**FSW-140: BOB ROSS**

**Set Operations**

**Query:**

SELECT Episode, Title, Barn, Boat FROM mysecondTable

UNION ALL

SELECT Episode, Title, Barn, Beach FROM elements\_by\_episode

ORDER BY Episode;

**Expected output:**

The expected output will be double the rows as the tables have duplicated. In UNION ALL the output shows duplicated rows. Union takes information from two different tables and combines them into One as the output.

**Screenshot:**

A screenshot of a computer

Description automatically generated with medium confidence

**Subqueries:**

**QUERY:**

SELECT TITLE, EPISODE FROM elements\_by\_episode

WHERE TITLE = (

SELECT TITLE FROM elements\_by\_episode

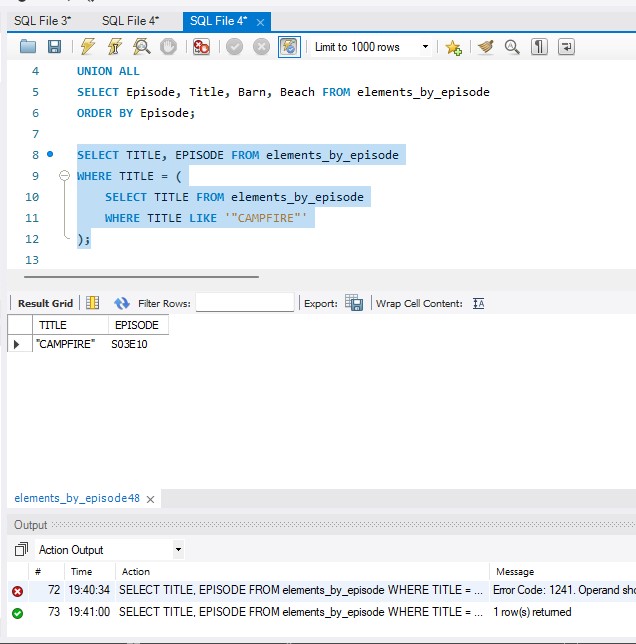
WHERE TITLE LIKE '"CAMPFIRE"'

);

**Expectations:**

The expected output is one row with the Title and Episode where the title is campfire. In subqueries the inner query (the one in parenthesis) executed first then the outer query executes giving a narrowed down query

**Screenshot:**



**Order of operation of queries:**

**Query:**

**SELECT TITLE, EPISODE FROM elements\_by\_episode**

**WHERE TREES >= 1**

**ORDER BY TITLE ASC**

**LIMIT 10;**

**Expected results:**

The expected result is 10 rows that show the title and episode with trees limited to 10 rows. The order of operations the query is executed by is it takes is FROM, WHERE, SELECT, ORDER BY and Then LIMIT. So it retrieves data from the table using the WHERE clause then it SELECTS the columns to return ORDERs the columns BY either ASC/DESC then executes the LIMIT clause.

**Screenshot:**

A screenshot of a computer

Description automatically generated with medium confidence

**Creating, altering, and dropping tables:**

**Query:**

ALTER TABLE mysecondtable

ADD WEATHER varchar(50);

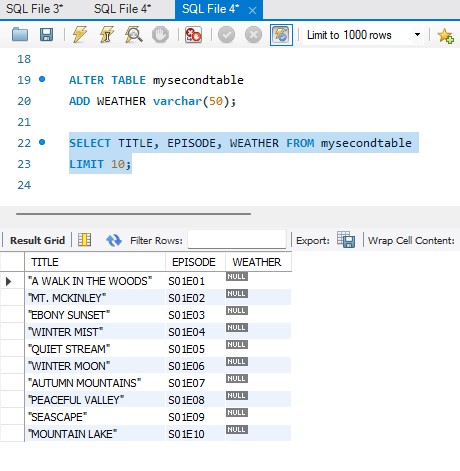
SELECT TITLE, EPISODE, WEATHER FROM mysecondtable

LIMIT 10;

**Expected results:**

Here I altered mysecondtable by adding a column of weather to it then I used a second query to show the column in the altered table

**Screenshot:**

****

**Associations:**

**Query:**

ALTER TABLE elements\_by\_episode

ADD COLUMN ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY;

ALTER TABLE mysecondtable

ADD COLUMN ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY;

ALTER TABLE mysecondtable MODIFY ID INT NOT NULL;

ALTER TABLE mysecondtable DROP PRIMARY KEY;

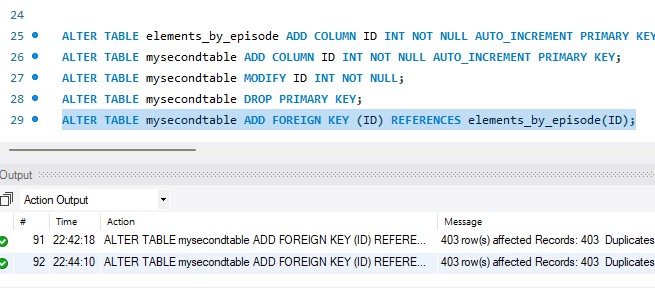
ALTER TABLE mysecondtable

ADD FOREIGN KEY (ID) REFERENCES elements\_by\_episode(ID);

**Expected results:**

I had to look this one up and what I found is that associations is what tables have in common together like an ID. Then you can query the data into a table of the common Column. By Adding PRIMARY and FOREIGN KEYS I was able to make the tables have an Association.

Screenshot:



**Joins and multiple table joins:**

**Query:**

SELECT elements\_by\_episode.FIRE, mysecondtable.CABIN FROM elements\_by\_episode

JOIN mysecondtable

WHERE mysecondtable.BUSHES = 1;

**Expected results:**

Using Join you can take data from two tables and join them together. The expected output would be of from both tables.

Screenshot:

