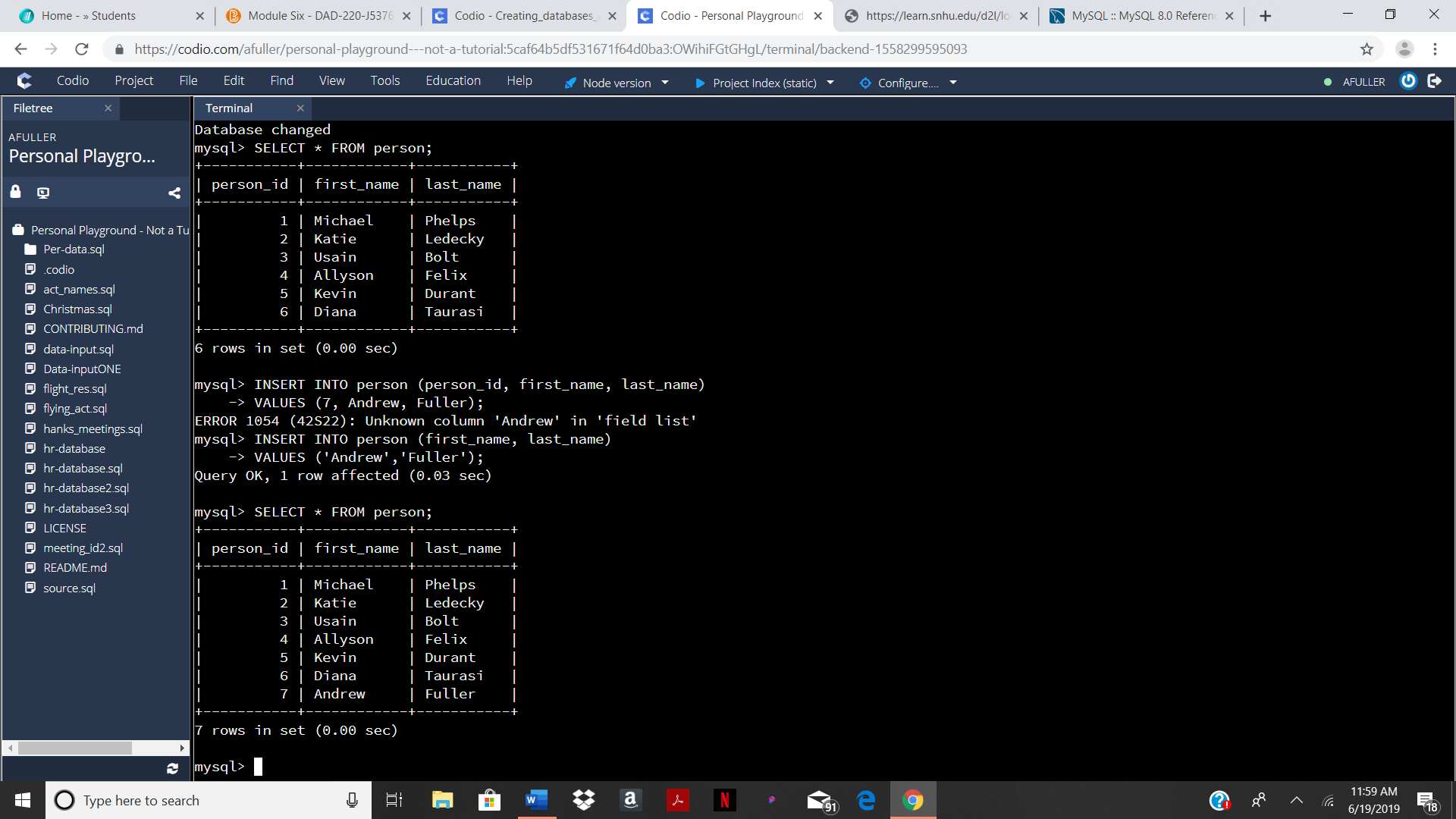
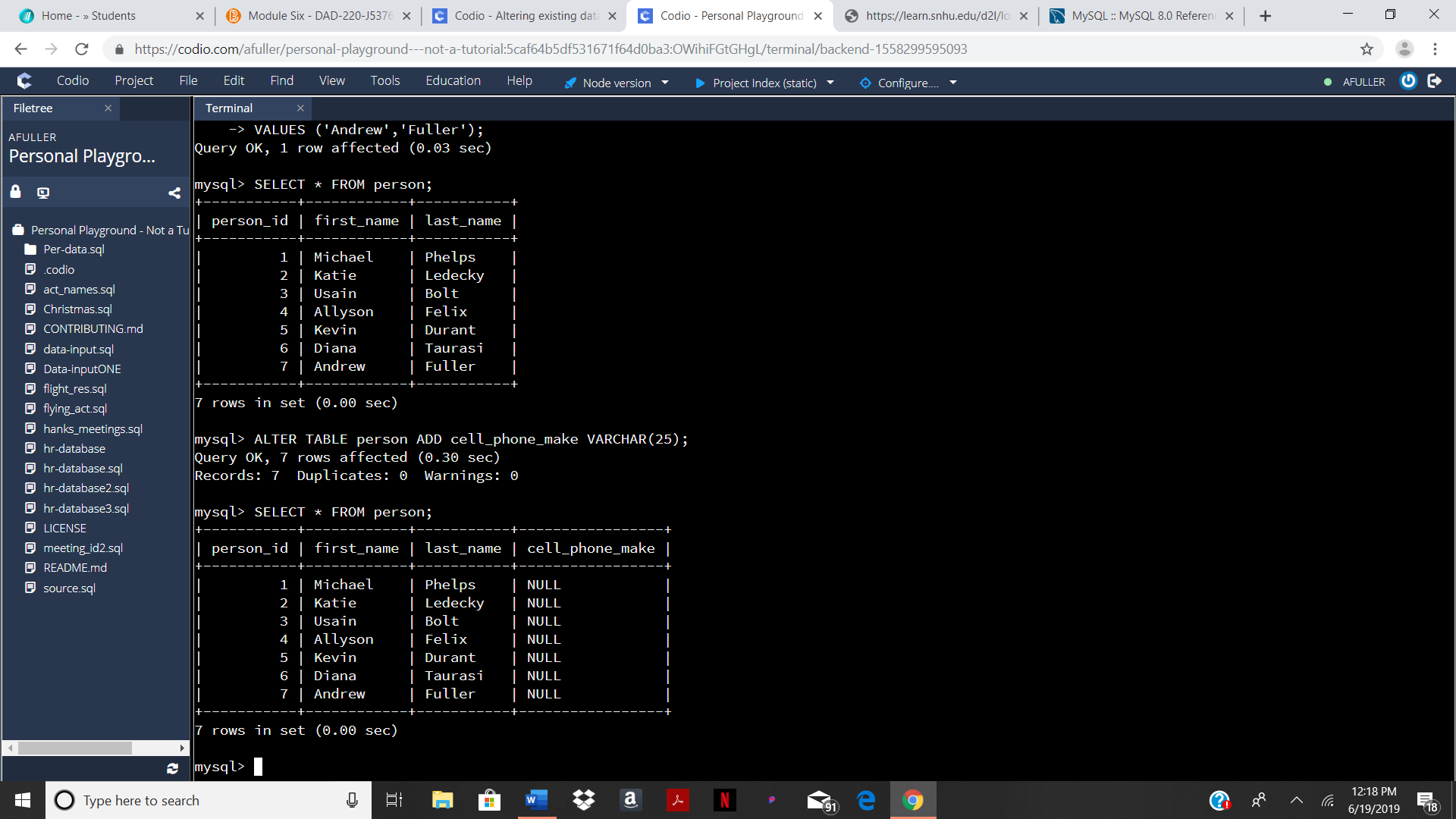
FINAL MILESTONE

TASK 1



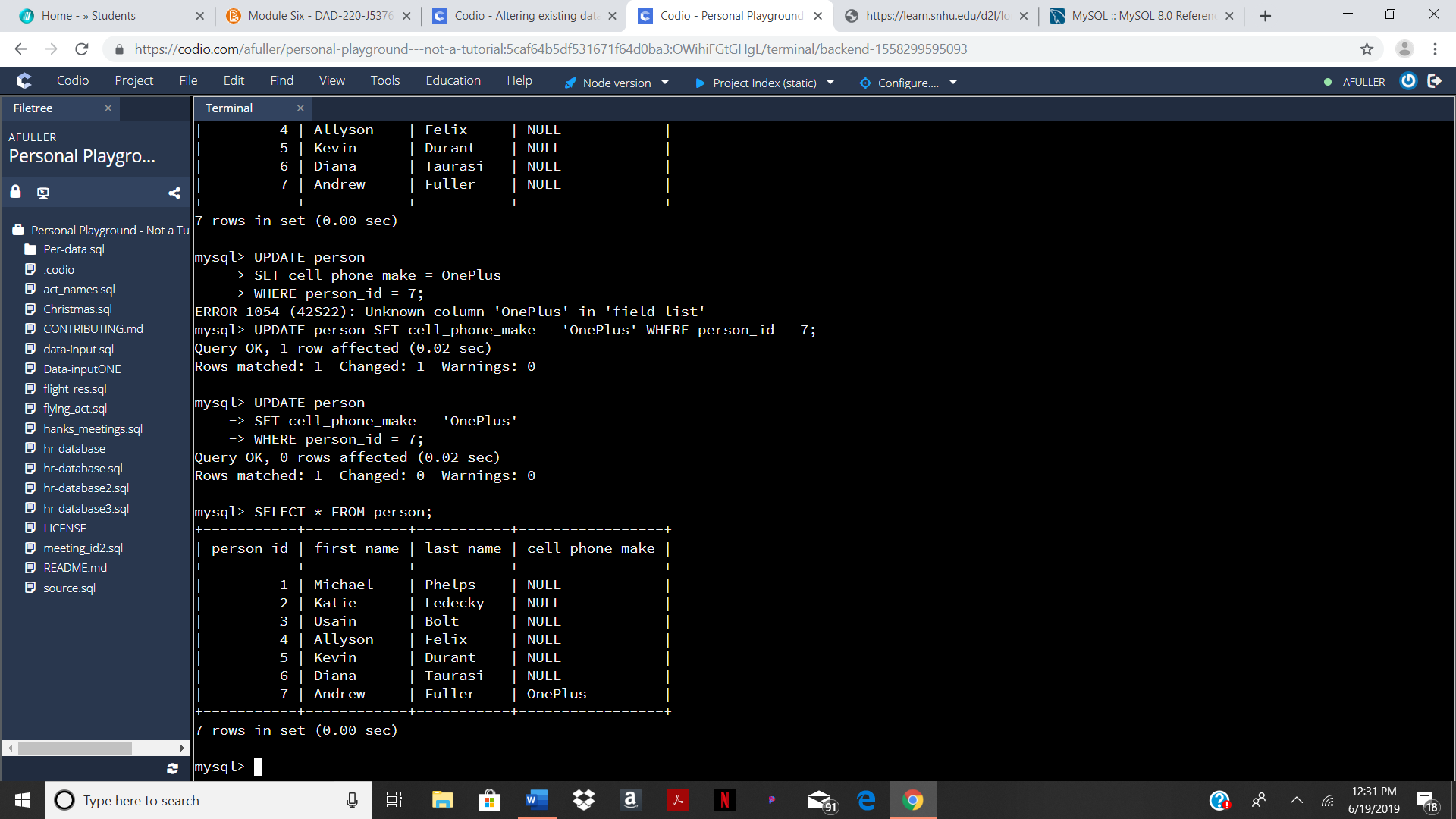
A simple INSERT INTO statement was used in order to input my name into the *person’s* table. The line of code begins with the INSERT INTO syntax fallowed by the name of the table, the desired columns, and the VALUES that will be entered into those columns.

TASK 2



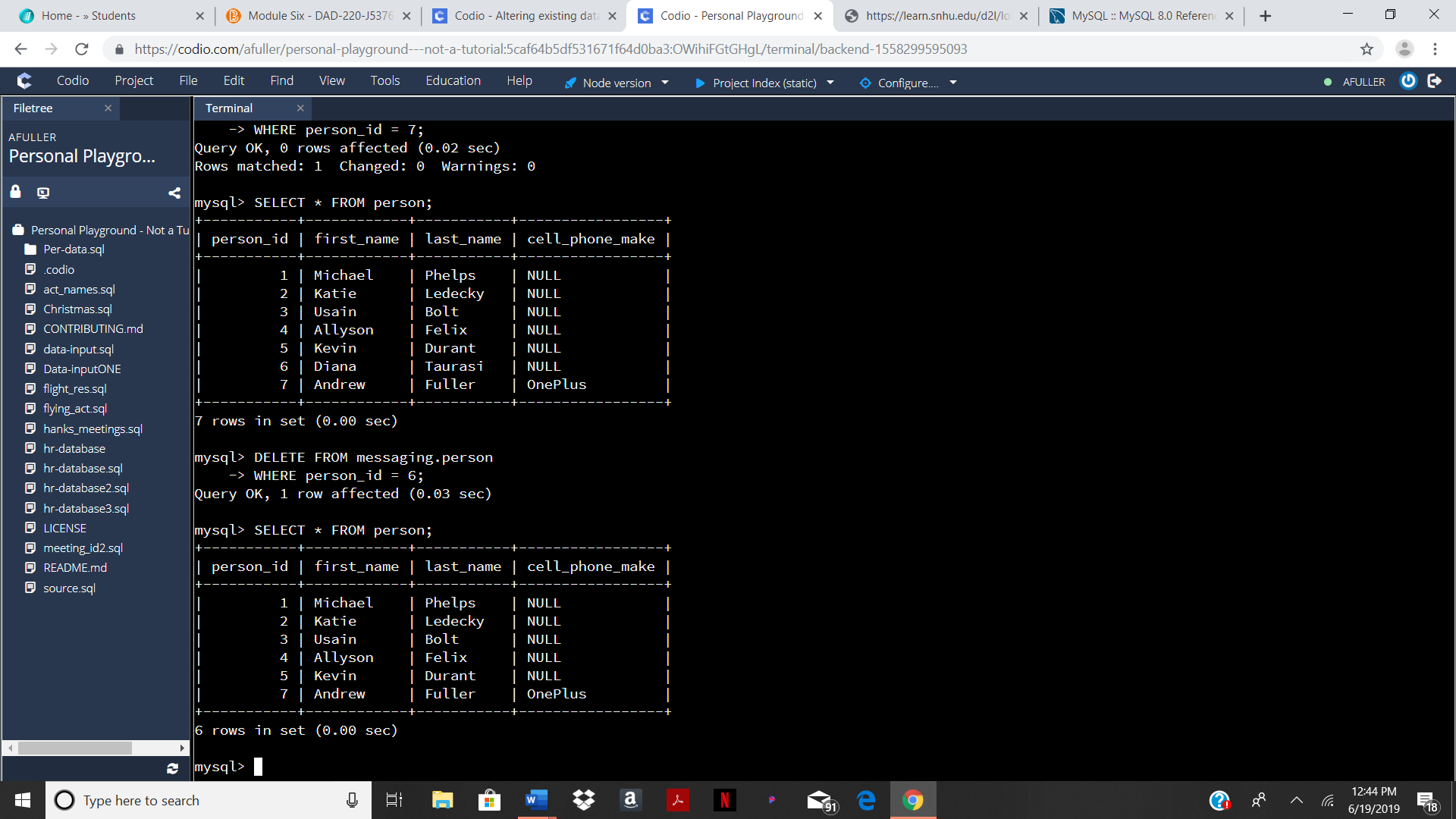
In order to add a column to the *person’s* table I used an ALTER TABLE statement. The name of the table must fallow ALTER TABLE in order to tell the computer what it is altering. In this line of code, a column is being added to the table. To do this, the syntax ADD is placed right before the column name, data type, and constraints that are being added.

TASK 3



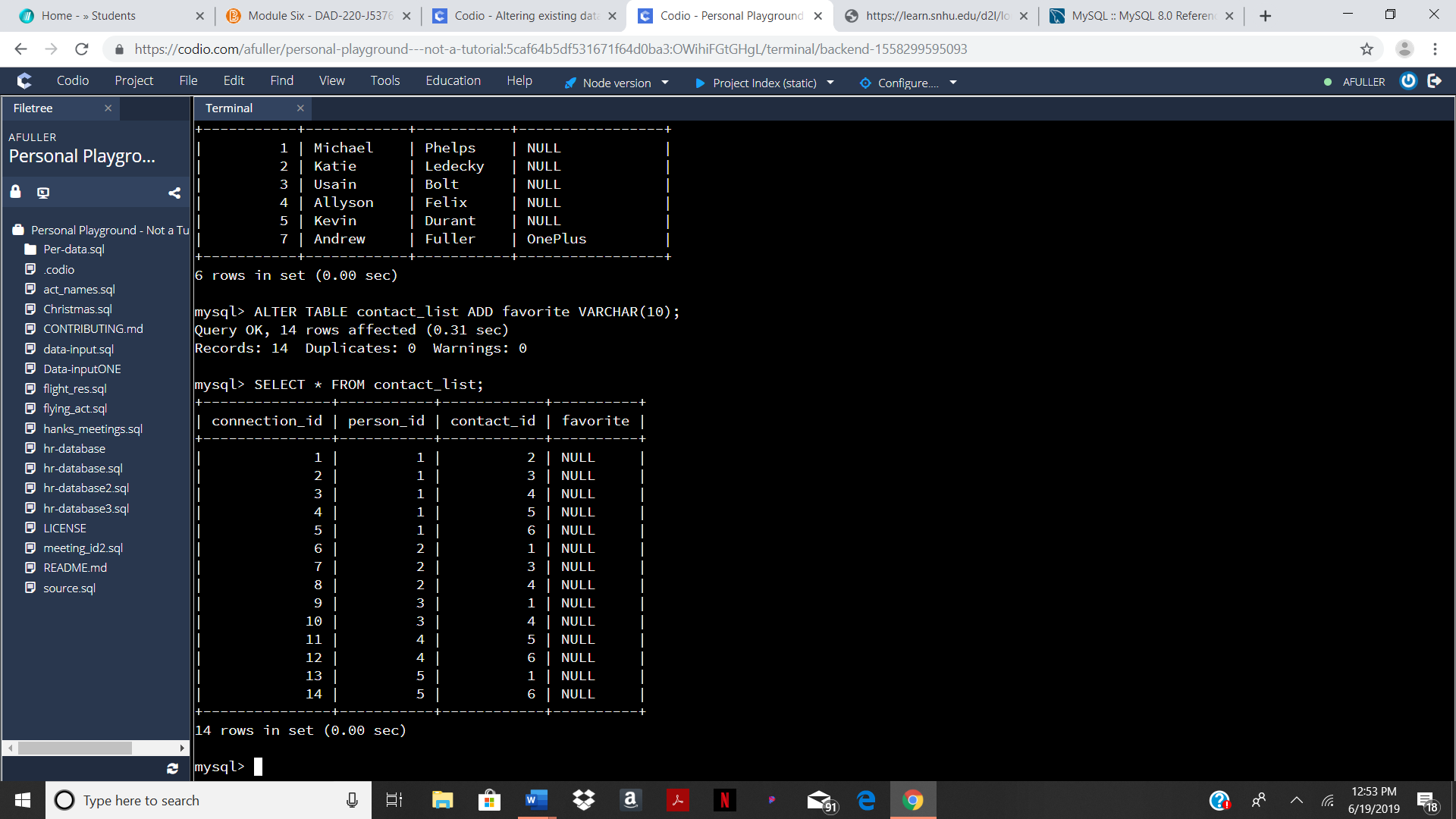
In this task the *cell\_phone\_make* column was updated to output “OnePlus” for my name. To accomplish this, I wrote an UPDATE statement that SET *cell\_phone\_make* to “OnePlus” WHERE *person\_id* was equal to 7. This changed only the data in the *cell\_phone\_make* column for my name because my *person\_id* is 7.

TASK 4



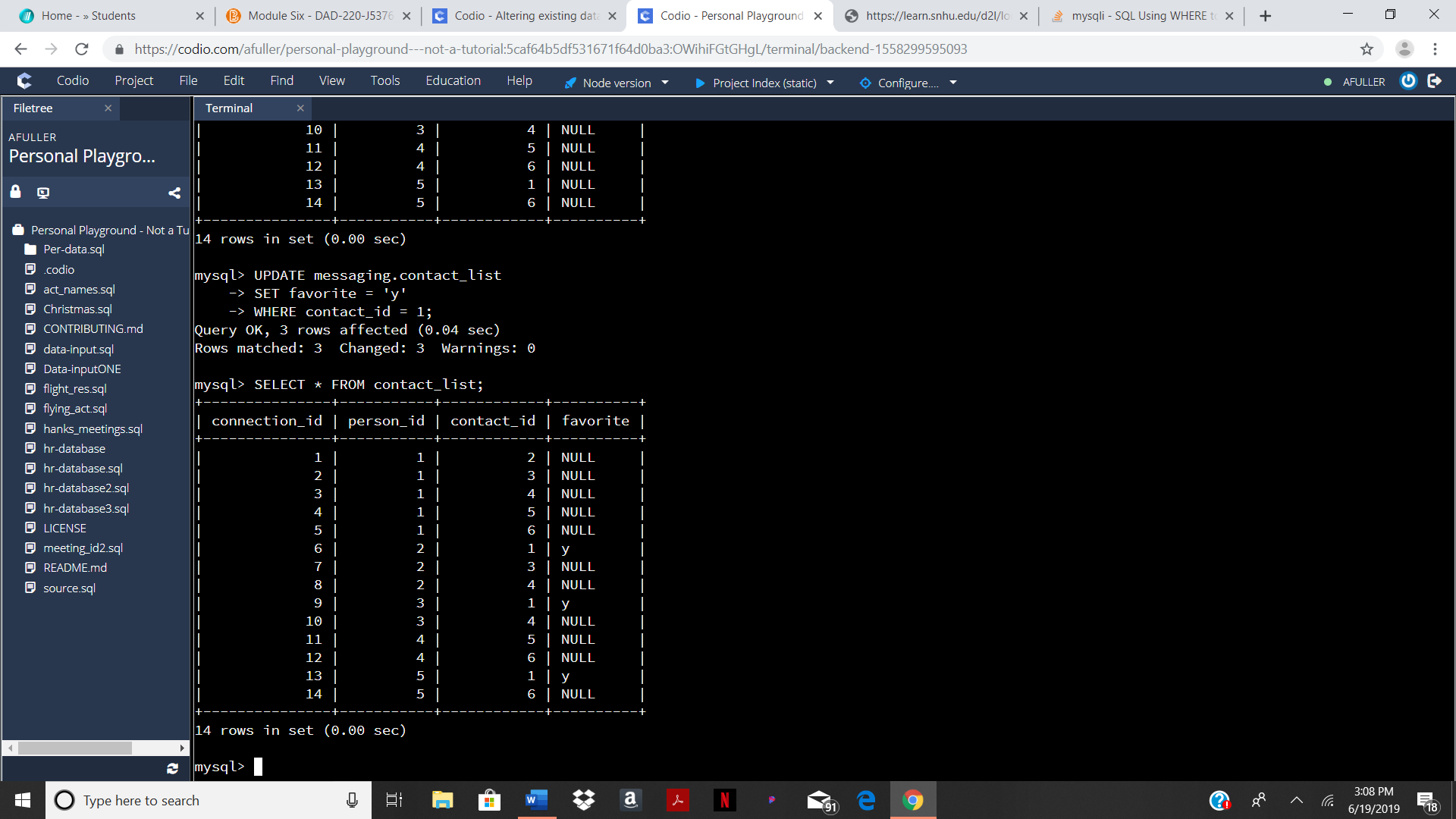
To delete data from a table a DELETE FROM statement can be used. To write this statement, two pieces of information are needed: which table is the information being deleted from and what information within that table is being deleted. “Diana Taurasi” was deleted from this table by using the WHERE clause to delete the *person\_id* that was equal to 6.

TASK 5



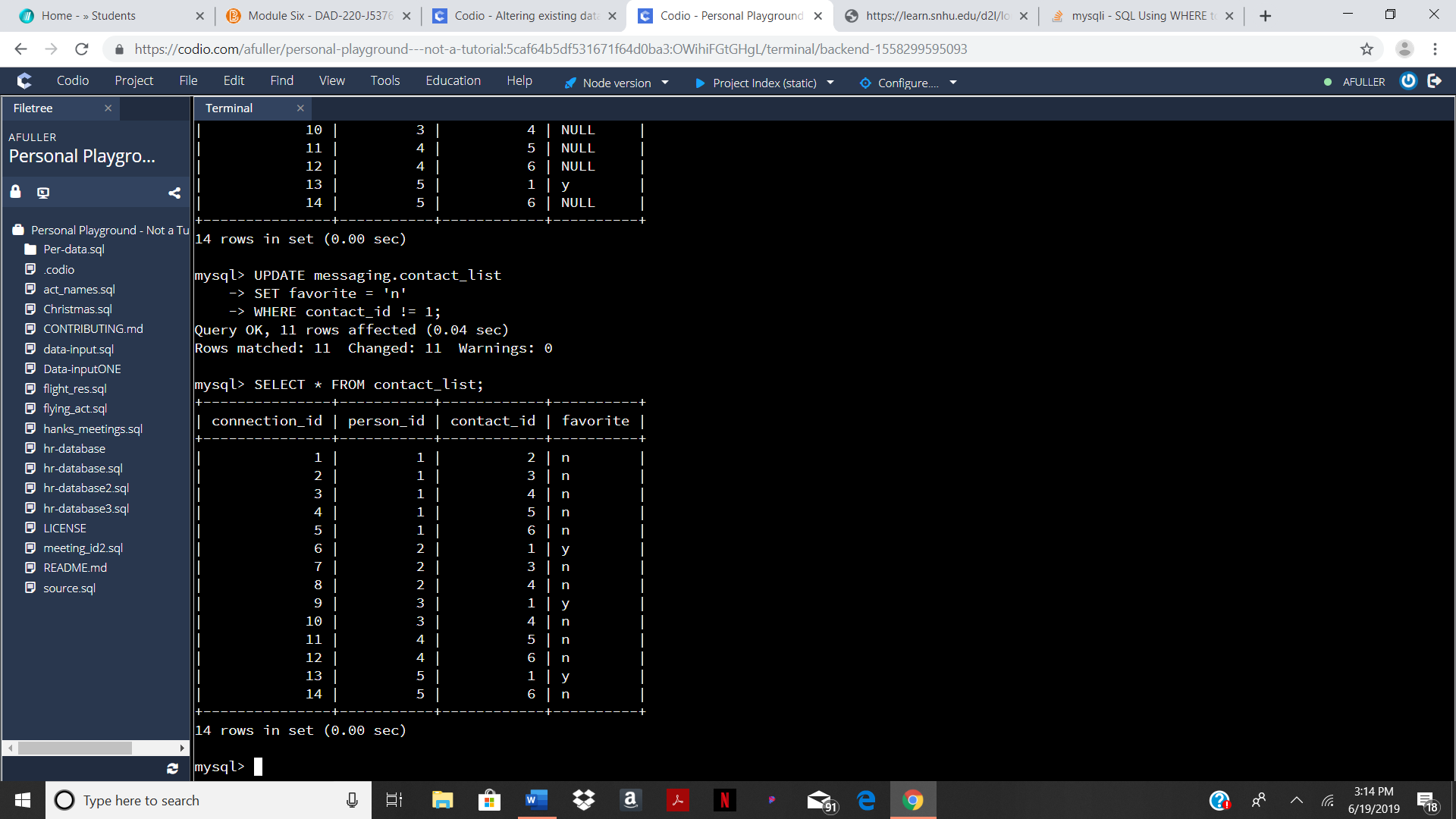
Just like in task two an ALTER TABLE statement is needed to add a column to a table. Once the table is selected that the column will be added to, the syntax ADD fallowed by the column name, data type, and constraints need to be typed in.

TASK 6



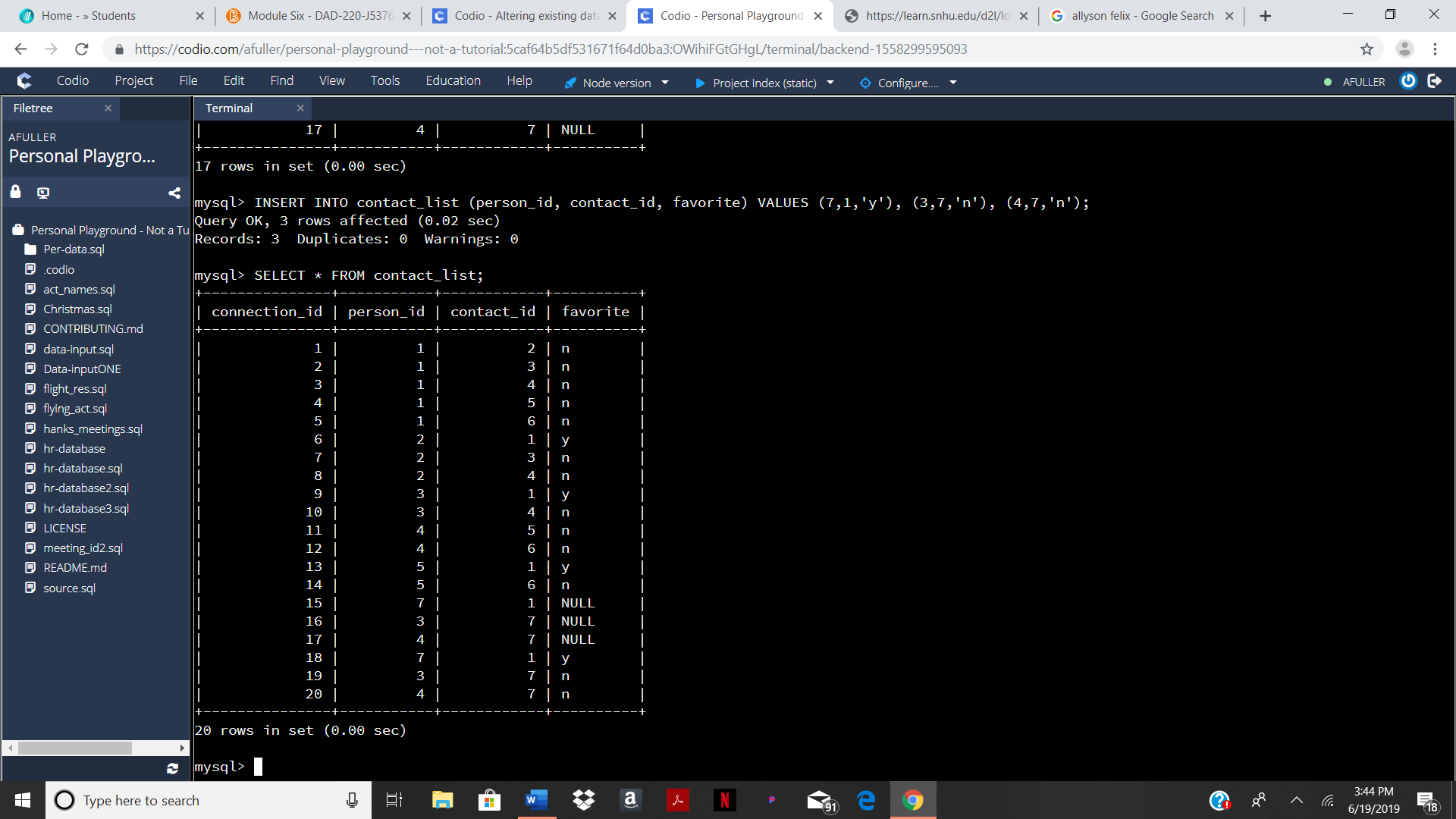
In this task an UPDATE statement is used to change the value in the favorite column to “y” if the *contact\_id* was equal to 1. The value in the favorite column remained null for the columns that did not have a 1 in the *contact\_id* column.

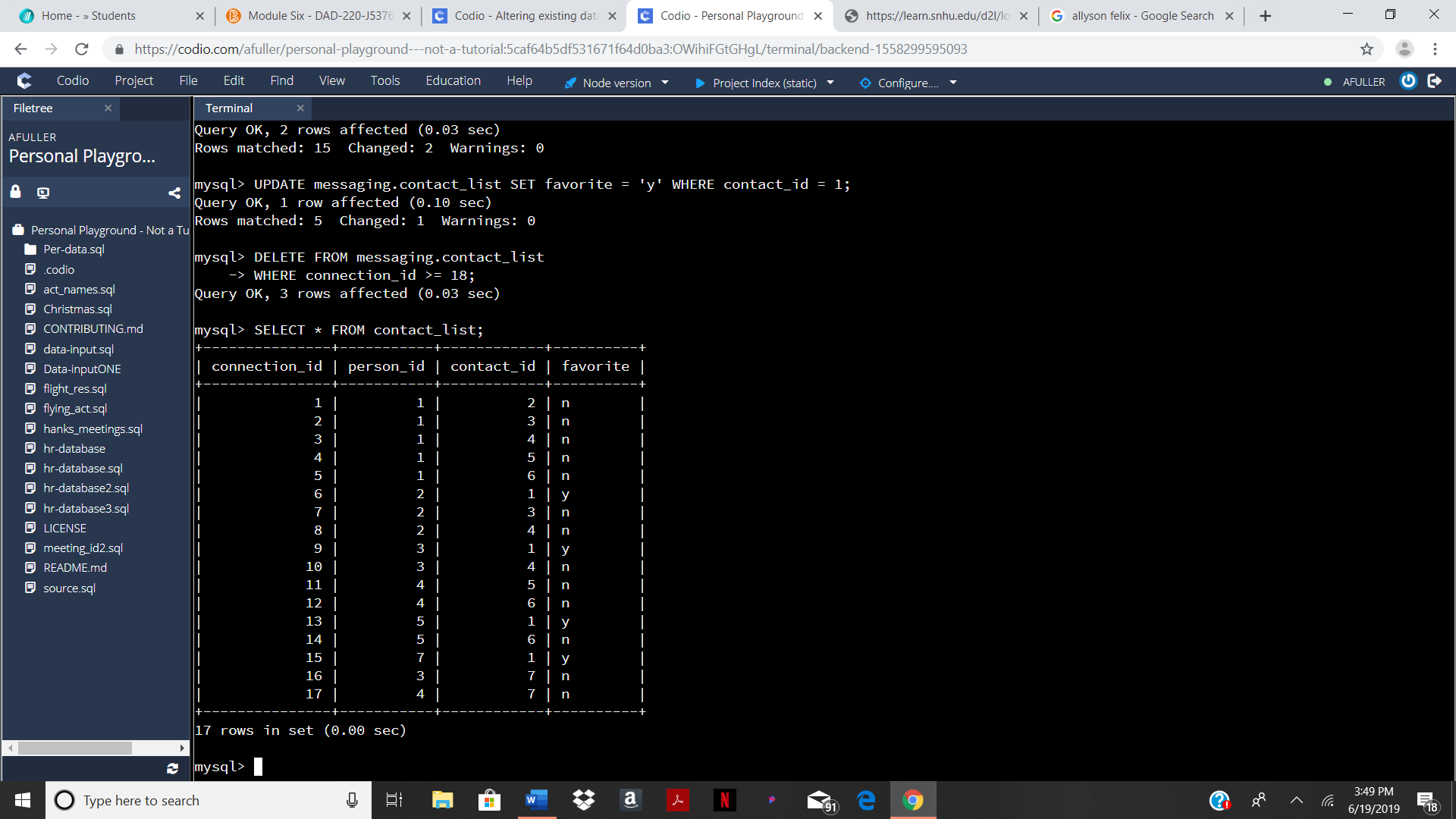
TASK 7



This task can be done by using almost the same statement as the previous task. There are only two alterations that must be done. Instead of the updating the *favorite* column with “y” WHERE *contact\_id* is equal to 1, you must change the “y” to “n” and update the data WHERE *contact\_id* does not equal 1.

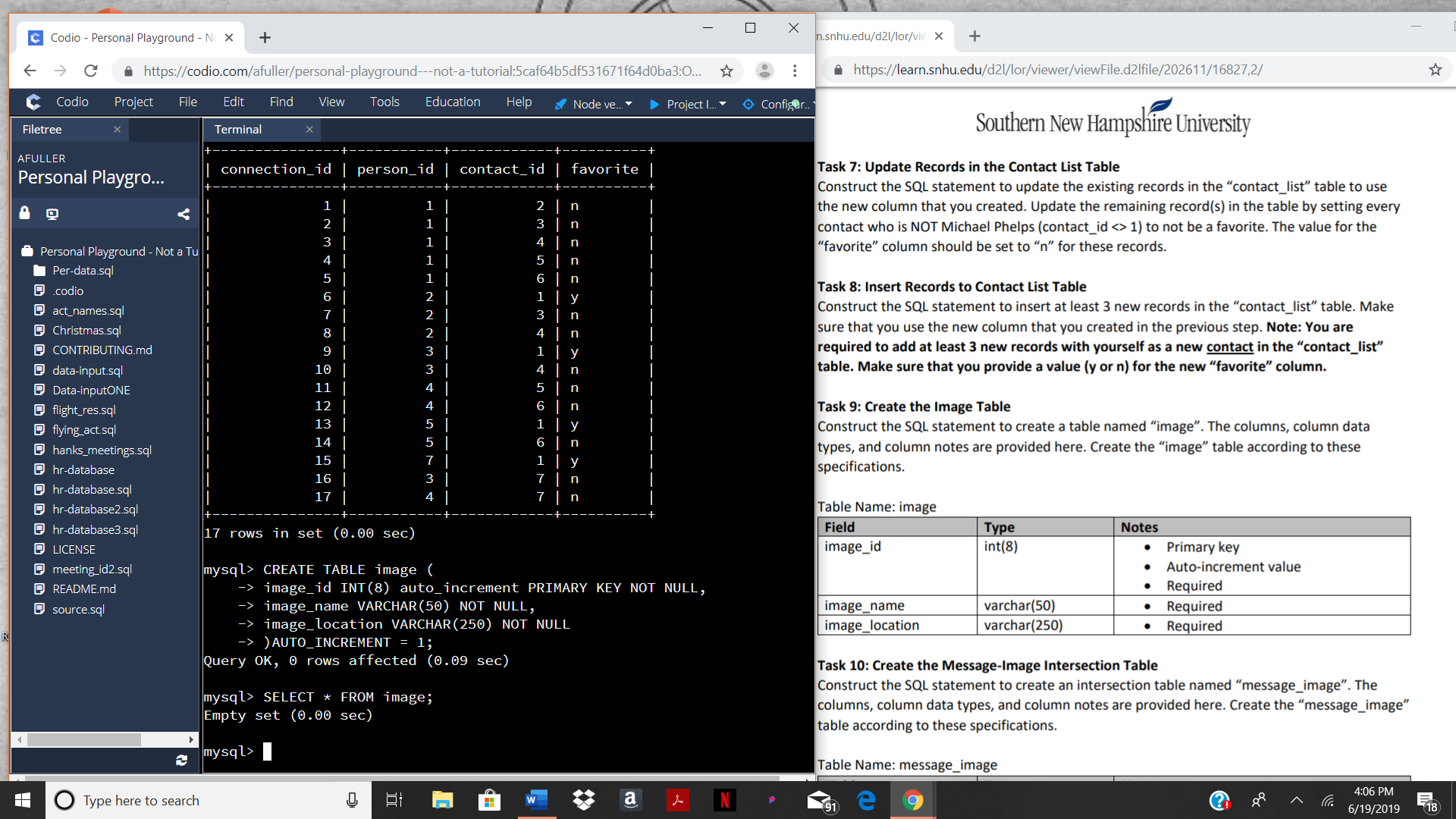
TASK 8

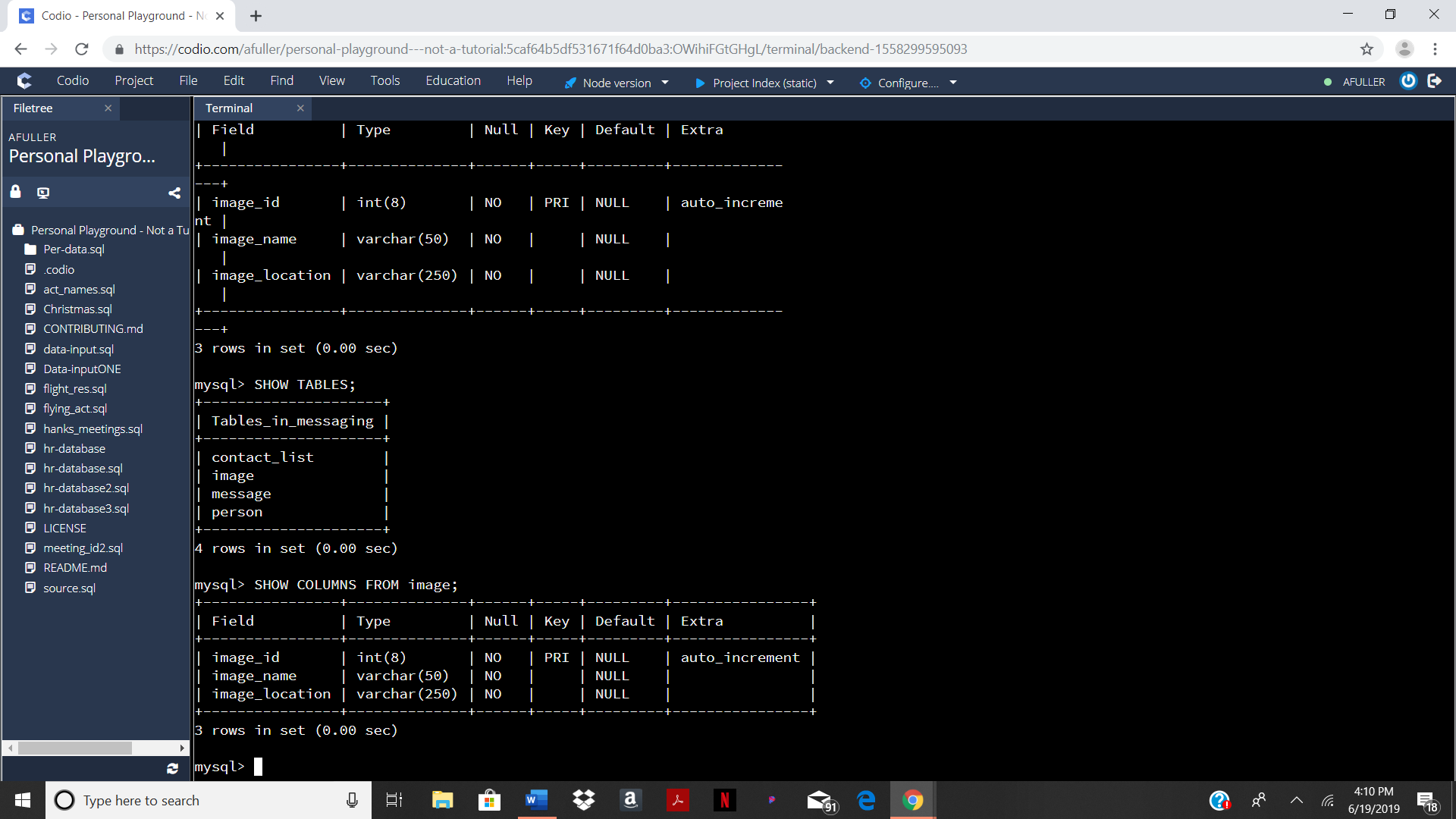




Similar to task one, an INSERT INTO statement was created to add *connection\_id’s* 15 – 17.

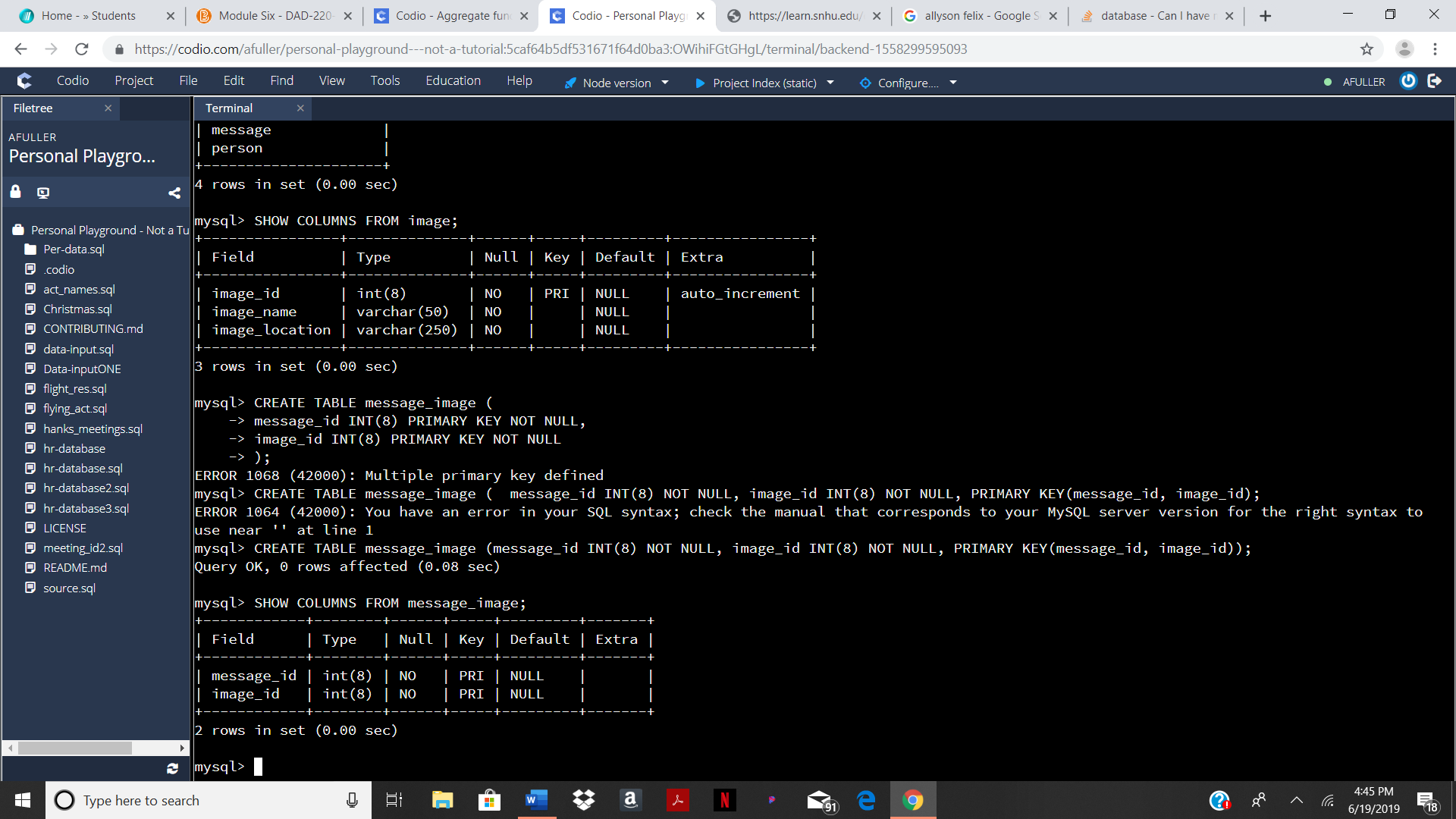
TASK 9





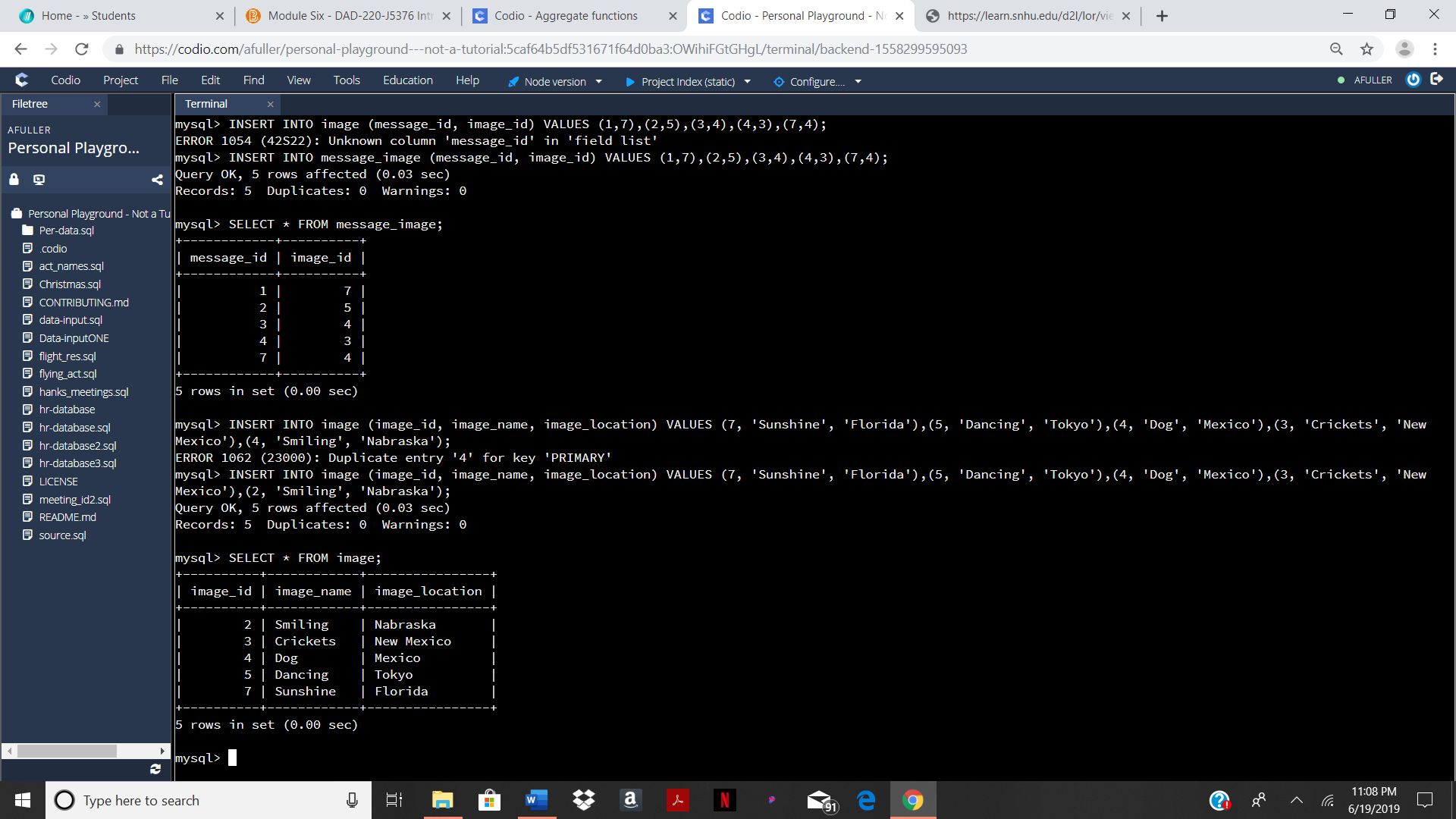
For this task a CREATE TABLE statement is required. In this statement the table is named, data types are given, and constraints are put into place.

TASK 10



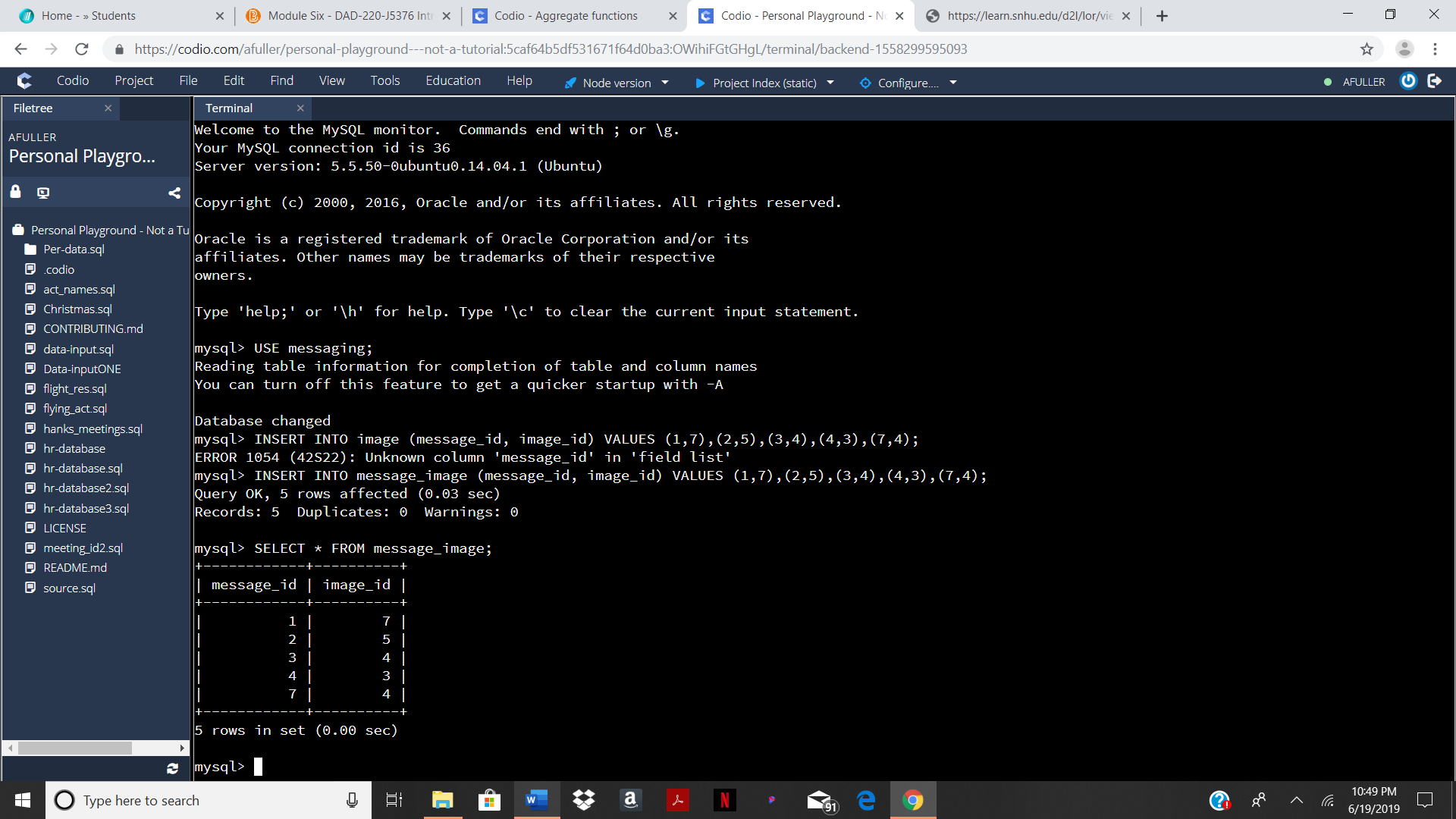
In this task there are two PRIMARY KEY’s which have a different syntax then a statement that creates a table with just one PRIMARY KEY. In order for there two be multiple PRIMARY KEY’s, there must be a line fallowing the column names, data types, and constraints that begins with PRIMARY KEY and has the PRIMARY KEY column names in parenthesis.

TASK 11



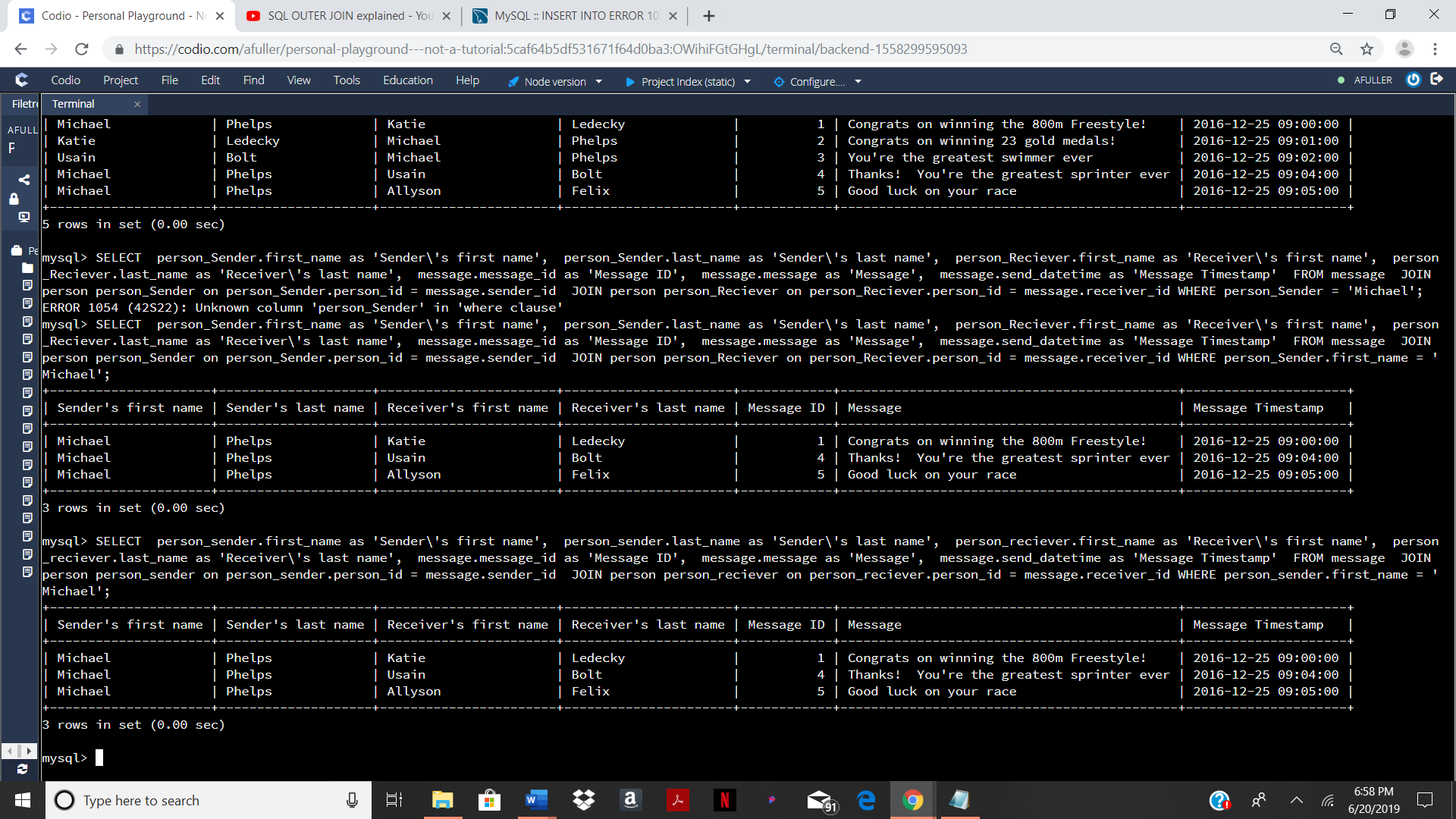
Similar to tasks 1 and 8, this task requires an INSERT INTO statement. For each picture there is an ID, name, and location.

TASK 12



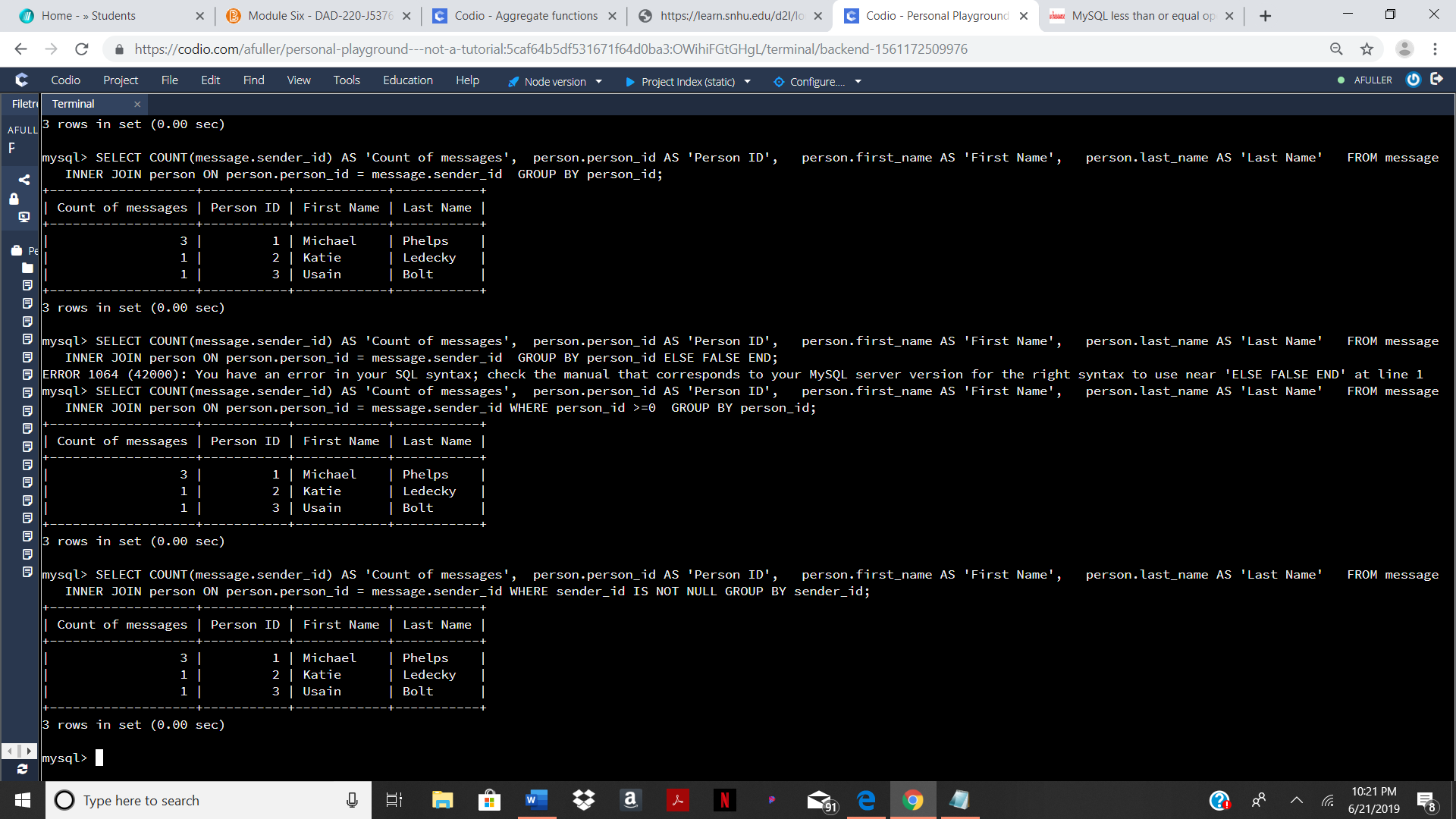
This task is similar to tasks 1, 8, and 11, this task can be completed by writing an INSERT INTO statement for the *message\_image* table. There are two columns in this table, and both need to be filled.

TASK 13



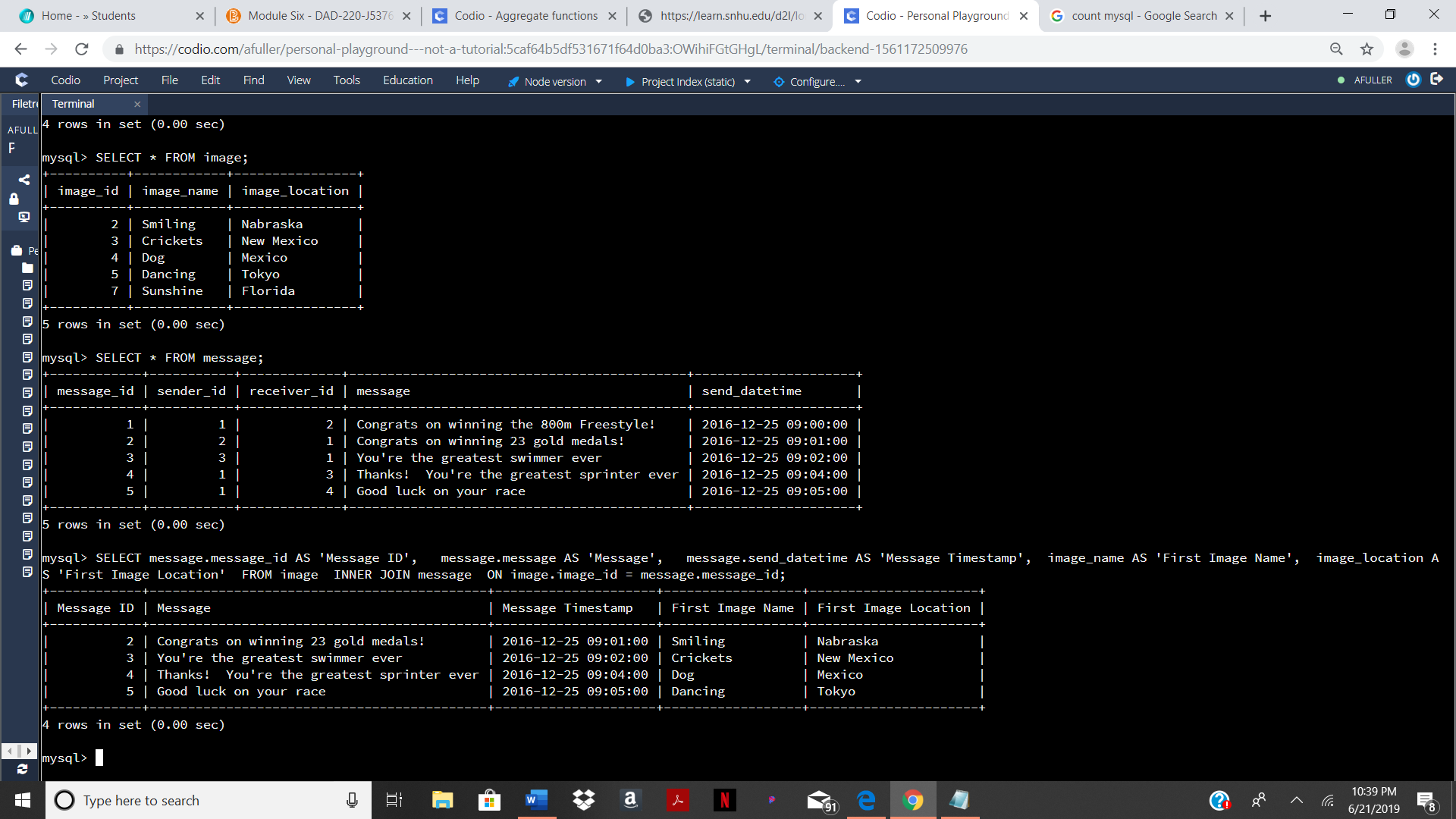
This is a fairly elaborate statement in comparison to previous tasks. In order to get the sender’s and receiver’s first and last names, the name columns need to be combined with the sender and receiver columns using a JOIN statement. By using the AS syntax, the column names can be changed when they are selected.

TASK 14



This task requires the use of the COUNT() function in order to count the amount of messages that were sent by each of the Olympians. An INNER JOIN is also used to combine selected data from the *person* and *message* table.

TASK 15



This task requires and INNER JOIN in order to combine data from the *image* and *message* tables. The *image\_id* and *message\_id* columns were INNER JOINED in order to display only messages that contained image(s).