**How to handle dates?**

In Cassandra, dates can be handled using the date or timestamp data types. Here are some guidelines on how to handle dates in Cassandra:

1. Data Model: Determine how you want to store your date data. Cassandra provides two main data types for handling dates:
   1. timestamp: This data type represents a point in time with millisecond precision. It can store both date and time information.
   2. date: This data type represents a date without any time information.
2. Choosing the appropriate data type: Decide whether you need to store only the date or both date and time. If you require only the date, use the date data type. If you need both date and time, use the timestamp data type.
3. Inserting dates: When inserting dates into Cassandra, you can use the appropriate data type and format the date accordingly. For example, if you're using the timestamp data type, you can insert a date as a Unix timestamp (number of milliseconds since January 1, 1970). If you're using the date data type, you can insert a date as a string in the format 'YYYY-MM-DD'.  
   Here's an example of inserting a date using CQL (Cassandra Query Language) with the timestamp data type:

| INSERT INTO my\_table (id, event\_time) VALUES (1, '2023-06-08 10:30:00'); INSERT INTO my\_table (id, event\_date) VALUES (1, '2023-06-08'); |
| --- |

1. Querying dates: When querying date data, you can use CQL's built-in functions to perform operations like filtering by date range, comparing dates, or extracting components from dates. Here are some examples:
   1. Filtering by date range:

| SELECT \* FROM my\_table WHERE event\_date >= '2023-06-01' AND event\_date <= '2023-06-30'; |
| --- |

* 1. Comparing dates:

| SELECT \* FROM my\_table WHERE event\_date > '2023-06-01'; |
| --- |

* 1. Extracting components from dates:

| SELECT dateOf(event\_time) FROM my\_table; |
| --- |

A duration can be added (+) or subtracted (-) from a timestamp or a date to create a new timestamp or date. So for instance:

| SELECT \* FROM myTable WHERE t = '2017-01-01' - 2d; |
| --- |

will select all the records with a value of t which is in the last 2 days of 2016

| SELECT sensor, day, min(value), max(value), avg(value) FROM sensor\_data WHERE sensor = ? AND day = currentdate() - 1d; |
| --- |

You can express durations as (quantity unit)+ like 12h30m where the unit can be:

* y: years (12` months)
* mo: months (1 month)
* w: weeks (7 days)
* d: days (1 day)
* h: hours (3,600,000,000,000 nanoseconds)
* m: minutes (60,000,000,000 nanoseconds)
* s: seconds (1,000,000,000 nanoseconds)
* ms: milliseconds (1,000,000 nanoseconds)
* us or µs : microseconds (1000 nanoseconds)
* ns: nanoseconds (1 nanosecond)