INF01006 - Projeto de Banco de Dados
Plano de ensino

Carlos A. Heuser

2009/1

1 Identificação

Nome do departamento: Informática Aplicada
Nome da atividade de ensino: INF01006 - Projeto de Banco de Dados
Curso de oferecimento: CIC – Ciência da Computação
Pré-requisito: Fundamentos de Banco de Dados
Etapa aconselhada no curso: 4 ^{a}
Corpo docente: Carlos A. Heuser
Créditos/carga horária: 4/4

2 Objetivos

A disciplina tem por objetivo permitir ao aluno aprofundar os conhecimentos de projeto de banco de dados que teve na na disciplina introdutória de BD. Consta da execução de um projeto completo (especificação, projeto e construção) de uma aplicação de BD por grupos de alunos.

3 Conteúdo Programático

1. Definição da aplicação de BD. 2. Definição da metodologia. 3. Especificação de requisitos, modelagem de dados, especificação das transações. 4. Projeto e definição da plataforma de implementação.

4 Metodologia Adotada

A disciplina está montada ao redor de um projeto de porte a ser executado em grupos de dois alunos.

O semestre é dividido em duas etapas.

Na primeira etapa, ocorre um conjunto de aulas expositivas tratando dos problemas que serão resolvidos no projeto (XML e similaridade). Serão propostos exercícios de fixação de conteúdo a ser resolvidos de forma individual.

Na segunda parte da disciplina, os alunos serão divididos em grupos de dois, cada grupo encarregado de um dos projetos da disciplina (ver detalhes dos projetos abaixo). A capacidade de atendimento do professor é de 20 grupos. Assim, se a turma estiver completa (40 alunos) não serão admitidos grupos de um único aluno.

Cada grupo de alunos executará um projeto. Um projeto consta de:

- Definição de um projeto entre os propostos para a disciplina.
- Estudo de um artigo fornecido pelo professor, que descreve o problema a resolver, bem como as técnicas a utilizar em detalhe. Os artigos escolhidos são, em princípio, auto-contidos, mas pode ser útil a leitura de algum artigo referenciado para facilitar a compreensão.
- Definição dos experimentos que serão realizados. Para todos projetos, serão realizados experimentos que objetivam avaliar a eficiência e/ou eficácia da solução usada. Cada grupo deve definir ao menos:
  - Objetivos que tem com os experimentos.
  - Dados a utilizar nos experimentos.
  - Procedimentos a utilizar nos experimentos.
  - Forma pela qual pretende avaliar os resultados dos experimentos.
- Apresentação do projeto e do plano de experimentos para os colegas.
- Projeto detalhado da base de dados e do software para realizar os experimentos.
- Carga da base de dados e realização dos experimentos.
- Apresentação dos experimentos para os colegas.
- Redação de artigo curto (formato ACM, máximo de 6 páginas) relatando os resultados alcançados.

Nesta etapa, cada grupo terá um encontro semanal de 12 minutos com o professor, em horário a ser sorteado, dentro do horário da disciplina. A presença dos dois membros do grupo neste encontro é obrigatória, pois é a forma de avaliar o andamento dos alunos no projeto. O encontro serve para orientar os alunos em seu projeto.

5 Cronograma de Atividades

Aula#	Formato	Conteúdo
1	aula	Apresentação da disciplina - Aplicações típicas de XML e dados WEB
2	aula	XML - o modelo de dados e a DTD
3	aula	Entidades, APIS e namespaces
4	aula	XPath
5	aula	XQuery
6	aula	XML Schema
7	aula	Integração de esquemas e instâncias: funções de similaridade e avaliação de qualidade
8-10	orientação	Discussão do artigo com o professor
11-12	orientação	Discussão do plano de experimentos com o professor
13-14	**apresentação**	**Grupos apresentam seus projetos**
15-18	orientação	Projeto da base de dados, projeto do software, obtenção dos dados para experimentos
19-24	orientação	Implementação da base de dados, implementação do software e primeira execução dos experimentos
25-26	orientação	Discussão dos resultados obtidos
27-28	**apresentação**	**Grupos apresentam resultados**
29-30	orientação	Revisão do artigo

6 Critérios de Avaliação

O conceito do aluno será obtido levando em conta os seguintes quesitos:

Peso	Quesito
10%	Resolução dos exercícios referentes às aulas expositivas.
50%	Qualidade de trabalho realizado demonstrado nas reuniões de orientação ao longo do semestre (qualidade do trabalho, prazos).
20%	Qualidade das apresentações.
20%	Qualidade do artigo apresentado no final da disciplina.

O prazo para recursos é de uma semana após a divulgação das notas.

6.1 Atividades de Recuperação

Pela natureza da disciplina, não há atividade de recuperação.

7 Bibliografia

A bibliografia varia de acordo com o projeto escolhido por cada grupo de alunos. Além dos projetos abaixo, outros poderão ser definidos de comum acordo com os alunos.

7.1 Armazenamento de XML

Neste projeto vamos tratar o problema do armazenamento de grandes documentos XML em uma base de dados relacional. Vamos considerar que o usuário deseje:

1. Armazenar um documento XML completo;
2. Recuperar um documento XML completo;
3. Editar o documento por partes, ou seja, não ler e escrever o documento por completo de cada vez;
4. Usar o suporte do SGBD relacional para executar consultas XPath ou XQuery.

Sobre este assunto há vários artigos na literatura (todos estão disponíveis em PDF no site Moodle da disciplina.

- Artigo com várias alternativas para armazenar XML. Artigo básico e referência para os demais.
  Daniela Florescu - Donald Kossmann, *A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database*, Technical Report, INRIA, Maio 1999

- Popõe alternativas de armazenamento levando em conta a execução de consultas XPath (na realidade, de um precursor de XPath, chamado XML/QL).
  Jayavel Shanmugasundaram et al. *Relational Databases for Querying XML Documents: Limitations and Opportunities* 25th VLDB Conference, Edinburgh, Scotland, 1999
  http://www.vldb.org/conf/1999/P31.pdf

- Artigo que demonstra a tradução de XPath para SQL.
  Li, Q. and Moon, B. 2001. *Indexing and Querying XML Data for Regular Path Expressions*. In Proceedings of the 27th international Conference on Very Large Data Bases (September 11 - 14, 2001).
  http://www.vldb.org/conf/2001/P361.pdf

- Este artigo mostra um esquema para traduzir XQuery para SQL.
  David DeHaan, David Toman, Mariano P. Consens, M. Tamer Özsu; *A Comprehensive XQuery to SQL Translation using Dynamic Interval Encoding*. SIGMOD 2003, June 9-12, 2003, San Diego, CA.
  http://portal.acm.org/citation.cfm?id=872757.872832

- Este artigo introduz o problema de armazenar documentos XML em que os elementos estão *ordenados*.
  Igor Tatarinov; *Storing and Querying Ordered XML Using a Relational Database System*. ACM SIGMOD'2002, June 4-6, Madison, Wisconsin, USA.
  http://portal.acm.org/citation.cfm?id=564691.564715

7.2 Detetando mudanças em documentos

Um problema que aparece em muitas aplicações XML é o da detecção de mudanças (edições) em documentos. Por exemplo, um documento pode ser exportado de uma base de dados para edição por dois revisores. Quando do retorno a base de dados, é necessário descobrir que mudanças foram realizadas.

Existem vários algoritmos que descobrem o que mudou em documento (chamados algoritmos de *diff*), como por exemplo:

- Yuan Wang, David J. DeWitt, Jin-Yi Cai, *X-Diff: An Effective Change Detection Algorithm for XML Documents*, 19th International Conference on Data Engineering (ICDE'03) , p. 519, 2003.
  http://www.cs.wisc.edu/ yuanwang/papers/xdiff.pdf

Para grandes documentos que não cabem na memória, o ideal é encontrar as diferenças através de processamento direto do documento em uma base de dados relacional. Um artigo que apresenta uma solução deste tipo é:

- Erwin Leonardi and Sourav S. Bhowmick. *Xandy: A scalable change detection technique for ordered XML documents using relational databases* Data & Knowledge Engineering, Volume 59, Issue 2, November 2006, Pages 476-507
http://dx.doi.org/10.1016/j.datak.2005.06.006

7.3 Junção por similaridade

Bases de dados que contém dados coletados de fontes pouco estruturadas como a Web, ou dados fornecidos diretamente por usuários finais, podem apresentar a necessidade de execução de junções por similaridade.

A junção por similaridade é uma operação semelhante a junção clássica da álgebra relacional, com a diferença de que ela não exige a igualdade de valores de campos, mas aceita campos *similares* (por exemplo, com abreviaturas ou erros de digitação).

Este tipo de junção pode ser implementada diretamente sobre bases relacionais, como mostra o artigo:

- Luis Gravano, Panagiotis G. Ipeirotis, Nick Koudas, Divesh Srivastava: , *Text joins in an RDBMS for web data integration*. WWW 2003: 90-101
http://doi.acm.org/10.1145/775152.775166

7.4 Deduplicação de registros

Um problema que aparece em muitas aplicações de integração de dados é o da deduplicação de registros, isto é, de descobrir quando dois registros diferentes representam a mesma entidade do mundo real. Este problema aparece quando dados são digitados com diferentes convenções (ordem de palavra, abreviaturas, erros de digitação).

Uma forma interessante de resolver o problema é usando uma base de dados relacional, como mostra o artigo:

- N. Koudas, A. Marathe, and D. Srivastava. Flexible string matching against large databases in practice. Proceedings of the Thirtieth International Conference on Very Large Data Bases, pp. 1078-1086 2004.
http://www.vldb.org/conf/2004/IND3P3.PDF