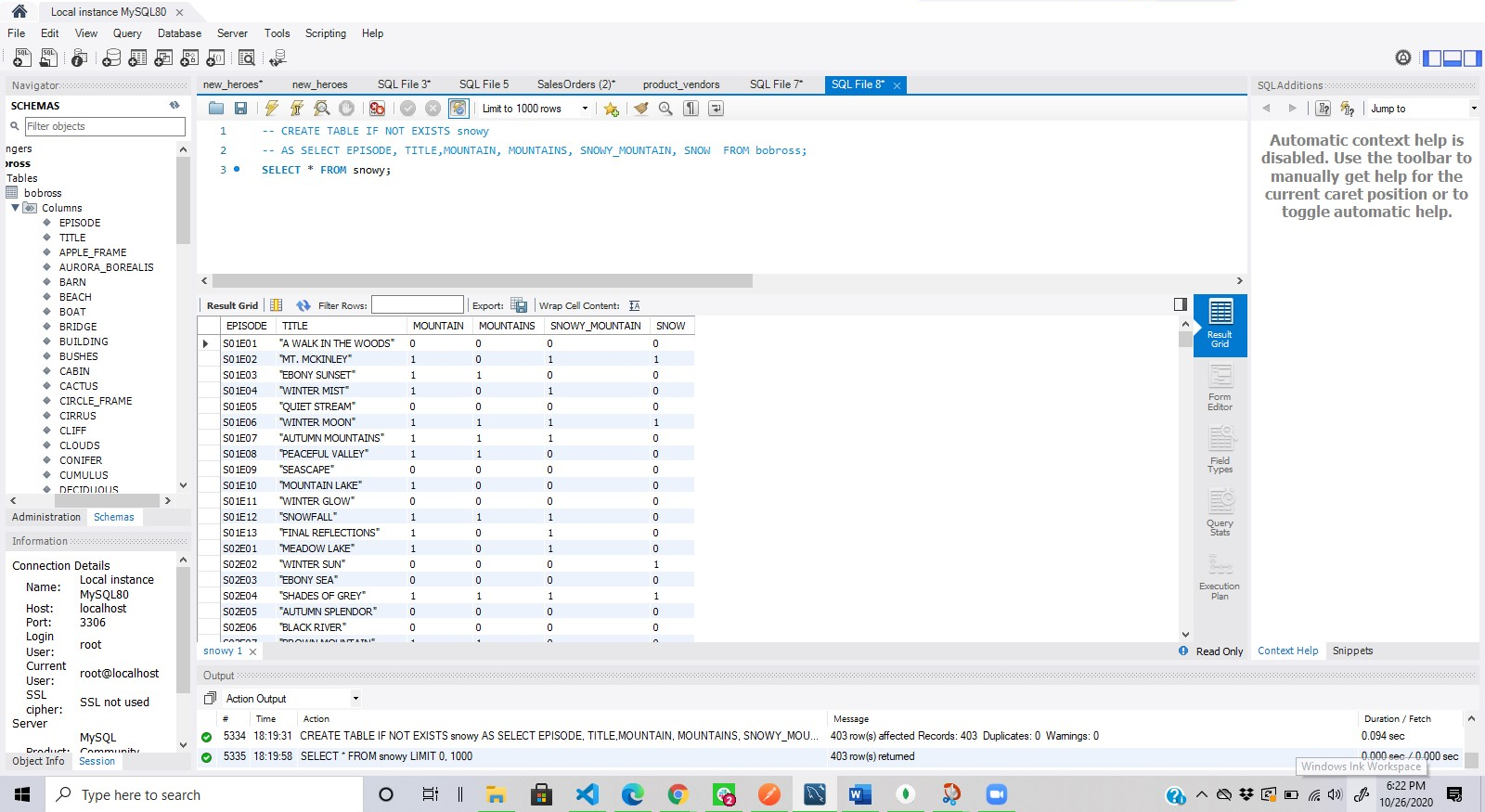
**Week 5. Bob Ross assignment.**

**1. CREATE & SUBQUERY.**

-- CREATE TABLE IF NOT EXISTS snowy

-- AS SELECT EPISODE, TITLE, MOUNTAIN, MOUNTAINS, SNOWY\_MOUNTAIN, SNOW FROM bobross;

SELECT \* FROM snowy;

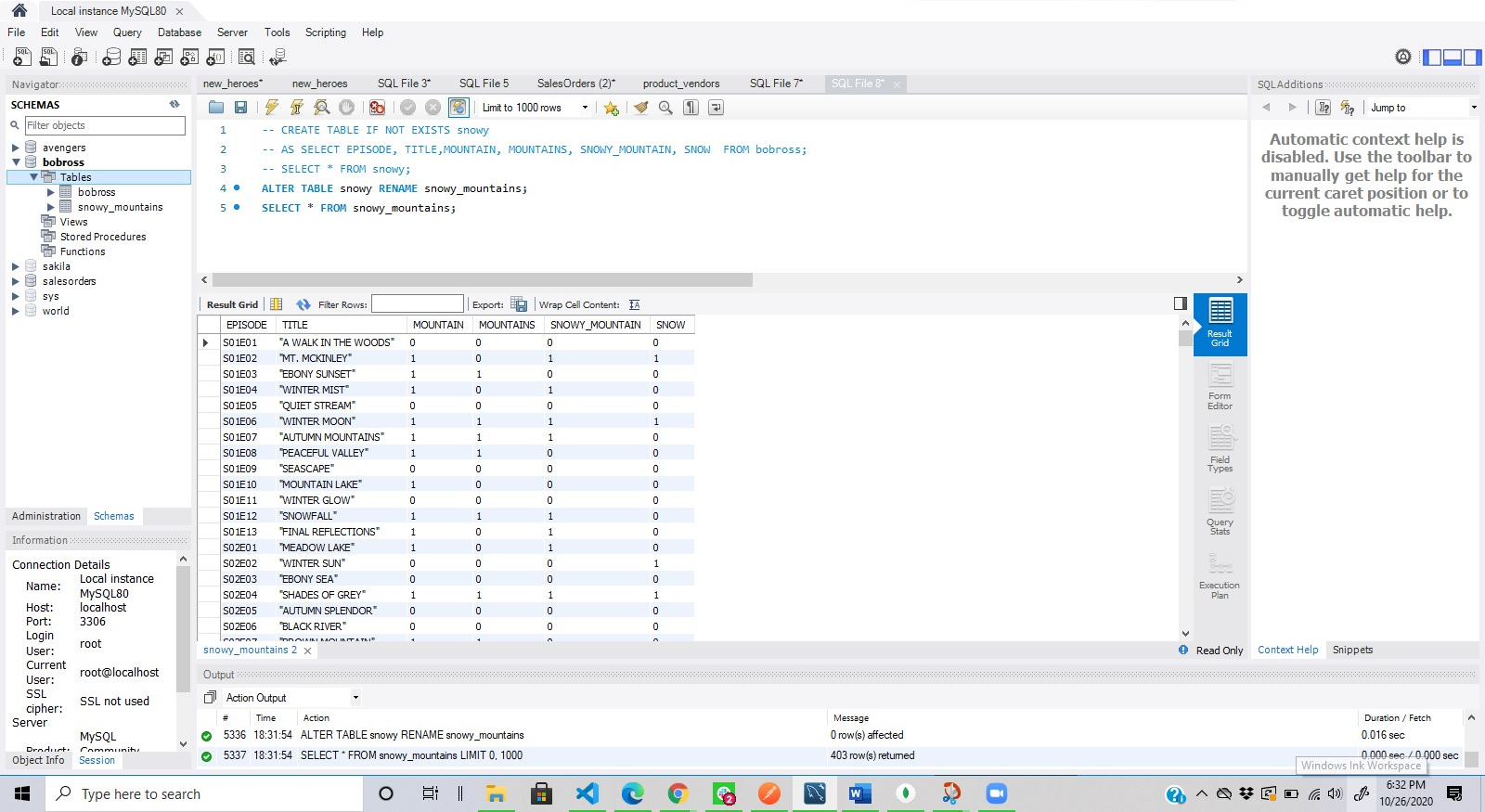


Above, I created a table with episode, title, mountain, mountains, snowy\_mountain, snow, from the Bob Ross table with all the values it already has by using the subquery that selects all the rows of this column of Bob Ross. I used the subquery “SELECT EPISODE, TITLE, MOUNTAIN, MOUNTAINS, SNOWY\_MOUNTAIN, SNOW FROM bobross;” in order to specify which columns I wanted to pull from bobross to my new query.

**2. ALTER, NAME CHANGE.**

ALTER TABLE snowy RENAME snowy\_mountains;

SELECT \* FROM snowy\_mountains;



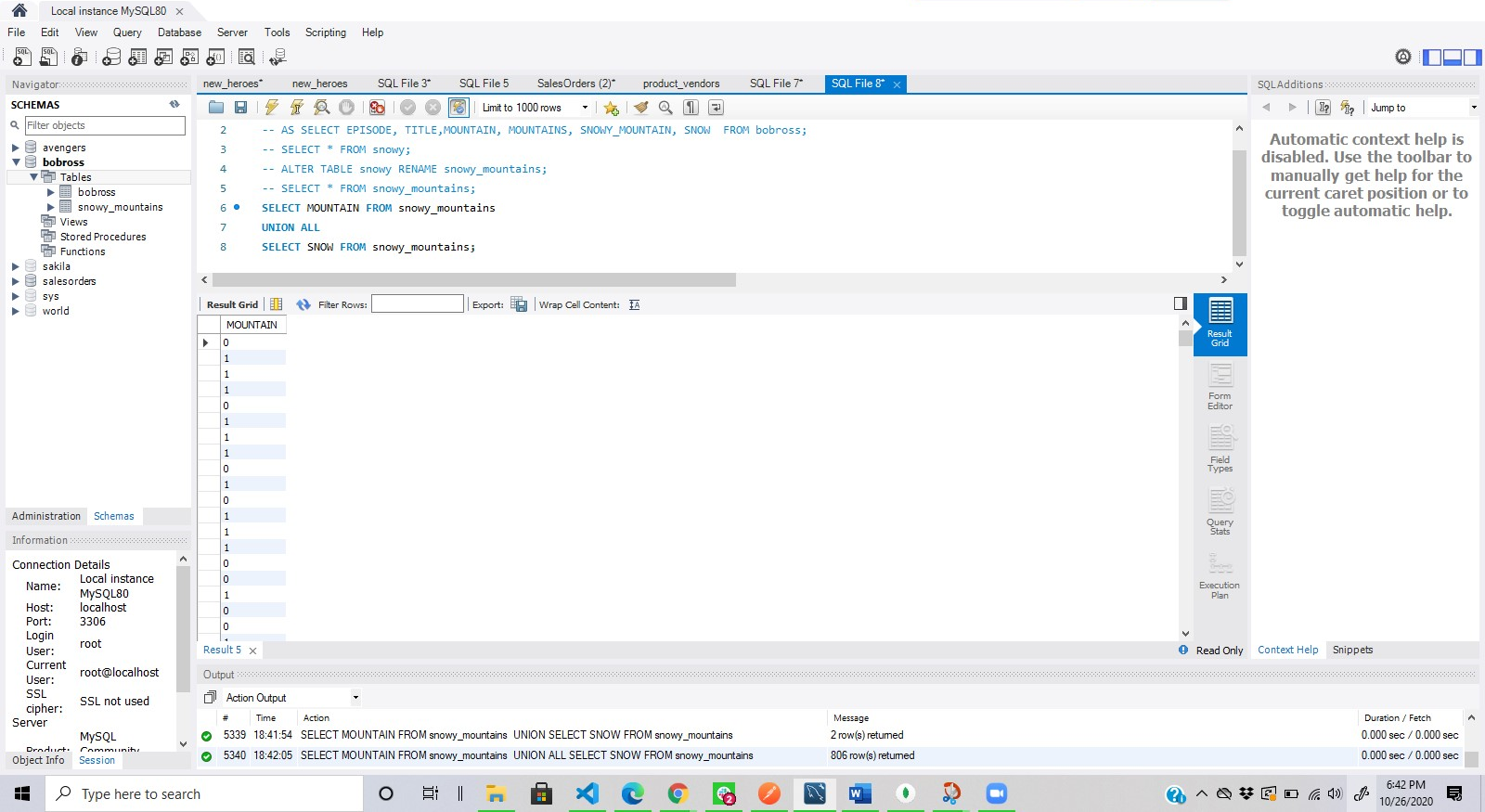
Above we are using the ALTER query to rename the table, from snowy to snowy\_mountains.

**3. SET OPERATIONS (UNION ALL).**

SELECT MOUNTAIN FROM snowy\_mountains

UNION ALL

SELECT SNOW FROM snowy\_mountains;

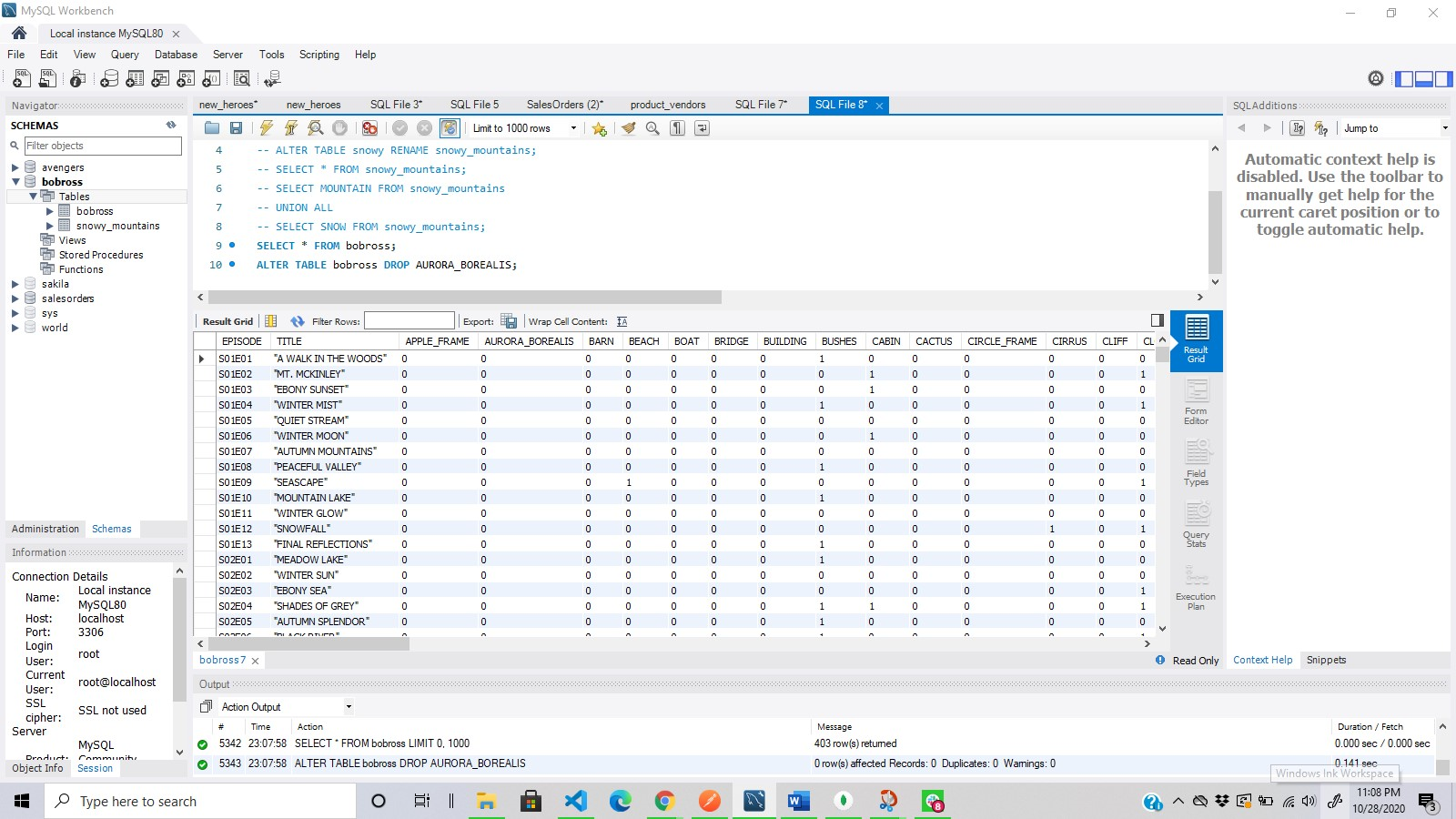


This query above, represent all the values from the MOUNTAIN & SNOW columns, from the snowy\_mountain table. Since its UNION ALL it brings all the values in a single column even the repeated ones.

**4. ALTER TABLE (DROP).**

SELECT \* FROM bobross;

ALTER TABLE bobross DROP AURORA\_BOREALIS;



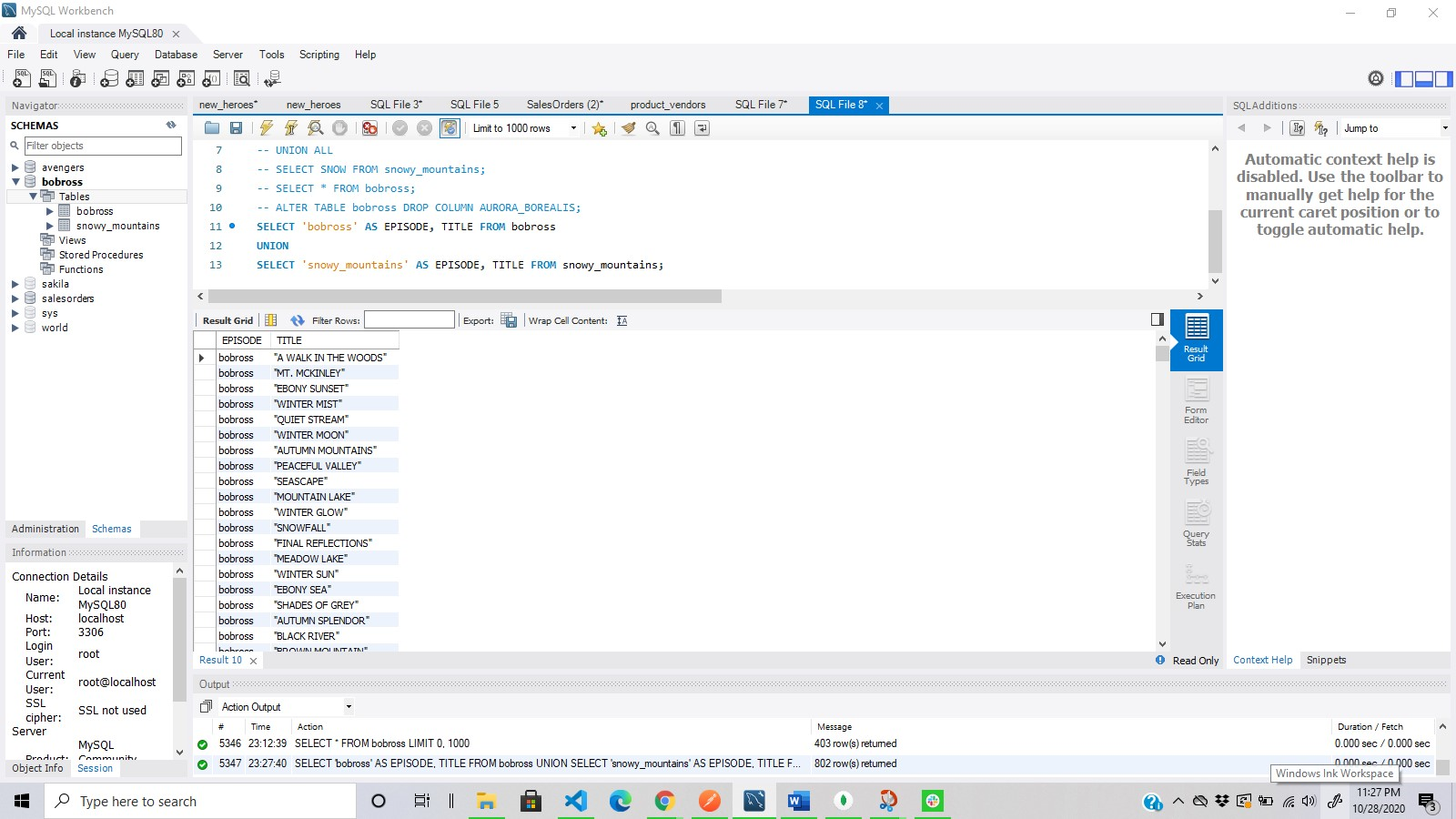
This query (the drop column command) is meant to delete a column from the existing table.

**5. UNION.**

SELECT 'bobross' AS EPISODE, TITLE FROM bobross

UNION

SELECT 'snowy\_mountains' AS EPISODE, TITLE FROM snowy\_mountains;

****

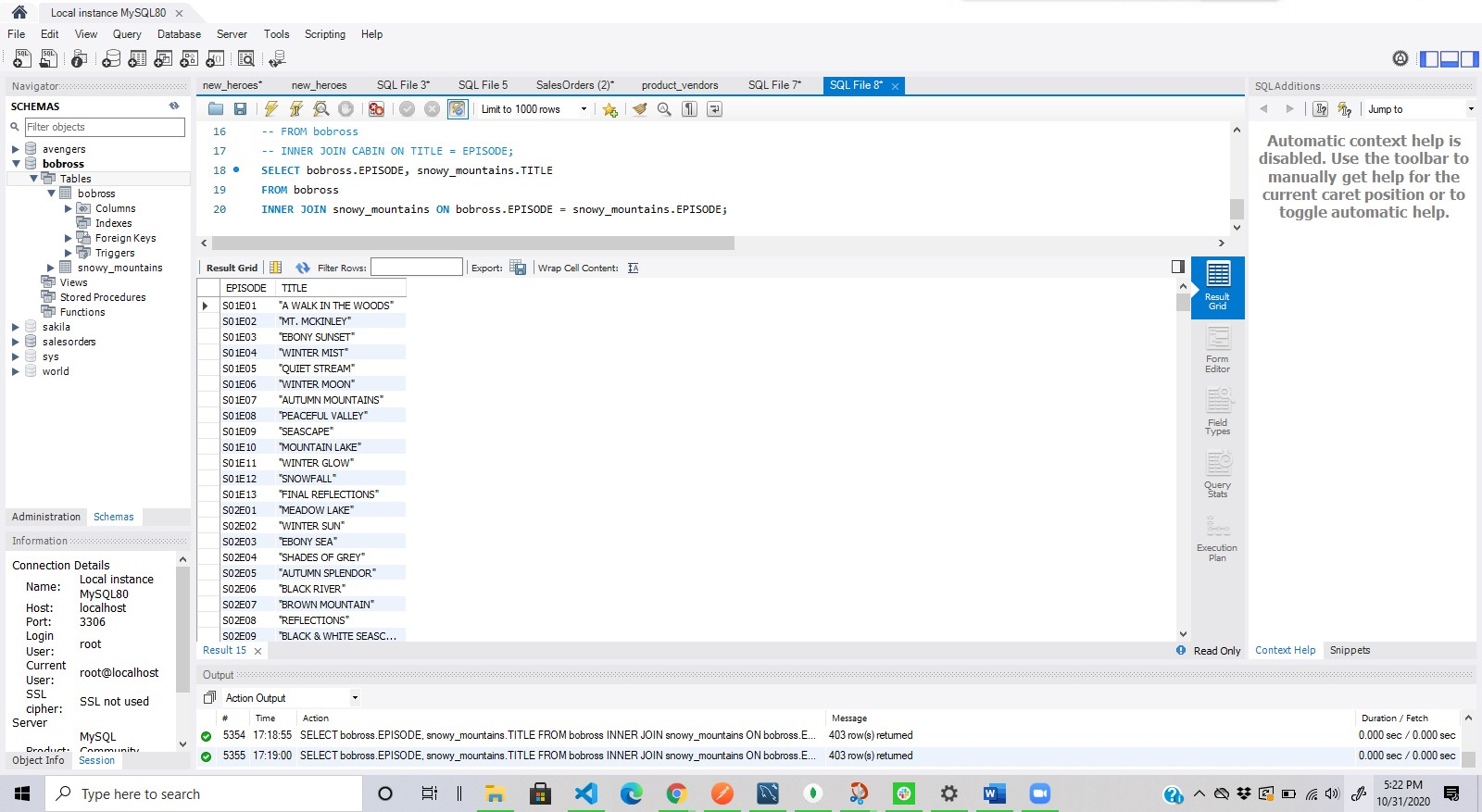
In the UNION query above, the command was meant to combine the result set of two SELECT statements I made, but just distinct values without repetition.

**6. ASSOCIATION & INNER JOIN.**

SELECT bobross.EPISODE, snowy\_mountains.TITLE

FROM bobross

INNER JOIN snowy\_mountains ON bobross.EPISODE = snowy\_mountains.EPISODE;



Here we are using INNER JOIN to select all of the titles from snowy\_mountain that have an association through the episode column with the episode column from the bobross table.

**7. ORDER OF OPERATIONS.**

SELECT bobross.EPISODE, snowy\_mountains.TITLE

FROM bobross

INNER JOIN snowy\_mountains ON bobross.EPISODE = snowy\_mountains.EPISODE

ORDER BY TITLE;

****

The first thing that happens in the query is the FROM and JOIN, then the SELECT limits which columns we need and finally ORDER BY ordered the output by alphabetical order.