Manual framework (prefSQL )

* Entry point for the Framework
  + prefSQL.SQLParser.SQLCommon
* Parse a prefSQL String to a usable SQL String
  + Parse a string with the “parsePreferenceSQL” function.

SQLCommon parser = new SQLCommon();

string strSQL = parser.parsePreferenceSQL(strPrefSQL);

* Define the skyline algorithm
  + Select the algorithm with the attribute “SkylineType”.

SQLCommon parser = new SQLCommon();

parser.SkylineType = new SkylineBNLSort();

string strSQL = parser.parsePreferenceSQL(strPrefSQL);

* + The following algorithms are available:

SkylineSQL //Works with ANSI-SQL syntax

SkylineBNL //Block nested loops (supports incomparable tuples)

SkylineBNLSort //Block nested loops with presort (supports incomparable tuples)

SkylineDQ //Divide and Conquer

SkylineHexagon //Hexagon Augsburg (supports incomparable tuples)

MultipleSkylineBNL //Multiple Skyline (define levels with SkylineUptoLevel variable)

* Show Skyline attributes
  + In some cases (i.e. to create a dominance graph) it might be helpful to have the skyline values in the select list. Use the option “ShowSkylineAttributes” for that.

SQLCommon parser = new SQLCommon();

parser.ShowSkylineAttributes = true;

string strSQL = parser.parsePreferenceSQL(strPrefSQL);

* Return multiple skylines
  + SkylineUpToLevel limits the amount of skylines (a field is added to the select list with the level of the tuple)

SQLCommon parser = new SQLCommon();

parser.SkylineType = new MultipleSkylineBNL();

parser.SkylineUpToLevel = 3;

string strSQL = parser.parsePreferenceSQL(strPrefSQL);

* Return the datatable
  + The method “parseAndExecutePrefSQL” returns the datatable instead of a string. The advantage is that it does not rely on MS SQL CLR.

SQLCommon parser = new SQLCommon();

string cnn = "Data Source=localhost;Initial Catalog=eCommerce;Integrated Security=True";

string driver = "System.Data.SqlClient";

parser.SkylineType = new SkylineSQL();

DataTable dt = parser.parseAndExecutePrefSQL(cnn, driver, strPrefSQL);