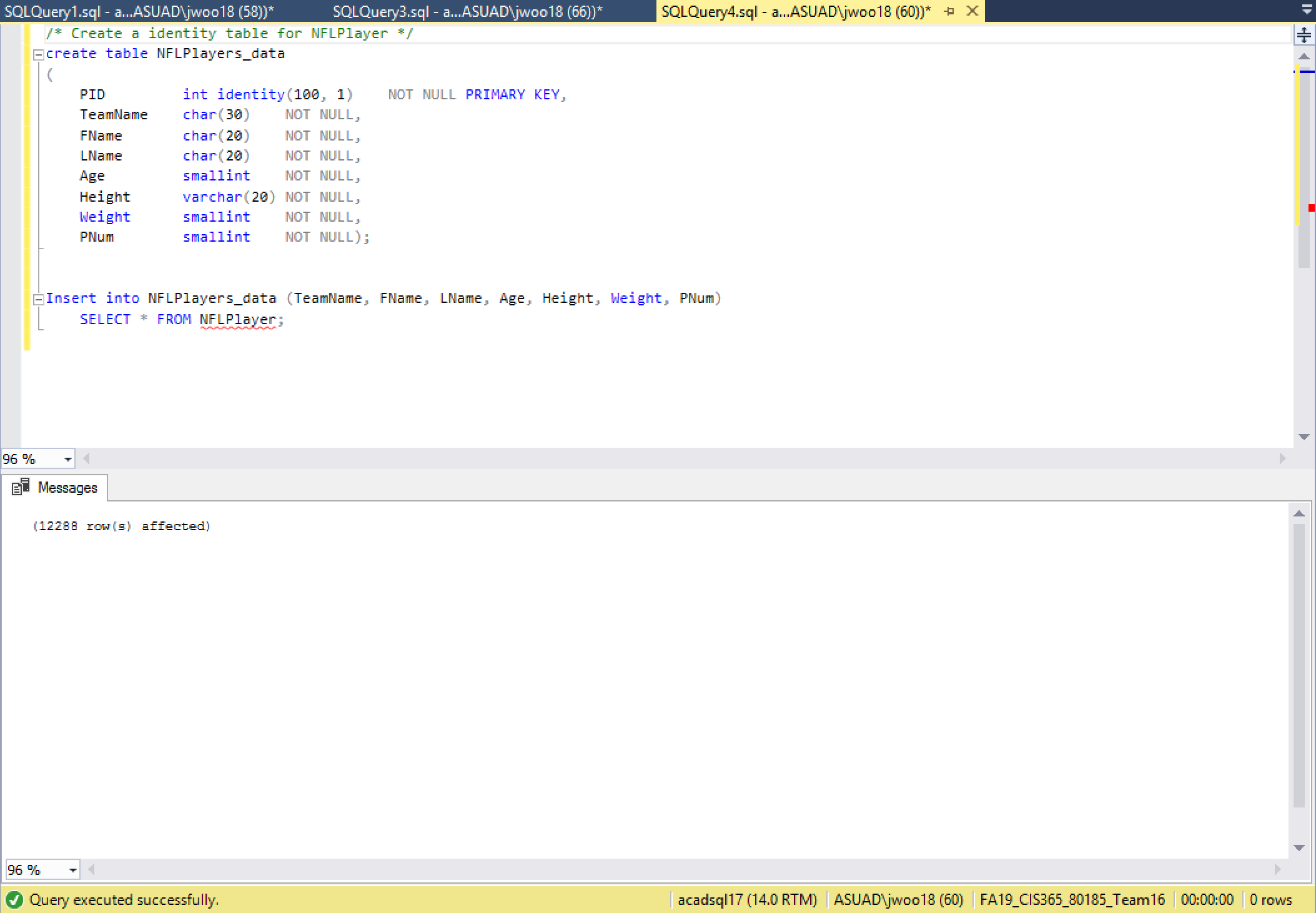
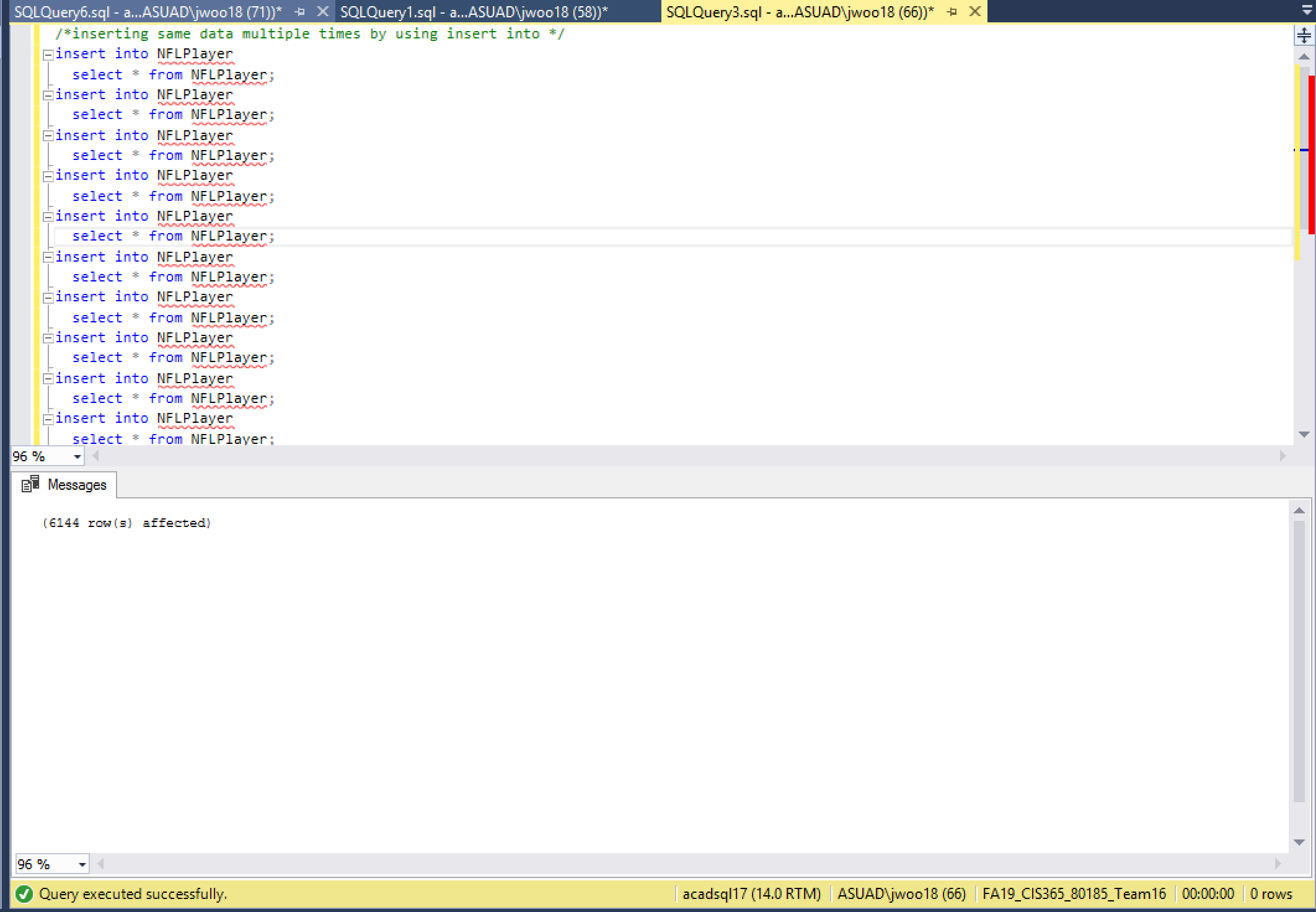
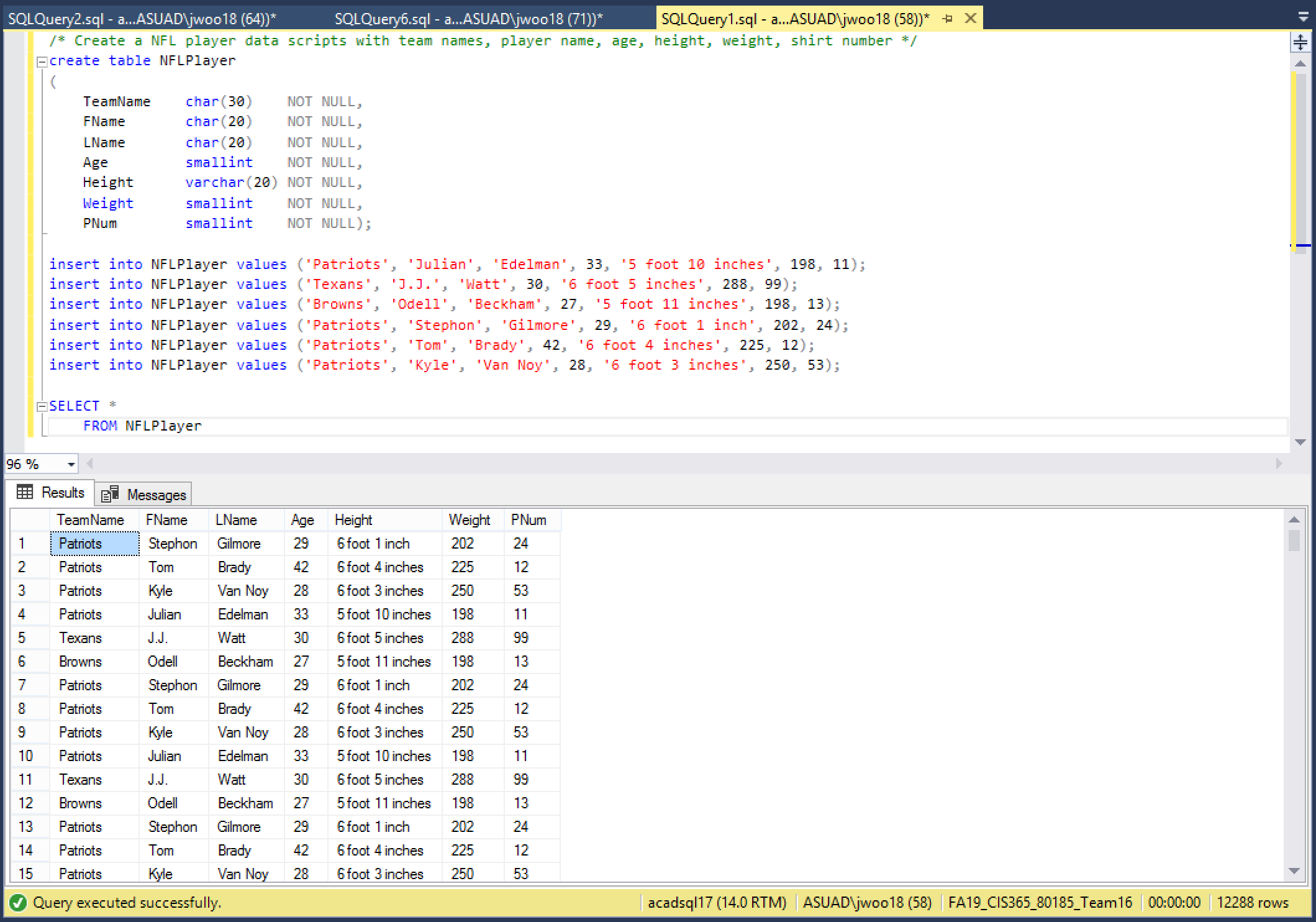
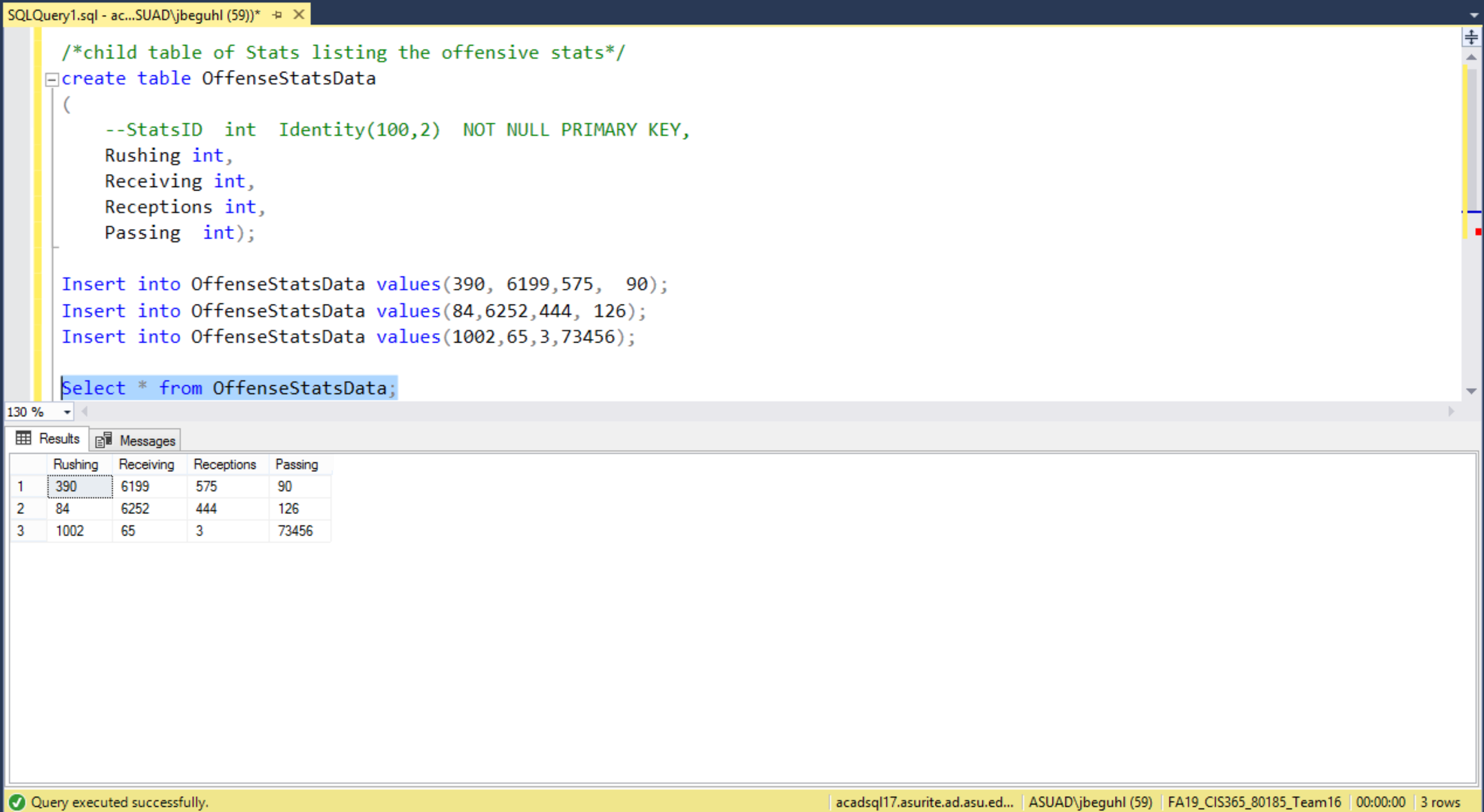
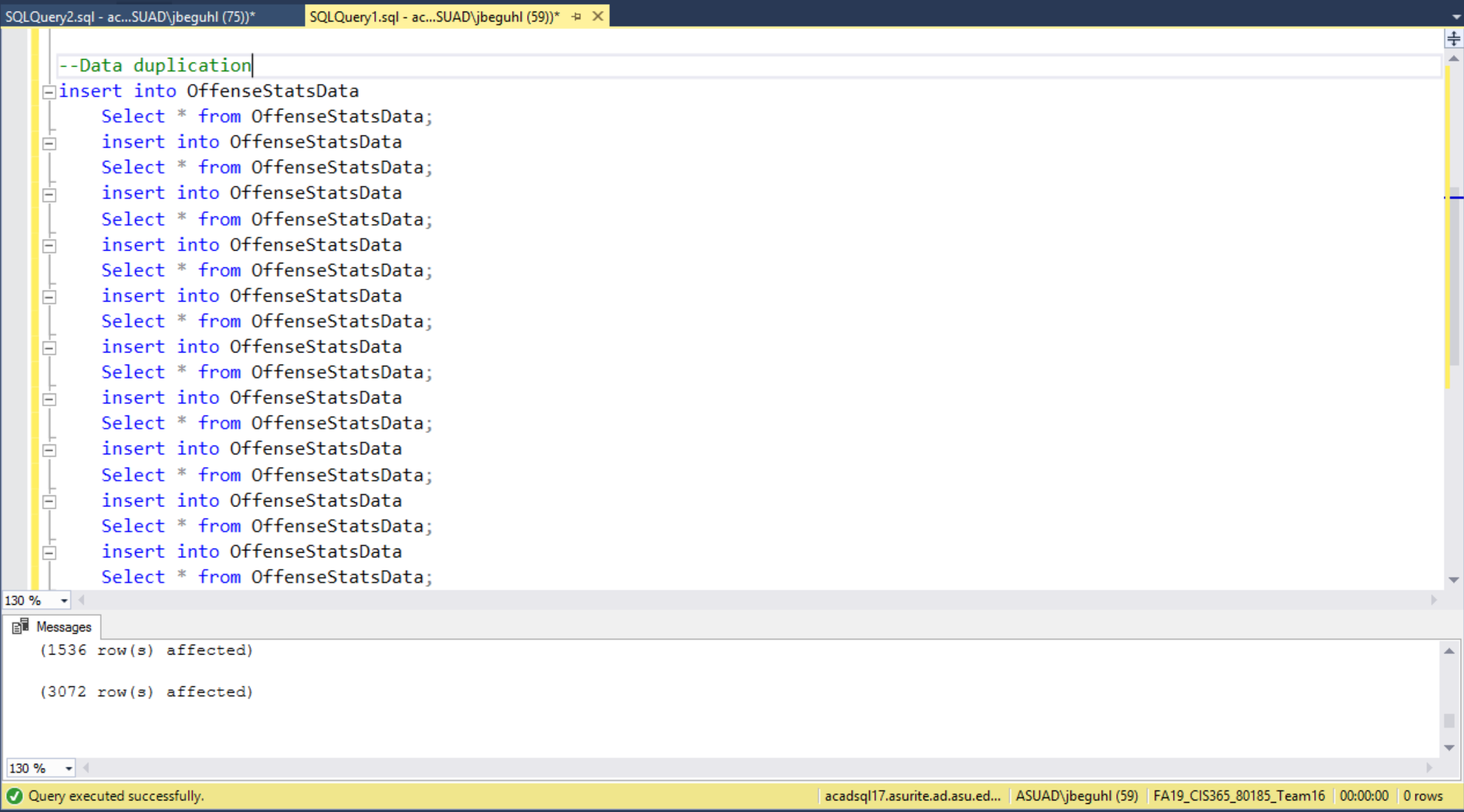
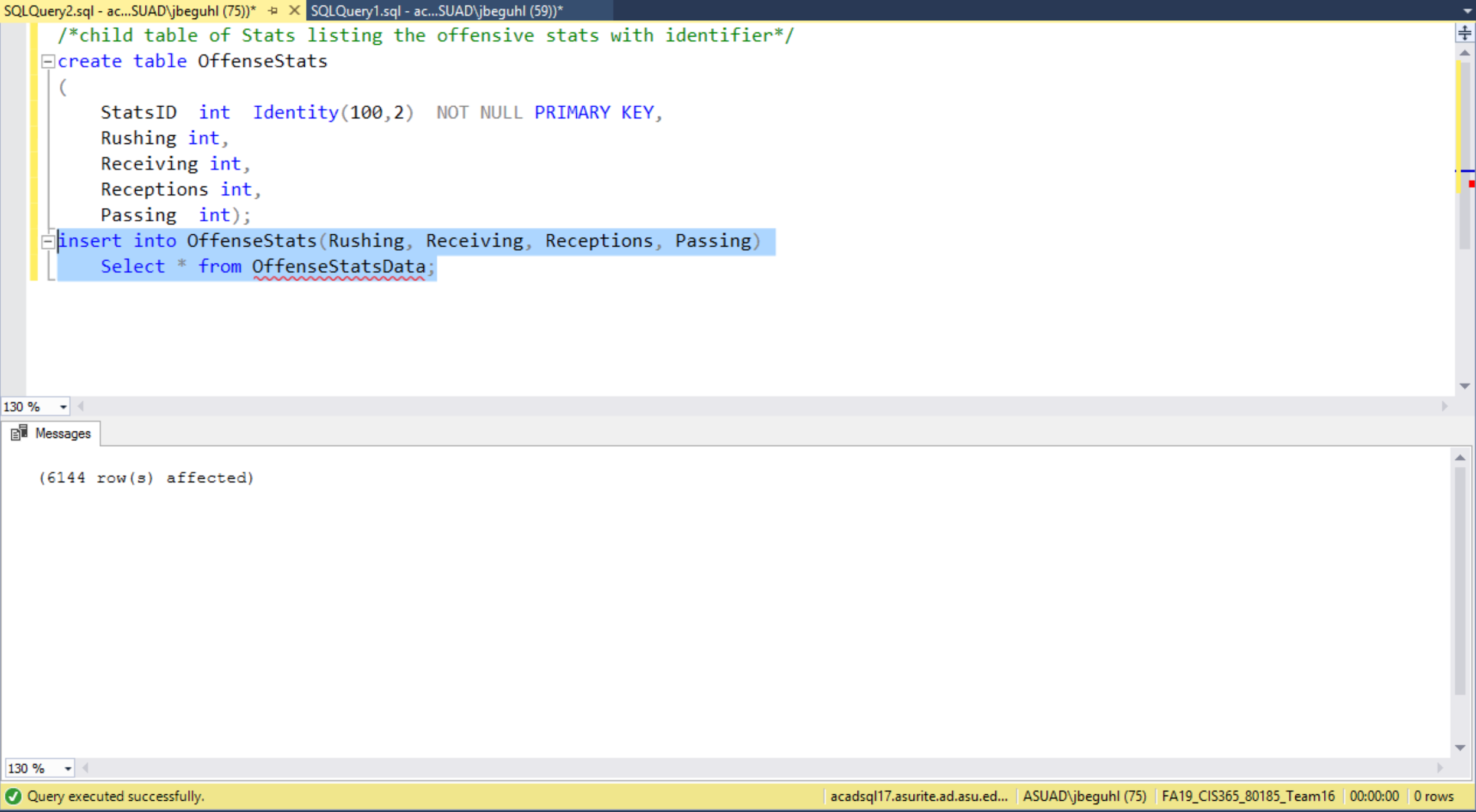
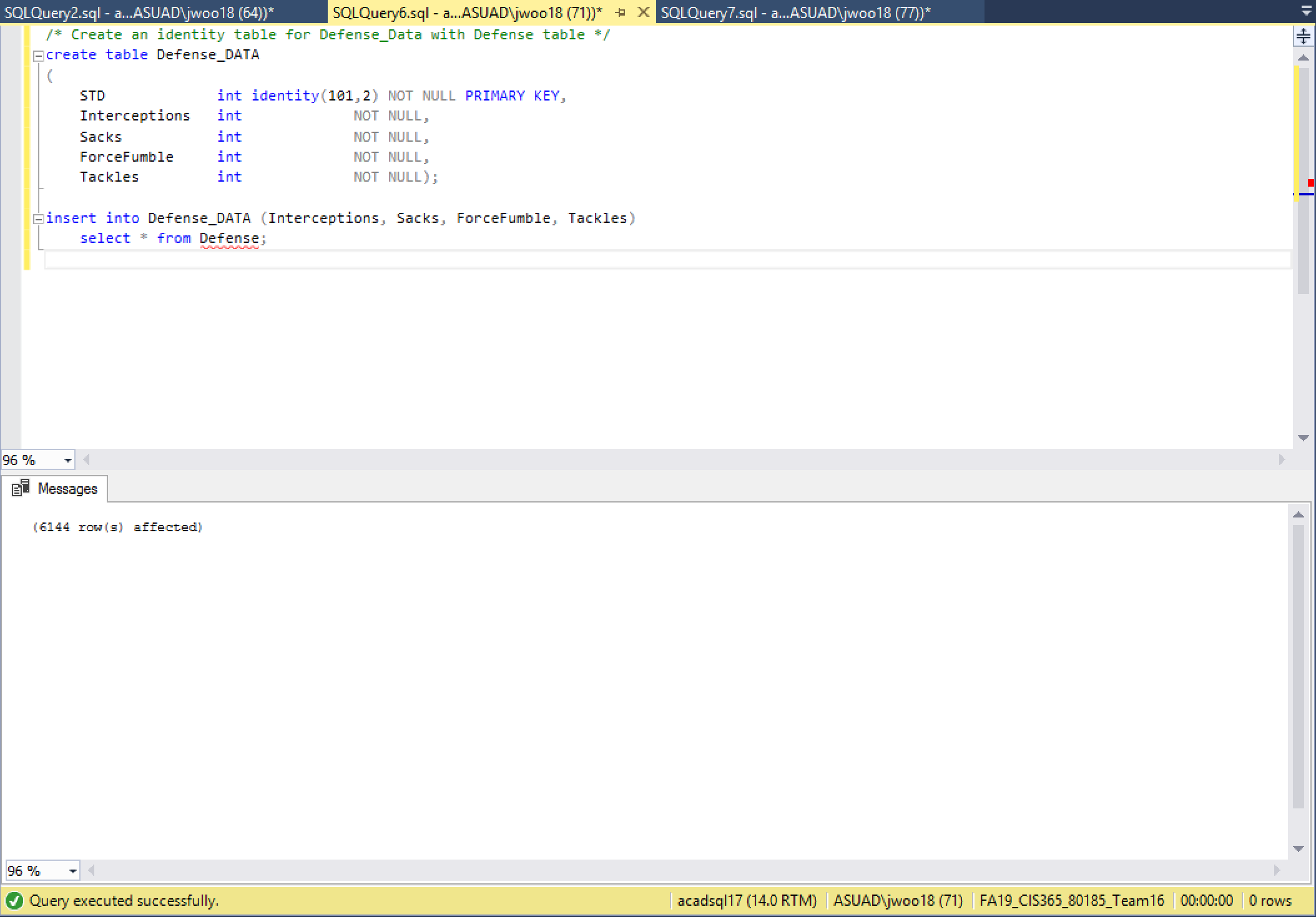
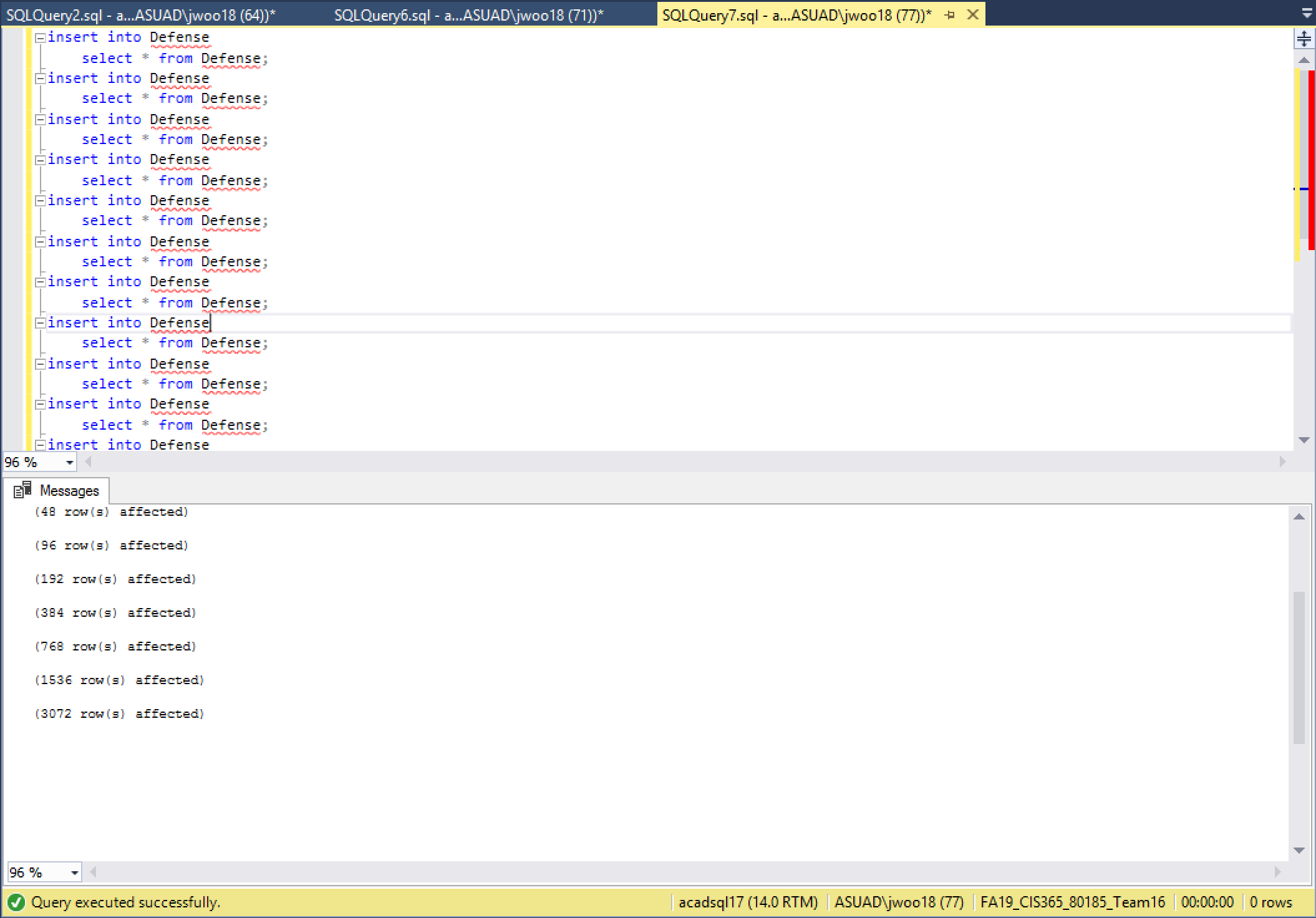
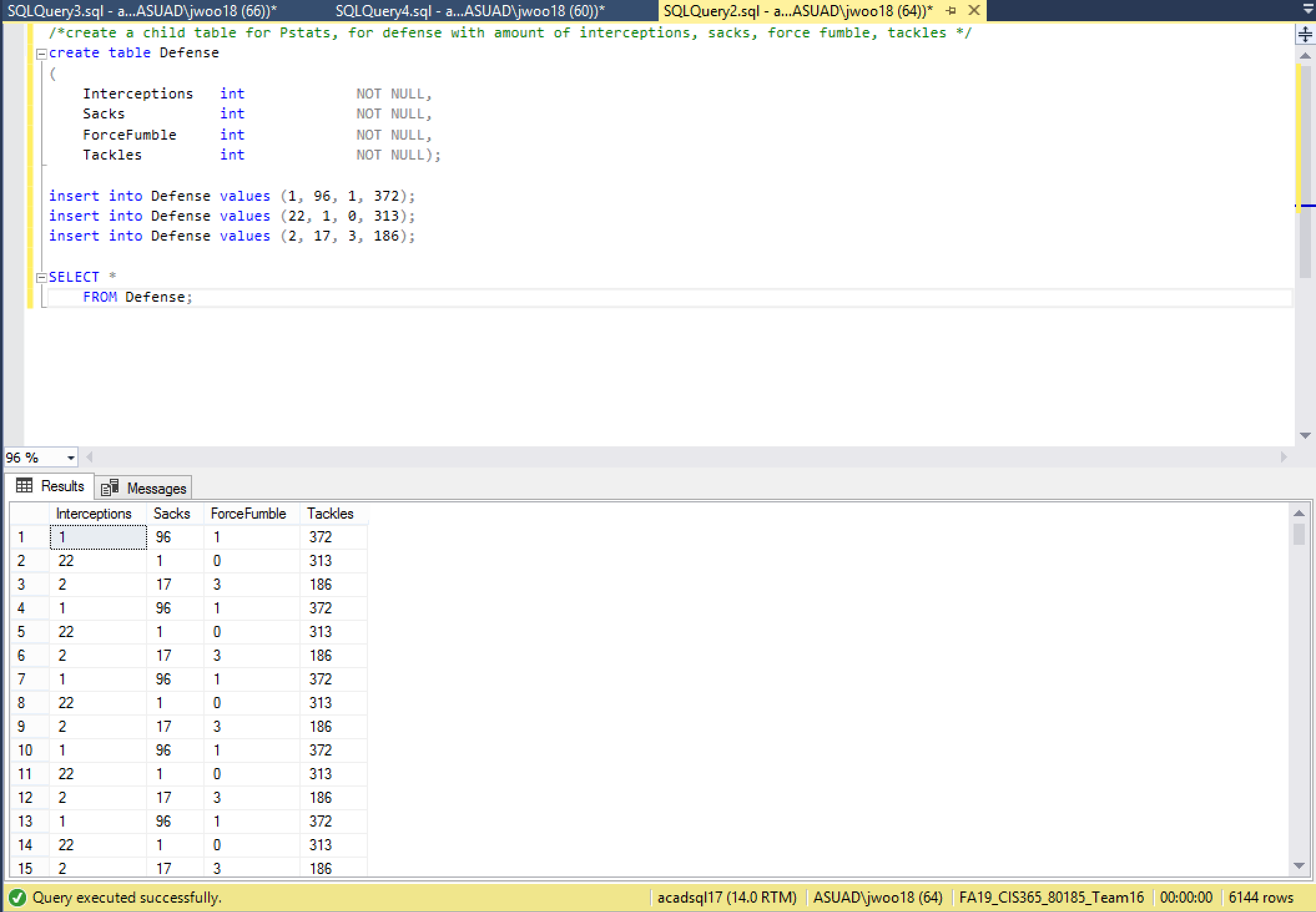
***NFL PLAYER***

***OFFENSE***

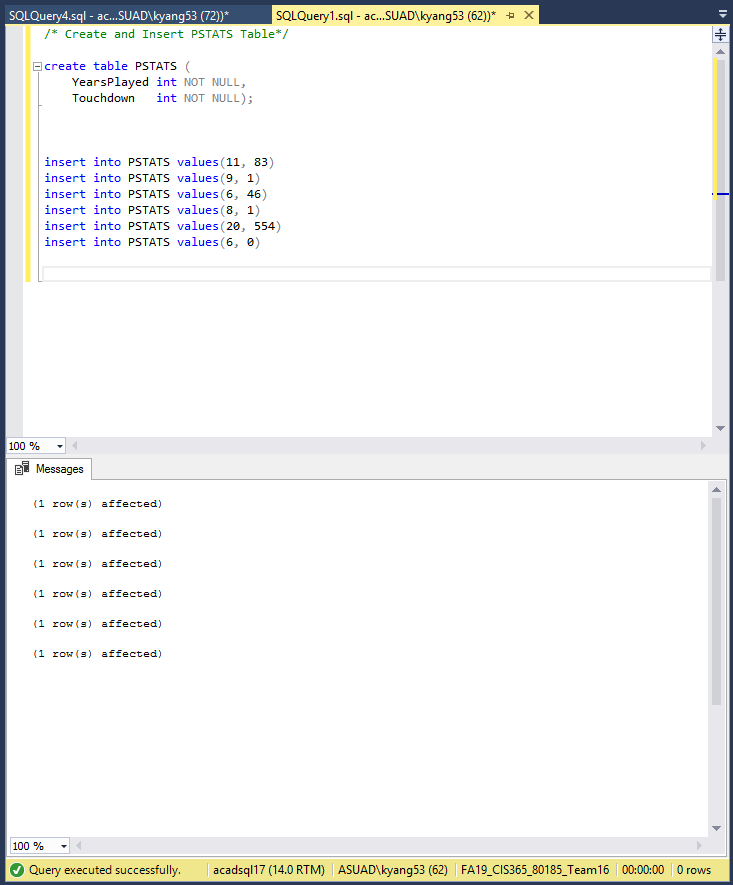


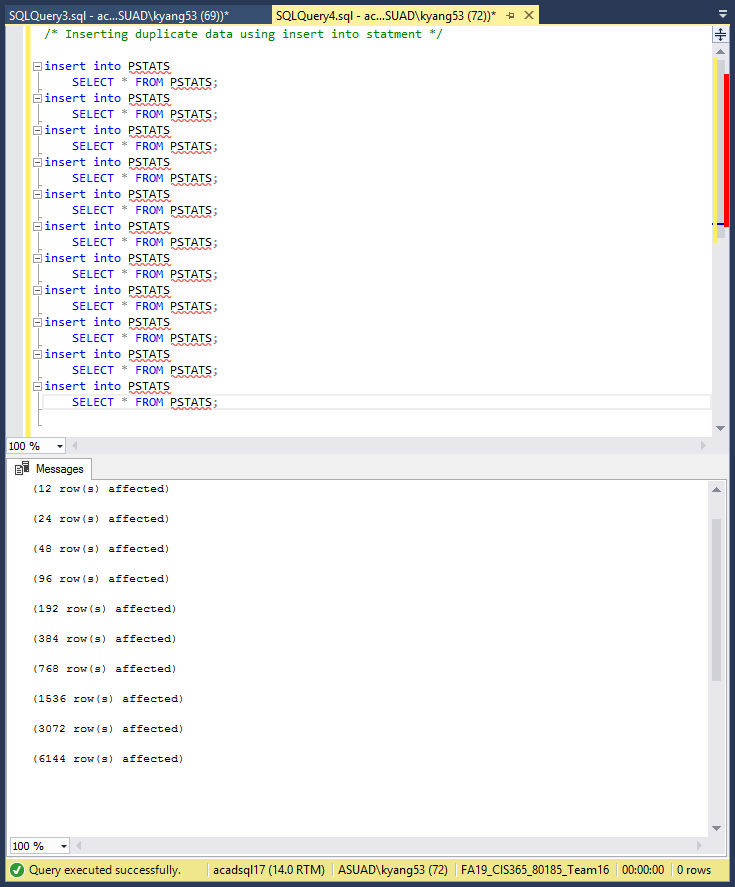


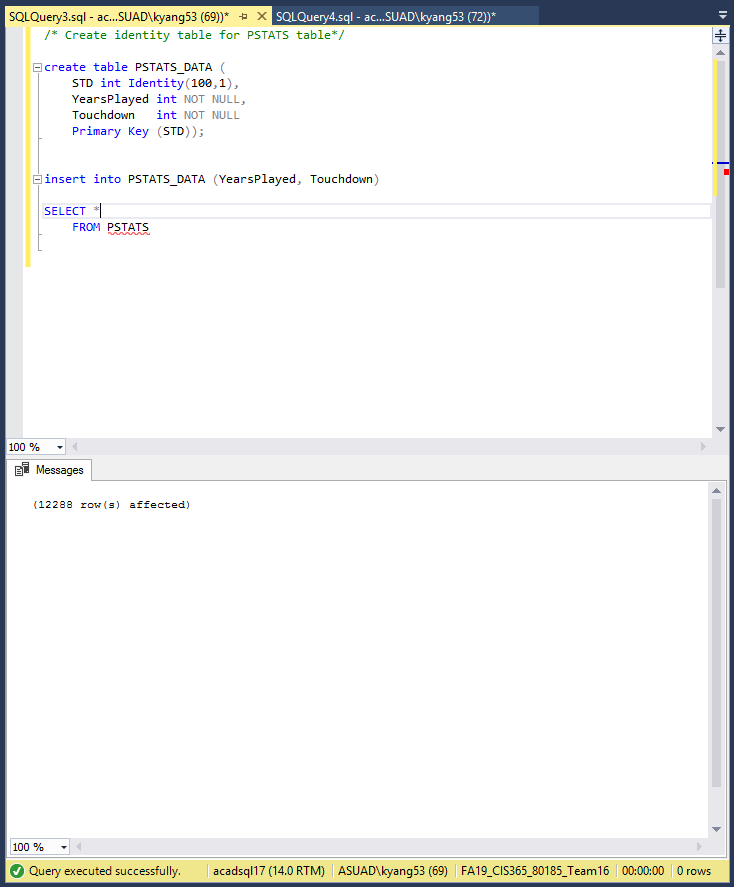


***DEFENSE***

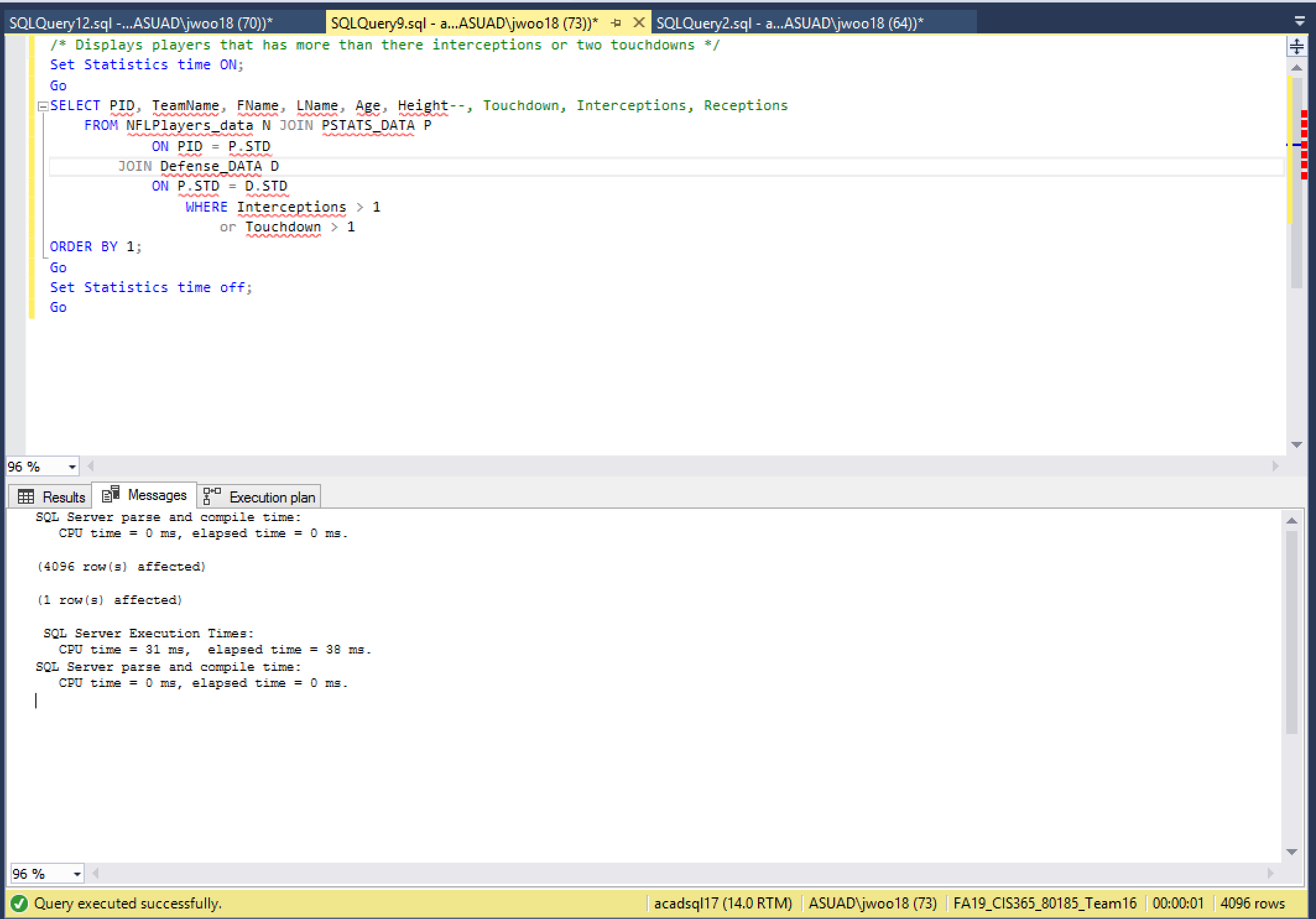
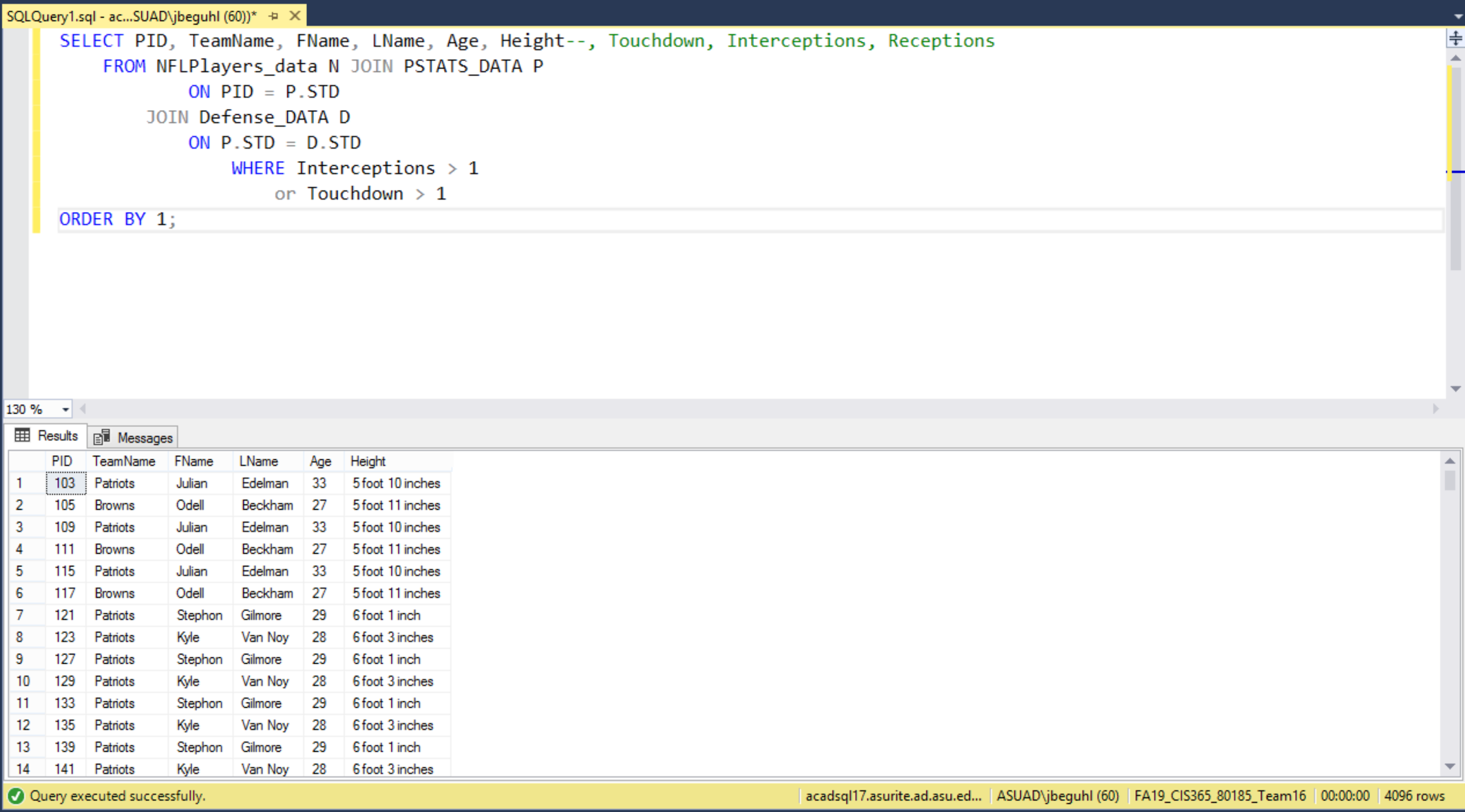
***PLAYER STATS***

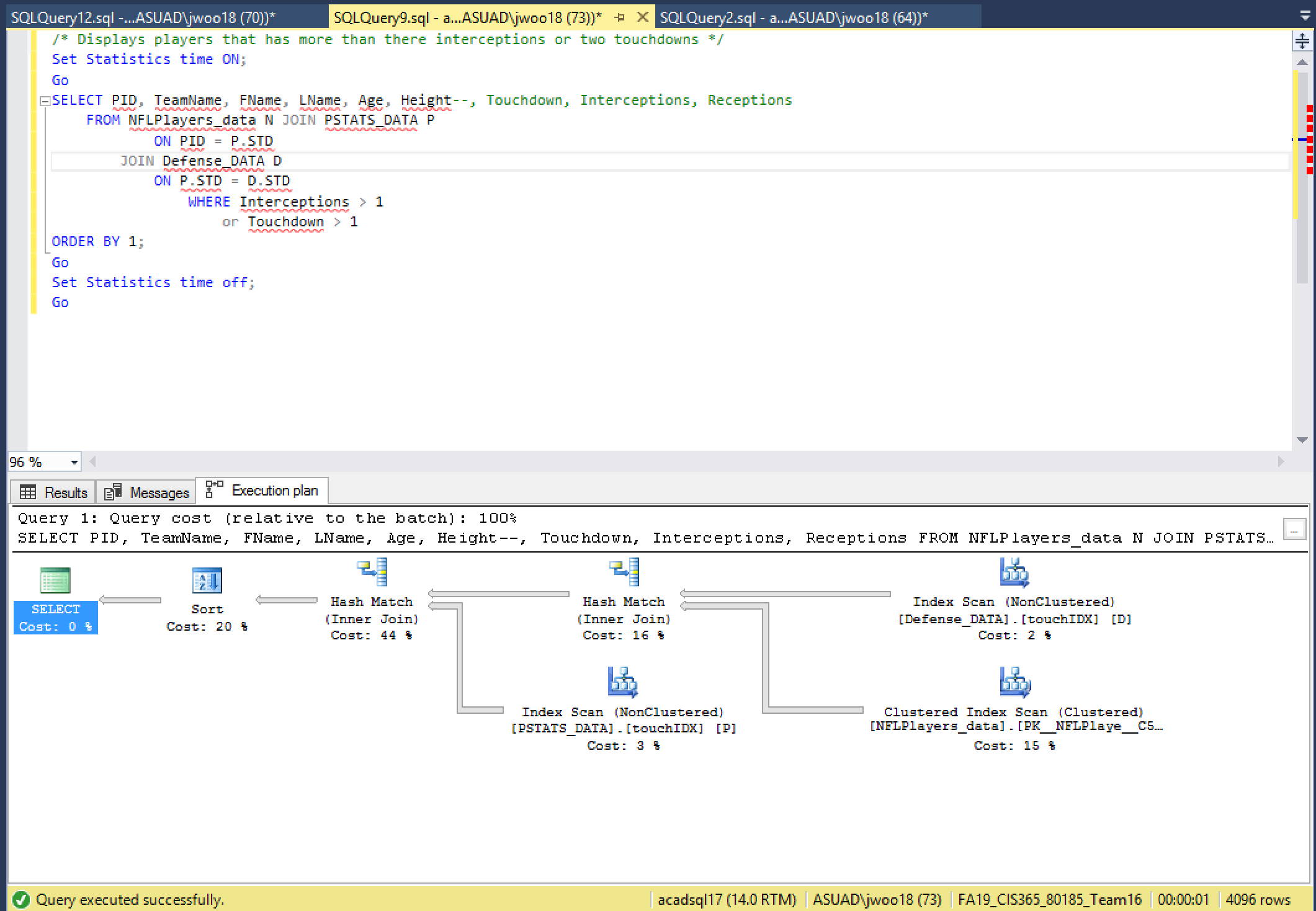
******

******

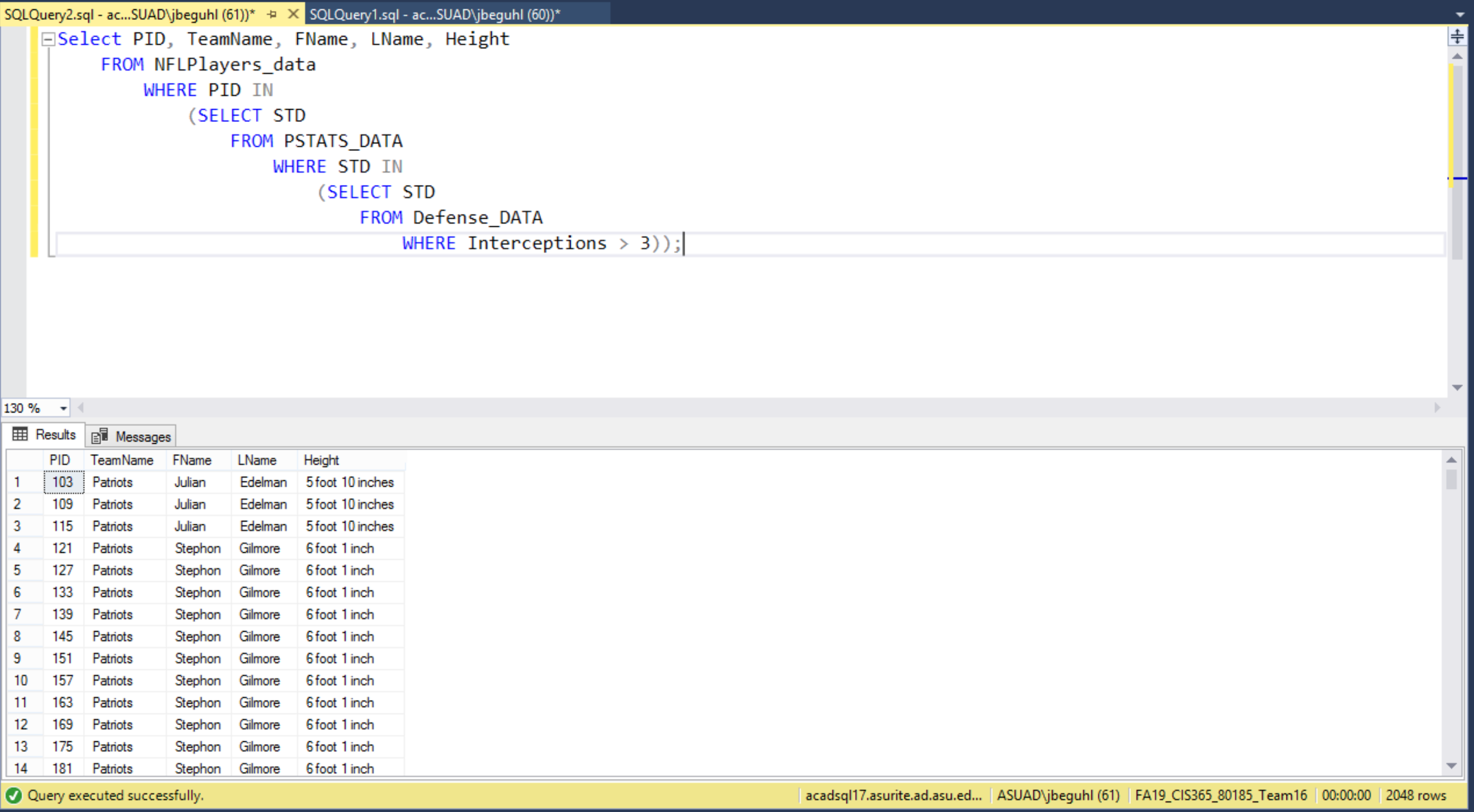
******

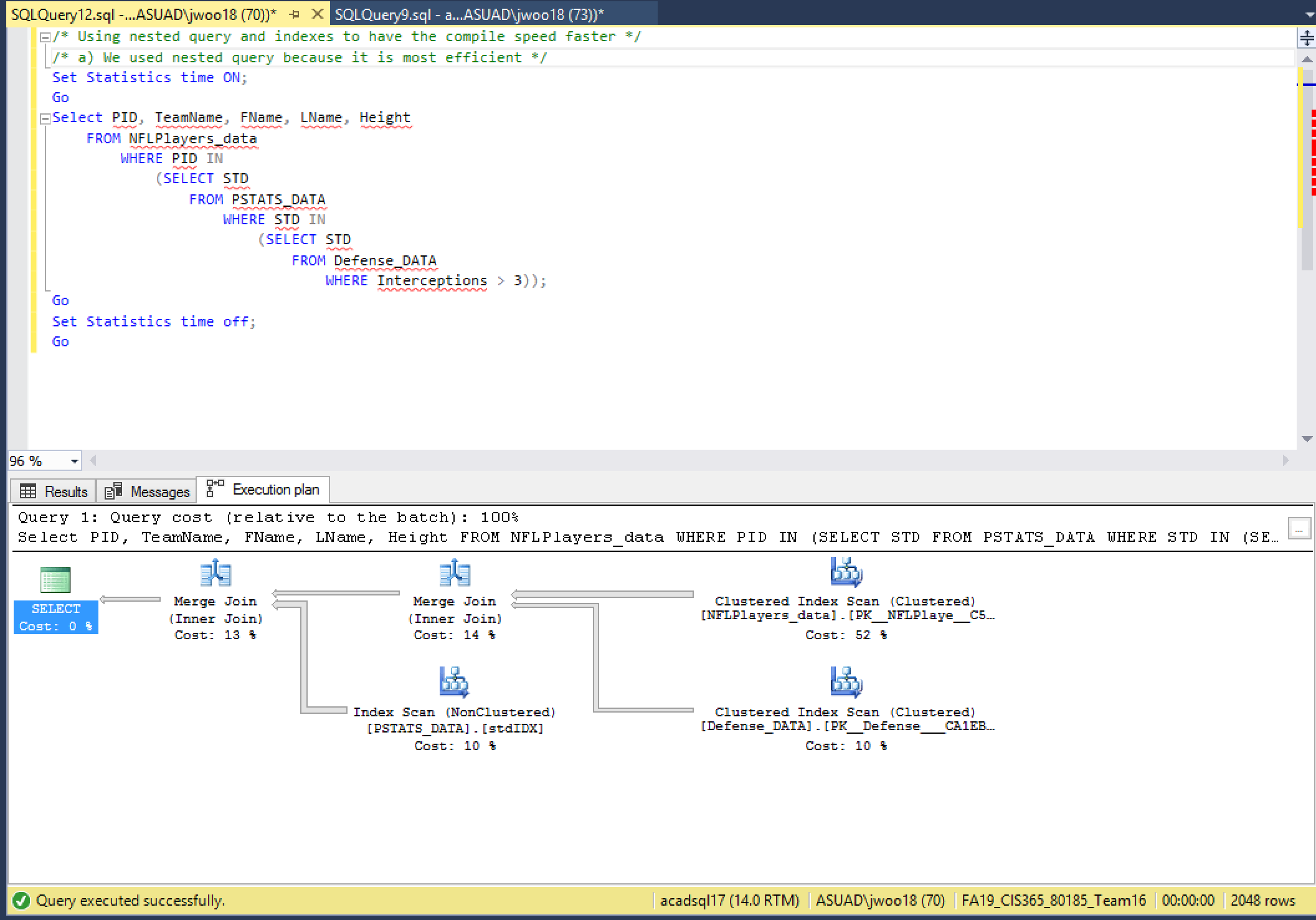
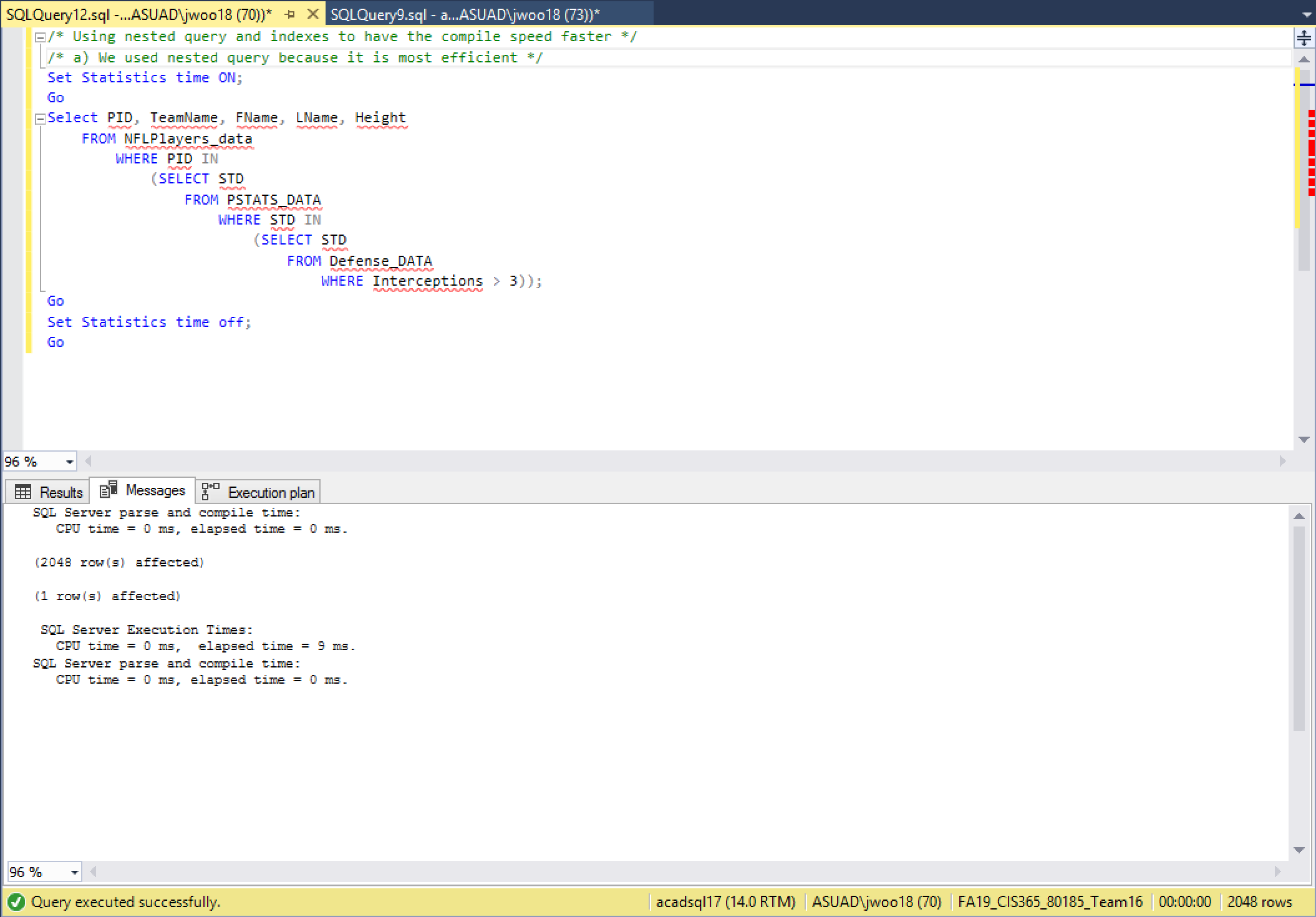
***JOIN BEFORE INDEX***

******

******

***NESTED QUERY AFTER INDEX***

******

******

***3. QUESTIONS***

1. We choose the query with multiple joins in it because we know from what Dr. Moser said, “Joins are more expensive because they are taxing on DBMS” which will give an outcome of longer execution time. We believe that physically changing the scripts and database with indexes will improve the execution time drastically because nested queries are must more efficient than join while accomplishing the same functions as joins.
2. First, we brainstormed what story ideas to base our database. Then we all agreed on making our database based on NFL players, teams and stats. Once we figured out what we wanted our database to show, we figured out what different tables, attributes and fields we needed in order to accomplish this. Then we drew out the EERD on the whiteboard to have a full visualization of our database. Then we created the tables and inserted data. We brainstormed again to figure out what query to make in order to tax the DBMS heavily. We ran our first query and recorded the execution time and query plan. We then optimized query by using nested query instead of joins statement and then we added indexes. We recorded the execution time and query plan before and after adding indexes.
3. With the joins statement, the execution time was 38 ms. With nested query before adding indexes, the execution time was 12 ms and after we added indexes, the execution time was 9 ms, improving the execution time by four times.
4. The main reason why the execution time was faster is because of optimization. We expected the result to be more drastic because of how many records we have and amount of joins in the query. The actual result was an improvement but as much as we hoped it to be.
5. If we had more time to improve our database, we would definitely add more tables and record in the database. By adding more data in a DBMS, this will give us more data to the query by adding more where clause, order by and joins, which will result in a more drastic improvement after optimization.