Principles of Database Systems (CS307)

Lecturer's Cut: Miscellaneous Sections

Yuxin Ma

Department of Computer Science and Engineering
Southern University of Science and Technology

- Most contents are from slides made by Stéphane Faroult and the authors of Database System Concepts (7th Edition).
- Their original slides have been modified to adapt to the schedule of CS307 at SUSTech.

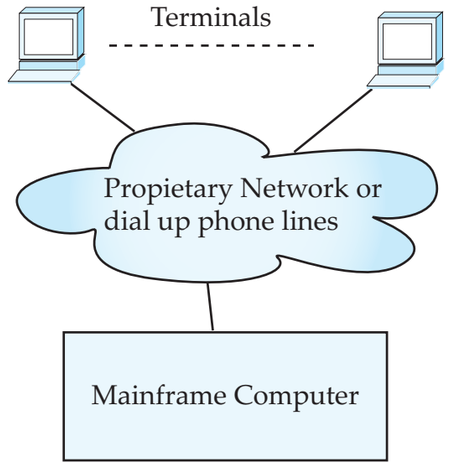
Application Development

Application Programs and User Interfaces

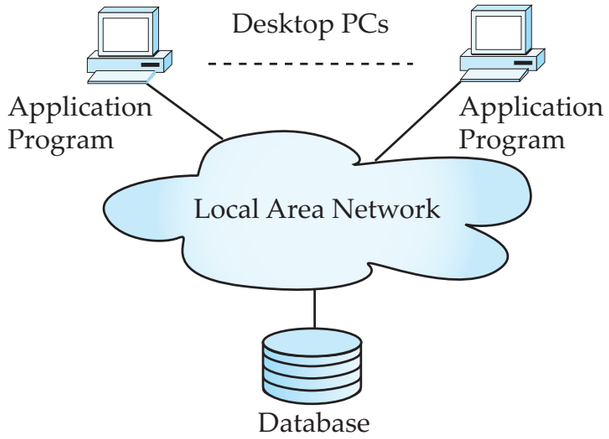
- Most database users do *not* use a query language like SQL
- An application program acts as the intermediary between users and the database
  - Applications split into
    - front-end
    - middle layer
    - backend
- Front-end: user interface
  - Forms
  - Graphical user interfaces
  - Many interfaces are Web-based

Application Architecture Evolution

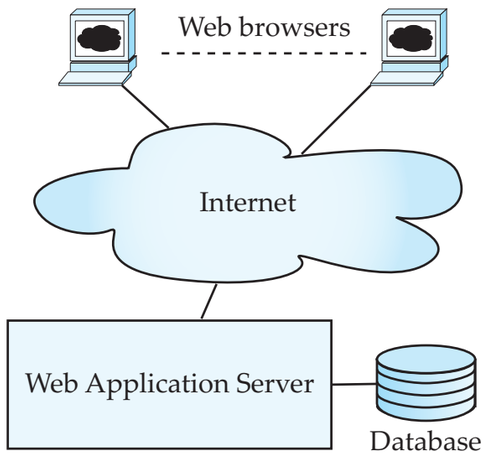
- Three distinct era's of application architecture
  - Mainframe (1960' s and 70' s)
  - Personal computer era (1980' s)
  - Web era (mid 1990' s onwards)
  - Web and Smartphone era (2010 onwards)



(a) Mainframe Era



(b) Personal Computer Era



(c) Web era

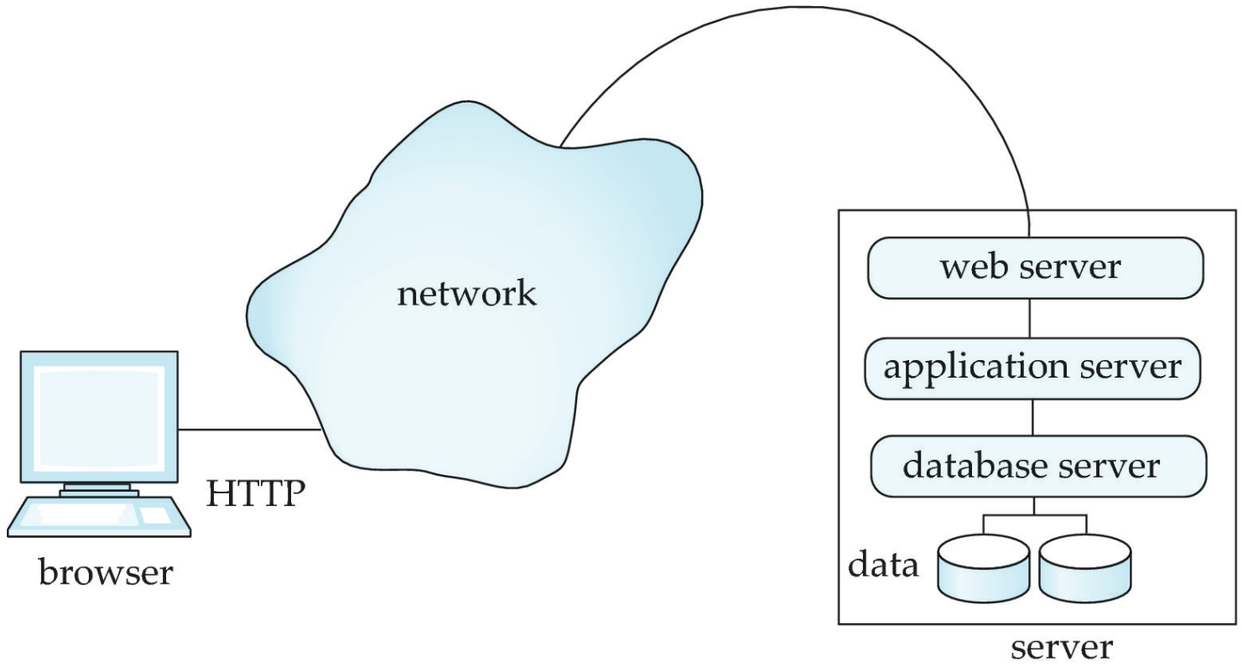
Web Interface

- Web browsers have become the de-facto standard user interface to databases
  - Enable large numbers of users to access databases from anywhere
  - Avoid the need for downloading/installing specialized code, while providing a good graphical user interface
    - JavaScript, ~~Flash~~ and other scripting languages run in browser, but are downloaded transparently
  - Examples: banks, airline and rental car reservations, university course registration and grading, and so on.

The World Wide Web

- The Web is a distributed information system based on hypertext.
- Most Web documents are hypertext documents formatted via the HyperText Markup Language (HTML)
- HTML documents contain
  - text along with font specifications, and other formatting instructions
  - hypertext links to other documents, which can be associated with regions of the text.
  - forms, enabling users to enter data which can then be sent back to the Web server

Three-Layer Web Architecture



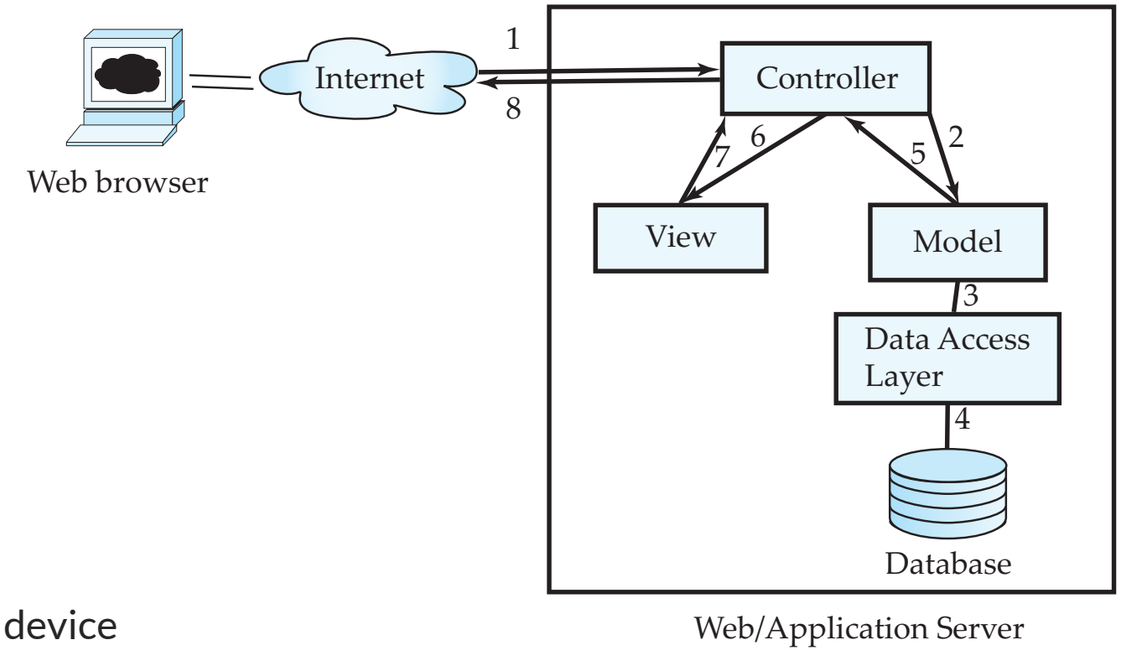
HTML and HTTP

- HTML provides formatting, hypertext link, and image display features
  - including tables, stylesheets (to alter default formatting), etc.
- HTML also provides input features
  - Select from a set of options
    - Pop-up menus, radio buttons, check lists
  - Enter values
    - Text boxes
  - Filled in input sent back to the server, to be acted upon by an executable at the server
- HyperText Transfer Protocol (HTTP) used for communication with the Web server

JavaScript

- JavaScript very widely used
  - Forms basis of new generation of Web applications (called Web 2.0 applications) offering rich user interfaces
- JavaScript functions can
  - Check input for validity
  - Modify the displayed Web page, by altering the underling document object model (DOM) tree representation of the displayed HTML text
    - Communicate with a Web server to fetch data and modify the current page using fetched data, without needing to reload/refresh the page
    - Forms basis of AJAX technology used widely in Web 2.0 applications
    - E.g. on selecting a country in a drop-down menu, the list of states in that country is automatically populated in a linked drop-down menu

Application Architectures



- Application layers
  - Presentation or user interface
    - model-view-controller (MVC) architecture
      - model: business logic
      - view: presentation of data, depends on display device
      - controller: receives events, executes actions, and returns a view to the user
  - business-logic layer
    - provides high level view of data and actions on data
      - often using an object data model
    - hides details of data storage schema
  - data access layer
    - interfaces between business logic layer and the underlying database
    - provides mapping from object model of business layer to relational model of database

Business Logic Layer

- Provides abstractions of entities
  - E.g., students, instructors, courses, etc
- Enforces business rules for carrying out actions
  - E.g., student can enroll in a class only if she has completed prerequsites, and has paid her tuition fees
- Supports workflows which define how a task involving multiple participants is to be carried out
  - E.g., how to process application by a student applying to a university
  - Sequence of steps to carry out task
  - Error handling
    - E.g. what to do if recommendation letters not received on time

Object-Relational Mapping

- Allows application code to be written on top of object-oriented data model, while storing data in a traditional relational database
  - Alternative: implement object-oriented or object-relational database to store object model
    - Has not been commercially successful
- Schema designer must provide a mapping between object data and relational schema
  - E.g., Java class Student mapped to relation student, with corresponding mapping of attributes
  - An object can map to multiple tuples in multiple relations
- Application opens a session, which connects to the database
- Objects can be created and saved to the database using session.save(object)
  - Mapping used to create appropriate tuples in the database
- Query can be run to retrieve objects satisfying specified predicates

Web Services

- Allow data on Web to be accessed using remote procedure call mechanism
- Two approaches are widely used
  - Representation State Transfer (REST): allows use of standard HTTP request to a URL to execute a request and return data
    - Returned data is encoded either in XML, or in JavaScript Object Notation (JSON)
  - Big Web Services:
    - Uses XML representation for sending request data, as well as for returning results
    - Standard protocol layer built on top of HTTP

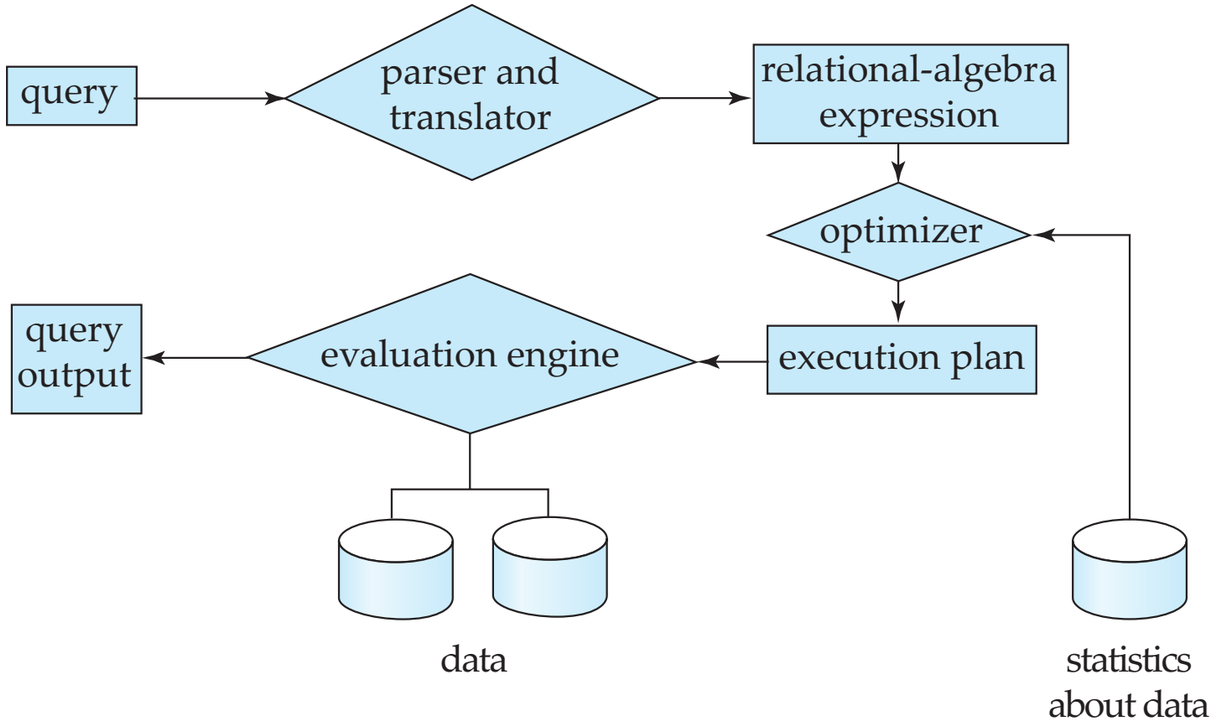
Self Study

- Key points:
  - Techniques / libraries / APIs to connect to a database (e.g. PostgreSQL) and run SQL querys in a program
    - ODBC, JDBC?
    - SQLAlchemy? Spring Boot? Hibernate?
  - (Web) Backend Frameworks
    - Spring Boot, Flask, Django
    - NodeJS
  - Frontend Frameworks
    - React, Vue

Query Planning and Optimization

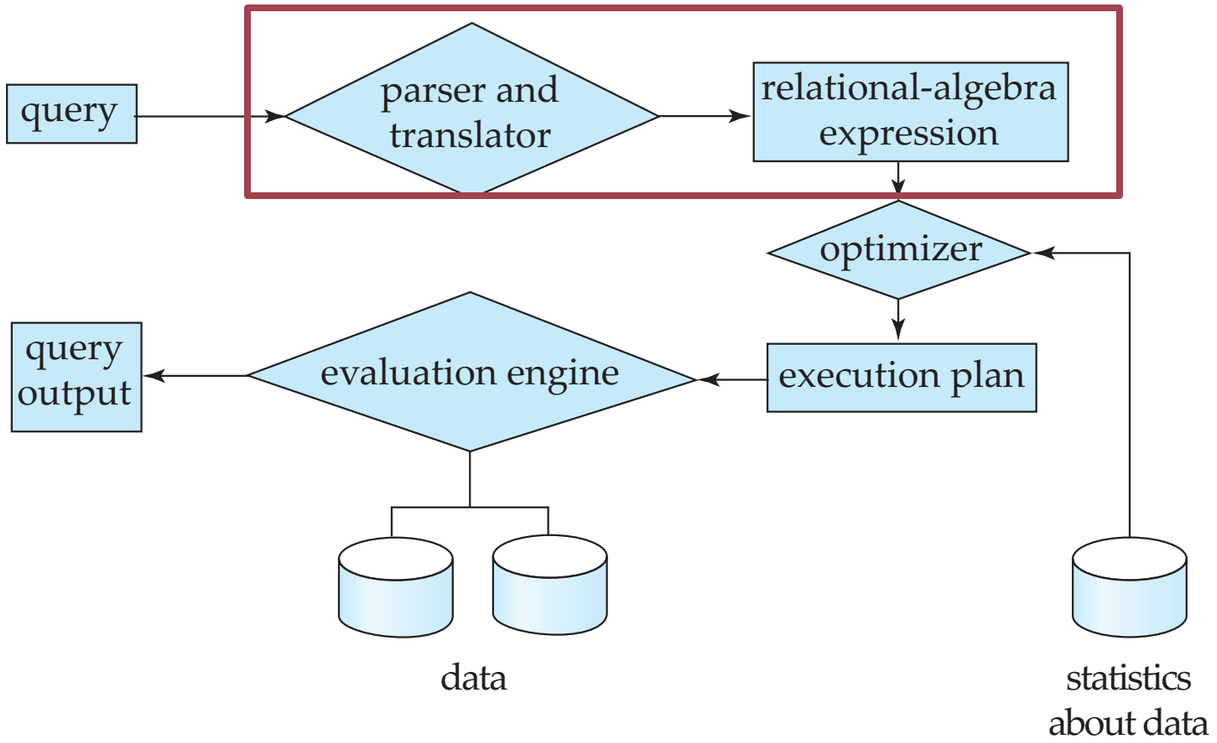
Basic Steps in Query Processing

- Parsing and Translation
- Optimization
- Evaluation



Basic Steps in Query Processing

- Parsing and Translation
  - Translate the query into its internal form
    - The internal form is then translated into relational algebra
  - Parser checks syntax and verifies relations



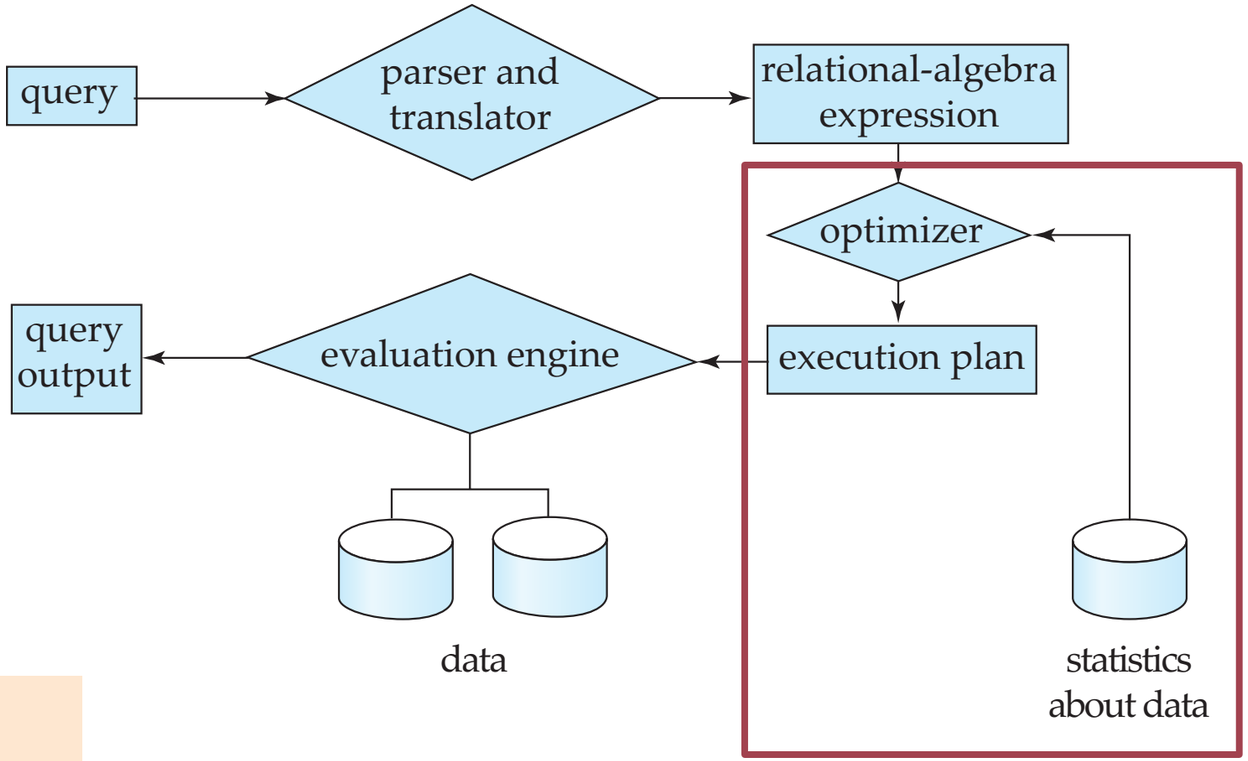
Basic Steps in Query Processing

- Optimization
  - A relational algebra expression may have many equivalent expressions
  - E.g.,

$$\sigma_{salary<75000}(\prod_{salary}(\text{instructor}))$$

is equivalent to

$$\prod_{\text{salary}}(\sigma_{\text{salary}<75000}(\text{instructor}))$$



But the number of rows involved in the projection operation may be (significantly) smaller in the second expression

Basic Steps in Query Processing

- Optimization
  - A relational algebra expression may have many equivalent expressions
    - … and each relational algebra operation can be evaluated using one of several different algorithms
  - *Correspondingly, a relational-algebra expression can be evaluated in many ways*

Basic Steps in Query Processing

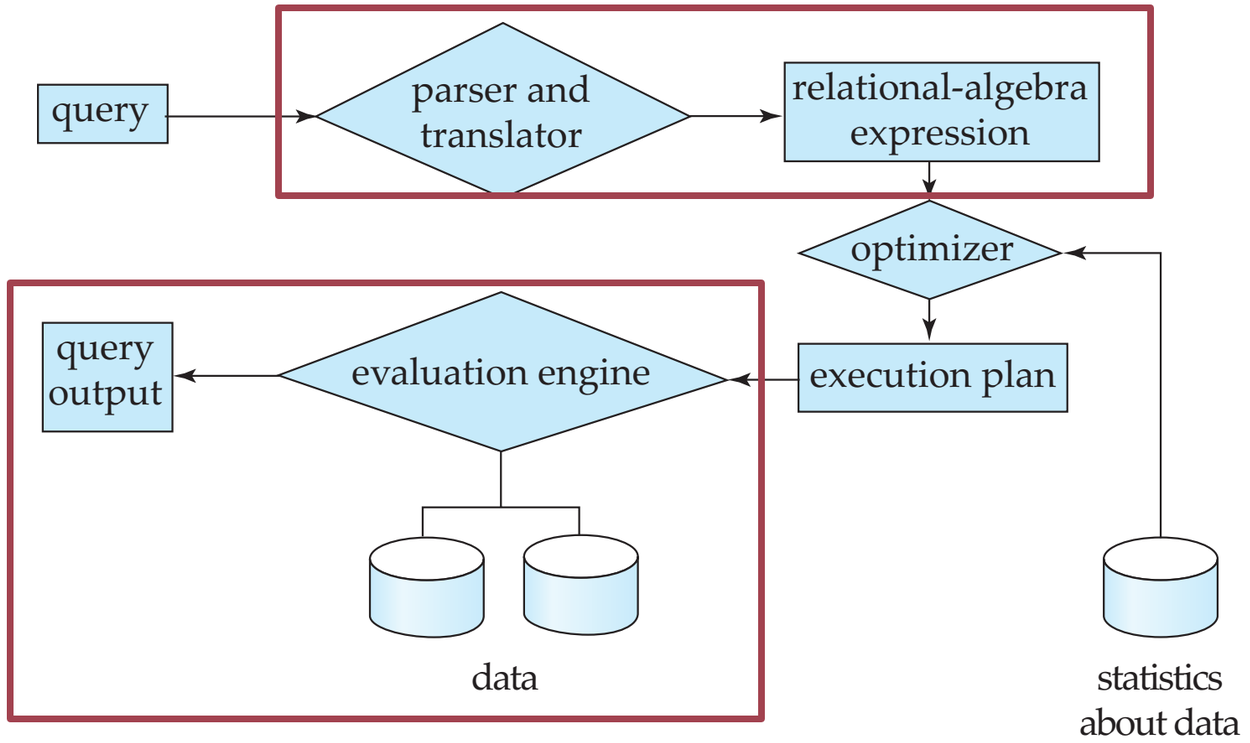
- Optimization
  - **Evaluation Plan**: Annotated expression specifying detailed evaluation strategy
  - E.g.,:
    - Use an index on `salary` to find instructors with `salary<75000`
    - Or perform complete relation scan and discard instructors with `salary<75000`

Query Optimization: Choose the one with the lowest cost among all equivalent evaluation plans

  - Cost can be estimated using statistical information from the database catalog
    - E.g., Number of tuples in each relation, size of tuples, etc.

Basic Steps in Query Processing

- Evaluation
  - The query-execution engine takes a query-evaluation plan, executes that plan, and returns the answers to the query



Further Reading

- Database System Concepts , 7th Edition
  - Part Six
    - Chapter 15 “Query Processing”
    - Chapter 16 “Query Optimization”

Beyond Tables: More Data Types

Semi-Structured Data

- Many applications require storage of complex data, whose schema changes often
- The relational model's requirement of atomic data types may be **an overkill**
  - E.g., storing set of interests as a set-valued attribute of a user profile may be simpler than normalizing it
- Data exchange can benefit greatly from semi-structured data
  - Exchange can be between applications, or between back-end and front-end of an application
  - Web-services are widely used today, with complex data fetched to the front-end and displayed using a mobile app or JavaScript
- **JSON** and **XML** are widely used semi-structured data models

Features of Semi-Structured Data Models

- Flexible schema
  - Wide column representation: allow each tuple to have a different set of attributes, can add new attributes at any time
  - Sparse column representation: schema has a fixed but large set of attributes, by each tuple may store only a subset

Features of Semi-Structured Data Models

- Multivalued data types
  - Sets, multisets
    - E.g.,: set of interests: `{"basketball", "cooking", "anime", "jazz"}`
  - Key-value map (or just map for short)
    - Store a set of key-value pairs
    - E.g.,
      - `{(brand, Apple), (ID, MacBook Air), (size, 13), (color, silver)}`
    - Operations on maps
      - `put(key, value)`
      - `get(key)`
      - `delete(key)`

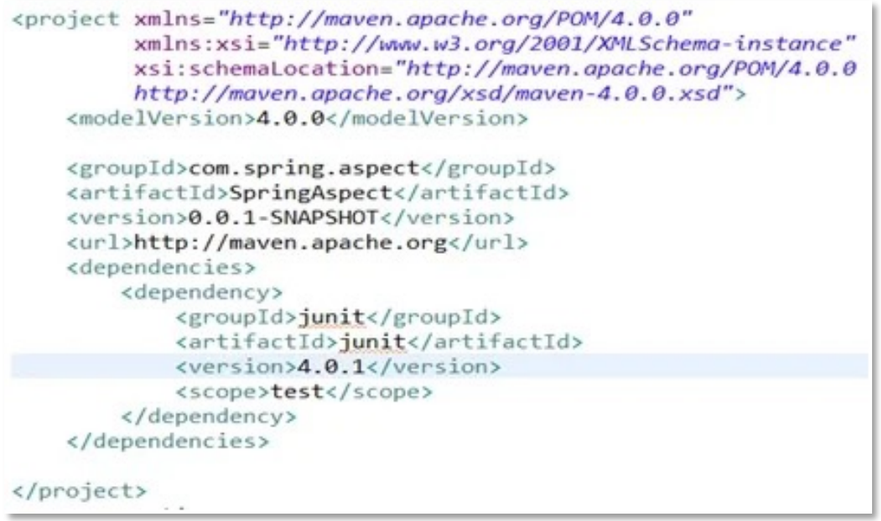
Features of Semi-Structured Data Models

- Arrays
  - Widely used for scientific and monitoring applications
  - E.g., readings taken at regular intervals can be represented as array of values instead of (time, value) pairs
    - [5, 8, 9, 11] instead of {(1,5), (2, 8), (3, 9), (4, 11)}
- Array database: a database that provides specialized support for arrays
  - E.g., compressed storage, query language extensions, etc.
  - Oracle GeoRaster, PostGIS, SciDB, etc

Nested Data Types

- Hierarchical data is common in many applications

- **JSON** (JavaScript Object Notation)
  - Widely used today
- **XML** (eXtensible Markup Language)
  - Earlier generation notation, still used extensively



JSON

- Textual representation widely used for data exchange

- Types: integer, real, string, and
  - Objects: key-value maps, i.e. sets of (attribute name, value) pairs
  - Arrays: also key-value maps (from offset to value)

```
{
    "ID": "22222",
    "name": {
        "firstname: "Albert",
        "lastname: "Einstein"
    },
    "deptname": "Physics",
    "children": [
        {"firstname": "Hans", "lastname": "Einstein" },
        {"firstname": "Eduard", "lastname": "Einstein" }
    ]
}
```

JSON

- JSON is ubiquitous in data exchange today
  - Widely used for web services
    - Most modern applications are architected around on web services
- PostgreSQL supports JSON format columns

```
create table json_test (
    id serial not null primary key,
    student json not null
);

insert into json_test (student) values ('{"name": "aaa", "age": 20, "major": {"primary": "cs", "minor":
"math"}}');
insert into json_test (student) values ('{"name": "bbb", "major": {"primary": "math", "minor": "physics"}}');
insert into json_test (student) values ('{"name": "ccc", "age": 19, "major": {"primary": "biology"}}');
```

JSON

- JSON is ubiquitous in data exchange today
  - Widely used for web services
    - Most modern applications are architected around on web services
- PostgreSQL supports JSON format columns

```
-- select all content from the column
select * from json_test;

-- select a value of a key with "->"
select student -> 'major' -> 'minor' from json_test;
```

	id	student
1	1	{"name": "aaa", "age": 20, "major": {"primary": "cs", "minor": "math"}}
2	2	{"name": "bbb", "major": {"primary": "math", "minor": "physics"}}
3	3	{"name": "ccc", "age": 19, "major": {"primary": "biology"}}

	?column?
1	"math"
2	"physics"
3	<null>

XML

- XML uses tags to mark up text
  - Tags make the data self-documenting
  - Tags can be hierarchical

```
<course>
    <course id>CS-101</course id>
    <title>Intro. to Computer Science</title>
    <dept name>Comp. Sci.</dept name>
    <credits>4</credits>
</course>
```

```
<info>
    <name>aaa</name>
    <age>20</age>
    <major>
        <primary>cs</primary>
        <minor>math</minor>
    </major>
</course>
```

Textual Data

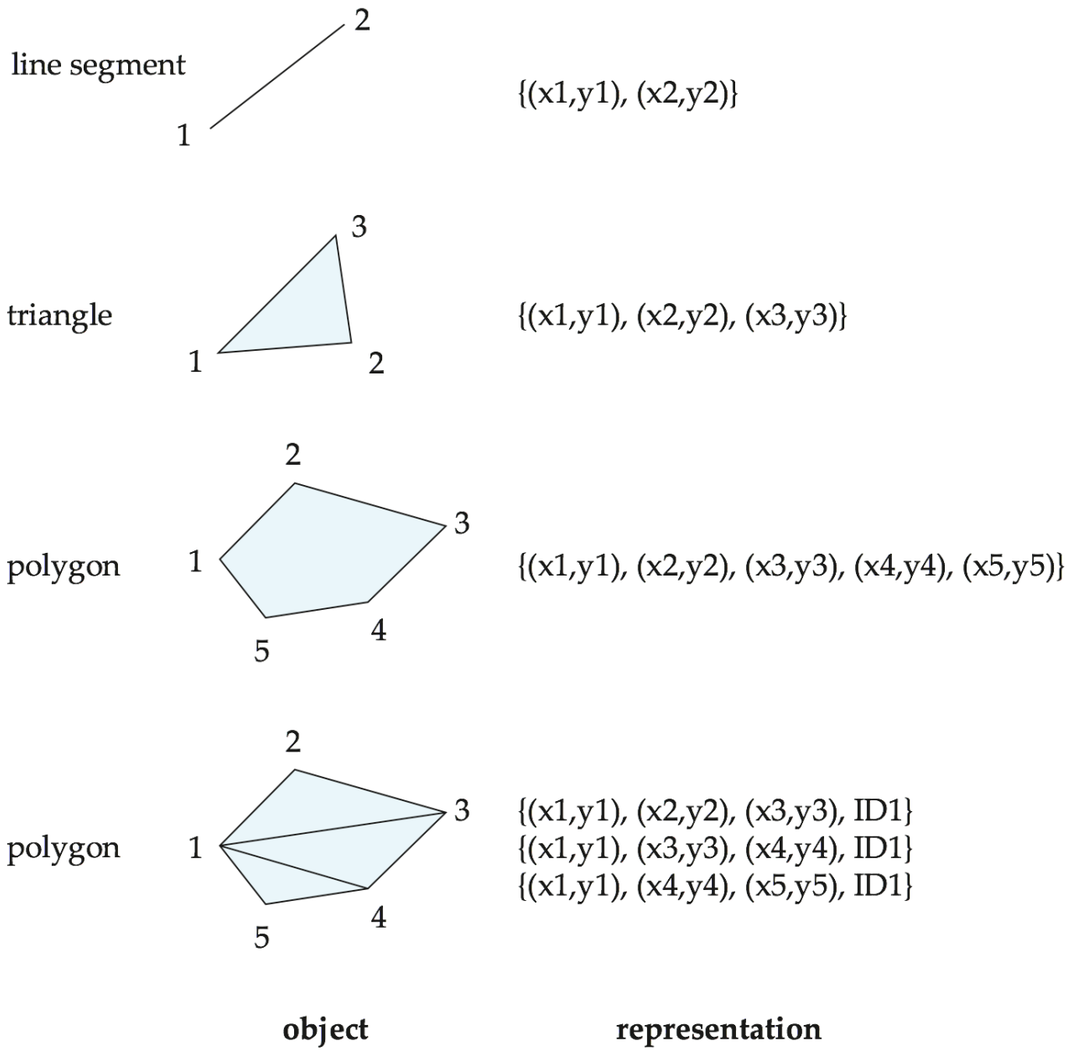
- Information retrieval: querying of unstructured data
  - Simple model of keyword-based queries
    - Given query keywords, retrieve documents containing all the keywords
  - More advanced models rank relevance of documents
  - Today, keyword queries return many types of information as answers
    - E.g., a query “cricket” typically returns information about ongoing cricket matches
- Relevance ranking
  - Essential since there are usually many documents matching keywords

Spatial Data

- **Spatial databases** store information related to spatial locations, and support efficient storage, indexing and querying of spatial data.
  - **Geographic data**: road maps, land-usage maps, topographic elevation maps, political maps showing boundaries, land-ownership maps, and so on.
    - Geographic information systems (GIS) are special-purpose databases tailored for storing geographic data.
    - Round-earth coordinate system may be used
      - (Latitude, longitude, elevation)
  - **Geometric data**: design information about how objects are constructed
    - E.g., designs of buildings, aircraft, layouts of integrated-circuits.
    - 2 or 3 dimensional Euclidean space with (X, Y, Z) coordinates

Representation of Geometric Information

- Various geometric constructs can be represented in a database in a normalized fashion



NoSQL Database

- “Not Only SQL”
  - Useful when working with a huge quantity of data when nature of data does not require a relational model
  - Usually not built on tables and queried by SQL
- Examples
  - Document store – MongoDB
  - Graph structure – Neo4j
  - Key-value storage – Redis, LevelDB
  - Tabular – Apache Hbase (Hadoop-based)

Beyond PostgreSQL: More DBMS

Commercial & Open-Source Solutions

Commercial Relational DBMS:

- Oracle Database
- Microsoft SQL Server
- IBM DB2
- …

Open-Source Counterparts:

- MySQL (MariaDB)
- PostgreSQL
- …

Commercial & Open-Source Solutions

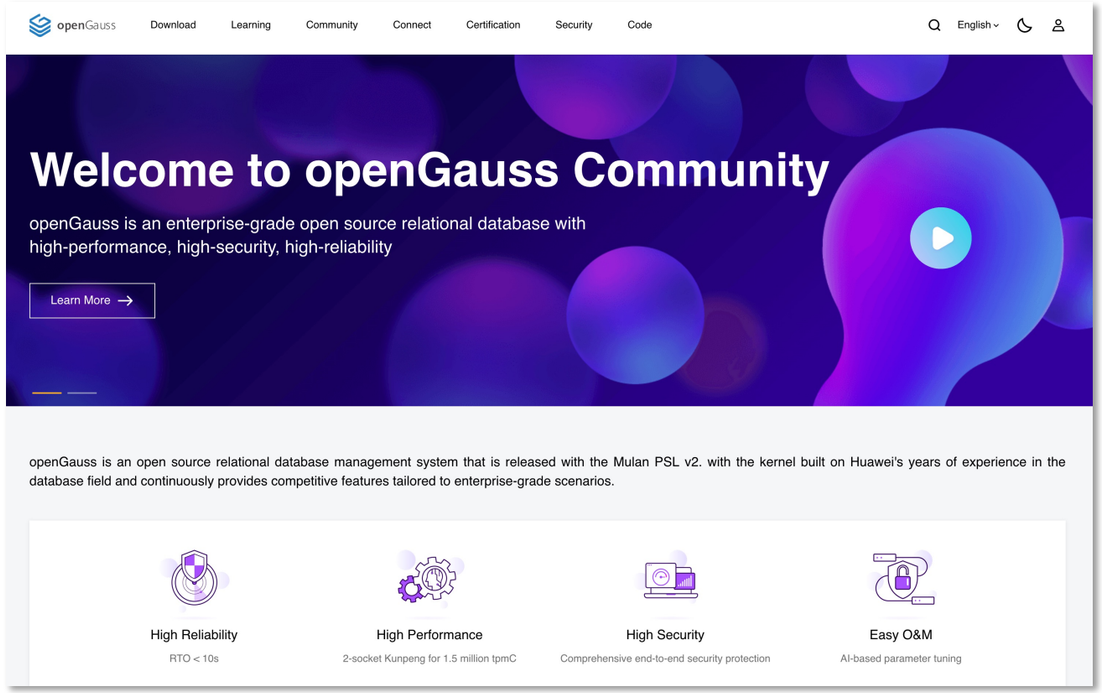
- Factors to consider open-source databases
  - Cost
    - Open-source databases are generally free
  - Customizability
    - Add your own features in the source code
  - Community support
    - Documentations, bug fixes, discussions

Commercial & Open-Source Solutions

- Factors to consider commercial databases
  - Technical support
    - Guaranteed professional services
  - Usability
    - Generally easier to deploy and use
    - Can be seamlessly integrated into other commercial products
  - Feature support
    - Enterprise-level feature extension (More functions, useful SQL syntax, etc.)

openGauss

- Relational DBMS from Huawei
  - Enterprise-grade open-source relational database
  - Client-server architecture
  - High-performance, high-reliability, high-security
  - Community support
  - * Compatible to PostgreSQL clients



https://www.opengauss.org/en/

Key Differences between openGauss and PostgreSQL

- originated from PostgreSQL-XC (eXtensible Cluster)
- Fundamental differences in the architecture and key technologies, especially in the storage engine and query optimizer

关键差异化因素		openGauss	PostgreSQL
运行时模型	执行模型	线程池模型，高并发连接切换代价小、内存损耗小，执行效率高，一万并发连接比最优性能损耗<5%。	进程模型，数据库进通过共享内存实现通讯和数据共享。每个进程对应一个并发连接，存在切换性能损耗，导致多核扩展性问题。
事务处理	并发控制	64位事务ID，使用CSN解决动态快照膨胀问题; NUMA-Aware引擎优化改造解决“五把大锁。	事务ID回卷，长期运行性能因为ID回收周期大幅波动；存在“五把大锁”的问题，导致事务执行效率和多处理器多核扩展性存在瓶颈。
	日志和检查点	增量Checkpoint机制，实现性能波动<5%。	全量checkpoint，性能短期波动>15%。
	鲲鹏NUMA	NUMA改造、cache-line padding、原生spin-lock。	NUMA多核能力弱，单机两路性能TPMC <60w。
数据组织	多引擎	行存、列存、内存引擎，在研DFV存储和原位更新。	仅支持行存。
SQL引擎	优化器	支持SQL Bypass, CBO吸收工行等企业场景优化能力。	支持CBO，复杂场景优化能力一般。
	SQL解析	ANSI/ISO标准SQL92、SQL99和SQL2003 和企业扩展包。	ANSI/ISO标准SQL92、SQL99和SQL2003。

Key Differences between openGauss and PostgreSQL

- originated from PostgreSQL-XC (eXtensible Cluster)
- Fundamental differences in the architecture and key technologies, especially in the storage engine and query optimizer

	openGauss	**PostgreSQL**
Execution Model	Thread pool-based (higher concurrency performance)	Process-based
Data Organization	Multiple engines: Row-oriented, column-oriented, in-memory storage	Only row-oriented
SQL Optimization	More complex enterprise-level optimization	Cost-based optimization
SQL Parsing	ANSI/ISO SQL92, SQL99, SQL2003 w/ enterprise-level extensions	ANSI/ISO SQL92, SQL99, SQL2003

Beyond Storage: Data Analytics

What is Data (Revisited)

data noun, plural in form but singular or plural in construction, often attributive

Save Word

da·ta | \ ˈdā-tə, ˈda- *also* ˈdä- \

Definition of *data*

1 : factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation

// the *data* is plentiful and easily available
— H. A. Gleason, Jr.

// comprehensive *data* on economic growth have been published
— N. H. Jacoby

2 : information in digital form that can be transmitted or processed

3 : information output by a sensing device or organ that includes both useful and irrelevant or redundant information and must be processed to be meaningful

factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation

https://www.merriam-webster.com/dictionary/data

Basic Statistical Descriptions

- Overall picture of your data
- Basis of exploratory data analysis

- Mean

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} = \frac{x_1 + x_2 + \cdots + x_N}{N}.$$

- Median

$$\mathrm{Q}_{\frac{1}{2}}(x) = \begin{cases} x'_{\frac{n+1}{2}}, & \text{if } n \text{ is odd.} \\ \frac{1}{2}\left(x'_{\frac{n}{2}} + x'_{\frac{n}{2}+1}\right), & \text{if } n \text{ is even.} \end{cases}$$

- Variance

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 = \left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \bar{x}^2,$$

Relationship between Data Objects: Data (Dis)Similarity

- Measurement of relationships
  - Commonly used in many statistical methods and data mining algorithms

- Dissimilarity Matrix & Distance Measures

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{bmatrix}$$

Euclidean	$d(x,y) = \sqrt{\sum (x_i - y_i)^2}$
Squared Euclidean	$d(x,y) = \sum (x_i - y_i)^2$
Manhattan	$d(x,y) = \sum \lvert (x_i - y_i) \rvert$
Canberra	$d(x,y) = \sum \frac{\lvert x_i - y_i \rvert}{\lvert x_i + y_i \rvert}$
Chebychev	$d(x,y) = \max(\lvert x_i - y_i \rvert)$
Bray Curtis	$d(x,y) = \frac{\sum \lvert x_i - y_i \rvert}{\sum x_i + y_i}$
Cosine Correlation	$d(x,y) = \frac{\sum (x_i y_i)}{\sqrt{\sum (x_i)^2 \sum (y_i)^2}}$
Pearson Correlation	$d(x,y) = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Uncentered Peason Correlation	$d(x,y) = \frac{\sum x_i y_i}{\sqrt{\sum (y_i - \overline{y})^2} \sqrt{\sum (y_i - \overline{y})^2}}$
Euclidean Nullweighted	Same as Euclidean, but only the indexes where both x and y have a value (not NULL) are used, and the result is weighted by the number of values calculated. Nulls must be replaced by the missing value calculator (in dataloader).

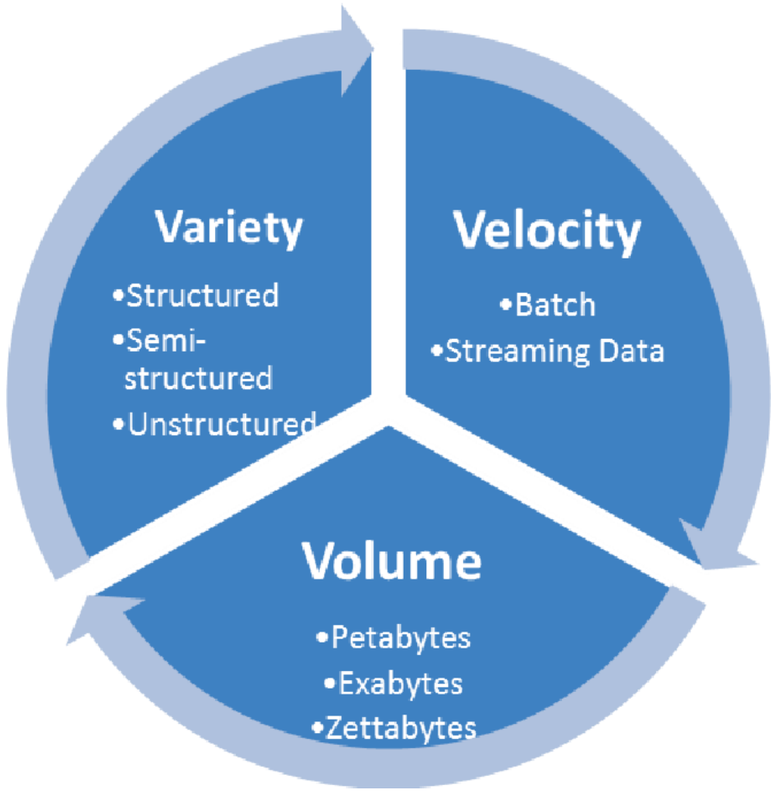
What is Big Data?

- A collection of data sets so large and complex

Three Dimensions of Big Data

- **Volume**
  - From GB to TB, PB, or higher
- **Velocity**
  - Processing speed
- **Variety**
  - Text, sensor data, multimedia, …

- Other (new) aspects:
  - Veracity: Trustworthiness
  - Value: Worth of data



The Emergence of Data Science

- 2016: “Trump vs. Clinton: How Big Data and scientists helped Trump win the election”

Digital campaigning

The role of technology in the presidential election

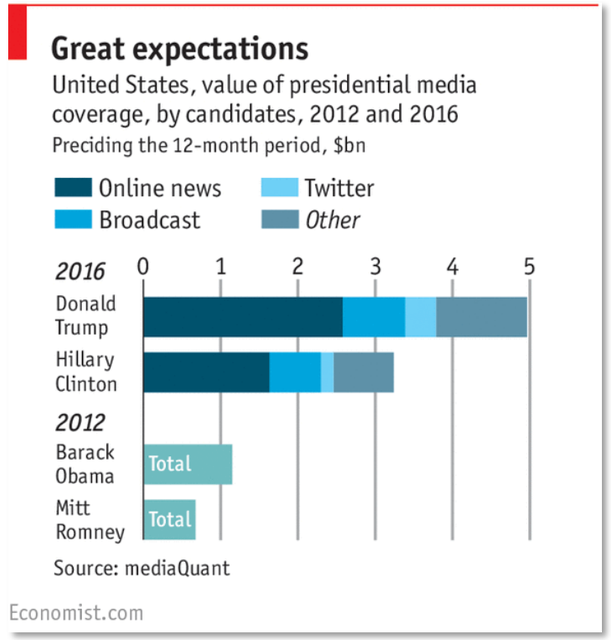
All latest updates

From fake news to big data, a post mortem is under way

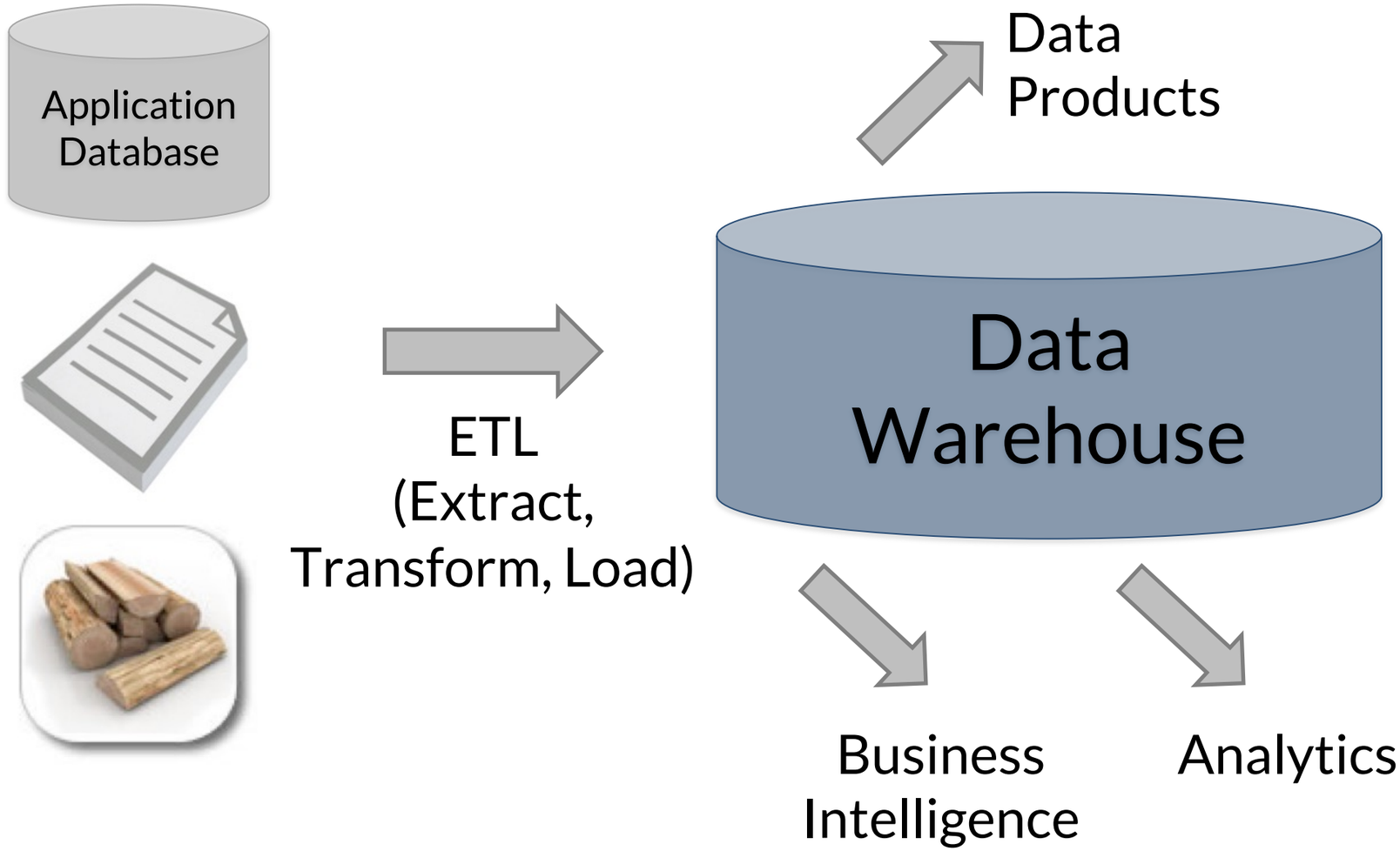
Nov 20th 2016 | United States

Timekeeper Like 916 Tweet

EARLY in America’s presidential campaign, pundits compared the contest between Hillary Clinton and Donald Trump to a fight between a large tanker and Somali pirates. This turned out to be particularly true of the digital campaigns: a massive data battleship lost to a chaotic flotilla of social-media speedboats. The big question now is what this means for future elections, both in America and abroad.



Standard Architecture



Instantiations(1) - Businesspersons

- Data Sources
  - Web pages
  - Excel
- Extract-Transform-Load (ETL)
  - Copy & paste
- Data Warehouse
  - Excel
- BI and Analytics
  - Excel functions
  - Excel charts
  - VB scripts?
  - Visualization tools: Power BI, Tableau

Instantiations(2) - Programmers

- Data Sources
  - Web scraping, web services API
  - CSV files
  - Database queries
- ETL
  - wget, curl, Beautiful Soup, lxml, …
- Data Warehouse
  - Files
- Analytics
  - Numpy, pandas, Matplotlib, R, Octave, …

Instantiations(3) - Enterprises

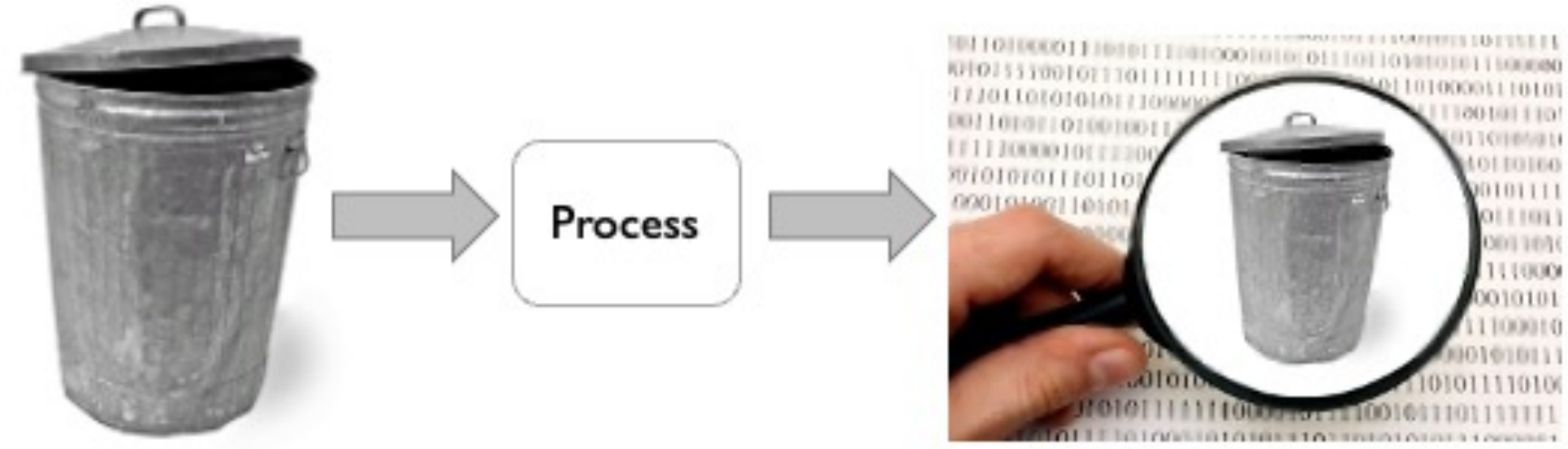
- Data Sources
  - Application databases(Oracle, IBM, …)
  - Intranet files
  - Application log files
- ETL
  - Infomatica, IBM DataStage, …
- Data Warehouse
  - Teradata, Oracle, IBM DB2, …
- Business Intelligence & Analytics
  - SAS, SPSS, R, …
  - Power BI, Tableau, Spotfire, …

Instantiations(4) – Web Companies

- Data Sources
  - Application databases
  - Logs
  - Web crawl data
- ETL
  - Apache Flume, Apache Sqoop, …
- Data Warehouse
  - Hadoop-based: Hive, Hbase
  - Microsoft Azure, Amazon Redshift
- Business Intelligence & Analytics
  - Argus, R, …

“Garbage in, garbage out.”

- Raw data can always be **DIRTY**!



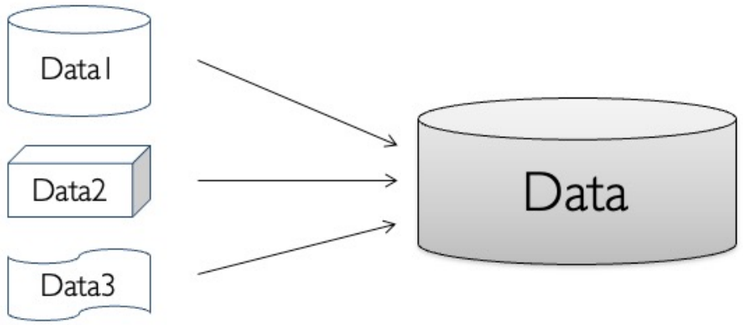
Data Quality

- Data quality: data has quality if it satisfies the requirements of its intended use
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Interpretability

Data Integration

- Data integration involves combining data residing in different sources and providing users with a unified view of these data.
  - Remember “views” in DBMS?

- Management of data from multiple sources



Customer (source 1)

CID	*Name*	*Street*	*City*	*Sex*
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Client (source 2)

Cno	*LastName*	*FirstName*	*Gender*	*Address*	*Phone/Fax*
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Customers (integrated target with cleaned data)

No	*LName*	*FName*	*Gender*	*Street*	*City*	*State*	*ZIP*	*Phone*	*Fax*	*CID*	*Cno*
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503-5998	444-555-6666		11	493
2	Smith	Christian	M	2 Hurley Place	South Fork	MN	48503-5998			24	
3	Smith	Christoph	M	23 Harley Street	Chicago	IL	60633-2394	333-222-6542	333-222-6599		24

Typical Data Cleaning and Integration Workflow

- Data analysis
  - Detailed inspection before operations
- Conflicts resolution
  - Resolve data confliction between data sources to be integrated
- Definition of transformation workflow and mapping rules
  - Workflow methods for schema adaption and transformation
- Verification of Workflow
  - Verify each steps
- Transformation
  - start the process

Load and Store Data

- File-based Storage
  - Simplest way & easy to manage
  - Scalability is low

- **Database & DBMS**
  - What we have learned for 10+ weeks

- Data Warehouse

Data Warehouse

A data warehouse is a **subject-oriented**, **integrated**, **time-varient**, and **nonvolatile** collection of data in support of management's decision making process.

-- W. H. Inmon, "Building the Data Warehouse". 1996.

Loosely Speaking, a data warehouse refers to a data repository that is **maintained separately** from an organization's operational databases.

-- J. Han and M. Kamber, "Data Mining: Concepts and Techniques", 3rd ed., 2011.

Differences between Databases and Data Warehouses

	DB	DW
Characteristics	operational processing	informational processing
Orientation	transaction	analysis
User	terminal users: clerk, database administrator(DBA)	knowledge workers: manager, analyst, executive
Function	everyday operations	long-term informational requirements decision support
Data	current, up-to-date	historic, accuracy maintained over time
Access	read/write	mostly read
Focus	data in	information/knowledge out
Size	GB to high-order GB	>=TB

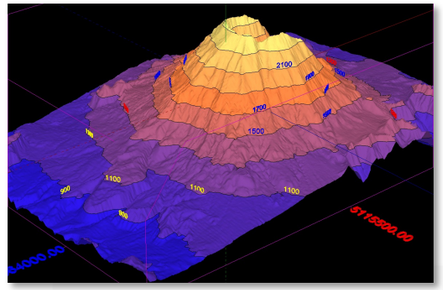
Data Analysis

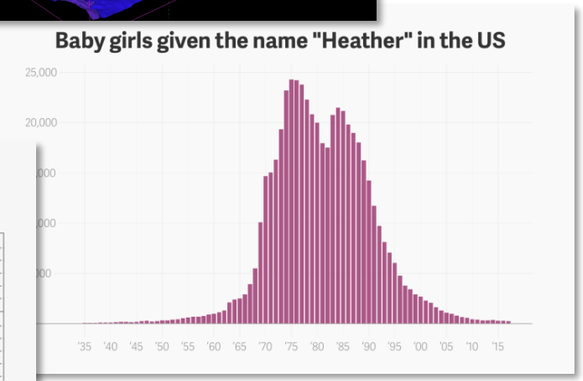
- Exploratory Data Analysis
- Data Mining

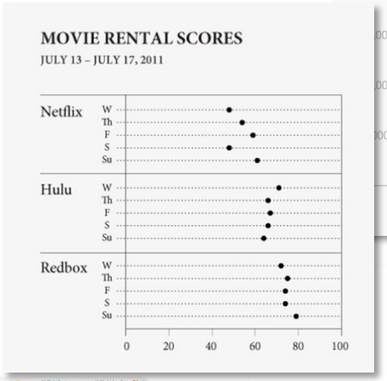
Exploratory Data Analysis (EDA)

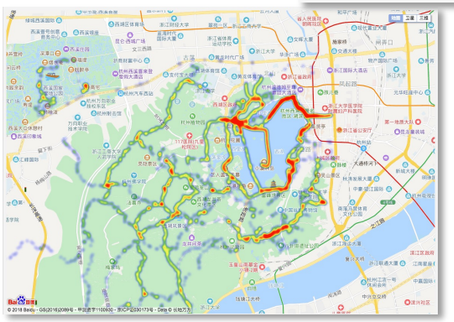
- Based on statistics
  - Data visualization-driven method
  - Summary of main characteristics in easy-to-understand form

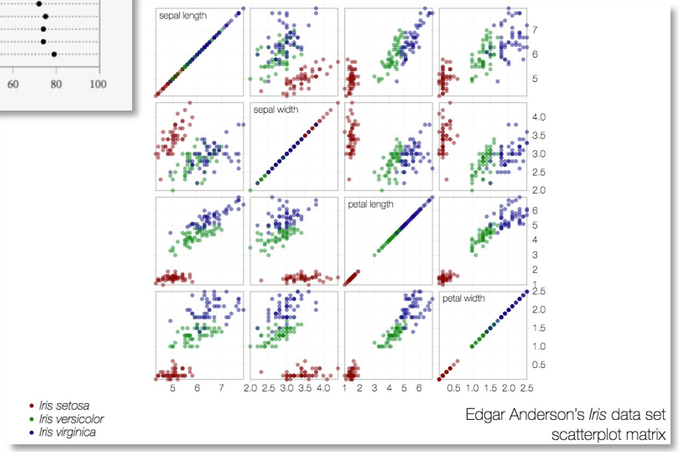
- Types of data visualization methods in EDA:
  - Plotting of raw data
  - Plotting of statistical values
  - Multiple coordinated views (Dashboard)







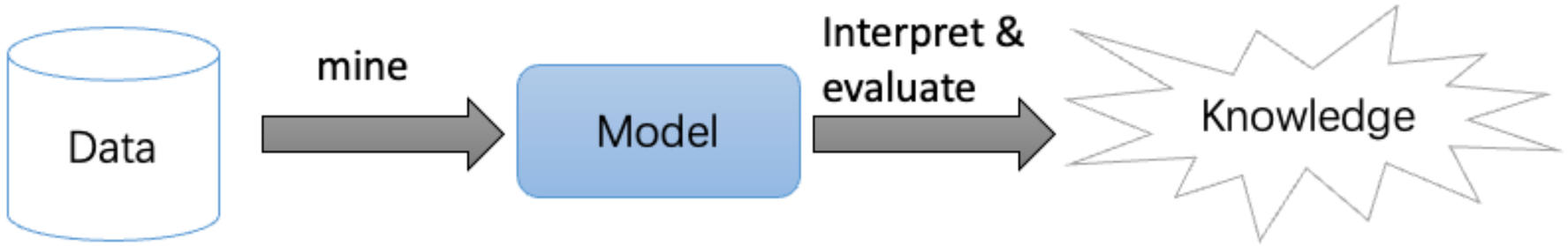




Data Mining

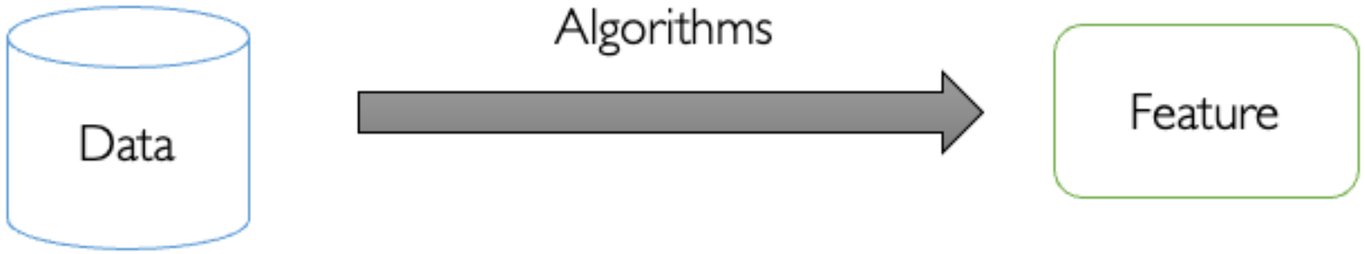
"**Data Mining**, also popularly referred to as **knowledge discovery** from data (KDD), is the automated or convenient **extraction of patterns representing knowledge** implicitly stored or captured in large databases, data warehouses, the Web, other massive repositories, or data streams."

-- H. Jiawei and M. Kamber, "Data Mining: Concepts and Techniques", 3rd ed., 2011.

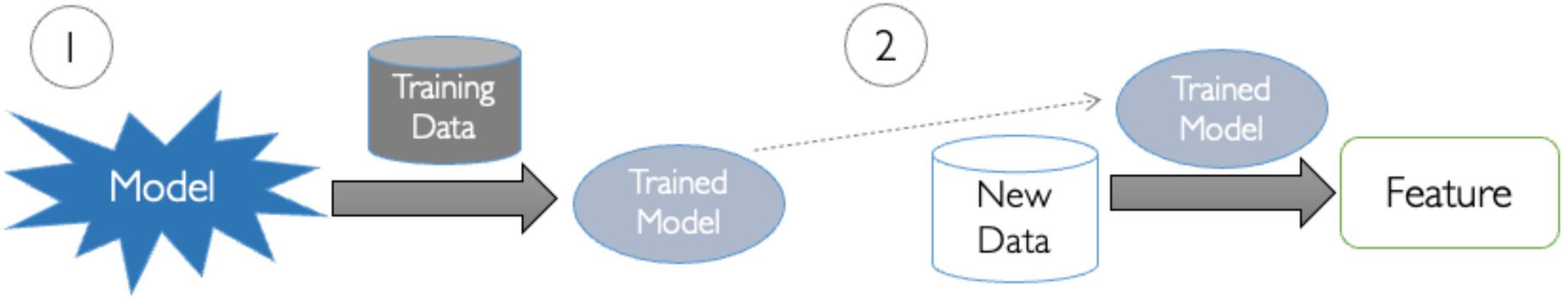


Tasks in Data Mining

- Descriptive Tasks



- Predictive Tasks



Descriptive Tasks

- Concept Description
  - Describe features of data directly
- Association Analysis
  - Analyze “feature-value” pairs that occur frequently in data
- Clustering
  - Group data on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity
- Outlier Detection
  - Analyze objects that do not comply with the general behavior or model of the data

Predictive Tasks

- Regression
  - Model the relationship between a scalar response and a number of variables
- Classification
  - Find a model/function that describes and distinguish data classes or concepts based on analysis of a set of training data
- Evolution Analysis
  - Analyze temporal and spatial patterns in dataset, model these patterns and predict data in unknown spatio-temporal positions

Data Visualization

- **Visualization** creation and **study** of t visual **representation of data**

Input: data

Output: visual form

Goal: insight

Why Do We Need Visualization?

- Sometimes, statistics may not work

Set A		Set B		Set C		Set D	
X	Y	X	Y	X	Y	X	Y
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.1	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.1	4	5.39	19	12.5
12	10.84	12	9.11	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

Summary Statistics
$u_X = 9.0$ $\sigma_X = 3.317$
$u_Y = 7.5$ $\sigma_Y = 2.03$

Linear Regression
$Y = 3 + 0.5\,X$
$R^2 = 0.67$

[Anscombe 73]