: Reconnaissance 侦查 (identify source + check structure and content) + Webpage Retrieval 检索 (download one or multiple pages from source + autogenerates new URLs based on website structure and its URL format) + Data Extraction 获取 + Data Cleaning + Data Storage. **Tools helping HTML** as it' s not always well-formed – Unix command line tool (curl, grep, awk···) + 3rd party tools (google spreadsheets using ImportHTML(URL, "table/list", index of which list/table to import from webpage) function for single webpages + can be expensive) + programming libraries (Requests + BeautifulSoup library). Robots Exclusion Standard – robots.txt file suggests that web crawlers should check it before starting crawling. Extracting **Data from HTML and Intro to XML – web retrieval in python (single pages):** requests.get(URL, [params]) + requests.post(URL, params). **Web Crawling in Python (multiple pages):** scrapy (python framework to implement a web 'spider' that follows multiple links along the web) + selenium.**HTML** (hypertext markup language) – webpages are written in HTML + markup via open&closing tags in (<title>···</title>). **Web page structure**: Head (title, style sheets, metadata) + Body (headings, text, tables, images). **Select content in a webpage/query or filter XML:** text patterns (simple but no good for complex patterns) + **DOM navigation** (document object model) + CSS selectors (based on the tag types, class specifications and IDs elements + easy to specify) + **XPath expressions** (can query single values, nodes or whole subtrees within one XML document). **HTML vs XML:** XML has use-defined tags + mainly used for data exchange, while HTML as pre-defined tags according to the WWW standard + used for web page design. XML-Conformance: well-formed (document satisfies XML syntax constraints = matching tags) + valid (well-formed & satisfies DTD) Structure: both refer to its objects as elements + top element (root or document element) + tree structure + solely data type for leaf elements. **Storing extracted data** – file systems (CSV or XML files)+Database
W7
XML:

```
<person name="John Smith">
    <address street="1 Cleveland Street"
             city="Sydney"
             state="NSW"
             zipcode="2006" />
    <phoneNumbers>
      <phoneNumber type="work" number="9351 0000" />
      <phoneNumber type="fax"  number="9351 3838" />
    </phoneNumbers>
</person>
```

JSON:

```
{
  "name": "John Smith",
  "address": {
      "street":  "1 Cleveland Street",
      "city":    "Sydney",
      "state":   "NSW",
      "zipcode": 2006
  },
  "phoneNumbers": [
    { "type": "work",  "number": "9351 0000" },
    { "type": "fax",   "number": "9351 3838" }
  ]
}
```

Web service (API):
Getting data via web APIs – programmable APIs to expliciti request data
Semi-structured Data (JSON + XML) – characteristics: self-describing data with flexible structure + nested data model with tree structure + optional attributes, grammar, schema and vocabulary.
Document Type Definition (DTD): shows which tags are allowed in an XML document.
XML (stronger in database area due to SQL/XML, XPath···) vs JSON (unordered sets of name-value pairs with nesting, another semi-structured format + lower-overhead (read less) format).
Structured Data: schema-first + only atomic type attributes allowed many tables & joins needed.
Storing semi – Databases: some relational DBMS (PostgreSQL) +SQL/XML (provides XML data type to store XML natively as tree structure + integrates XML support function) + JSON in relational DB (JSON type stores exactly as JSON data & JSONB type stores a binary, decomposed versions) + NoSQL DB – non-relational DB.
Criticism for traditional DBMS –scalability in size and number of nodes + schema flexibility (sometimes unknown schema) + advanced data models (how to analyse large graphs in the scale of FB, Ebay···) + less need for traditional
"ACID" transaction processing.
MongoDB for JSON: document-oriented storage with flexible schema + collections of 'documents' which are nested key-value pairs.
SQL Part:
AND 优先级高于 OR
DESC: from biggest to smallest
The format of date type: YYYY-MM-DD
CURRENT_DATE refer to the current date data.
By using LIMIT to limit the number of column in the result.
random() can make the order random.
ROUND(attribute or 计算公式, number) 保留 number 位小数
||: By using || to combine several value. E.g. air_temp || "C" AS "temperature"
TO_CHAR(): By using this function to change the format e.g. TO_CHAR(obsdate, 'DD Mon YYYY' AS "date")
CASE WHEN<condition> THEN<expression> 将符合 condition 描述为 expression
Join part:
<table 1> JOIN <table 2> ON(condition)
<table 1> JOIN <table 2> USING(field)
RIGHT OUTER JOIN keep the column from the right table although there is no join partner
LEFT OUTER JOIN
FULL OUTER JOIN
EXTRACT(year FROM obsdate) 提取 date data 里面的年
HAVING() 在 GROUP BY 之后进行
UPPER(column) LOWER(column) 将 column 统一变成大写或小写
IN (range e.g. "" , "")



LIKE ··%... ; '$_ _'
INSERT INTO <table>(attributes) VALUES (values)
REFERNENCES the key refer to the foreigner key.
CREATE VIEW name AS <queries>