# Sample join Query with where clause:

### Context:

The following query joins the patient table with the observations table from native schema and displays the details of the patients, along with age and observations from 2023 (latest), who have low BP.

select \*

from

native.observations obs inner join native.patients pat

on obs.patient=pat.id

inner join (select distinct patient,value,date

from native.observations

where description =' Diastolic Blood Pressure') bp

on obs.patient = bp.patient

where

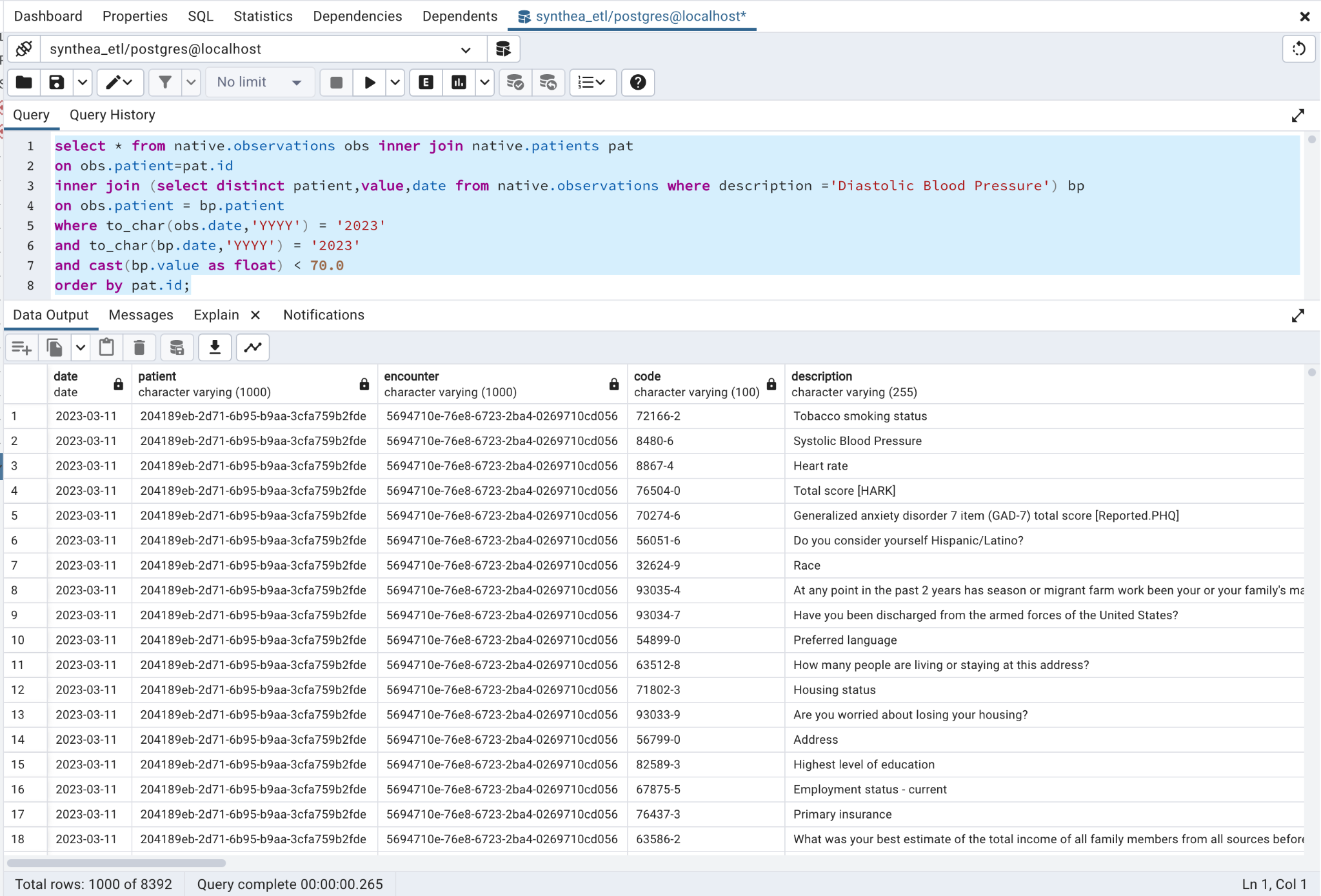
to\_char(obs.date,'YYYY') = '2023'

and to\_char(bp.date,'YYYY') = '2023'

and cast(bp.value as float) < 70.0

order by pat.id;

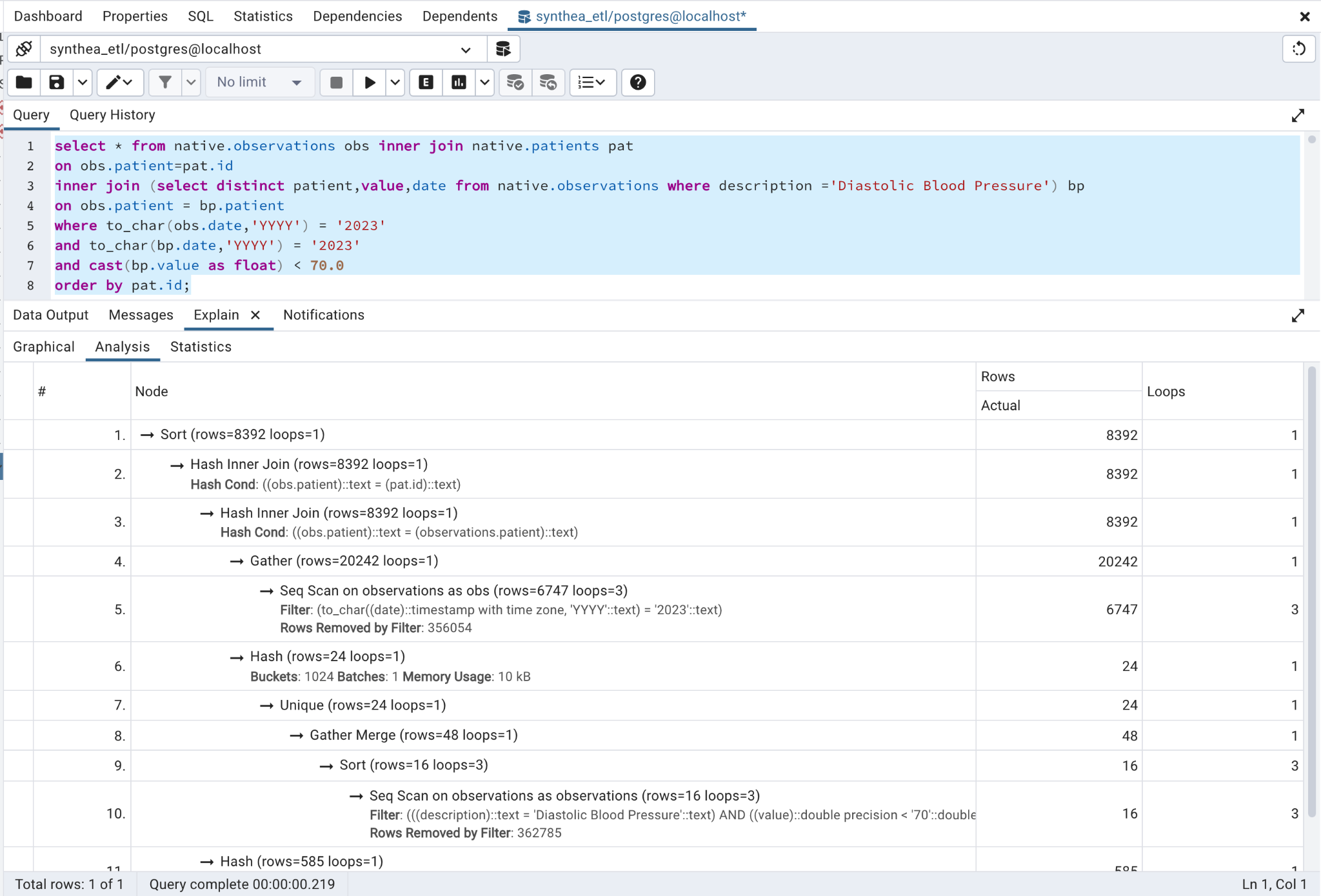
### Query Results:



### Explain Analysis:

The following explain analysis shows the execution analysis stats.

* Each action is shown with the rows returned and what condition was mentioned in action
* Since the observations table has more than 1 million records, the seq scan is the costliest action. It's scanning the data in loops.



### Improvisation:

* Instead of using SELECT \*, using the specific columns is always recommended
* Since we need only the data from the year 2023, we can apply this as a filter in the sub-query, bp, so that only required data is fetched and stored in memory.
* If some of the columns are often used to filter data and thus appear in the where clause, we can create indexes on these columns and thus the performance will improve.
* Whenever possible, if some calculations like Age appear often in the queries, then such calculations can be added as a column in the table directly, instead of calculating it again and again.