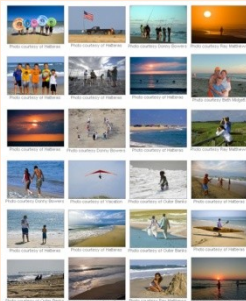
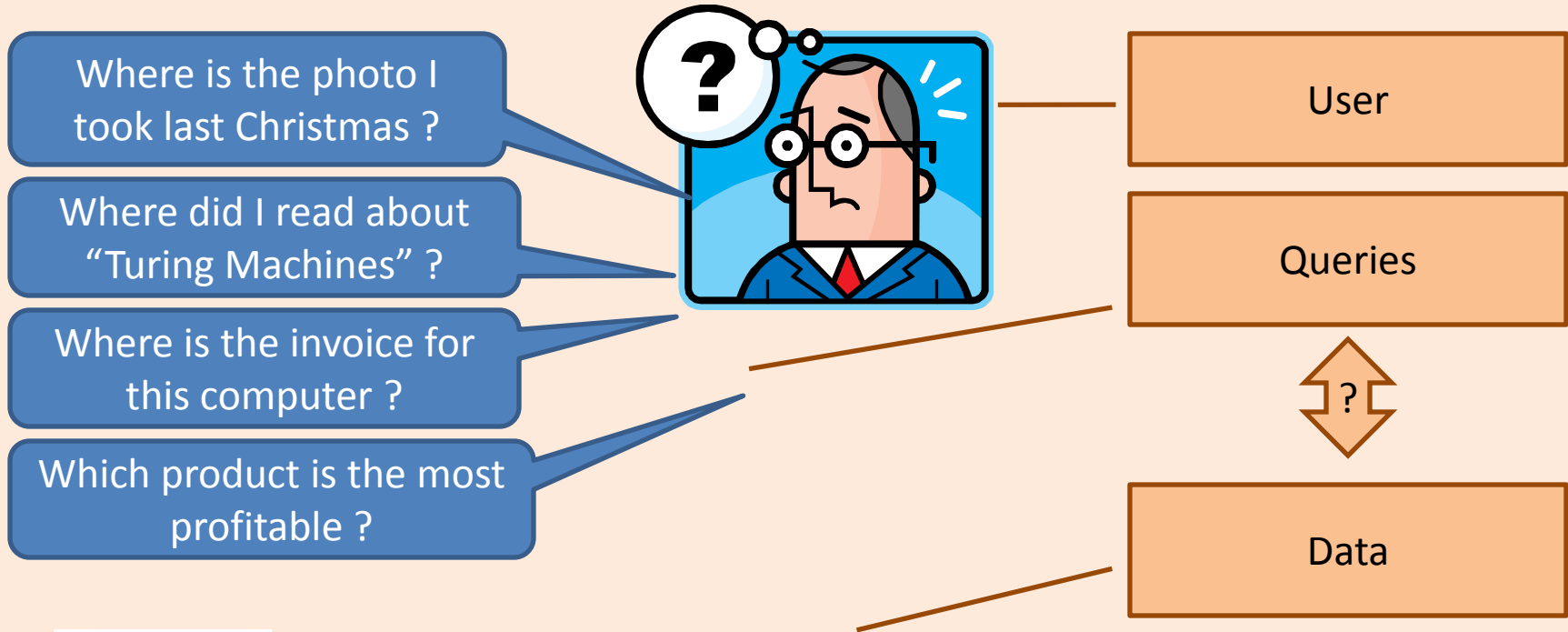


ICS 101 Fall 2013

Introduction to Data Management

Asst. Prof. Lipyeow Lim
Information & Computer Science Department
University of Hawaii at Manoa

The Data Management Problem



What is “data” ?

- **Data** are known facts that can be recorded and that have implicit meaning.
- Three broad categories of data
 - Structured data
 - Semi-structured data
 - Unstructured data
- “**Structure**” of data refers to the organization within the data that is identifiable.

What is a database ?

- A **database** : a collection of related data.
 - Represents some aspect of the real world (aka universe of discourse).
 - Logically coherent collection of data
 - Designed and built for specific purpose
- A **data model** is a collection of concepts for describing/organizing the data.
- A **schema** is a description of a particular collection of data, using the a given data model.

The Relational Data Model

- *Relational database*: a set of *relations*
- A *relation* is made up of 2 parts:
 - *Instance* : a *table*, with rows and columns.
#Rows = *cardinality*, #fields = *degree / arity*.
 - *Schema* : specifies name of relation, plus name and *domain/type* of each column or attribute.
 - E.G. Students(sid: string, name: string, login: string, age: integer, gpa: real).
- Can think of a relation as a *set* of rows or *tuples* (i.e., all rows are distinct).

Example Instance of Students Relation

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Q1. What is the cardinality of the relation instance?
(a) 1 (b) 2 (c) 3 (d) 4
- Q2. What is the degree/arity of the relation instance?
(a) 2 (b) 3 (c) 4 (d) 5

Why is the relational model useful ?

- Supports simple and powerful query capabilities!
- Structured Query Language (SQL)

```
SELECT S.sname  
FROM Students S  
WHERE S.gpa>3.5
```

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

What is a DBMS ?

- A **database management system (DBMS)** is a collection of programs that enables users to
 - **Create** new DBs and specify the structure using data definition language (DDL)
 - **Query** data using a query language or data manipulation language (DML)
 - **Store** very large amounts of data
 - Support **durability** in the face of failures, errors, misuse
 - Control **concurrent** access to data from many users

Types of Databases

- On-line Transaction Processing (**OLTP**)
 - Banking
 - Airline reservations
 - Corporate records
- On-line Analytical Processing (**OLAP**)
 - Data warehouses, data marts
 - Business intelligence (BI)
- Specialized databases
 - Multimedia
 - XML
 - Geographical Information Systems (GIS)
 - Real-time databases (telecom industry)
- Special Applications
 - Customer Relationship Management (CRM)
 - Enterprise Resource Planning (ERP)
- Hosted DB Services
 - Amazon, Salesforce

A Bit of History

- 1970 **Edgar F Codd** (aka “Ted”) invented the **relational model** in the seminal paper “A Relational Model of Data for Large Shared Data Banks”
 - Main concept: relation = a table with rows and columns.
 - Every relation has a schema, which describes the columns.
- Prior 1970, no standard data model.
 - Network model used by Codasyl
 - Hierarchical model used by IMS
- After 1970, IBM built System R as proof-of-concept for relational model and used **SQL** as the query language. SQL eventually became a standard.

Transactions

- A transaction is the DBMS's abstract view of a user program: a sequence of reads and writes.
 - Eg. User 1 views available seats and reserves seat 22A.
- A DBMS supports **multiple users**, ie, multiple transactions may be running **concurrently**.
 - Eg. User 2 views available seats and reserves seat 22A.
 - Eg. User 3 views available seats and reserves seat 23D.

ACID Properties of Transactions

- **Atomicity** : all-or-nothing execution of transactions
- **Consistency**: constraints on data elements is preserved
- **Isolation**: each transaction executes as if no other transaction is executing concurrently
- **Durability**: effect of an executed transaction must never be lost

Q3. Why use a DBMS ?

- a) The data is too large to manage in excel files
- b) I do not want to write my own programs to find something in the data
- c) I do not want to write my own program to manage multiple users and transactions
- d) All of the above.

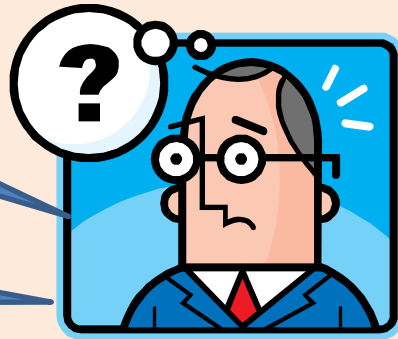
The Data Management Problem

Where is the photo I took last Christmas ?

Where did I read about "Turing Machines" ?

Where is the invoice for this computer ?

Which product is the most profitable ?

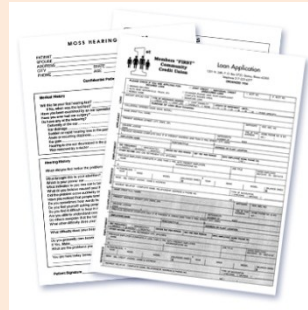
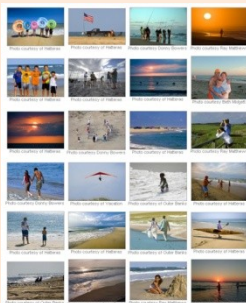


User

Queries



Data



Unstructured Data

- What are some examples of unstructured data?
- How do we model unstructured data ?
- How do we query unstructured data ?
- How do we process queries on unstructured data ?
- How do we index unstructured data ?

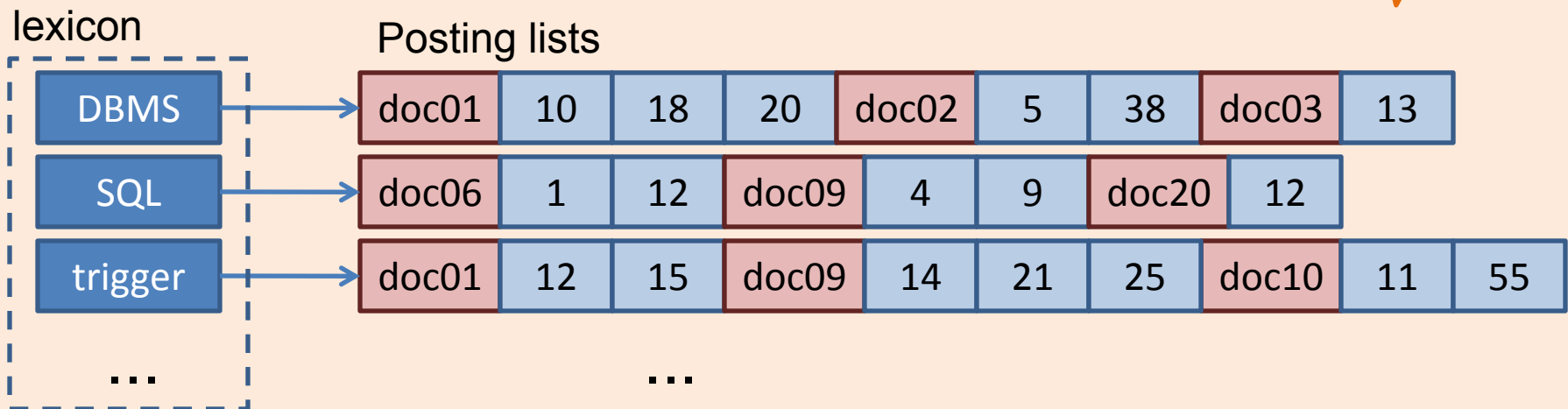
Unstructured Text Data

- Field of “**Information Retrieval**”
- Data Model
 - Collection of documents
 - Each document is a **bag of words (aka terms)**
- Query Model
 - **Keyword** + Boolean Combinations
 - Eg. DBMS and SQL and tutorial
- Details:
 - Not all words are equal. “**Stop words**” (eg. “the”, “a”, “his” ...) are ignored.
 - **Stemming** : convert words to their basic form. Eg. “Surfing”, “surfed” becomes “surf”

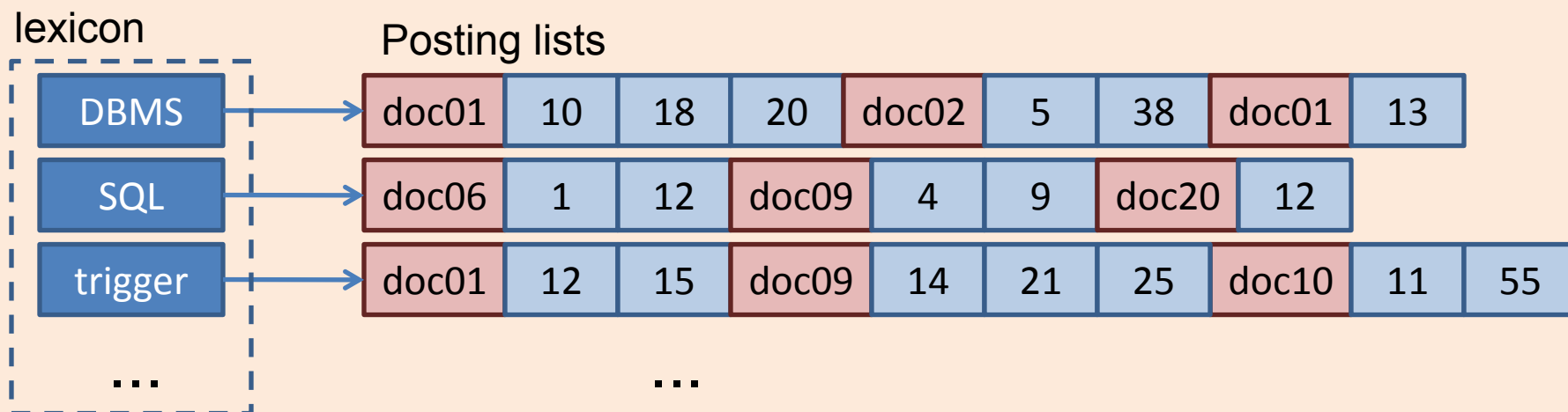
Inverted Indexes

- Recall: an index is a mapping of search key to data entries
 - What is the search key ?
 - What is the data entry ?
- Inverted Index:
 - For each term store a list of postings
 - A posting consists of <docid,position> pairs

What is the data in an inverted index sorted on ?



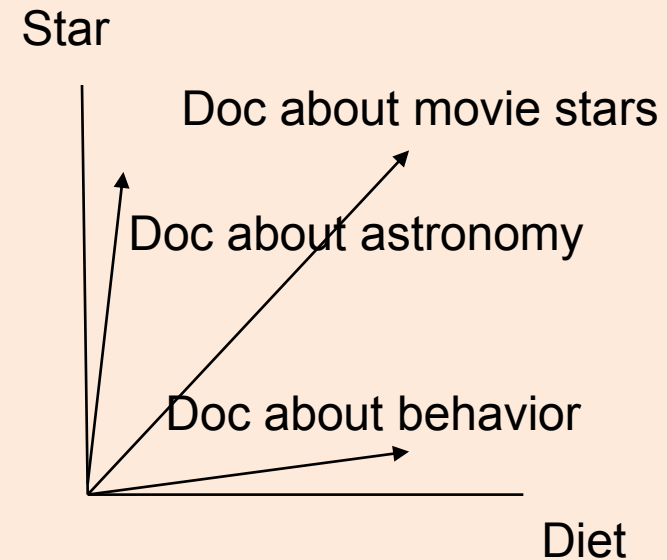
Lookups using Inverted Indexes



- Given a **single keyword query "k"** (eg. SQL)
 - Find k in the lexicon
 - Retrieve the posting list for k
 - Scan posting list for document IDs [and positions]
- What if the query is **"k1 and k2"** ?
 - Retrieve document IDs for k1 and k2
 - Perform intersection

Too Many Matching Documents

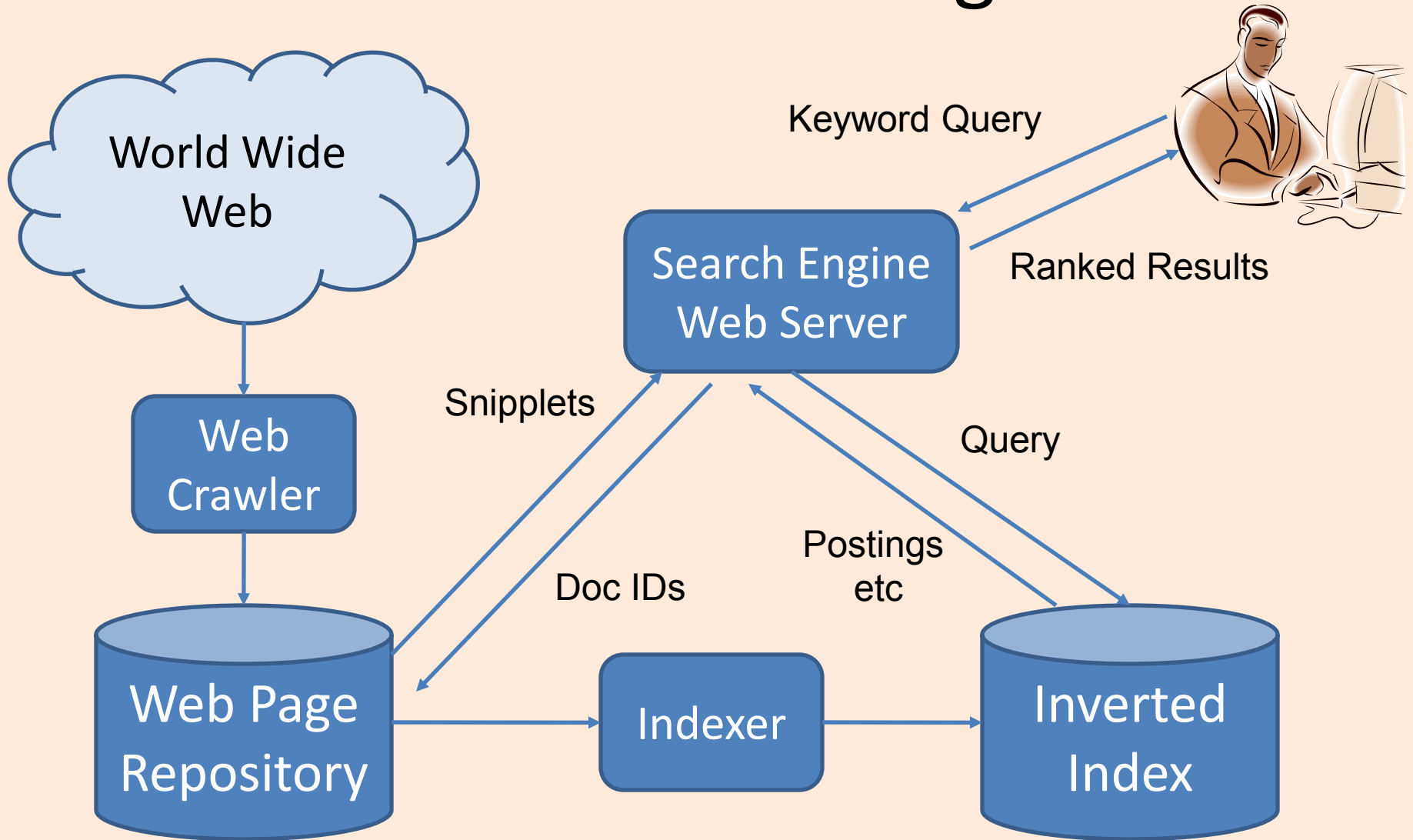
- Rank the results by “relevance”!
- Vector-Space Model
 - Documents are **vectors** in hi-dimensional space
 - Each dimension in the vector represents a term
 - **Queries** are represented as **vectors** similarly
 - **Vector distance** (dot product) between query vector and document vector gives ranking criteria
 - **Weights** can be used to tweak relevance
- PageRank (later)



Q4. Which of the following is the most similar to an inverted index ?

- a) Bookmarks.
- b) Content page of a book.
- c) The index at the end of a book.
- d) A deck of playing cards.

Internet Search Engines



Basic Web Search

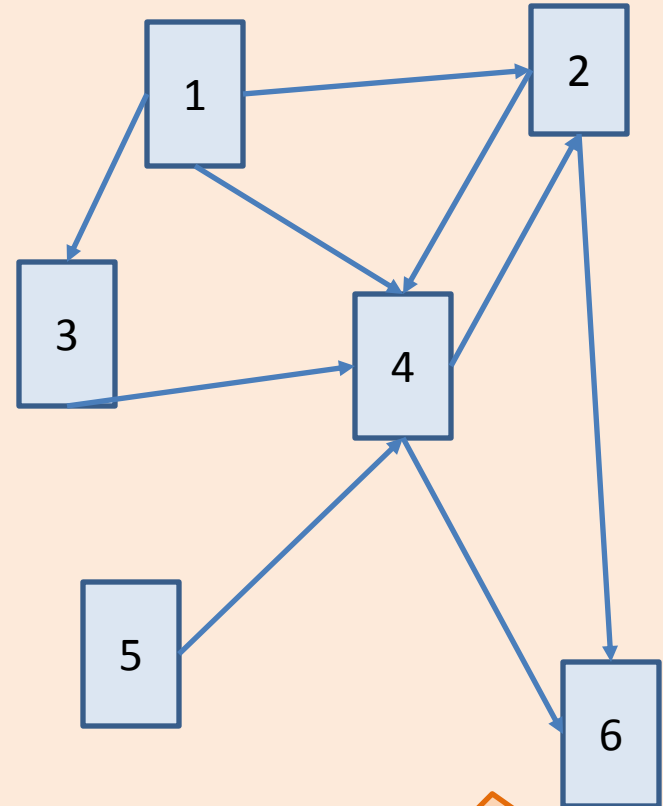
- http://www.googleguide.com/advanced_operators_reference.html

Query Expression	What it means
Biking italy	Biking AND italy
Recycle steel OR iron	Recycle AND (steel OR iron)
"I have a dream"	"I have a dream" treated as one term
Salsa -dance	Salsa AND NOT dance

Other nifty expressions	What it means
$12 + 34 - 56 * 7 / 8$	Evaluates the arithmetic expression
300 Euros in USD	Converts 300 euros to US currency

Ranking Web Pages

- Google's **PageRank**
 - Links in web pages provide clues to **how important a webpage** is.
- Take a **random walk**
 - Start at some webpage p
 - Randomly pick one of the links and go to that webpage
 - Repeat for all eternity
- The **number of times** the walker visits a page is an indication of how **important** the page is.



Vertices represent web pages.
Edges represent web links.

Semi-structured Search

Web pages are not really unstructured! Click “view source” to view HTML.

Query Expression	What it means
define: imbroglio	Find definitions of “imbroglio”
Halloween site: www.census.gov	Restrict search for “halloween” to US census website
Form 1098-T IRS filetype: pdf	Find the US tax form 1098-T in PDF format
link: warriorlibrarian.com	Find pages that link to Warrior Librarian's website
Dan Shugar intext: Powerlight	Find pages mentioning Dan Shugar where his company, Powerlight , is included in the text of the page, i.e., less likely to be from the corporate website.
allintitle: Google Advanced Operators	Search for pages with titles containing "Google," "Advanced,", and "Operators"

Summary

- Data Management Problem
 - How do we pose and answer queries on data?
- Structured data
 - Relational Data Model
 - SQL
 - Relational DBMS
 - Transactions
- Unstructured data
 - Bag of terms
 - Boolean combination of keyword queries
 - Inverted Indexes (Web Search Engines)
- Semi-structured data
 - Could use techniques from either structured or unstructured
 - More sophisticated keyword queries