MLDS-413 Introduction to Databases and Information Retrieval

Lecture 3
Text, Date and Time Representations

Instructor: Nikos Hardavellas

Slides adapted from Steve Tarzia

Last Lecture

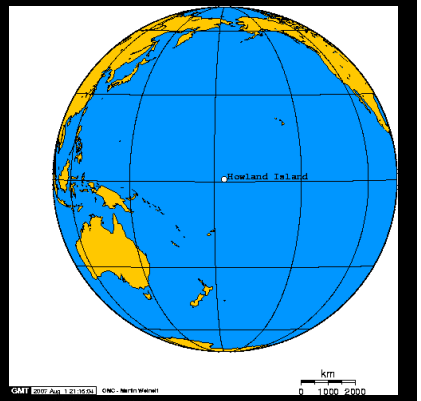
- Described floating point: a binary representation of scientific notation
- Floating point can represent
  - Fractional numbers, even very small ones (very close to 0, + or -)
  - Wide range (can represent very large and very small values)
- But floating point numbers have fewer significant digits than integers
  - Single precision: ~7 decimal significant digits
    - Can perfectly represent integers requiring up to 24 bits
  - Double precision: ~16 decimal significant digits
    - Can perfectly represent integers requiring up to 53 bits
- Counting: use integers
  - Use unsigned integers only if you really need the additional range
- Measuring: use floats
- Use fixed-point on special cases

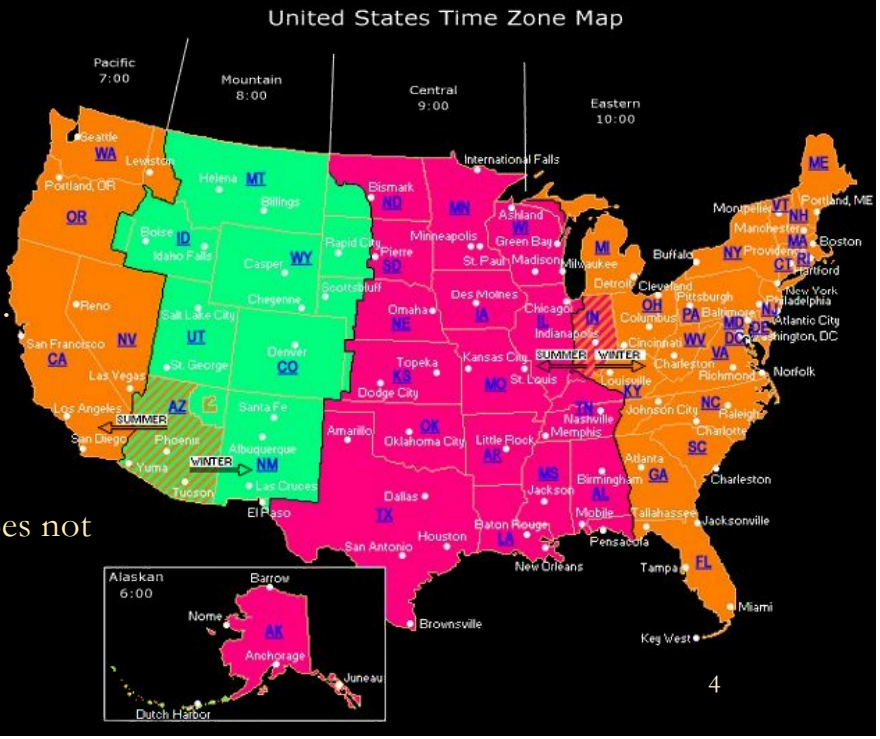
Date and Time

- Seems simple, but date and time can cause all sorts of problems:
- Calendars are complex!
  - Feb 29th on leap years
  - Sometimes a *leap second* gives 61 seconds in the last minute of the year
  - Wikipedia: "Every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but these centurial years are leap years if they are exactly divisible by 400. For example, the years 1700, 1800, and 1900 are not leap years, but 2000 is."
  - Equinox alignment: Oct 4, 1582 → Oct 15, 1582

Time Zones are Complex!

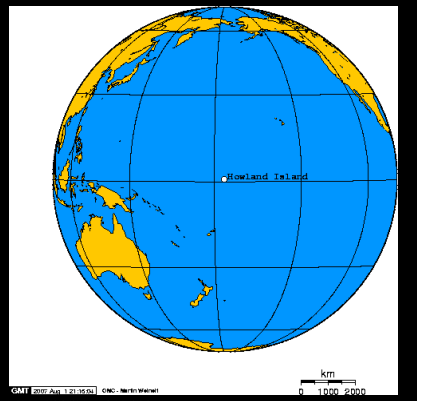
- Irregularly drawn time zones; differing state/nation time standards
  - Some states have many time zones
    - Florida, Indiana, Kentucky, Michigan and Tennessee (Eastern + Central)
    - Nebraska, Kansas, Texas, North / South Dakota (Central + Mountain)
    - Idaho, Nevada, Oregon (Mountain + Pacific)
    - Alaska (Alaska + Hawaii-Aleutian)
  - Some have none: Baker, Howland, Wake Islands
  - Some locations use daylight savings, others do not
    - Non-DST time zones: American-Samoa, Chamorro
    - Some states don't observe DST: Hawaii, Navassa Island, ...
    - Indiana: not following DST since 1970, but it does 2006+
    - Not Arizona (Energy Policy Act of 2005)
      - ...but, Navajo Nation (in Arizona) follows DST
      - ...but, Hopi Reservation (surrounded by Navajo Nation) does not
  - Date of daylight savings varies
    - March 10, 2019 (US) vs. March 31, 2019 (Europe)
  - AoE

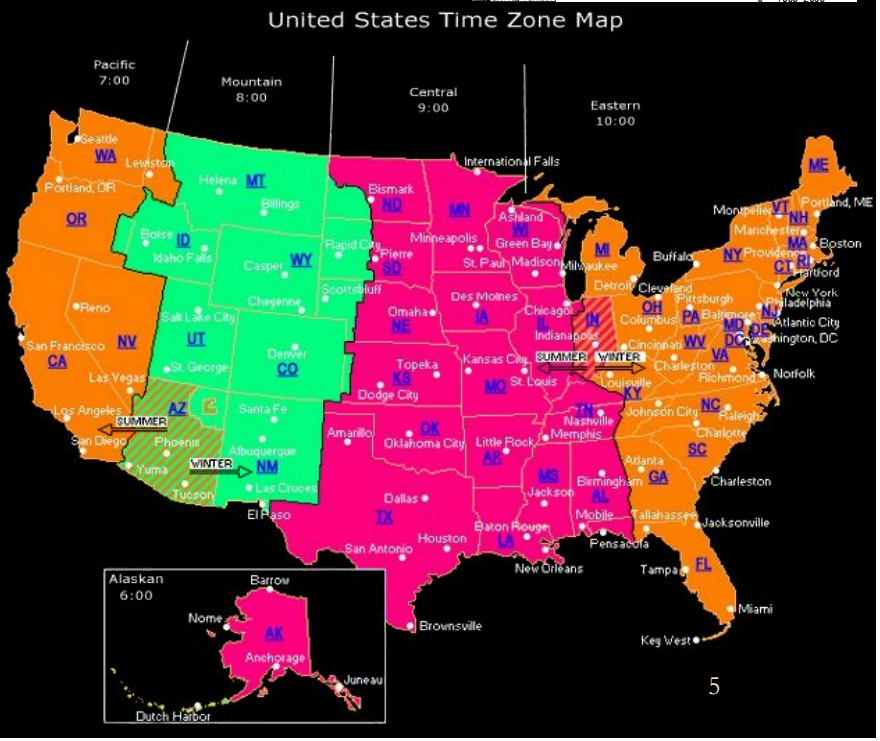




Time Zones are Complex!

- Case Study: Indiana
  - 1918 CT
  - 1961 divided to CT/ET
  - 1967-77 12 counties observed DST/CT, but 82 non/ET
  - …but 6 of them unofficially DST
  - 1977 Pike moved from CT to ET no DST
  - 1991 Starke followed
  - 2006 8 counties returned to CT/DST
  - …but on March 2007 one of them returned to ET
  - …and Nov 2007 5 more did

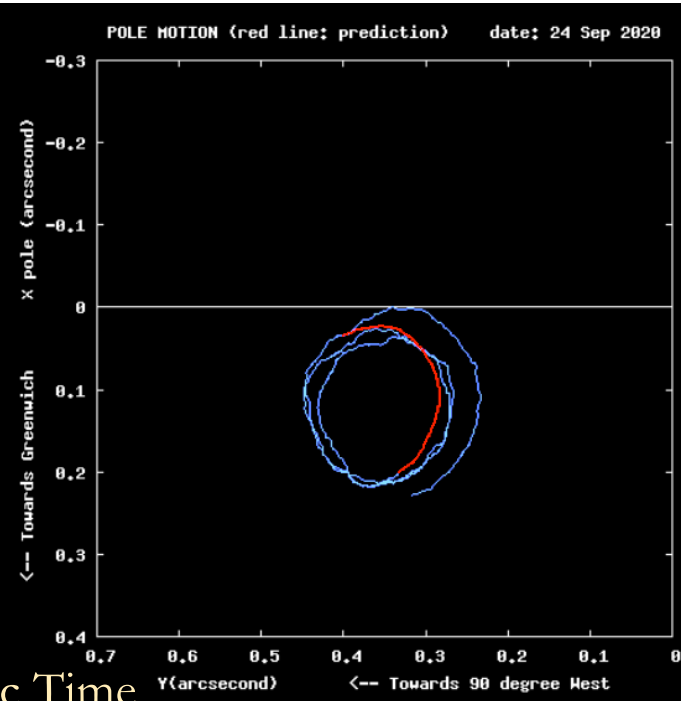


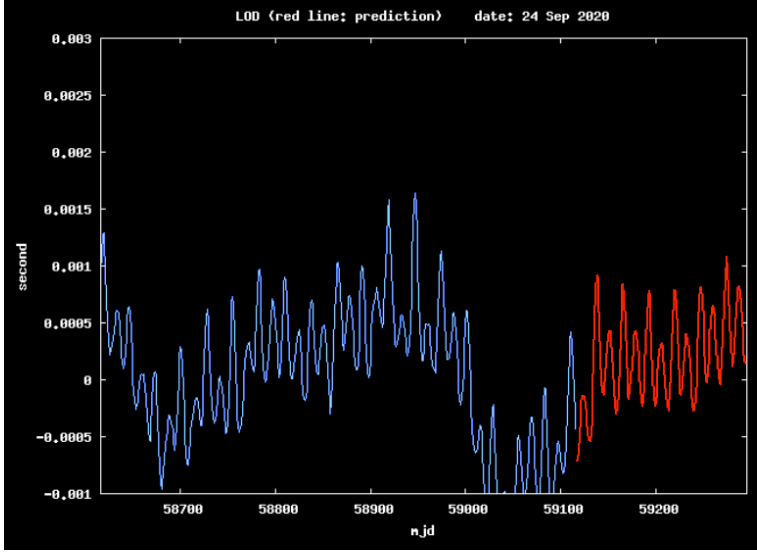


Got that?

Physics is Complex!

- Consider GPS
  - Special relativity: -7μs/day dilation (orbital speeds)
  - General relativity: +45.9μs/day (gravitational freq. shift)
  - Combined shift: +38 μs/day
  - Accuracy requirement: 20ns w.r.t. TAI – 19s





- Time standards evolve
  - UTC = Universal Time
  - TAI = International Atomic Time
  - TAI – UTC = ? periodically announced "leap seconds"
    - International Earth Rotation and Reference Service
    - Full table at Earth Orientation Center
      - See leap second file at http://hpiers.obspm.fr/eop-pc/index.php
  - Ephemeris (1952), Barycentric Dynamical (1984/2006), Terrestrial (1991/2000), Geocentric Coordinate (1991), Barycentric Coordinate (1991), Sidereal (1970s/2003)

Death by Dates

- "12:45pm on Sept 26, 2017" is actually difficult to interpret and compare to times that may have been observed in different locations
- A purchase was made in Japan at 13:01 local time on January 3rd, 1991
  - How long ago was that in seconds if I am in **Pike County, IN**?
- Don't ever, ever write your own calendar code
  - Your database management system or a standard library has already done it

Epoch time

- For simplicity, times are most often represented in "epoch time"
  - Defined as the number of seconds since January 1st, 1970 in London, England.
  - Simply use a 32-bit unsigned integer to count seconds since 1970 or a 64-bit unsigned integer for milliseconds or microseconds, if desired.
  - Beginning of class, 9/28/2023 @ 9:30AM, epoch time = 1,695,911,400
  - Epoch time does not account for leap seconds, so durations are not truly precise.
- Calendar libraries convert epoch time to a human-readable format in the time-zone of interest.
- Ignores Einstein's theory of relativity, but that's OK
- The UTC time standard is used in systems that wish to report *human-readable* times in a time-zone-independent way.
  - Uses mean solar time at the Greenwich Meridian (London), without daylight savings
  - For example, the event logs in a multinational ecommerce platform

More about bits

- When measuring data, 8 bits are called a **byte**.
- Bytes are the standard unit of data measurement.
- However, kilobytes, megabytes, gigabytes, and terabytes actually grow by factors of 1024 (2^{10}), not 1000 (10^3).
  - 1 GB is actually 2^{30} bytes = 1024^3 bytes = 1,073,741,824 bytes
  - Sometimes, GiB is used (instead of GB) to denote 2^{30} (instead of 10^9)
    - Similarly, KiB, MiB
- Hexadecimal notation refers to groups of four bits with the characters: 0 1 2 3 4 5 6 7 8 9 A B C D E F
  - So, 00111100 in hex notation is "3C" sometimes written as 0x3C
  - 0xFFFF is the hex notation for 16 ones (in binary)
  - Hex is a much shorter way for humans to read and write bit values

Text encodings

- How do computers store text as ones and zeros?
- Early standard is called the American Standard Code for Information Interchange (ASCII)
  - Developed in the 1960s
  - Uses seven bits per character,
    but in practice each character is stored in 8 bits and the top bit is zero.
- ASCII text includes:
  - Lowercase letters, uppercase letters, numbers, punctuation, other symbols
  - Whitespace characters: space, tab, newline, carriage return
  - Control characters: null, line feed, vertical tab, bell, escape, delete, backspace, etc.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

“Hello!” in ASCII

	H	e	l	l	o	!
hex	48	65	6C	6C	6F	21
binary	0100 1000	0110 0101	0110 1100	0110 1100	0110 1111	0010 0001

A Tale of Two Cities

Book the First--Recalled to Life

I. The Period

It was the best of times,
it was the worst of times,
it was the age of wisdom,
it was the age of foolishness,
it was the epoch of belief,
it was the epoch of incredulity,

Encoded in ASCII:

```
42 6f 6f 6b 20 74 68 65  20 46 69 72 73 74 2d 2d  |Book the First--|
52 65 63 61 6c 6c 65 64  20 74 6f 20 4c 69 66 65  |Recalled to Life|
0d 0a 0d 0a 0d 0a 0d 0a  0d 0a 49 2e 20 54 68 65  |..........I. The|
20 50 65 72 69 6f 64 0d  0a 0d 0a 0d 0a 49 74 20  | Period......It |
77 61 73 20 74 68 65 20  62 65 73 74 20 6f 66 20  |was the best of |
74 69 6d 65 73 2c 0d 0a  69 74 20 77 61 73 20 74  |times,..it was t|
68 65 20 77 6f 72 73 74  20 6f 66 20 74 69 6d 65  |he worst of time|
73 2c 0d 0a 69 74 20 77  61 73 20 74 68 65 20 61  |s,..it was the a|
67 65 20 6f 66 20 77 69  73 64 6f 6d 2c 0d 0a 69  |ge of wisdom,..i|
74 20 77 61 73 20 74 68  65 20 61 67 65 20 6f 66  |t was the age of|
20 66 6f 6f 6c 69 73 68  6e 65 73 73 2c 0d 0a 69  | foolishness,..i|
74 20 77 61 73 20 74 68  65 20 65 70 6f 63 68 20  |t was the epoch |
6f 66 20 62 65 6c 69 65  66 2c 0d 0a 69 74 20 77  |of belief,..it w|
61 73 20 74 68 65 20 65  70 6f 63 68 20 6f 66 20  |as the epoch of |
69 6e 63 72 65 64 75 6c  69 74 79 2c 0d 0a 69 74  |incredulity,..it|
```

What about the thousands of other characters we might want to use?

- ¿Español?, 中文, Ελληνικά
- 
- Different currency symbols
- Even American English uses “weird punctuation” sometimes
  - e.g., dash, en dash, em dash
- A single 8-bit byte will not be enough to store all the possible characters

UTF-8 to the rescue!

- UTF-8 is now the most common text encoding.
- The Sep 2021 version includes 144,697 symbols, and more can be added.
  - Can eventually be expanded to more than one million characters
- It's a variable-length encoding
  - Characters are represented with one, two, three, or four bytes.
- Backward-compatible with ASCII
  - ASCII text is also valid UTF-8
  - Previous version of Unicode (such as UTF-16) were not widely adopted due to incompatibility with ASCII.

Variable length character encoding with UTF-8

1st byte	2nd byte	3rd byte	4th byte	# of free bits
`0... ....`				7 (ASCII)
`110. ....`	`10.. ....`			11
`1110 ....`	`10.. ....`	`10.. ....`		16
`1111 0...`	`10.. ....`	`10.. ....`	`10.. ....`	21

- Single-byte characters are identical to ASCII
- First byte tells you how many total bytes to expect
- Every "extra" byte starts with "10"
  - If you start reading in the middle of a character you'll know it.
  - It's very easy to know where each new character starts.

Example: 500 euros "€500" in UTF-8

	€			5	0	0
hex	E2	82	AC	35	30	30
binary	**1110** 0010	**10**00 0010	**10**10 1100	**0**011 0101	**0**011 0000	**0**011 0000

1st byte	2nd byte	3rd byte	4th byte	# of free bits
`0... ....`				7 (ASCII)
`110. ....`	`10.. ....`			11
`1110 ....`	`10.. ....`	`10.. ....`		16
`1111 0...`	`10.. ....`	`10.. ....`	`10.. ....`	21