CSCB20: Lecture 1

January 7, 2019 8:20 AM

what is a Database?

- It is a collection of interrelated data.
- The data is relevant to an enterprise (FB, Google, Amazon)
  - eg amazon shows you items related to you.

what is a database management system (DBMS)?

- A database and a set of programs to access database.
- Provides ways to store and retrieve database information.
- Must be convenient and efficient.
- Two things database should do efficiently: write, read

Where are databases used?

- Enterprise Information
  - Sales
  - Accounting
  - HR - Human Resource
  - Manufacturing
  - Online retailers
- Banking and Finance
  - Banking
  - Credit card transaction
  - finance
- Other applications?
  - Universities
  - Airlines
  - Telecommunication
  - more!

Add1	Track1
Add2	Track2
⋮	⋮
AddN	TrackN

when did databases occur?

- 1960's data storage changed from ~~tape~~ to direct access. → has memory address (location)
- This allowed shared interactive data use.
- Early databases were navigational which was very inefficient for searching
- Edgar Codd created a new system in the 1970s based on the relational model.
- Late 1970's and early 1980's SQL was developed based on a relational model, which is the foundation of current databases and what we will study. → relational model is improvement of navigational model ⇒ Table Form.
- In 2000's, with increasingly large datasets, new XML databases and NoSQL databases are becoming more prevalent.

Why use databases?

- Commercialized management of large amounts of data.
- Ability to update and maintain data
- keep track of relationships between subsets of the data
- efficient access and searching capabilities
- multiple users can access and share data.
- Ability to limit access to a certain portion of the data according to user type and enables security of data.
- Minimizes redundancy of multiple data sets
- Enables consistency constraints
- Allows users an abstract view of the data which hides the details of how the data are stored and maintained.

Data Abstraction - how?

- Physical level:
  - lowest level, how the data are actually stored
  - Usually in complex low-level data structures
- Logical Level: Ways to connect data (eg. kindle users are likely to purchase ebooks.
  - what data are stored in the database and what relationships exist between the data.
  - Implementing the simple structure of the logical level may require complex physical low level structures.
- View level - logical table [$a+b+c+d=17$], what is $a+b$? $x=a+b$ then $x+c+d=17$ | Intermediate table that can be developed.
  - Highest level of abstraction - describes only a small portion of the data base.
  - Allows used to simplify their interaction with the database system.
  - Can have many views, why is this good?

eg.

Name	Quiz 1
A	⋮
⋮	⋮
Z	⋮

An intermediate table, give every student with first name starting with a "C".

New table

Name	Quiz 1
C	⋮
⋮	⋮
C	⋮

} views!

why do we want views?

- simplifies interaction with users.

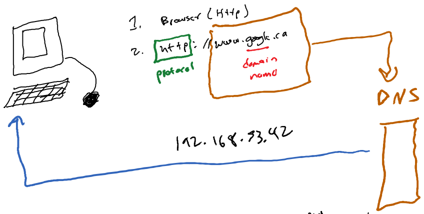
Protocle: allows to communicate with a specific rules.

eg. browser - Http

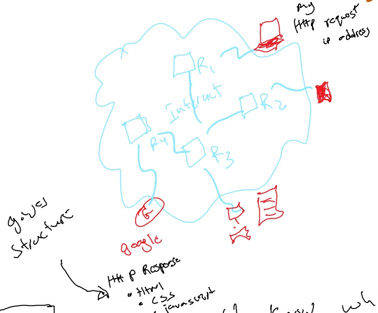
Prof - english

speak in the protocle

Browser understands Http protocol.



= IP address of DNS is provided by your IP service provider eg. rogers, bell ...



IP address: # that identifies a location of hosts / destination

IP addresses are hard to memorize, so we map it to a domain using DNS.

Think of DNS as a machine to translate IP address to domains vice versa

You don't know which path you take to get to your end destination. You just know the destination. Each router takes the best next path. Think of it like a post office. whichever gets you to your destination quicker.

your browser on your computer understands Http. And sends Http request

~~to~~ understands Http. And sends Http response.

On googles side - a web server

Html - skeleton

CSS - look and feel

Just Html and CSS provides a picture

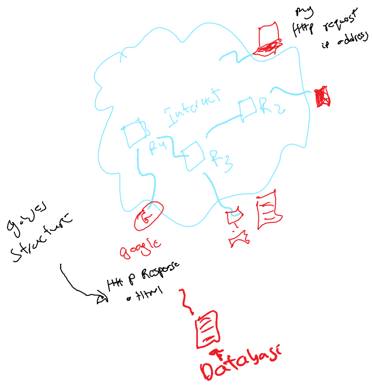
Javascript - makes it interactive

To ban an IP Address: block the end point IP Address

- Use VPN to by pass
- Eg to watch US netflix, use VPN to use a US router.

Where does databases fit in the web?

- Username and passwords

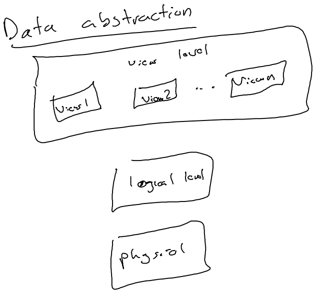


Our journey today starts from databases.

why do we want

- simplifics interaction with users.

Data abstraction



Relational model

Table = relation
row = tuples
column = attributes

Database is a collection of tables each having a unique name

Instance of a database is the info stored at a particular moment in time.

Schema: the overall design of the database

instances changes quick, schema usually doesn't

domain? limits of column such as credits and salary. Actual range for which the values are being pulled off to create a table.

Database

Terminology

Database Schema: The logical design of the database

Database instance: snapshot of the data in the database

Relation Schema:

How do we uniquely refer to a tupple or row in a schema?

Super key: set of attributes that allow you to differentiate one row from another row.

What are possible super keys for the instructor relation?

instructor (ID, name, dept_name, salary) ← still a super key cause id is present and unique.

Teaches relation

ID is not a super key.

Good super key: ID and course ID? no cause multiple selection possible

Better to take everything.

Candidate key: chooses a minimal super key. Not long of attribute!

Eg: Super key for relation instructor

$\underbrace{\{ID\}}_{SK1}, \underbrace{\{name, dept\_name\}}_{SK2}, \underbrace{\{ID, name\}}_{SK3}$

which is a SK?

A. {ID}, {Name, dept_name}

contains enough attributes to uniquely identify