

CPrefSQL-Tool: Uma Ferramenta Web para Consultas com Suporte a Contextos e Preferências do Usuário

Vinicius V. S. Dias¹, Sandra de Amo¹

¹ Faculdade de Computação

Universidade Federal de Uberlândia (UFU) – Uberlândia, MG – Brazil

viniciusvdias@comp.ufu.br, deamo@ufu.br

Abstract. *In an ubiquitous computing scenario users access to databases will not occur at a unique context, and consequently their expectation when querying data will vary according to a multitude of parameters including location, time and surrounding influences. In this article we present CPrefSQL-Tool, a web tool allowing to execute queries and filter the results according to the user preferences and contexts. Some of the functionalities of the CPrefSQL-Tool are: (1) queries may be specified either by using standard SQL commands or by using CPrefSQL, an extension of SQL with specific operators for computing the top – k and the most preferred tuples according to a set of contextual preference rules; (2) preference rules are easily specified by means of a simple and intuitive interface; (3) queries are executed by using the CPrefSQL query processor; (4) the filtered results of a query execution are traceable and explained on user demand.*

Resumo. *Num cenário típico de computação ubíqua, os acessos dos usuários a um banco de dados não ocorrem em um único contexto e consequentemente suas expectativas ao consultar os dados podem variar de acordo com uma multitude de parâmetros incluindo sua localização, o momento presente e as influências do ambiente. Este artigo apresenta CPrefSQL-Tool, uma ferramenta Web que permite a execução de consultas e a filtragem dos resultados levando em consideração as preferências e o contexto do usuário. Algumas das funcionalidades de CPrefSQL-Tool são: (1) as consultas podem ser especificadas por meio de comandos SQL ou por meio de comandos da linguagem CPrefSQL, uma extensão de SQL com operadores específicos para computar as k melhores tuplas de acordo com um conjunto de regras de preferências contextuais fornecido; (2) as regras de preferência são facilmente especificadas através de uma interface simples e intuitiva; (3) as consultas são executadas por meio do processador de consultas da linguagem CPrefSQL; (4) os resultados filtrados após a execução de uma consulta podem ser rastreados e explicados caso o usuário assim o desejar.*

1. Introdução

A incorporação de recursos permitindo levar em conta as preferências e o contexto dos usuários em mecanismos de busca ou em linguagens tradicionais de consultas em banco de dados tem despertado cada vez mais interesse na comunidade científica, abrindo possibilidades de pesquisa em áreas como banco de dados, inteligência artificial e recuperação de informação. Isso pode ser observado em aplicações sensíveis a contexto, como definições de perfis de usuário na Web [Suryanarayana and Hjelm 2002]

e computação móvel [Chen and Kotz 2000], que garantem uma experiência personalizada para o usuário. No que diz respeito à área de banco de dados, a pesquisa tem se concentrado na incorporação de operadores de preferências às linguagens de consultas tradicionais para manipulação de dados, como o SQL [de Amo and Pereira 2010, K.Stefanidis and E.Pitoura 2008]. Recentemente, foi proposta a linguagem CPrefSQL [de Amo and Pereira 2010], que estende o SQL padrão com dois operadores que permitem filtrar todas as melhores tuplas ou as k melhores tuplas de acordo com uma hierarquia de *preferências contextuais* informada pelo usuário através de comandos específicos da linguagem. O modelo de preferências adotado em CPrefSQL adapta o formalismo lógico de [Wilson 2009].

Neste artigo é apresentada *CPrefSQL-Tool*, uma ferramenta *Web* desenvolvida com o propósito de possibilitar que usuários, entendidos ou não da linguagem CPrefSQL, possam gerenciar e aplicar suas preferências num banco de dados de sua escolha. O foco principal deste trabalho está em disponibilizar as funcionalidades da linguagem CPrefSQL através de uma interface simples e intuitiva. A ferramenta também permite ao usuário entender como o processador de CPrefSQL executa a filtragem das tuplas de acordo com as regras de preferências contextuais fornecidas pelo usuário. Mais ainda, a ferramenta permite comparar os resultados retornados por consultas tradicionais (SQL padrão) e consultas CPrefSQL. Está disponível em <http://www.lsi.ufu.br/cprefsq>.

2. A Linguagem CPrefSQL

A linguagem CPrefSQL possui três funcionalidades básicas, que envolvem operações de (1) criação de preferências contextuais, (2) consulta de tuplas mais preferidas e (3) *ranking* de tuplas preferidas. A seguir, será apresentada cada uma das funcionalidades da linguagem juntamente com seu comando correspondente. Considere, para os exemplos ao longo deste texto, um banco de dados composto pela relação *hospedagem*(*hotel*, *cidade*, *avaliacao*, *preco*, *distancia*, *finalidade*). Cada tupla desta relação descreve uma opção de hospedagem, com informações de nome de hotel, localização, preço da diária, distância e finalidade da mesma respectivamente.

2.1. Criação de Preferências Contextuais

No modelo considerado, definimos uma preferência de usuário como um conjunto de regras do tipo "*if <contexto> then <preferencia>*", onde *<preferencia>* da regra indica a vontade do usuário em uma situação que satisfaz o *<contexto>*. Dito isso, a criação de uma preferência consiste em executar um comando CPrefSQL que recebe um conjunto de regras deste tipo e as armazena no catálogo do SGBDR (Sistema Gerenciador de Banco de Dados Relacional) através de um identificador. A Figura 1(a) ilustra o comando de criação de preferências contextuais. O comando da Figura 1(b) ilustra a criação de uma preferência sobre a relação *hospedagem*. Basicamente ele cria uma preferência com o nome "mypref-hospedagem" que envolve atributos da relação *hospedagem*. Por exemplo, a primeira regra pode ser traduzida como: "*Se a distância for menor do que 500km, então prefiro hospedagens com diária inferior a R\$ 250,00 desde que as finalidades respectivas das mesmas sejam idênticas.*"

<pre>CREATE PREFERENCES <nome-pref> FROM <tabelas> AS <regras-pref> (a)</pre>	<pre>CREATE PREFERENCES mypref-hospedagem FROM hospedagem AS IF distancia<500 THEN preco< 250 > preco≥ 250 [1,2,3] AND IF preco>400 THEN avaliacao=5 > avaliacao=4 [1,2] AND finalidade = 'férias' > finalidade = 'trabalho' [1,2,5] (b)</pre>
---	--

Figura 1. Sintaxe do Comando de Criação de Preferências

2.2. Tuplas mais Preferidas e *Ranking* de Tuplas

Após a criação de uma preferência, a mesma pode ser usada em alguma consulta de CPrefSQL. Para entender como as preferências são usadas pelo processador CPrefSQL, é necessário discutir brevemente a semântica de um conjunto de regras de preferências. Um tal conjunto induz uma ordem natural sobre as tuplas de um banco de dados. Esta ordem constitui sua *semântica*. Cada regra de preferência isolada induz uma relação de dominância entre as tuplas de forma natural. Por exemplo, considere as tuplas $t_1 = (\text{Excelsior}, \text{Madri}, 3, 190, 400, \text{férias})$ e $t_2 = (\text{St Germain}, \text{Paris}, 5, 300, 400, \text{férias})$. A primeira regra “**IF** distancia < 500 **THEN** (preco < 250) > (preco ≥ 250) [1,2,3]” permite comparar t_1 e t_2 e concluir que t_1 é preferida a t_2 (ou t_1 *domina* t_2). Repare que duas tuplas podem ser comparadas se elas indiferem nos atributos que não aparecem na regra (finalidade - atributo 6). Os números entre colchetes [1,2,3] referem-se respectivamente aos atributos *hotel*, *cidade* e *avaliação* que não precisam ser idênticos nas duas tuplas a fim de que estas possam ser comparadas. A relação de ordem final (ou relação de *dominância* entre as tuplas) correspondente a um *conjunto* de regras consiste em considerar a união das ordens induzidas por cada regra individualmente e depois considerar o fecho transitivo da relação obtida.

A relação de dominância pode ser visualizada através de um *Grafo de Dominância*. Dado este grafo, diz-se que uma tupla t_1 domina uma tupla t_2 se existir um caminho direcionado que parta de t_1 e chegue em t_2 . A ausência de arcos entre tuplas indica tuplas *indiferentes* e arcos tracejados representam uma dominância por transitividade. Assim, no exemplo da Figura 2, temos que: t_1 é *dominante* em relação à t_2 e t_3 , t_2 é *dominante* em relação à t_3 mas *dominada* em relação à t_1 .

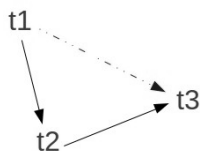


Figura 2. Grafo de Dominância

Deste modo, a ideia dos algoritmos que implementam a linguagem CPrefSQL é, justamente, percorrer todas as tuplas em questão e calcular a dominância entre as mesmas, que é definida para cada par de tuplas consideradas. No final resta a exibição das tuplas escolhidas para a resposta de acordo com a operação solicitada (mais preferidas ou *ranking*).

Dois tipos de consultas podem ser feitas: (1) perguntar pelas tuplas que melhor se adequam a uma dada preferência de usuário; (2) pedir as *top-k* tuplas, isto é as primeiras k tuplas preferidas de acordo com o *ranking* estabelecido pelas regras de preferências criadas. No primeiro tipo de consulta, as dominâncias entre as tuplas são calculadas e as consideradas *dominadas* (possuem pelo menos um arco de entrada de acordo com o *Grafo de Dominância* correspondente) são excluídas da resposta. Aqui, a ordem de apresentação das tuplas do resultado não tem significado, a única certeza sobre elas é que são preferidas (*dominantes*). No segundo tipo de consulta, as tuplas da resposta são ordenadas em ordem decrescente de preferência e as k primeiras são retornadas. Essa ordenação é feita do seguinte modo: (1) a cada tupla é associado um nível, (2) de acordo com o *Grafo de Dominância*, o processo é iniciado atribuindo-se o nível 0 às tuplas que não são dominadas por nenhuma outra tupla (as que não possuem nenhum arco de entrada), (3) para as

de seu nome (ou identificador). Tais regras já devem ter sido criadas através do comando CPrefSQL (ver Figura 1(a)) ou usando a interface da ferramenta própria para isso.

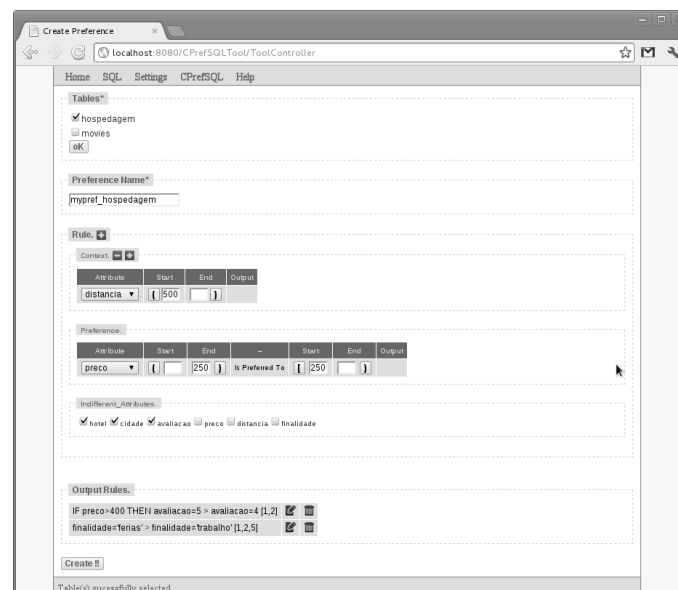


Figura 4. Interface de Inserção das Regras de Preferências

Execução de Consultas: De acordo com a configuração feita em (E), o usuário (1) insere a consulta SQL e o nome do arquivo de preferências ou (2) insere o comando CPrefSQL correspondente à sua consulta (caso conheça seja familiarizado com a sintaxe desta linguagem). A Figura 5 ilustra a execução da consulta da Figura 3(b), em que as seguintes configurações foram declaradas: (C) apresentar apenas o resultado da consulta CPrefSQL, (D) rastreamento de resultado desativado e (E) definido como SQL padrão mais a escolha do nome de um conjunto de regras de preferência criado.

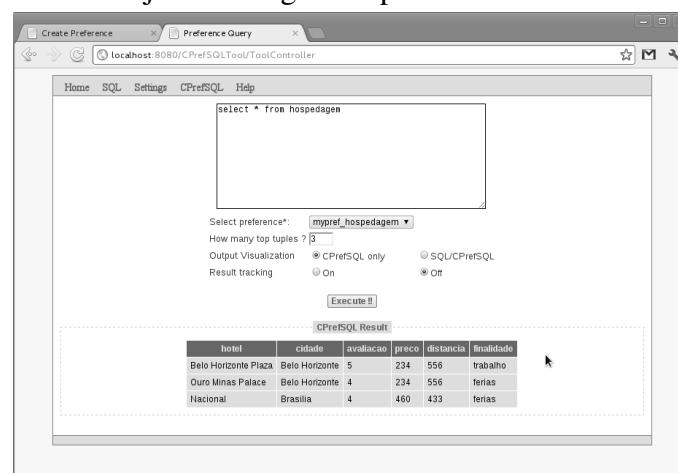


Figura 5. Execução de uma consulta

4. Implementação e Ambiente

A ferramenta foi implementada como uma aplicação *Java-Web* por motivos de portabilidade. Assim, a aplicação oferece a interface para manipulação de preferência sobre bancos de dados armazenados em um SGBDR, local ou remoto. Além disso, devido às características originalmente propostas para a linguagem CPrefSQL e suas extensões, a ferramenta usa o PostgreSQL como SGBDR. A arquitetura simplificada da ferramenta pode ser observada na Figura 6. Assim: (1) um usuário interage com o servidor *Web* da

ferramenta através de uma interface HTTP Requisição/Resposta, (2) o servidor se comunica com o banco de dados através de consultas (SQL ou CPrefSQL) e obtém respostas do mesmo, (3) no caso de a requisição ao SGBDR envolver preferências CPrefSQL, é necessário que o processador de consultas se comunique com o diretório de preferências.

Outro ponto importante é que o gerenciador de banco de dados a ser usado deve suportar a linguagem CPrefSQL, demandando uma pré-configuração desse ambiente.

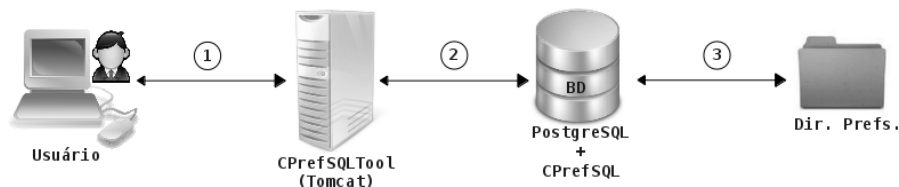


Figura 6. Arquitetura da ferramenta CPrefSQLTool

5. Conclusões e Extensões Futuras

Neste artigo foi apresentada *CPrefSQL-Tool*, uma ferramenta *Web* que permite a execução de consultas com suporte a preferências e contextos do usuário de forma simples e amigável. Com estas características, *CPrefSQL-Tool* é capaz, através do exercício prático, de estabelecer fortes vantagens em se utilizar linguagens com suporte a preferências como a CPrefSQL. Neste aspecto, o objetivo é que um usuário possa criar preferências sobre um banco de dados qualquer e, através de consultas envolvendo as mesmas, perceber as diferenças na resposta obtida quando comparada a uma consulta SQL padrão. Diferenças que, em geral, fazem com que a partir de uma linguagem com suporte a preferências os piores casos (consultas sem resposta ou com respostas inviavelmente volumosas) sejam evitados. Um trabalho futuro é a incorporação de um módulo de mineração de preferências [de Amo et al. 2012, da Silva and de Amo. 2011] na ferramenta, possibilitando uma funcionalidade de extração automática de preferências do usuário a partir de suas escolhas passadas.

Referências

- Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. *Technical report tr2000, Departament of Computer Science, Dartmouth College, Hanover, New Hampshire, USA*, page 381.
- da Silva, N. F. F. and de Amo., S. (2011). Cprefminer: A bayesian miner of conditional preferences. *Journal of Information and Data Management (JIDM)*, 2 (1):35–42.
- de Amo, S., Diallo, M. S., Diop, C. T., Giacometti, A., Li, H. D., and Soulet, A. (2012). Mining contextual preference rules for building user profiles. In *14th International Conference on Data Warehousing and Knowledge Discovery - DaWaK 2012, Vienna (Austria)*.
- de Amo, S. and Pereira, F. (2010). Evaluation of conditional preference queries. *Journal of Information and Data Management (JIDM)*, 1(3):521–536.
- K.Stefanidis and E.Pitoura (2008). Fast contextual preference scoring of database tuples. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 344–355.
- Suryanarayana, L. and Hjelm, J. (2002). Profiles for the situated web. *11th International World Wide Web Conference (WWW 2002)*, pages 200–209.
- Wilson, N. (2009). Extending cp-nets with stronger conditional preference statements. *AAAI*.