

Utilizando Regras baseadas no Contexto para Reescrever Consultas

Aluno: Antônio Ezequiel de Mendonça

E-mail: aem@cin.ufpe.br

Orientadores

Ana Carolina Salgado

E-mail: acs@cin.ufpe.br

Damires Yluska de Souza Fernandes

E-mail: damires@ifpb.edu.br

Nível – Mestrado

Dissertação de Mestrado em andamento

Ingresso: março/2012

Conclusão prevista: fevereiro/2014

Etapas Concluídas: Créditos, Seminário de Acompanhamento, Definição do Problema, Especificação, Implementação de Protótipo.

Etapas Futuras: Escrita da Dissertação, Finalização da Implementação, Realização de Experimentos.

Previsão de Defesa: janeiro/2014.

Universidade Federal de Pernambuco – UFPE

Programa de Pós Graduação em Ciência da Computação

Área de Concentração: Ciência da Computação.

Linha de Pesquisa: Banco de dados, Contexto Computacional.

Abstract

When users access applications, they aim to obtain useful information. Sometimes, however, the user needs to reformulate submitted queries several times and go through many answers until a satisfactory set of answers is achieved. In this scenario, the user may be in different contexts, and these contexts may change frequently. A context surrounding his tasks (e.g., queries), for instance, may be built from his specific interests, location or expertise. In this work, we address the issue of rewriting queries considering the context acquired at query submission time. To this end, we propose a query rewriting approach, which makes use of context-based rules to produce new related expanded or relaxed queries. In this paper, we describe our approach and present some results we have obtained.

Keywords

Query Rewriting; Context; Rules.

1. Introdução e Motivação

O desenvolvimento de aplicações computacionais cada vez mais complexas e, ao mesmo tempo, adaptáveis e flexíveis, fez surgir a necessidade de mecanismos que pudessem ajudar no desenvolvimento destas. Um desses mecanismos diz respeito à utilização do *contexto* no qual uma interação entre um usuário e a aplicação acontece. Aplicações em geral fazem uso intensivo de dados e lidam com diversos tipos de usuários que interagem em momentos diferentes, por vezes, utilizando dispositivos ou ambientes também diferentes. Nessa perspectiva, o contexto pode ser usado como recurso computacional que possibilita às aplicações adaptarem-se às necessidades específicas de cada usuário ou ambiente em questão. Para Bolchini *et al.* (2009), o contexto é definido como um conjunto de variáveis que podem ser de interesse de um agente, podendo então influenciar as ações em uma dada tarefa.

Em aplicações que fazem uso intensivo de dados, às vezes uma consulta submetida pelo usuário pode conter uma descrição incompleta das informações de que ele necessita. Até mesmo quando a descrição é bem definida o mecanismo de consulta pode não ser capaz de retornar respostas que o usuário deseja. Nestes casos, argumentamos que o contexto pode ser utilizado para proporcionar a reescrita das consultas, de modo que respostas mais adequadas à necessidade do usuário possam ser retornadas. Com isto em mente, propomos uma abordagem de reescrita de consultas chamada CORE – *Context-based Rules for rEwriting*, que oferece expansão e/ou relaxamento da consulta de acordo com o contexto adquirido.

Para se trabalhar com o contexto, este precisa ser identificado e gerenciado. Particularmente em aplicações que usam SGBDs, algumas estratégias de gerenciamento de contexto vêm sendo utilizadas: (i) a nativa que consiste do desenvolvimento de um código dentro do próprio SGBD, usando operadores físicos e algoritmos específicos [Levandovski *et al.* 2010] e (ii) a externa que, por meio de um *plug-in* ou de um *middleware*, permite a tradução de uma consulta, sem modificar a implementação original do SGBD [Koutrika *et al.* 2010]. A abordagem CORE é baseada na segunda solução e é parte de uma arquitetura proposta por Maciel *et al.* (2013), que visa fornecer recursos de sensibilidade ao contexto em um SGBD. Este artigo tem como foco apresentar a abordagem CORE que vem sendo desenvolvida por meio de um componente de reescrita de consultas desta arquitetura de referência. Na CORE, utiliza-se consultas relacionais no padrão SQL92¹ e regras de produção [Newell *et al.* 1973] baseadas no contexto capturado. No processo de reescrita de consultas, o usuário acessa uma aplicação e submete consultas que serão alteradas de acordo com o contexto identificado. Dessa forma, o usuário irá receber respostas mais relevantes associadas a seu contexto.

Particularmente, definimos o nosso problema da seguinte forma: *Dada uma consulta do usuário Q, expressa em SQL, nosso objetivo é gerar uma consulta reescrita Q', a qual é semanticamente relacionada à consulta original Q. A relação semântica entre as duas consultas é determinada pelo contexto adquirido, que é especificado por meio de regras.*

O processo de criação de regras é desempenhado por um *especialista no domínio* (ED) da aplicação que expressa seu conhecimento por meio de regras de produção. Os elementos que formam a condição da regra são denominados *elementos contextuais*

¹ <http://www.sqlteam.com/>.

(EC) e representam os fatos a serem usados pelo motor de inferência. Os ECs referem-se a qualquer dado, informação ou conhecimento que permite caracterizar uma entidade em um domínio [Vieira *et al.* 2011]. Por exemplo, *localização* e *preferências* são elementos contextuais associados a uma entidade *Usuário*.

Há algumas maneiras em que a nova consulta Q' pode ser semanticamente relacionada com a consulta original Q. Neste trabalho, duas técnicas são utilizadas: (i) *expansão de consultas*, onde é realizada a adição de novos termos adquiridos a partir do processamento das regras baseadas em contexto; (ii) *relaxamento de consultas*, onde se faz uma redução do escopo de consultas através da remoção de alguns de seus termos, a fim de otimizar a consulta e reduzir o número de respostas incorretas ou redundantes. Para viabilizar o processo de reescrita, um conjunto de operadores (e.g., *trunk*, *order_value*, *etc*) vem sendo especificado. Estes operadores são utilizados na definição das regras e visam facilitar o processo de confecção das mesmas. Com base nas definições da abordagem, foi implementado uma versão inicial da *CORE*, juntamente com uma aplicação front-end. Estes tópicos serão abordados ao longo deste artigo.

Este artigo está organizado como segue: a Seção 2 introduz alguns conceitos básicos; a Seção 3 caracteriza as contribuições do trabalho; a Seção 4 apresenta a avaliação dos resultados obtidos até o momento. A Seção 5 descreve alguns trabalhos relacionados. A Seção 6 tece algumas considerações e indica o desenvolvimento necessário para a conclusão.

2. Fundamentação Teórica

Para Godfrey e Gryz (1996), *reescrita de consultas* é uma técnica que utiliza algum tipo de conhecimento semântico, como caches semânticas, visões materializadas ou o conhecimento do domínio, a fim de realizar uma tradução da consulta. Ao reescrever uma consulta Q em uma consulta Q', técnicas de enriquecimento ou relaxamento podem ser empregadas. Assim, a *expansão de consultas* é definida por Andreou *et al.* (2005) como um processo de inclusão de novos termos em uma consulta submetida pelo usuário, com a finalidade de melhorar as respostas obtidas. Por outro lado, o *relaxamento de consultas* refere-se ao processo ao qual a consulta é simplificada por restrições de enfraquecimento das expressões que são responsáveis por falhas [Stuckenschmidt *et al.* 2005]. O objetivo é generalizar uma consulta que contém falhas, criando uma consulta mais eficiente, eliminando ou relaxando algumas restrições da consulta original.

Segundo Dey e Abowd (2000), contexto se refere a qualquer informação que caracteriza a situação de uma entidade, onde uma entidade é uma pessoa, lugar ou objeto considerado relevante para a interação entre um usuário e uma aplicação. O contexto pode ser utilizado para ampliar o conhecimento que se tem sobre uma determinada situação, desempenhando um papel importante em qualquer domínio que envolva requisitos como compreensão, raciocínio, resolução de problemas ou aprendizado [Vieira *et al.* 2011]. Neste sentido, este trabalho busca usar o contexto identificado no momento da submissão de uma consulta, de forma associada às técnicas de expansão e relaxamento para reescrever consultas. Para viabilizar a inferência do contexto, utiliza-se regras de produção, aqui denominadas regras baseadas em contexto.

3. Caracterização da Contribuição

A abordagem *CORE* é parte integrante da arquitetura *Texere* [Maciel *et al.* 2013]. O objetivo da *Texere* é proporcionar características de sensibilidade contextual a um

SGBD tradicional, por meio de alguns componentes, como, por exemplo, editor de regras, definição de modelo de persistência do contexto e o componente de reescrita de consultas – foco deste trabalho.

Nesse panorama, a abordagem CORE usa as diretivas de reescrita retornadas pelo motor de inferência, a partir do processamento das regras de produção para realizar a reescrita da consulta Q. A diretiva de reescrita é um comando destinado a produzir um pedaço de código, o qual é traduzido para o padrão SQL 92. As regras criadas pelo Especialista de Domínio (ED), serão utilizadas para realizar a inferência do contexto e produzir novos fatos. Estes podem ser uma caracterização do contexto ou uma diretiva para reescrita da consulta.

O processo de reescrita faz uso de duas estratégias: expansão ou relaxamento. A decisão de qual estratégia utilizar na reescrita depende do contexto adquirido. Por exemplo, se o usuário, no momento de submissão da consulta, estiver utilizando um *smartphone*, o resultado de uma regra disparada para esse contexto é uma diretiva do tipo *revisão trunc 200*. Isso implica em um comando de expansão, pois a consulta original Q não previa essa restrição, em que o campo *revisão* deve apenas retornar os 200 primeiros caracteres.

O processo de reescrita se inicia quando o usuário conectado a uma aplicação envia uma consulta denominada Q, e esta é capturada pelo CORE. Ao final do processamento das regras, um conjunto de diretivas é devolvida ao CORE. Cada diretiva é traduzida para um comando em SQL 92, que é integrado à consulta original, gerando uma nova consulta Q'. A diretiva pode ser de dois tipos: relaxamento e expansão. Nesta, novos elementos serão incorporados à consulta original; naquela, a diretiva leva à subtração de algum elemento ou, até mesmo, o aumento da abrangência de alguma restrição. Finalmente, a consulta reescrita Q' é executada no banco de dados da aplicação e os resultados obtidos são devolvidos ao usuário. A Figura 1 ilustra uma visão geral da CORE. Os principais componentes são apresentados a seguir:

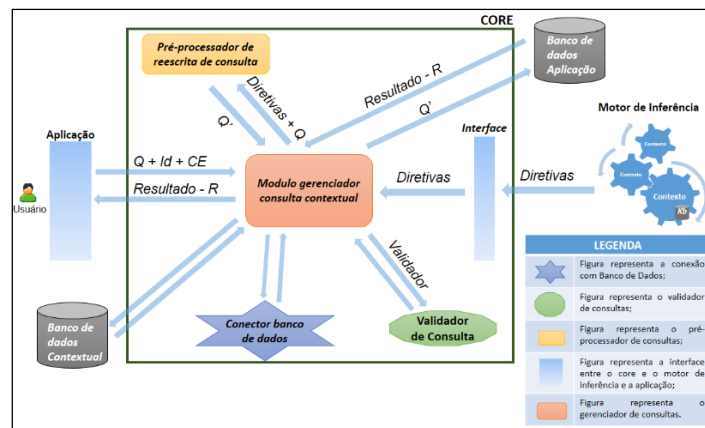


Figure 1. Principais componentes do CORE.

- (I) **Conector do Banco de Dados** – gerência as conexões com os bancos de dados necessários ao processo de reescrita, como banco de dados *metadata* e o da aplicação;
- (II) **Validador de Consulta** – valida sintaticamente as consultas enviadas pela aplicação e a consulta final Q', resultante do processo de reescrita;
- (III) **Pré-processador de reescrita de consulta** – recebe a consulta Q e as diretivas, posteriormente traduz cada uma para um comando SQL padrão. Em seguida, verifica se a diretiva é uma operação de expansão ou de relaxamento. Então é executando a

reescrita da consulta Q, com a operação apropriada a diretiva traduzida; (IV) **Interface do motor de inferência** – Recebe as diretivas de reescrita de consulta enviadas pelo motor de inferência; (V) **Gerenciador de consulta contextual** – instancia os ECs, realiza a integração dos módulos do *CORE* e coordena o processo de reescrita.

Ao criar regras, o ED usa uma linguagem de alto nível. Esta linguagem é semelhante à linguagem natural e tem seus próprios operadores. Por exemplo, o operador **Trunk**, que não existe no padrão SQL, foi criado para possibilitar restringir a quantidade de caracteres retornados em uma coluna específica (ver operadores na Tabela 1). Por exemplo, em uma diretiva que contenha o operador **Trunk**, este será transformado em uma função *substring* (*coluna, caraterInicial, quantidadeCaracteresRetornados*).

Tabela 1 – Alguns Operadores usados no CORE.

Escrita	Ação	Exemplo
Trunk	Limita o retorno de uma quantidade x de caracteres de um campo ao submeter a consulta.	IF dispositivo igual celular Then revisão trunc 200
Order Value	Ordena a consulta por um valor específico de uma coluna.	IF idioma igual frances Then lingua order_value 'francesa'

4. Avaliação dos Resultados e Estado Atual do Trabalho

Para validar a proposta desse trabalho, foi implementado um protótipo que contempla todos os componentes da *CORE* descritos na Seção 3 foi implementado. Como estudo de caso inicial, desenvolvemos uma aplicação *front-end* de consulta a livros denominada TexereLibrary (cuja interface é mostrada na Figura 2), que usa o *CORE* para obter a reescrita das consultas com base no contexto identificado. O usuário pode realizar consultas, escrevendo comandos SQL padrão e habilitando a opção de consultas *com* ou *sem* contexto.

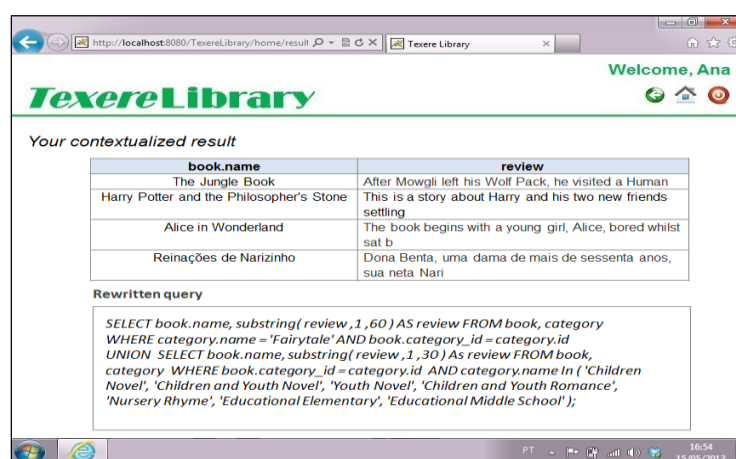


Figura 2 – Página de Resultados do TexereLibrary.

Como ilustração, suponha que o usuário Ana é uma menina de nove anos que mora no Brasil e está usando o TexereLibrary. Ana se registra na aplicação e recebe um id (user_id = 10) e uma sessão (session_id = 15). Ana está usando um *smartphone* (device = smartphone) e o mês atual é julho. Esses ECs são capturados pelo *CORE* e

armazenados. Ana então submete a seguinte consulta (SQL) Q: *Select name, review From book*. O CORE, após a submissão da consulta e a execução das regras definidas na Tabela 2 pelo motor de inferência, recebe as diretivas de reescrita.

Tabela 2 – Regras disparadas.

IF <i>mobile_device</i> Then <i>truncate_information</i> (revisão);
IF <i>user_age</i> < 12 Then <i>show (book_category)</i> in (Children Novel, Children Youth);
IF estação in (verão) Then <i>school_vacation</i> is true;
IF <i>school_vacation</i> is true and <i>user_age</i> <= 12 Then <i>category_order_value</i> = 'Fairytale';

Após o processamento das regras e considerando as diretivas de reescrita apresentadas em cada regra na cláusula *Then*, a consulta Q' é reescrita como segue:

*Select book.name, substring(review, 1, 200) As review From book, category Where category.name = 'Fairytale' AND book.category_id = category.id Union
Select book.name, substring(review, 1, 200) As review From book, category Where book.category_id = category.id AND category.name In ('Children Novel', 'Children Youth');*

Esta consulta é executada e seus resultados são retornados a Ana.

Experimentos estão sendo realizados com dois objetivos básicos: (i) obter um feedback dos usuários quanto à utilização do contexto para reescrever consultas (em termos de relevância das respostas), e (ii) verificar o desempenho da aplicação acoplada ao serviço de reescrita, em termos de tempo de resposta. O primeiro está sendo montado por meio de um formulário que será preenchido pelos usuários. O segundo já foi realizado e, diante de seus resultados, foi implementado um mecanismo de cache que melhorou sensivelmente o tempo de resposta das consultas.

5. Trabalhos Relacionados

Quanto à utilização de contexto em consultas, Amo e Pereira (2012) apresentam uma extensão da linguagem SQL denominada CPrefSQL. Para realizar a personalização, o usuário precisa informar previamente suas preferências (regras) que são do tipo "IF <contexto> THEN <preferencia>", onde <preferencia> indica o desejo do usuário em uma situação que satisfaz o <contexto>. Levandoski *et al.* (2010) trabalham com um sistema de banco de dados sensível ao contexto, implementado dentro do SGBD PostgreSQL, chamado CareDB. Este tem por objetivo redefinir a resposta de uma consulta tradicional, através da integração de vários tipos de contextos. Hachani *et al.* (2006) apresenta uma abordagem para estender os critérios de restrição de uma consulta, a fim de fornecer respostas aproximadas. A abordagem utiliza o relaxamento das restrições, bem como o contexto do usuário no processo de reescrita.

Comparando estes trabalhos, Amo e Pereira (2012) concentram-se em uma extensão de SQL. O usuário precisa registrar suas preferências (regras) em um momento imediatamente anterior ao de realizar consultas, tornando o processo de customização mais caro e estático. Em Levandoski *et al.* (2010), a abordagem exige um grande esforço para desenvolver um código nativo no SGBD, a fim de proporcionar a criação de consultas contextuais. Outra desvantagem é a impossibilidade de usar este mesmo código em outro SGBD. No trabalho de Hachani *et al.* (2006), a reescrita é feita apenas

com o relaxamento das restrições, limitando assim as possibilidades de customização da consulta. Em nossa abordagem, a geração de regras é feita pelo ED, que identifica e registra as regras sem a necessidade de desativar o serviço do *CORE*. Trabalhamos com o padrão SQL, por isso não há necessidade de alterar o algoritmo interno do SGBD relacional subjacente. Assim, aumentamos a portabilidade do sistema, reduzindo os custos com a implementação nativa e permitindo que qualquer SGBD relacional compatível com o padrão SQL 92 use o *CORE*.

6. Considerações e Desenvolvimento Necessário para Conclusão

Este trabalho apresentou a abordagem *CORE*, a qual foi especificada e uma primeira versão foi implementada. Atualmente, estamos estendendo os operadores definidos que suportam a identificação de diretivas e o processamento das regras. Além disso, após o processo de reescrita serão incorporadas técnicas de *tuning* sobre Q', a fim de melhorar o desempenho de execução da consulta reescrita.

Referências

- Amo, S. and Pereira, F. (2010) "Evaluation of conditional preference queries". Journal of Information and Data Management (JIDM). v. 1, p. 521–536.
- Andreou A. (2005) "Ontologies and query expansion". M.S. thesis. School of Informatics, University of Edinburgh, Edinburgh, UK.
- Bolchini, C. Curino, A. Quintarelli, E. Schreiber, A. e Tanca, L. (2009) "Context information for knowledge reshaping". Int. J. Web Eng. and Tech., Milano, Italia.
- Dey, A. K., Abowd, G. D. (2000) "Towards a Better Understanding of Context and Context-Awareness", In: Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands.
- Godfrey, P. Gryz, J. (1996) "A framework for intensional query optimization". In DDLP, p. 57–68.
- Hachani, N. Ali, M. Hassine, B. and Ounelli, H. (2009) "Cooperative Answering of Fuzzy Queries". Journal of Computer Science and Technology, p. 675-686
- Levandovski J. J., Mokbel M. F., and Khalefa M. E. (2010) "CareDB: A Context and Preference -Aware Location-Based Database System". In Proceedings of the VLDB Endowment, p. 1529-1532.
- Koutrika, G. (2010) "Query Personalization based on User Preferences". V. 35, Abril, New York, USA.
- Newell A. (1973) "In Visual Information Processing. Academic Press". Chase E. (editor), New York, USA, p.283-308.
- Maciel, Paulo. (2013) "Texere, a Context-aware System for Improving Database Queries". Technical Report, Federal University of Pernambuco, Brazil.
- Stuckenschmidt H., Giunchiglia F., and van Harmelen F. (2005) "Query processing in ontology-based peer-to-peer systems". In V. Tamma, S. Craneeld, T. Finin, and S. Willmott, editors, Ontologies for Agents: Theory and Experiences. Birkhuser.
- Vieira, V., Tedesco, P., and Salgado A. C. (2011) "Designing Context-Sensitive Systems: Na integrated Approach". Expert Systems with Applications 38. p. 1119-1138.