

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/255634082>

# Uma Ferramenta de Apoio à Normalização de Tabelas Relacionais Baseada na Análise de Dados

## Article

CITATIONS

0

READS

2,569

## 2 authors:



[Michel Leite de Ávila](#)

Federal University of Santa Catarina

1 PUBLICATION 0 CITATIONS

SEE PROFILE



[Ronaldo Mello](#)

Federal University of Santa Catarina

94 PUBLICATIONS 733 CITATIONS

SEE PROFILE

# Uma Ferramenta de Apoio à Normalização de Tabelas Relacionais Baseada na Análise de Dados

Michel Leite de Ávila<sup>1</sup>, Ronaldo dos Santos Mello<sup>1</sup>

<sup>1</sup>Depto. de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)  
Caixa Postal 476 - 88.049-900 – Florianópolis – SC - Brasil

{mlavila,ronaldo}@inf.ufsc.br

**Abstract.** *This paper presents a relational table normalization tool for helping developers to reduce their efforts and optimize the quality in bottom-up relational database projects. By analyzing table data, the tool discovers all functional dependencies that must be removed. The designer's only effort is to evaluate the dependencies between attributes when the tool's analysis is not completely sure, and decide whether it is a functional dependency or not. The tool keeps decomposing the main table in simpler and functional dependency safe tables until the third normal form is reached. It also offers support for the importing relational data from plain text and XML documents, as well as required pre-processing and primary key definition during the application of the normal forms.*

**Resumo.** *Este artigo apresenta uma ferramenta de apoio ao processo de normalização de tabelas, visando contribuir principalmente com projetos bottom-up de bancos de dados relacionais, reduzindo os esforços do projetista e melhorando a qualidade do resultado. Através da análise de dados, a ferramenta descobre dependências funcionais existentes que devem ser removidas. O único esforço do projetista é avaliar as relações entre atributos quando a ferramenta não consegue determinar com segurança se uma dependência funcional está ocorrendo. A ferramenta decompõe sistematicamente a tabela principal em tabelas mais simples e livres de dependências funcionais até alcançar a terceira forma normal. Ela também oferece suporte aos passos de importação de dados relacionais de arquivos texto ou XML, assim como atividades de pré-processamento e eleição de chave primária, eventualmente necessárias durante a aplicação das formas normais.*

## 1. Introdução

A normalização de tabelas foi proposta por E. Codd em 1970, juntamente com o próprio modelo de dados relacional, como uma técnica para eliminar redundâncias de informações e evitar anomalias causadas pela inserção, atualização e remoção de tuplas [Codd 1970]. A aplicação desta técnica não é trivial, exigindo conhecimento sobre formas normais e dependências funcionais, além da teoria de bancos de dados relacionais. O sucesso da normalização depende, também, da intimidade do projetista com o domínio dos dados. Essa combinação de requisitos faz com que esta etapa seja complexa, pelo tempo que consome e pela carga de conhecimento teórico e do domínio

dos dados requeridos. Como consequência, a normalização é difícil de ser totalmente automatizada, deixando geralmente a qualidade da execução do projeto fortemente dependente da experiência do projetista, o que nem sempre gera bons resultados.

Este artigo apresenta uma ferramenta para apoiar a etapa de normalização de tabelas relacionais em projetos *bottom-up*, com foco no auxílio à descoberta de dependências funcionais através da análise dos próprios dados das tabelas, atenuando assim a carga de trabalho por parte do projetista. A ferramenta transporta os dados de um arquivo de texto ou XML para uma tabela inicial não-normalizada, analisa os dados e monta uma base de informações para apoiar a aplicação de cada passo da normalização, solicitando a intervenção do usuário apenas quando as informações extraídas não são capazes de decidir por uma decomposição com segurança. O resultado do processo é um conjunto de tabelas normalizadas até a terceira forma normal.

A ferramenta pode dar suporte tanto para projetos *bottom-up* de novos bancos de dados relacionais [Heuser 2004] quanto para o re-projeto de tabelas relacionais já existentes. Comparada com trabalhos relacionados, a ferramenta diferencia-se por auxiliar o usuário na tomada de decisões em cada etapa da normalização com base em informações coletadas a partir dos dados existentes nas tabelas.

O restante do artigo está organizado da seguinte forma. A seção 2 aborda os trabalhos relacionados e suas contribuições para este trabalho. A seção 3 apresenta a ferramenta proposta, detalhando o funcionamento de cada módulo. Por fim, a seção 4 apresenta as conclusões deste trabalho.

## 2. Trabalhos Relacionados

Os trabalhos que mais se aproximam da ferramenta aqui apresentada estão relacionados com a automatização da engenharia reversa de bancos de dados [Codd 1970]. Os processos propostos utilizam documentos físicos, esquemas, análise de dependências funcionais, código fonte da aplicação e código SQL, juntamente com dados. No entanto, não há abordagem que se proponha a normalizar um conjunto de dados com base apenas na análise dos dados de entrada, dispensando o conhecimento sobre o seu domínio.

Em [Chen 2007] são apresentadas algumas abordagens complementares ao processo tradicional de normalização, introduzindo o conceito de *independência funcional* e melhorando a qualidade do resultado da normalização. Uma independência funcional ocorre quando a combinação de todos os valores entre dois atributos é válida, ou seja, não se fere nenhuma restrição semântica. Desta forma, o trabalho define a Forma Normal da Independência Funcional (FINF). Uma tabela está na FINF se, e somente se, estiver na Forma Normal de Boyce-Codd (FNBC) e, para todos os pares de atributos “X” e “Y” for verificado que  $X \rightarrow Y$  ou  $Y \rightarrow X$  ou  $X \succ Y$  (esta última a notação formal para independência funcional). Isto implica a ausência de redundâncias causadas por dependências funcionais de subdomínio (quando ocorre apenas para alguns valores do atributo) e outras formas de relacionamento entre atributos.

Em [Soutou 1998] é apresentada uma técnica de extração de cardinalidades via comandos SQL, gerados dinamicamente sobre um dicionário de dados, para aperfeiçoar a engenharia reversa de bancos de dados. Essa abordagem pode ser aplicada para extrair a cardinalidade entre atributos, auxiliando na descoberta de dependências funcionais. Os

resultados podem ser aplicados em esquemas ER (Entity-Relationship), MERISE, ECR(+) (Entity-Relationship Complete), OMT (Object-Modeling Technique) e ODMG (Object Data Management Group), ou ainda na integração de ferramentas comerciais que oferecem engenharia reversa de bancos de dados.

Em [Yeh 2005] é proposto um processo de extração de um diagrama ER de um banco de dados legado, com pouca informação sobre os atributos e nenhuma informação sobre as chaves, analisando os próprios dados armazenados e telas de formulário do sistema que alimenta o banco. A semântica dos dados é descoberta através do preenchimento dos formulários do sistema e análise do posterior armazenamento das informações de entrada no banco de dados.

Grande parte das pesquisas nesta área de extração de esquemas conceituais de dados baseia-se na utilização do esquema relacional como entrada principal. A semântica dos atributos de um banco de dados é vital para a compreensão do funcionamento do mesmo, e geralmente é pobre ou até mesmo inexistente, justificando abordagens como esta, que utilizam a análise dos formulários do sistema para ajudar na reconstrução de esquemas ER.

Este e os demais trabalhos apresentados contribuíram para o amadurecimento geral da ferramenta. No entanto, este trabalho se diferencia destes ao propor uma ferramenta de apoio à normalização que utiliza apenas os dados como entrada, dos quais se extrai as informações necessárias para efetuar grande parte da normalização praticamente de forma automática. Outro aspecto é a disponibilização de uma interface com o usuário dotada de recursos para apresentar a teoria envolvida em todo o processo de normalização, distribuindo-a em módulos e passos, conferindo um aspecto acadêmico à ferramenta.

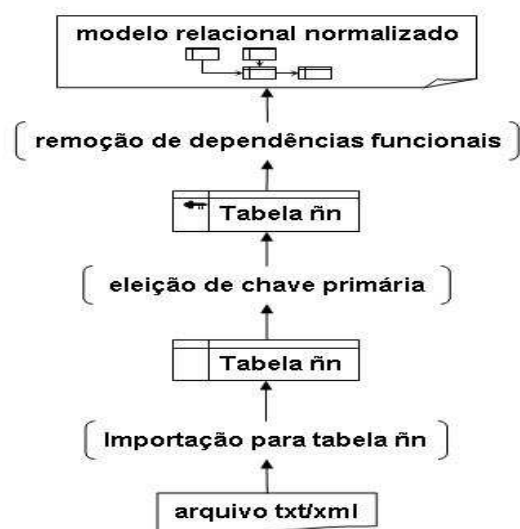
### **3. A Ferramenta**

#### **3.1. Funcionamento Geral**

A Figura 1 apresenta uma visão geral do funcionamento da ferramenta. A entrada da ferramenta é um arquivo de dados em texto ou XML. Após a importação, o usuário pode aplicar um pré-processamento, alterando os nomes dos atributos e seus tipos. Em seguida, o usuário elege uma chave primária, auxiliado pela ferramenta, que descobre os atributos mais aptos a compô-la. Definida a chave primária, o usuário pode então aplicar a 1FN (1ª Forma Normal), 2FN (2ª Forma Normal) e, por fim, a 3FN (3ª Forma Normal). A tabela sofre a aplicação de cada uma das formas normais em passos separados, onde a ferramenta identifica, classifica e remove cada dependência encontrada. Ao término do processo são apresentadas as tabelas resultantes da normalização, sobre as quais a ferramenta pode ser aplicada novamente, de forma recursiva.

#### **3.2. Módulos da Ferramenta**

A plataforma de desenvolvimento dos módulos da ferramenta está toda em SQL, sendo constituída de tabelas auxiliares e *stored procedures* que encapsulam os algoritmos de cada tarefa. O sistema gerenciador de bancos de dados escolhido foi o MySQL 5.1 e a interface gráfica foi desenvolvida utilizando-se a versão 5.5.1 do Netbeans.



**Figura 1. Funcionamento geral da ferramenta.**

### 3.2.1. Módulo de Importação

Este primeiro módulo extrai os dados de um arquivo de texto ou XML e insere em uma tabela preliminar. Para arquivos de texto, o usuário define os caracteres que serão utilizados para quebrar os registros em tuplas, não sendo necessário que o usuário conheça a formatação do arquivo. Uma funcionalidade de *pré-visualização* da ferramenta permite que o usuário observe uma amostra dos dados e descubra quais caracteres separam adequadamente os registros, informe estes caracteres à ferramenta e visualize novamente como os dados ficariam após a importação. Uma vez que o usuário tenha julgado qual o caractere separador adequado, ele realiza a importação propriamente dita.

No caso de arquivos XML, a ferramenta automaticamente efetua a transformação das *tags* em linhas e colunas, ignorando a *tag* raiz e transformando cada ocorrência da *tag* imediatamente descendente em uma tupla. As demais *tags* descendentes, incluindo seus respectivos atributos, são interpretadas como colunas.

A Figura 2 apresenta a tela deste módulo, durante a carga de um arquivo de dados de ligações telefônicas, para um domínio de uma operadora de telefonia.

### 3.2.2. Módulo de Pré-Processamento

Este módulo permite excluir, alterar o nome e o tipo de dado de cada atributo do arquivo importado. A ferramenta analisa previamente os dados e sugere o tipo mais adequado para cada atributo, cabendo ao usuário aceitá-lo ou não. Em caso de dúvida, o usuário pode observar uma amostra dos dados, além de ser impedido de escolher tipos cuja transformação possa causar perda de informação.

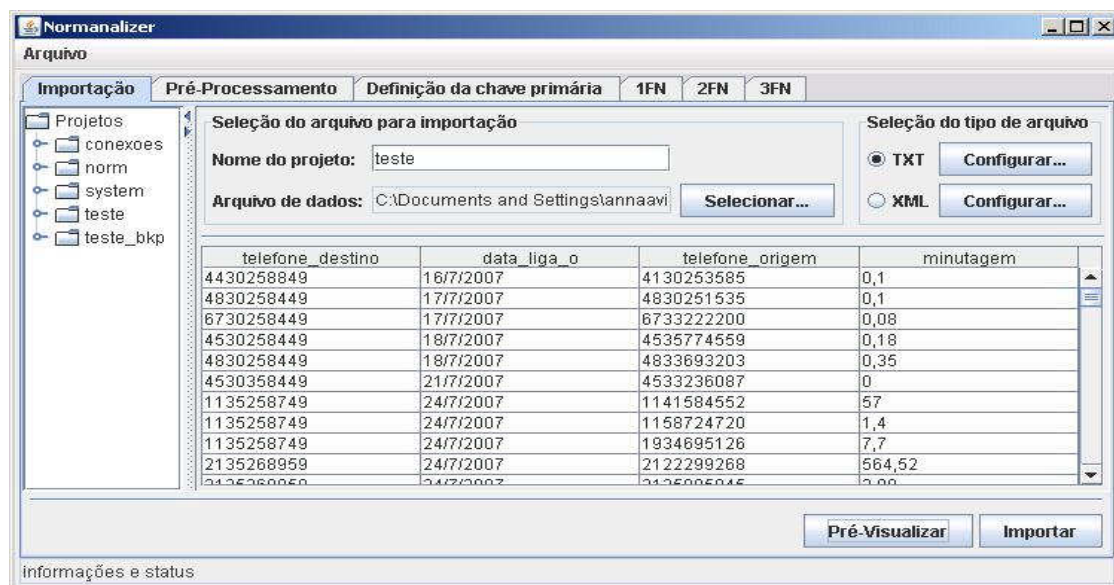


Figura 2. Tela do módulo de Importação.

A Figura 3 apresenta a tela deste módulo. A primeira coluna contém os nomes dos atributos do arquivo importado. A segunda sugere um novo tipo para o atributo. A terceira oferece a opção de renomear o atributo e a última oferece a opção de selecionar um tipo de dado diferente do sugerido pela ferramenta.

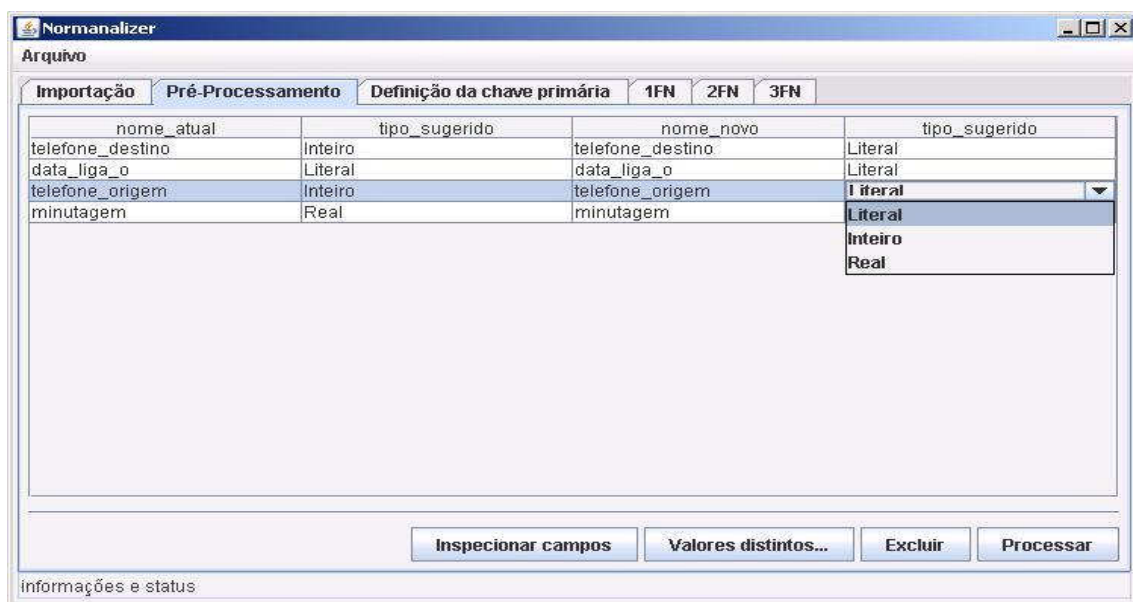
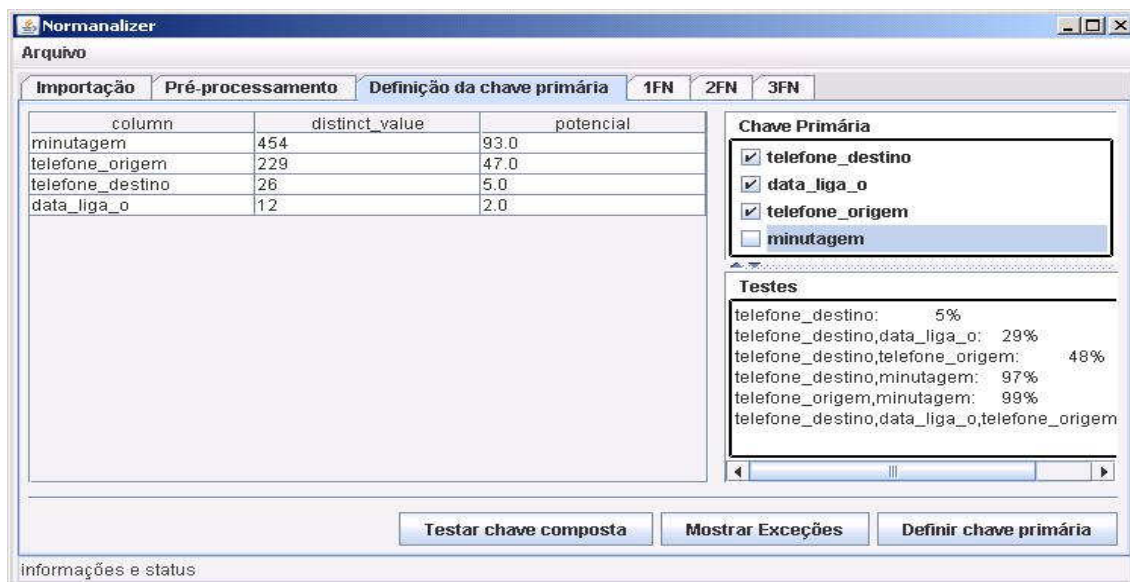


Figura 3. Tela do Módulo de Pré-Processamento.

### 3.2.3. Módulo de Eleição de Chave Primária

Este módulo auxilia o usuário a compor a chave primária da tabela inicial, analisando os dados importados e calculando o potencial de cada atributo (ou combinação de atributos) para ser a chave primária. Este potencial resulta da divisão do número de valores distintos do atributo pelo número total de tuplas da tabela, variando de 0% a 100%, onde 100% indica que os valores do atributo nunca se repetem. A ferramenta

oferece uma amostra dos valores que se repetem, permitindo identificar tuplas integralmente duplicadas, valores corrompidos ou nulos, que podem mascarar uma possível chave primária. Neste caso, a ferramenta adverte o usuário que a escolha de um ou mais atributos de potencial inferior a 100% resultará em perda de informação. O usuário pode, ainda, optar pela geração de uma chave primária artificial. A Figura 4 apresenta a tela deste módulo.



**Figura 4. Tela do módulo de Eleição da Chave Primária.**

### 3.2.4. Módulo da 1FN

O Módulo da 1FN auxilia o usuário na eliminação de atributos multivalorados e tabelas aninhadas, para que a tabela fique em conformidade com a 1FN. A ferramenta varre os dados carregados, a procura de valores não atômicos, isto é, que representem dois ou mais valores. A busca tem como foco caracteres frequentemente utilizados para separar valores em arquivos de dados, como “,” (vírgula), “;” (ponto e vírgula), tabulação, “ ” (espaço), “|” (barra vertical). Para cada aninhamento encontrado, a ferramenta permite ao usuário executar uma dentre as seguintes ações:

- gerar uma nova tupla (repetindo os demais atributos);
- gerar novos atributos para melhor identificar os conteúdos do aninhamento;
- selecionar apenas o primeiro valor;
- ignorar o aninhamento, caso o atributo não seja de fato multivalorado ou o usuário o julgue irrelevante.

A Figura 5 apresenta a tela deste módulo. A primeira coluna lista os atributos, a segunda a frequência de aninhamentos em cada atributo, a terceira o caractere usado para separar os valores e a última as ações possíveis para eliminar os aninhamentos. Se houver aninhamentos, após a correção dos mesmos recomenda-se aplicar novamente o módulo de eleição da chave primária, pois uma nova chave primária pode ser encontrada.

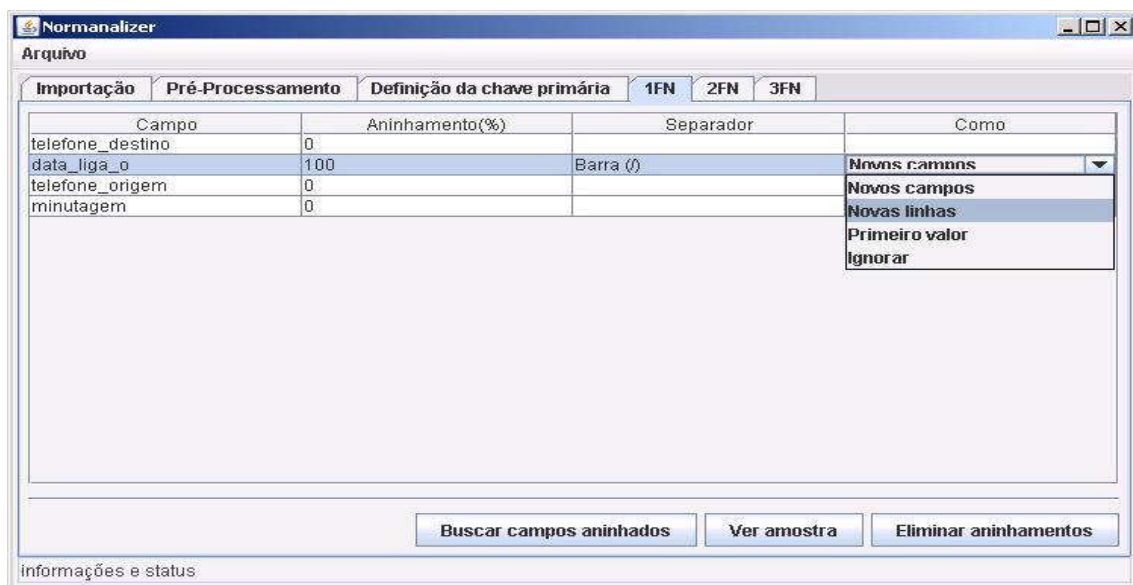


Figura 5. Tela do módulo da 1FN.

### 3.2.5. Módulo da 2FN

O módulo da 2FN ajuda a encontrar dependências funcionais parciais (atributos não-chave cujos valores são determinados por apenas parte dos atributos que compõem a chave primária), através da análise dos dados, e removê-las. Apesar de existirem algoritmos capazes de alcançar a 3FN diretamente, a 2FN é enfatizada para preservar a natureza didática desta ferramenta. A Figura 6 mostra a tela deste módulo.

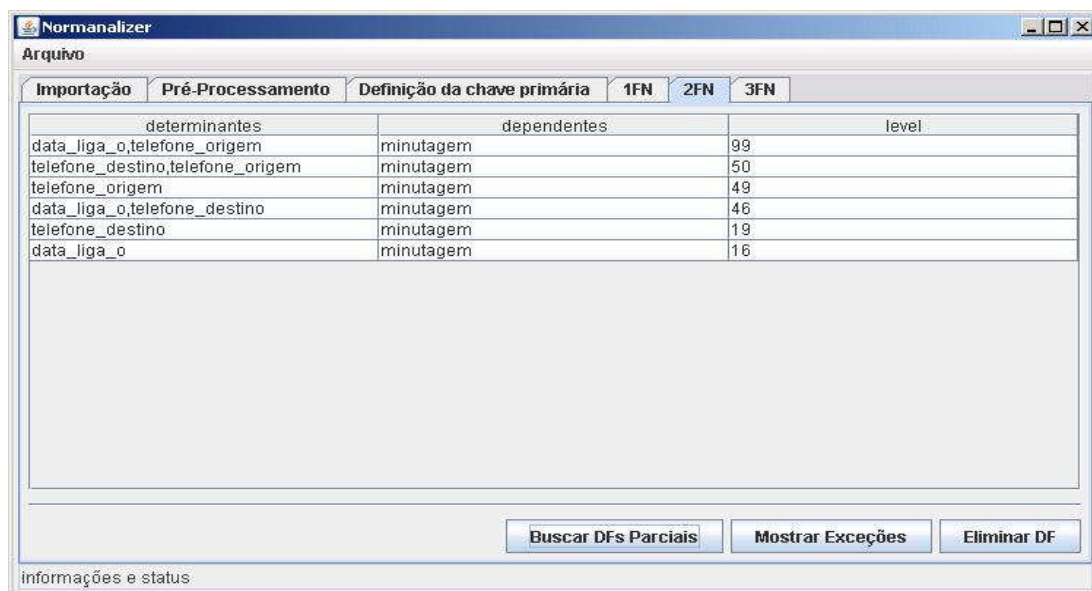


Figura 6. Tela do módulo da 2FN.

Para evitar que erros nos dados possam prejudicar este processo, a ferramenta calcula a frequência com que a dependência funcional ocorre para cada combinação de atributos possível e gera um índice, que varia de 0% a 100%, onde 0% indica que um



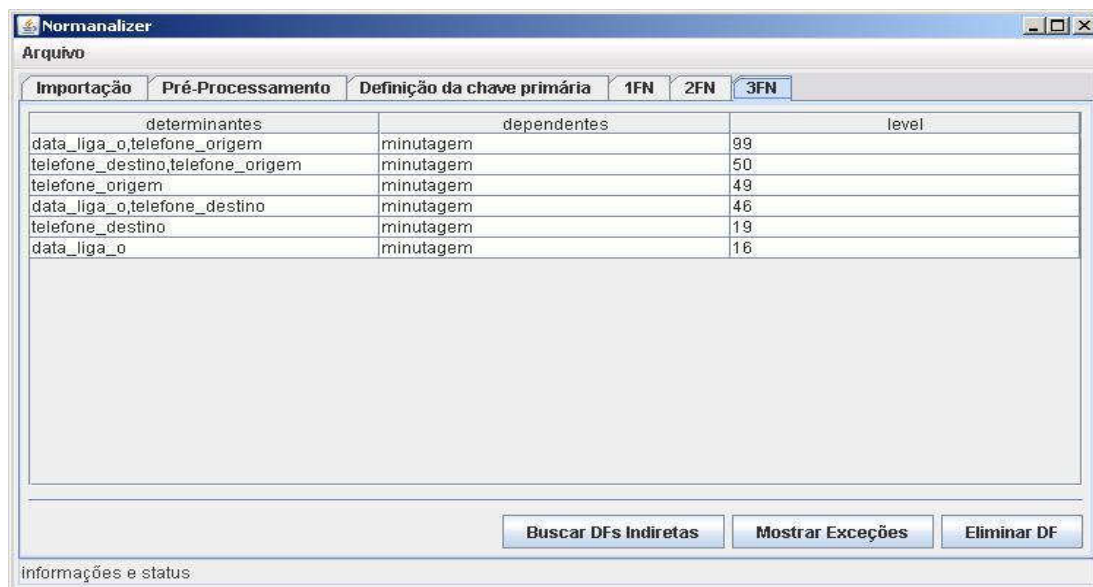
atributo não determina o outro em nenhum de seus valores e 100% indica que sempre determina, o que corresponde à tradicional dependência funcional.

Em caso de dúvidas a respeito do nível de alguma dependência funcional, a ferramenta possibilita a visualização de uma amostra dos tuplas onde a mesma foi violada (exceções), para descobrir se o nível aferido é reflexo de erros ou são características naturais dos dados.

Neste ponto, o usuário remove as dependências funcionais de nível igual a 100% e analisa amostras das que tem nível inferior a 100%, para removê-las também caso observe que não atingiram nível igual a 100% em virtude de erros nos dados. A cada dependência funcional removida, as tabelas resultantes são apresentadas ao usuário. Após a remoção de todas as dependências funcionais, o usuário adquire acesso ao módulo da 3FN.

### 3.2.6. Módulo da 3FN

Este último módulo funciona exatamente como o módulo da 2FN, exceto pelo fato de que a dependência funcional procurada é a indireta, relativa à 3FN. A Figura 7 apresenta a tela deste módulo.



determinantes	dependentes	level
data_liga_o,telefone_origem	minutagem	99
telefone_destino,telefone_origem	minutagem	50
telefone_origem	minutagem	49
data_liga_o,telefone_destino	minutagem	46
telefone_destino	minutagem	19
data_liga_o	minutagem	16

Figura 7. Tela do módulo da 3FN.

Uma vez finalizada a execução deste módulo pelo usuário, obtém-se um esquema relacional, que é o resultado da ferramenta. Este esquema pode ser persistido em um arquivo XML e posteriormente ser, por exemplo, utilizado para gerar um banco de dados relacional ou servir de entrada para outra ferramenta participante de um projeto *bottom-up* de banco de dados. A Figura 8 apresenta um exemplo de XML resultante de uma normalização, para o domínio exemplificado anteriormente.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<schema>
  <table>
    <name>conexoes</name>
    <primary_key>id_telefone_destino_telefone_origem</primary_key>
    <attribute>data_liga_o</attribute>
    <attribute>minutagem</attribute>
  </table>
  <table>
    <name>telefone_destino_telefone_origem</name>
    <primary_key>id_telefone_destino_telefone_origem</primary_key>
    <attribute>telefone_destino</attribute>
    <attribute>telefone_origem</attribute>
  </table>
</schema>

```

**Figura 8. Exemplo do esquema resultante em XML.**

#### 4. Conclusão

O processo de normalização de tabelas, em virtude da carga de conhecimento teórico e intimidade com o domínio dos dados exigidos do projetista, pode se tornar complexo e pouco eficiente, caso não se disponha de um profissional adequadamente qualificado. Além disso, quando se parte de arquivos de dados legados ou tabelas já populadas, há outros fatores agravantes, como a qualidade dos dados, que pode impedir que dependências funcionais sejam descobertas.

O objetivo deste artigo é apresentar uma ferramenta semi-automática para este processo, capaz de reduzir o conhecimento necessário sobre normalização e aproximar o usuário do domínio dos dados. O diferencial da abordagem apresentada está nos algoritmos capazes de detectar dependências funcionais em tabelas, sem que erros nos dados comprometam o resultado da detecção. A ferramenta facilita a execução do processo de normalização, já que grande parte da teoria envolvida está nela embutida, cabendo ao usuário apenas avaliar os cenários oferecidos e decidir a ação a ser tomada. No âmbito acadêmico, esta ferramenta pode ser utilizada como auxílio ao aprendizado da teoria da normalização em disciplinas de bancos de dados.

Testes preliminares realizados com a ferramenta, utilizando arquivos de dados de diversos domínios, foram considerados satisfatórios pelos usuários especialistas nos domínios em questão em termos de usabilidade. Mesmo assim, sabe-se que algumas etapas da normalização são de natureza altamente intelectual e dependem de um projetista que tenha certa intimidade com o domínio dos dados. Portanto, dificilmente a ferramenta poderá vir a realizar um processo totalmente automatizado de normalização. Foram utilizados arquivos de três domínios: pessoas físicas, ligações telefônicas e endereços, com um especialista para cada domínio.

A ferramenta pode ser estendida para detectar dependências funcionais mais complexas, como a 4FN (4ª Forma Normal) e a BCNF (Boyce-Codd Normal Form) [Korth 2006]. Além disso, ela pode ser integrada a ambientes de desenvolvimento de projetos de bancos de dados, já que estes não oferecem nenhuma solução tão detalhada e

flexível para a execução da normalização. Outro possível trabalho futuro é definir uma interface *Web* para a ferramenta, pela alta disponibilidade que este ambiente oferece.

## Referências

- Codd, E. F. 1970. A relational model of data for large shared data banks. *Commun. ACM* 13,6 (Jun.1970), 377-387. DOI=<http://doi.acm.org/10.1145/362384.362685>.
- Heuser, C.A. Projeto de Banco de Dados. 5a edição. Série Livros Didáticos – Instituto de Informática da UFRGS, número 4. Editora Sagra-Luzzatto, 2004.
- Soutou, C. 1998. Relational database reverse engineering: algorithms to extract cardinality constraints. *Data Knowl. Eng.* 28, 2 (Nov. 1998), 161-207. DOI=[http://dx.doi.org/10.1016/S0169-023X\(98\)00017-2](http://dx.doi.org/10.1016/S0169-023X(98)00017-2)
- Korth, H. F.; Sudarshan, S; Silberschatz, A. Sistema de Banco de Dados. 5a edição. Editora Campus, 2006.
- D. Bitton, J. Millman, S. Torgersen, A feasibility and performance study of dependency inference. In: *Proc. 5th Int. Conf. on Data Engineering* (Feb. 1989) pp. 635-641.
- M. Castellanos, F. Saltor, Extraction of data dependencies. In: *Report LSI-93-2-R*, University of Catalonia, Barcelona (1993).
- Chen, T. X., Liu, S. S., Meyer, M. D., and Gotterbarn, D. 2007. An introduction to functional independency in relational database normalization. In *Proceedings of the 45th Annual Southeast Regional Conference* (Winston-Salem, North Carolina, March 23 - 24, 2007). *ACM-SE 45*. ACM Press, New York, NY, 221-225. DOI=<http://doi.acm.org/10.1145/1233341.1233381>.
- Yeh, D. and Li, Y. 2005. Extracting Entity Relationship Diagram from a Table-Based Legacy Database. In *Proceedings of the Ninth European Conference on Software Maintenance and Reengineering* (March 21 - 23, 2005). *CSMR*. IEEE Computer Society, Washington, DC, 72-79. DOI= <http://dx.doi.org/10.1109/CSMR.2005.31>.