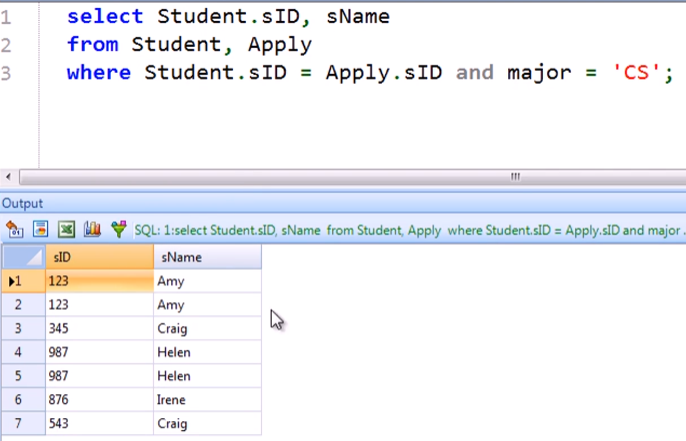
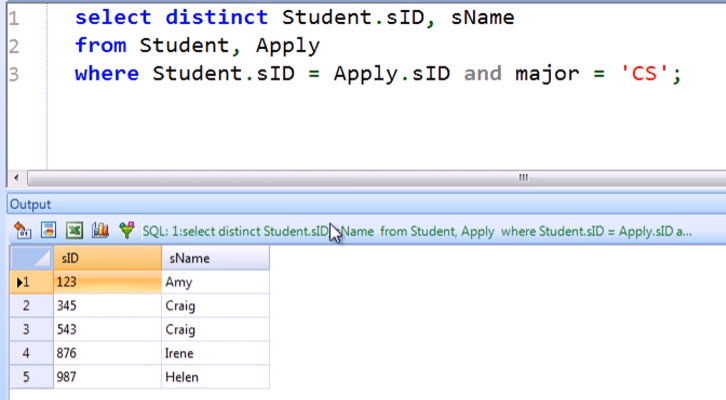
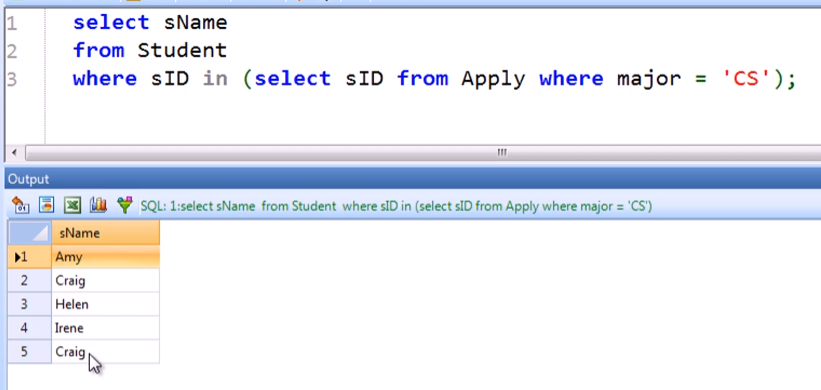


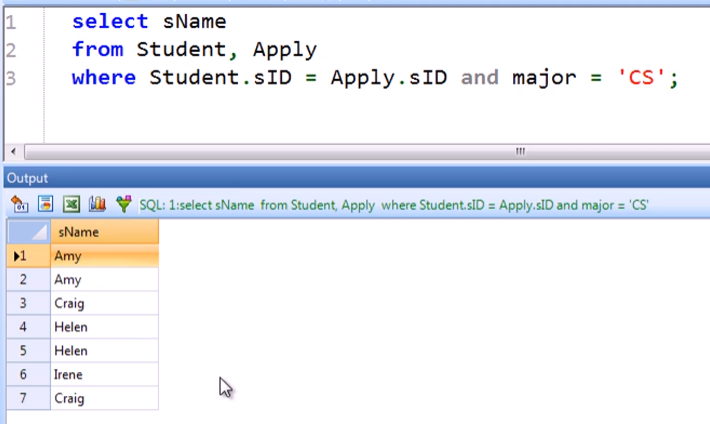
* Same without a subquery:

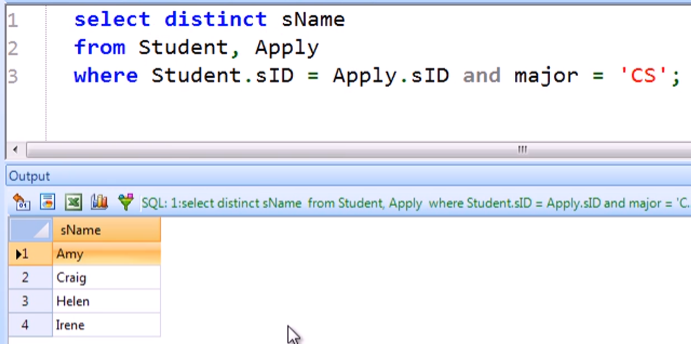






* Same without a subquery:

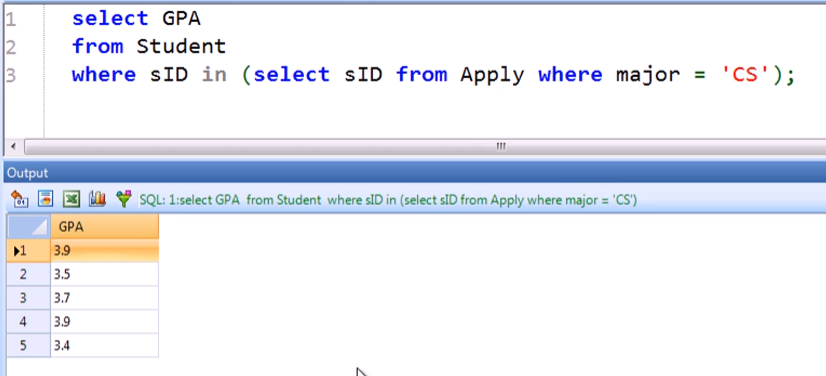


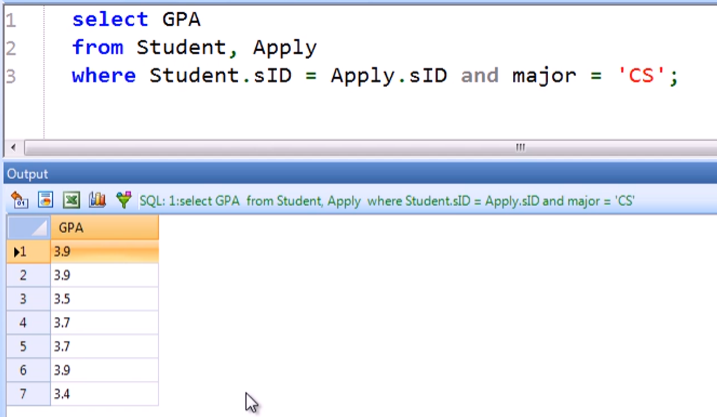
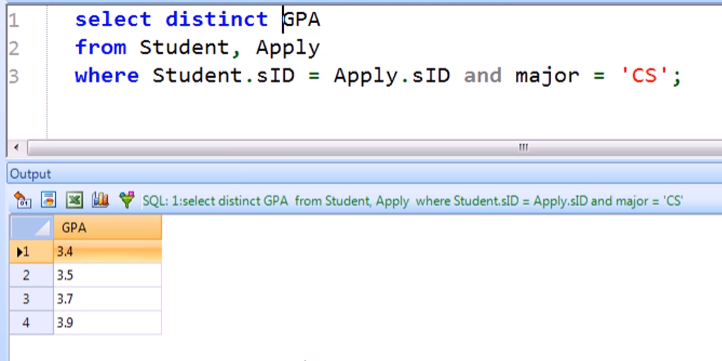


* + WRONG
  + Craig is tagged as duplicate because the only distinction is the name. The ids of different Craig are not being accounted for

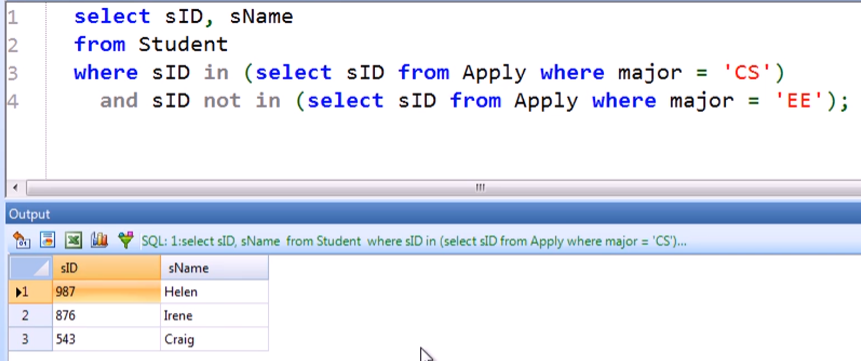
Why some duplicates are important

* Example: getting the average GPA of CS Applicants

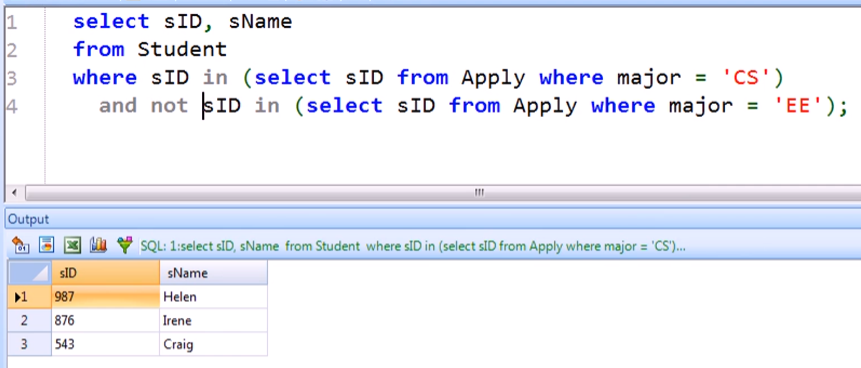


* Using join instead of subquery:
  + 
    - We will be counting double GPAs here and the Average will not be the same
  + 
    - In this case, we will be short by one GPA and the Average will not be the same!

Writing the query equivalent to the EXCEPT operator

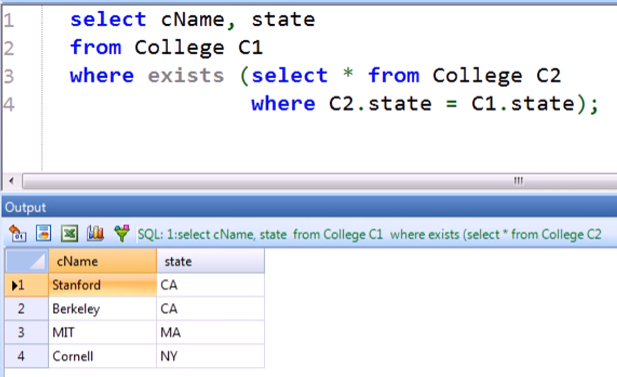


Or

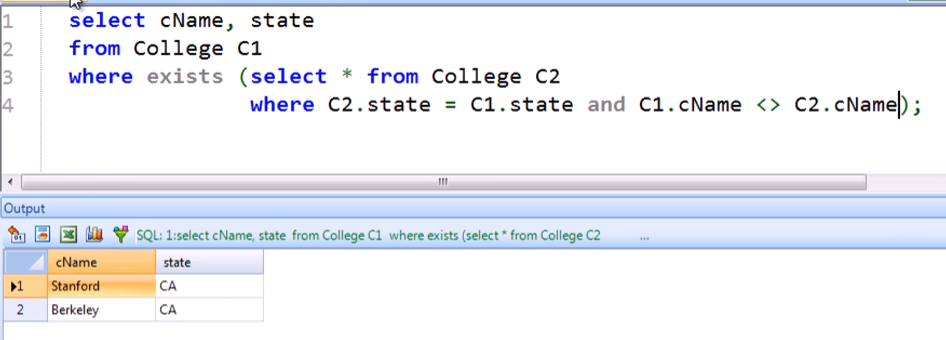


EXISTS operator

* To check whether a subquery is empty or not empty



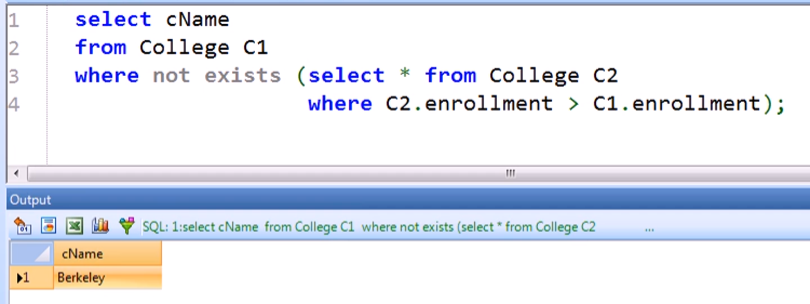
* Wrong answer!

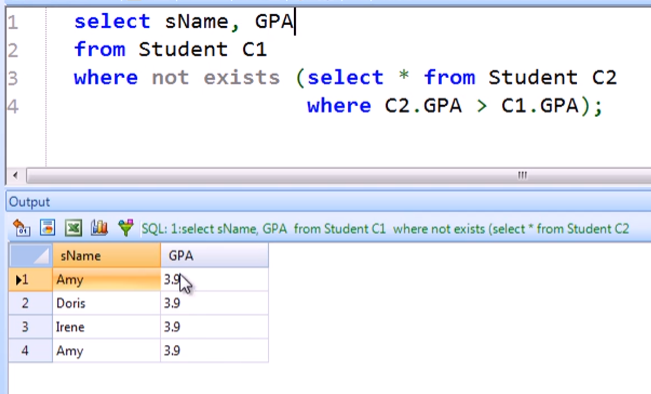


* Fixed by adding a condition where cName must not be matched with itself

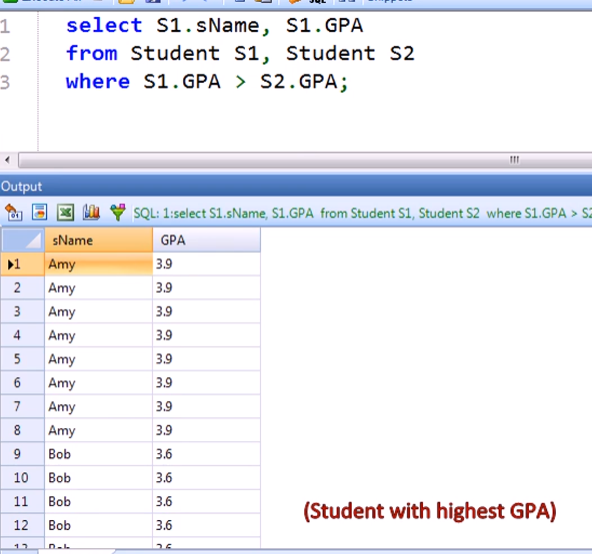
Querying like the MAX operator without using it

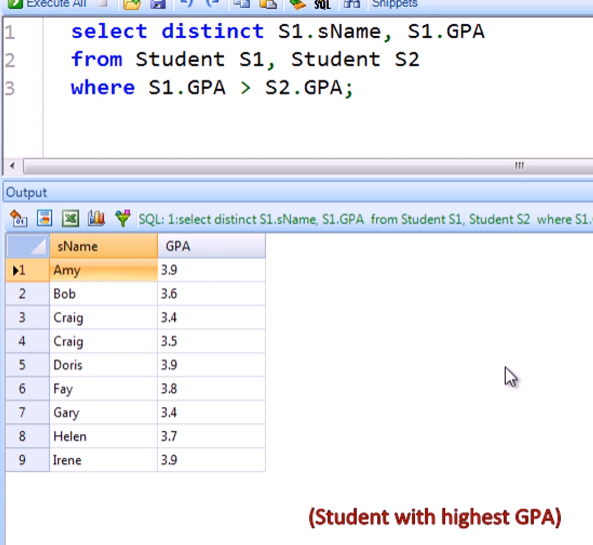
* Eg. College with highest enrolment





Trying to query like MAX without using subqueries

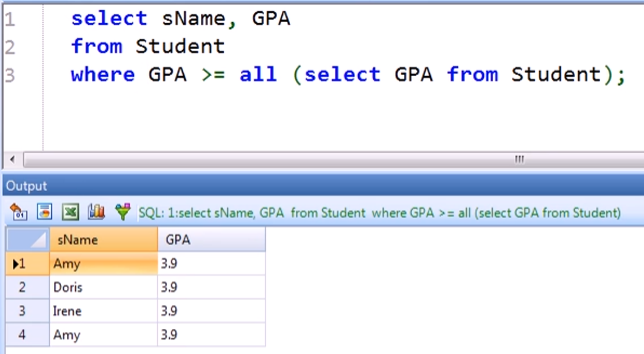




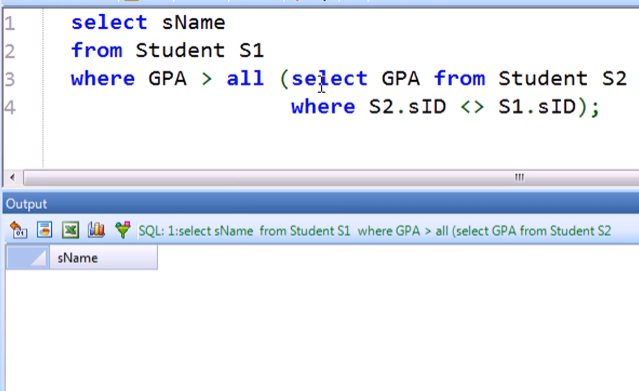
* THESE ARE WRONG!
* This is only getting ALL students’ GPA EXCEPT the LOWEST

Querying like MAX using the ALL operator

* This is like AND operator that is passed or executed from each record

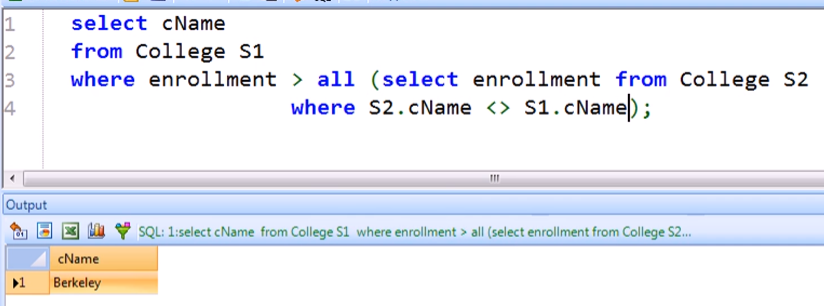


* Correct!



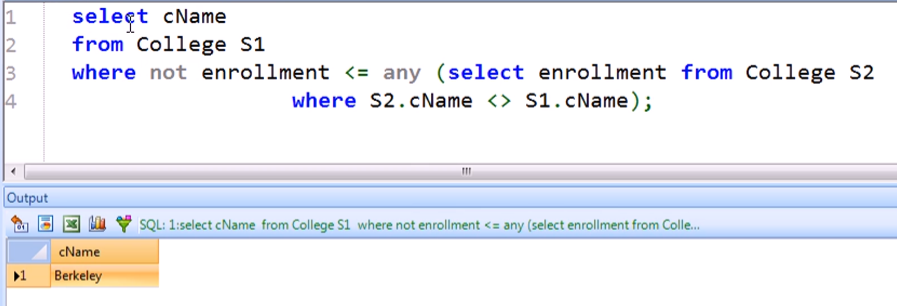
* Incorrect! Because there are 4 students that have the EQUAL max GPA
* If there is only ONE student that have the max GPA, this query will work!

Applying that using enrolment since enrolment numbers are unique!

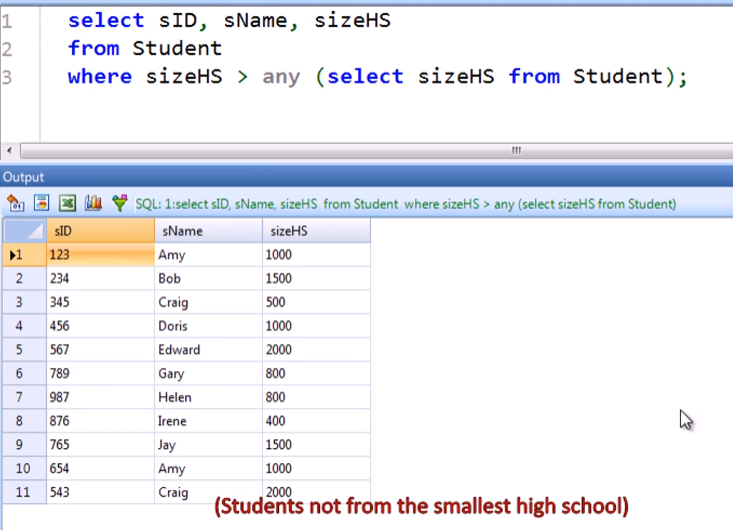


ANY operator

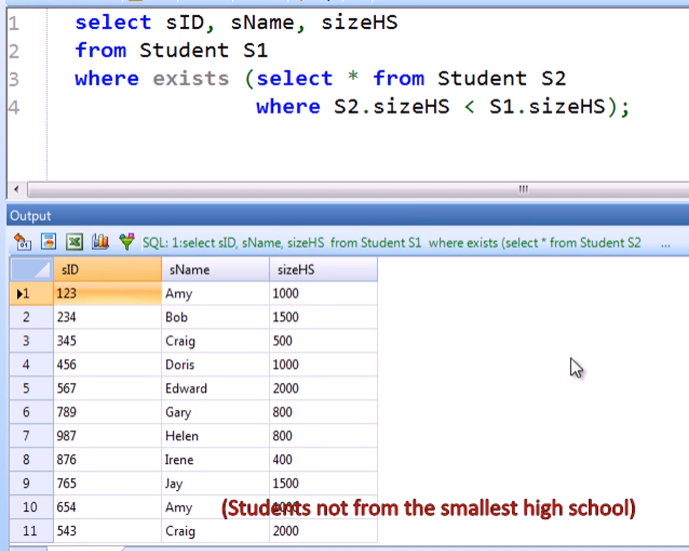
* Just satisfy at least ONE
* Like the OR operator passed or executed from each record



* Getting max enrolment using ANY

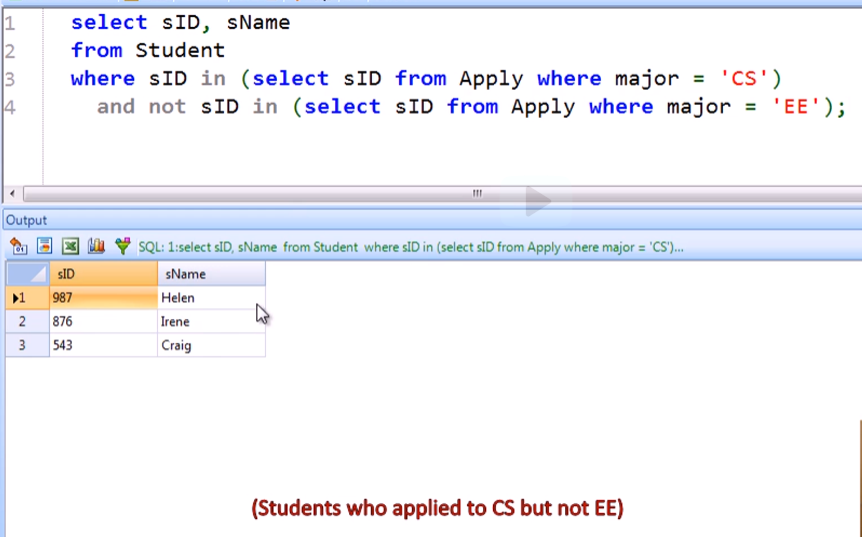


Same query written without using ANY operator

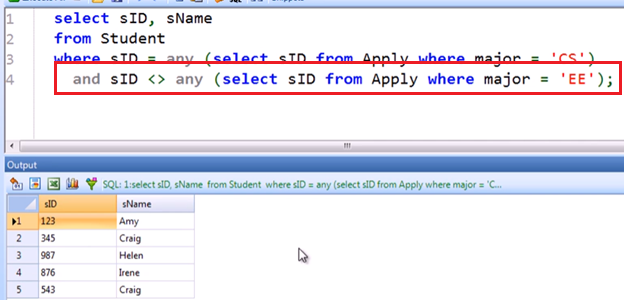


Going back to the big query of Students who applied to CS but not EE

Using IN

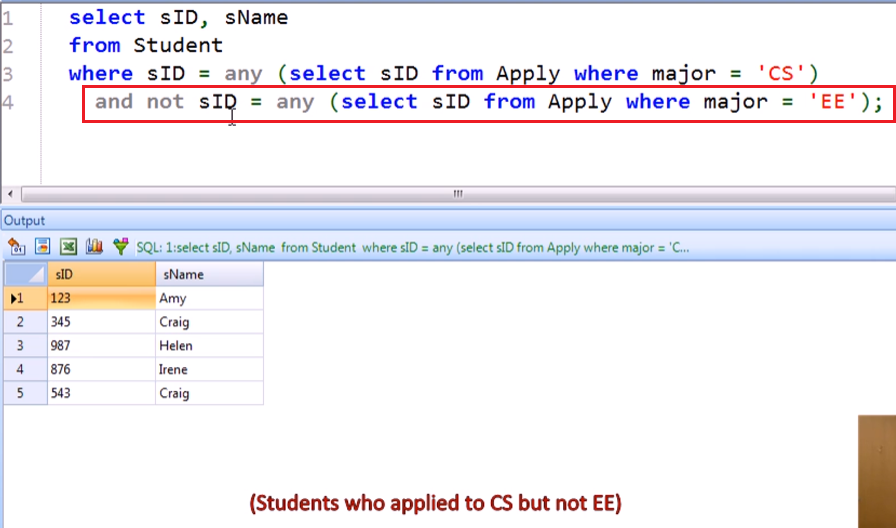


Using ANY



* Wrong answer!
* Because of the highlighted query
  + The condition was satisfied even if we are not looking at the same student ID since we used any!

Correcting the using ANY



* We first check if the SID is in the sIDs of major EE, then negate it!