

Leap in Time

Claudia Huber



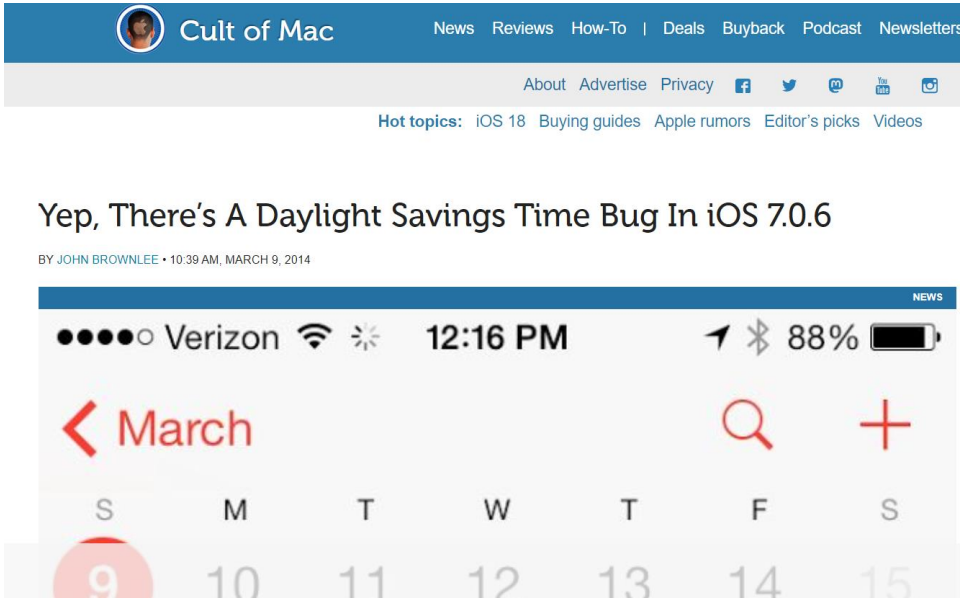
20 SQL KONFERENZ 2024



Time Zone are fun

In Real Life

Bugs ...



http://www.cultofmac.com/269245/yep-theres-daylight-savings-time-bug-ios-7-0-6/

20 SQL KONFERENZ 2024

Adhoc political changes ...

BBC

NEWS

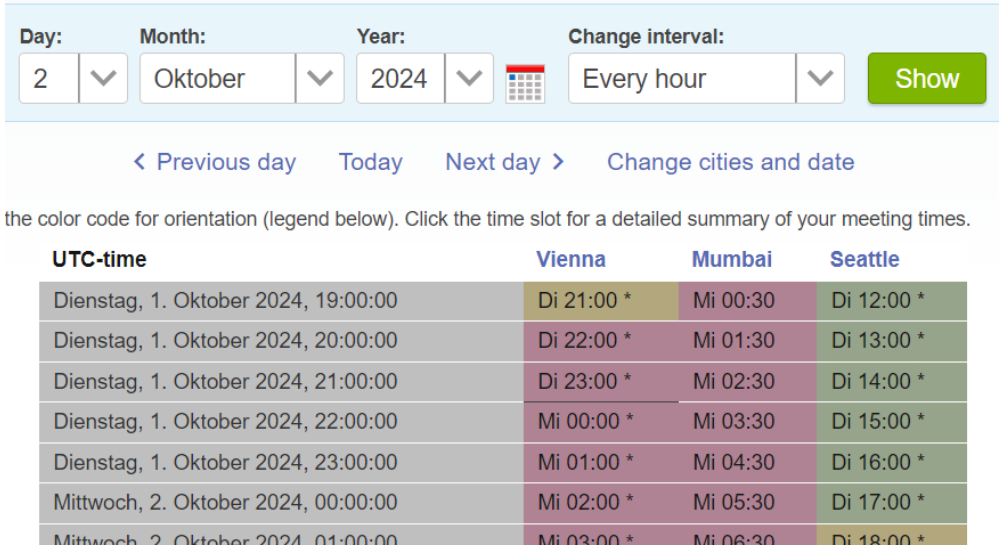
Confusion as Turkey clocks defy time-change delay

25 October 2015

https://www.bbc.com/news/world-europe-34631326



Meeting planning ...



https://www.timeanddate.com/worldclock/meetingtime.html

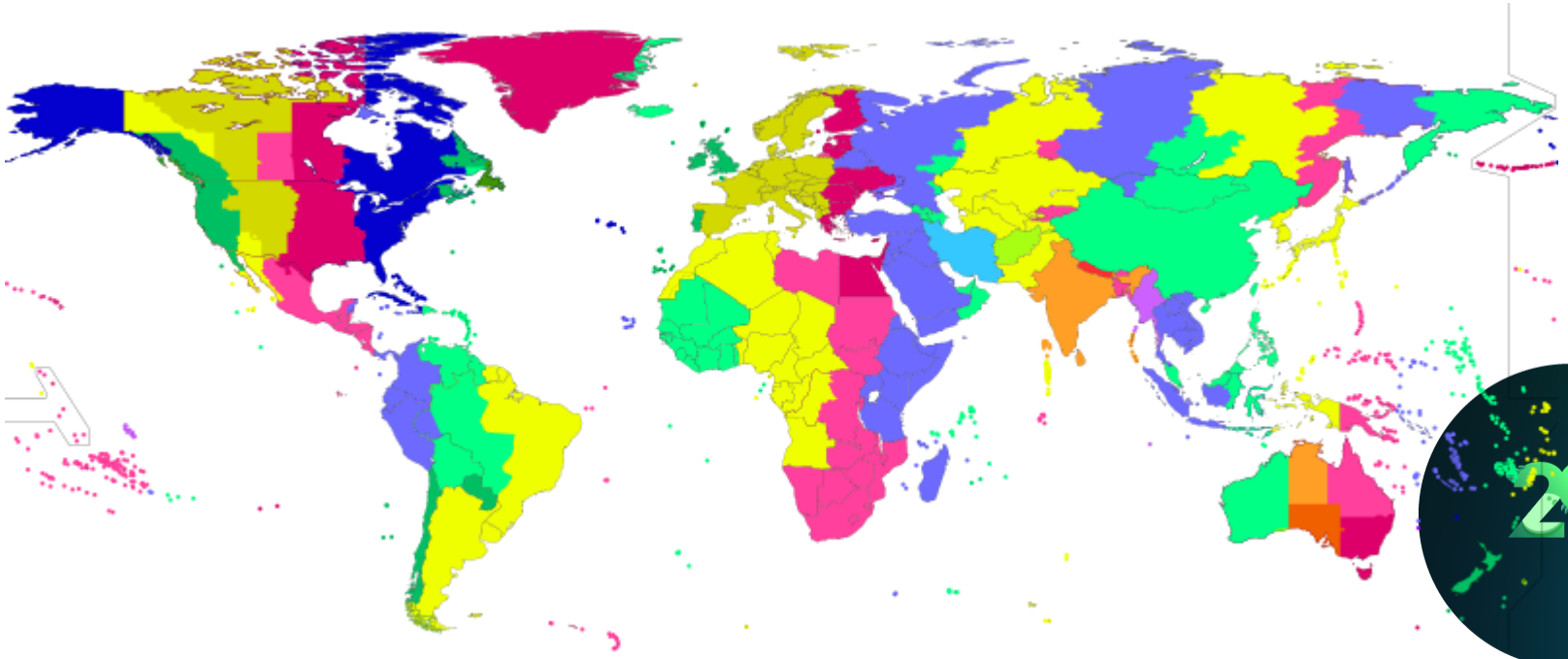
20 SQL KONFERENZ 2024



Time Zones

Definitions and Names ...

Time Zones of the World (UTC offsets)



SQL KONFERENZ 2024

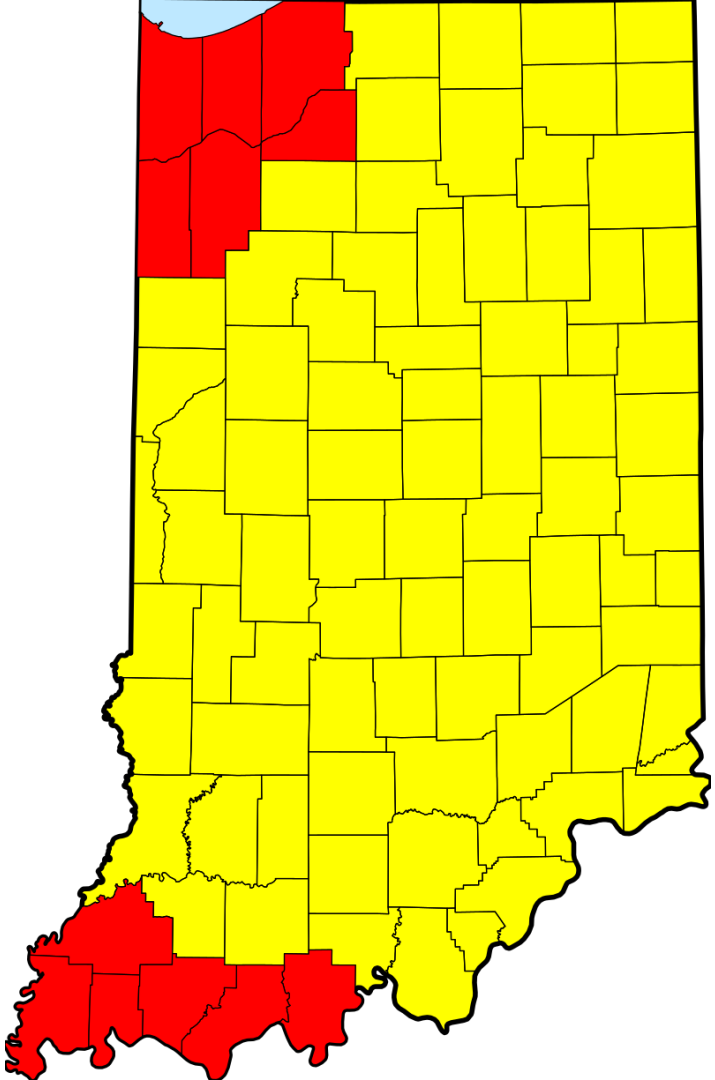
Names of Timezone

- Are not standardized enough in daily usage.
- Timezone information can look like this:
  - GMT +2:00
  - UTC +2:00
  - Europe/Vienna
  - CET (Central European Time) / CEST Central European Summer Time

Complexity of Rules for Time Zones

- Are a moving target due to political changes
- There is not only time zones using UTC +/- x full hours, e.g:
  - Indian Standard Time = UTC +5:30
  - New Zealand, Chatham Islands = UTC +12:45 / +13:45
- Do have Summer- / Wintertime or not
- Begin and End of Daylight Saving time does not happen on the same day in every country. (In Europe there is a standard).

20 SQL KONFERENZ 2024



mezone

ɔ determine a

ı one time zone e.g.

te is enough

ones
ne_in_Indiana

SQL KONFERENZ 2024



Time Zones

Data Arriving ...

Mix of Time Zones



Source Data in different (local) Time Zones

- The source application was build to work for one local time zone.
- There is user generated data with timestamps in local time.
- The source application only gives access to data in the end users default time zone - a local time.
- Legacy data in different time zones
- Data e.g. in Web Analytics is typically recorded in UTC.

Time Zone information in the data

- There might be no information about the time zone in the source data itself. = You need to know the default of the source
- A mix of time zones might exist in a single source with or without proper information.

Daylight Saving Time

Time Zones



Daylight Savings Time

Many countries (but not all) adjust their clock twice a year to Summer / Winter Time.



SQL KONFERENZ 2024

The missing hour 1

- In spring we will advance all clocks by one hour. (Winter to Summer time)
- In Central European Time we switch from 2:00 to 3:00 on the Sunday, 30th March 2025
- This can lead to issues if your code is not ready …

The missing hour 2

- "2025-03-30 01:58" plus 20 minutes
  - = " 2025-03-30 02:19"
    This is an invalid timestamp!
- "2025-03-30 01:58" plus 20 minutes
  - = " 2025-03-30 03:19"
    This is correct ☺

The missing hour 3

- Let’s caluclate how much time lapsed between event A "2025-03-30 01:50" and event B " 2025-03-30 03:05"
  - = 75 minutes
    This is incorrect!
- …
  - = 15 minutes
    This is correct ☺

The missing hour 4

Misleading results can show up because:

- An hourly aggregate of any measure will show a gap on this day.
- For a daily average there is only 23 hours instead of 24 hours of business to happen if you simply use calendar days as one day.

The double hour 1

- In autumn we will set back all clocks by one hour. (Summer to Winter time)
- In Central European Time we switch from 3:00 to 2:00 again on the Sunday, 26th October 2025
- This can lead to issues if your code is not ready …

The double hour 2

- We see the opposite effects to the missing hour

Misleading results can happen because:

- The sort order of single events get's lost. Whenever you want to analyze events in relation to each other.
- Aggregation of the data will show data from two hours and thus can show up as too high numbers.



Solution Patterns

Technology Independent

UTC or UNIX (POSIX) Timestamp



Store Data in UTC 1

- „Coordinated Universal Time (UTC) is the primary time standard globally used to regulate clocks and time. It establishes a reference for the current time, forming the basis for civil time and time zones.“ https://en.wikipedia.org/wiki/Coordinated_Universal_Time
- There is no daylight saving time for UTC! No issues here.
- Observes Leap Seconds (Might not be relevant)
- If your data is already in UTC – you are fine ☺

Store Data in UTC 2

- Calculations will work fine with this data (no double hours etc.)
- Conversion to different time zones (e.g to show to your users) will work fine

Store Data in Unix (Posix) Timstamp?

- „Unix time[a] is a date and time representation widely used in computing. It measures time by the number of non-leap seconds that have elapsed since 00:00:00 UTC on 1 January 1970, the Unix epoch.“ https://en.wikipedia.org/wiki/Unix_time
- Gives an integer number.
- Ignores Leap seconds
- You will usually want to convert to UTC (easy) to make use of date and time functions in your preferred language.

UTC / GMT / Unix?

- GMT (Greenwitch Mean Time) is the older standard and often mixed up with UTC. The difference is seconds.
- For most purposes you are fine with all of those.
- **Recommendation: Use UTC**
  - Newer standard
  - Enables usages of date and time functions in your preferred language (e.g. SQL)

Store Data with UTC offset information 1

- A timestamp in ISO 8601 with time zone information looks like this:
  "2017-02-26 01:00 +0100"
- Either store timestamps
  - In a data type that can store the offset information
  - Or save the offset / time zone in a separate column

SQL KONFERENZ 2024

Store Data with UTC offset information 2

- No information is lost
- Calculations with the data can be done properly – even across time zones ☺
- To show the data to the user you might want to hide the offset
- Conversions to other timezones will work (no double hour etc.)

Time Zone Data with offset information lost 1

You will lose precision during **double hours**. You will need to

- Decide which of the two potentially matching UTC hours you chose.
- or do special calculations like e. g. assigning half the revenue of a transaction to each of the two hours.

20 SQL KONFERENZ 2024

Time Zone Data with offset information lost 2

For the **missing hour** there is no issue when converting to UTC as you still get a unique result. But:

- Error handling to check for the missing hour might be helpful.
  - A real-life example: A single incorrect data point was delivered within the missing hour (human error) - and due to no error handling for this case it went "missing" during the process.
- Business Logic for this day might still need logic to adjust for 23 hours in local time.

20 SQL KONFERENZ 2024

Multiple Timezones - recommendations

- Do not lose the original timezone information or utc timestamp, even you only need one local timezone right now – it might be needed later!
- Unify timezones to UTC or use a data type with offset information to allow logic to work friction free
- If you need to show data in specific timezone(s) to your end users:
  - Pre-calculate in the best case in your gold layer. Doing this live in a report costs performance and is rarely needed.

20 SQL KONFERENZ 2024

Maintaining Time Zone Rules

- First choice: check **functionality offered by your vendor** / language of choice.
- Also checkout **IANA Time Zone Data Base** (TZDB)
  - https://www.iana.org/time-zones
  - TZDB is actively maintained by volunteers, and it is the most complete time zone information I found.
  - De facto standard in the Unix/Linux world and many other systems.
  - Used by many libraries for diverse programming languages
  - Subscribe to their newsletter if time zones are critical for you





Capabilities Microsoft

Data Types TSQL (SQL Server and more) 1

Data type	Format	Range	Accuracy	Storage size (bytes)	User-defined fractional second precision	Time zone offset
time	HH:mm:ss[.nnnnnnn]	00:00:00.0000000 through 23:59:59.9999999	100 nanoseconds	3 to 5	Yes	No
date	yyyy-MM-dd	0001-01-01 through 9999-12-31	1 day	3	No	No
smalldatetime	yyyy-MM-dd HH:mm:ss	1900-01-01 through 2079-06-06	1 minute	4	No	No
datetime	yyyy-MM-dd HH:mm:ss[.nnn]	1753-01-01 through 9999-12-31	0.00333 second	8	No	No
datetime2	yyyy-MM-dd HH:mm:ss[.nnnnnnn]	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999	100 nanoseconds	6 to 8	Yes	No
datetimeoffset	yyyy-MM-dd HH:mm:ss[.nnnnnnn] [+|-]HH:mm	0001-01-01 00:00:00.0000000 through 9999-12-31 23:59:59.9999999 (in UTC)	100 nanoseconds	8 to 10	Yes	Yes

- https://learn.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver16#date-and-time-data-types

SQL KONFERENZ 2024

Data Types TSQL (SQL Server and more) 2

- Use **datetime2** over datetime:
  - Higher precision
  - Compatible with ANSI und ISO 8601 (datetime is NOT compliant with those standards)
  - Potentially smaller data type if no milliseconds are needed (6 bit for *datetime2(0)* instead of 8 bit)
- Use **datetimeoffset** for timestamps with offset information



Time Zone conversion in TSQL (SQL Server and more)

- AT TIME Z
  - https:
    sql?vi
  - Uses

```
--Time before DST change has standard time offset (+01:00)
SELECT CONVERT(DATETIME2(0), '2022-03-27T01:01:00', 126)
AT TIME ZONE 'Central European Standard Time';
--Result: 2022-03-27 01:01:00 +01:00

/*
  Adjusted time from the "gap interval" (between 02:00 and 03:00)
  is moved 1 hour ahead and presented with the summer time offset
  (after the DST change)
*/
SELECT CONVERT(DATETIME2(0), '2022-03-27T02:01:00', 126)
AT TIME ZONE 'Central European Standard Time';
--Result: 2022-03-27 03:01:00 +02:00
--Time after 03:00 is presented with the summer time offset (+02:00)
SELECT CONVERT(DATETIME2(0), '2022-03-27T03:01:00', 126)
AT TIME ZONE 'Central European Standard Time';
--Result: 2022-03-27 03:01:00 +02:00
```

20 SQL KONFERENZ 2024

Data Types in Spark (Notebooks in Azure Synpase / Microsoft Fabric)

- Datetime type
  - `DateType`: Represents values comprising values of fields year, month and day, without a time-zone.
  - `TimestampType`: Timestamp with local time zone(TIMESTAMP_LTZ). It represents values comprising values of fields year, month, day, hour, minute, and second, with the session local time-zone. The timestamp value represents an absolute point in time.
  - `TimestampNTZType`: Timestamp without time zone(TIMESTAMP_NTZ). It represents values comprising values of fields year, month, day, hour, minute, and second. All operations are performed without taking any time zone into account.
    - Note: TIMESTAMP in Spark is a user-specified alias associated with one of the TIMESTAMP_LTZ and TIMESTAMP_NTZ variations. Users can set the default timestamp type as `TIMESTAMP_LTZ`(default value) or `TIMESTAMP_NTZ` via the configuration `spark.sql.timestampType`.

https://spark.apache.org/docs/latest/sql-ref-datatypes.html



Time Zone Conversion in Spark (Notebooks in Azure Synapse / Micro

- For PyS
  least –
  - ht
    ys
  - Us

	Runtime 1.1	Runtime 1.2	Runtime 1.3
Apache Spark	3.3.1	3.4.1	3.5.0
Operating System	Ubuntu 18.04	Mariner 2.0	Mariner 2.0
Java	8	11	11
Scala	2.12.15	2.12.17	2.12.17
Python	3.10	3.10	3.11
Delta Lake	2.2.0	2.4.0	3.2
R	4.2.2	4.2.2	4.4.1

```
|2015-04-08 15:00:00|
+-------------------+
```

https://learn.microsoft.com/en-us/fabric/data-engineering/runtime

SQL KONFERENZ 2024

Time Zone Conversion in Spark (Notebooks in Azure Synpase / Microsoft Fabric) 2

- Workaorund with **pytz** library and user defined function for older versions of spark / runtime

```
1   def convtz(inputz, outputtz, inputdt):
2       itz=pytz.timezone(inputz)
3       otz=pytz.timezone(outputtz)
4       return itz.localize(inputdt).astimezone(otz)
```

✓ <1 sec - Command executed in 228 ms by Huber, Claudia on 3:04:02 PM, 10/02/24

```
1   udf_convtz = F.udf(convtz, TimestampType())
```

✓ <1 sec - Command executed in 256 ms by Huber, Claudia on 3:04:36 PM, 10/02/24

SQL KONFERENZ 2024

Power BI Data Types

- **Date/Time** represents both a date and time value. The underlying **Date/Time** value is stored as a **Decimal number** type, so you can convert between the two types. The time portion stores as a fraction to whole multiples of 1/300 seconds (3.33 ms). The data type supports dates between years 1900 and 9999.
- **Date/Time/Timezone** represents a UTC date/time with a timezone offset, and converts into **Date/Time** when loaded into the model. The Power BI model doesn't adjust the timezone based on a user's location or locale. A value of 09:00 loaded into the model in the USA displays as 09:00 wherever the report is opened or viewed.

  - ...And More
  - https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-data-types



Time Zone Conversion Power Query M-Code (Power BI and more)

- **DateTimeZone.SwitchZone**
  - https://learn.microsoft.com/de-de/powerquery-m/datetimezone-switchzone
  - Uses Hours and Minutes to shift time zones
  - You have to take care for daylight saving yourself!

Power Query M Copy

```
DateTimeZone.SwitchZone(#datetimezone(2010, 12, 31, 11, 56, 02, 7, 30), 0, -30)
```

SQL KONFERENZ 2024

Time Zone Conversion in DAX?

- **No specific function**
- Pre-calculating time zones might be a better idea





Demo Time

A small demo with Fabric

Thank you



Claudia Huber

Teamlead Data & Analytics

adesso Austria GmbH