CMSC424: Database Design
Introduction/Overview

Instructor: Amol Deshpande
amol@cs.umd.edu

Today

- Motivation: Why study databases ? What is databases ?
- Administrivia
  - Workload etc.
- Current Industry Outlook
- A typical DBMS at a glance
- No laptop use allowed in the class !!

Some To-Dos

- Sign up for CampusWire !
- Set up the computing environment (project0), and make sure you can run Docker, PostgreSQL, IPython, etc.
- Upcoming: Reading Homework 1, Project 1: SQL

Motivation: Data Overload

- Explosion of data, in pretty much every domain
  - Sensing devices and sensor networks that can monitor everything 24/7 from temperature to pollution to vital signs
  - Increasingly sophisticated smart phones
  - Internet, social networks makes it easy to publish data
  - Scientific experiments and simulations produce astronomical volumes of data
  - Internet of Things
  - Dataification: taking all aspects of life and turning them into data (e.g., what you like/enjoy turned into a stream of your "likes")
- How to handle that data? How to extract interesting actionable insights and scientific knowledge?
- Data volumes expected to get much worse

Four V's of Big Data

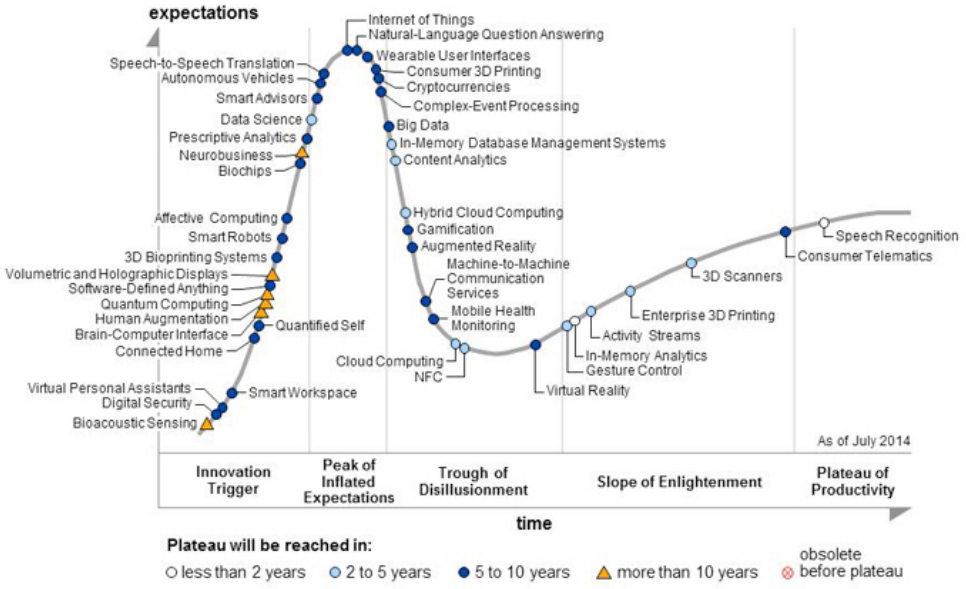
- Increasing data Volumes
  - Scientific data: 1.5GB/genome -- can be sequenced in .5 hrs; LHC generates 100TB of data a day
  - 500M tweets per day (as of 2013)
  - As of 2012: 2.5 Exabytes of data created every day
  - EBay: Two data warehouses with 7.5PB and 40PB
  - Walmart: 583 terabytes of sales and inventory data
  - FICO monitors 2.5 billion active accounts worldwide
- Variety:
  - Structured data, spreadsheets, photos, videos, natural text, ...
- Velocity
- Veracity

Four V's of Big Data

- Increasing data Volumes
- Variety
- Velocity
  - Sensors everywhere -- can generate tremendous volumes of "data streams"
  - Real-time analytics requires data to be consumed as fast as it is generated
- Veracity
  - How do you decide what to trust? How to remove noise? How to fill in missing values?

Big Data and Data Science to the Rescue

- Terms increasingly used synonymously: also data analytics, data mining, business intelligence
  - Loosely used for any process where interesting things are inferred from data
  - Google search: “How Big Data Will Change”
- Data scientist called the sexiest job of the 21st century
  - The term has becoming very muddled at this point
- Overhyped words
  - We are headed toward the trough of Disillusionment



Is it all hype?

- No: Extracting insights and knowledge from data very important, and will continue to increase in importance
  - Big data techniques are revolutionizing things in many domains like Education, Food Supply, Disease Epidemics, ...
- But: it is not much different from what we, especially statisticians, have been doing for many years
- What is different?
  - Much more data is digitally available than was before
  - Inexpensive computing + Cloud + Easy-to-use programming frameworks = Much easier to analyze it
  - Often: large-scale data + simple algorithms > small data + complex algorithms
    - Changes how you do analysis dramatically

Motivation: Data Overload

- How do we do anything with this data?
- Where and how do we store it ?
  - Disks are doubling every 18 months or so -- not enough
  - In many cases, the data is not actually recorded as it is; *summarized* first
- What if the disks crash ?
  - Very common, especially with 10,000's of disks
- How do we ensure "correctness" ?
  - What if the system crashes in the middle of an ATM transaction ?
    - Can't have money disappearing
  - What happens when a million people try to buy tickets to *<your favorite artist>'s* concert at the same time ?

Motivation: Data Overload

- What to do with the data ? How to process/analyze it ?
  - text search ?
    - Very limited
  - “find the stores with the maximum increase in sales in last month”
    - We can’t expect the users to write Java programs
  - “how much time from here to Pittsburgh if I start at 2pm ?”
    - Data is there; more will be soon (GPS, live traffic data)
    - Requires predictive capabilities
  - Increasing need to convert “information” to “knowledge”: **Data mining**
    - “How many DVDs should we order?” (Netflix)
    - Find videos with this type of an event (say car break-ins)
    - Mine the “blogs” to detect “buzz”

Motivation: Data Overload

- Speed !!
  - With TB's of data, just finding something (even if you know what), is not easy
    - Reading a file with TB of data can take hours
  - Imagine a bank and millions of ATMs
    - How much time does it take you to do a withdrawal ?
    - The data is not local
- How do we guarantee the data will be there 10 years from now ?
- Privacy and security !!!
  - Every other day we see some database leaked on the web
  - How to make sure different users' data is protected from each other

Why not use file systems ?

- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

Why not use file systems ?

- Drawbacks of using file systems to store data:
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - Example: Transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent access needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time
  - Security problems
    - Hard to provide user access to some, but not all, data

Today

- Motivation: Why study databases ? What is databases ?
- Administrivia
  - Workload etc.
- Current Industry Outlook
- A typical DBMS at a glance
- No laptop use allowed in the class !!

Today

- Motivation: Why study databases ? What is databases ?
  - Key Concept: Data Modeling
  - Key Concept: Data Abstraction
  - Database Design

- Administrivia
  - Workload etc.
- Current Industry Outlook
- A typical DBMS at a glance
- No laptop use allowed in the class !!

DBMSs to the Rescue

- Provide a systematic way to answer many of these questions…
- Aim is to allow easy management of high volumes of data
  - Storing , Updating, Querying, Analyzing ….

- What is a Database ?
  - A large, integrated collection of (mostly *structured*) data
  - Typically models and captures information about a real-world ***enterprise***
    - Entities *(e.g. courses, students)*
    - Relationships *(e.g. **John** is taking **CMSC 424**)*
    - Usually also contains:
      - Knowledge of constraints on the data *(e.g. course capacities)*
      - Business logic *(e.g. pre-requisite rules)*
      - Encoded as part of the data model (preferable) or through external programs

DBMSs to the Rescue

- Massively successful for *highly structured data*
  - Why ? Structure in the data (if any) can be exploited for ease of use and efficiency
    - If there is no structure in the data, hard to do much
    - Contrast managing emails vs managing photos
  - Much of the data we need to deal with is highly structured
  - Some data is *semi-structured*
    - E.g.: Resumes, Webpages, Blogs etc.
  - Some has complicated structure
    - E.g.: Social networks
  - Some has no structure
    - E.g.: Text data, Video/Image data etc.

Structured vs Unstructured Data

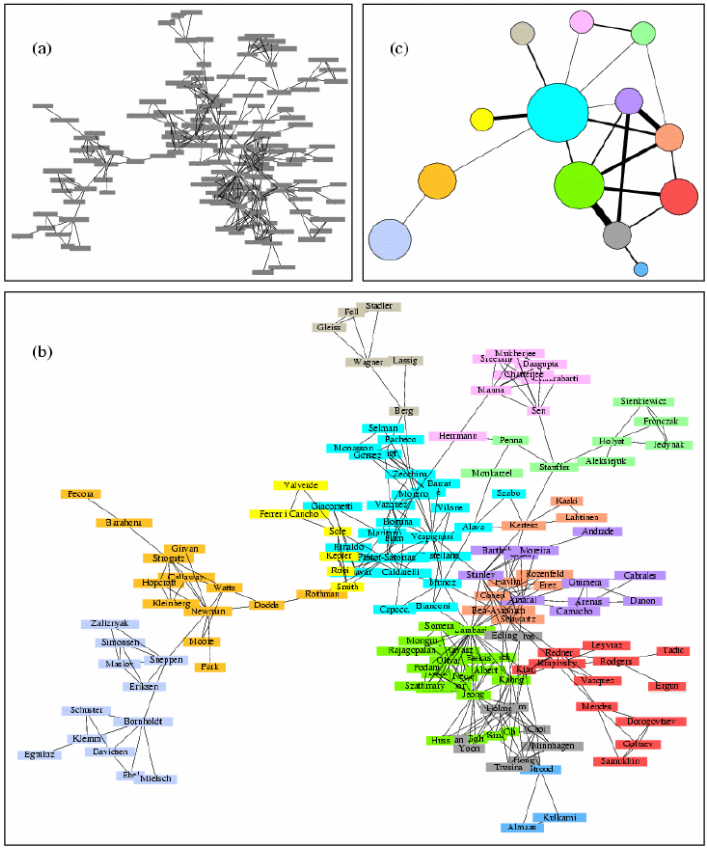
- A lot of the data we encounter is structured
  - Some have very simple structures
    - E.g. Data that can be represented in tabular forms
  - Significantly easier to deal with
  - We will focus on such data for much of the class

Account		
bname	**acct_no**	**balance**
Downtown	A-101	500
Mianus	A-215	700
Perry	A-102	400
R.H	A-305	350

Customer		
cname	**cstreet**	**ccity**
Jones	Main	Harrison
Smith	North	Rye
Hayes	Main	Harrison
Curry	North	Rye
Lindsay	Park	Pittsfield

Structured vs Unstructured Data

- Some data has a little more complicated structure
  - E.g graph structures
    - Map data, social networks data, the web link structure etc
  - Can convert to tabular forms for storage, but may not be optimal
  - Queries often reason about graph structure
    - *Find my "Erdos number"*
    - *Suggest friends based on current friends*
  - Growing importance in recent years in a variety of domains: Biological, social networks, web…



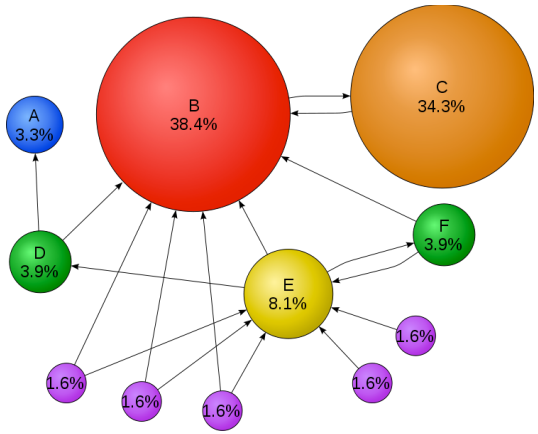
Structured vs Unstructured Data

- Increasing amount of data in a semi-structured format
  - XML – Self-describing tags (HTML ?)
  - Complicates a lot of things
  - We will discuss this toward the end

```
    <Symbol>List</Symbol>
    <Function>
     <Symbol>List</Symbol>
     <Symbol>Automatic</Symbol>
     <Number>4.</Number>
    </Function>
    <Function>
     <Symbol>List</Symbol>
     <Symbol>Automatic</Symbol>
     <Number>6.</Number>
    </Function>
   </Function>
  </Option>
 </Options>
</Notebook>
```

- A huge amount of data is unfortunately unstructured
  - Books, WWW
  - Amenable to pretty much only text search… so far
    - Information Retreival research deals with this topic
  - What about Google search ?
    - Google search is mainly successful because it uses the structure (in its original incarnation)

- Video ? Music ?
  - Can represent in DBMS's, but can't really operate on them



circle size == page importance == **pagerank**
more incoming links → higher pagerank
incoming links from important pages → higher pagerank

DBMSs to the Rescue

- Massively successful for *highly structured data*
  - Why ? Structure in the data (if any) can be exploited for ease of use and efficiency
  - How ?
  - Two Key Concepts:
    - Data Modeling: Allows reasoning about the data at a high level
      - e.g. “emails” have “sender”, “receiver”, “…”
      - Once we can describe the data, we can start “querying” it
    - Data Abstraction/Independence:
      - Layer the system so that the users/applications are insulated from the low-level details

DBMSs to the Rescue: Data Modeling

- Data modeling
  - **Data model**: A collection of concepts that describes how data is represented and accessed
  - **Schema:** A description of a specific collection of data, using a given data model

  - Some examples of data models that we will see
    - Relational, Entity-relationship model, XML. JSON…
    - Object-oriented, object-relational, semantic data model, RDF…

  - Why so many models ?
    - Tension between descriptive power and ease of use/efficiency
    - More powerful models → more data can be represented
    - More powerful models → harder to use, to query, and less efficient

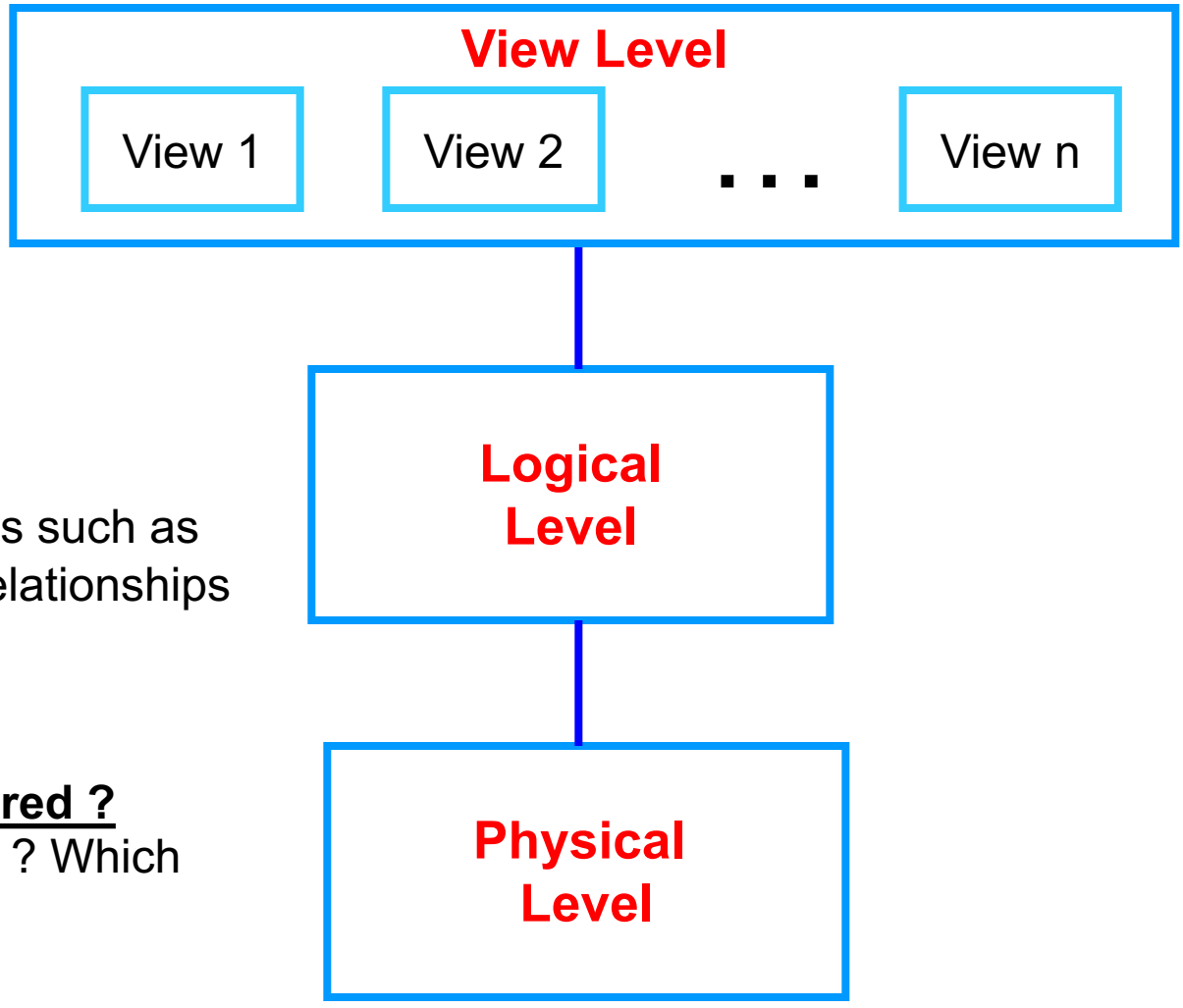
DBMSs to the Rescue: Data Abstraction

- Probably *the* most important purpose of a DBMS
- Goal: Hiding *low-level details* from the users of the system
  - Alternatively: the principle that
    - *applications and users should be insulated from how data is structured and stored*
  - Also called *data independence*

- Through use of *logical abstractions*

Data Abstraction

What data users and application programs see ?



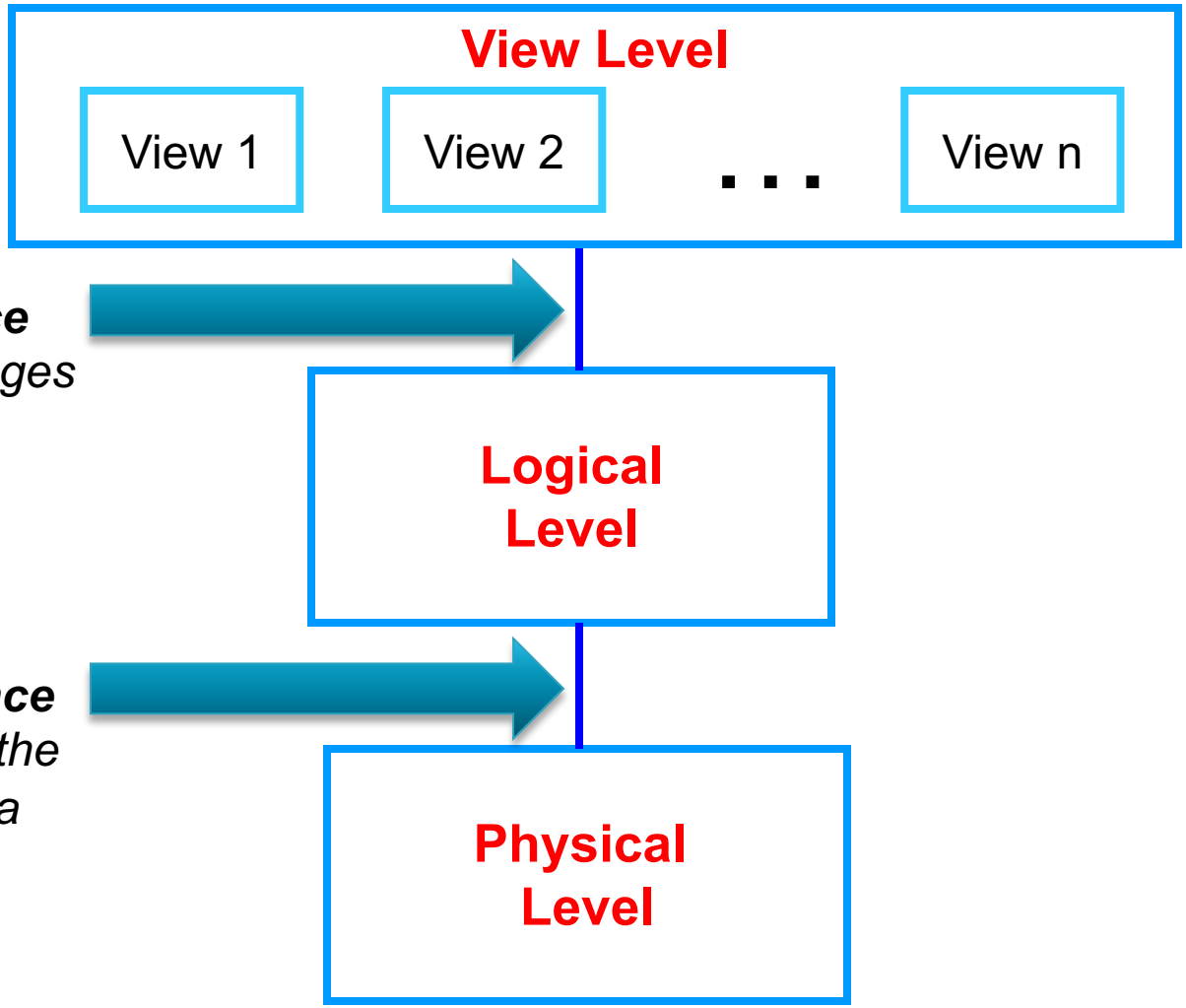
What data is stored ?

describe data properties such as data semantics, data relationships

How data is actually stored ?

e.g. are we using disks ? Which file system ?

Data Abstraction

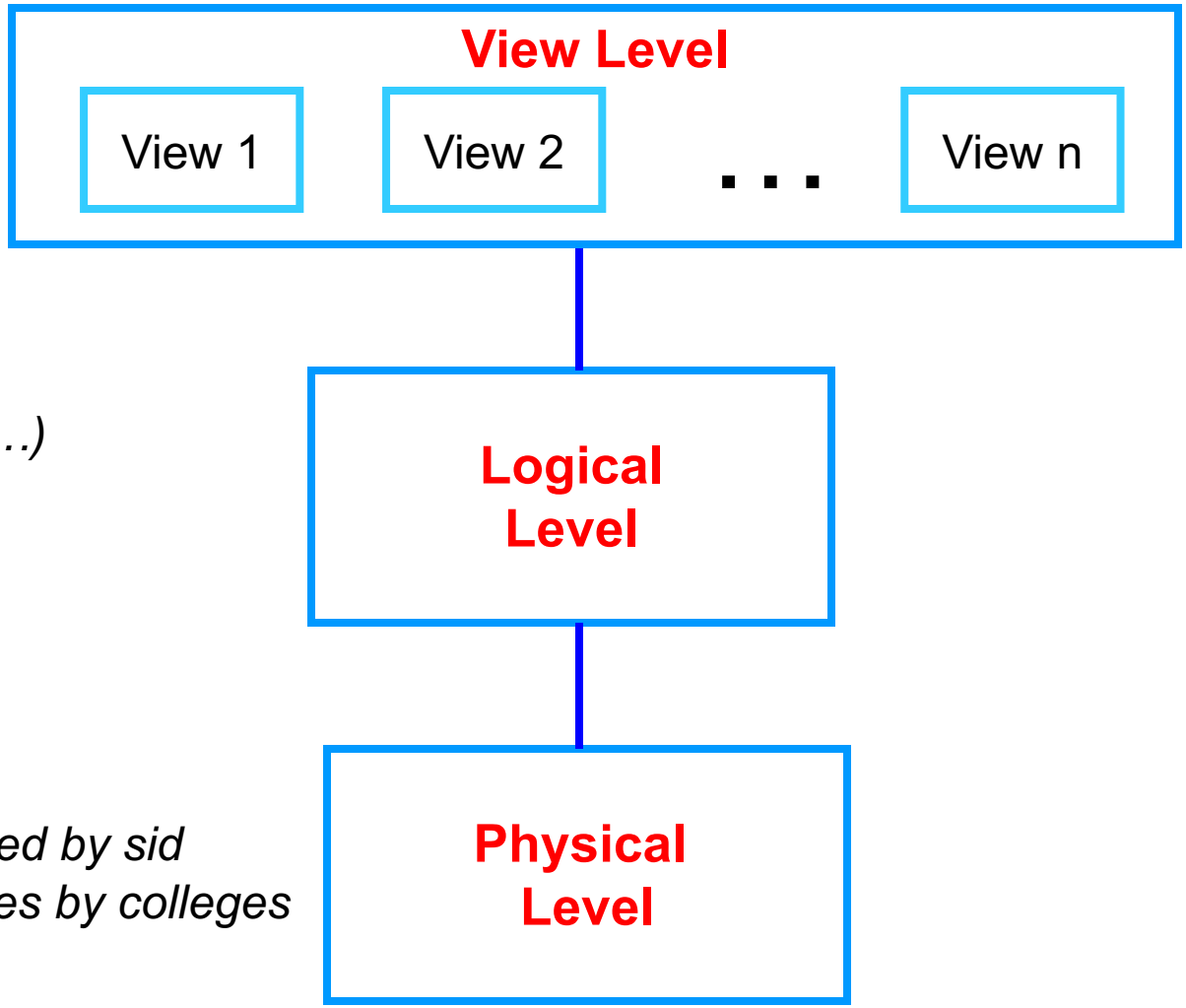


Logical Data Independence
Protection from logical changes to the schema

Physical Data Independence
Protection from changes to the physical structure of the data

Data Abstractions: Example

A View Schema
course_info(#registered,…)



Logical Schema
students(sid, name, major, …)
courses(cid, name, …)
enrolled(sid, cid, …)

Physical Schema
all students in one file ordered by sid
courses split into multiple files by colleges

What about a Database System ?

- A DBMS is a software system designed to store, manage, facilitate access to databases

- Provides:
  - Data Definition Language (DDL)
    - For defining and modifying the schemas
  - Data Manipulation Language (DML)
    - For retrieving, modifying, analyzing the data itself
  - Guarantees about correctness in presence of failures and concurrency, data semantics etc.

- Common use patterns
  - Handling transactions (e.g. ATM Transactions, flight reservations)
  - Archival (storing historical data)
  - Analytics (e.g. identifying trends, **Data Mining**)

Relational DBMS: SQL

- **SQL** (sequel): Structured Query Language

- **Data definition (DDL)**
  - **create table** *instructor* (
    ID **char**(5),
    name **varchar**(20)**,**
    dept_name **varchar**(20),
    salary **numeric**(8,2))

- **Data manipulation (DML)**
  - Example: Find the name of the instructor with ID 22222

    select *name*
    from *instructor*
    where *instructor.ID* = '22222'

Current Industry Outlook

- Relational DBMSs
  - Oracle, IBM DB2, Microsoft SQL Server, Sybase, Amazon RDS/Aurora
- Open source alternatives
  - MySQL, PostgreSQL, BerkeleyDB (mainly a storage engine – no SQL) …
- Other Data Models
  - Neo4j (Graph), MongoDB (Document), CosmosDB (many)
- Data Warehousing Solutions
  - Geared towards very large volumes of data and on analyzing them
  - Long list: Teradata, Oracle Exadata, Netezza (based on FPGAs), Aster Data (founded 2005), Vertica (column-based), Kickfire, Xtremedata..
  - Usually sell package/services and charge per TB of managed data
  - Many (especially recent ones) start with MySQL or PostgreSQL and make them parallel/faster etc..

Web Scale Data Management, Analysis

- Ongoing debate/issue
  - Cloud computing seems to eschew DBMSs in favor of homegrown solutions
  - E.g. Google, Facebook, Amazon etc…
- MapReduce: A paradigm for large-scale data analysis
  - Hadoop: An open source implementation
  - Apache Spark: a better open source implementation
- Why ?
  - DBMSs can't scale to the needs, not fault-tolerant enough
    - These apps don't need things like transactions, that complicate DBMSs (???)
  - Mapreduce favors Unix-style programming, doesn't require SQL
    - Try writing SVMs or decision trees in SQL
  - Cost
    - Companies like Teradata may charge $100,000 per TB of data managed

Current Industry Outlook

- Bigtable-like
  - Called “key-value stores”
  - Think highly distributed hash tables
  - Allow some transactional capabilities – still evolving area
  - PNUTS (Yahoo), Apache Cassandra (Facebook), Dynamo (Amazon), and many many others
- Mapreduce-like
  - Hadoop (open source), Pig (@Yahoo), Dryad (@Microsoft), Spark
  - Amazon EC2 Framework
  - Not really a database – but increasing declarative SQL-like capabilities are being added (e.g. HIVE at Facebook)
- Much ongoing research in industry and academia

DBMS at a glance

- Data Models
  - Conceptual representation of the data
- Data Retrieval
  - How to ask questions of the database
  - How to answer those questions
- Data Storage
  - How/where to store data, how to access it
- Data Integrity
  - Manage crashes, concurrency
  - Manage semantic inconsistencies

- Not fully disjoint categorization !!

What we will cover…

- We will mainly discuss structured data
  - That can be represented in tabular forms (called Relational data)
  - We will spend some time on XML
  - We will also spend some time on Mapreduce-like stuff
- Still the biggest and most important business (?)
  - Well defined problem with really good solutions that work
    - Contrast XQuery for XML vs SQL for relational
  - Solid technological foundations
- Many of the basic techniques however are directly applicable
  - E.g. reliable data storage etc.
  - Cf. Many recent attempts to add SQL-like capabilities, transactions to Mapreduce and related technologies
    - E.g., Spark DataFrames

What we will cover…

- representing information
  - data modeling
  - semantic constraints
- languages and systems for querying data
  - complex queries & query semantics
  - over massive data sets
- concurrency control for data manipulation
  - ensuring transactional semantics
- reliable data storage
  - maintain data semantics even if you pull the plug
  - fault tolerance

What we will cover…

- representing information
  - data modeling: *relational models, E/R models, XML/JSON*
  - semantic constraints: *integrity constraints, triggers*
- languages and systems for querying data
  - complex queries & query semantics*: SQL*
  - over massive data sets*: indexes, query processing, optimization, parallelization/cluster processing, streaming*
- concurrency control for data manipulation
  - ensuring transactional semantics: *ACID properties, distributed consistency*
- reliable data storage
  - maintain data semantics even if you pull the plug: *durability*
  - fault tolerance: *RAID*

Summary

- Why study databases ?
  - Shift from *computation* to *information*
    - Always true in *corporate* domains
    - Increasing true for *personal* and *scientific* domains
  - Need has exploded in recent years
    - Data is growing at a very fast rate
  - Solving the data management problems is going to be a key
- Database Management Systems provide
  - Data abstraction: Key in evolving systems
  - Guarantees about data integrity
    - In presence of concurrent access, failures…
  - Speed !!

Logistics

- Instructor: Amol Deshpande
  - 5154 IRB
  - amol@cs.umd.edu
  - Class Webpage:
    - Off of http://www.cs.umd.edu/~amol,
    - Or http://www.cs.umd.edu/class

- Email to me: write CMSC424 in the title.

- TAs: Shruti Bidwalkar, Richard Johnson, Abhilasha Sancheti, Wichayaporn Wongkamjan

Logistics

- Textbook:
  - Database System Concepts
    - Sixth Edition
    - Abraham Silberschatz, Henry F. Korth, S. Sudarshan
- Lecture notes will be posted on the webpage
- CampusWire
  - We will use this in place of a newsgroup
  - First resort for any questions
  - General announcements will be posted there
  - Register today !

Administrivia Break

- Workload:
  - 6 (individual) programming projects (30%)
    - 10 late days in total, no more than 4 for any project
  - 2 midterms (30%), Final (25%)
  - Reading homeworks (11%)
    - One every week (can get full credit with 11/13)
    - Assigned reading, simple questions on the reading (to ensure you read it) and homework on the previous week's material
      - Readings will refer to the Sixth edition of the book
    - Expect to spend about 1.5-2 hours on each
      - With the exception of the ones for the cancelled classes
  - Class/Forum participation (4%)
    - May do in-class activities later

Logistics

- Project 1: SQL (out by tomorrow)
  - May want to get started on this soon since it covers the same stuff as the first reading homework

Reading Homeworks Due			Midterms/Final		Projects Due		
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
January	27	28	29	30	31	1	2
February	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	1
March	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	31	1	2	3	4	5
April	6	7	8	9	10	11	12
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	1	2	3
May	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18						

Logistics

- Grading
  - Approximate cut-offs
  - 80+: A
  - 70+: B
  - 60+: C
  - 60-: D/F
- Most had 35+ on non-exams last two times (out of 45)
  - Exams are usually somewhat harder (no curves)
  - We would enforce a minimum passing grade on the total exam score

Some To-Dos

- Sign up for CampusWire !
- Set up the computing environment (project0), and make sure you can run Vagrant+VirtualBox, PostgreSQL, Jupyter, etc.
- Upcoming: Reading Homework 1 (Due next Monday), Project 1: SQL