DROP TABLE rivers\_and\_waterfalls; --dropping table so everything can be created again

CREATE TABLE rivers\_and\_waterfalls( Rivers VARCHAR(200), Waterfalls VARCHAR (200)); --creating table rivers\_and\_waterfalls

ALTER TABLE rivers\_and\_waterfalls ADD COLUMN episodes VARCHAR(100); --altering table to add column episodes

INSERT INTO rivers\_and\_waterfalls(episodes) SELECT episode FROM bobross; --episodes from rivers\_and\_waterfalls and bobross.episode is the association between bobross table and rivers\_and\_waterfalls table.

UPDATE rivers\_and\_waterfalls SET rivers = (SELECT (

SELECT title FROM bobross WHERE bobross.episode = rivers\_and\_waterfalls.episodes)

)FROM bobross WHERE bobross.episode = rivers\_and\_waterfalls.episodes AND bobross.river= 1; ---subqueries --bringing data to the new table. Titles of the episodes that have rivers

UPDATE rivers\_and\_waterfalls SET waterfalls = (SELECT (

SELECT title FROM bobross WHERE bobross.episode = rivers\_and\_waterfalls.episodes)

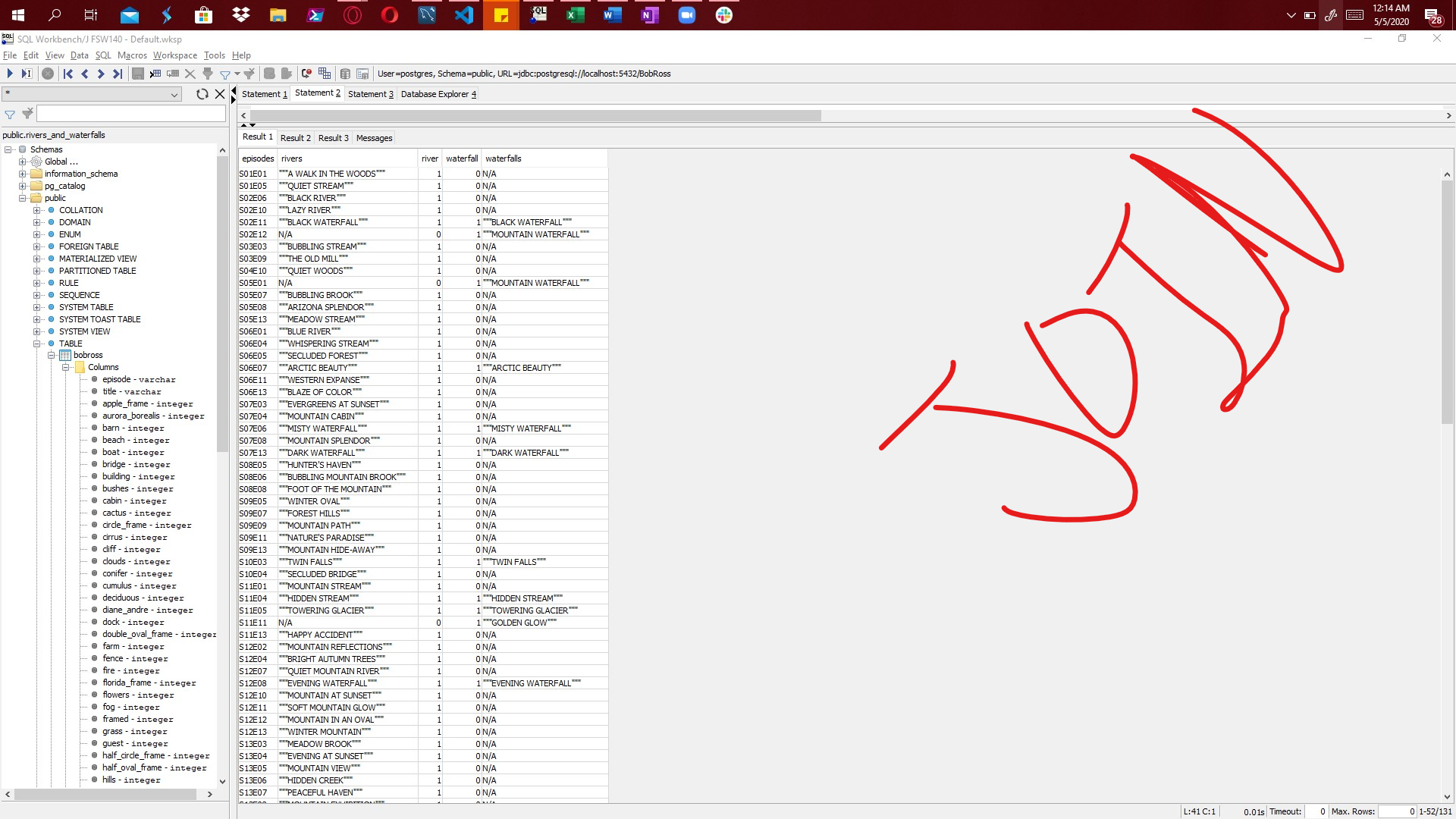
)FROM bobross WHERE bobross.episode = rivers\_and\_waterfalls.episodes AND bobross.waterfall= 1; --bringing data to the new table. Titles of the episodes that have waterfalls

DELETE FROM rivers\_and\_waterfalls WHERE waterfalls IS NULL AND rivers IS NULL; --getting rid of episodes where there is no waterfall nor river

UPDATE rivers\_and\_waterfalls SET rivers = 'N/A' WHERE waterfalls IS NOT NULL AND rivers IS NULL; --labeling “N/A” where the is one of them but not the other. Here we label rivers.

UPDATE rivers\_and\_waterfalls SET waterfalls = 'N/A' WHERE rivers IS NOT NULL AND waterfalls IS NULL; --labeling “N/A” where the is one of them but not the other. Here we label waterfalls

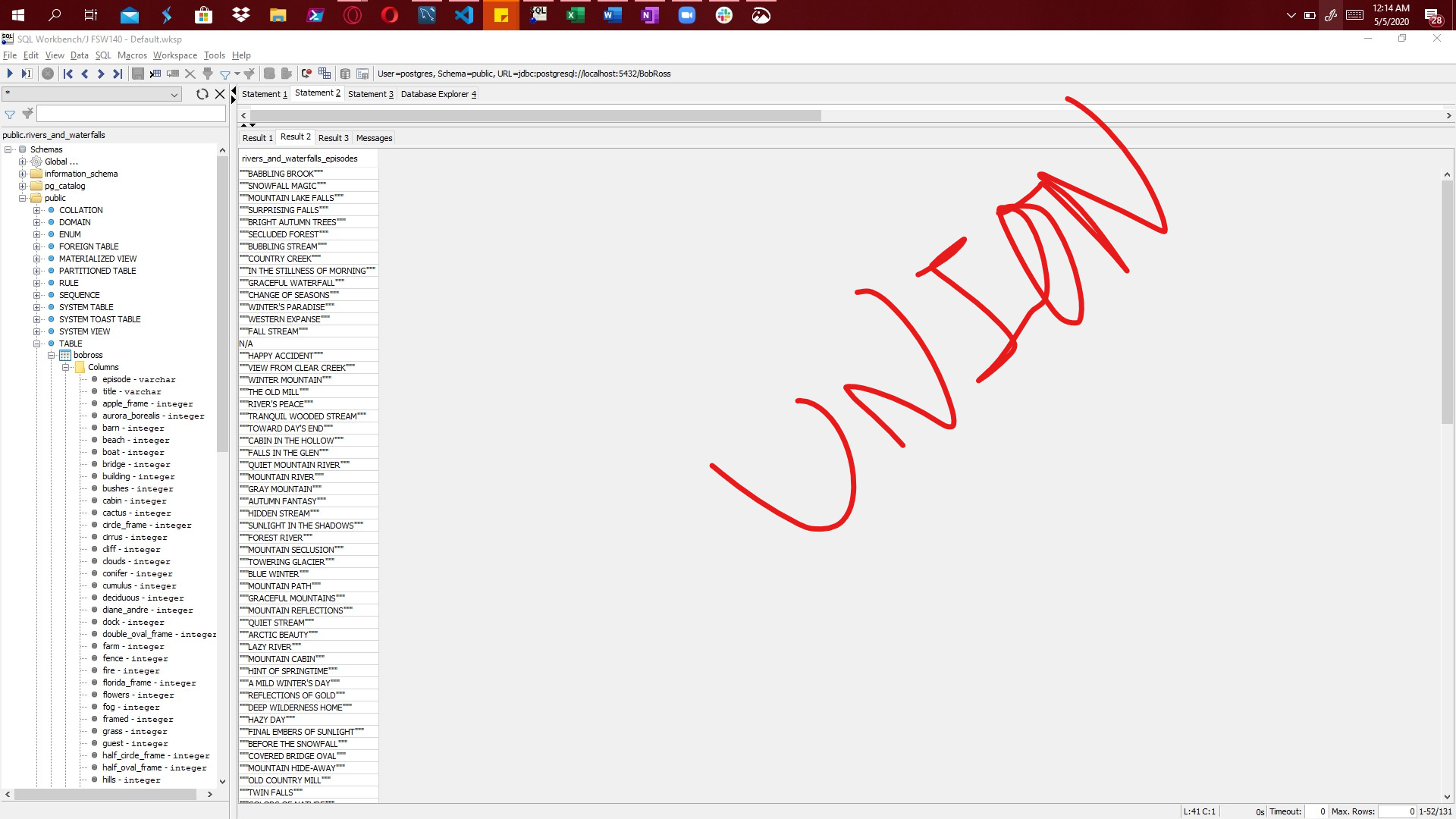
SELECT episodes, rivers, river, waterfall, waterfalls FROM rivers\_and\_waterfalls JOIN bobross ON rivers\_and\_waterfalls.episodes = bobross.episode --multiple table joins expected: columns episodes, waterfalls and rivers from rivers\_and\_waterfalls along side river and waterfall from bobross table



SELECT waterfalls as rivers\_and\_waterfalls\_episodes FROM rivers\_and\_waterfalls

UNION

SELECT rivers FROM rivers\_and\_waterfalls; --set operations; ex: UNION expected: waterfall rows + rivers rows with no repetition of rows that are shared by both waterfalls and rivers.



SELECT DISTINCT COUNT(\*) FROM rivers\_and\_waterfalls WHERE rivers IS NOT NULL AND waterfalls IS NOT NULL GROUP BY rivers; --order of operations ; expected 1 and 5 – ALL titles that have waterfall and river are present only one per group, except for one that is 5, hence I added distinct to it and all that returns is the unique or distinct results which are 1 and 5.

