LINQ – Querying DBLP Data

DBLP is an online bibliographical database for computer science containing around 1 million references. Its content is publically available in XML format.

Since this content is more than 400 MB a small excerpt of this data will be used for this exercice.

The DBLP collection follows the BibTeX format and contains the following types of references: article, inproceedings, proceedings, book, incollection, phdthesis, masters-thesis, and www. The fields describing the above types of references are the following: author, editor, title, booktitle, pages, year, address, journal, volume, number, month, url, ee, cdrom, cite, publisher, note, crossref, isbn, series, school, and chapter. Notice that the not all fields are allowed in all reference types; please refer to the DTD file for this information.

Write the LINQ code for the following queries:

1. The types of publications in the file
2. The number of publications of each types
3. The list of author names
4. The number of authors
5. The list of author names that are also editors
6. The number of publications by author
7. The authors ordered by the number of publications, in descending order
8. The author(s) having the highest number of publications
9. Give for each author the number of publication types
10. Give for each author the total number of publications and the number of publications by type
11. The list proceedings that have at least one editor that is also author of at least one article in the proceedings
12. Give for each author the number of co-authors and the number of joint publications with each of them
13. For each proceedings give its title and the titles of articles appearing in it
14. Give the transitive co-authors of Frank Manola
15. Give the distance of Frank Manola with respect to other authors. Two authors that write together a publication have distance 0. If an author a write a publication with author b and if author b write a publication with author c, then a is at distance 1 from c if a and c have not published together. If an author a is at distances d1 and d2 from an author b, where these distances are obtained following different paths, the minimum value of d1 and d2 will be given as answer.

Solutions

```
var xml = XElement.Load
   (@"C:\Users\sboucher\Desktop\dblp-small.xml");

var publications = xml.Elements();
var q1 = (from p in publications select p.Name).Distinct();

var q2 = from p in publications group p by p.Name into type
         select new {
           Type = type.Key,
           Count = type.Count()
         };
var authors = (from a in publications.Elements("author")
               select a.Value.ToString()
              ).Distinct();
var q3 = authors;
var q4 = q3.Count();
var editors = (from e in publications.Descendants("editor")
               select e.Value.ToString());
var q5 = from author in authors
         where editors.Contains(author)
         select author;
var author_publications = from author in authors
                          let ap = (from p in publications
                                    where (from a in p.Elements("author")
                                           select a.Value.ToString()
                                          ).Contains(author)
                                    select p)
                                   select new {
                                     Name = author,
                                     Publications = ap
                                   };
var q6 = from ap in author_publications
         select new {
            Name = ap.Name,
            PublicationsCount = ap.Publications.Count()
         };
var q7 = from ap in q6
         orderby ap.PublicationsCount descending
         select ap;
var q8 = from ap in q6
         where ap.PublicationsCount == (from ap2 in q6
                                        select ap2.PublicationsCount).Max()
         select ap;
var q9 = from ap in author_publications
        select new {
          Name = ap.Name,
          PublicationsTypeCount = (from p in ap.Publications
                                   select p.Name.ToString()
                                  ).Distinct().Count()
        };
```

```
var q10 = from ap in author_publications
          select new {
            Name = ap.Name,
            Count = ap.Publications.Count(),
            Breakdown = (from p in ap.Publications
                         group p by p.Name into g
                         select new {
                           Type = g.Key,
                           Count = g.Count()
                         })
        };
var q11 = from p in publications
          where (from editor in p.Elements("editor")
                 where (from p2 in publications
                        where (from a in p2.Elements("author")
                               select a.Value.ToString()
                              ).Contains(editor.Value.ToString())
                       &&
                       p2.Attribute("crossref") == p.Attribute("key")
                  select p2).Any()
              select editor
        ).Any()
        select p;
var q12 = from ap in author_publications
        select new {
         Name = ap.Name,
         CoAuthors = (from coauthor in
                        (from p in ap.Publications
                         from a in p.Elements("author")
                         where  a.Value.ToString() != ap.Name
                         select a.Value.ToString()).Distinct()
                      select new {
                        CoAuthorName = coauthor,
                        Count = (from pub in ap.Publications
                                 where(from a in pub.Elements("author")
                                       select a.Value.ToString()
                                      ).Contains(coauthor)
                                 select pub).Count()
                      })
        };
var q13 = from proceeding in xml.Elements("proceedings")
          select new {
            Title = proceeding.Element("title").Value.ToString(),
            Articles = (from  inproceeding in xml.Elements("inproceedings")
                        where
                         inproceeding.Element("crossref").Value.ToString()
                         == proceeding.Attribute("key").Value.ToString()
                        select inproceeding.Element("title")
                                           .Value.ToString())
           };
```