Anelim: Uma Ferramenta de Geração Automática de Dados para Banco de Dados Relacional em Ambientes de Testes

Angelo Brayner[1], **F. Ronald Araújo B.**[2], **José Maria Monteiro** [1]

[1] Departamento de Computação – Universidade Federal do Ceará (UFC)

[2]Departamento de Engenharia de Teleinformática – Universidade Federal do Ceará (UFC)

{brayner,monteiro}@dc.ufc.br, f.ronaldaraujo@gmail.com

Resumo. *Há uma crescente demanda pela geração de dados de testes, tanto para validar o projeto lógico e físico de bancos de dados, quanto para testar artefatos de software. Ademais, as aplicações atuais manipulam volumes de dados da ordem de magnitude de terabytes e até petabytes. Contudo, a criação manual de dados para tais cenários é inviável. Neste sentido, foi desenvolvida a ferramenta Anelim*[1]*, a qual tem por finalidade possibilitar a geração automática de grandes volumes de dados de testes. A partir da análise dos resultados obtidos, pode-se afirmar que, diferentemente de outras soluções já existentes, a Anelim gera uma massa de dados consistente, garantindo todas as restrições de integridade especificadas no esquema do banco de dados relacional.*

Abstract. *There is a growing demand for test data, both for validating the logical and physical design of databases, and for testing software artifacts. Nowadays, applications should handle very large databases. However, for such scenarios, creating test datasets in a manual way is not feasible. Thus, we proposed a tool, called Anelim*[1]*, which aims to enable the automatic creation of very large test datasets. From the analysis of the results, we can conclude that, unlike other already existing solutions, Anelim generates a mass of consistent data, ensuring all data integrity restrictions defined in a relational database schema.*

1. Introdução

Atualmente, as aplicações computacionais manipulam volumes de dados nunca antes imaginados. Para ilustrar este fato, pode-se destacar que no final de 2017 existiam 364 milhões de cartões de crédito em uso nos EUA [2]. Neste sentido, para que seja possível testar tais aplicações é imprescindível a existência de uma massa de dados de testes na mesma ordem de grandeza dos banco de dados reais (ou de produção). Desta forma, torna-se imperiosa a utilização de ferramentas que possibilitem a geração automática de conjuntos de dados de teste.

Existem inúmeras ferramentas que povoam automaticamente bancos de dados. Contudo, tais ferramentas não garantem as restrições de integridade especificadas no esquema do banco de dados, como, por exemplo a integridade referencial [Garcia-Molina et al. 2011]. Muitas destas ferramentas geram dados para tabelas específicas e não para o banco de dados como um todo. Na Seção 4.2, várias destas ferramentas serão analisadas e comparadas com a ferramenta proposta.

[1]https://youtu.be/eEpFBpOdNmQ

[2]https://www.creditcards.com/credit-card-news/ownership-statistics.php

Neste artigo será apresentada a ferramenta Anelim, para a geração automática de dados para banco de dados de testes. Estes dados produzidos a partir dos metadados existentes no esquema do banco de dados. Diferentemente das ferramentas existentes, a Anelim gera uma massa de dados consistente, onde as restrições de integridade do modelo relacional especificadas no esquema são garantidas.

Este artigo está estruturado da seguinte forma. A Seção 2 descreve a ferramenta proposta. Na Seção 3, são apresentados os tipos de dados suportados pela *Anelim*. Por sua vez, a Seção 4 apresenta resultados obtidos com a execução da *Anelim*, bem como, analisa ferramentas concorrentes da apresentada neste artigo. Por fim, a Seção 4.2 conclui este trabalho.

2. A *Anelim*

A ferramenta *Anelim* foi desenvolvida com o objetivo de gerar dados aleatórios para bancos de dados relacionais a partir do seu esquema lógico. Para tanto, a ferramenta deve apresentar as seguintes propriedades. Primeiramente, deve ter a capacidade de ler o esquema do banco de dados. Com esta propriedade pretende-se garantir as restrições de domínio (tipo de dados), chave (chave primária e chave candidata) e de integridade referencial especificadas nos metadados. Em segundo lugar, deve-se garantir ao usuário a possibilidade de especificar o volume de dados a ser gerado. Este volume refere-se ao banco de dados todo ou pode ser especificado para cada tabela. Por último, deve-se garantir ao usuário a opção de inserir ou não os dados automaticamente no banco de dados. Estas três propriedades, implementadas na ferramenta desenvolvida, introduziram um alto grau de complexidade à sua construção.

Com relação à garantia das restrições de integridade, a real dificuldade está em garantir a restrição de integridade referencial (IR). Isto se deve ao fato que a IR impõe uma ordem de inserção de tuplas nas tabelas, para que não seja violada a IR. Para ilustrar este problema, considere a modelagem conceitual entre os conjuntos de entidades Departamentos e Funcionários, apresentado na Figura 1. Observe que uma relação de cardinalidade um-para-muitos entre os dois conjuntos de entidades, onde Departamentos situa-se no lado *um* e Funcionários no lado *muitos*.

Desta forma, no esquema do banco de dados correspondente, o atributo *DepartamentoId* da tabela Funcionários é definido como chave estrangeira. Por este motivo, *DepartamentoId* deve possuir um valor que faz referência a um valor de chave primária na tabela Departamentos. Neste caso específico, o atributo *DepartamentoId* não pode assumir valor nulo, pois Funcionários apresenta participação total no relacionamento, ou seja todo funcionário deve estar lotado em um departamento. Portanto, não é possível inserir um funcionário sem informar em que departamento o mesmo está associado.

Em outras palavras, para gerar dados para o atributo *DepartamentoId*, a ferramenta precisa ler os valores gerados para a chave primária da tabela Departamentos. Caso contrário, a inserção de dados em Funcionários seria afetada, pois poderia haver muitos erros de violação da integridade referencial.

Para garantir todas as restrições de integridade do modelo relacional, a ferramenta proposta tem como entrada um arquivo de configuração, contendo os metadados do banco de dados. O formato padrão escolhido foi o *JavaScript Object Notation* (JSON). O ar-

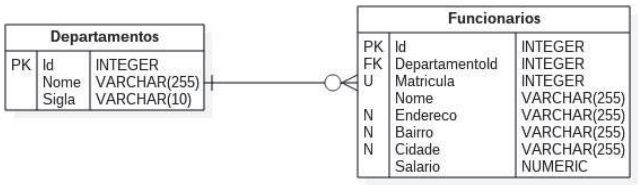

Figura 1. Entidades Departamentos e Funcionários.

quivo de configuração contém os metadados e parâmetros de geração de dados apresentados na Tabela 1. Adicionalmente, a *Anelim* solicita ao usuário os parâmetros apresentados na Tabela 2.

Tabela 1. Atributos do arquivo de configuração

Nome	Especificação	Obrigatório
tables	*Array* de objetos contendo as características de cada tabela.	Sim
number_inserts	Número de tuplas a ser inseridas em cada tabela.	Sim
tables → *name*	Nome da tabela.	Sim
fields	*Array* de objetos contendo as características de cada atributo da tabela.	Sim
fields → *name*	Nome do atributo.	Sim
primary_key	Informa se o atributo é chave primeira.	Não
type	Informa qual o tipo do atributo.	Sim
foreign_key	Informa se o atributo é chave estrangeira.	Não

3. Tipos de Dados Suportados

Atualmente, a *Anelim* suporta dois sistemas gerenciadores de bancos de dados (SGBDs), a saber: SQL Server e PostgreSQL. Cada um desses sistemas de bancos de dados apresenta um conjunto de diferentes tipos de dados. Todavia, oferecer suporte a todos esses tipos de dados específicos aumenta a complexidade do processo de geração de dados. Neste sentido, definiu-se um conjunto comum de tipos de dados suportados. O critério estabelecido para a escolha desses tipos foi a interseção entre estes conjuntos. A Tabela 3 mostra os tipos de dados que a *Anelim* pode gerar.

4. Experimentos e Análise Comparativa

4.1. Análise de Desempenho da *Anelim*

O principal objetivo desta análise foi averiguar o comportamento da *Anelim* no ato da geração e inserção dos dados nos dois SGBDs suportados. Para realizar o comparativo de desempenho, foi usado o esquema do banco de dados *Northwind*. Este banco de dados é composto de oito tabelas. O relacionamento entre estas tabelas é apresentado na Figura 2.

Por sua vez, a figura 3 mostra o gráfico dos tempos de execução para geração e inserção de dados no banco de dados, tanto no SQL Server, quanto no PostgreSQL.

Tabela 2. Atributos do arquivo de configuração

Nome	Valor Padrão	Funcionalidade
-f, --file	schema.json	Permite que o usuário informe o nome do arquivo contendo o schema do banco de dados.
-t, --target	mssql	Permite que o usuário informe a sintaxe que deverá ser utilizada na criação dos dados e/ou tabelas.
-d, –drop	false	Quando sinalizada com true, gera um script de “DROP TABLE” acima do script de inserção.
-c, --create	false	Quando sinalizado com true, gera um script de “CREATE TABLE” acima do script de inserção.
-i, --insert	false	Quando sinalizado com true, executa de forma transacional o script dentro do banco de dados.
--debug	false	Quando sinalizado com true, ativa o modo debug da ferramenta informando o seu passo-a-passo.

Tabela 3. Relação de tipos de dados aceitos

Tipo do dado	SQL Server	PostgreSQL
smallint	×	×
integer	×	×
bigint	×	×
decimal	×	×
real	×	×
serial		×
money	×	×
varchar	×	×
date	×	×
time with time zone		×
boolean		×
uuid		×
bit	×	×

Observe que cada tabela do Northwind foi povoada com uma quantidade de 10, 100 e 1000 tuplas.

Vale destacar que os resultados apresentados na Figura 3 foram obtidos com a execução de comandos de *insert* convencional. Em outras palavras, para cada *insert*, o SGBD verificava a garantia das restrições de integridade, como, por exemplo, restrição de chave, de identidade e integridade referencial. Atualmente, estamos implementando estratégias mais eficientes para inserção de grandes volumes de dados. Para o SQL Server, por exemplo, estamos implementado a estratégia de *bulk insert*.

4.2. Análise Comparativa

Esta seção apresenta uma análise comparativa entre a *Anelim* e as principais ferramentas relacionadas: DataFiller, Database Test Data, GenerateData, DGMaster e Mockaroo. Os critérios utilizados nesta comparação foram: i) a observâncias das restrições de integridade definidas no esquema lógico e ii) os SGBDs suportados.

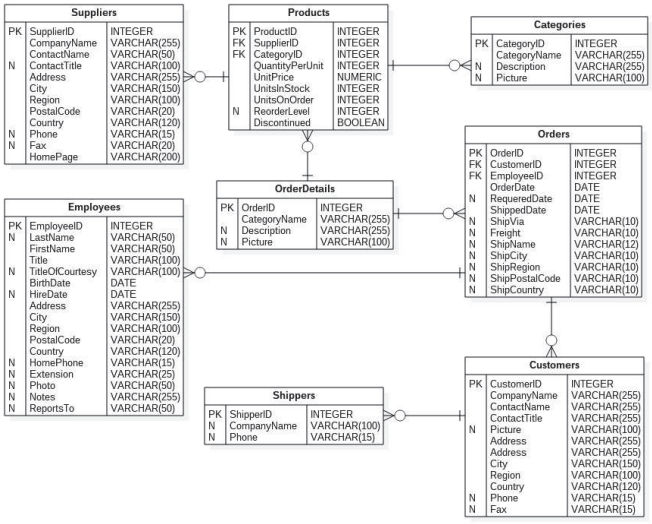

Figura 2. Banco de dados Northwind.

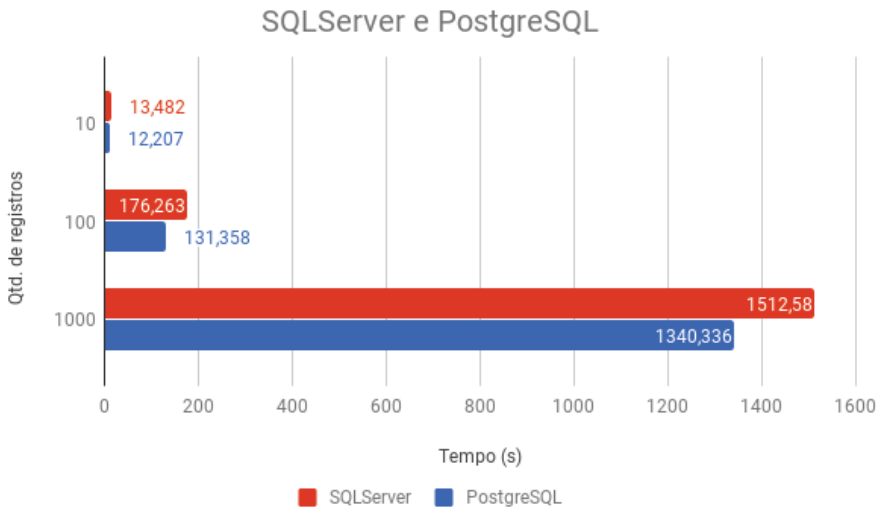

Figura 3. Desempenho da ferramenta.

A ferramenta DataFiller [DataFiller 2014] implementa uma estratégia de inserção por tabela. Assim, é possível informar a quantidade e probabilidade de geração de dados para cada atributo da tabela. Contudo, esta ferramenta não garante a restrição de integridade referencial de forma satisfatória. Tal característica, induz a muitos erros de violação de integridade referencial durante a inclusão dos dados gerados. Consequentemente, a DataFiller não garante que a quantidade de dados solicitados pelo usuário seja gerada de forma satisfatória. Por fim, esta ferramenta só consegue gerar e inserir dados para o PostgreSQL.

As ferramentas GenerateData [GenerateData 2017] e Database Test Data [Test Data Generator 2018] não permitem a inserção automática no banco de dados, apenas geram arquivos contendo os dados gerados. Consequentemente, não garantem as restrições de integridade do modelo relacional.

Por sua vez, a DGMaster [DGMaster 2017] apresenta suporte a vários tipos de

dados, gerando-os em diversos formatos (por exemplo, texto e XML). Contudo, além de apresentar uma interface confusa e de difícil utilização, apresenta instabilidade operacional. Finalmente, com a capacidade de gerar grandes volumes de dados em vários modelos de dados diferentes (relacional, XML, planilhas Excel, entre outros), a ferramenta [Mockaroo 2017] não apresenta suporte para povoar o banco de dados de forma automática.

A Tabela 4 ilustra o resultado da análise comparativa realizada. Pode-se perceber, analisando a Tabela 4, que *Anelim* é a única ferramenta a respeitar as restrições de integridade definidas no esquema lógico. Ademais, a *Anelim* fornece suporte para dois diferentes SGBDs.

Tabela 4. Análise comparativa entre ferramentas de geração de dados

Ferramenta	*Respeita as RIs*	*SGBDs Suportados*
Anelim	Sim	2
Database Test Data	Não	0
DataFiller	Não	1
GenerateData	Não	0
DGMaster	Sim	0
Mockaroo	Não	0

5. Considerações Finais

Neste trabalho, apresentou-se uma ferramenta, denominada Anelim, capaz de gerar automaticamente dados de teste em banco de dados relacionais. Atualmente, a Anelim é capaz de povoar bancos de dados em dois SGBDs, o SQLServer e o PostgreSQL. A partir da análise dos resultados obtidos, pode-se inferir que a *Anelim* apresenta diversas vantagens em relação às ferramentas concorrentes avaliadas neste trabalho. Atualmente, estão sendo implementadas estratégias para inserção de grandes volumes de dados e suporte para inserção de dados no MySQL.

6. Referências

Referências

DataFiller (2014). Generate random data from database schema. `https://www.cri.ensmp.fr/people/coelho/datafiller.html`. Abril.

DGMaster (2017). Data generator: simple, free, extensible. `http://dgmaster.sourceforge.net/`. Maio.

Garcia-Molina, H., Ullman, J., and Widom, J. (2011). *Database Systems: The Complete Book*. Pearson Education.

GenerateData (2017). Script generate random data from a database schema. `http://www.generatedata.com/`. Abril.

Mockaroo (2017). Random Data Generator and API Mocking Tool. `https://mockaroo.com/`. Maio.

Test Data Generator, D. (2018). Fill your database with random test data. `http://www.databasetestdata.com/`. Abril.