CS549 Distributed Information Systems - Coursework Assignment

Group Project: Submission Deadline 28th/March/2024

This coursework is a group project that requires each group to design and implement data models for a given scenario as the following:

- **Design and implement relational database for the given scenario.**
- **Design and implement XML file for the given scenario.**
- **Design and build an ontology for the given scenario.**
- **Write SQL, XQuery and SPARQL queries to answer the given five business requirements.**
- **Write one paragraph (Max.150 words) to state some extra knowledge that can be retrieved from the ontology but not from the relational or XML data models**

By doing so, students can develop a comparative understanding of how to design a distributed information system by designing different data models that can be used to store, integrate, and share data.

This is a group project and must be accomplished within collaboration among each group members. However, collaboration, collusion or the **sharing of solutions among groups are not allowed**. As the submission will go first **to Turnitin (a plagiarism checker)** which will check the submission similarities with others. So, it is important that each group keeps its own work in unshared. Failure to follow this instruction will result in disciplinary action being taken.

Consider The following Business Scenario:

A Software development company has several departments such as marketing, HR, Programming, etc. Each department has employees, and an employee can be either on an admin-based job or a software-based job such as software engineer, frontend developer, full stack developer, etc. For example, John Smith is an employee on a software-based job, and his role is a software engineer. Similarly, Kate Smith is an employee on an admin-based job, and she is a finance manager. An employee on a software-based job can be assigned an admin role, such as a software engineer, and a project manager. However, this is not valid for admin employees as they can only be assigned an admin role. The company works on software projects, and software projects have a type, such as a Web App, Desktop App, etc. Also, each project has a category, such as educational software, enterprise software, etc. Each project comprises many tasks such as frontend design, frontend development, backend design, etc. Naturally, projects may share tasks, and therefore a task may be found in more than one project. For example, the task frontend design may exist in educational and application projects. Each task has a code that identifies this task, a name, and a given weight that can be between 1 and 5. Each project is assigned to a team (a team is a group of employees on a software-based job). Each team is managed by one of its members, and this member is assigned a project manager role in addition to his/her first role. Each team member is assigned a project task. Each team member may be assigned more than one project task. The project-record stores information about the monthly progress of each task of the project and hence the project progress. A project-record includes data such as project ID, task code, employee ID, month ID, and completion percentage. Any task is completed when the sum of its completion percentage is 100%. Similarly, a project is completed when each of its tasks has a completion percentage equal to 100%. Naturally, some projects can take more than a year to be completed so it is possible to understand the annual completion percentage for each project task based on the data stored in the project-record. Each member of staff is based in an office where many staff can share the same office.

The given business scenario focuses on the main and important aspects of the information system flows. However, there are some attributes that are not mentioned but they are necessarily important and exist. For example, the above scenario does not state that each employee has a unique emp_ID,

however, naturally, this information has to be included as you analyse the above scenario. In a similar way, there are some important attributes that you will discover and have to include them while analysing the scenario. This helps acquiring important skills that enable analysing real-world business scenarios and thus design and implement a distributed information system.

1. Coursework requirements

Each group is required to accomplish the following steps:

1- Design, draw, and implement an ERD (states all the **entities**, **relationships,** and **attributes** for each entity). Then. insert test data[1] using SQL statements (five rows in each entity). Finally, write SQL queries that will answer the given five business requirements.
2- Design an XML file and **its DTD** so that your XML file should be valid against its own DTD contained in it. Also, populate this XML file with test data[1] then write XQuery queries that will answer the given five business requirements.
3- Design and develop an Ontology, insert instances (test data[1]) then write SPARQL queries that will answer the given five business requirements.
4- State some extra knowledge (Max 150 words) which can be retrieved from the developed ontology but not from the relational or XML data models.

The test data to be used in Step 1, Step 2, and Step 3 should allow you to get similar results when writing queries. For example, consider the test data about employees. The five employees stored in the relational DB should be the same employees stored in the XML file. Also, they are the same employees stored as individuals in the developed ontology. Therefore, an SQL, XQuery, and SPARQL query to answer the first business requirement will give you the same results.

The **five business requirements** are:

1. List the ID and name of all employees.
2. List the name of all projects together with their type.
3. List the ID and name of all employees together with the project name and project category that they work on.
4. List the ID and name of all tasks together with the name and ID of the employee who works on this task.
5. List the ID and name of all employees together with the office number that they are based in.

2. Preparing for submission

The submission of your coursework is two files (**Please use the file names as shown below**):

1. "courswork_group**ID**.pdf" (change ID to group No): The contents of this PDF file are shown in the template file on Myplace "courswork_tmp.docx".
2. "group**ID**.zip" (change ID to group No)**:** This Zip file must contain the following **six** files:
  2.1. "**sql_db.sql**": An SQL file that contains SQL statements which create and insert data on your relational db.
  2.2. "**sql_query.sql**": An SQL file that contains five SQL queries to answer the business requirements from the relational database.
  2.3. "**file_db.xml**": An xml file with both DTD and **test data** included.
  2.4. "**xquery.txt**": A text file that contains XQuery queries to answer the business requirements from the XML file.
  2.5. "**Ont.owl**": A file that stores the developed ontology in XML/RDF format.
  2.6. "**sparql.txt**": A file that contains SPARQL queries to answer the business requirements from the developed ontology.

Please note that for the purpose of assessment, the six files included in the ZIP file will be used to replicate your work so make sure these are correct.

Once you have the two files (PDF and ZIP files) ready, please use the link provided on Myplace to submit your coursework (PDF and ZIP files) on the Turnitin by 5.00pm on Thursday 28th March.

3. Tools that you may use

- For relational database implementation, test and queries, you may use phpMyAdmin to implement the ERD, insert data, and query your db.
- For XML work, a text editor such as VSCode IDE or Notepad++ may be used to create an XML file with DTD and write test data inside it. To write an XQuery query, you may use the same text editor.
- For ontology work, you should use Protégé (http://protege.stanford.edu/) to design the ontology. To test the SPARQL queries, you can use Protégé as well.

4. Marking Schema

- First, each group will be marked based on the marking scheme shown below. This will result in a **common group mark**. Then, each group member will get a final individual mark that is calculated as (**common group mark) * (peer review ratio)**. The peer review ratio is your contribution to the coursework assessed by the rest of your groups. For example, if your group members assign you the full peer review mark, then your final mark will be the same as your common group mark.

Step No	Mark	70-100%	60-69%	50-59%	<50% fail
Step 1	**9**	An excellent design that captures the stated requirements through modelling the required aspects and can potentially suit possible future requirements. The implementation allows for answering business requirements and other key requirements. The test data used is representative, and the queries are correct. The design, implementation, and queries are replicable from the submission file.	A very good design that captures the stated requirements. However, future requirements are not considered. The implementation allows for answering business requirements. The data used is largely representative and the queries are correct. The design, implementation, and queries are replicable from the submission file.	A good design with some redundancies or omissions, however the business requirements are largely considered. The implementation allows for answering some business requirements. The data used is not representative, and some queries are incorrectly stated. The design, implementation, and queries are replicable, with some errors, from the submission file.	A limited design that captures some business requirements. However, it is unable to answer most of the required business requirements. The implementation is thus limited in terms of answering the business requirements. The test data used is not representative and most of the queries are incorrect. The design, implementation and queries are not replicable from the submission file.
Step 2	**9**				
Step 3	**9**				
Step 4	**3**	The justification about the extra knowledge is sound and supported by the developed ontology design.	The justification about the extra knowledge is very good with some reference to the developed ontology design.	The justification about the extra knowledge is good and partially supported by developed ontology design.	The justification about the extra knowledge is rather arbitrary stated.
	30				

Dr AbdulSalam Kalaji
January 2024