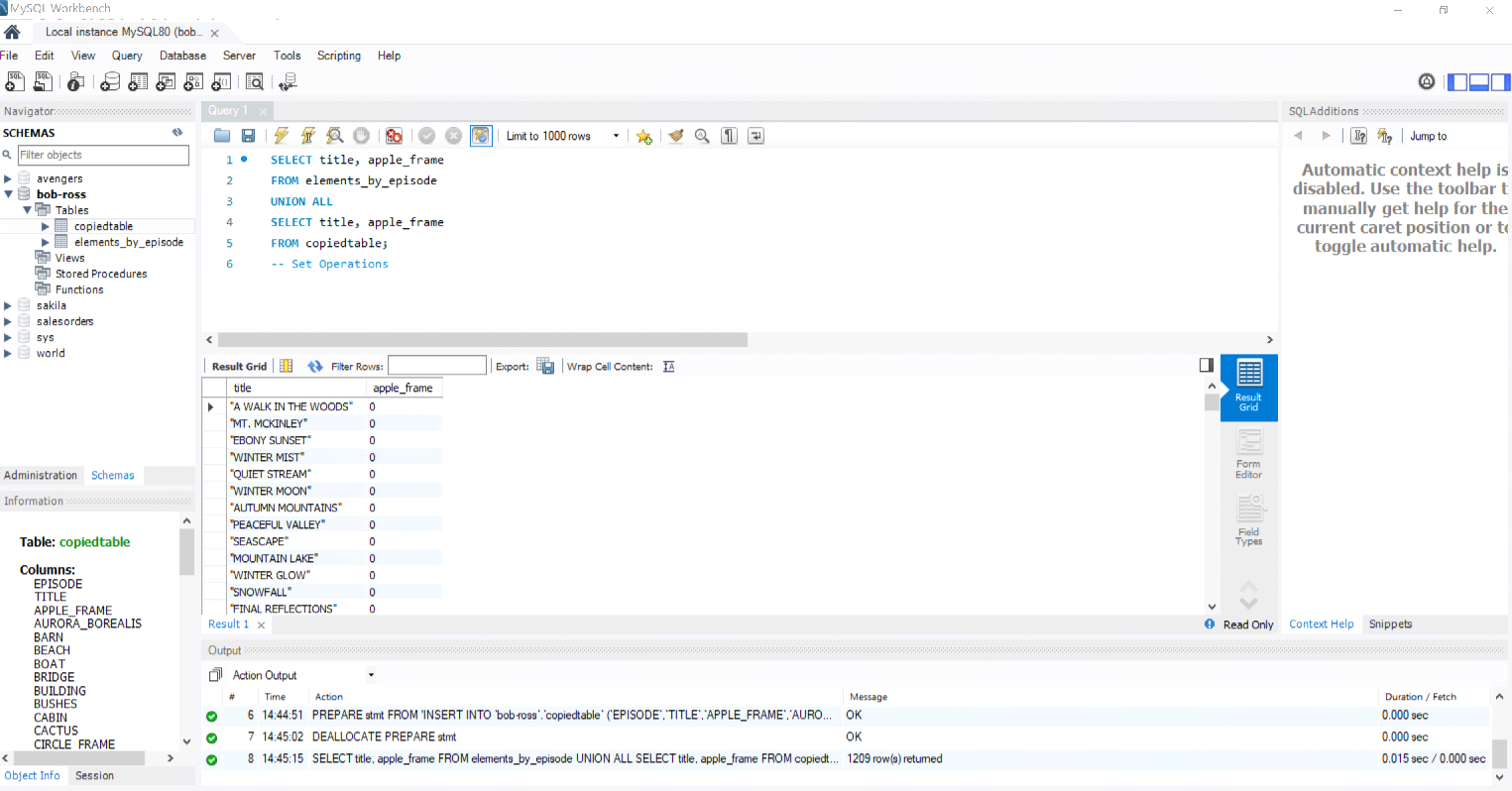
SET Operations

* Query:

SELECT title, apple\_frame  
FROM elementsbyepisode  
UNION ALL  
SELECT title, apple\_frame  
FROM copiedtable;



* I had just created a copied table from the same data. The set operator UNION takes similar data from both tables and joins them together.

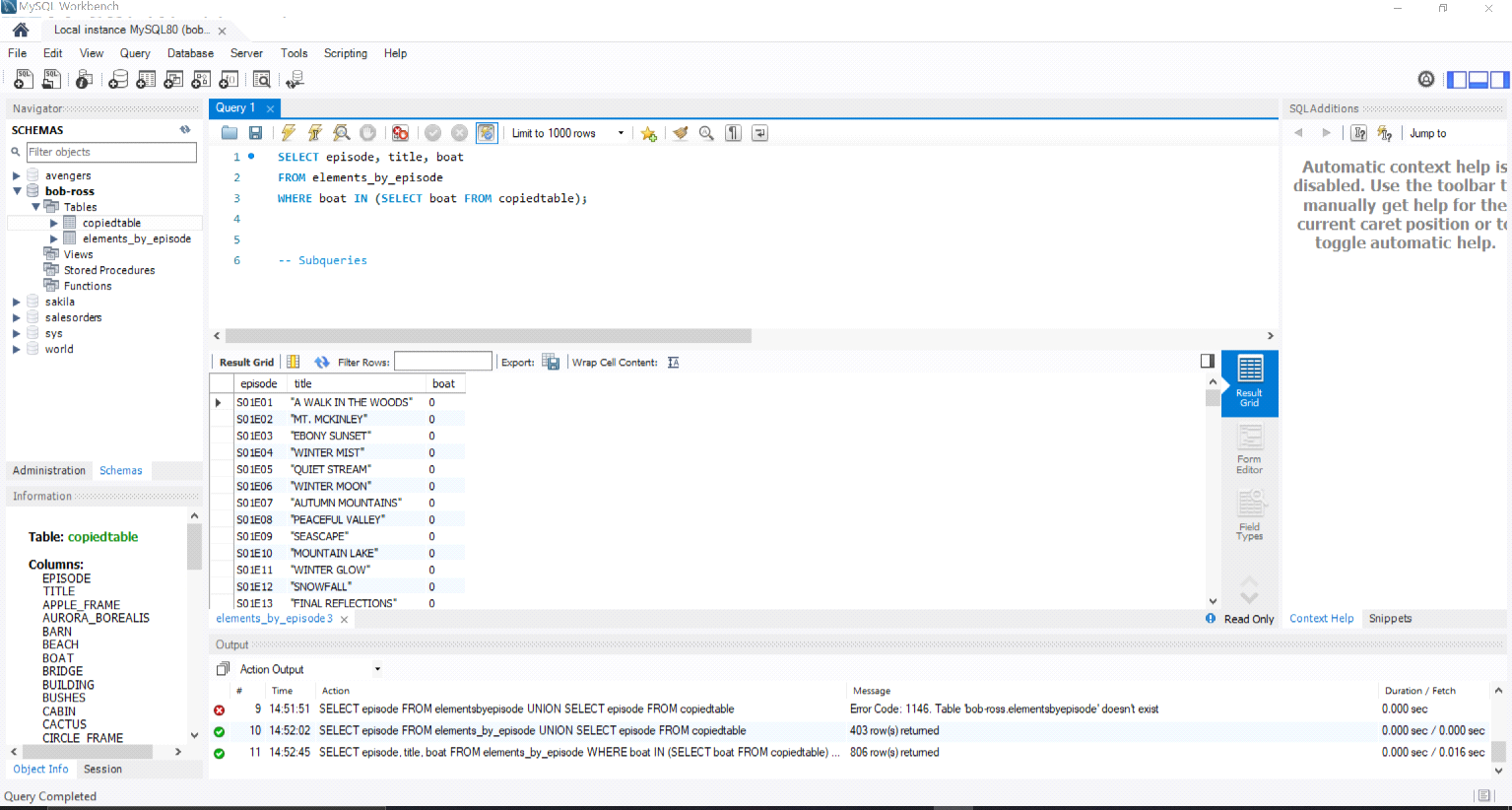
SubQuery

* Query:

SELECT episode, title, boat

FROM elementsbyepisode

WHERE boat IN (SELECT boat FROM copiedtable);



* The subquery is useful for pulling data with additional restrictions from other tables.

Order of Operations

* Query:

SELECT \*

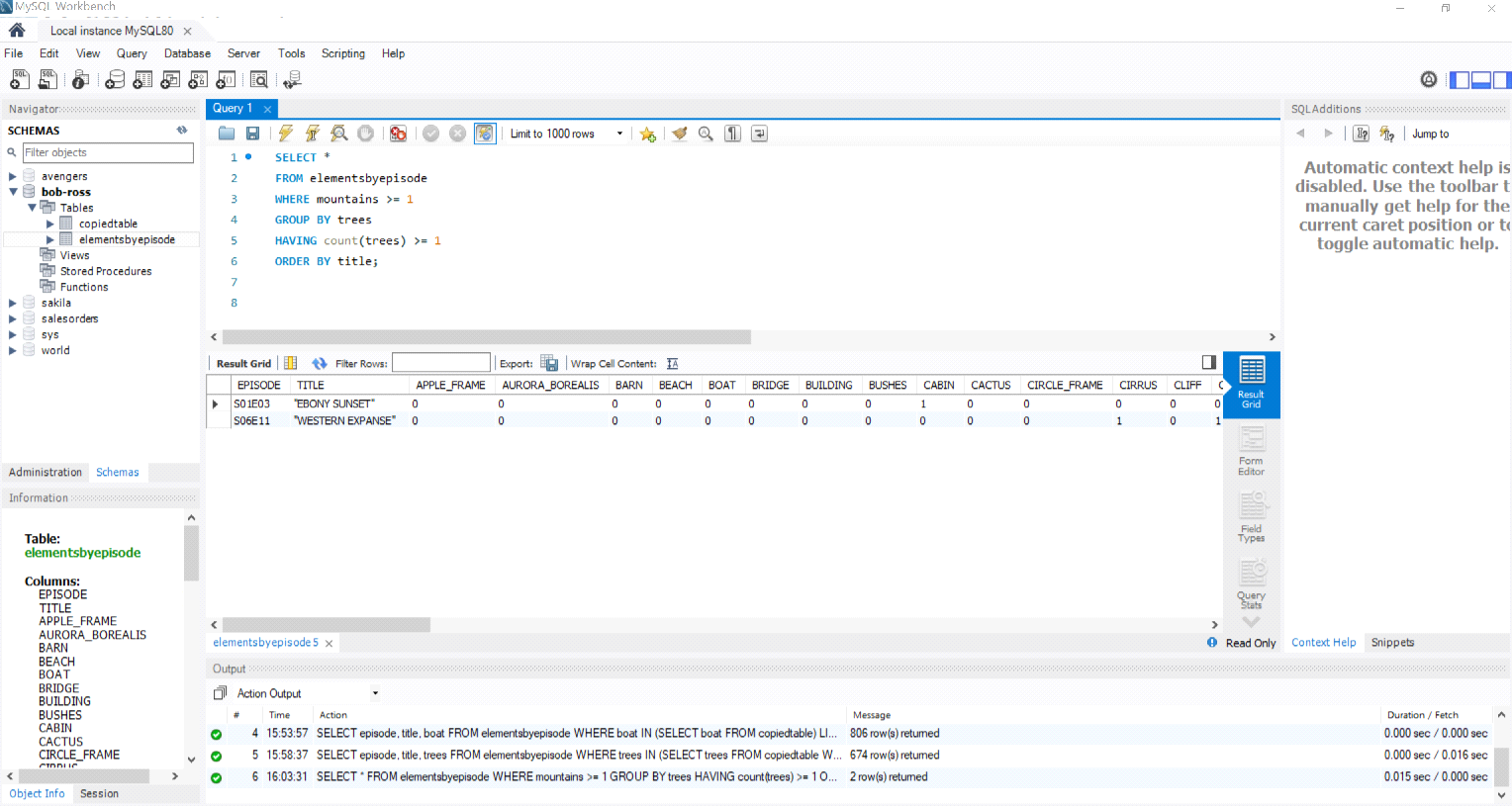
FROM elementsbyepisode

WHERE mountains >= 1

GROUP BY trees

HAVING count(trees) >= 1

ORDER BY title;

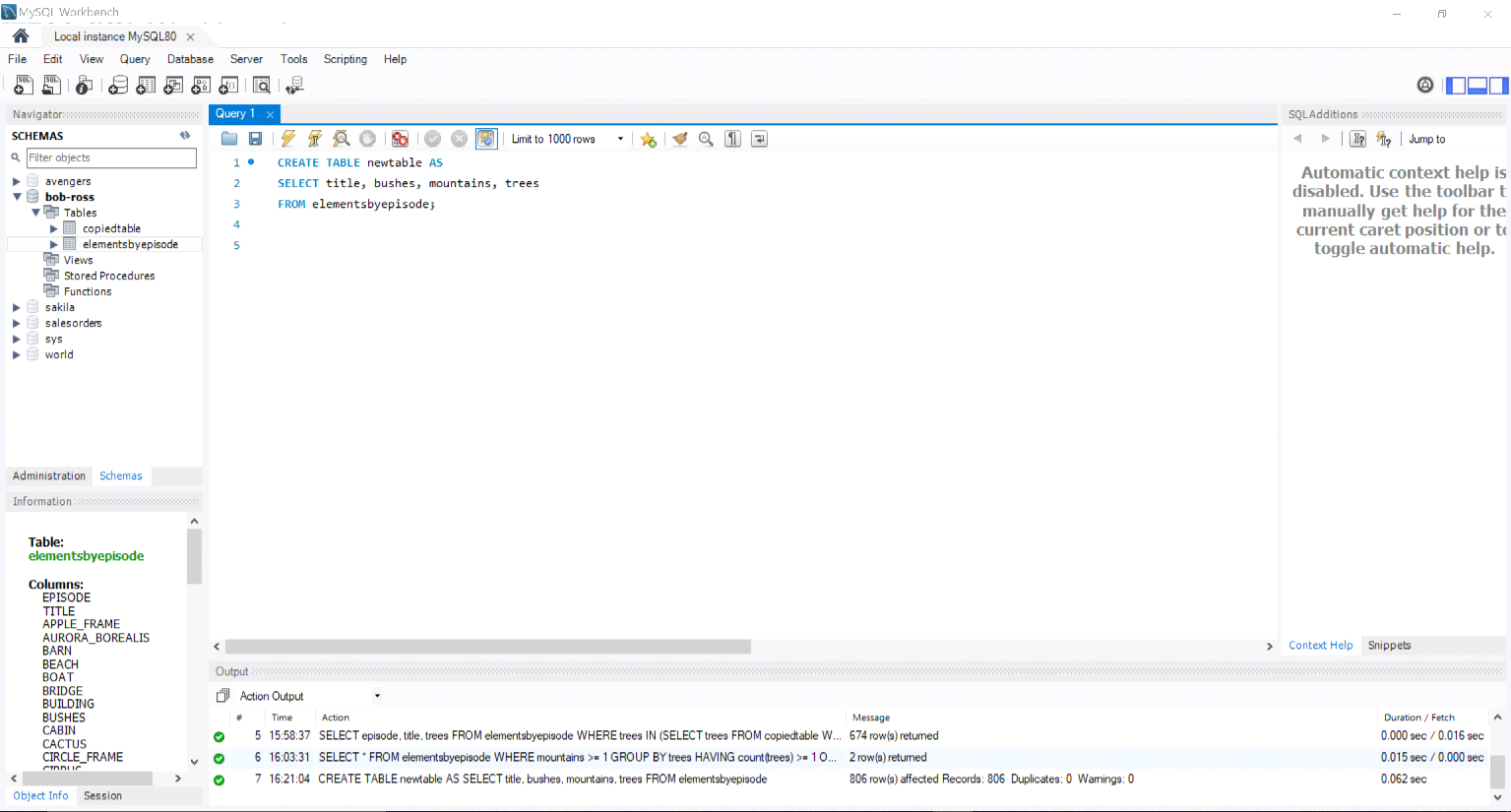


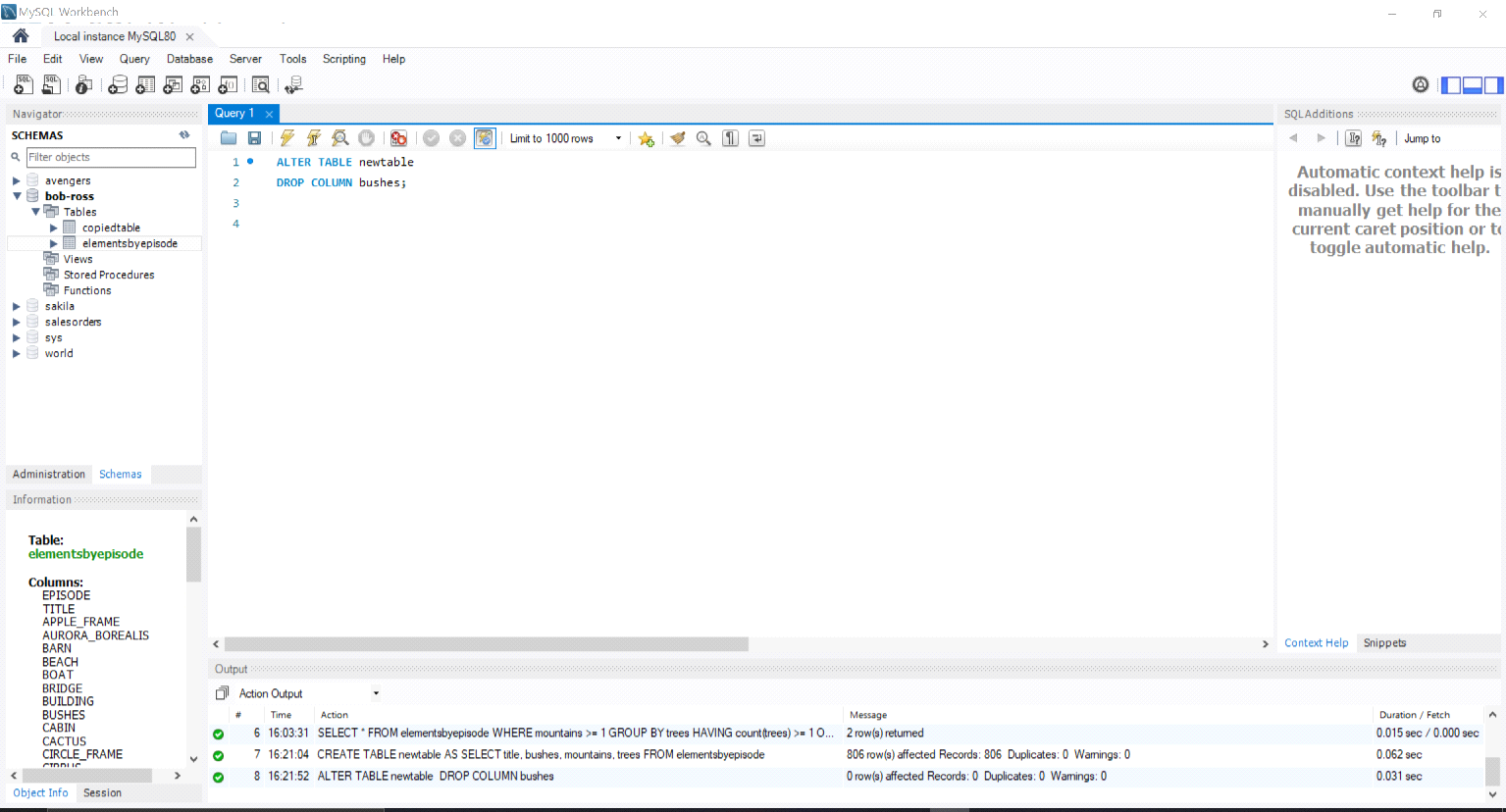
* Order of operations is the order that each part of the query is performed. Here I have the data pulled with each condition in the specific order required.

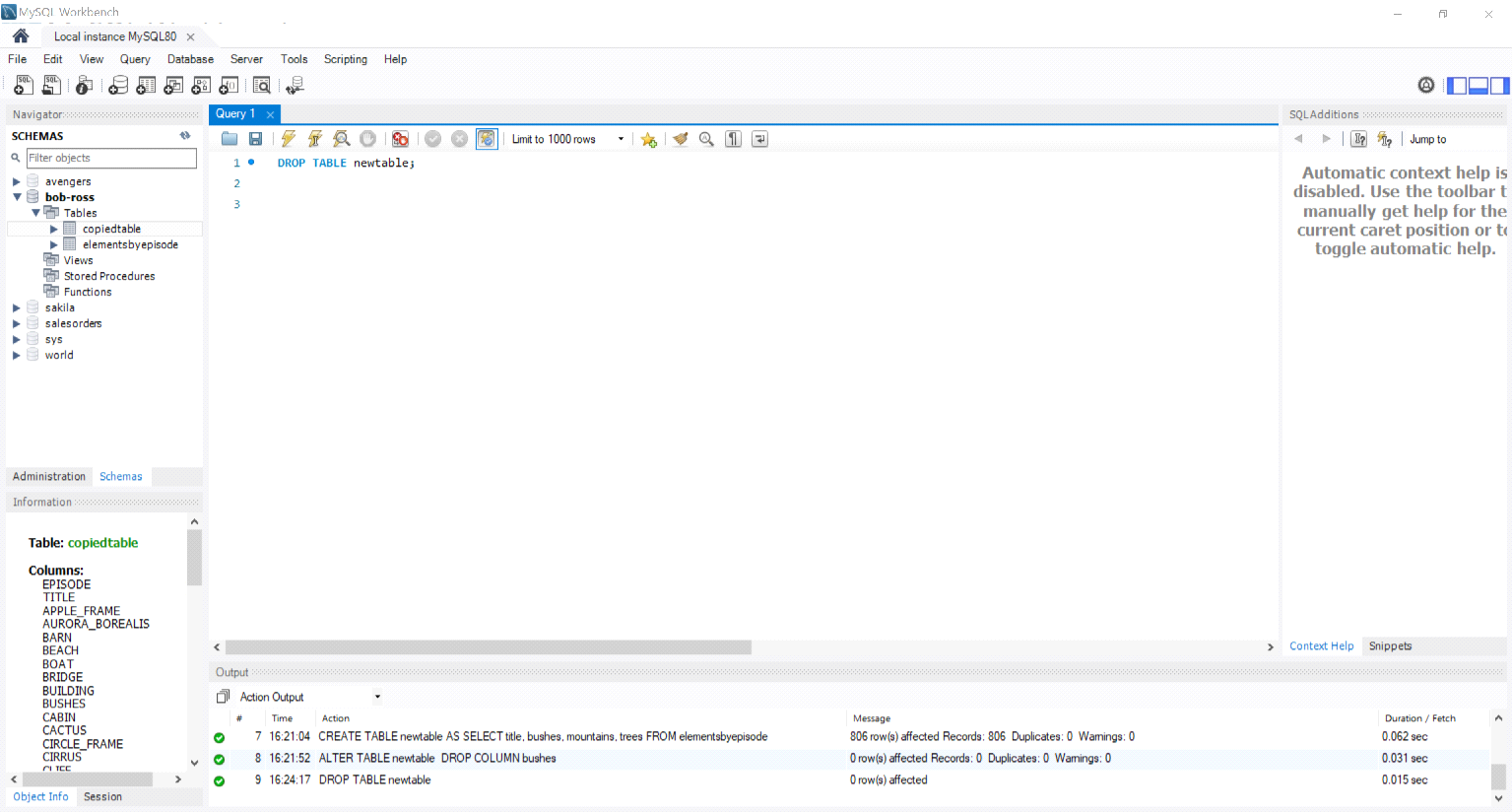
Create, Alter, Drop

* Querys:

1. CREATE TABLE newtable AS  
   SELECT title, bushes, mountains, trees  
   FROM elementsbyepisode;
2. ALTER TABLE newtable   
   DROP COLUMN bushes;
3. DROP TABLE copiedtable;





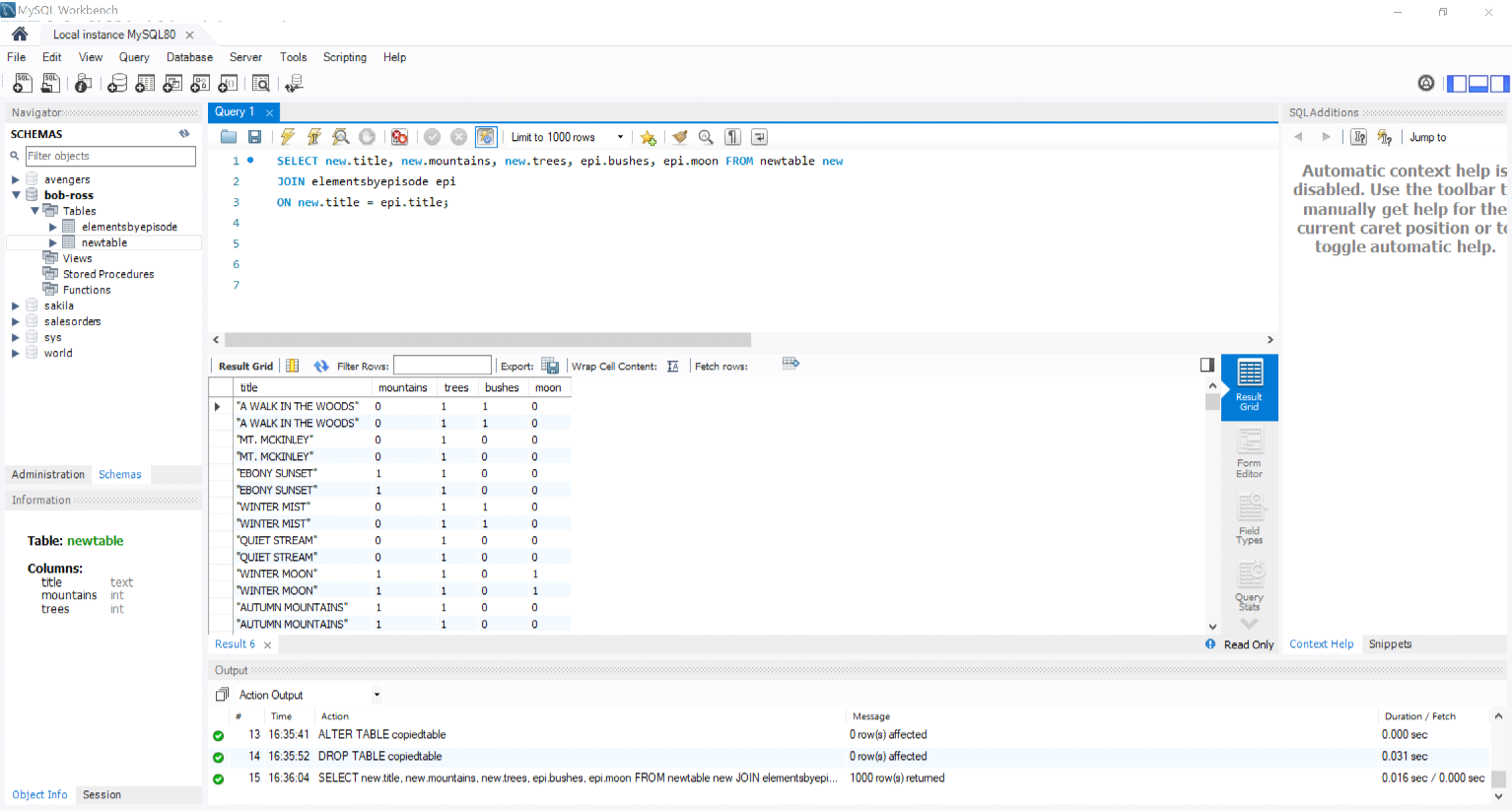


* Here, I created a new table from the bob ross data. Altered the table by dropping a column. Then dropped the table all together.

ASSOCIATIONS

* Query:

SELECT new.title, new.mountains, new.trees, epi.bushes, epi.moon FROM newtable new  
JOIN elementsbyepisode epi  
ON new.title = epi.title;



Joining Mutiple

* Query:

SELECT epi.title, epi.boat, epi.cactus, new.mountains, new.trees

FROM elementsbyepisode epi

JOIN newtable new

ON epi.title = new.title

JOIN anothernewtable anew

ON epi.title = anew.title;

