

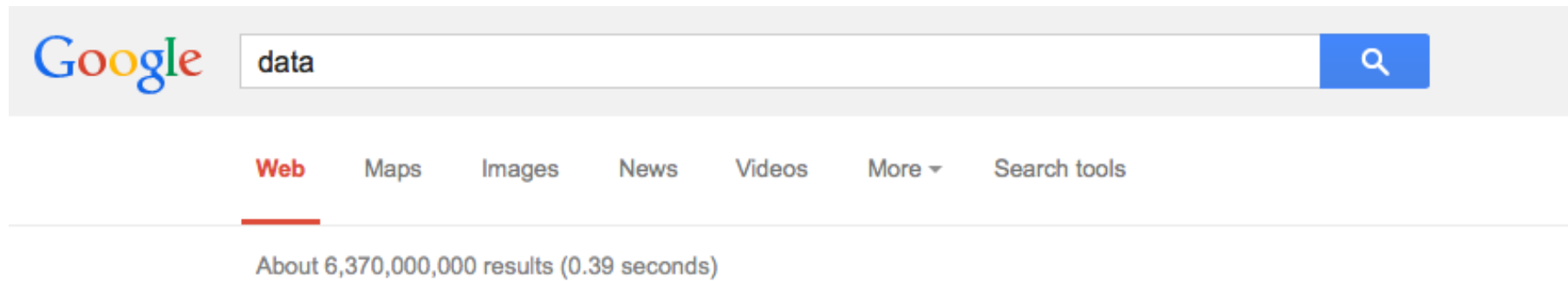


# Managing Data & Databases

Session 2

Saving, retrieving and exchanging data

# The data



If it is so  
abundant,  
why is it so  
valuable?

## What does that mean?

- Pre-analytical, pre-factual...
- Different from facts, evidence...
- Different from information, knowledge...
- Not ontological, not epistemological, but rhetorical...
- What the... :D



And what defines a data type?

$2 + 2 = ?$

$"2" + "2" = ?$

# What's the deal with data structures?

Structured = Fixed structure

Semi-structured = Floating structure

Unstructured = No readily identifiable/described structure

But why should we structure the data?

# What is qualitative anyway?

Storage?  
Structure?  
Interpretation?



# The statisticians' conception of data types

- Nominal
  - Notion of categories
  - Married > Unmarried? Unmarried > Married? Married < Divorced?
  - If the comparison is subjective, you are not dealing with data anymore
- Ordinal
  - Notion of order
  - Medium > Small, Large > Medium => Large > Small
  - But is  $L - M = M - S$ ? Or at least  $L - M = f(M - S)$
  - If the answer is subjective, you are not dealing with data anymore
- Interval
  - Notion of difference / subtraction
  - Discrete = Count, Continuous = Almost all measures
  - Intervals can have a fixed point = Ratio
  - Intervals can have be non-linear = Decibels, Richter



# The computer scientists' perception of data types



- Primitive
  - Numerical: Digits, Integers, Real numbers, Hexadecimal numbers, Large integers, Rational numbers, Double precision floating point real numbers!
  - Logical: Booleans, Factors
  - Characters, Strings (of characters), Text
    - Code is machine-interpretable / executable text
  - Pointers, References, Links
- Composite
  - Any larger data structure encompassing a combination of the above
- Binary
  - Machine-readable anything
  - Where we can't deal with the raw data
  - Compiled code, encrypted text, images, audio, video, multimedia



# Some simple composite data types

- 1D
  - Tuples (Pairs)
  - Ranges
  - Linked Lists (or Lists)
  - Vectors
  - Rows
  - Columns
  - Collections
- 2D or more
  - Arrays (Vectors of vectors)
  - Tables (Collection of columns and rows)



## Some more complex data structures

- Linked tables
- Trees (and Hierarchies)
- Graphs (and Networks)
  - Multi-graphs
  - Multi-mode Graphs
  - Hyper-graphs
- Documents
- And most importantly: **Objects**
  - Collections of diverse variables / attributes / properties and coded behaviors

# How do we save it?

- Serialization
  - That's what you do when you write down your fluffy ideas
- Storage formats
  - Human-readable (Regularly used for data-interchange)
    - Linear
      - Plain text
    - Two-dimensional
      - Comma-delimited or tab-delimited text
      - Fixed-width text
    - Tree
      - Markup languages (HTML, XML, CSS)
      - Object notations (YAML, JSON)
  - Machine-readable (Regularly used for permanent data storage)
    - Most databases (apart from some new NoSQLs)
    - Most videos, most images, most audio formats, etc...

## Comma-delimited text

"Date","Pupil","Grade"

"25 May","Bloggs, Fred","C"

"25 May","Doe, Jane","B"

"15 July","Bloggs, Fred","A"

"15 April","Muniz, Alvin ""Hank""","A"

# XML



```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook V3.1//EN">
<glossary><title>example glossary</title>
  <GlossDiv><title>S</title>
    <GlossList>
      <GlossEntry ID="SGML" SortAs="SGML">
        <GlossTerm>Standard Generalized Markup Language</GlossTerm>
        <Acronym>SGML</Acronym>
        <Abbrev>ISO 8879:1986</Abbrev>
        <GlossDef>
          <para>A meta-markup language, used to create markup
languages such as DocBook.</para>
          <GlossSeeAlso OtherTerm="GML">
            <GlossSeeAlso OtherTerm="XML">
              </GlossDef>
            <GlossSee OtherTerm="markup">
              </GlossEntry>
            </GlossList>
          </GlossDiv>
        </glossary>
```

# JSON



```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to create markup languages such as DocBook.",
            "GlossSeeAlso": ["GML", "XML"]
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

Now, does all this precision mean  
that honestly gathered, non-  
fabricated is objective?

Why not?



# MySQL Data Types

Data Type	Storage Required
TINYINT	1 byte
SMALLINT	2 bytes
MEDIUMINT	3 bytes
INT, INTEGER	4 bytes
BIGINT	8 bytes
FLOAT(p)	4 bytes if $0 \leq p \leq 24$ , 8 bytes if $25 \leq p \leq 53$
FLOAT	4 bytes
DOUBLE [PRECISION], REAL	8 bytes
DECIMAL(M,D), NUMERIC(M,D)	Varies; see following discussion
BIT(M)	approximately $(M+7)/8$ bytes



# MySQL Data Types

Data Type	Storage Required
YEAR	1 byte
DATE	3 bytes
TIME	3 bytes + fractional seconds storage
DATETIME	5 bytes + fractional seconds storage
TIMESTAMP	4 bytes + fractional seconds storage
ENUM('value1','value2',...)	1 or 2 bytes, depending on the number of enumeration values (65,535 values maximum)
SET('value1','value2',...)	1, 2, 3, 4, or 8 bytes, depending on the number of set members (64 members maximum)

# MySQL Data Types

Data Type	Storage Required
CHAR(M)	M × w bytes, 0 ≤ M ≤ 255, where w is the number of bytes required for the maximum-length character in the character set. See Section 14.2.15.7, “Physical Row Structure” for information about CHAR data type storage requirements for InnoDB tables.
BINARY(M)	M bytes, 0 ≤ M ≤ 255
VARCHAR(M), VARBINARY(M)	L + 1 bytes if column values require 0 – 255 bytes, L + 2 bytes if values may require more than 255 bytes
TINYBLOB, TINYTEXT	L + 1 bytes, where L < 28
BLOB, TEXT	L + 2 bytes, where L < 216
MEDIUMBLOB, MEDIUMTEXT	L + 3 bytes, where L < 224
LOB, LONGTEXT	L + 4 bytes, where L < 232

# What is all the fuss about metadata?

Is metadata data?  
What kind of data?

