

MODELAGEM DE PROCESSOS DE REFATORAÇÃO DE BANCO DE DADOS UTILIZANDO BPMN

Márcia Beatriz Pereira Domingues (Universidade de São Paulo, São Paulo, Brasil) - beatrizpereira@usp.br

Jorge Rady de Almeida Jr. (Universidade de São Paulo, São Paulo, Brasil) - jorgerady@usp.br

Francisco Carlos Paletta (Universidade de São Paulo, São Paulo, Brasil) - fcpaletta@usp.br

Projecting and maintaining a database is an important challenge, due to the frequent requirement of changes demanded by the users. To support those changes, the database schema must go through structural modifications that, many times, affect its performance and the queries' results, such as: un necessary relationships, primary and foreign keys strongly attached to the domain; obsolete attributes and types of attributes which are inadequate. The literature about Agile Methods for software development suggests the use of refactoring for the evolution of the database's schema, when there are requirement changes. A refactoring is a simple change that improves the design but does not alter the database semantics nor adds new functionalities. Each refactoring has a group phases to be performed, which makes the implementation process possible. However, some changes on the database demand many refactoring to be made in the same process, which means that they can be combined and many steps can be suppressed. In this context, the present work aims at presenting how the refactoring phases can be modeled using processes notation

Keywords: Database Refactoring, BPMN, Database Maintenance

MODELAGEM DE PROCESSOS DE REFATORAÇÃO DE BANCO DE DADOS UTILIZANDO BPMN

O projeto e manutenção de bancos de dados é um importante desafio, tendo em vista as frequentes mudanças de requisitos solicitados pelos usuários. Para acompanhar essas mudanças o modelo do banco de dados deve passar por alterações estruturais que muitas vezes prejudicam o desempenho e o projeto das consultas, tais como: relacionamentos desnecessários, chaves primárias ou estrangeiras criadas fortemente acopladas ao domínio, atributos obsoletos e tipos de atributos inadequados. A literatura sobre Métodos Ágeis para desenvolvimento de software propõe o uso de refatorações para evolução do modelo do banco de dados quando há mudanças de requisitos. Uma refatoração é uma alteração simples que melhora o design, mas não altera a semântica do modelo de dados, nem adiciona novas funcionalidades. Cada refatoração possui um conjunto de etapas a serem executadas, o que possibilita a implementação de processos. No entanto, algumas alterações no banco de dados exigem que várias refatorações sejam feitas em um mesmo processo, ou seja elas podem ser combinadas e muitos passos podem ser suprimidos. Nesse contexto, o presente trabalho tem como objetivo, apresentar como as etapas das refatorações podem ser modeladas utilizando a notação de processos BPMN (*Business Process Model and Notation*) e apresentar uma proposta para executar um processo no qual, seja realizada uma grande alteração no banco de dados. Os recursos da notação BPMN são utilizados para auxiliar na documentação, formalização e execução de processos de refatoração de banco de dados.

Palavras-chave: Refatoração de Banco de Dados, BPMN, Manutenção de Banco de Dados.

1. INTRODUÇÃO

A manutenção de bancos de dados tem sido reconhecida como um dos principais obstáculos que dificultam as atualizações em sistemas de informação (Ambler & Sadalage, 2006). Estudos mostram que essas dificuldades aumentaram com o uso de sistemas Web, nos quais a frequência de mudanças é muito grande e praticamente não há tolerância para erros ou indisponibilidade dos serviços.

Alguns conceitos do modelo tradicional foram substituídos pelas Metodologias Ágeis para Desenvolvimento de Software, por modelos iterativos e incrementais, que não precisam ter todo o modelo lógico e físico do banco de dados prontos para o início do desenvolvimento do software (Fowler, Beck, Brant, Opdyke, & Roberts, 1999). Isso significa organizar o desenvolvimento do sistema em várias iterações para entregas parciais ao cliente. Cada iteração contempla todas as fases já conhecidas nos modelos tradicionais como planejamento, especificação de requisitos, projeto, codificação e testes (Pressman, 2011). Essas iterações que normalmente duram em torno de duas semanas, têm como principais vantagens a detecção antecipada de erros de requisitos e de implementação e também a flexibilidade de mudar os requisitos no decorrer do projeto. Para atender a essas mudanças de requisitos é necessário que o banco de dados também tenha um desenvolvimento iterativo, ou seja, à medida que o software é desenvolvido, o banco de dados precisa ser alterado para acompanhar os novos requisitos (Roddick, 1995). As alterações no modelo de banco de dados são chamadas de refatorações de banco de dados.

Segundo Fowler et al., (1999), refatorar um banco de dados é mais trabalhoso do que refatorar um código-fonte de uma aplicação, embora possa seguir as mesmas premissas. Tal afirmação baseia-se no fato de que é preciso se preocupar com todas as instâncias das aplicações que utilizam o banco de dados; realizar as alterações respeitando os requisitos dos usuários; preservar os dados existentes; não modificar a semântica do modelo atual e manter a integridade do banco de dados.

As técnicas de refatoração de código foram adaptadas para banco de dados e inseridas na literatura de Métodos Ágeis para Desenvolvimento de Software. Uma refatoração de banco de dados foi definida como uma pequena mudança, que melhora o projeto, não adiciona funcionalidades e não altera as informações ou comportamentos do banco de dados. Ambler e Sadalage (2006), organizaram as refatorações conhecidas na literatura de banco de dados em quatro grupos: estrutural, qualidade de dados, integridade referencial e arquitetural. Cada refatoração possui um conjunto de etapas a serem executadas. Cada etapa possui um conjunto de restrições que não devem ser negligenciados para que se garanta a integridade, consistência e desempenho do banco de dados.

Um dos grandes desafios é implementar um processo no qual, várias refatorações possam ser executadas, de modo a respeitar a ordem, na qual cada etapa deve ser executada. Esse processo também deve ser capaz, de prevenir erros, de modo que em uma

eventual falha, todas as refatorações, ou etapas de refatorações já completadas, possam ser refeitas sem que os dados armazenados sejam perdidos.

Ambler e Sadalage (2006) apresentaram um catálogo de refatorações que incluem os passos para execução de cada uma. No entanto, algumas alterações no modelo exigem que várias refatorações sejam feitas em um mesmo processo, ou seja, elas podem ser combinadas e muitos passos podem ser suprimidos.

Nesse contexto, o presente trabalho tem como objetivo, apresentar como as etapas das refatorações podem ser modeladas em BPM e apresentar uma proposta para executar um processo no qual, seja realizada uma grande alteração no banco de dados.

A representação em BPMN abrange não somente os aspectos físicos do modelo do banco de dados, mas também as interações e decisões do usuário, a documentação de todo o processo e a possibilidade de retornar ao estado antes das refatorações, caso haja erros ou os resultados não sejam satisfatórios.

1.1. Refatoração de Banco de Dados

Segundo Ambler e Sadalage (2006) uma refatoração em um banco de dados pode ser definida como uma alteração simples em seu modelo, com o objetivo de melhorar seu projeto, preservando sua semântica informacional e sua semântica comportamental. Assim, para se realizar uma refatoração não é possível adicionar novas funcionalidades ou modificar os dados existentes.

Em outras palavras, após a execução de uma refatoração ou um conjunto de refatorações no modelo de dados, o banco de dados deve responder às mesmas consultas que eram realizadas antes deste processo de refatoração.

A estratégia de refatoração definida por Ambler e Sadalage (2006) é aplicada em ambientes complexos, nos quais existem várias aplicações acessando simultaneamente o mesmo banco de dados. Quando esse banco precisa ser alterado, praticamente todas as aplicações precisam sofrer algum tipo de adaptação, sendo que não é possível supor que todas essas aplicações serão alteradas e implantadas no mesmo momento (Domingues, 2011). Por esse motivo, necessita-se de um período de transição entre o início e o fim de uma refatoração, no qual o modelo antigo e o modelo novo coexistam, como mostrado na Figura 1.

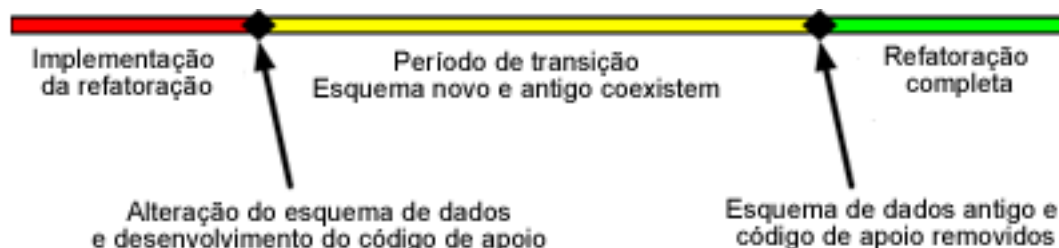


Figura 1 - Ciclo de vida de uma refatoração

Fonte: adaptado de Ambler & Sadalage (2006)

Ambler e Sadalage (2006) destacaram alguns problemas no banco de dados que podem exigir refatorações como: tabelas ou colunas com múltiplas funções; tabelas com muitas colunas; e tabelas com colunas ou associações obsoletas.

Para abranger todos esses problemas e outros que podem interferir no bom desempenho do modelo de dados, e também para atender mudanças de requisitos, Ambler e Sadalage (2006) escreveram um catálogo para as refatorações, o qual foi amplamente discutido por Domingues (2011).

Essas refatorações são apresentadas a seguir, divididas em 4 grupos: estrutural, qualidade de dados, integridade referencial e arquitetural.

i. Estrutural

As refatorações estruturais têm como objetivo melhorar o modelo do banco de dados. Mudanças de requisitos são as principais causas para se alterar a estrutura do modelo do banco de dados. As refatorações estruturais podem incluir alterações no modelo do banco de dados, tais como:

- Remover tabelas, colunas ou visões;
- Dividir ou unir tabelas ou colunas;
- Renomear tabelas, colunas ou visões.

ii. Qualidade de Dados

As refatorações de qualidade de dados têm como objetivo aumentar as validações e diminuir as inconsistências do banco. As refatorações de qualidade de dados podem incluir ações como:

- Introduzir restrição de coluna, valor padrão ou formato comum;
- Aplicar código ou tipo padrão;
- Consolidar estratégias de chaves;
- Remover restrição de coluna, valor padrão ou restrição de não nulo;
- Alterar colunas para receber valores nulos.

iii. Integridade Referencial

As refatorações de integridade referencial têm como objetivo manter consistentes as referências entre as tabelas, incluindo ou removendo regras, armazenando históricos ou implementando lógicas da aplicação no banco de dados. As refatorações de qualidade de integridade referencial podem incluir ações como:

- Adicionar ou remover restrição de chave estrangeira;
- Adicionar ou remover a exclusão em cascata.

iv. Arquitetural

O objetivo principal das refatorações arquiteturais é mover uma parte do código dos aplicativos para o banco de dados (lógica no banco de dados). Os objetivos secundários são organizar, padronizar e melhorar o desempenho do banco de dados. As refatorações arquiteturais podem incluir ações como:

- Introduzir índice, tabela espelho, tabela somente para leitura;
- Adicionar métodos (*Stored Procedures*).

1.2. Modelagem de Processos

A ideia básica de um processo é identificar o conjunto de atividades que tem de ser realizada para alcançar um objetivo específico e definir uma ordem em que essas tarefas devem ser realizadas (Decker, Che, Oberweis, Stürzel, & Vogel, 2010).

A modelagem visa criar um modelo de processos por meio da construção de desenhos e diagramas operacionais sobre seu comportamento. A modelagem valida o projeto, testando suas reações sobre diversas condições para certificar que seu funcionamento atenderá aos requisitos globais estabelecidos, além de medir seu desempenho no ponto de vista de qualidade, desempenho e custo (ABPMP, 2009; Silva, 2013; Valle & Oliveira, 2013) (ABPMP, 2009; Silva, 2013).

Um modelo de processo pode conter: um ou mais diagramas; informações de objetos; informações de relacionamentos entre os objetos; informações de iteração das atividades do processo com pessoas, com sistemas de informações ou funções automatizadas (ABPMP, 2009).

1.3. Modelagem de Processos

Dentre as técnicas para modelagem de processos mais difundidas atualmente estão: Fluxogramas, BPMN (*Business Process Modeling Notation*), UML (*Unified Modeling Language*), IDEF0 e IDEF3 (*Integrated Definition*) e Redes de Petri (Giaglis, 2001). Entretanto BPMN é uma das notações mais amplamente utilizada para modelar processos (Valle & Oliveira, 2013). Essa notação é mantida pela OMG (*Object Management Group*) com suporte de empresas como IBM, Microsoft, Unisys e outras. A OMG também é responsável por manter as especificações da UML.

Em 2005, um grupo conhecido por *Business Process Management Initiative* (BPMI) iniciou o desenvolvimento de uma notação gráfica para representação de processos de negócio chamada BPMN. Neste mesmo ano, o *Workflow Management Coalition* (WfMC), também começou a construir um padrão formal, escrito para a representação de processos, chamado *XML Process Definition Language* (XPDL), porém sem representação gráfica. A XPDL 2.0 é uma linguagem para descrição de *workflows* (Valle & Oliveira, 2013).

A notação BPMN passou a ser mantida pela OMG em 2006 que uniu a notação BPMN com a notação XPDL e hoje a notação BPMN é considerada uma linguagem formal de representação. Essa integração facilitou o desenvolvimento e utilização em ferramentas diferentes, desde que essas ferramentas de modelagem sejam construídas conforme os padrões publicados pela OMG.

Segundo Valle e Oliveira (2013) um dos pontos fortes do BPMN é o fato de que os diagramas possam ser integrados a um ambiente operacional e automatizado. Outro ponto forte é assegurar que as linguagens XML desenhadas para execução de processos de negócio, como BEPEL4WS (*Business Process Execution Language for Web Services*), possam ser visualizadas dentro de uma notação orientada a processos.

A sequência de passos necessários para que se possa atingir a automação de processos de negócio, de acordo com um conjunto de regras definidas é chamada de *workflow* (ABPMP, 2009). Esta automação pode ser no todo ou em parte do processo de negócio, durante a qual documentos, informações, ou tarefas são passadas de um participante para outro (Aalst, 2004).

A automação de processos é feita por meio de uma categoria de *software* chamada de BPMS (*Business Process Management Systems*) que automatiza a gestão de processos de negócio.

Há no mercado diversas ferramentas que dão suporte a construção de *workflows* como: BizAgi BPM Suite, Bonita, AgilePoint BPMS, JBossjBPM e Oracle BPM Suite. A maioria dessas ferramentas disponibiliza, gratuitamente, o módulo para modelagem de processos e cobra para utilizar módulos que dão suporte a automatização, integração com bancos de dados ou monitoramento em tempo real.

1.4. Conceitos Básicos da notação BPMN

As atividades de um processo podem ser executadas por diferentes atores: usuários e sistemas, portanto, um processo é uma forma de trabalho colaborativo. Um processo de negócio pode incluir a execução de atividades em paralelo e iterações entre subprocessos (Decker et al., 2010).

Embora a notação BPMN seja rica em elementos de modelagem, os elementos mais utilizados na modelagem de processos de negócio são 4: atividades (*tasks*), eventos (*events*), decisões (*gateways*) e sequências de fluxos (*sequenceflows*), como mostra a Figura 2. Com apenas esses quatro elementos é possível construir modelos de processos bastante intuitivos e de fácil leitura (Valle & Oliveira, 2013).

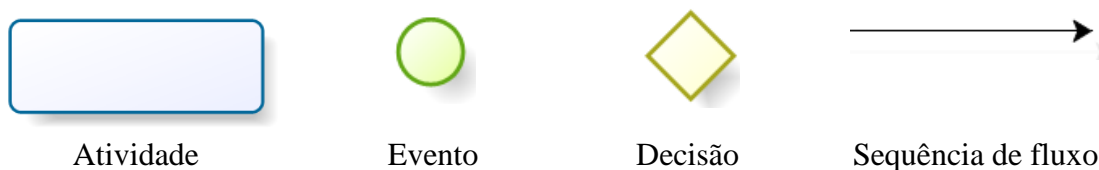


Figura 2 – Elementos Básicos da notação BPMN

Fonte: Adaptado de OMG (2011)

- **Atividade (*Activity*):** Uma Atividade é um termo genérico do trabalho realizado em um processo. Uma atividade pode ser atômica ou composta. Se ela for atômica não possui outras atividades dentro dela, então ela é definida também uma Tarefa (*task*). Se ela for composta, ou seja, contém outras atividades dentro dela, ela é classificada como um processo ou subprocesso, como é mostrado na Figura 3.



Figura 3 – Tipos de Atividade da notação BPMN

Fonte: Adaptado de OMG (2011)

- **Evento (*Event*):** Um evento é algo que acontece durante o curso de um processo e afeta o seu fluxo. Há três tipos de eventos, com base em quando e como eles afetam o fluxo: de início, intermediário e de fim, como mostra a Figura 4.



Figura4 – Tipos de Eventos da notação BPMN

Fonte: Adaptado de OMG (2011)

- **Conector de Sequência de fluxo (*Sequenceflow*):** Um fluxo de sequência mostra a ordem em que as atividades serão executadas no processo. O conector indica que existe uma dependência e uma prioridade entre as atividades conectadas para seu início e fim. Uma sequência de fluxo conecta atividades, decisões e eventos uns aos outros dentro de uma piscina (pool) (Rademakers, 2012). Tipicamente, uma atividade terá uma única entrada e uma única saída (White & Miers, 2008). A Figura 5 mostra os diferentes tipos de fluxo de sequência.

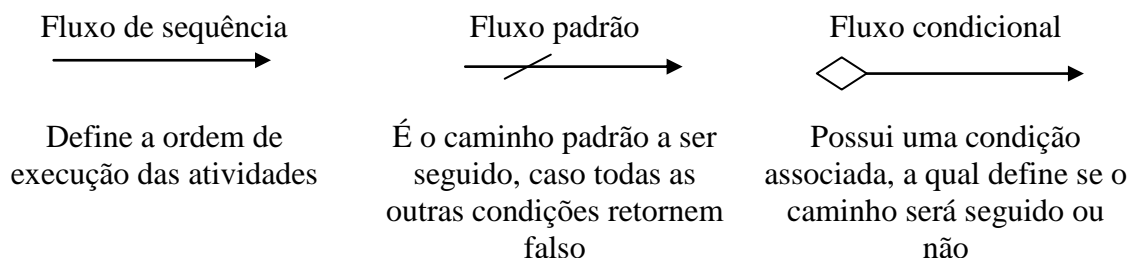


Figura 5 – Tipos de fluxo de sequência da notação BPMN

Fonte: adaptado de SILVA, 2013

- **Decisão (gateway):** É um elemento de modelagem utilizado para controlar como a sequência do fluxo interage dentro de um processo ao convergir ou divergir. Um *gateway* pode ser entendido como uma pergunta que é feita em um ponto no fluxo do processo. A Figura 6 mostra os principais tipos de decisão da notação BPMN e abaixo são descritos seus diferentes tipos de uso.

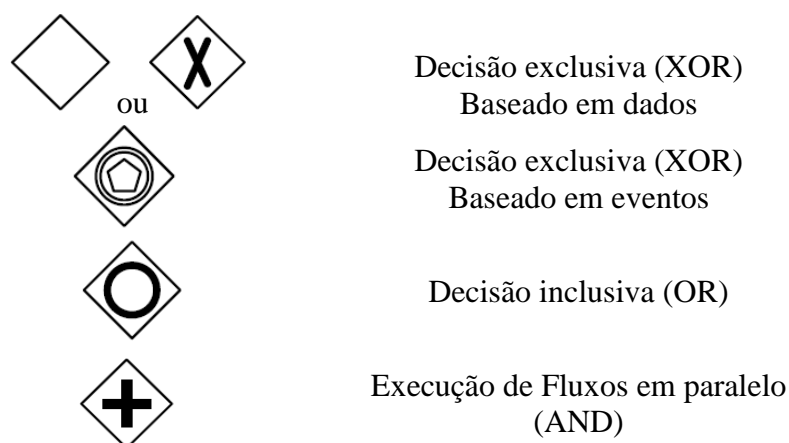


Figura 6 – Tipos de Gateways da notação BPMN

Fonte: adaptado de SILVA, 2013

2. MODELAGEM DE REFATORAÇÕES DE BANCO DE DADOS EM BPMN

A Modelagem de processos de negócio é uma solução muito utilizada para análise de requisitos de sistemas em conjunto com o desenvolvimento de software tradicional (Phalp & Shepperd, 2000). O avanço de sistemas como de comércio eletrônico levaram profissionais de Tecnologia da Informação (TI) a buscar alternativas para gerenciar os fluxos de atividades em seus sistemas. Isso em conjunto com o avanço das ferramentas e linguagens para modelagem de processos possibilitaram que profissionais de TI utilizassem recursos que antes eram empregados somente por profissionais de administração, engenharia de produtos e engenharia de produção para modelagem ou melhoria de processos internos (Mili et al., 2010).

O uso de processos para refatorações de banco de dados foi proposto por Ambler e Sadalage (2006), porém os autores não fizeram uma análise do processo de refatoração, levando em conta seu diagrama, as atividades e condições envolvidas. A notação BPMN mostrou-se uma linguagem simples para entender e ser usada por diferentes perfis de usuários (administradores, desenvolvedores, analistas ou usuários finais) e o ponto de vista de refatoração de bancos de dados, abrange todas as necessidades de modelagem de uma grande alteração.

2.1. Estudo de Caso

Para demonstrar o uso da notação BPMN para modelagem de refatorações de banco de dados, foi utilizado um banco de dados geográfico do sistema desenvolvido no projeto FINEP-ESALQ do Laboratório de Automação Agrícola da Escola Politécnica da Universidade de São Paulo (USP) (<http://www.laa.pcs.usp.br/html/pagina.php?p=projetos>). Diversos aspectos do banco de dados foram analisados para a conclusão que ele seria adequado para esta modelagem, tais como: problemas no projeto, consultas lentas e necessidade de diminuir a complexidade de *scripts* de seleção e inserção de dados.

Esse sistema tem como objetivos: monitoramento, tomada de decisão, recomendação e intervenções para aplicação de fertilizantes em taxa variável e avaliação de indicadores em amostras de solo. As amostras são relacionadas à adubação nitrogenada, prioritariamente nas culturas da cana-de-açúcar, algodão, trigo e milho. Para modelar no banco de dados as áreas onde serão coletadas as amostras de solo, as fazendas foram divididas em várias partes chamadas de talhões. Os talhões são criados para alternar as culturas de acordo com a safra, possibilitando um controle mais apurado dos custos de produção e as necessidades de reposição de nutrientes de cada talhão.

Os talhões são representados no banco de dados por suas dimensões geográficas, assim é possível utilizando ferramentas de georeferenciamento do SGDB (Sistema de Gerenciamento do Banco de Dados) mostrar um desenho geométrico do talhão. Com o desenho geométrico do talhão pronto, é possível dividir esse espaço em pequenos polígonos que representam o local geográfico de cada amostra de solo coletada para análise. Como resultado dessa análise, o sistema fornece ao usuário uma recomendação das taxas de nutrientes que precisam ser repostas ao solo. Uma abstração da descrição desse trabalho é mostrada na Figura 7.

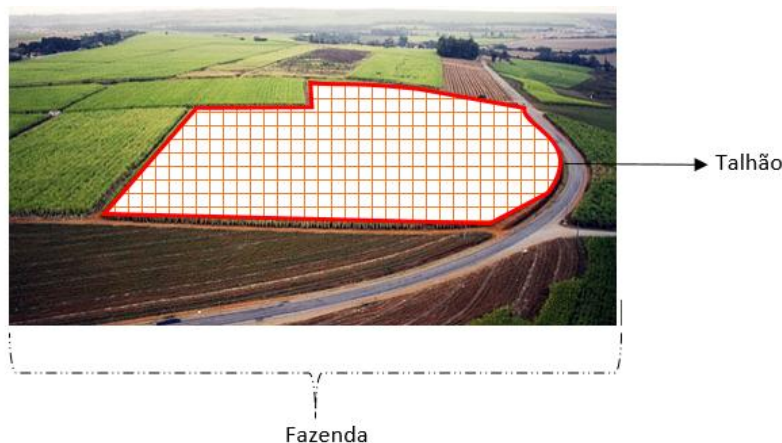


Figura 7 – Exemplo de Fazenda e Talhão

Fonte: Autor

O sistema precisa arquivar, recuperar e processar dados para análises futuras sem negligenciar os aspectos do modelo de entidade-relacionamento (Chen, 1976). O modelo de banco de dados original do sistema é apresentado na Figura 8, onde há muitos relacionamentos que descrevem os estágios para analisar os ciclos de uma cultura (Costa, Santana, Saraiva, & Molin, 2011).

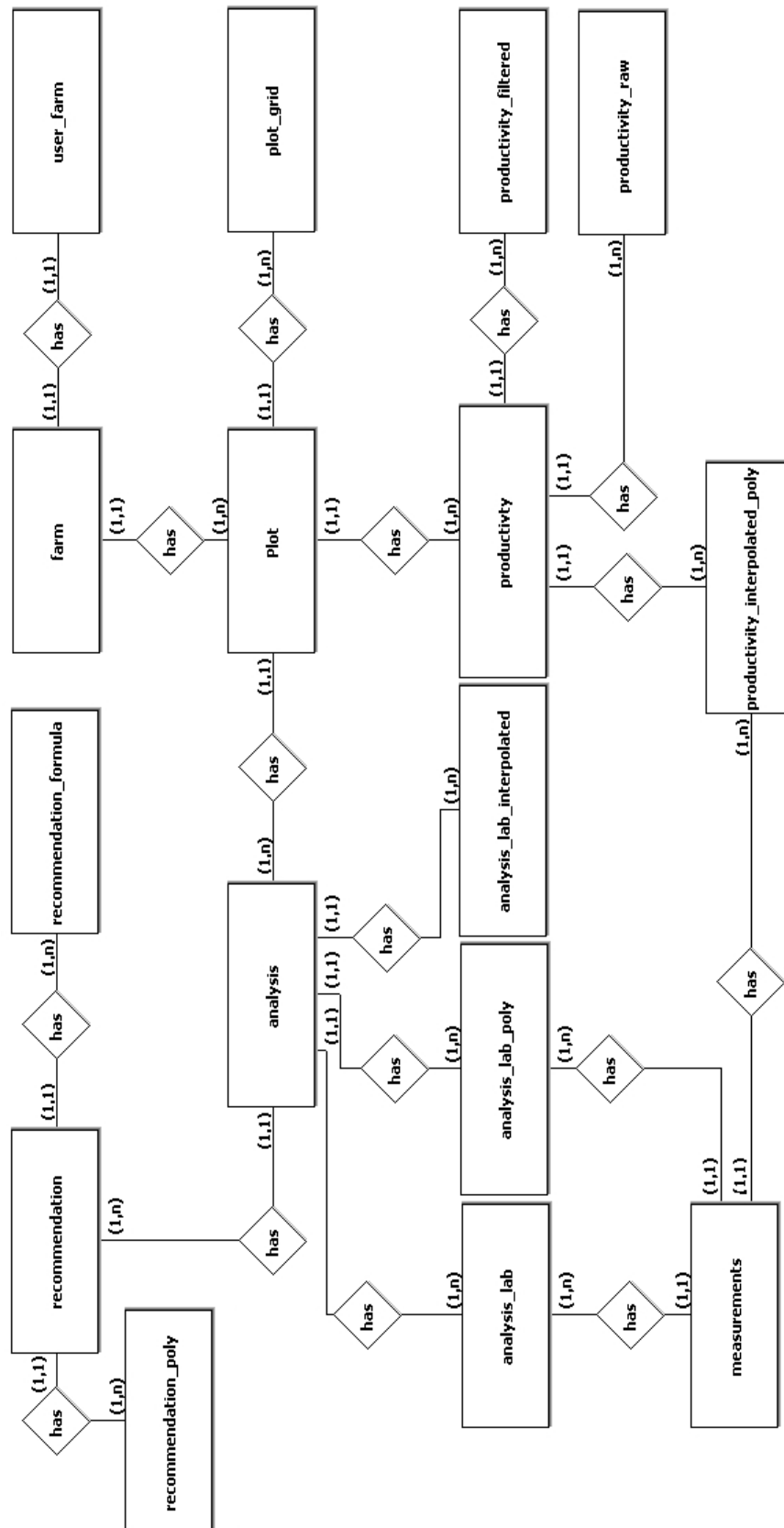


Figura 8 – Modelo de dados Original
 Fonte: Autor

Uma breve descrição das tabelas que fazem parte do modelo da Figura 8 é apresentada a seguir:

- a) **Plot:** um contorno gráfico representa uma parcela da fazenda que é usada como campo de cultura, também chamado de talhão. Esta entidade deve ter um polígono associado devido ao fato de que esta informação é usada para os métodos de geoprocessamento (por exemplo, a produtividade da máquina de filtragem e métodos de interpolação)
- b) **Farm:** representa o conjunto de parcelas utilizadas como campos de culturas agrícolas. A fazenda pode ser representada por uma geometria de múltiplos polígonos.
- c) **Productivity:** a produtividade representa a colheita de um campo de cultura. Esta entidade contém os dados que descrevem a colheita tais como data de início e fim da colheita. Os dados reais da produtividade estão representados nas entidades *Productivity_filtered* e *Productivity_raw*.
- d) **Productivity_raw:** esta entidade possui os dados obtidos da máquina colheitadeira sem qualquer tipo de limpeza de dados.
- e) **Plot_grid:** representação matricial de todos os pontos utilizados para interpolações de amostras química/física do solo ou produtividade.
- f) **Produtivity_interpolated_poly:** resultado do processamento do *raster (bitmap)* gerado na interpolação para representação vetorial, com o objetivo de acelerar a visualização nos servidores de mapas.
- g) **Produtivity_filtered:** pontos com informação de produtividade da colheitadeira resultantes do algoritmo de limpeza de dados de produtividade. São removidos pontos inválidos, com resultados nulos ou fora do talhão.
- h) **Analysis:** entidade que armazena os tipos de análises efetuadas nos talhões. As análises podem ser de atributos físicos/textura do solo, nutrientes químicos (Nitrogênio, Fósforo e Potássio), calagem ou outras medições.
- i) **Analysis_lab:** pontos amostrados no talhão para análise efetuada em laboratório.
- j) **Analysis_lab_interpolated:** pontos contendo o resultado da interpolação dos dados oriundos da tabela **Analysis_lab**.
- k) **Analysis_lab_poly:** resultado do processamento do *raster (bitmap)* gerado na interpolação para representação vetorial dos pontos amostrados no talhão, oriundos da tabela **Analysis_lab**.
- l) **Recommendation:** entidade de registro de criação de recomendação para aplicação de insumos.

- m) **Recommendation_poly**: representação em polígonos, representando a grade de pontos para a aplicação da recomendação gerada, utilizando a fórmula gravada na tabela **Recommendation_formula**.
- n) **Recommendation_formula**: fórmulas ou equações utilizadas para o cálculo da recomendação para aplicação de insumos e reposição de nutrientes.
- o) **Measurements**: unidades de medida utilizadas em agricultura.
- p) **User_farm**: armazena o cadastro de usuários do sistema (agricultores, técnicos agrícolas e pesquisadores).

A Figura 9 mostra as tabelas *farm*, *plot*, *productivity* e *productivity_raw* que foram selecionadas para demonstrar a modelagem BPMN deste artigo, pois armazenam os dados mais significativos do sistema e que são usados pelo sistema para gerar gráficos e relatórios para a análise dos usuários.

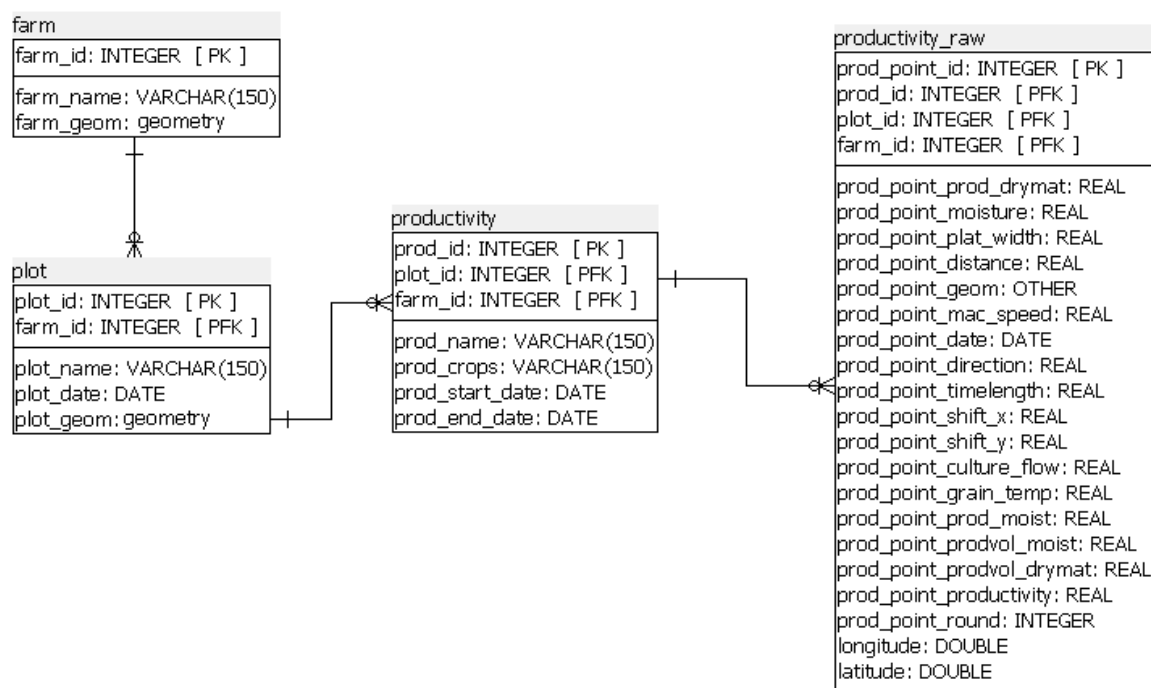


Figura 9 – Tabelas do modelo original, antes das refatorações.

Fonte: Autor

2.2. Processos para Refatorações de Banco de Dados

O trabalho inicial do administrador de banco de dados é coletar os requisitos de alterações dos usuários, identificar os problemas de modelagem ou que podem influenciar o desempenho da consulta. Esse conjunto de alterações no modelo de banco de dados será a entrada do processo de refatoração que levará o modelo do banco de dados do modelo original até o final. A Figura 10 mostra o contexto do processo de refatoração proposto.



Figura 10– Contexto do processo da refatoração.

Fonte: Autor

Na Figura 9 verifica-se o uso de chaves compostas como chave primária e há um forte acoplamento com o domínio de agricultura de precisão. Quanto mais simples a modelagem do banco, melhor o desempenho das consultas, assim a Tabela 1 apresenta as refatorações que foram modeladas em BPMN e executadas no processo de refatoração do banco de dados, para atribuir chave de identificação única (id) para todas as tabelas e refazer as referências de chaves-estrangeiras.

Tabela 1 – Conjunto de refatorações utilizadas no estudo de caso

	Refatoração	Descrição	Tipo	Impacto
1	Introduzir chave de identificação	Troca múltiplos atributos por uma chave primária	Estrutural	Melhora o modelo e o desempenho
2	Adicionar restrição de chave estrangeira	Adicionar uma restrição para garantir a integridade referencial	Integridade referencial	Melhora o modelo
3	Eliminar coluna	Excluir do modelo uma coluna	Estrutural	Melhora o modelo

Os principais passos para a realização do processo de refatoração do estudo de caso foram:

1. Identificar as entidades envolvidas nas refatorações:

a) *Introduzir chave de identificação*

Todas as tabelas do exemplo da Figura 10 não têm uma coluna **id** auto incrementada para ser utilizada como chave de identificação. A ordem das tabelas refatoradas foi: *farm*, *plot*, *productivity* e *productivity_raw*. A Figura 11 apresenta os passos da refatoração **Introduzir chave de identificação** modelados na notação BPMN.

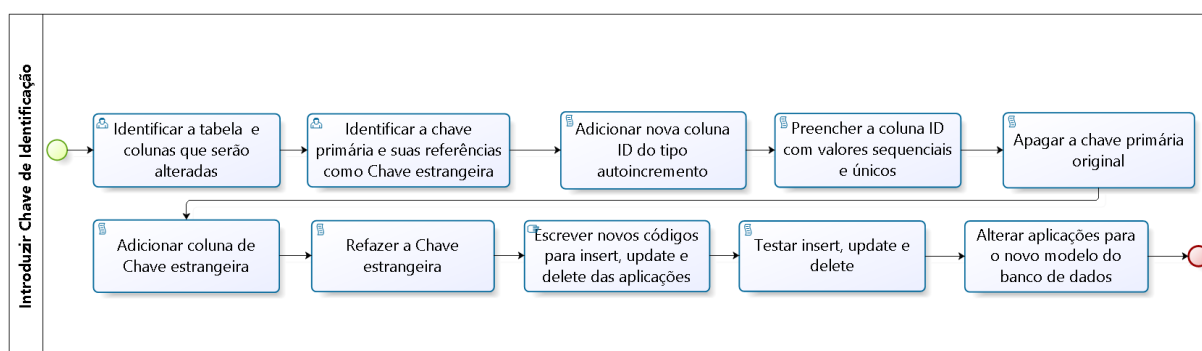


Figura 11 – Refatoração Introduzir chave de identificação.

Fonte: Autor

b) *Adicionar restrição de chave estrangeira*

Após a refatoração anterior, foi necessário adicionar restrições de chave estrangeira nas seguintes tabelas:

- i. *plot*, coluna *farm_id* é chave estrangeira para a tabela *farm*, coluna *id*
- ii. *productivity*, coluna *plot_id* é chave estrangeira para a tabela *plot*, coluna *id*
- iii. *productivity_raw*, coluna *prod_id* é chave estrangeira para a tabela *productivity*, coluna *id*

Neste caso a ordem afetará a refatoração. Assim as tabelas foram ordenadas da seguinte forma: *productivity_raw*, *productivity* e *plot*. A Figura 12 apresenta os passos da refatoração **Adicionar restrição de chave estrangeira** modelados na notação BPMN.

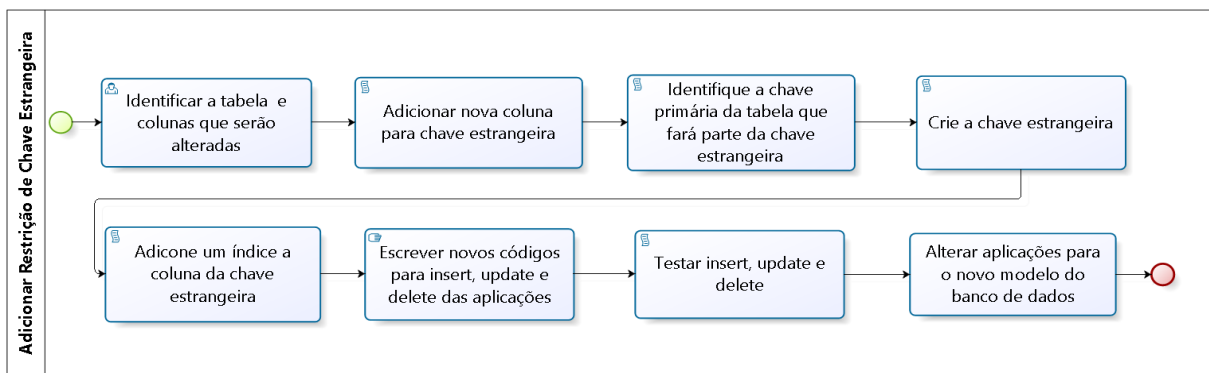


Figura 12 – Refatoração Adicionar restrição de chave estrangeira.

Fonte: Autor

c) *Eliminar coluna*

Após a refatoração anterior, foi necessário eliminar as seguintes colunas nas tabelas a seguir:

- iv. *farm*, eliminar a coluna *farm_id*
- v. *plot*, eliminar as colunas *plot_id*
- vi. *productivity*, eliminar as colunas *prod_id* e *farm_id*
- vii. *productivity_raw*, eliminar as colunas *plot_id*, *farm_id* e *prod_point_id*
- viii. *productivity_raw*

Neste caso a ordem afetará a refatoração. Assim as tabelas foram ordenadas da seguinte forma: *productivity_raw*, *productivity*, *plot* e *farm*.

- d) A Figura 13 apresenta os passos da refatoração ***Eliminar coluna*** modelados na notação BPMN.

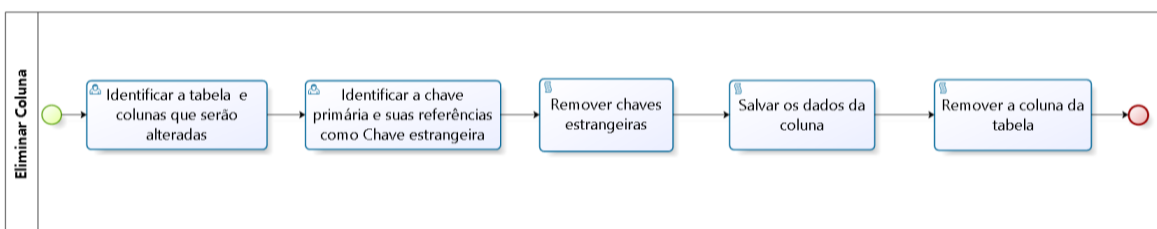


Figura 13 – Refatoração Eliminar coluna.

Fonte: Autor

Os processos apresentados nas Figuras 11, 12 e 13 podem ser reutilizados para serem aplicados em todo o modelo do banco de dados da Figura 8. Para isso foi necessário desenvolver um processo que possibilite agrupar as várias refatorações necessárias. Deste modo, os processos apresentados nas Figuras 11, 12 e 13, tornam-se sub-processos na Figura 14, na qual é apresentado o processo para grande alteração em um banco de dados. Os passos do processo da Figura 14 serão discutidos a seguir.

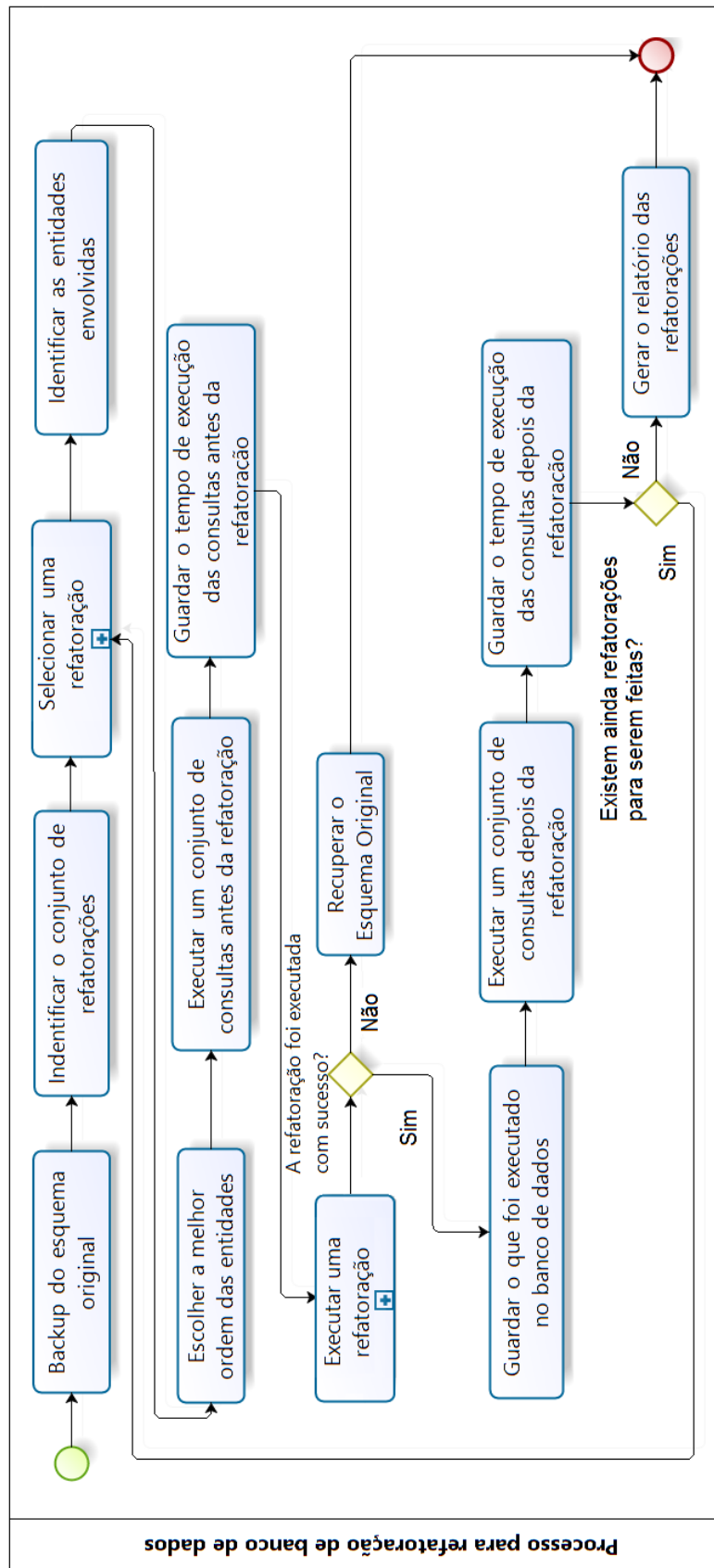


Figura 14– Processo para grande alteração em um banco de dados

Fonte: Autor

1. Backup do Esquema Original

Antes de se iniciar todas as refatorações, é necessário fazer um *backup* do esquema (modelo do banco de dados) que será alterado. Em qualquer ambiente, essa tarefa é necessária. Mesmo no ambiente de desenvolvimento, é importante que se faça o teste do *backup* para a recuperação do esquema alterado.

2. Backup do Esquema Original

A entrada do processo é um conjunto de requisitos do usuário e uma lista dos problemas identificados no ambiente de produção. Os problemas e os requisitos devem ser traduzidos para uma lista de refatorações. Esta tarefa deve ser feita pelo administrador do banco de dados que conhece o domínio do negócio e também toda sua estrutura. O mapeamento de um requisito ou problema para refatoração tem uma grande influência da experiência do administrador do banco de dados, mas o processo possibilita também um administrador que não tem muita experiência fazer uma refatoração e verificar se o resultado foi o esperado.

3. Backup do Esquema Original

O administrador deverá selecionar, do conjunto de refatorações, uma refatoração para começar. O administrador deve verificar somente se a refatoração escolhida tem pré-requisitos. Não é necessário, em um primeiro momento, escolher uma melhor ordem de execução das refatorações, pois uma melhor ordem vai depender muito do ambiente, do banco de dados e do tipo do problema ou requisito que está sendo resolvido. As refatorações serão executadas primeiro no ambiente de desenvolvimento, depois no ambiente de testes e por último no ambiente de produção. Conforme as refatorações sejam executadas, o administrador vai acumular experiência para reordenar as refatorações para gastar menos tempo.

4. Backup do Esquema Original

Muitas refatorações exigem alterações em várias entidades. Nestes casos, o administrador, conhecedor da aplicação, deve listar todas as entidades envolvidas na refatoração e para executar a próxima tarefa, também documentar as dependências entre elas.

5. Escolher a Melhor Ordem das Entidades

Neste momento é necessário ordenar as entidades que serão alteradas durante a refatoração. A melhor ordem é aquela que permite que todas as entidades sejam alteradas sem a perda de integridade, dados ou referências.

6. Executar um Conjunto de Consultas Antes da Refatoração

Para cada entidade envolvida na refatoração, é necessário escrever consultas que tragam os dados de suas tabelas e também das tabelas mães e filhas (aquelas que fazem referência e aquelas referenciadas). Uma possibilidade de automatizar essa

tarefa é coletar do *log* do gerenciador de banco de dados, por um período representativo, todas as consultas feitas ao banco de dados. Identificar as mais executadas e as mais complexas para que sejam utilizadas nesta tarefa.

7. Guardar o Tempo de Execução das Consultas Antes da Refatoração

Mesmo que a refatoração não tenha como objetivo melhorar o desempenho do banco de dados, é muito importante coletar e guardar o tempo de execução das consultas antes de fazer qualquer refatoração. O objetivo é medir o impacto no desempenho que uma refatoração pode provocar. Caso o objetivo não seja melhorar o desempenho, também é necessário ter o cuidado de não piorar o desempenho do banco de dados.

8. Executar uma Refatoração

Esta tarefa irá alterar o esquema do banco original em direção ao modelo final esperado. Cada refatoração irá exigir um conjunto de ações que é muito específico. Assim, o administrador de banco de dados deve ler com atenção as instruções existentes nos catálogos de refatorações, existentes na literatura de banco de dados, como o desenvolvido por Ambler e Sadalage (2006) e o catálogo reorganizado e em português desenvolvido por Domingues (2011). Em BPMN esta atividade é um subprocesso da Figura 15.

9. Recuperar o Esquema Original

Esta tarefa somente será executada se houver falha na execução de uma das refatorações. Para garantir a segurança e integridade dos dados, todo o processo será cancelado, não importando quantas refatorações já tenham sido feitas com sucesso.

10. Salvar o Que Foi Executado No Banco de Dados

Todos os *scripts* executados no banco de dados devem ser salvos para que seja possível executá-los em outros ambientes. Além deste objetivo, é importante que se salve todos os *scripts* para que seja possível realizar uma auditoria, caso ocorra algum problema.

11. Executar um Conjunto de Consultas Depois da Refatoração

Com a refatoração executada, repete-se a execução do conjunto de consultas ao banco de dados. A execução com sucesso desta tarefa, indica que a refatoração foi executada conforme o planejado. Pode ocorrer que alguma consulta específica não seja executada, devido às alterações realizadas, mas é importante obter e salvar os *logs* de erros. Os erros ocorridos na execução desta consulta são indicativos que a refatoração realmente foi realizada, provocando a interrupção do funcionamento de uma consulta. Um conjunto desses erros, pode ser usado para exemplificar, no final de todo o processo, o que os desenvolvedores não podem mais realizar nas suas aplicações.

12. Guardar o Tempo de Execução das Consultas Depois da Refatoração

Para aquelas consultas que foram executadas com sucesso, é necessário coletar o tempo de execução ou qualquer outra métrica (por exemplo: plano de execução) para que seja possível verificar o impacto no desempenho do banco de dados.

13. Gerar o Relatório das Refatorações

Esta tarefa do processo é executada quando se esgotaram todas as refatorações do conjunto definido na tarefa: “Identificar o conjunto de refatorações”. O objetivo é compilar todas as informações existentes no processo para que se tenha um relatório. As principais informações que devem fazer parte deste relatório são:

- Lista das refatorações que foram executadas e a ordem;
- *Scripts* que *fizeram* as alterações no banco de dados;
- *Scripts* que contêm as consultas realizadas antes e depois de cada refatoração;
- Tempos antes e depois da execução destas consultas;
- Lista das observações, comentários ou alertas dos problemas encontrados durante o processo.

Os relatórios permitem uma análise de impacto deste processo, documentar tudo o que foi feito e divulgar o novo modelo de banco de dados para todos envolvidos no processo.

3. CONCLUSÕES

A notação BPMN mostrou-se adequada para a modelagem de refatoração de banco de dados. Cada refatoração descritas por Ambler possui passos bem definidos, que são facilmente representadas pela notação BPMN. É possível modelar restrições, estabelecer ordem para as tarefas serem realizadas e documentar cada alteração realizada no banco de dados, assim é possível desfazer as alterações realizadas e retornar ao estado inicial, sem prejuízo aos dados armazenados. Os processos isolados das refatorações de Ambler podem ser reaproveitados, quando for preciso realizar uma grande alteração no banco de dados, na qual é necessário repetir algumas refatorações para que o processo seja completado.

Os processos apresentados nesse trabalho podem ser incorporados a sistemas de informação e serem executados de forma automática e programada pelo administrador do banco de dados (DBA). Essa automatização, além de agilizar o trabalho do DBA, ainda previne erros que afetam a segurança, integridade e consistência do banco de dados.

4. REFERÊNCIAS

- Aalst, W. M. P. van der. (2004). Business process management demystified: A tutorial on models, systems and standards for workflow management. *Lectures on Concurrency and Petri Nets*, 1–65. doi:10.1007/978-3-540-27755-2_1
- ABPMP. (2009). *CBOK - Business Process Management Common Body of Knowledge (BPM CBOK)*. (APBMP, Ed.). Chigago: ABPMP.
- Ambler, S. W., & Sadalage, P. J. (2006). *Refactoring databases: evolutionary database design*. New York: Addison-Wesley Professional.
- Chen, P. P. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1), 9–36.
- Costa, W. F., Santana, F. S., Saraiva, A. M., & Molin, J. P. (2011). Banco de dados geográficos para a construção de serviços web para agricultura de precisão. In *In: VIII Congresso Brasileiro de Agroinformática*. Bento Gonçalves, Rio Grande do Sul, Brasil.
- Decker, M., Che, H., Oberweis, A., Stürzel, P., & Vogel, M. (2010). Modeling Mobile Workflows with BPMN. In *In: 2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)* (pp. 272–279). IEEE. doi:10.1109/ICMB-GMR.2010.12
- Domingues, H. H. (2011). *Replicação Assíncrona de banco de dados evolutivos*. Instituto de Matemática e Estatística, Universidade de São Paulo.
- Fowler, M., Beck, K., Brant, J., Opdyke, W., & Roberts, D. (1999). *Refactoring: Improving the Design of Existing Code*. New York: Addison-Wesley Professional.
- Giaglis, G. M. (2001). A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques. *International Journal of Flexible Manufacturing Systems*, 13, 209–228. doi:10.1023/a:1011139719773
- Mili, H., Tremblay, G., Jaoude, G. B., Lefebvre, É., Elabed, L., & Boussaidi, G. El. (2010). Business process modeling languages. *ACM Computing Surveys*, 43(1), 1–56. doi:10.1145/1824795.1824799

- Phalp, K., & Shepperd, M. (2000). Quantitative analysis of static models of processes. *Journal of Systems and Software*, 52(2-3), 105–112. doi:10.1016/S0164-1212(99)00136-3
- Pressman, R. S. (2011). *Engenharia de Software - Uma Abordagem Profissional. Development* (7^a ed., p. 752). New York, NY, USA: Mc Graw Hill.
- Rademakers, T. (2012). *Activiti in Action: Executable business processes in BPMN 2.0* (p. 456). Shelter Island, NY: Manning Publications. Retrieved from <http://www.amazon.com/Activiti-Action-Executable-business-processes/dp/1617290122>
- Roddick, J. F. (1995). A survey of schema versioning issues for database systems. *Information and Software Technology*, 37(7), 383–393. doi:10.1016/0950-5849(95)91494-K
- Silva, A. C. L. (2013). BPMN 2.0 - Notação e Modelo de Processo de Negócio (Pôster). Retrieved March 18, 2014, from <http://www.itposter.net/itPosters/bpmn/bpmn.htm>
- Valle, R., & Oliveira, S. B. de. (2013). *Análise e Modelagem de Processos de Negócio: Foco da Notação BPMN (Business Process Modeling Notation)*. (E. A. S. A., Ed.) (1. ed.). São Paulo: ATLAS. Retrieved from <http://books.google.com.br/books?id=8hARQgAACAAJ>
- White, S. A., & Miers, D. (2008). *BPMN Modeling and Reference Guide*. Lighthouse Point, FL USA: Future Strategies Inc. Retrieved from <http://books.google.co.bw/books?id=0Z2Td3bCYW8C>