TASK 1 CREATE NEW DATABASE

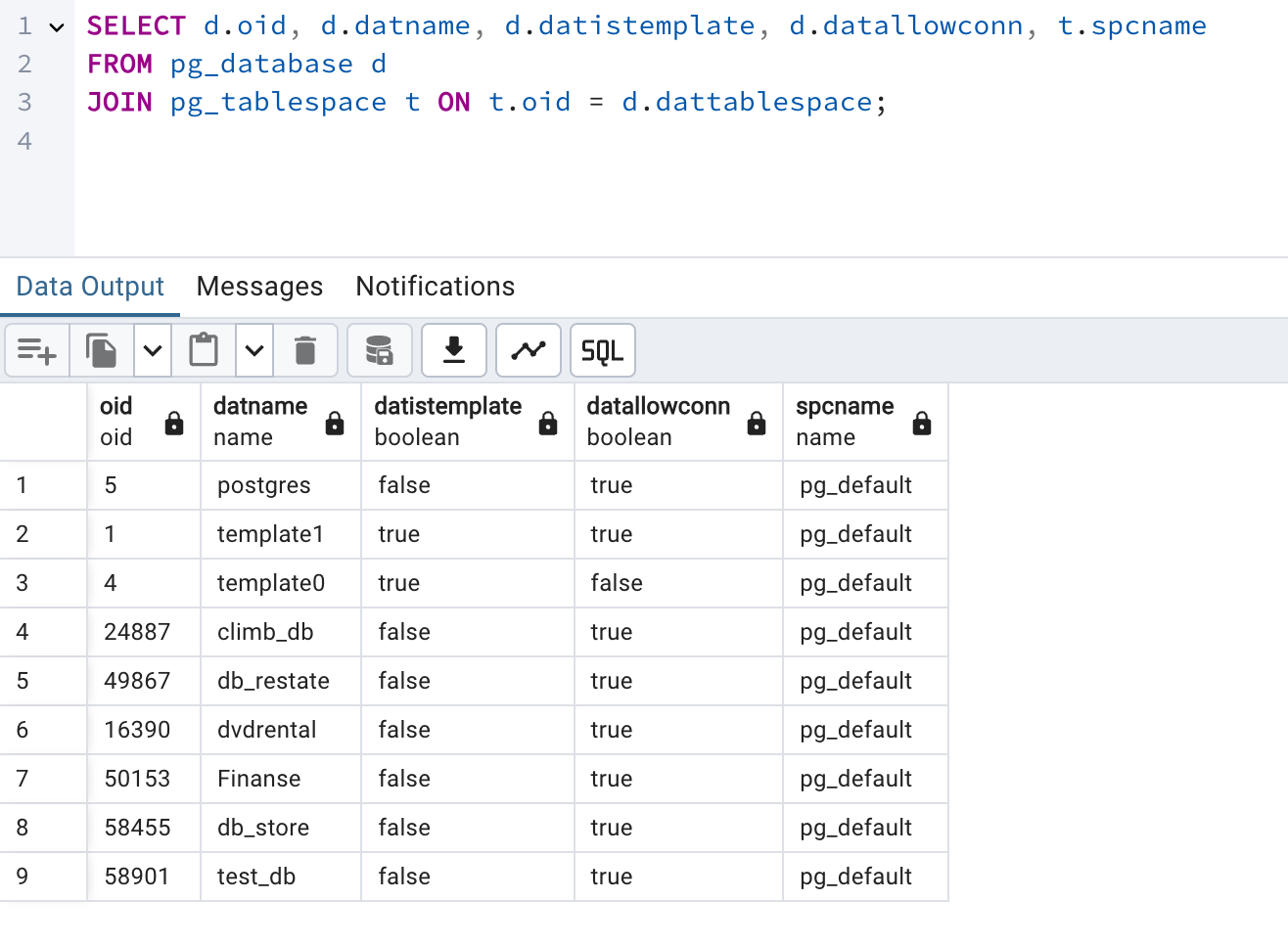
1.Connect to the postgres Database and create a new one named “test\_db”.

2. Run query, investigate result:

SELECT d.oid, d.datname, d.datistemplate, d.datallowconn, t.spcname

FROM pg\_database d

JOIN pg\_tablespace t ON t.oid = d.dattablespace;



test\_db is a user-created database just now.

Oid is unique identifier.

Datname is name of the database.

datistemplateindicates whether the database is a template and template databases are used as the basis for creating new databases.

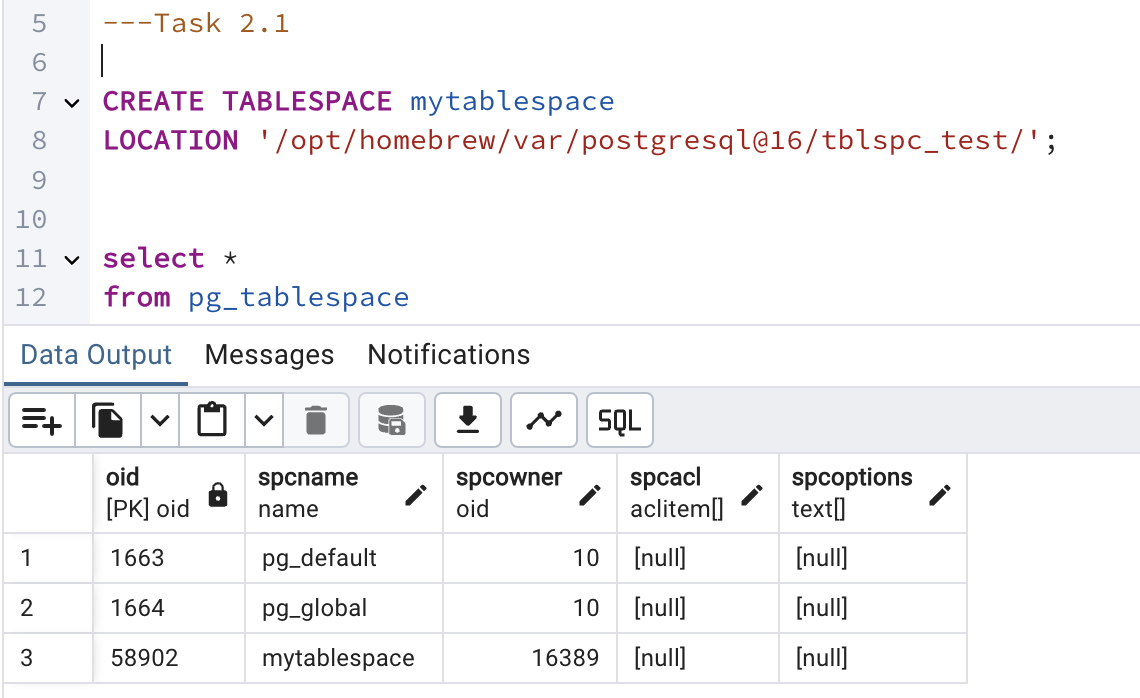
datallowconn indicates whether connections to the database are allowed

spcnameIndicates the tablespace associated with the database. pg\_default is the default tablespace.

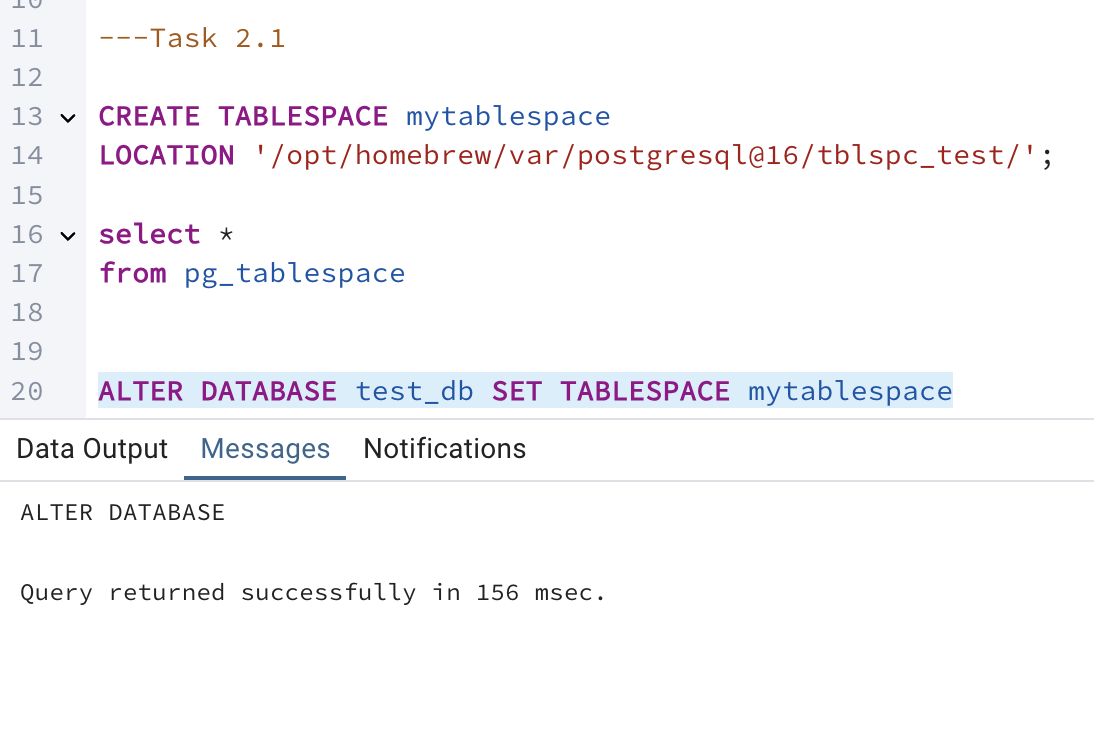
TASK 2 CREATE NEW TABLESPACE

**select** \*

**from** ***pg\_tablespace***

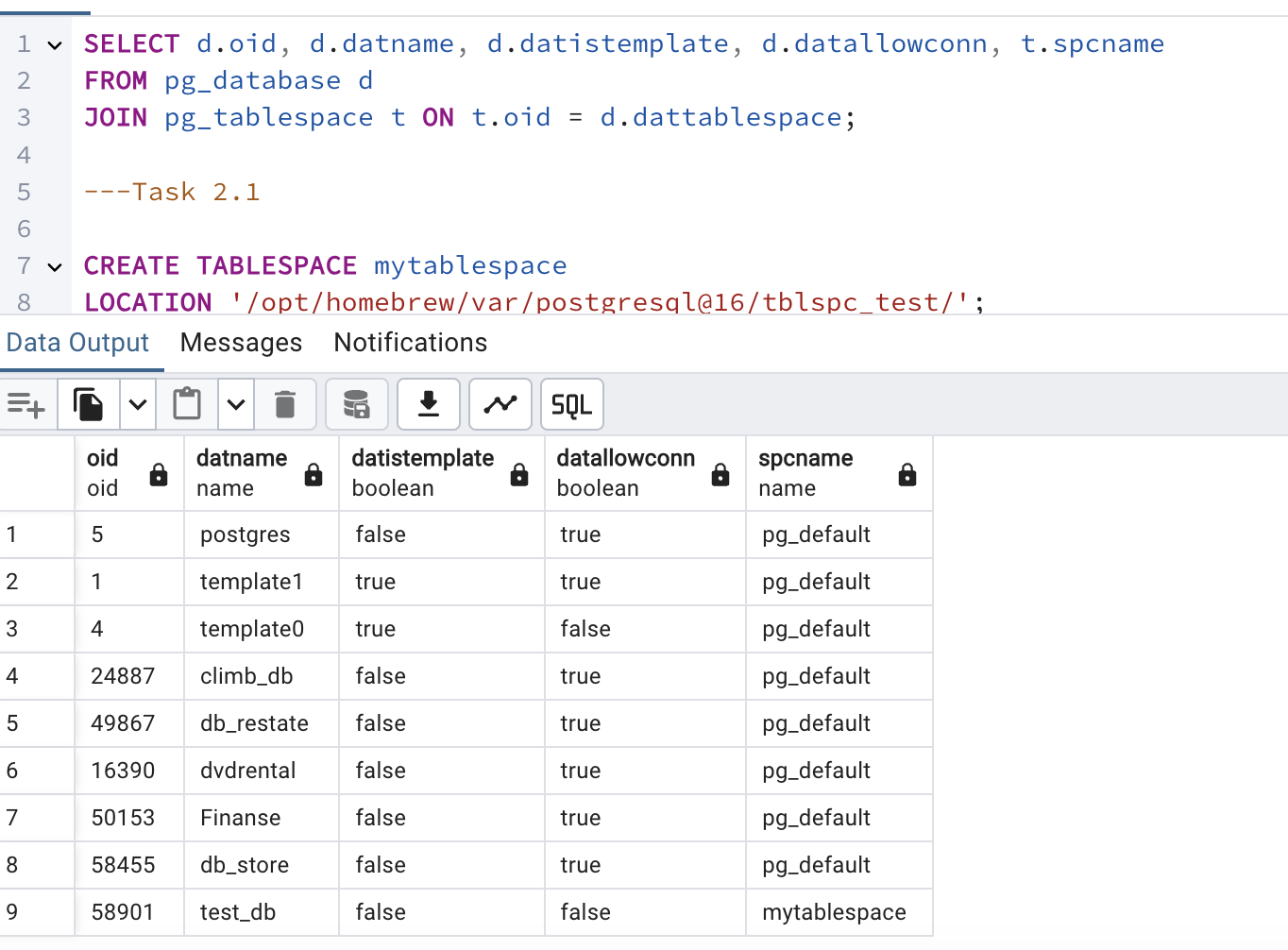
******

Here you can see it’s created successfully.

It tells names of spcnames. 

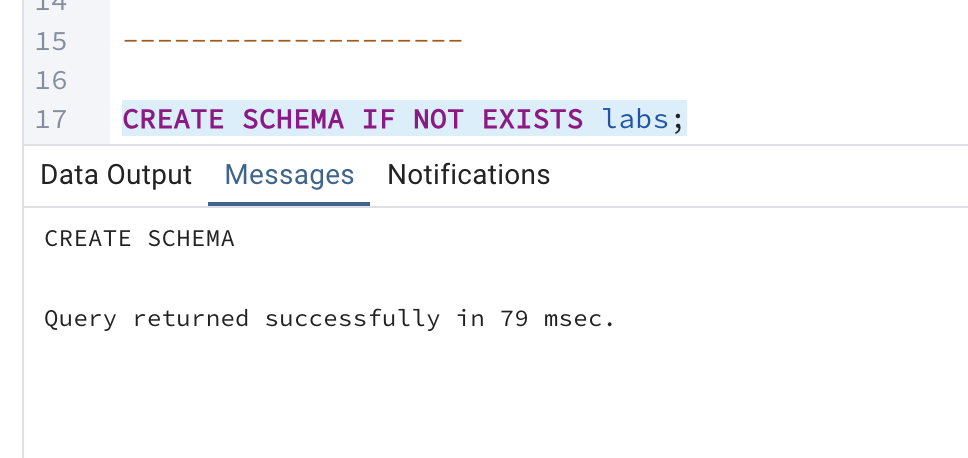
ALTER DATABASE test\_db SET TABLESPACE mytablespace

I altered tablespace



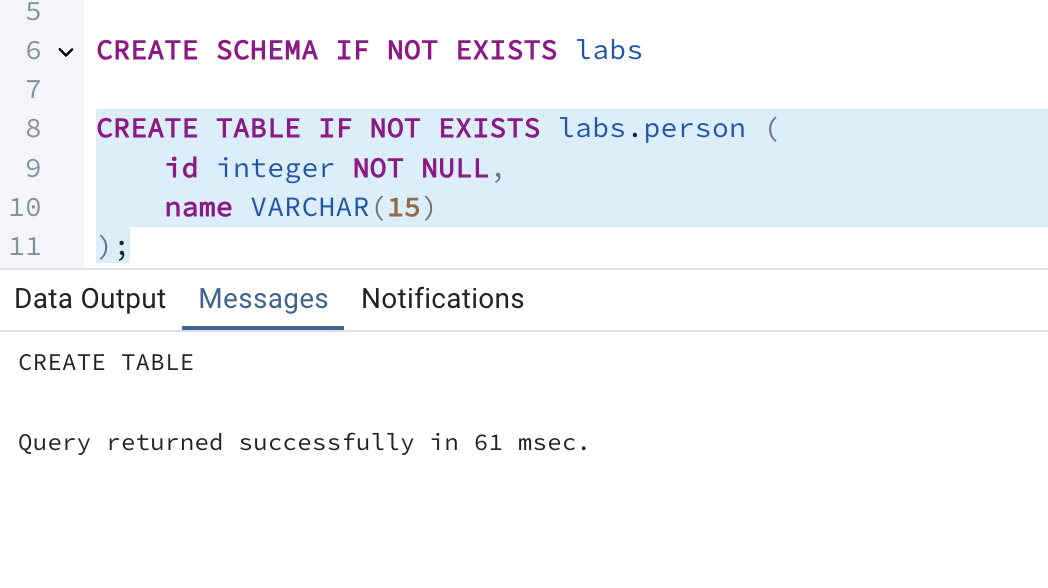
Tablespace changed for our test\_db database, after this query all data will be located in specified location.

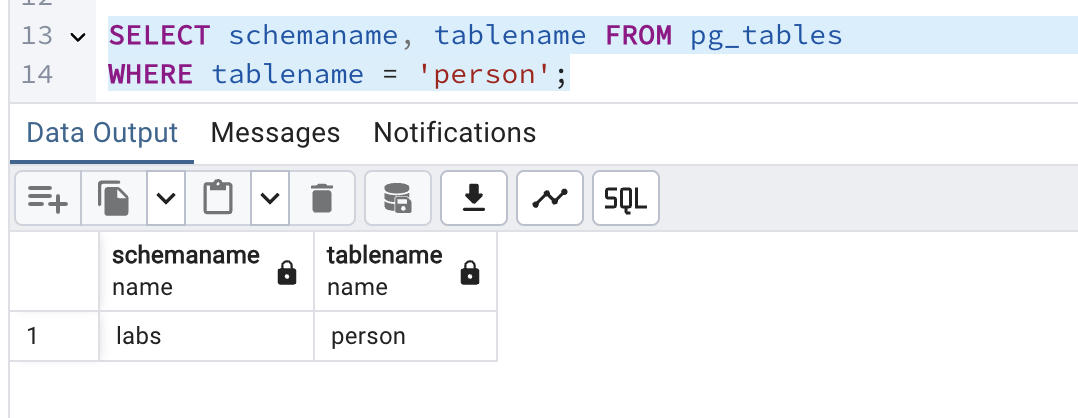
2.3 TASK 3 CREATE NEW SCHEMA

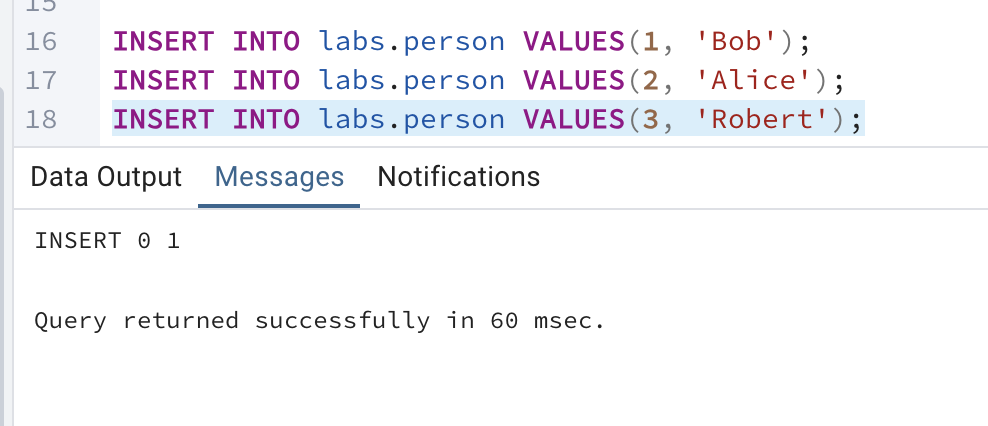


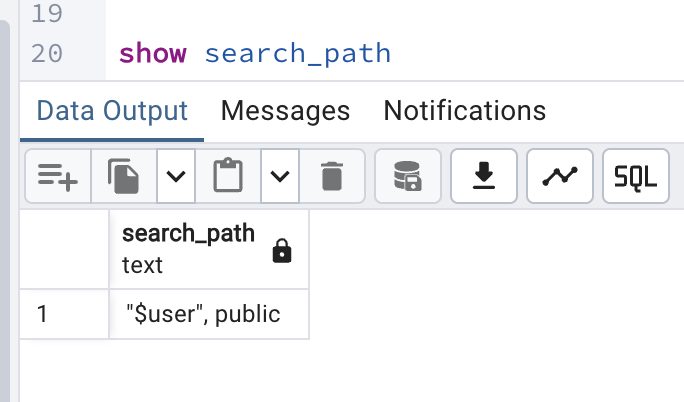
Schema created

TASK 3 CREATE NEW SCHEMA

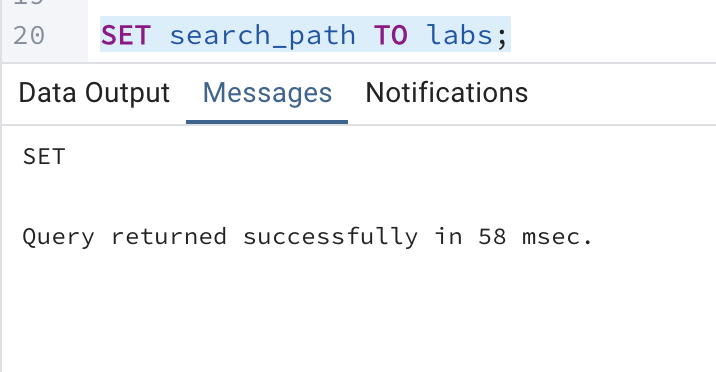


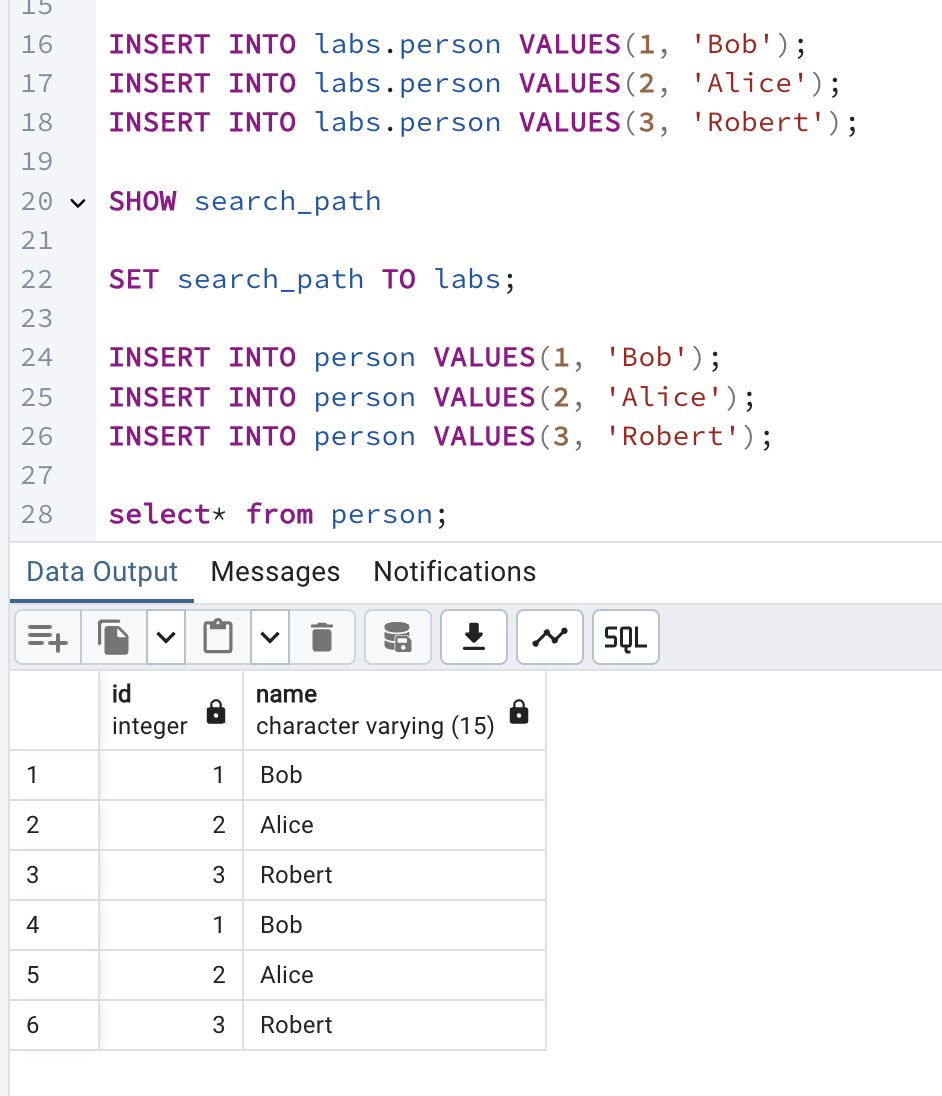


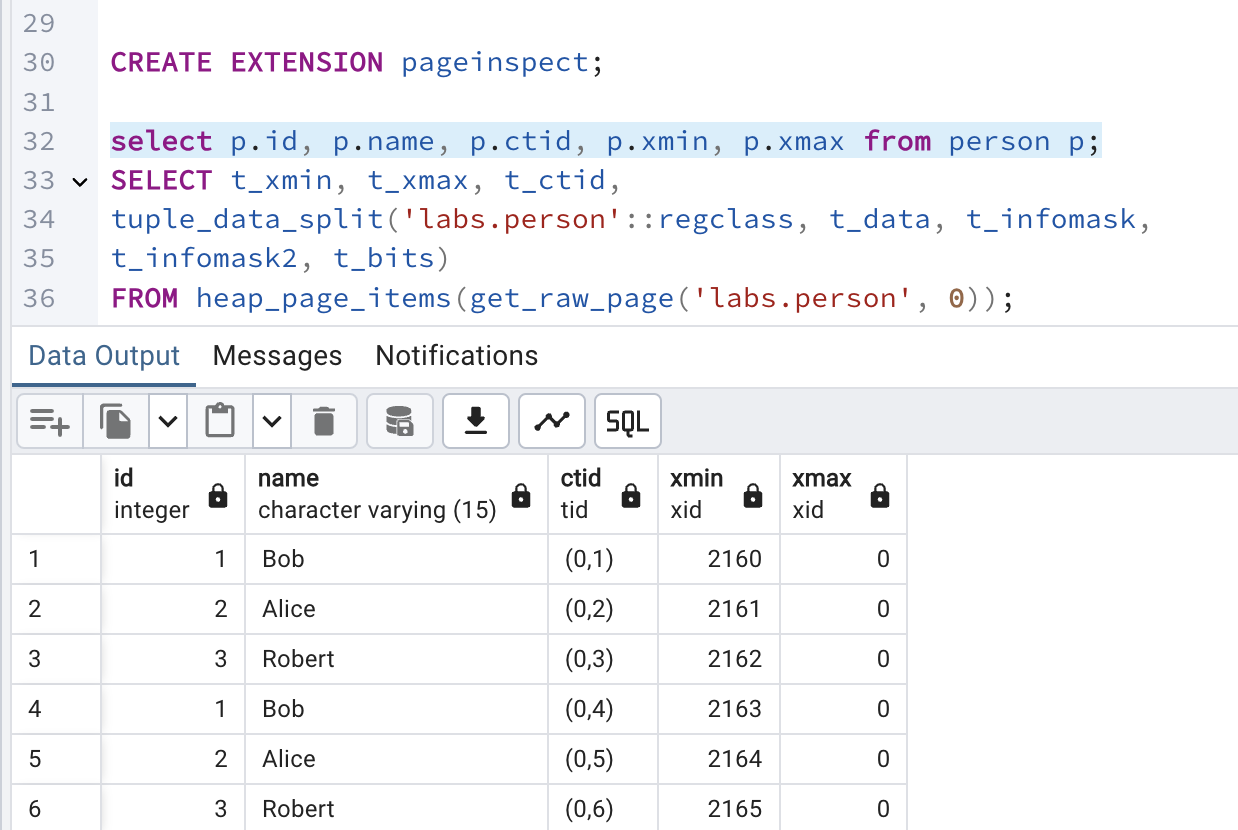




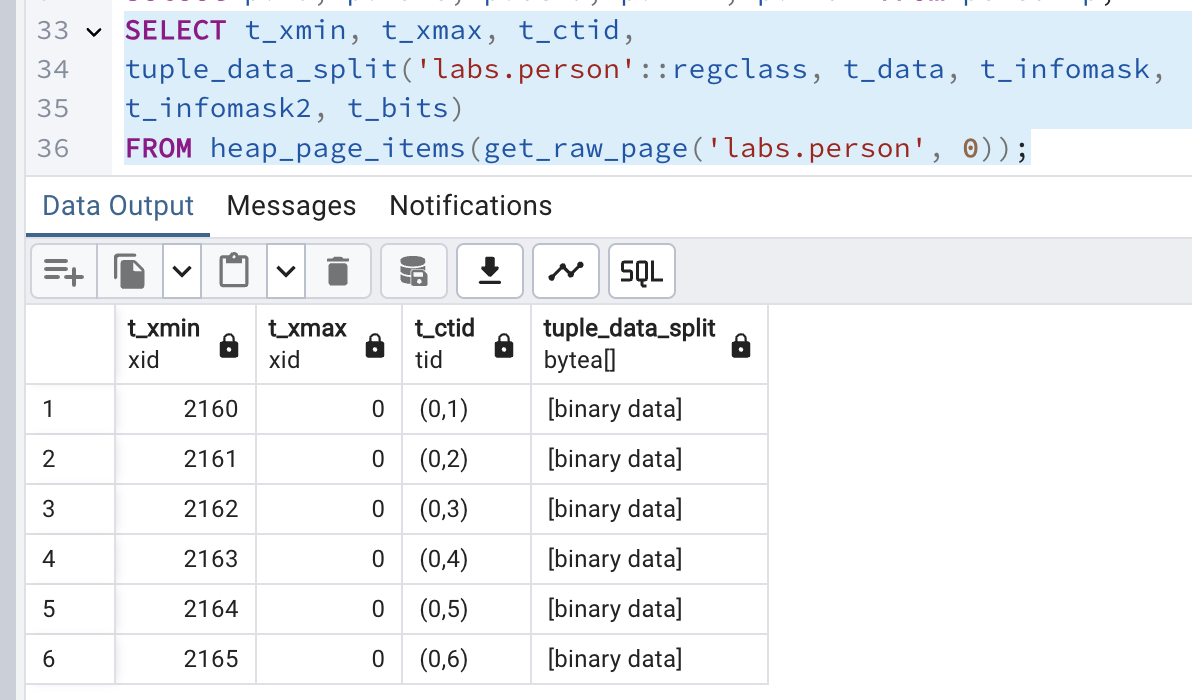
This query shows the default search schema name , in which postgres will search for tables if we didn’t specify schema name in the query.



Here we set the schema name to tell postgres that we are working in labs schema.

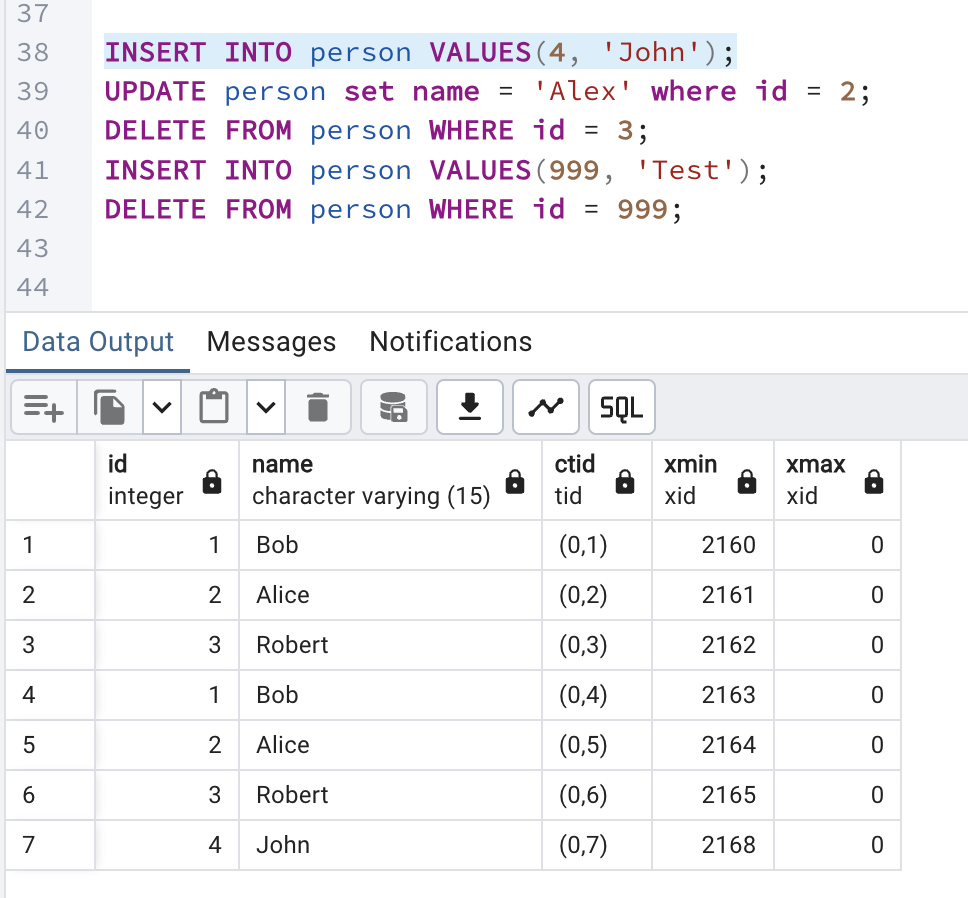
TASK 4 INVESTIGATE MVCC\*

Extension created. This query shows following information id show’s unique identifier of the row, name is the name we set when creating this table, ctid is a unique identifier for the row within its table. It consists of the block number and the position within the block. Xmin shows transaction ID of the INSERT transaction.xmax shows deleting transaction ID , it show’s 0 as the rows are not deleted right now.

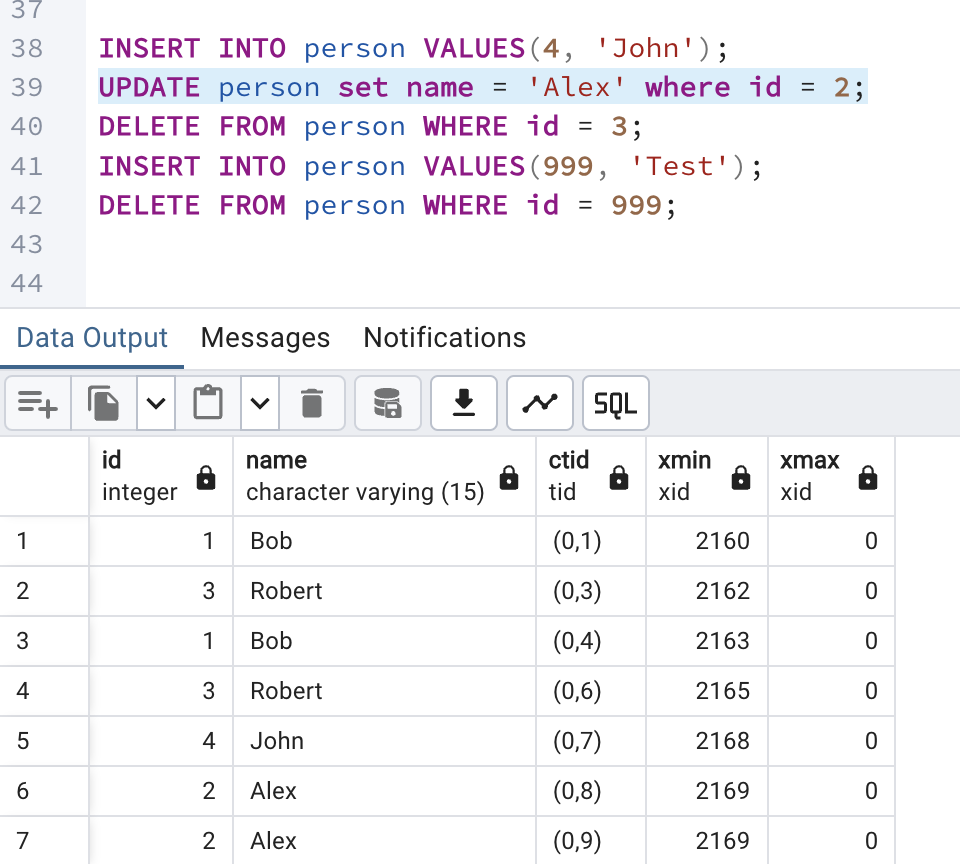


get\_raw\_page function retrieves the raw data for page 0 of the labs.person table. heap\_page\_items function takes the raw page data and returns a set of rows, each representing a tuple (row) in the specified page.

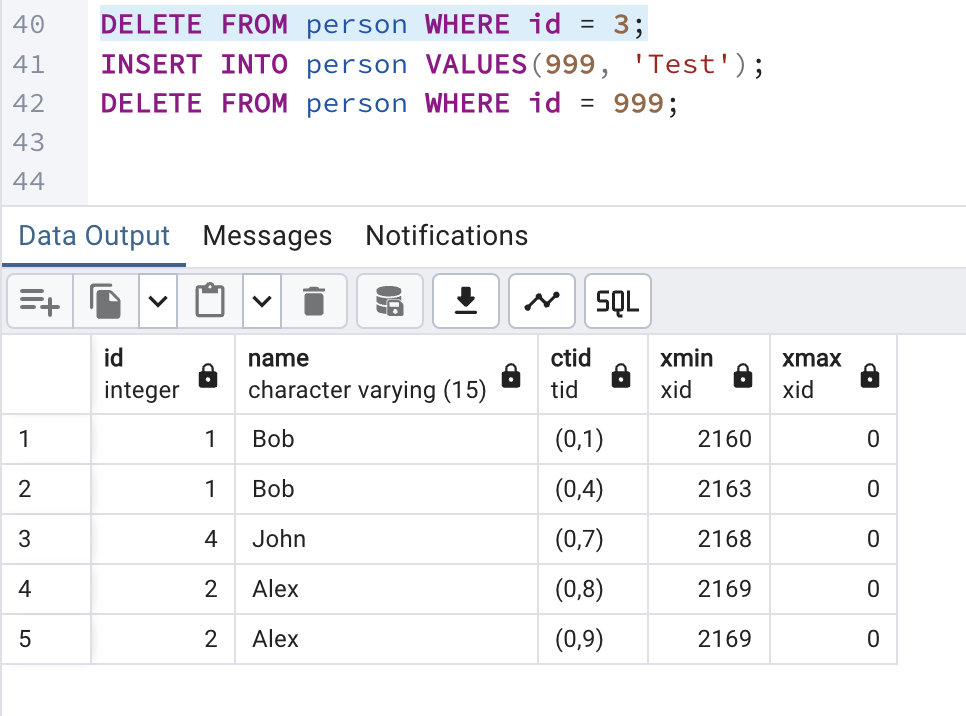
A t\_xmax of zero means the tuple is currently not marked for deletion or update



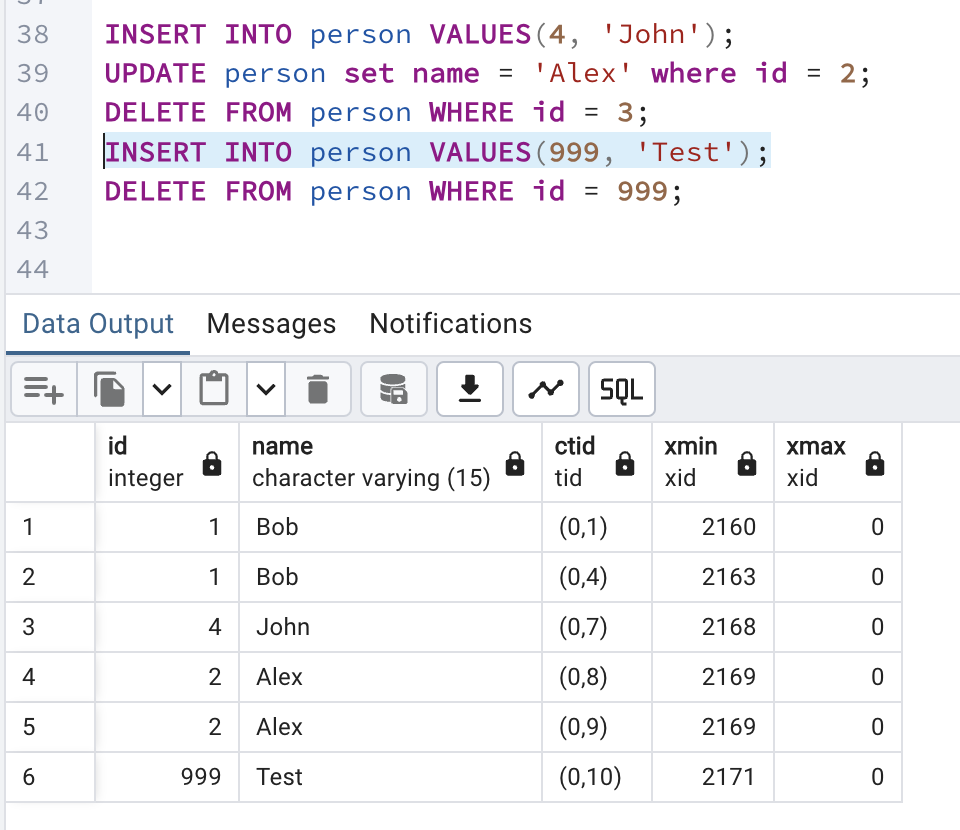
Inserted new row , and the xmin id is 2168



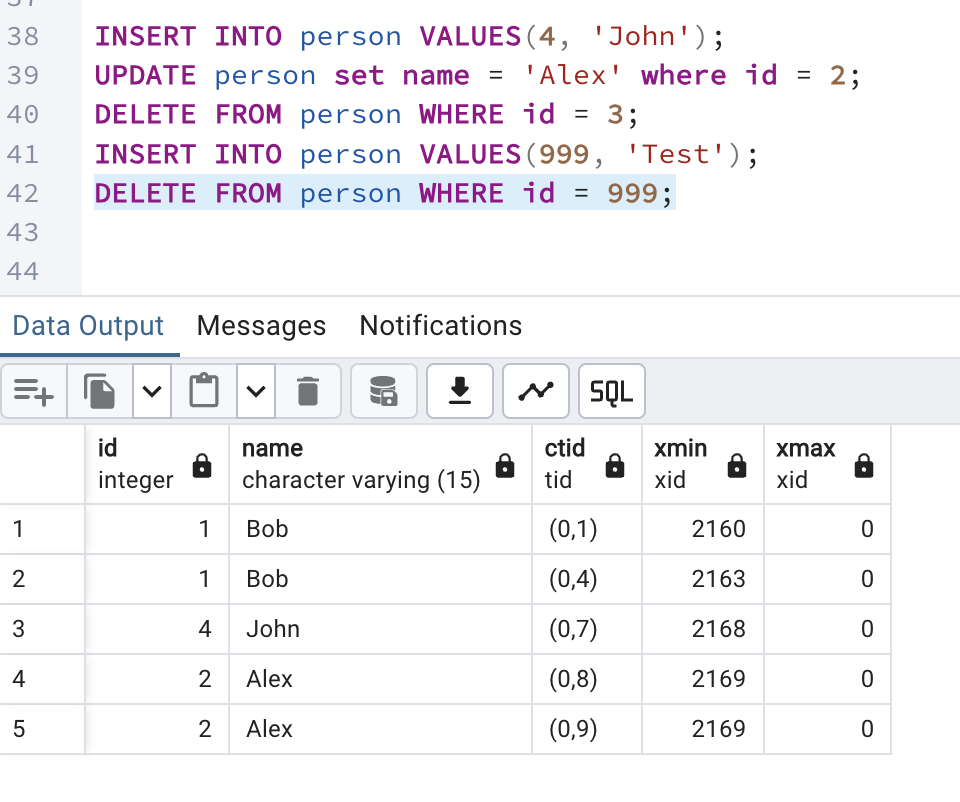
When we update existing row its got new ctid and and the xmin ID was updated to last transaction ID.



Deleted id 3

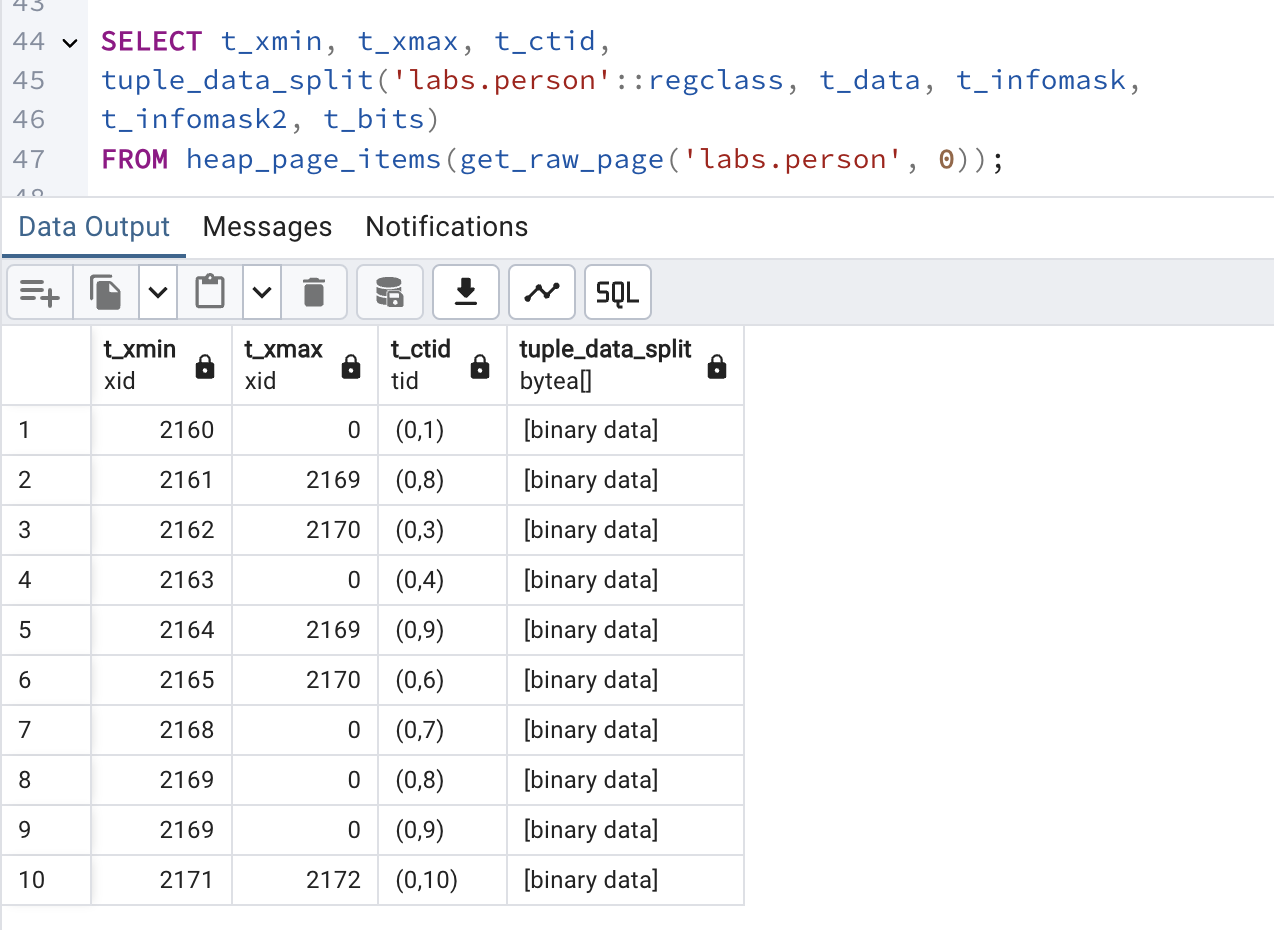


Insert id 999

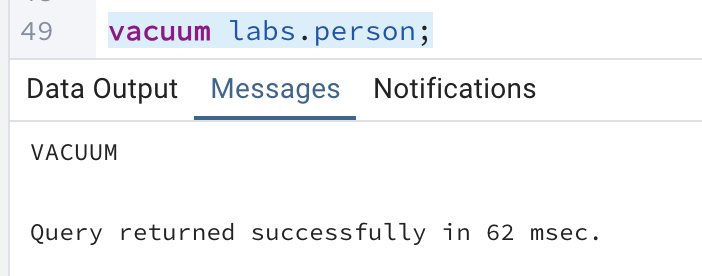


and now I deleted id 999

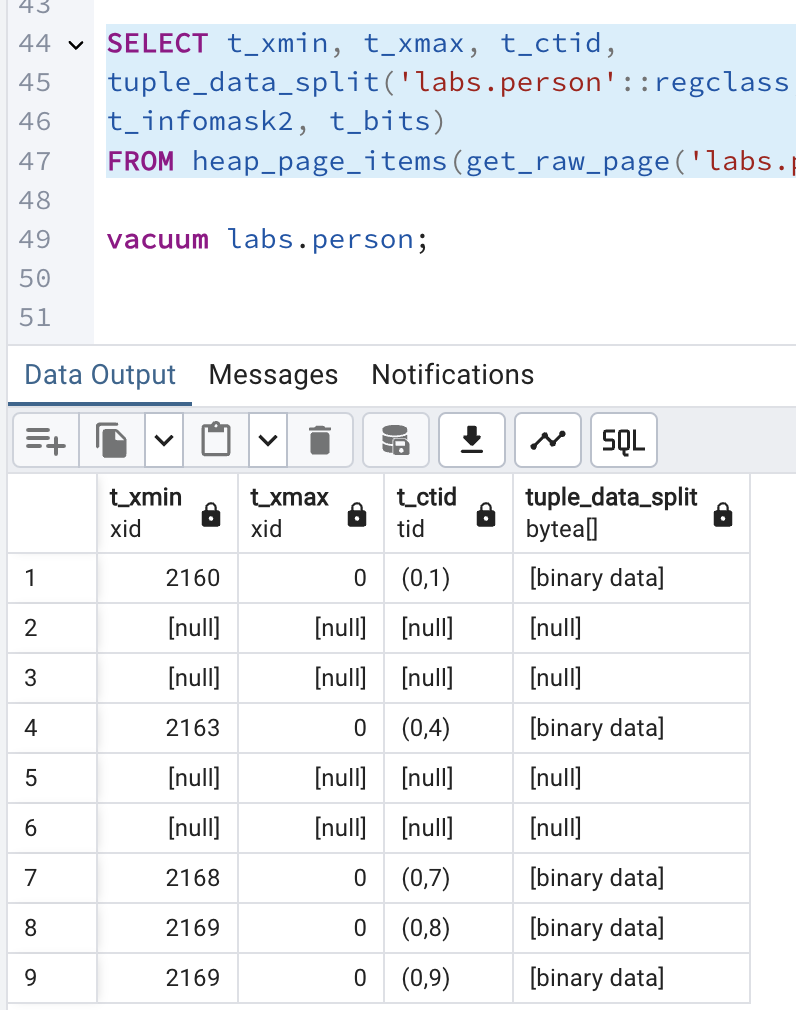
TASK 5 INVESTIGATE VACUUM



When we delete some rows the t\_xmax id shows that the row is deleted by that transaction ID but it’s remains heap\_page\_items so , to clear all deleted rows we need to execute VACUUM

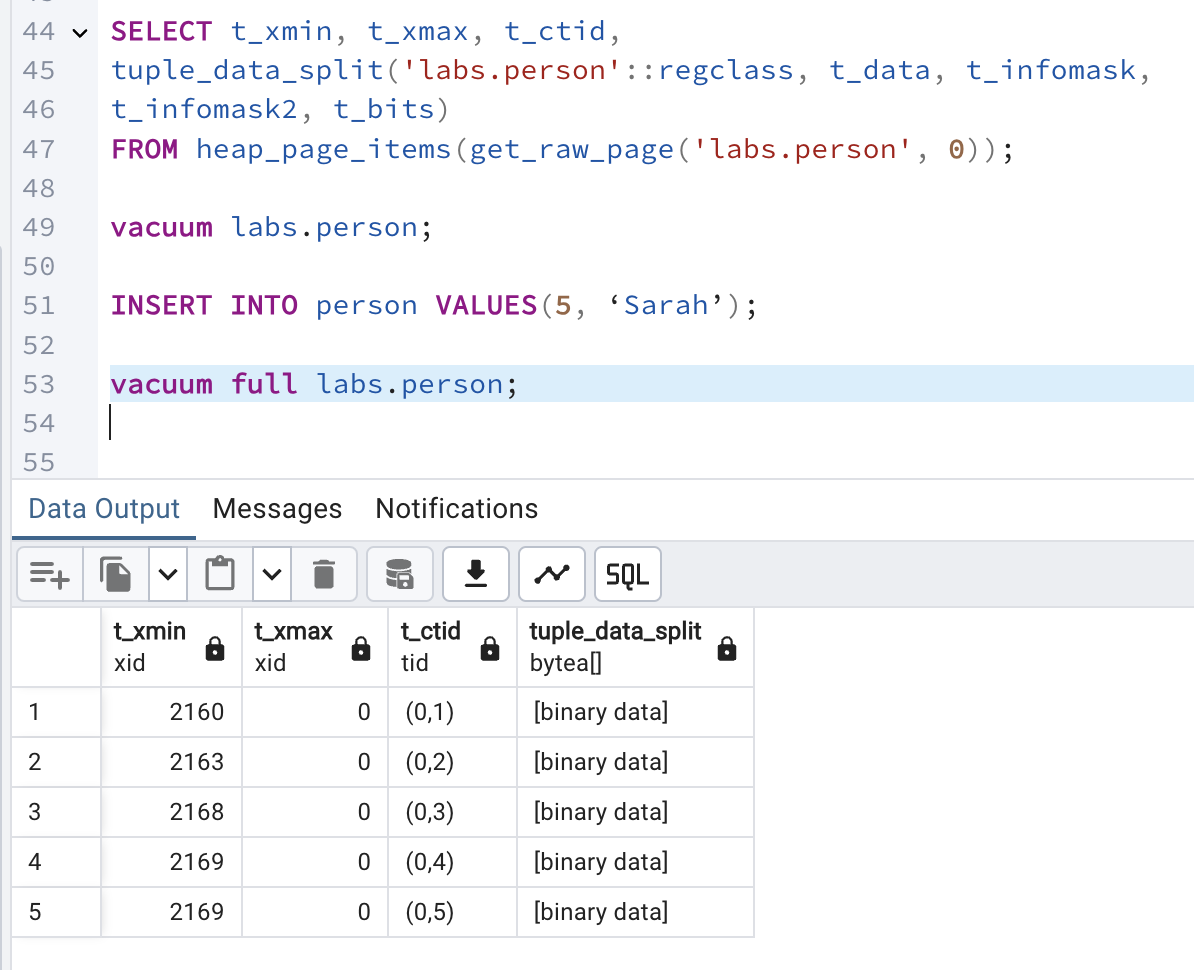


After vacuum



After executing vacuum we see that deleted row gone.

After vacuum full the result is:



VACUUM FULL rewrites the entire contents of the table into a new disk file with no extra space, allowing unused space to be returned to the operating system.