COMP212 Programming 3
2021W - Test2
Implement WPF app to access/manipulate data of Database through EF6

Rubric: Total marks = 25

Cases
Database-First concept must be enforced. That is reverse engineering from database to code must be done first.
When you start your app, it should right away show the initial data in the relevant DataGrid: Fruit related info in DataGrid-1 (left DataGrid), Planet related info in DataGrid-2 (middle DataGrid). This initial data should come from the database tables created from the given .sql file.
Access and manipulation of database tables from the GUI can be done.
Three DataGrid, two ComboBoxes, six Buttons are present in the GUI layout.
When you select an item in Fruit ComboBox, the selected item gets appended in a new row in DataGrid-1.
When you select an item in Planet ComboBox, the selected item gets appended in a row in DataGrid-2.
When you select an item in the DataGrid-1, the selection (if any) at DataGrid-2 should get unselected.
When you select an item in the DataGrid-2, the selection (if any) at DataGrid-1 should get unselected.
An element in DataGrid (i.e. a grid location) should not be editable.
On clicking button “LINQ Project QS” shows relevant rows in DataGrid-3 (right DataGrid). “LINQ Project QS” implies executing LINQ command Project in query syntax.
On clicking button “LINQ Filter QS” shows relevant rows in DataGrid-3. “LINQ Filter QS” implies executing LINQ command Filter in query syntax.
On clicking button “LINQ Order Ascending QS” shows relevant rows in DataGrid-3. “LINQ Order Ascending QS” implies executing LINQ command **Order Ascending** in query syntax.
On clicking button “LINQ Inner Join QS” shows relevant rows in DataGrid-3. “LINQ Inner Join QS” implies executing LINQ command **Inner Join** in query syntax.
Button “Clear” clears all three DataGrids, clears the relevant entries in the database, clears the selections in every ComboBox.
Button “Delete selected row” deletes the selected row in DataGrid-1 (left DataGrid) **OR** the selected row in DataGrid-2 (middle DataGrid), clears the relevant entry in the database.

Button “Delete selected row” should not delete any selected row in DataGrid-3 (right DataGrid).
When DataGrids are empty (i.e. they have no rows of items), ComboBoxes should be in reset condition (i.e. they should have no item selected)
When DataGrids are empty, clicking any button should have no effect on the GUI.
GUI should be resizable.

Submission of Test-2:

- This test must be completed individually.
- Submit your solution **through Quiz system**. You must name your submission according to the following rule: **studentID(yourlastname)_Test2.zip**

Evaluation Focus:

- Correct implementation of requirements
- Explanation of solution through demo video

Develop a WPF application that interacts with database through Entity Framework 6 (EF6)

1. There is a **.sql** file provided to you. You must use the given **.sql** file first to enforce Database-First concept.

2. When you start your app, it should right away show the initial data. This initial data should come from the database tables created from the **.sql** file.

3. Below are several pictures of how the structure of your app should look. **The rubric in Page 1 of this document tells you how your app should work. Read the figure captions carefully as well;** they also provide hint on how the app should work. Examples of names and colors of fruit and planet entries are shown in the pictures.

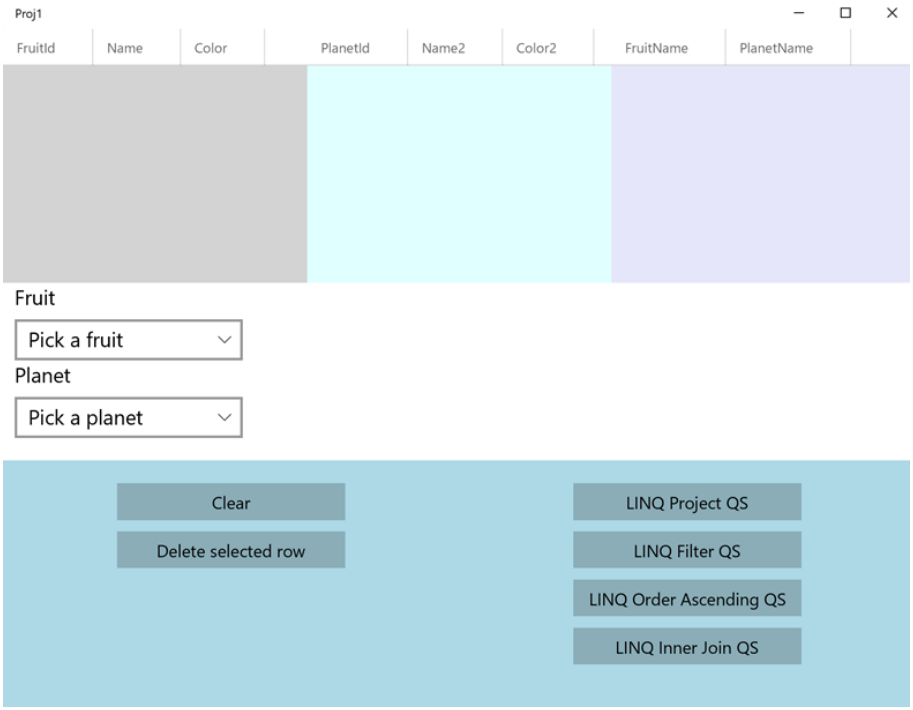

Figure 1. The App when launched. What is not shown here is the initial data. Note that your GUI must show initial data in the relevant DataGrids when the App is launched. This initial data should come from the database tables created from the .sql file.

4. There are Three DataGrids (in top row), two ComboBoxes (in middle row), six Buttons (in bottom row) present in the GUI layout. **Refer to the rubric for more information**.

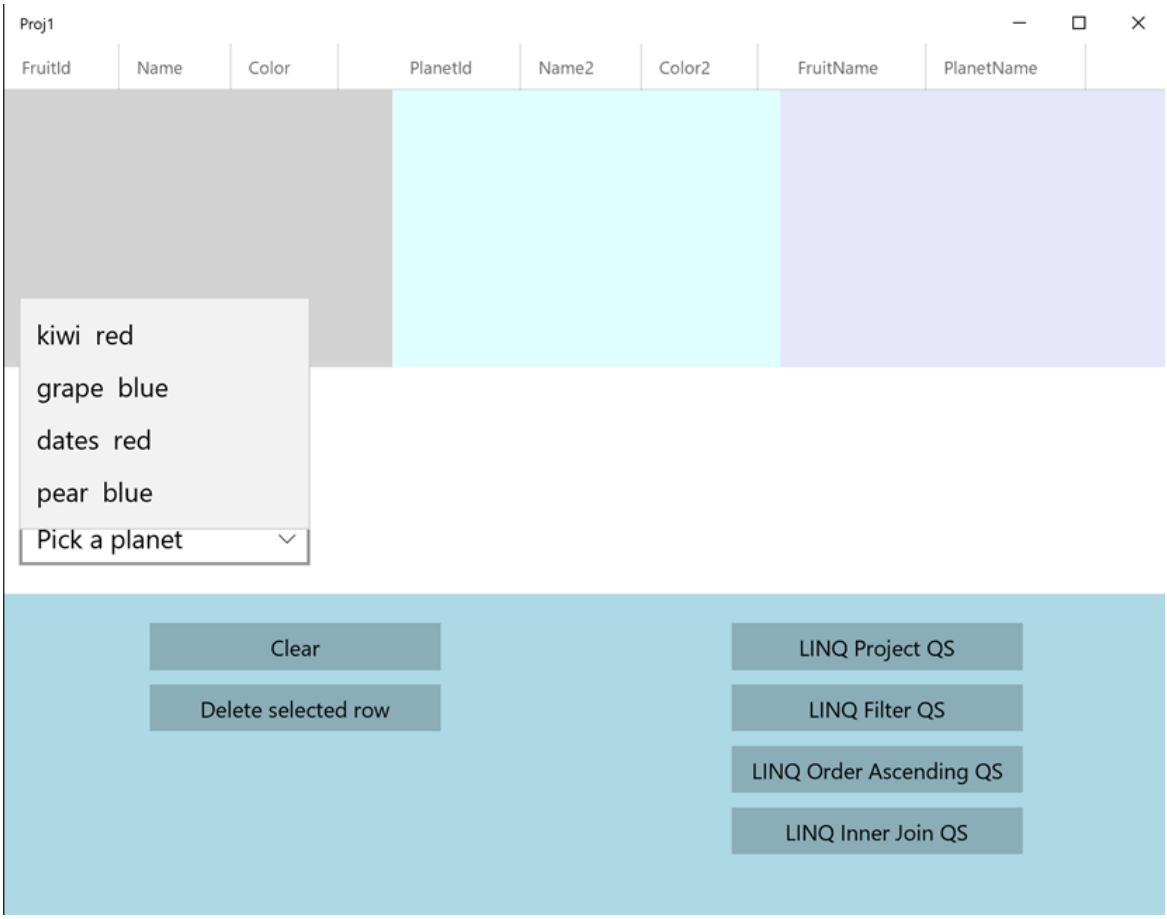

Figure 2. Example of Items on clicking Fruit ComboBox.

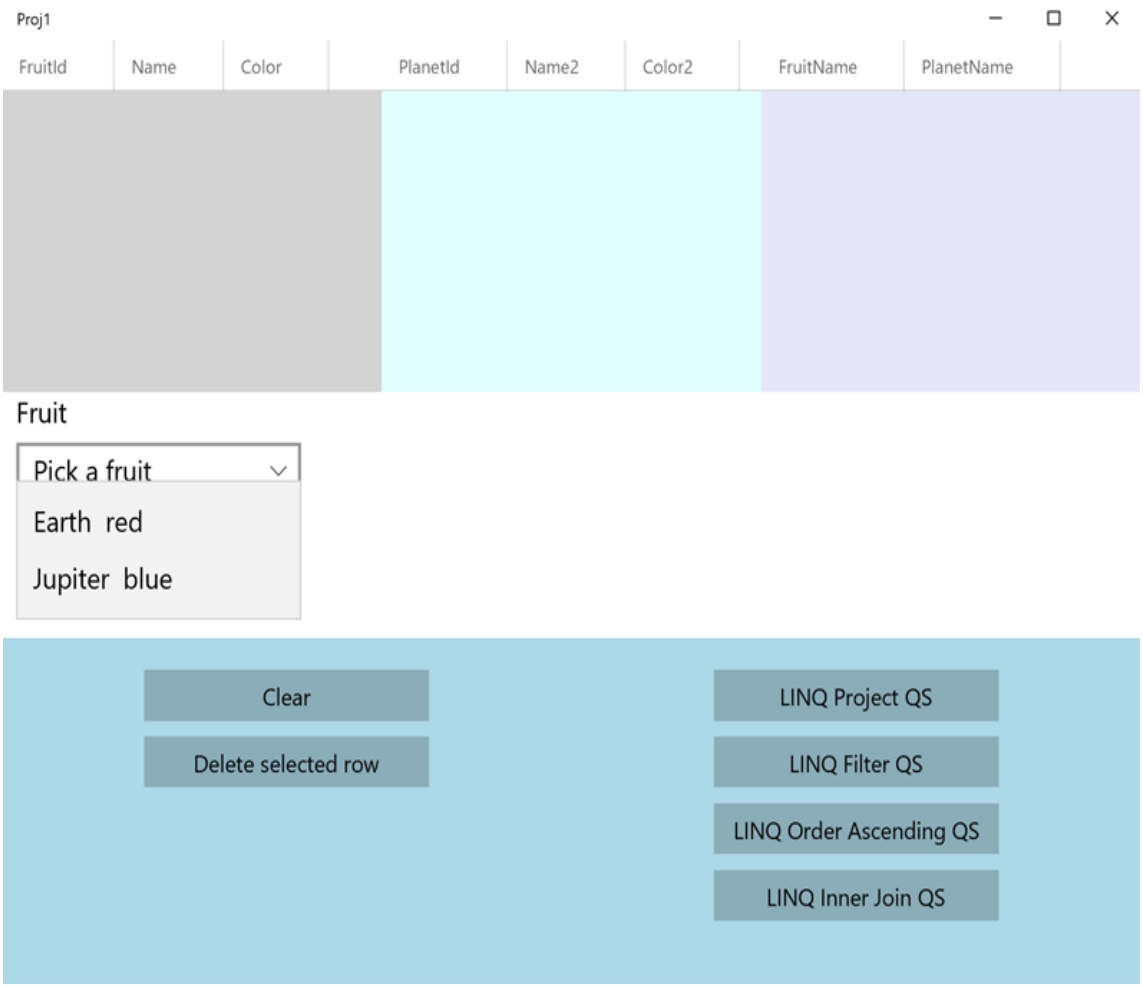

Figure 3. Example of Items on clicking Planet ComboBox.

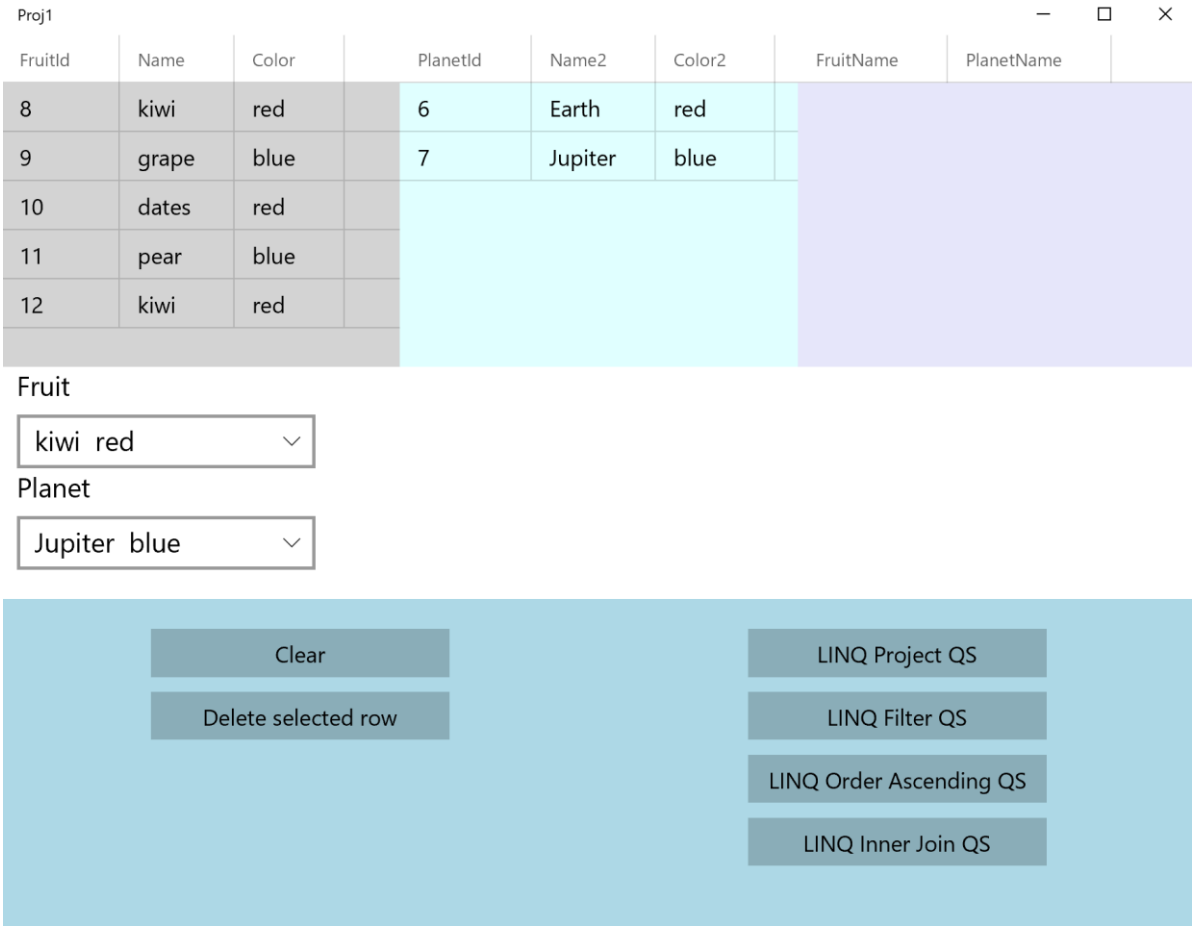

Figure 4. An example set of items in DataGrid-1 (left datagrid) and DataGrid-2 (middle datagrid). **Note:** The **FruitId** value **MUST NOT** be set by you. It should be automatically created for you when a Fruit object is saved in the database as a row. Same argument holds for the **PlanetId** of a Planet object.

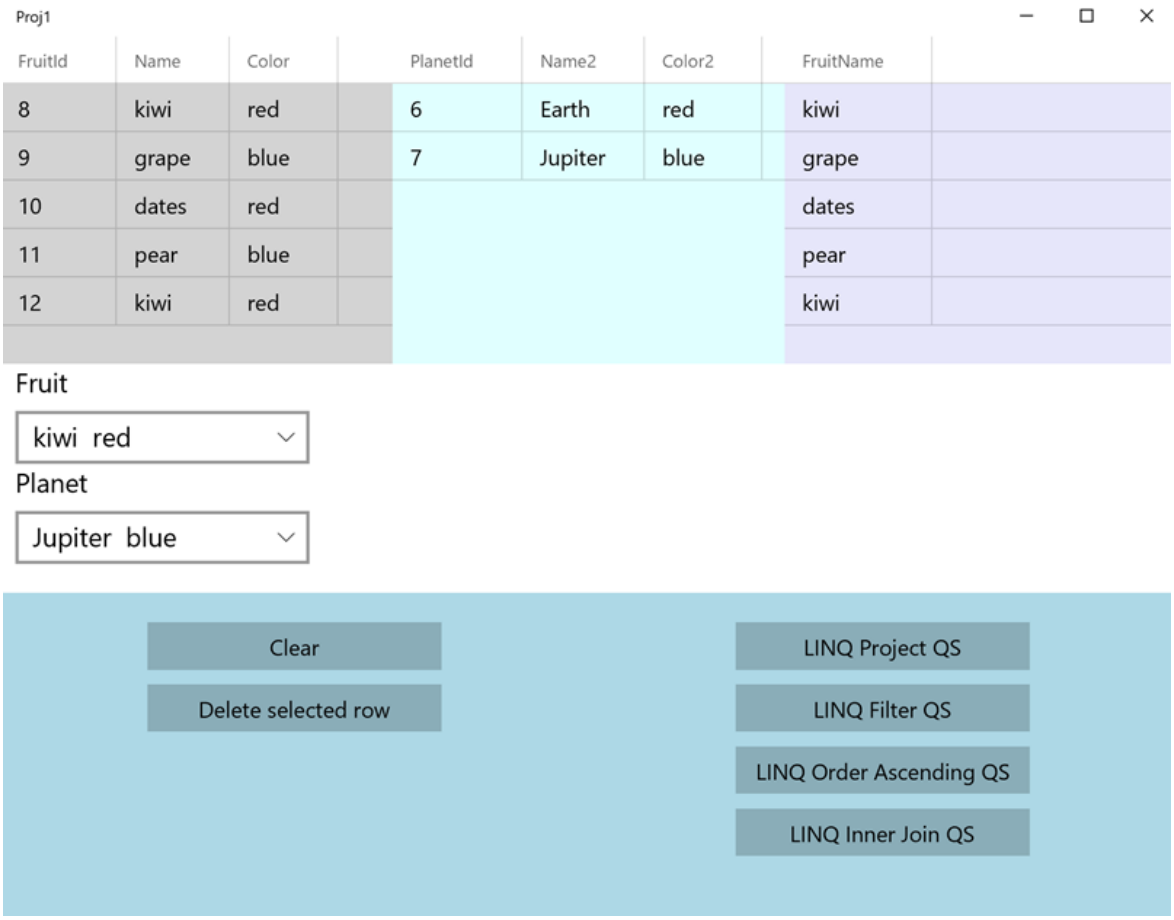

Figure 5. The relevant items in DataGrid-3 (right datagrid) on clicking button "LINQ Project QS" (given the example set of items).

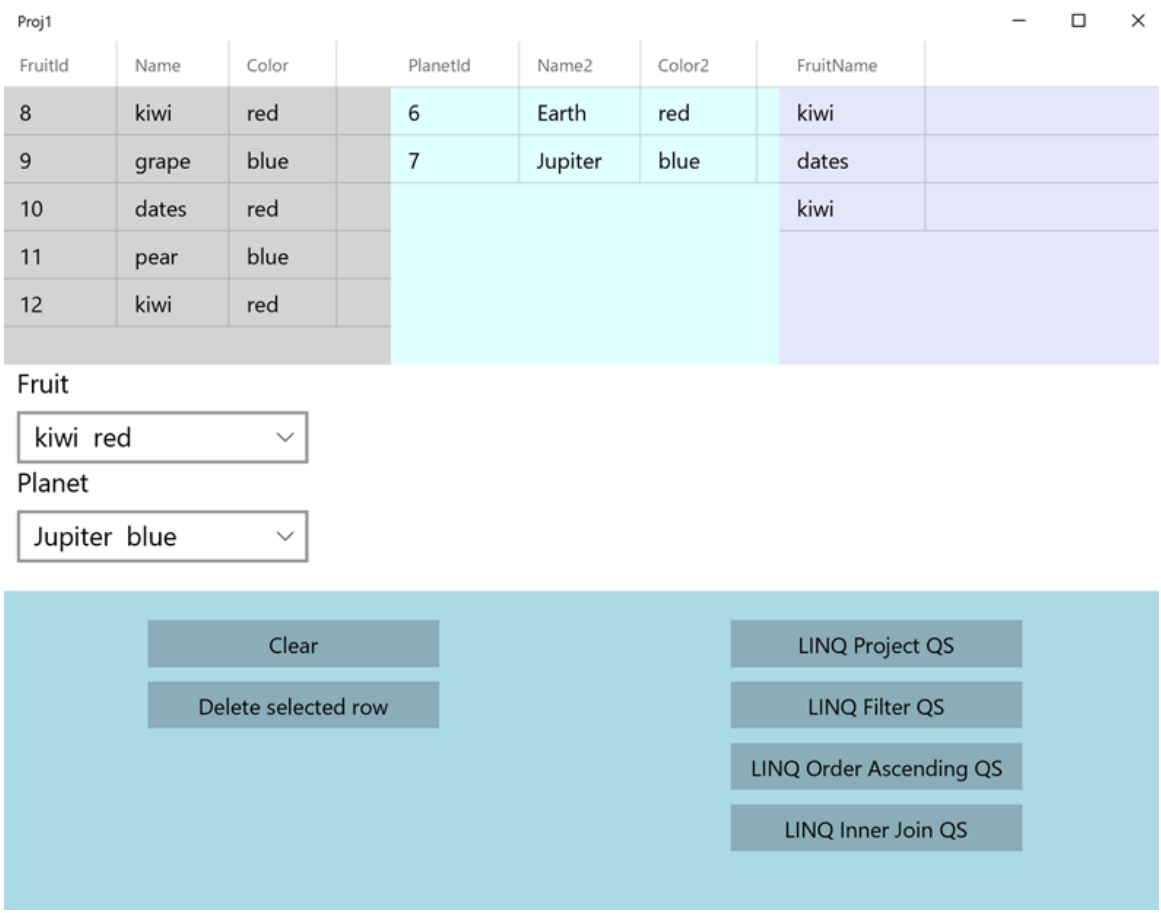

Figure 6. The relevant items in DataGrid-3 (right datagrid) on clicking button "LINQ Filter QS". Here the "filter" is the color **red**.

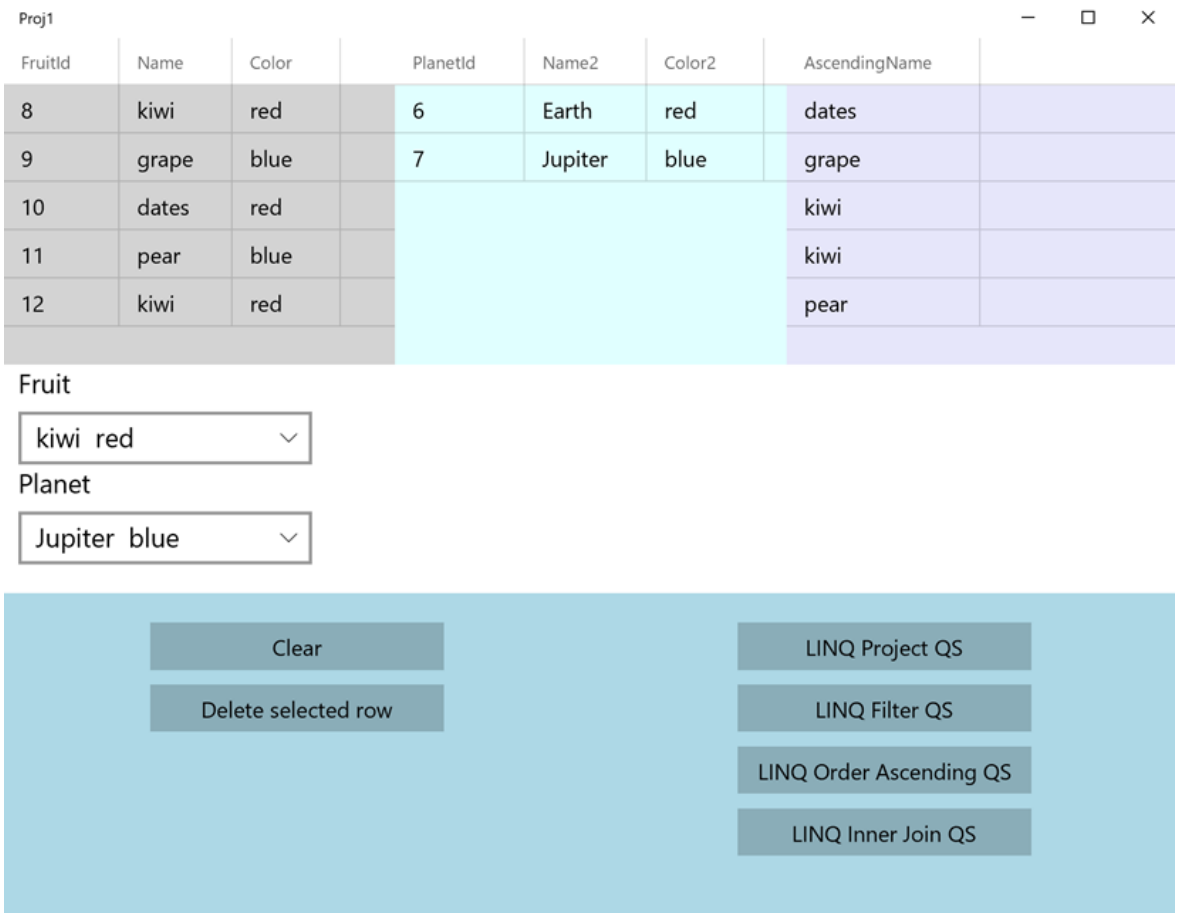

Figure 7. The relevant items in DataGrid-3 (right datagrid) on clicking button "LINQ Order Ascending QS".

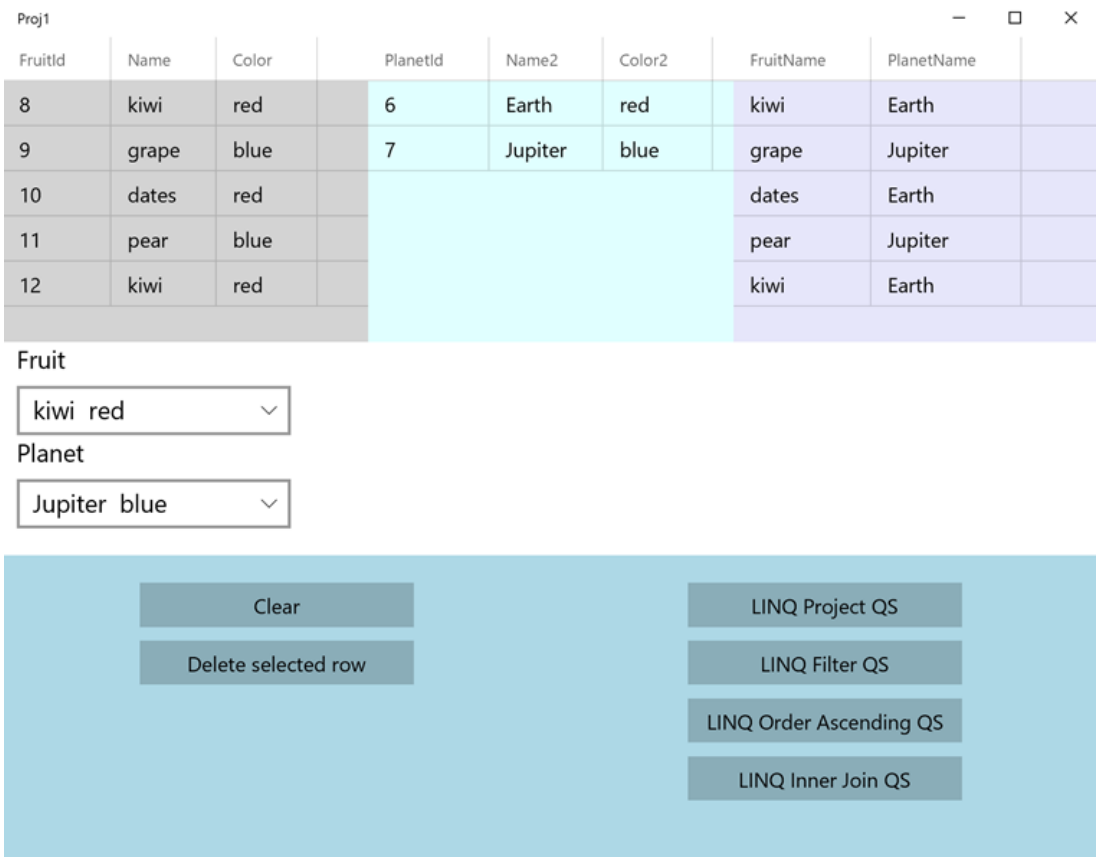

Figure 8. The relevant items in DataGrid-3 (right datagrid) on clicking button "LINQ Inner Join QS".