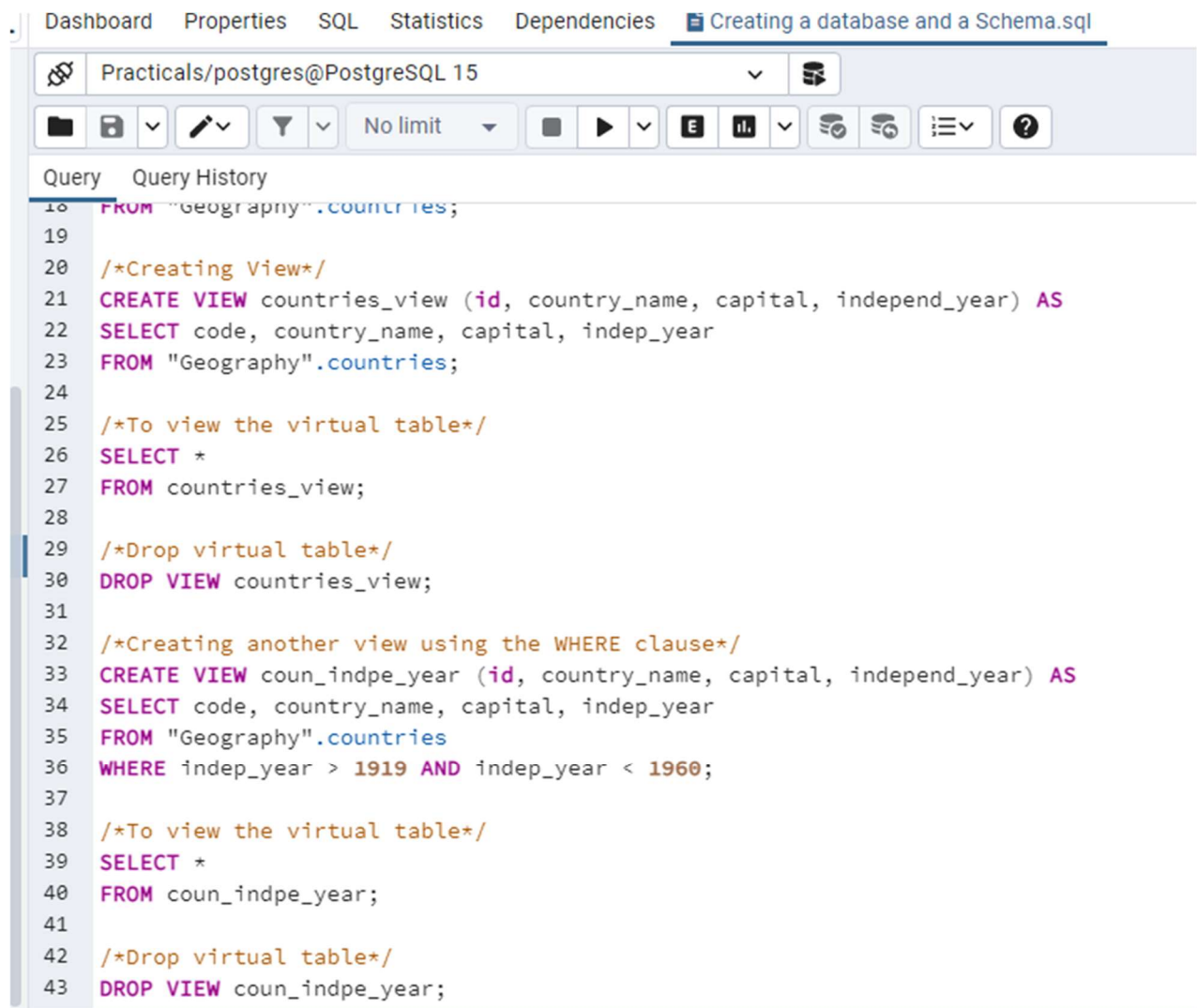


DAY 5: CREATING SQL VIEWS

A SQL view is a virtual table that does not actually exist. It does not have any data stored in it. It exists only as a definition of a schema object. This virtual table can be invoked by calling its name in a query as you would a base table. The simplest type of view to create is one that references only one table and retrieves data from columns within the table without modifying that data.

The first line is the name of the view, COUNTRIES_VIEW, followed by the name of each of the columns: ID, COUNTRY_NAME, CAPITAL, and INDEPEND_YEAR. If the column names were omitted, the view columns would inherit the names from the original table columns. The DROP VIEW statement is used to remove the virtual table; however, none of the data in the original table is affected.



The screenshot shows a PostgreSQL SQL editor window titled "Creating a database and a Schema.sql". The connection is "Practicals/postgres@PostgreSQL 15". The SQL editor contains the following code:

```
18 FROM "Geography".countries;
19
20 /*Creating View*/
21 CREATE VIEW countries_view (id, country_name, capital, independ_year) AS
22 SELECT code, country_name, capital, indep_year
23 FROM "Geography".countries;
24
25 /*To view the virtual table*/
26 SELECT *
27 FROM countries_view;
28
29 /*Drop virtual table*/
30 DROP VIEW countries_view;
31
32 /*Creating another view using the WHERE clause*/
33 CREATE VIEW coun_indpe_year (id, country_name, capital, independ_year) AS
34 SELECT code, country_name, capital, indep_year
35 FROM "Geography".countries
36 WHERE indep_year > 1919 AND indep_year < 1960;
37
38 /*To view the virtual table*/
39 SELECT *
40 FROM coun_indpe_year;
41
42 /*Drop virtual table*/
43 DROP VIEW coun_indpe_year;
```