Assumptions

- As per the general notation, columns with '_date' suffix contain only date type values, and columns with '_time' suffix contain timestamp type values. As 'trunc' function used in the sample code, I assume that dss_start_date column contains timestamp values as well, in contrary with the general notation. If, aligned with the general notation '_date' suffix column has only date type values using ‘trunc(dt,'day')’ would only cause unnecessary computation power, as it would give the same result with itself.

- Since it is not specified, aligned with the general ETL logic, I assume that the `load_date` field refers the data which ETL is executed for, which is, generally, one day prior to 'date' field.

  load_date = yesterday

  date_add(date,-1,'day') = load_date

- As date_add(trunc(dt,'month'),-1,'day') (last day of the previous month) used in the sample code, I assume that, the data is being tried to capture to show as of the last day of the previous month.

- To reduce the data size, I assumed that the 'update_date' is being filled only if there is an alter/change/update in this field exist, otherwise it will be kept as NULL. Also, as ifnull function used in the sample code, I assume that, NULL values are there in insert entries.

- I was not familiar with 'dss_', as per my research, I am assuming that you are using a plugin called 'DSS' (Dataiku Data Science Studio for SQL) to manage ETL data transfer to Snowflake. And, 'dss_' naming convention was used to identify those tables which is generated by this plugin.

- _be_ may be used to identify Belgium related data.

- I assume, the requirement is to get the data for a given data and having a snapshot data as of that date. As per my experience, having data of a snapshot date has limited use-case as compared to the last date snapshot, For that reason, I would generate a dataset and write this data to a permanent table instead of keeping this as a view, this will serve much more use-cases, and reduce the error risk of using wrong queries by using this Single Source of Truth.

- I would prefer using '>= and <' instead of 'between' in most cases, I would only use 'between' if the columns has date type values only (between may bring less records especially in .59.999 values). Also, this will reduce the computation power slightly, as I do not need to use date_add function
  instead of:
  between trunc(rel.dss_start_date, 'day') and date_add(day, - 1, trunc(rel.dss_end_date, 'day'))
  I would prefer:
  trunc(rel.dss_start_date, 'day') >= dd.load_date AND trunc(rel.dss_end_date, 'day') < dd.load_date
  and even if _date field are date type, simply:
  rel.dss_start_date >= dd.load_date and rel.dss_end_date < dd.load_date