Misc: OLAP and Data Cubes; Information Retrieval

Amol Deshpande
CMSC424

Spring 2020 – Online Instruction Plan

- Week 1: File Organization and Indexes
- Week 2: Query Processing
- Week 3: Query Optimization; Parallel Databases 1
- Week 4: Parallel Databases; Mapreduce; Transactions 1
- Week 5: Transactions 2
- Week 6: Homework Due May 8
  - Transactions: Recovery
  - Misc 1: Distributed Transactions, and Object-oriented/Object-relational databases
  - Misc 2: OLAP and Data Cubes, and Information Retrieval

OLAP and Data Cubes

- Book Chapters
  - 5.7
- Key topics:
  - Data Warehouses
  - Star and Snowflake Schemas
  - Data Cubes

Data Warehouses

- A repository of integrated information for querying and analysis purposes
- A (usually) stand-alone system that integrates data from everywhere
  - Read-only, typically not kept up-to-date with the *real* data
  - Geared toward business analytics, data mining etc…
  - HUGE market today
- Heavily optimized
  - Specialized query processing and indexing techniques are used
  - High emphasis on pre-computed data structures like summary tables, **data cubes**
- Analysis cycle:
  - Extract data from databases with queries, visualize/analyze with desktop tools
  - E.g., Tableau

Data Warehouses

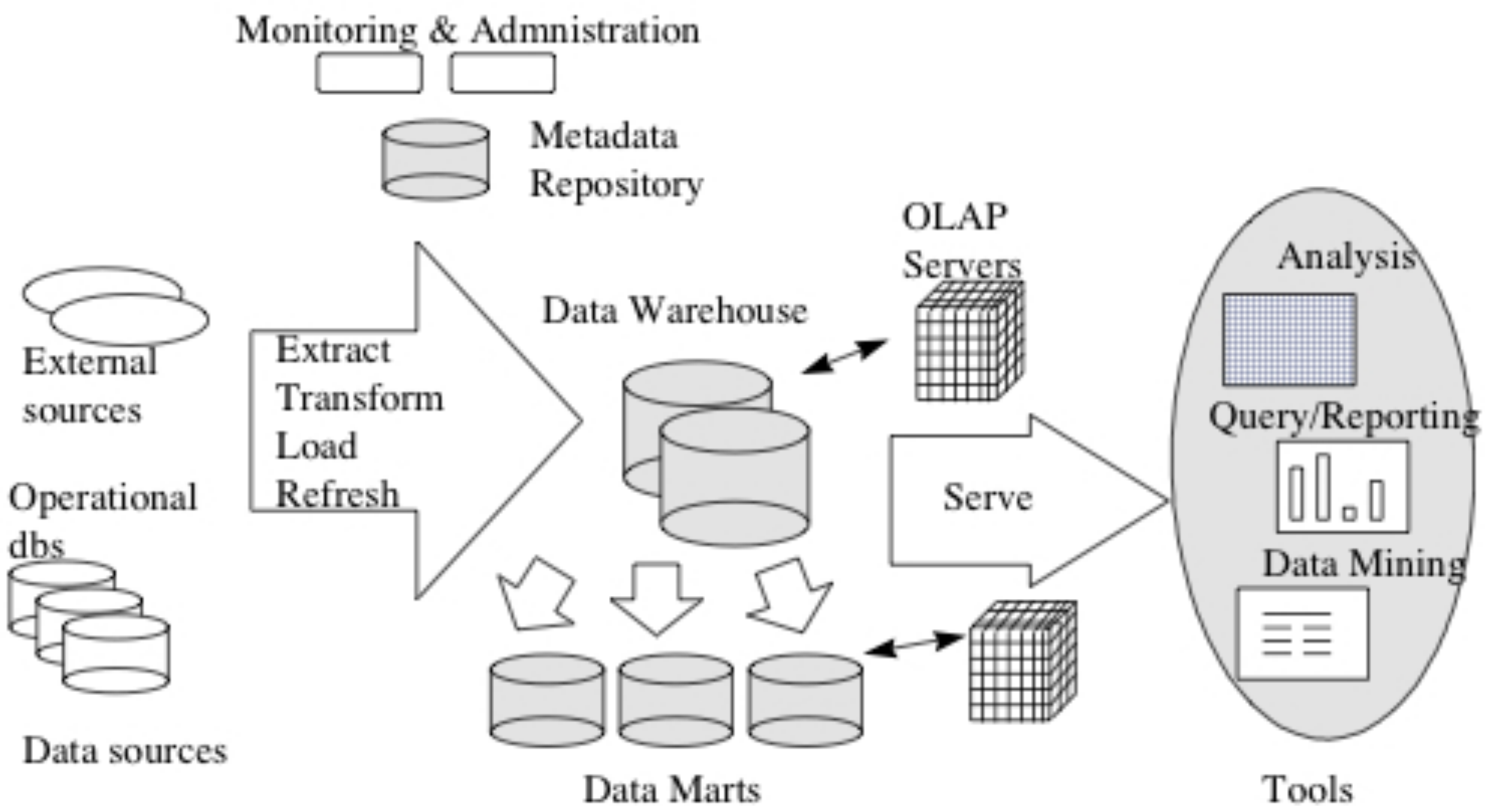


Figure 1. Data Warehousing Architecture

Data Warehouses

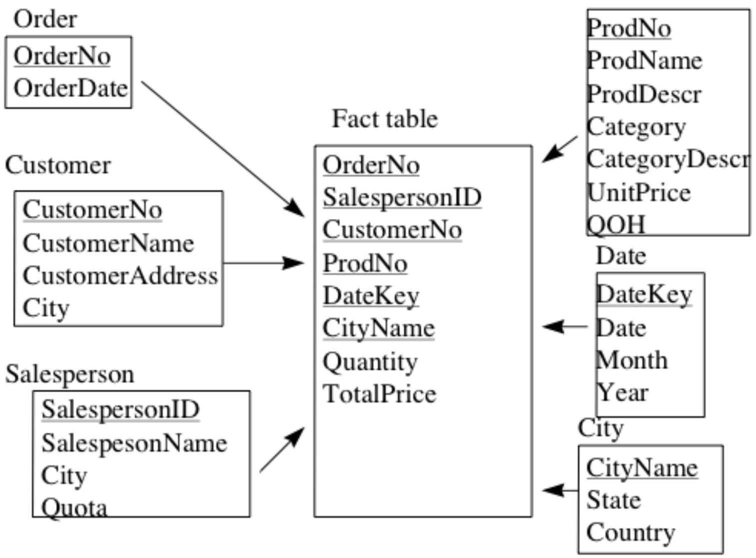


Figure 3. A Star Schema.

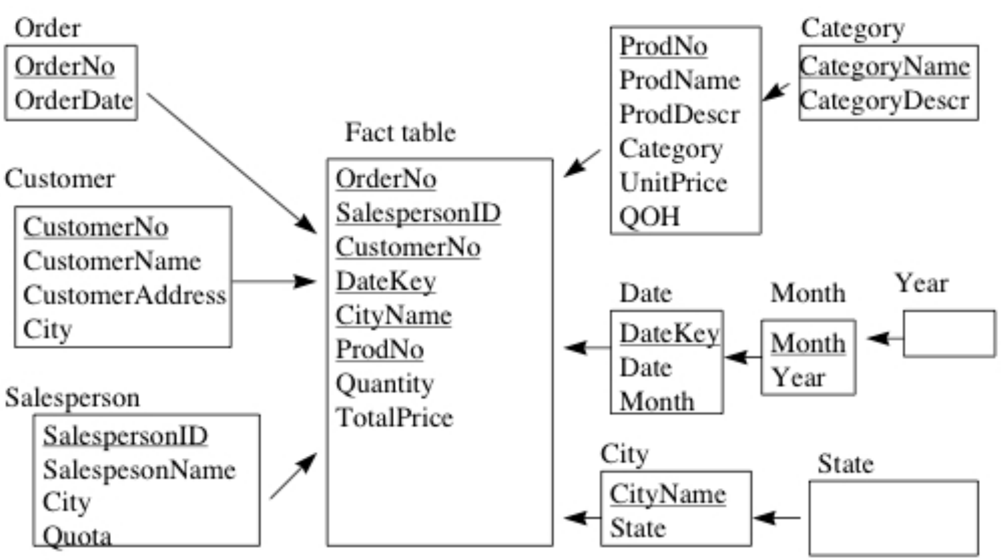


Figure 4. A Snowflake Schema.

Query processing algorithms heavily optimized for these types of schemas

Many queries of the type:

Selections on dimension tables (e.g., state = 'MD')

Join fact table with dimension tables

Aggregate on a "measure" attribute (e.g., Quantity, TotalPrice)

For example:

select c_city, o_year, SUM(quantity)
from Fact, Customer, Product
where p_category = 'Tablet';

OLAP

- On-line Analytical Processing
- Why ?
  - Exploratory analysis
    - Interactive
    - Different queries than typical SPJ SQL queries
  - Data CUBE
    - A summary structure used for this purpose
      - E.g. *give me total sales by zipcode; now show me total sales by customer employment category*
    - Much much faster than using SQL queries against the raw data
      - The tables are *huge*
- Applications:
  - Sales reporting, Marketing, Forecasting etc etc

Data Analysis and OLAP

- **Online Analytical Processing (OLAP)**
  - Interactive analysis of data, allowing data to be summarized and viewed in different ways in an online fashion (with negligible delay)
- Data that can be modeled as dimension attributes and measure attributes are called **multidimensional data**.
  - **Measure attributes**
    - measure some value
    - can be aggregated upon
    - e.g., the attribute *number* of the *sales* relation
  - **Dimension attributes**
    - define the dimensions on which measure attributes (or aggregates thereof) are viewed
    - e.g., attributes *item_name, color,* and *size* of the *sales* relation

Example sales relation

item_name	*color*	*clothes_size*	*quantity*
skirt	dark	small	2
skirt	dark	medium	5
skirt	dark	large	1
skirt	pastel	small	11
skirt	pastel	medium	9
skirt	pastel	large	15
skirt	white	small	2
skirt	white	medium	5
skirt	white	large	3
dress	dark	small	2
dress	dark	medium	6
dress	dark	large	12
dress	pastel	small	4
dress	pastel	medium	3
dress	pastel	large	3
dress	white	small	2
dress	white	medium	3
dress	white	large	0
shirt	dark	small	2
shirt	dark	medium	6
...	...	...	...
...	...	...	...

Cross Tabulation of *sales* by *item_name* and *color*

clothes_size **all**

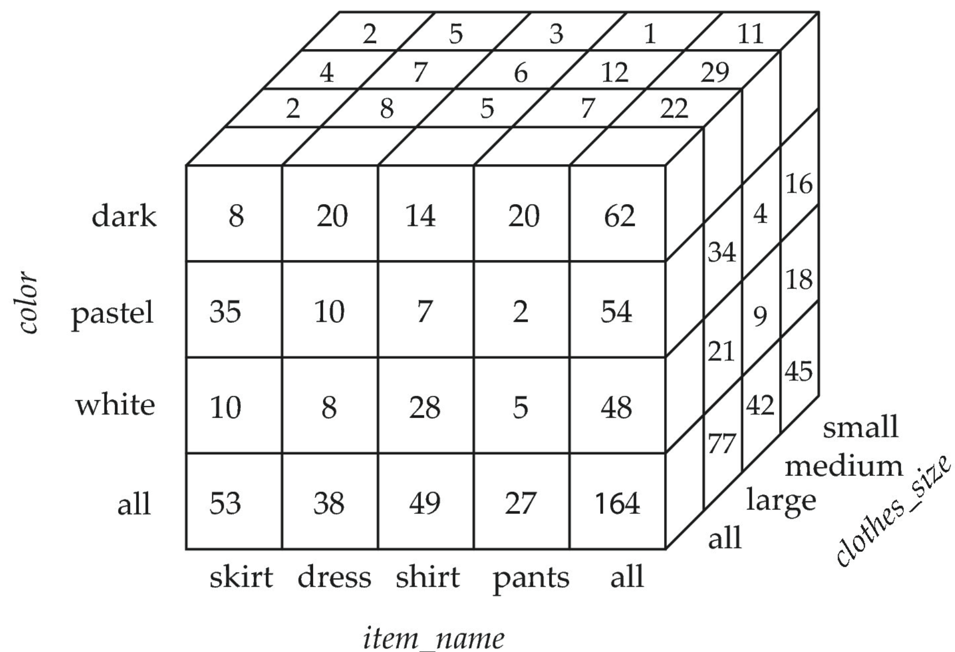
color

item_name	dark	pastel	white	total
skirt	8	35	10	53
dress	20	10	5	35
shirt	14	7	28	49
pants	20	2	5	27
total	62	54	48	164

- Example of a **cross-tabulation** (**cross-tab**), or a **pivot-table**.
  - Values for one of the dimension attributes form the row headers
  - Values for another dimension attribute form the column headers
  - Other dimension attributes are listed on top
  - Values in individual cells are (aggregates of) the values of the dimension attributes that specify the cell.

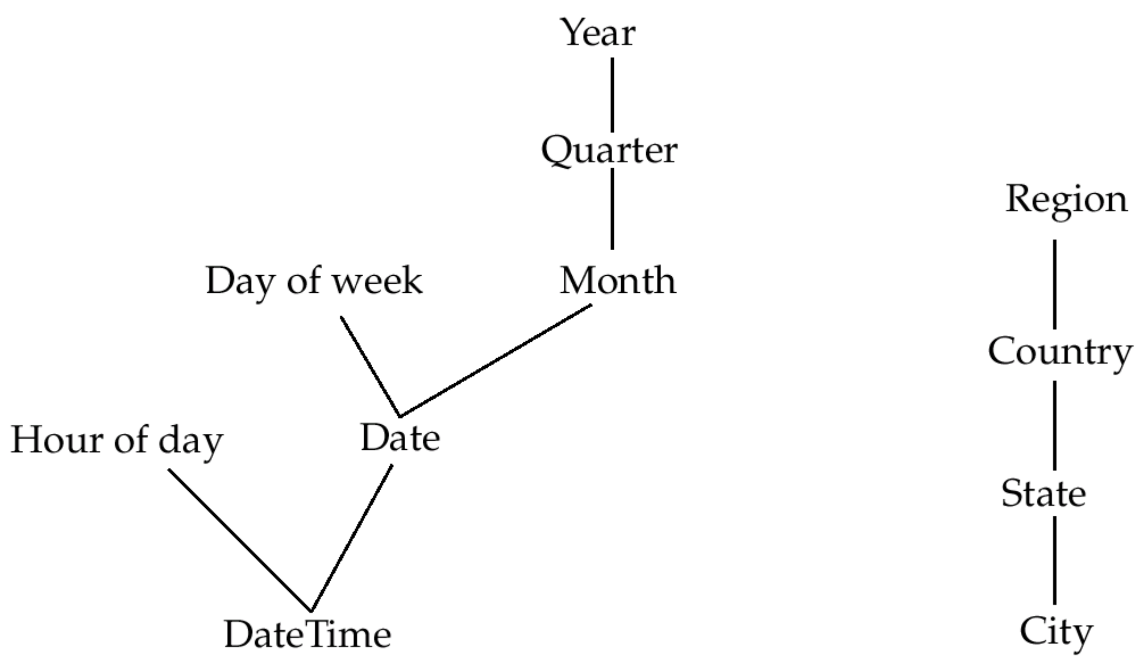
Data Cube

- A **data cube** is a multidimensional generalization of a cross-tab
- Can have *n* dimensions; we show 3 below
- Cross-tabs can be used as views on a data cube



Hierarchies on Dimensions

- **Hierarchy** on dimension attributes: lets dimensions to be viewed at different levels of detail
  - ★ E.g., the dimension DateTime can be used to aggregate by hour of day, date, day of week, month, quarter or year



a) Time Hierarchy

b) Location Hierarchy

Cross Tabulation With Hierarchy

- Cross-tabs can be easily extended to deal with hierarchies
  - Can drill down or roll up on a hierarchy

clothes_size: **all**

category	*item_name*	*color*				
		dark	pastel	white	total	
womenswear	skirt	8	8	10	53	
	dress	20	20	5	35	
	subtotal	28	28	15		88
menswear	pants	14	14	28	49	
	shirt	20	20	5	27	
	subtotal	34	34	33		76
total		62	62	48		164

Relational Representation of Cross-tabs

- Cross-tabs can be represented as relations
  - We use the value **all** is used to represent aggregates.
  - The SQL standard actually uses null values in place of **all** despite confusion with regular null values.

item_name	*color*	*clothes_size*	*quantity*
skirt	dark	**all**	8
skirt	pastel	**all**	35
skirt	white	**all**	10
skirt	**all**	**all**	53
dress	dark	**all**	20
dress	pastel	**all**	10
dress	white	**all**	5
dress	**all**	**all**	35
shirt	dark	**all**	14
shirt	pastel	**all**	7
shirt	White	**all**	28
shirt	**all**	**all**	49
pant	dark	**all**	20
pant	pastel	**all**	2
pant	white	**all**	5
pant	**all**	**all**	27
all	dark	**all**	62
all	pastel	**all**	54
all	white	**all**	48
all	**all**	**all**	164

Extended Aggregation to Support OLAP

- The **cube** operation computes union of **group by**' s on every subset of the specified attributes

  sales(*item_name, color, clothes_size, quantity*)

- Consider the query

  select *item_name, color, size,* **sum**(*number*)
  from *sales*
  group by cube(*item_name, color, size*)

Computers a union of eight different groupings of the *sales* relation:

{ (*item_name, color, size*), (*item_name, color*),
(*item_name, size*), (*color, size*),
(*item_name*), (*color*),
(*size*), ( ) }

where ( ) denotes an empty **group by** list.

Extended Aggregation (Cont.)

- The **rollup** construct generates union on every prefix of specified list of attributes

- E.g.,

  select *item_name*, *color*, *size*, **sum**(*number*)
  from *sales*
  group by rollup(*item_name, color, size*)

  Generates union of four groupings:

  { (*item_name, color, size*), (*item_name, color*), (*item_name*), ( ) }

Extended Aggregation (Cont.)

- Multiple rollups and cubes can be used in a single group by clause
  - Each generates set of group by lists, cross product of sets gives overall set of group by lists
- E.g.,

  select *item_name, color, size*, **sum**(*number*)
  from *sales*
  group by rollup(*item_name*), **rollup**(*color, size*)

  generates the groupings

  {item_name, ()} X {(color, size), (color), ()}

  = { (*item_name, color, size*), (*item_name, color*), (*item_name*), (*color, size*), (*color*), ( ) }

Online Analytical Processing Operations

- **Pivoting:** changing the dimensions used in a cross-tab is called
- **Slicing:** creating a cross-tab for fixed values only
  - Sometimes called **dicing**, particularly when values for multiple dimensions are fixed.
- **Rollup:** moving from finer-granularity data to a coarser granularity
- **Drill down:** The opposite operation - that of moving from coarser-granularity data to finer-granularity data

OLAP Implementation

- The earliest OLAP systems used multidimensional arrays in memory to store data cubes, and are referred to as **multidimensional OLAP (MOLAP)** systems.
- OLAP implementations using only relational database features are called **relational OLAP (ROLAP)** systems
- Hybrid systems, which store some summaries in memory and store the base data and other summaries in a relational database, are called **hybrid OLAP (HOLAP)** systems.

Data Mining

- Searching for patterns in data
  - Typically done in data warehouses
- Association Rules:
  - When a customer buys X, she also typically buys Y
  - Use ?
    - Move X and Y together in supermarkets
  - A customer buys a lot of shirts
    - Send him a catalogue of shirts
  - Patterns are not always obvious
    - Classic example: It was observed that men tend to buy *beer* and *diapers* together (may be an urban legend)
- Other types of mining
  - Classification
  - Decision Trees

Summary

- Data analytics a major industry right now, and likely to grow in near future
  - BIG Data !!
  - Extracting (actionable) knowledge from data really critical
    - Especially in real-time
- Some key technologies:
  - Parallelism – pretty much required
  - Column-oriented design
    - Lay out the data column-by-column, rather than row-by-row
  - Heavy pre-computation (like Cubes)
  - New types of indexes
    - Focusing on bitmap representations
  - Heavy compression
  - Map-reduce??

Information Retrieval

- Book Chapters
  - Chapter 21 – at a fairly high level
- Key topics:
  - What is Information Retrieval?
  - IF-TDF
  - Web crawling and searching

Information Retrieval Systems

- **Information retrieval (IR)** systems use a simpler data model than database systems
  - Information organized as a collection of documents
  - Documents are unstructured, no schema
- Information retrieval locates relevant documents, on the basis of user input such as keywords or example documents
  - e.g., find documents containing the words “database systems”
- Can be used even on textual descriptions provided with non-textual data such as images
- Web search engines are the most familiar example of IR systems

Information Retrieval Systems (Cont.)

- Differences from database systems
  - IR systems don't deal with transactional updates (including concurrency control and recovery)
  - Database systems deal with structured data, with schemas that define the data organization
  - IR systems deal with some querying issues not generally addressed by database systems
    - Approximate searching by keywords
    - Ranking of retrieved answers by estimated degree of relevance

Keyword Search

- In **full text** retrieval, all the words in each document are considered to be keywords.
  - We use the word **term** to refer to the words in a document
- Information-retrieval systems typically allow query expressions formed using keywords and the logical connectives *and, or,* and *not*
  - *And*s are implicit, even if not explicitly specified
- Ranking of documents on the basis of estimated relevance to a query is critical
  - Relevance ranking is based on factors such as
    - Term frequency
      - Frequency of occurrence of query keyword in document
    - Inverse document frequency
      - How many documents the query keyword occurs in
        - Fewer ➔ give more importance to keyword
    - Hyperlinks to documents
      - More links to a document ➔ document is more important

Relevance Ranking Using Terms

- **TF-IDF** (Term frequency/Inverse Document frequency) ranking:
  - Let $n(d)$ = number of terms in the document d
  - $n(d, t)$ = number of occurrences of term t in the document d.
  - Relevance of a document d to a *term* t

$$TF(d, t) = \log\left(1 + \frac{n(d, t)}{n(d)}\right)$$

    - The log factor is to avoid excessive weight to frequent terms
  - Relevance of document to *query Q*

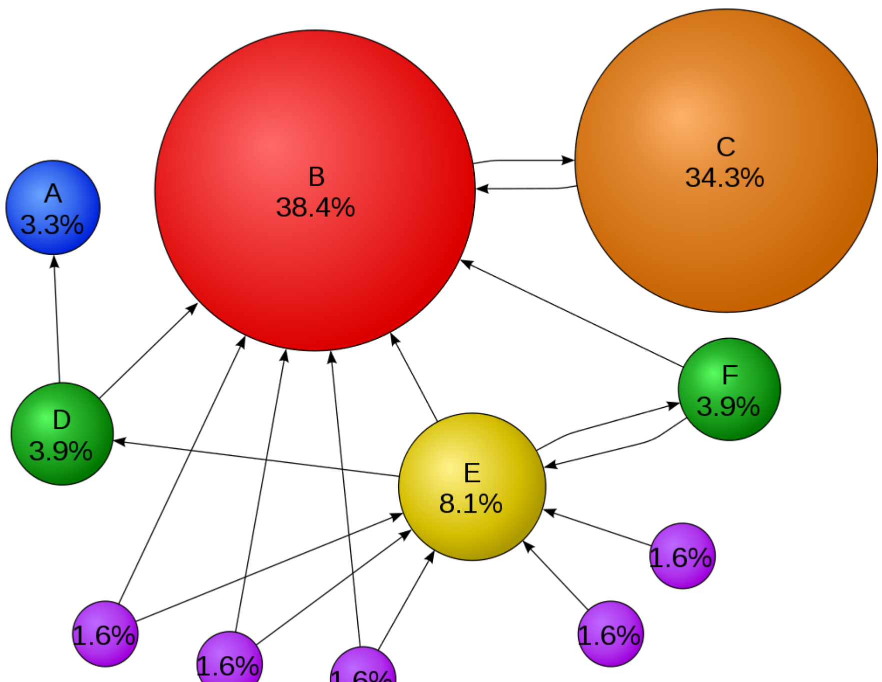
$$r(d, Q) = \sum_{t \in Q} \frac{TF(d, t)}{n(t)}$$

Relevance Ranking Using Terms (Cont.)

- Most systems add to the above model
  - Words that occur in title, author list, section headings, etc. are given greater importance
  - Words whose first occurrence is late in the document are given lower importance
  - Very common words such as “a”, “an”, “the”, “it” etc. are eliminated
    - Called **stop words**
  - **Proximity**: if keywords in query occur close together in the document, the document has higher importance than if they occur far apart
- Documents are returned in decreasing order of relevance score
  - Usually only top few documents are returned, not all

PageRank: Ranking based on hyperlinks

- The probability that a random surfer (who follows links randomly) will end up at a particular page
  - **Intuitively:** Higher the probability, the more important the page
- Surfer model:
  - Choose a random page to visit with probability "alpha"
  - If the number of outgoing edges = n, then visit one of those pages with probability (1 – alpha)/n



Indexing of Documents

- An inverted index maps each keyword K_i to a set of documents S_i that contain the keyword
  - Documents identified by identifiers
- Inverted index may record
  - Keyword locations within document to allow proximity based ranking
  - Counts of number of occurrences of keyword to compute TF
- **and** operation: Finds documents that contain all of K_1, K_2, ..., K_n.
  - Intersection $S_1 \cap S_2 \cap \ldots\ldots \cap S_n$
- **or** operation: documents that contain at least one of K_1, K_2, …, K_n
  - union, $S_1 \cap S_2 \cap \ldots\ldots \cap S_n$,.
- Each S_i is kept sorted to allow efficient intersection/union by merging
  - "**not**" can also be efficiently implemented by merging of sorted lists

Measuring Retrieval Effectiveness

- Information-retrieval systems save space by using index structures that support only approximate retrieval. May result in:
  - **false negative (false drop)** - some relevant documents may not be retrieved.
  - **false positive** - some irrelevant documents may be retrieved.
  - For many applications a good index should not permit any false drops, but may permit a few false positives.
- Relevant performance metrics:
  - **precision** - what percentage of the retrieved documents are relevant to the query.
  - **recall** - what percentage of the documents relevant to the query were retrieved.

Measuring Retrieval Effectiveness

- Recall vs. precision tradeoff:
    - Can increase recall by retrieving many documents (down to a low level of relevance ranking), but many irrelevant documents would be fetched, reducing precision
- Measures of retrieval effectiveness:
  - Recall as a function of number of documents fetched, or
  - **Precision as a function of recall**
    - Equivalently, as a function of number of documents fetched
  - E.g., “precision of 75% at recall of 50%, and 60% at a recall of 75%”
- Problem: which documents are actually relevant, and which are not

Web Search Engines

- **Web crawlers** are programs that locate and gather information on the Web
  - Recursively follow hyperlinks present in known documents, to find other documents
    - Starting from a *seed* set of documents
  - Fetched documents
    - Handed over to an indexing system
    - Can be discarded after indexing, or store as a *cached* copy
- Crawling the entire Web would take a very large amount of time
  - Search engines typically cover only a part of the Web, not all of it
  - Take months to perform a single crawl

Summary and More

- Information retrieval a very mature field, that developed largely in parallel to databases

- Much work on:
  - similarity search (to find similar documents)
  - better search and ranking algorithms
  - natural language question/answering
  - answer diversification (imagine searching for “apple”)
  - … and so on

- SIGKDD, SIGIR, WWW the main research conferences
  - Vs SIGMOD, VLDB, ICDE for databases