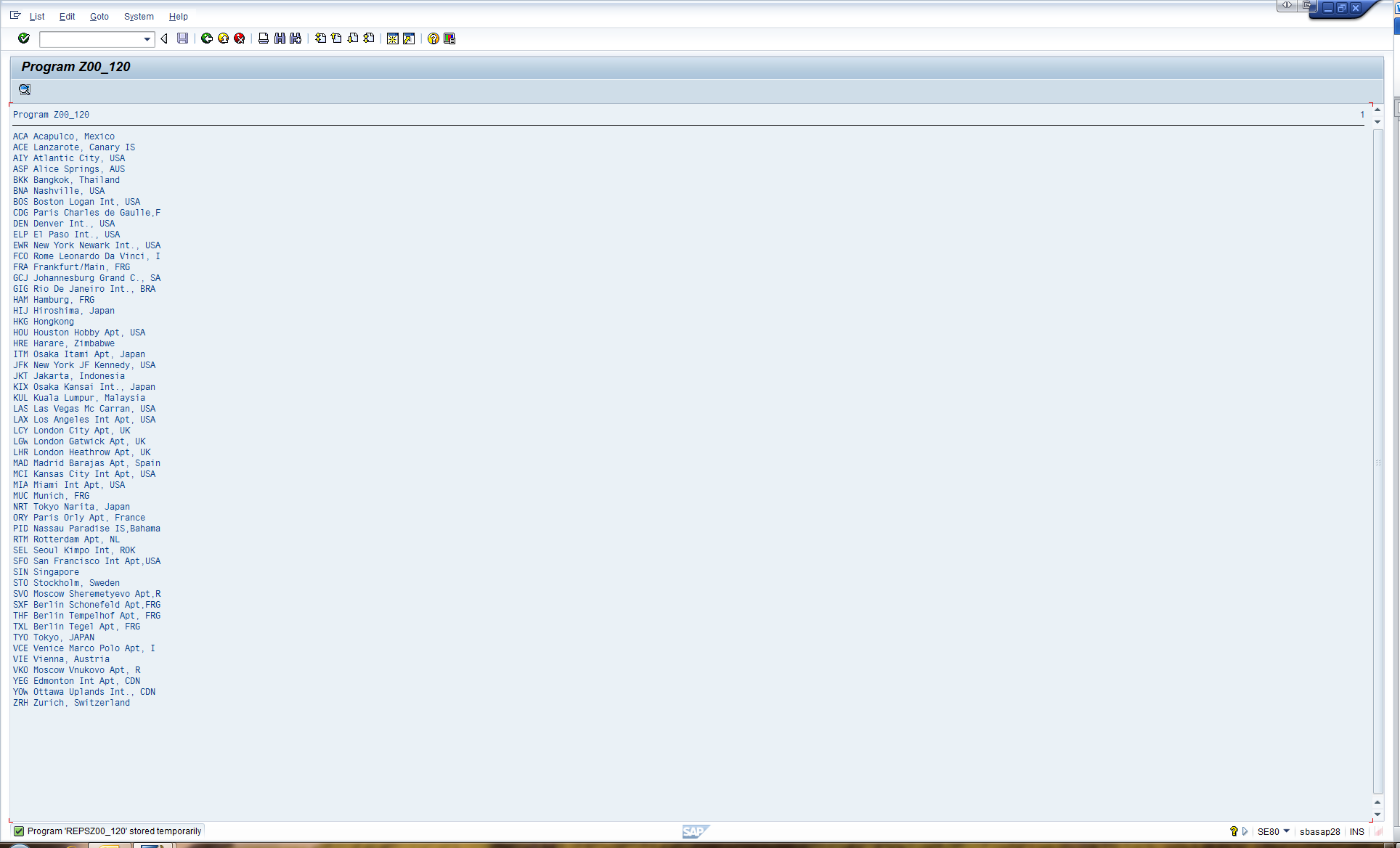
Activity: This lab provides opportunity to practice creating drilldowns in list output

Scenario:Our reservation office wants to be able to see a list of airports which we use for travel. From this list they want to be able to select an airport and see all the flights to and from that airport. From this flights list they want to be able to select a flight and see the fare of that flight.

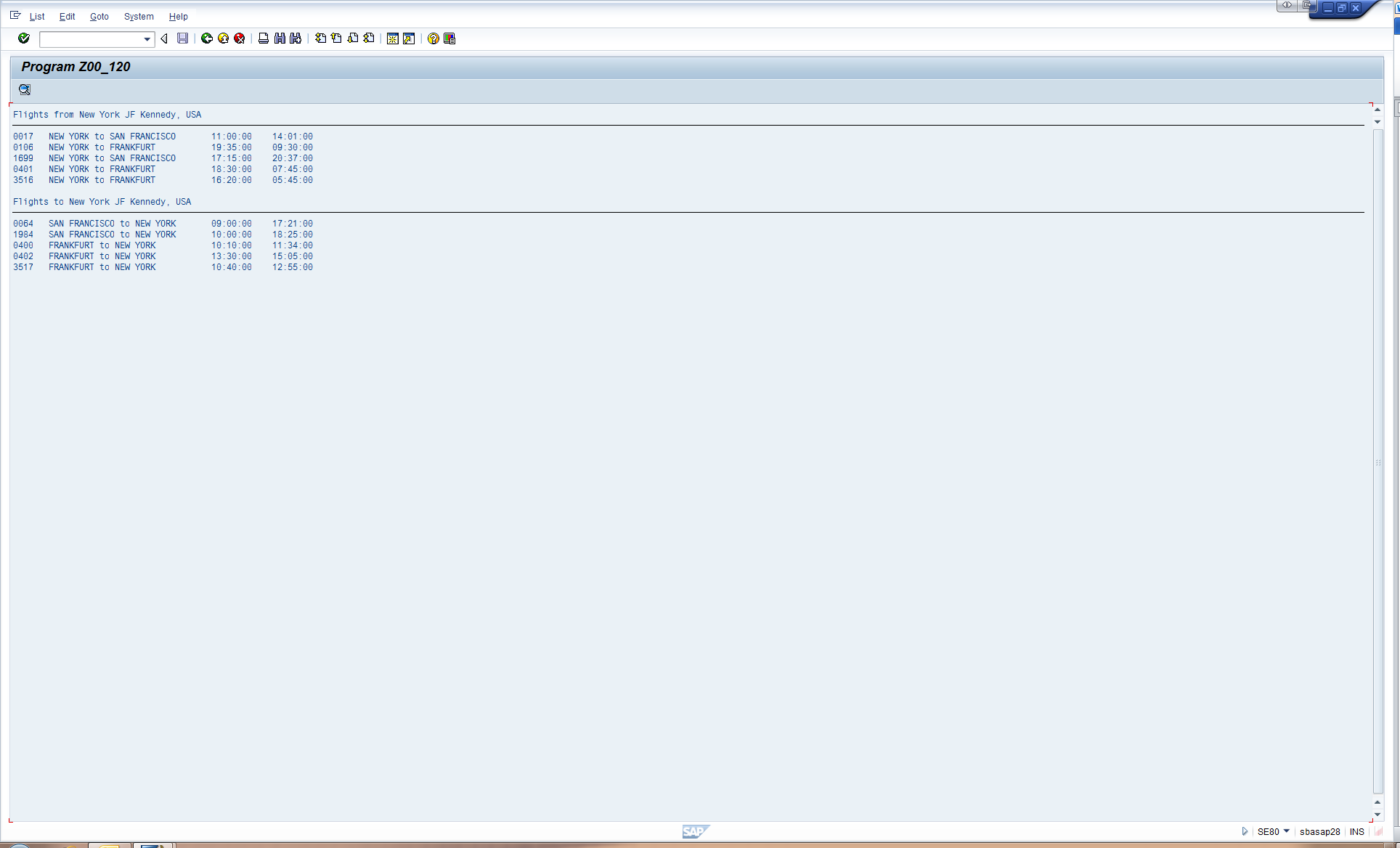
The following tables are used in this exercise: sflight, spfli, sairport. Examine each table to determine which is needed for each aspect of the solution. No joins are needed, just basic queries.

A “prototype” has been developed to guide you in your work. Make your output match the prototype.

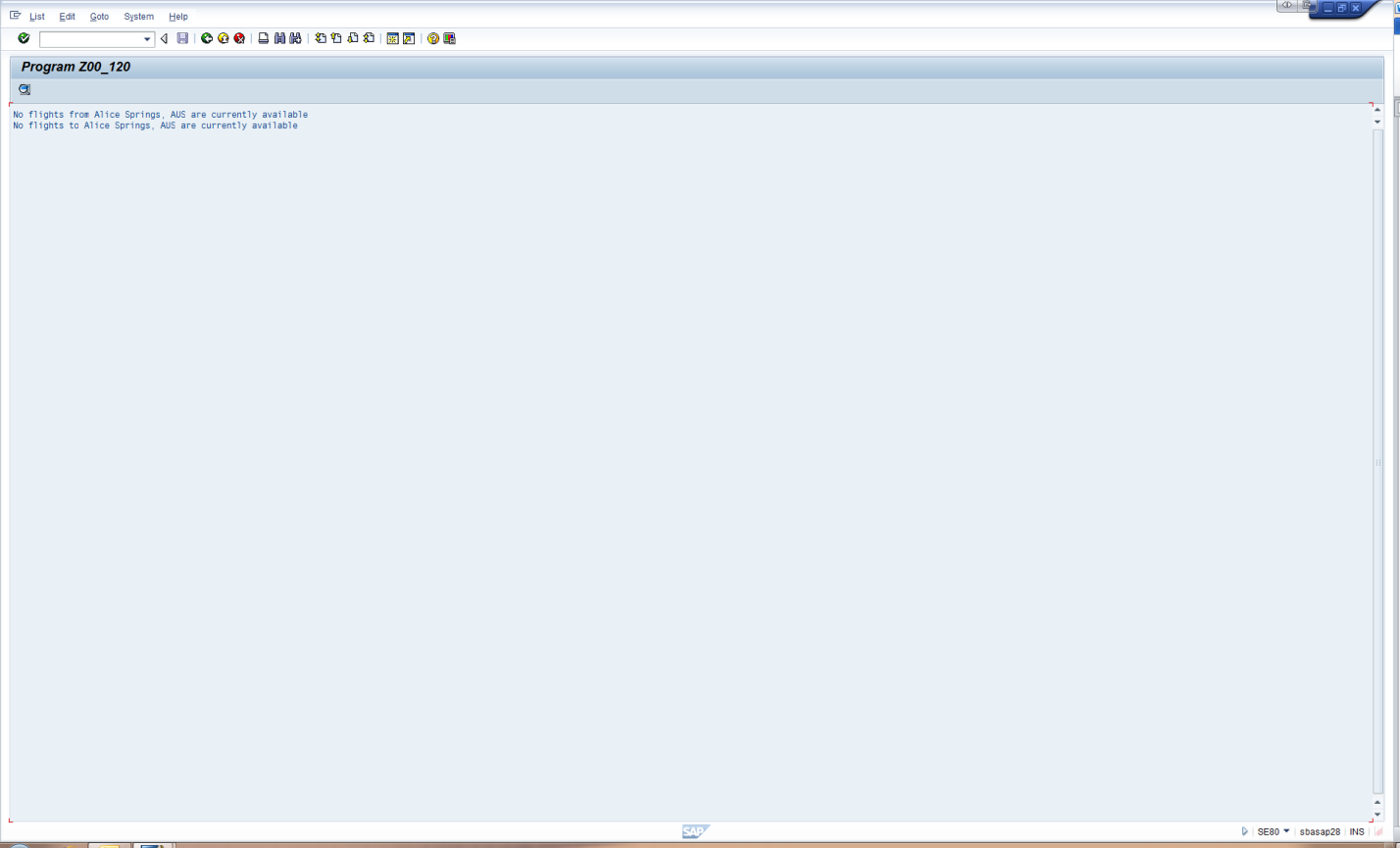
Screen One:

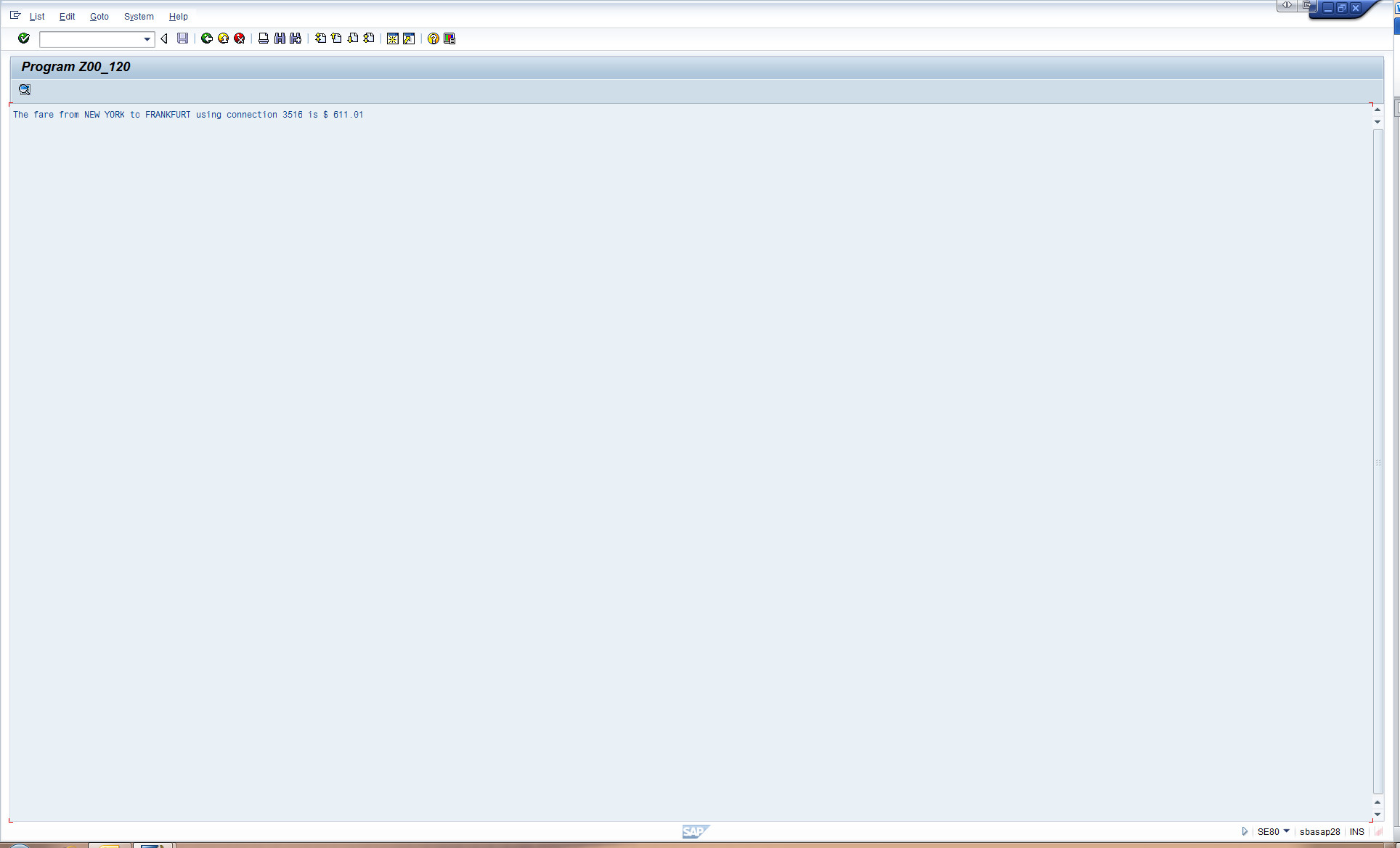
A list of airport codes and their respective airport names is displayed.

When the user double-clicks on or chooses a line, Screen Two is displayed.

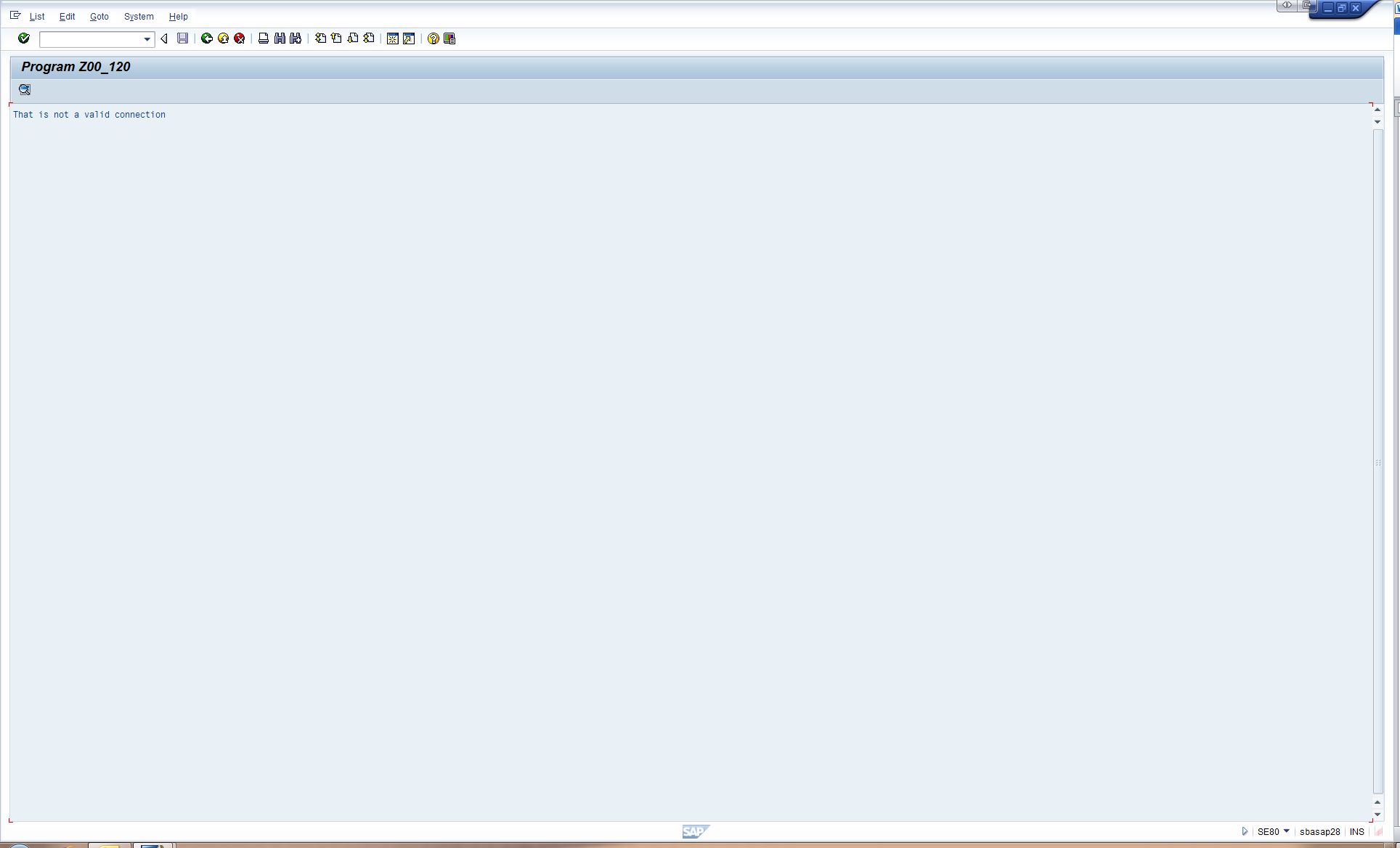
Screen Two:

If flights exist related to the chosen airport, the user is shown the information at the left. Note the separation of the display into 2 lists and the various formatting elements (column alignment, underlining, etc.) Your output should resemble what is shown here.

If no flights exist related to the chosen airport, the user receives the message shown at the left. Note that it is possible for there to be flights from but not flights to or vice versa.

Screen Three:

If the user drills down on a flight in screen two, they can receive fare information as shown above.

Be sure to manage places where the user might try to drill down improperly. Make sure your program displays valid output, such as the following.

Note the following supplemental information:

* For each list item that supports drilldown, hide only the minimal data necessary to achieve the drilldown operation. (Consider the query to be executed at the “next level”.)
* Create needed data objects in your event handers only as needed. (i.e., Don’t create the data objects for screen 3 until/unless the user actually drills down from screen 2.)
* Be very attentive to spacing in your output. Make your screens resemble those shown in the prototype.
* There are multiple fare records for each connection in the database (reflecting different flight dates), but each of those records have the same fare. Use a database query for retrieving the fare that best reflects this situation.
* Be sure to think about how to handle what happens if (a) the user drills down from screen 3 and/or (b) the user double clicks on things like the header lines on screen two.

**Submit to the Lab 6 D2L dropbox the standard submission elements by the due date posted in D2L.**