CSCI 585 – Database Systems – Fall 2014

Homework Assignment 3

Due: December 2, 2014 @ 12.00 PM (noon) PST

[Electronic submission via D2L]

In this assignment you will explore the design and implementation of XML Schemas, XML Stylesheets and the use of the XML query language XQuery to query XML data.

While working on this assignment, you can use one of the following XML editors:

1. Altova XMLSpy 2015 (for Windows):
    - Download: http://www.altova.com/download.html
    - Documentation: http://www.altova.com/solutions.html
2. EditiX XML Editor 2014 SP1 (for Windows, Mac & Linux)
    - Download: http://www.editix.com/download.html
    - Documentation: http://www.editix.com/doc/manual12/index.html

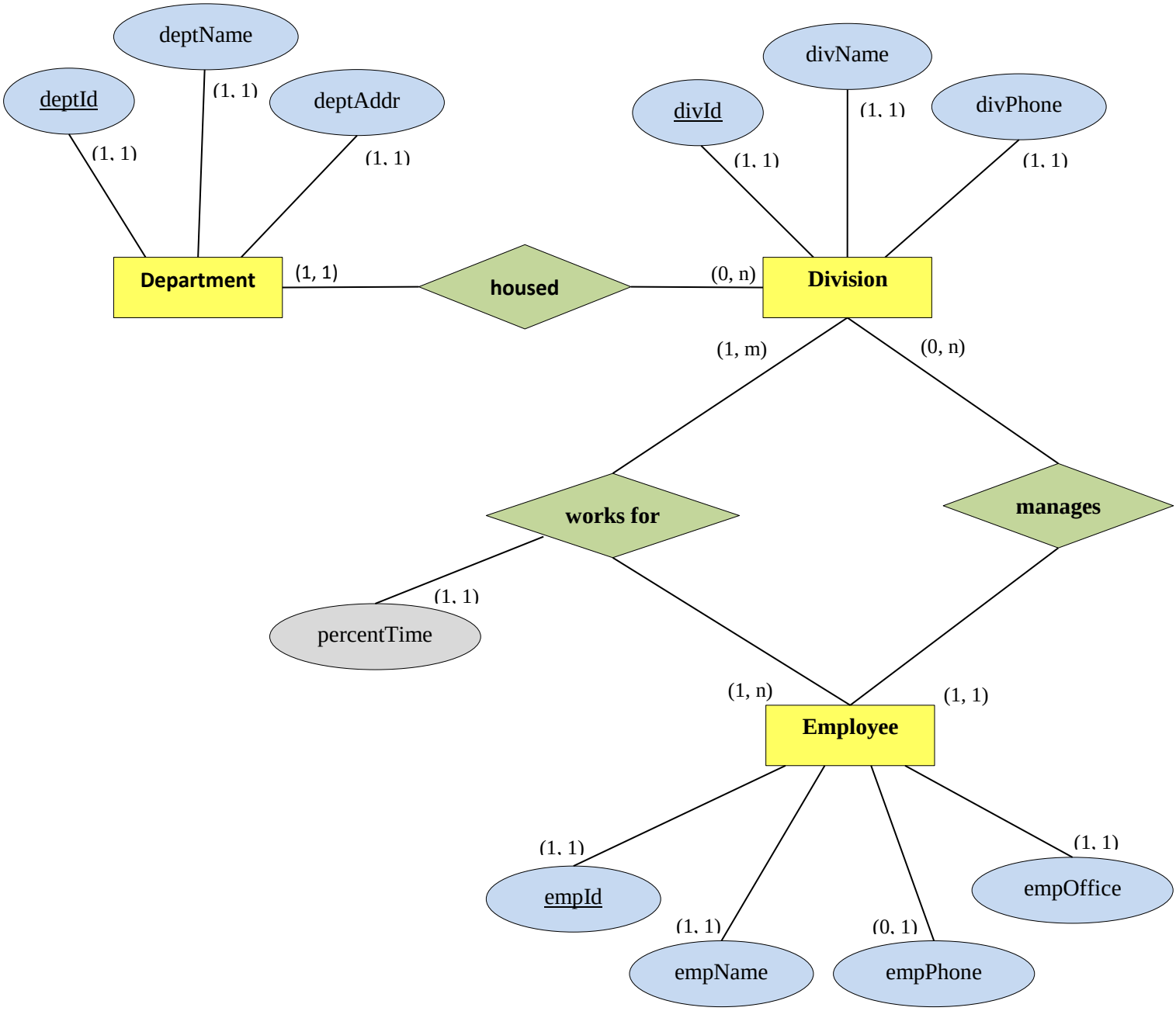
Part 1: Design XML schema (15 points)

A company XYZ has several departments. Each department houses many divisions. However, each division is housed in a single department. Each employee in the company can work for one or more divisions. The WorksFor relation lists the percent-time an employee works-for each Division. Consider the following schema for the company:

Department (deptId, deptName, deptAddr)
Division (divId, divName, divPhone, housedDeptId, managerEmpId)
    FK housedDeptId references Department
    FK managerEmpId references Employee
Employee (empId, empName, empPhone, empOffice)
WorksFor (divId, empId, percentTime)
    FK divId references Division
    FK empId referenced Employee

The underlined attributes in the above schema represent primary keys.

Following is the ER diagram depicting the above schema:



The cardinalities shown near the attributes describe the minimum and maximum number of values of the attribute associated with each instance of an entity. For example, cardinality (1, 1) shown near deptId attribute of Department implies that deptId is a required attribute and has 1 occurrence.

Following are the format restrictions that apply to the attributes, which should be reflected in the XML schema design:
Format for deptId: 3 UPPERCASE letters from A-Z
Format for phone number: xxx-yyy-zzzz, where x, y, and z are numbers and xxx does not start with a "0"

Use the information from the schema and the ER diagram to create a valid XML Schema `company.xsd`. Make sure to use appropriate data types, including defining your own complex types as applicable.

Part 2: Create XML data file (10 points)

Create a valid XML data file `company.xml` based on the XML Schema in part 1 for the data given in the Excel file, "company_data.xslx". Department, Employee, Division, and WorksFor data is provided in different tabs of the Excel workbook.

Part 3: Design XML stylesheet (15 points)

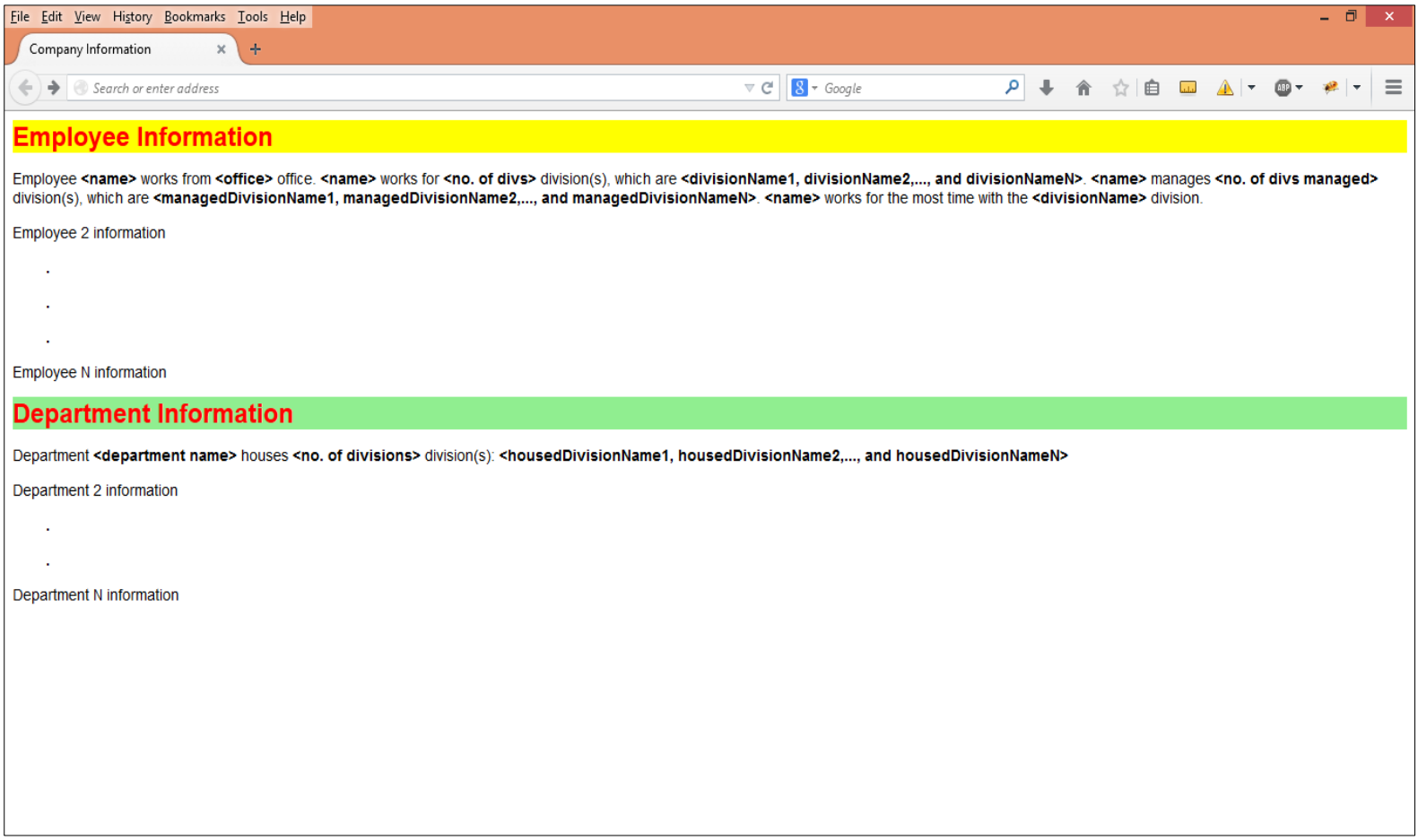
Using the XML Schema in part 1, create a valid XML stylesheet `company.xsl` so that the XML file you created in Part 2 can be displayed in the format given below in **Firefox** browser.

Specifications

1. Employee information header:
    a. font-size: 24 pixels, background-color: Yellow, font-weight: bold, color: red.
2. Department information header:
    a. font-size: 24 pixels, background-color: LightGreen, font-weight: bold, color: red.
3. Font to be used for the entire page is Arial, and except the headers mentioned above, rest of the text should have a font-size of 12 pixels.
4. Text to be displayed for each employee:
    a. Employee **<name>** works from **<office>** office. **<name>** works for **<no. of divs>** division(s), which are **<divisionName1, divisionName2, ... , and divisionNameN>**. **<name>** manages **<no. of divs managed>** division(s), which are **<managedDivisionName1, managedDivisionName2, ... , and managedDivisionNameN>**. **<name>** works for the most time with the **<divisionName>** division.
    b. Replace the text in the angular brackets with appropriate values, and display the text without the angular brackets. For example, sentence "Employee **<name>** works from **<office>** office." should be displayed as "Employee **PSmith** works from **LA** office."
    c. Display replaced text in bold in the output.
5. Text to be displayed for each department:
    a. Department **<department name>** houses **<no. of divisions>** division(s): **<housedDivisionName1, housedDivisionName2, ..., and housedDivisionNameN>**.

b. Replace the text in the angular brackets with appropriate values, and display the text without the angular brackets similar to the instruction in 4b.
c. Display replaced text in bold in the output.

6. **Note:** Make sure to pretty format the sequences of divisions with "and" displayed before the last division. For example, displaying "…works for **<no. of divs>** divisions, which are **d1, d2, d3,** ." is not pretty formatted. It should be displayed as "…works for **<no. of divs>** divisions, which are **d1, d2, and d3**."



Part 4: Query processing using XQuery (60 points)

Using XQuery, query your `company.xml` file to answer the following. You should submit the respective query files (not the results!) as `query1.xquery` to `query6.xquery`. Enclose the results of each of your queries in an appropriate XML root tag, e.g.:
`<query1> ... </query1>`

1. Find the names of all departments that house a division in which an employee named "PSmith" or "Jack" works for at least 50% time.

2. Find names of divisions that EVERY employee works for.

3. Find names of the departments that house at least one division managed by "PSmith" but no divisions managed by "Wong".

4. Find employees who do not manage any division in the company. Display their names, phone numbers, and offices, ordered by name in descending order.

5. Find the average number of divisions that each employee works for. (E.g.: if the company has 2 employees, with emp1 working for 4 divisions and emp2 working for 3 divisions, then the average number of divisions = (4+3)/2 = 3.5)

6. Find the name, phone, and office of the employee who works for the maximum number of divisions.

Submission Guidelines:

1. You should have the following files in your submission:
   a. `readme.txt` – Containing your name, id and email address. It should also contain a list of all the submitted files.
   b. XML Schema: `company.xsd`
   c. XML Data: `company.xml`
   d. XML Stylesheet: `company.xsl`
   e. Six XQuery files: `query1.xquery, ..., query6.xquery`

2. Compress the above mentioned files into a single zip archive with the filename format `lastname_firstname_hw3.zip` and submit via D2L only. Email submissions will not be accepted. Failure to adhere to filenames and/or file formats will result in a penalty of 10 points.

3. Do **not** submit the XQuery results. We will run your queries to obtain the results.

4. It is ok to modify and re-submit your entire file, but **only until the deadline. Late amendments are not accepted.** Only the most recent on-time submission is graded.

Useful references:

1. w3schools.com has tutorials on XML, XML Schema, XPath, XSLT and XQuery
2. http://www.xml.com/
3. XML: http://www.w3.org/TR/xml/
4. XML Stylesheet: http://www.w3.org/TR/xml-stylesheet/
5. XML Schema: http://www.w3.org/TR/xmlschema-1/
6. XQuery: http://www.w3.org/TR/xquery/