



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

UMA FERRAMENTA PARA GERENCIAMENTO DE
RESTRIÇÕES SEMÂNTICAS TEMPORAIS EM BANCO DE DADOS

FABIO DE NEGRI BAUMGRATZ

ORIENTADOR: PROF.DR. LUIZ CAMOLESI JUNIOR

PIRACICABA, SP

2010

**UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**UMA FERRAMENTA PARA GERENCIAMENTO DE
RESTRIÇÕES SEMÂNTICAS TEMPORAIS EM BANCO DE DADOS**

FABIO DE NEGRI BAUMGRATZ

ORIENTADOR: PROF.DR. LUIZ CAMOLESI JUNIOR

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA, SP

2010

**UMA FERRAMENTA PARA GERENCIAMENTO DE
RESTRIÇÕES SEMÂNTICAS TEMPORAIS EM BANCO DE DADOS**

Autor: Fabio De Negri Baumgratz

Orientador: Prof.Dr. Luiz Camolesi Junior

Dissertação de Mestrado defendida e aprovada em 19 de Fevereiro de 2010, pela Banca Examinadora constituída dos Professores:

Prof.Dr. Luiz Camolesi Junior

UNICAMP

Prof^a. Dr^a. Marina Tereza Pires Vieira

UNIMEP

Prof^a. Dr^a. Regina Lúcia de Oliveira Moraes

UNICAMP

À

Minha família, especialmente ao meu avô Luigi, que faz parte da minha vida e o mantenho vivo em minha memória. À minha mãe Ângela por todo o suporte e por ser uma pessoa que tenho como exemplo de vida. Ao meu pai Leonardo, ao meu irmão Rafael e à minha querida Adriana.

AGRADECIMENTOS

Ao professor Luiz Camolesi Junior por dividir comigo uma parte de seu conhecimento, pelo seu esforço e dedicação que mesmo se desligando da UNIMEP continuou a me orientar.

Ao professor Luiz Eduardo Galvão Martins que sempre esteve disposto a me ajudar, desde a minha graduação, sendo o meu orientador na faculdade de Ciência da Computação.

Aos professores Luis Henrique Sacchi e José Martins Júnior que, apesar de não terem participação direta no mestrado, foram professores excepcionais durante a minha graduação.

Às professoras Marina Teresa Pires Vieira e Ana Estela Antunes da Silva por serem sempre muito atenciosas comigo.

À Sra. Rosa Maria Alves, uma pessoa fundamental na conclusão deste mestrado, sempre me ajudando e elucidando as minhas dúvidas.

À todos os meus amigos que de forma direta ou indireta me incentivaram e me motivaram à conclusão do mestrado, especialmente ao colega de mestrado Leonardo de Oliveira Lombardi que compartilhou comigo diversas dificuldades, foi um grande amigo durante estes dois anos e espero que continue assim. Aos meus amigos de Águas de São Pedro que, apesar de não vê-los com frequência, acabavam recarregando as minhas energias nos finais de semana que passava na cidade, especialmente ao meu amigo de infância João Paulo Zampieri.

“Quem sabe concentrar-se numa coisa e insistir nela como único objetivo,
obtém, ao fim e ao cabo, a capacidade de fazer qualquer coisa.”

Mahatma Gandhi

(1869 - 1948)

SUMÁRIO

LISTA DE FIGURAS	IX
LISTA DE ABREVIATURAS E SIGLAS	XI
LISTA DE TABELAS E QUADROS	XII
1 INTRODUÇÃO	14
1.1 CONSIDERAÇÕES INICIAIS	14
1.2 OBJETIVO DO TRABALHO	15
1.3 MOTIVAÇÃO DO TRABALHO	15
1.4 METODOLOGIA.....	16
1.5 ESTRUTURA DA DISSERTAÇÃO	16
2 RESTRIÇÕES EM BANCO DE DADOS.....	18
2.1 CONSIDERAÇÕES INICIAIS.....	18
2.2 RESTRIÇÕES.....	18
2.2.1 RESTRIÇÕES DE BANCO DE DADOS	19
2.2.1.1 Exemplo de Aplicação de Restrição	22
2.2.2 CENÁRIO DAS RESTRIÇÕES	23
2.2.3 MODELANDO AS RESTRIÇÕES	27
2.3 TEMPO	29
2.4 RESTRIÇÕES NOS SGBDs.....	31
2.5 TRABALHOS CORRELATOS.....	34
2.6 CONSIDERAÇÕES FINAIS	36
3 MODELAGEM DA FERRAMENTA	37
3.1 CONSIDERAÇÕES INICIAIS.....	37
3.2 REQUISITOS.....	37
3.3 MODELAGEM	41
3.3.1 FUNÇÕES ESSENCIAIS	42
3.3.2 MANIPULAR RESTRIÇÕES	46
3.3.3 MANIPULAR VARIANTES E CENÁRIOS DE SOBREVIVÊNCIA	48
3.3.4 MANIPULAR VERSÕES.....	51
3.3.5 MANIPULAR PACOTES	53
3.4 PROJETO DO BANCO DE DADOS DA FERRAMENTA	56
3.5 PROJETO DA INTERFACE.....	64
3.6 CONSIDERAÇÕES FINAIS	68
4 GERENCIAMENTO DE CENÁRIOS E VARIANTES.....	70
4.1 CONSIDERAÇÕES INICIAIS.....	70
4.2 CRIAÇÃO DE CENÁRIOS DE SOBREVIVÊNCIA	70
4.2.1 CENÁRIO DE SOBREVIVÊNCIA NORMAL.....	72
4.2.2 CENÁRIO DE SOBREVIVÊNCIA SEM ABERTURA	72
4.2.3 CENÁRIO DE SOBREVIVÊNCIA SEM FECHAMENTO	73
4.2.4 CENÁRIO DE SOBREVIVÊNCIA GLOBAL	74
4.3 CRIAÇÃO DE VARIANTES.....	75
4.3.1 ASSOCIANDO O TEMPO À VARIANTE	77
4.3.2 ATIVAÇÃO E DESATIVAÇÃO DE UMA VARIANTE	78
4.4 CENÁRIOS DE SOBREVIVÊNCIA SOBREPOSTOS E CONFLITOS DE VARIANTES	80
4.5 CONSIDERAÇÕES FINAIS	88

5	CONCLUSÃO.....	90
5.1	CONSIDERAÇÕES INICIAIS	90
5.2	DIFICULDADES ENCONTRADAS	90
5.3	CONTRIBUIÇÕES	93
5.4	TRABALHOS FUTUROS	94
5.5	CONSIDERAÇÕES FINAIS	95

LISTA DE FIGURAS

FIGURA 2.1. MODELO DE CENÁRIOS PARA RESTRIÇÕES DE BANCO DE DADOS. (CAMOLESI JR., 2006).....	23
FIGURA 2.2. A) LS PERMANENTE E JS PERMANENTE. B) LS PERMANENTE E JSS TEMPORÁRIOS. (CAMOLESI JR., 2006)	25
FIGURA 2.3. UMA RESTRIÇÃO EM DIVERSOS CENÁRIOS DE APLICABILIDADE.	27
FIGURA 2.4. EXEMPLO DE MODELAGEM DO TEMPO.....	31
FIGURA 2.5. CONSULTA À VISÃO SYS.OBJECTS DO BANCO DE DADOS MSDB.....	33
FIGURA 2.6. CONSULTA AO OBJETO-PAI DA RESTRIÇÃO EM DESTAQUE NA FIGURA 2.5.....	34
FIGURA 2.7. CONSULTA À VISÃO DAS RESTRIÇÕES DO TIPO CHECK DO BANCO DE DADOS MSDB.....	34
FIGURA 3.1. INTERAÇÃO ENTRE O DBA (USUÁRIO) E AS RESTRIÇÕES DO BANCO.	38
FIGURA 3.2. CASO DE USO: VISÃO GERAL DA FERRAMENTA.....	41
FIGURA 3.3. CASO DE USO: FUNÇÕES ESSENCIAIS.	42
FIGURA 3.4. CASO DE USO: RESTRIÇÕES	46
FIGURA 3.5. CASO DE USO: VARIANTES E CENÁRIOS DE SOBREVIVÊNCIA.....	49
FIGURA 3.6. CASO DE USO: VERSÕES	52
FIGURA 3.7. CASO DE USO: PACOTES	54
FIGURA 3.8. DIAGRAMA DE CLASSES DO BANCO DE DADOS DA FERRAMENTA	57
FIGURA 3.9. INTERFACE GRÁFICA DO SISTEMA.....	64
FIGURA 3.10. ÁREA 1: BARRA DE FUNCIONALIDADES	65
FIGURA 3.11. ÁREA 2: ÁRVORE DE SERVIDORES E BANCO DE DADOS.....	66
FIGURA 3.12. ÁREA 3: ÁRVORE DE ELEMENTOS DO BANCO DE DADOS.....	67
FIGURA 3.13. ÁREA 4: JANELA DE EXTRAÇÃO DE BANCO DE DADOS	67
FIGURA 4.1. INTERFACE DA FERRAMENTA PARA CRIAR UM CENÁRIO DE SOBREVIVÊNCIA	71
FIGURA 4.2. CENÁRIO DE SOBREVIVÊNCIA NORMAL.....	72
FIGURA 4.3. CENÁRIO DE SOBREVIVÊNCIA SEM ABERTURA	73
FIGURA 4.4. CENÁRIO DE SOBREVIVÊNCIA SEM FECHAMENTO	74
FIGURA 4.5. INTERFACE DA FERRAMENTA PARA ADICIONAR NOVA VARIANTE.....	76

FIGURA 4.6. CRIAÇÃO DA VARIANTE CHAMADA DE MAIORIDADE	78
FIGURA 4.7. ATIVANDO UMA VARIANTE.....	79
FIGURA 4.8. DESATIVANDO UMA VARIANTE	79
FIGURA 4.9. EXEMPLO DE CENÁRIOS SOBREPOSTOS	81
FIGURA 4.10. CRIAÇÃO DOS CENÁRIOS DE SOBREVIVÊNCIA B E A	82
FIGURA 4.11. CRIAÇÃO DA VARIANTE C	82
FIGURA 4.12. CRIANDO A VARIANTE C2 COMO UMA VERSÃO DA VARIANTE C.....	83
FIGURA 4.13. CRIANDO A VARIANTE C1 COMO UMA VERSÃO DA VARIANTE C.....	83
FIGURA 4.14. ÁRVORE DE VERSÃO DAS VARIANTES.....	84
FIGURA 4.15. ASSOCIANDO A VARIANTE C1 AO CENÁRIO DE SOBREVIVÊNCIA B	85
FIGURA 4.16. TENTATIVA DE ASSOCIAÇÃO DAS VARIANTES C, C1 E C2 NO CENÁRIO A	86
FIGURA 4.17. ASSOCIAÇÃO DA VARIANTE C A UM CENÁRIO NÃO SOBREPOSTO AO CENÁRIO B.....	87
FIGURA 4.18. AVISO DE ABERTURA DE CENÁRIO NA BARRA DE TAREFAS	87
FIGURA 5.1. EXEMPLO DE ESTRUTURA EM ÁRVORE DE VERSÕES DE UMA RESTRIÇÃO	91
FIGURA 5.2. TIPOS DE SOBREPOSIÇÃO DE CENÁRIOS	92

LISTA DE ABREVIATURAS E SIGLAS

<i>BD</i>	Banco de Dados
<i>SGBD</i>	Sistema de Gerenciamento de Banco de Dados.
<i>DBA</i>	Administrador de Banco de Dados (Database Administrator)

LISTA DE TABELAS E QUADROS

Tabela 2.1. Exemplo de visões do sistema do Microsoft SQL Server 2005	33
Quadro 3.4.1 - Caso de uso : Registrar Servidor	44
Quadro 3.4.2 - Caso de uso : Excluir Servidor	45
Quadro 3.4.3 - Caso de uso : Excluir Banco de Dados	45
Quadro 3.4.4 – Caso de uso : Extrair Banco de Dados.....	45
Quadro 3.4.5 – Caso de uso : Atualizar Restrições.....	46
Quadro 3.4.6 - Caso de uso : Executar mudanças.....	46
Quadro 3.4.7 - Caso de uso : Criar restrição.....	48
Quadro 3.4.8 – Caso de uso : Alterar restrição	48
Quadro 3.4.9 - Caso de uso : Desativar restrição	49
Quadro 3.4.10 - Caso de uso : Ativar restrição	49
Quadro 3.4.11 - Caso de uso : Criar variante.....	51
Quadro 3.4.12 - Caso de uso : Criar cenário de sobrevivência.....	51
Quadro 3.4.13 - Caso de uso : Excluir cenário de sobrevivência.....	52
Quadro 3.4.14 - Caso de uso : Associar variante ao cenário de sobrevivência	52
Quadro 3.4.15 - Caso de uso : Criar versão de restrição	53
Quadro 3.4.16 - Caso de uso : Criar versão de variante	54
Quadro 3.4.17 - Caso de uso : Criar versão de pacote	54
Quadro 3.4.18 - Caso de Uso : Criar Pacote.....	56
Quadro 3.4.19 - Caso de Uso : Alterar Pacote.....	56
Quadro 3.4.20 - Caso de Uso : Excluir Pacote.....	56
Quadro 3.4.21 - Caso de Uso : Ativar Pacote	57
Quadro 3.4.22 - Caso de Uso – Desativar Pacote	57
Quadro 3.4.23 - Dicionário da Classe Servers	59
Quadro 3.4.24 - Dicionário da Classe Databases	60
Quadro 3.4.25 - Dicionário da Classe Schemas.....	60
Quadro 3.4.26 - Dicionário da Classe Objects	60
Quadro 3.4.27 - Dicionário da Classe ObjectsHistory	61
Quadro 3.4.28 - Dicionário da Classe Tables.....	61
Quadro 3.4.29- Dicionário da Classe Columns	61

Quadro 3.4.30 - Dicionário da Classe ColumnTypes	61
Quadro 3.4.31 - Dicionário da Classe Constraints	62
Quadro 3.4.32 - Dicionário da Classe ForeignKeys	62
Quadro 3.4.33 - Dicionário da Classe ValueConstraints	62
Quadro 3.4.34 - Dicionário da Classe FKReferences.....	62
Quadro 3.4.35 - Dicionário da Classe ColumnsConstraints	63
Quadro 3.4.36 - Dicionário da Classe Packages.....	63
Quadro 3.4.37 - Dicionário da Classe PackagesInfo.....	63
Quadro 3.4.38 - Dicionário da Classe LifeScenario.....	64
Quadro 3.4.39 - Dicionário da Classe Variants	64
Quadro 3.4.40 - Dicionário da Classe PackagesHistory.....	65

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

Os seres humanos civilizados estão acostumados a lidar com regras, pois crescem rodeados delas e também por suas versões mais rígidas, as leis. As restrições de bancos de dados podem ser comparadas às leis para as pessoas que vivem em sociedade. Quando elas são criadas e estão ativas, jamais serão transgredidas pelo banco de dados. Este trabalho trabalha principalmente com restrições semânticas. A restrição semântica é um tipo de restrição que envolve uma comparação, por exemplo, a restrição: “preço_mínimo>100” indica que o preço mínimo de algum produto cadastrado no BD deve ser maior que 100. Uma restrição semântica é uma restrição, porém uma restrição nem sempre é uma restrição semântica. Por isso, quando é citada a palavra “restrição” neste trabalho, dependendo do contexto, pode ser uma referência à restrição semântica.

Este trabalho envolve o gerenciamento das restrições dos bancos de dados e também propõe uma nova maneira de lidar com elas associando informações temporais. Isso permite, dentre outras vantagens que serão discutidas mais a frente, um gerenciamento melhor, mais preciso e automatizado, pois as restrições possuirão um escopo de vida e somente serão utilizadas quando forem necessárias. Isso tira uma preocupação extra dos DBAs, que é ativar e desativar restrições nos momentos certos, fomentando um uso maior das restrições dos bancos de dados, que atualmente não é utilizado tão frequentemente. Isto ocorre porque quando a tarefa de ativar ou desativar uma restrição se fizer necessária, ela será feita à mão no sistema de gerenciamento de banco de dados (SGBD) que está sendo utilizado, escrevendo o código para criar ou apagar uma restrição, o que torna a tarefa cansativa e também perigosa, pois é humanamente possível apagar uma restrição por engano e os SGBDs, normalmente, não armazenam um histórico das restrições.

Este trabalho envolve a criação de cenários temporais que serão responsáveis por fornecer um escopo de vida às restrições, tornando a ferramenta capaz de saber exatamente quando uma restrição deve ser ativada ou desativada. Espera-se que a ferramenta desenvolvida seja apenas o início, para proporcionar aos DBAs um controle das restrições cada vez melhor e mais eficiente.

1.2 OBJETIVO DO TRABALHO

O projeto tem como objetivo principal o desenvolvimento de uma ferramenta para o gerenciamento de restrições semânticas temporais em banco de dados. Esta ferramenta permite extrair todas as restrições semânticas de um banco de dados qualquer e, a partir desta extração, disponibilizar ao administrador do banco de dados (DBA), o controle das restrições para que ele possa manipulá-las e fazer todas as mudanças desejadas, inclusive a criação de novas restrições. As restrições criadas ou modificadas podem ser inseridas no banco de dados do qual foram extraídas.

A ferramenta, trabalhando em conjunto com os SGBDs, é um facilitador entre a comunicação do administrador do banco e o próprio banco de dados.

1.3 MOTIVAÇÃO DO TRABALHO

Nos dias atuais, a quantidade de bancos de dados existentes é gigantesca e pode variar desde um pequeno banco de dados de uma loja a um banco de dados de uma multinacional ou de uma instituição governamental. Independentemente da dimensão, uma característica comum entre esses bancos de dados é a dificuldade encontrada para manipular regras semânticas que estão contidas nesses bancos de dados e podem ser tanto uma regra simples de uma chave primária ou uma mais complexa como *triggers*. Como até o momento não se tem o conhecimento de alguma ferramenta com estas características, a criação

desta ferramenta e o fato dela trabalhar em uma área pouco explorada por pesquisas (cenários temporais para gerenciamento de restrições), abre um leque extenso de funcionalidades que podem ser implementadas no futuro. Essas foram as principais motivações para finalizar este projeto.

1.4 METODOLOGIA

Para atingir o objetivo deste trabalho, foi realizado o estudo sobre restrições em banco de dados, sobre os conceitos de intervalos temporais proposto por Allen (ALLEN, 1983) e obviamente as pesquisas relacionadas aos cenários temporais e restrições feitas por Camolesi (CAMOLESI JR., 2004) (CAMOLESI JR., 2006).

Existem diversas pesquisas relacionadas às restrições de banco de dados (CURRIM and RAM, 2008; DOUCET and MONTIES, 1997; FLESCA, GRECO, 2004), outras tantas relacionadas ao tempo (ALLEN, 1983; MOK, LEE, 2002), estudos para criar uma álgebra relacional temporal (DEY, BARRON, 1996), além de estudos relacionados a cenários temporais (FONTAINE and RAMAUX, 1998; HERTZUM, 2003), porém por se tratar de um ramo pouco explorado, o das restrições temporais em banco de dados, foi difícil encontrar artigos que se assemelhassem ao propósito desta dissertação. A base desta dissertação foram as pesquisas de Camolesi (CAMOLESI JR., 2004; CAMOLESI JR., 2006). Os artigos encontrados, mencionados anteriormente dentre outros, ajudaram a clarear algumas dúvidas para um melhor preparo para trabalhar com restrições em banco de dados e com a manipulação do tempo.

1.5 ESTRUTURA DA DISSERTAÇÃO

A dissertação está dividida em cinco capítulos. No primeiro, chamado introdução, está descrito o objetivo do trabalho, as razões que motivaram

este trabalho, a metodologia utilizada e esta subseção que descreve como está dividida a dissertação.

O segundo capítulo expõe os tópicos essenciais que são os pilares desta pesquisa. Nele estará contido informações sobre restrições de tal forma que uma pessoa com pouco conhecimento as compreenda e entenda qual o trabalho e a importância delas nos bancos de dados, bem como são fornecidas informações mais aprofundadas para as pessoas com mais conhecimento. Será discutido o tempo, isto é, como será abordada a questão temporal das restrições no banco de dados, a importância de dimensionar o tempo, os esforços de pesquisadores há décadas sobre este assunto e como é o padrão de tempo aceito atualmente pela ISO 8601(ISO, 2000). Na subseção 2.3 é descrito como as restrições se comportam nos SGBDs, informação esta necessária para tornar possível a comunicação entre o banco de dados alvo e a ferramenta que foi desenvolvida por este trabalho, tornando assim possível a extração e o armazenamento das restrições. Para efeito de ilustração, o SGBD exemplificado será o Microsoft SQL Server. É também neste capítulo que os trabalhos semelhantes a este serão comentados.

No terceiro capítulo é descrita a modelagem da ferramenta, iniciando com a proposta da ferramenta, em seguida é apresentada a estrutura do projeto e depois é detalhado como a ferramenta foi modelada, nesta subseção são expostos diversos diagramas de classes e alguns quadros fornecendo detalhes dos mesmos. Esse capítulo é finalizado com a apresentação da estrutura do banco de dados da ferramenta e com a demonstração de sua interface.

No quarto capítulo é descrito como é feita a criação e a manipulação dos cenários de sobrevivência e das restrições variantes, que é a funcionalidade inovadora desta ferramenta, já que, até então, as restrições em banco de dados eram consideradas simplesmente como invariantes.

No quinto e último capítulo está a conclusão da dissertação, com a discussão sobre as dificuldades encontradas para atingir o objetivo da proposta, as contribuições que esta ferramenta irá trazer para a área e uma breve descrição dos trabalhos futuros que esta ferramenta poderá implementar.

2 RESTRIÇÕES EM BANCO DE DADOS

2.1 CONSIDERAÇÕES INICIAIS

O objetivo maior deste trabalho é tratar as restrições temporais, também conhecidas por restrições variantes, isto é, restrições que variam conforme se passa o tempo ou que sua existência depende do tempo.

Neste capítulo são apresentados os conceitos de restrições, restrições temporais, tempo, cenários e como as restrições estão armazenadas nos SGBDs. Estes são os requisitos necessários para facilitar a compreensão deste trabalho.

2.2 RESTRIÇÕES

Restrição (*constraint*) é um elemento do banco de dados que estabelece limites a serem respeitados por usuários do banco. As restrições podem ser consideradas um sinônimo de regras, mais especificamente, regras de integridade.

As restrições dos bancos de dados jamais são transgredidas, deste modo, enquanto uma regra existir em um banco de dados, ela nunca será violada. Por exemplo, se por alguma razão uma empresa criou a restrição que clientes só poderão ser cadastrados no banco de dados sendo maiores de idade, enquanto esta restrição existir no banco de dados será impossível cadastrar alguém com idade inferior a 18 anos.

2.2.1 RESTRIÇÕES DE BANCO DE DADOS

As restrições são elementos importantíssimos em qualquer banco de dados, pois tornam possível o controle de ações ou de dados que serão inseridos, removidos ou atualizados para que não haja discrepância na informação, isto é, mantendo-os íntegros e com qualidade. Segundo Cordeiro (CORDEIRO, SANTOS, 2004) existem sete tipos de integridades relacionadas às informações contidas nos bancos de dados:

- **Integridade de Banco de Dados e Tabela:** é a forma mais geral de controle sobre integridade. Refere-se à relação entre entidades, colunas múltiplas em tabelas e múltiplas tabelas em um banco de dados;
- **Integridade Referencial:** é estabelecida na forma de campos (atributos) que fazem referência a outros campos ou entidades;
- **Integridade de Entidade:** define as regras para as colunas e linhas de uma tabela e tabelas como unidades. A unicidade é uma das regras primárias para assegurar a integridade da entidade;
- **Integridade de Domínio:** define um conjunto de valores que pode ser assumido por determinado campo;
- **Integridade Transacional:** define que um conjunto de operações sobre dados seja completamente efetivado ou as alterações são canceladas completamente e os valores alterados assumem os mesmos valores iniciais;
- **Integridade de Tipo:** define o formato do valor que um campo pode receber (moeda, decimal, inteiro etc.) ou se aquele campo pode ou não assumir um valor nulo;
- **Integridade de Estado:** define se o valor alterado de um campo está assumindo um valor válido.

A qualidade dos bancos de dados pode ser mensurada de diversas maneiras, entre elas, a política de segurança (FERNANDEZ-MEDINA, PIATTINI, 2001) e a manutenibilidade (DÍAZ and PIATTINI, 1999). Para se preservar a

qualidade das informações nesses bancos de dados uma modelagem correta e adequada dos recursos estratégicos é necessária (BUNEMAN, DAVIDSON, 1998). Os SGBDs controlam uma quantidade muito grande de informações com uma probabilidade muito alta de mudanças decorrentes de evoluções, involuções ou revoluções nos requisitos do mundo real, portanto, as restrições de banco de dados podem ser consideradas recursos estratégicos (CAMOLESI JR., 2006).

Por definição (RIORDAN, 1999; SILBERSHATZ, KORTZ, 1999), existem sete tipos de restrições:

- **Chaves primárias (*primary keys*):** Uma coluna ou um grupo de colunas que identificam unicamente um registro (linha) de uma tabela.
- **Chaves estrangeiras (*foreign keys*):** Consiste em uma coluna, ou um grupo de colunas, que participam de um relacionamento com uma chave primária de uma tabela.
- **Valores padrão (*defaults*):** Um valor padrão é um valor pré-definido pelo projetista do banco de dados, que será atribuído automaticamente ao atributo de uma tupla caso nenhum valor seja informado explicitamente pelo sistema ou pelo usuário.
- **Não Nulo (*NOT NULL*):** Certifica-se de que um atributo de uma tabela sempre tenha um valor, não podendo ser nulo.
- **Checagens (*checks*):** Limita a quantidade de valores que o sistema ou usuário pode atribuir para um atributo (coluna).
- **Regras (*rules*):** Executa a mesma função que a restrição de checagem, porém enquanto a restrição de checagem só pode ser aplicada na tabela à qual as regras foram atribuídas, as regras podem ser aplicadas a diversas tabelas.
- **Únicos (*uniques*):** Certifica-se que um registro de um atributo é único dentro de uma tabela, não permitindo assim valores repetidos.

Estes tipos podem ser classificados basicamente de duas formas: restrição estrutural (chave primária, por exemplo) ou restrição semântica. A restrição semântica de banco de dados tradicionalmente pode ser considerada um conjunto de predicados no formato: $P_1 \square P_2 \square \dots \square P_k$ sendo que cada predicado ($P_1, P_2, P_3,$

etc) tem o formato $C_1 \theta C_2$, aonde C_1 é um atributo, θ é um comparador e C_2 pode ser um atributo ou uma constante (CAMOLESI JR., 2006).

As restrições de banco de dados possuem algumas características usadas em qualquer sistema de controle/gestão de consistência (OAKASHA, CONRAD, 1999):

- **Identificador:** para manipulação e consulta cada regra precisa ter um nome único que a identifique;
- **Status:** propriedade temporária para representar o estado das restrições;
- **Escopo:** Conjunto de dados (classe) associados às restrições às quais se desejam manter a integridade e são dependentes do modelo de dados adotado;
- **Variabilidade:** Propriedade que estabelece o nível de satisfação de uma restrição para todos os objetos do escopo em todo e qualquer momento;
- **Inconstância:** Propriedade que estabelece o nível de modificações de uma restrição.

A característica *status* possui dois valores: ativo e inativo que, respectivamente, indicam se a restrição está sendo usada para o controle de integridade do banco de dados ou se a restrição não está em uso. Somente alguns usuários com privilégios podem mudar esse valor. A característica variabilidade, no escopo de restrições de integridade, pode ser referenciada como invariante porque em alguns casos não há outra opção a ser seguida pelo sistema. Para ilustrar pode-se lembrar o exemplo citado na subseção 2.1, em que todos os clientes precisam ser maiores de idade para serem cadastrados no banco de dados.

Os modelos de representação atuais usados para as restrições de banco de dados são pobres e inflexíveis para representar as regras encontradas no mundo real demandando, portanto, um grande esforço dos grupos de engenharia de software para sua manutenção (CAMOLESI JR., 2006). Currim & Ram (CURRIM and RAM, 2008) exemplificam isso quando apresentam um problema e expõem que algumas dessas regras não são tipicamente modeladas durante o projeto conceitual do banco de dados, os projetistas, na melhor das hipóteses, as documentam em

texto livre, mas muitas vezes acabam omitindo-as e esperam que elas sejam eventualmente implementadas em PL/SQL/Java/etc.

As restrições de dados podem e devem ser consideradas em todos os sistemas como requisitos dinâmicos, portanto demandando todos os cuidados associados (CAMOLESI JR., 2006). Além disso, surgem vários benefícios de se modelar as restrições semânticas durante o projeto conceitual, incluindo: custo reduzido para a correção de erros, execução consistente da aplicação, melhor capacidade para validar requisitos com o usuário e visibilidade das regras (CURRIM and RAM, 2008).

2.2.1.1 EXEMPLO DE APLICAÇÃO DE RESTRIÇÃO

O exemplo apresentado a seguir empregou conceitos de mineração de dados para solucionar um problema que existe no mundo real. Para tal, foi necessário usar uma base de dados. Não se pode afirmar que as informações contidas nesse banco de dados são reais, portanto, as regras apresentadas a seguir, por serem resultados de uma base de dados que pode ser fictícia, servem apenas para ilustrar o problema e não podem ser consideradas como uma representação fiel do mundo real.

Após a popularização da mineração de dados, alguns bancos americanos descobriram que existe uma maneira de conceder empréstimos a pessoas físicas com a menor chance possível de terem prejuízo. Diversas regras foram descobertas, entre elas, a que conceder crédito a uma pessoa com menos de 27 anos, com um salário menor que U\$ 4.210 e que seja solteira, pode ser arriscado demais, pois estatisticamente este é um perfil com alto índice de calote. Por outro lado, o risco é estatisticamente baixo ao conceder crédito a uma pessoa com idade superior a 27 anos, com saldo em conta corrente maior que U\$ 394 e salário maior que U\$ 2.623.

Com estas regras em mãos, o banco pode criar restrições de integridade e inserí-las no banco de dados para que a aprovação do empréstimo, em alguns casos, possa ser imediatamente concedida caso a restrição gerada pelo

segundo perfil seja satisfeita, ou não concedida caso a restrição gerada para o primeiro perfil seja satisfeita.

2.2.2 CENÁRIO DAS RESTRIÇÕES

Com o emprego de cenários para ajudar a modelagem do tempo das restrições, o controle de integridade dos dados dependerá de pré-condições que deverão ser usadas para a verificação do grupo de restrições que está ativo no momento em que uma atualização no banco de dados é feita. É com este propósito que existem duas classes de cenários de restrição (CAMOLESI JR., 2004):

- **Cenário de sobrevivência (*life scenario - LS*):** uma classe de cenários que estabelecem quando uma restrição *viva* pode ser *ativada* no sistema. Pode-se considerar que os LSs estão contidos em um escopo total e sem limite chamado *Cenário Global*.
- **Cenário de aplicabilidade (*job scenario - JS*):** Uma classe de intra-cenários, contidos no LS, com restrições ativadas, que podem ser usadas para verificação de integridade nos momentos chamados Pontos de Ação (*Action Points - APo*).

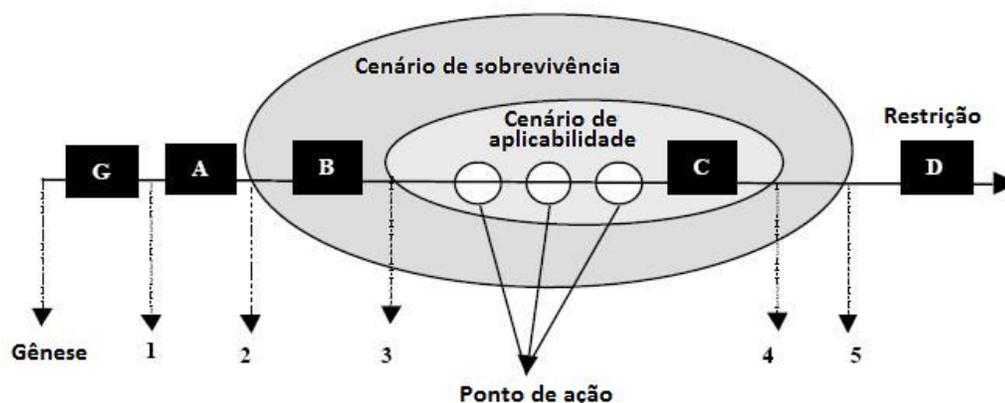


FIGURA 2.1. MODELO DE CENÁRIOS PARA RESTRIÇÕES DE BANCO DE DADOS. (CAMOLESI JR., 2006)

Imaginando uma linha do tempo na Figura 2.1, os Pontos de Ação são os momentos em que uma atualização no banco de dados (*insert, delete, update*) requer uma verificação de restrição, porém somente as restrições ativas devem ser verificadas. Como a quantidade de restrições a serem verificadas é reduzida, além de se manter a integridade, o banco ganha em desempenho.

Mudanças no mundo real precisam ser refletidas no banco de dados e, conseqüentemente, em suas restrições. Uma mudança no mundo real faz com que alguns cenários sejam fechados e outros abertos, portanto mudando o estado das restrições.

Na Figura 2.1, seguindo a linha do tempo observamos a ocorrência G (Gênese) que representa a fase de descoberta de restrições durante a engenharia de requisitos. Os outros pontos estão relacionados com as mudanças do mundo real que são: (1) desenvolvimento das restrições do banco está finalizado; (2) a abertura de um LS; (3) a abertura de um JS; (4) o fechamento do JS; (5) o fechamento do LS.

Sabendo da variabilidade das restrições de banco de dados, faz-se necessário conhecer o escopo das mesmas, assim possibilitando saber quando uma restrição está desativada ou quando está ativada e será usada pelo sistema para o controle de integridade.

Na análise baseada em cenários, as restrições possuem quatro estados durante o ciclo de vida, demonstrados por caixas-pretas na Figura 2.1 (CAMOLESI JR., 2006):

- **G. (desenvolvimento):** Restrições neste estado estão sendo projetadas (ou re-projetadas) e testadas. Somente o usuário pode remover a restrição deste estado;
- **A. (prontidão):** Restrições neste estado estão prontas para serem usadas. Somente o usuário pode remover a restrição deste estado;
- **B. (viva):** Restrições neste estado estão disponíveis para serem ativadas pelo usuário ou pelo SGBD;
- **C. (ativa):** Restrições neste estado podem ser usadas pelo sistema de controle de integridade para atender o SGBD;

- **D. (inativa):** Restrições neste estado não podem ser ativadas automaticamente. Somente o usuário pode “ressuscitar” a restrição criando um novo LS com um novo JS.

Os LSs permanentes possuem restrições sempre vivas enquanto os JSs permanentes possuem restrições sempre ativas. Os LSs temporários e os JSs temporários possuem restrições que serão desativadas quando o seu respectivo cenário fechar.

Uma ressalva interessante, ainda analisando a Figura 2.1, é que as ocorrências (2) e (3) podem aparecer simultaneamente resultando na criação, no mesmo instante, dos cenários LS e JS. Com isso a restrição, quando assumir o estado vivo (B), irá passar praticamente no mesmo instante para o estado ativo (C). Logicamente as ocorrências (4) e (5) poderão acontecer simultaneamente.

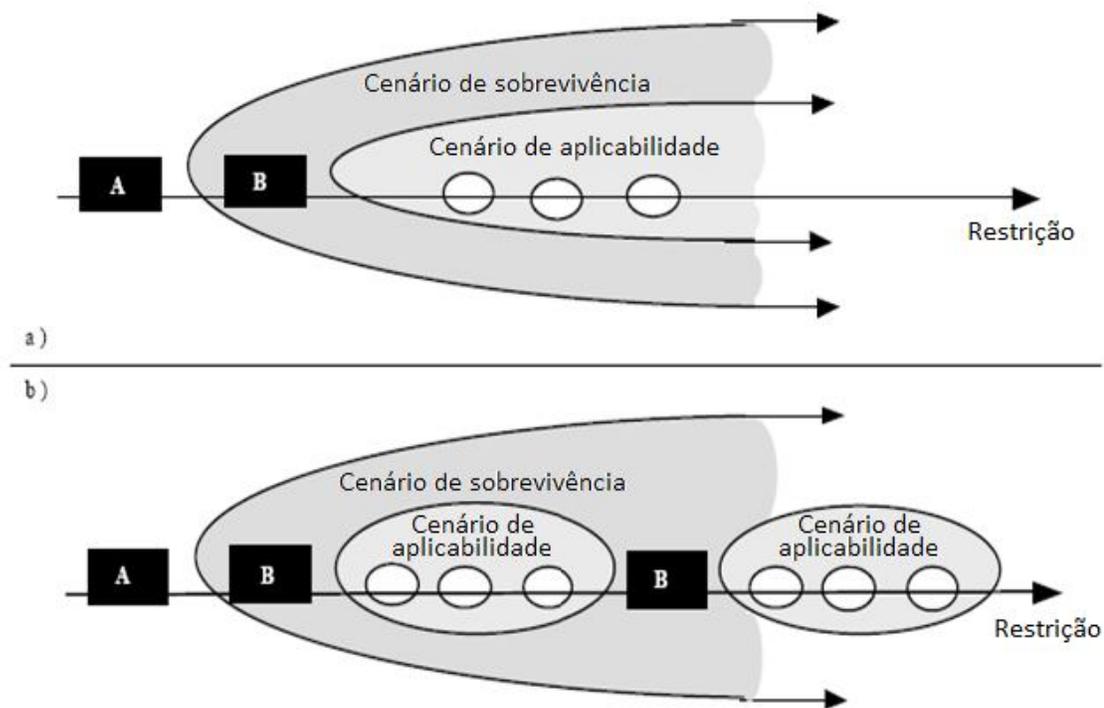


FIGURA 2.2. A) LS PERMANENTE E JS PERMANENTE. B) LS PERMANENTE E JSS TEMPORÁRIOS. (CAMOLESI JR., 2006)

As relações entre LS e JS podem prover alternativas importantes para o controle de integridade (CAMOLESI JR., 2006):

- **LS temporário e JS temporário (Figura 2.1):** adequado para situações de controle de integridade que demandam uma manutenção periódica e manual das restrições;
- **LS permanente e JS permanente (Figura 2.2a):** adequado para situações onde as condições de integridade são estáveis (somente condições invariantes);
- **LS permanente e JS temporário (Figura 2.2b):** adequado para situações onde as mudanças nas condições de integridade envolvem mudanças nos estados das restrições, que podem ser automáticas ou semi-automáticas.

Os cenários podem ser modelados com predicados contextuais. Após diversas análises em sistemas de informação (CAMOLESI JR., 2006), pode-se notar a existência de três contextos (CAMOLESI JR., 2004):

- **Conjunto de predicado de integridade (IP Set):** é composto para sentenças de condições clássicas $C_1 \theta C_2$, onde C_1 ou C_2 são atributos de objetos. Estes conjuntos de predicados estabelecem os Pontos de ação (APo) no ciclo de vida da restrição;
- **Conjunto de predicado de sobrevivência (SP Set):** é composto para sentenças de condições $Now \theta C_2$, envolvendo uma verificação de tempo ou intervalo definido em C_2 . Estes conjuntos de predicados não envolvem atributos dos objetos. *Now* é uma palavra opcional para representar o tempo real do sistema. Estes conjuntos de predicados estabelecem o cenário de sobrevivência (LS) para as restrições;
- **Conjunto de predicado de aplicabilidade (AP Set):** é composto para sentenças de condições $Now \theta C_2$, envolvendo uma verificação de tempo ou intervalo definido em C_2 . Estes conjuntos de predicados não envolvem atributos dos objetos. *Now* é uma palavra opcional para representar o tempo real do sistema. Estes conjuntos de predicados estabelecem o cenário de aplicabilidade (JS) para as restrições.

Por definição, cada restrição variante pode possuir vários conjuntos de predicados de aplicabilidade, porém apenas um conjunto de predicados de sobrevivência e um conjunto de predicados de integridade. Essa definição, como mostra a Figura 2.3, significa que uma restrição (caixa preta com a letra “C”), poderá estar ativada em diversos momentos e em diversos cenários de trabalho, mas sempre estará dentro do mesmo cenário de vida e sempre com a mesma tarefa a ser executada.

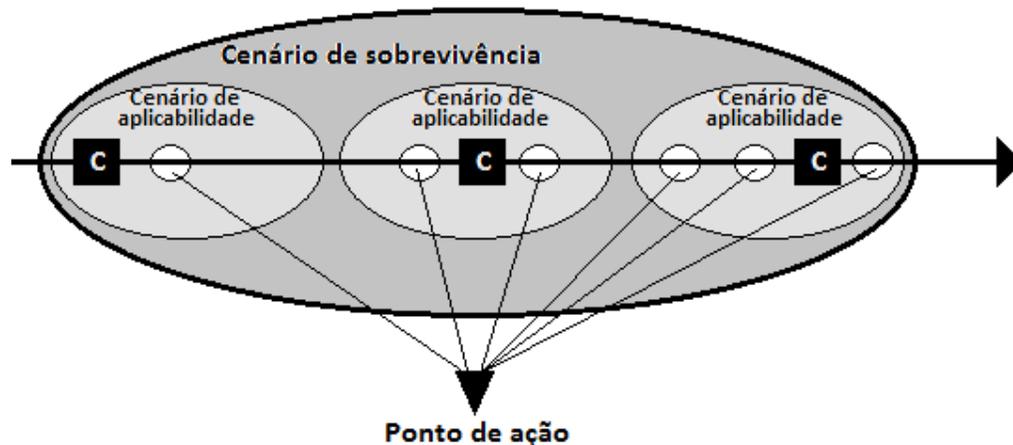


FIGURA 2.3. UMA RESTRIÇÃO EM DIVERSOS CENÁRIOS DE APLICABILIDADE.

2.2.3 MODELANDO AS RESTRIÇÕES

Como as necessidades do mundo real estão sujeitas a mudanças ao longo do tempo, o que é um processo natural, as restrições não podem ser mantidas como estão, elas também devem ser modificadas. É por isso que o conceito de cenário irá auxiliar este trabalho, pois consegue, satisfatoriamente, representar as necessidades de sistemas flexíveis, dinâmicos ou adaptativos (HERTZUM, 2003). Claramente o tempo se torna um atributo fundamental neste trabalho. Mais informações sobre o tempo estão descritas na subseção 2.3.

A associação do tempo com cenários recebe o nome de cenário com limites temporais (*temporal boundary scenario*) fornecendo recursos poderosos para modelar e representar a variabilidade das restrições de bancos de dados em dois

contextos: aplicabilidade e sobrevivência. Este *framework* conceitual será responsável por verificar se o comportamento das ações de integridade será determinado por asserções temporais, isto é, verificar a(s) restrição(ões) do banco de dados em um determinado tempo quando uma informação é inserida, removida ou atualizada, pois uma inserção feita em um momento poderá ter restrição(ões) distinta(s) de uma informação inserida no minuto seguinte. Baseado nas características para cenários definidas por Fontaine (FONTAINE and RAMAUX, 1998), Camolesi (CAMOLESI JR., 2004) estabeleceu algumas características para os cenários com limites temporais para restrições de banco de dados (CAMOLESI JR., 2004):

- Tempo de ocorrência e duração do intervalo são: condição de ativação e instante de ativação;
- O conjunto de fatos/eventos relevantes para cenários com limites temporais neste trabalho é a atualização de objeto (*object updating*);
- Os componentes temporais são as restrições de banco de dados (atores);
- O atraso de reconhecimento (*recognition delay*) depende da duração finita ou infinita do cenário, o qual define o ciclo de vida dos cenários em três fases:
 - Todos os cenários têm um momento de tempo quando começam a sua existência (abertura ou *opening*);
 - Começando por este momento, um cenário possui o desempenho de seu(s) ator(es) (atuação ou *acting*);
 - Cenários podem ter um momento de tempo o qual termina sua existência (fechamento ou *closing*);
- Nas relações de conjuntos, alguns cenários têm relacionamento de composição (é-parte-de) com cenários internos, seguindo as condições temporais básicas envolvendo os momentos de tempo de abertura e fechamento. Portanto, supondo o cenário *Ce1* e seu intra-cenário *Ce2*, temos: $Ce1.opening \leq Ce1.closing$; $Ce2.opening \leq Ce2.closing$; $Ce1.opening \leq Ce2.opening$; $Ce1.closing \geq Ce2.closing$.

Os cenários podem ser abertos e fechados (*opening/closing*) diversas vezes (cenários recorrentes). Pela especificação de existência temporal de

cenários temporais, a definição do momento de abertura (*opening*) e fechamento (*closing*) é obrigatória, salvo a exceção dos cenários permanentes que não possuem *closing* ou aqueles sem *opening*.

2.3 TEMPO

Como este trabalho abrange cenários e restrições temporais, é preciso uma definição de como será tratada a dimensão do tempo. Esta subseção é dedicada ao tratamento e como será modelada a dimensão do tempo, um problema que já foi bastante discutido e pesquisado (ALLEN, 1983; GORALWALLA, TANSEL, 1995; ISO, 2000) direcionando este trabalho para a melhor maneira de trabalhar com o tempo.

A ambiguidade é um dos maiores problemas de se trabalhar com o tempo, já que a interpretação de uma data ou de um horário pode variar. Por exemplo, nos Estados Unidos o mês é informado antes do dia, no Brasil o dia é escrito antes do mês. Portanto, a data 05/10 pode ser interpretada como sendo o quinto dia do mês de outubro ou como o décimo dia do mês de maio, depende se é um brasileiro ou um americano que está lendo a data. Por isso, algumas características para uma definição homogênea do tempo foram adotadas neste trabalho baseado em (CAMOLESI JR., 2006):

- Momento: um valor de um instante de tempo;
- Granularidade: precisão do domínio de um instante de tempo, baseado no modelo ISO (PnYnMnDnTnHnMnS), onde P é um sinalizador de que os próximos valores serão referentes a uma data, por exemplo: 4Y indica quatro caracteres para representar o ano, até encontrar ou não com o sinalizador T que indica que os próximos valores representarão o tempo. Ex.: [20081126101525, P4Y2M2DT2H2M2S].
- Orientação: sistema de referência para a representação temporal. Ex.: calendário gregoriano, chinês, judaico, etc.

- Direção: todas as orientações têm um momento de origem, que será zero. Provavelmente o mais conhecido ponto de origem é o seguido pelos cristãos que referencia o nascimento de Jesus (calendário gregoriano). Por conta disso, conseguimos ter uma noção melhor do tempo, sabemos que o ano de 1500 A.C é na realidade -1500, portanto, há 3510 anos em 2010.
- Aplicação: especificação do uso da representação temporal, possibilitando o reconhecimento semântico do tipo, independente do contexto inserido. Isto possibilitará a identificação não ambígua de uma data, já que [03,P2M] pode ser compreendida como três meses ou o mês de março. Os tipos de valores indicados são três:
 - Ocorrência: para especificar um momento (tempo ou intervalo) onde será realizada ou já foi realizada uma ação. Ex.: Mês em que será efetuada a compra de novos equipamentos [05, P2M] (neste exemplo é maio); meses para negociação comercial [10,12,P2M] (neste exemplo, entre outubro e dezembro);
 - Duração: usando tempo ou intervalo, para especificar a duração de tempo de uma ação (passada ou futura). Ex.: Duração de tempo para instalar novos equipamentos [05,P2M] (neste exemplo são cinco meses); meses para negociação comercial [10,12,P2M] (neste exemplo são três meses);
 - Frequência: para especificar um momento (tempo ou intervalo) usado para registrar repetições. Ex.: O começo do *Backup* [000001,T2H2M2S] (primeiro segundo de todos os dias).

O intervalo também é fundamental para a modelagem do tempo. Quando é utilizado, precisa de dois momentos de tempo que irão especificar o início e fim do intervalo (*StartMoment* e *EndMoment*) combinado do sinal + ou - que irão indicar a direção do tempo. Além desses atributos faz-se necessário o uso de outro, denominado composição discreta, representando o tempo contínuo ou discreto. A composição discreta deve ser ocultada somente quando o intervalo de tempo é contínuo. Quando o intervalo de tempo é discreto ela deve ser usada para

especificar a granularidade dos momentos de tempo. A granularidade da composição discreta deve ser sempre menor que a granularidade do intervalo, portanto, quando o intervalo é dado em anos, a granularidade da composição discreta deve ser especificada em meses, dias, etc.

Por exemplo, um cenário de sobrevivência qualquer, do tipo ocorrência, que foi iniciado em 2001 (*StartMoment*), sua validade é de 9 anos, portanto será fechado em 2010 (*EndMoment*), isso quer dizer que sua direção é +, sua orientação é dada pelo calendário gregoriano e se deseja uma granularidade mensal para sua composição discreta. Este exemplo encontra-se representado na Figura 2.4.

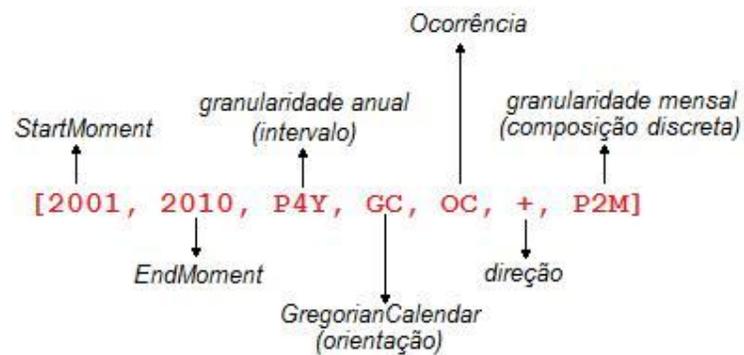


FIGURA 2.4. EXEMPLO DE MODELAGEM DO TEMPO

2.4 RESTRIÇÕES NOS SGBDs

A ferramenta criada neste trabalho extrai restrições de um banco de dados qualquer (através de consultas aos metadados) e depois as insere de volta com as devidas mudanças. Para tornar possível essa extração/inserção de restrições no banco de dados, é preciso ter conhecimento do SGBD que participará da transação.

Cada SGBD possui uma estrutura interna própria para armazenar seus metadados, que são dados que descrevem os dados. Com o conhecimento dos metadados do SGBD, normalmente eles estão dispostos em visões, é possível acessar a forma que as restrições são armazenadas e, conseqüentemente, às

próprias restrições. Nesta subseção é demonstrado como é a estrutura interna do SGBD Microsoft SQL Server através das tabelas de sistema, que são os metadados.

O Microsoft SQL Server cria visões de seus metadados, possibilitando, para o caso deste trabalho, a extração/inserção de restrições. A seguir é apresentada algumas visões deste SGBD.

Tabela 2.1. Exemplo de visões do sistema do Microsoft SQL Server 2005

Visão	Descrição
SYS.OBJECTS	Contém informações de todos os objetos do SGBD.
SYS.DATABASES	Contém os bancos de dados da instância do SGBD.
SYS.SCHEMAS	Contém informações dos esquemas existentes.
SYS.TYPES	Contém os tipos de colunas existentes.
SYS.TABLES	Contém as informações das tabelas dos bancos de dados.
SYS.COLUMNS	Contém as colunas das tabelas.
SYS.KEY_CONSTRAINTS	Contém dados de chaves primárias e restrições únicas.
SYS.INDEX_COLUMNS	Contém as aplicações das chaves primárias e restrições únicas.
SYS.FOREIGN_KEY	Contém as informações das chaves estrangeiras.
SYS.FOREIGN_KEY_COLUMNS	Contém as aplicações das chaves estrangeiras.
SYS.CHECK_CONSTRAINTS	Contém as informações das restrições de checagem.
SYS.DEFAULT_CONSTRAINTS	Contém as restrições <i>default</i> do banco de dados.

No exemplo a seguir, é demonstrada uma consulta simples na visão SYS.OBJECTS do banco de dados MSDB, que é um banco de dados ou esquema padrão do Microsoft SQL Server. Este banco de dados pertence a um grupo chamado *system databases*, que são os esquemas do sistema do SGBD da Microsoft.

Como se pode observar, a Figura 2.5 apresenta todos os objetos existentes neste banco de dados e varia desde uma tabela ou uma visão até um objeto referente a uma *stored procedure*, uma função e também às restrições. Na Figura 2.5 está destacada uma restrição do tipo chave estrangeira. Frequentemente, as restrições do banco de dados estão associadas a uma coluna ou a uma tabela, portanto, é necessário informar à qual tabela ou coluna a restrição pertence. Para

isso existe o atributo `parent_object_id` que informa qual é o objeto-pai do objeto. Por exemplo, na Figura 2.5, apesar de não ser possível ver o número inteiro deste atributo, o número do objeto-pai da restrição em destaque é 2146106686. Portanto, se uma consulta for feita nesta visão buscando o objeto que possui este id, será descoberto quem é o objeto responsável pela restrição. Como pode ser observado na Figura 2.6, o objeto responsável pela restrição é uma tabela chamada “restorefile”.

Além desta visão geral do banco de dados, existem outras visões mais específicas, como por exemplo, as visões das restrições do tipo *check* que existem no banco de dados MSDB. Uma demonstração de uma consulta geral nesta visão está apresentada na Figura 2.7, onde está em destaque a restrição temporal criada para um exemplo do capítulo 4. Como pode ser observado, existem diversos atributos nesta visão, dentre eles, os mais importantes são: o nome da restrição, o objeto-pai da restrição e, obviamente, o valor da restrição semântica, que neste caso informa que o atributo id da tabela à qual esta restrição pertence deve ser maior ou igual a quinze.

name	object_id	principal_id	sch...	parent...	type	type_desc	create_date	modify_date	is_ms...	is_pu...	is_sc...
37 sysxmlplacement	93	NULL	4	0	S	SYSTEM...	2005-10-14 01:36:23.127	2005-10-14 01:36:24.137	1	0	0
38 sysobjkeycrypts	94	NULL	4	0	S	SYSTEM...	2005-10-14 01:36:25.157	2005-10-14 01:36:25.177	1	0	0
39 sysasymkeys	95	NULL	4	0	S	SYSTEM...	2005-10-14 01:36:25.173	2009-11-28 20:10:58.447	1	0	0
40 syssqlguides	96	NULL	4	0	S	SYSTEM...	2005-10-14 01:36:25.190	2009-11-28 20:10:58.490	1	0	0
41 sysbinsubobjs	97	NULL	4	0	S	SYSTEM...	2005-10-14 01:36:22.110	2005-10-14 01:36:23.120	1	0	0
42 sp_verify_jobpro...	2099048	NULL	1	0	P	SQL_ST...	2005-10-14 01:54:36.520	2009-11-28 20:11:04.753	1	0	0
43 syssubsystems	5575058	NULL	1	0	U	USER_T...	2005-10-14 01:54:07.537	2009-11-28 20:10:58.520	1	0	0
44 sp_dts_getfolder	7671075	NULL	1	0	P	SQL_ST...	2005-10-14 01:55:59.863	2009-11-28 20:11:06.877	1	0	0
45 trig_sysmail_prof...	11147...	NULL	1	7986...	TR	SQL_TRI...	2005-10-14 01:55:32.520	2005-10-14 02:02:31.787	1	0	0
46 FK_restorefi_r...	14623...	NULL	1	2146...	F	FOREIG...	2005-10-14 01:55:15.677	2005-10-14 01:56:11.960	1	0	0
47 sp_downloaded...	18099...	NULL	1	0	P	SQL_ST...	2005-10-14 01:54:36.850	2009-11-28 20:11:04.763	1	0	0
48 sysproxysubst...	21575...	NULL	1	0	U	USER_T...	2005-10-14 01:54:07.647	2005-10-14 02:02:31.800	1	0	0
49 sp_dts_putpack...	23671...	NULL	1	0	P	SQL_ST...	2005-10-14 01:55:59.973	2009-11-28 20:11:06.887	1	0	0
50 trig_sysmail_ser...	27147...	NULL	1	1134...	TR	SQL_TRI...	2005-10-14 01:55:32.740	2005-10-14 02:02:31.850	1	0	0
51 restorefilegroup	30623...	NULL	1	0	U	USER_T...	2005-10-14 01:55:15.787	2005-10-14 02:02:31.850	1	0	0
52 sp_post_msx_o...	34099...	NULL	1	0	P	SQL_ST...	2005-10-14 01:54:37.193	2009-11-28 20:11:04.823	1	0	0
53 sysproxylogin	37575...	NULL	1	0	U	USER_T...	2005-10-14 01:54:07.757	2005-10-14 02:02:31.880	1	0	0
54 sp_dts_checkex...	39671...	NULL	1	0	P	SQL_ST...	2005-10-14 01:56:00.083	2005-10-14 02:02:31.897	1	0	0
55 trig_sysmail_ser...	43147...	NULL	1	1230...	TR	SQL_TRI...	2005-10-14 01:55:33.067	2005-10-14 02:02:31.910	1	0	0
56 FK_restorefi_r...	46623...	NULL	1	3062...	F	FOREIG...	2005-10-14 01:55:15.787	2005-10-14 01:56:02.413	1	0	0
57 sp_verify_perfor...	50099...	NULL	1	0	P	SQL_ST...	2005-10-14 01:54:37.507	2009-11-28 20:11:04.833	1	0	0
58 DF_sysproxylo...	53575...	NULL	1	3757...	D	DEFAULT...	2005-10-14 01:54:07.757	2005-10-14 01:56:02.413	1	0	0
59 sp_dts_addfolder	55671...	NULL	1	0	P	SQL_ST...	2005-10-14 01:56:00.193	2009-11-28 20:11:06.893	1	0	0
60 trig_sysmail_con...	59147...	NULL	1	1390...	TR	SQL_TRI...	2005-10-14 01:55:33.287	2005-10-14 02:02:31.943	1	0	0
61 logmarkhistory	62623...	NULL	1	0	U	USER_T...	2005-10-14 01:55:15.897	2009-11-28 20:10:58.543	1	0	0

FIGURA 2.5. CONSULTA À VISÃO `SYS.OBJECTS` DO BANCO DE DADOS MSDB

FABIODNB-PC\SQL... SQLQuery1.sql* Summary

```
use msdb;
select * from sys.objects where object_id=2146106686;
```

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_fulltext_catalog
1	restorefile	2146106686	NULL	1	0	U	USER_TABLE	2005-10-14 01:55:15.677	2005-10-14 02:02:35.927	1	0

Query executed successfully. FABIODNB-PC\SQLEXPRESS (9.0 SP3) FabioDNB-PC\Fabio DNB (52) msdb 00:00:00 1 rows

FIGURA 2.6. CONSULTA AO OBJETO-PAI DA RESTRIÇÃO EM DESTAQUE NA FIGURA 2.5

FABIODNB-PC\SQL... SQLQuery1.sql* Summary

```
use msdb select * from sys.check_constraints;
```

	name	object_id	parent_object_id	definition	create_date
1	CK_originating_server_id_MustBe_1	181575685	133575514	((originating_server_id)=(1))	2005-10-14 01:5
2	CK_master_server_MustBe_1	213575799	133575514	((master_server)=(1))	2005-10-14 01:5
3	Variante de teste dissertação	493244812	805577908	((id)>=(15))	2010-01-09 09:1
4	versao nova	679673469	1454628225	((attachment_encoding)='UUEN...	2009-12-05 13:4
5	sysmail_OutMailMustHaveAtleastOneRecipient	1566628624	1454628225	(NOT ([recipients] IS NULL AND...	2005-10-14 01:5
6	sysmail_OutMailRecipientCannotBeEmpty	1582628681	1454628225	((datalength(isnull([recipients],...))	2005-10-14 01:5
7	sysmail_OutMailImportanceMustBeValid	1614628795	1454628225	((importance)='HIGH' OR [import...	2005-10-14 01:5
8	sysmail_OutMailSensitivityMustBeValid	1630628852	1454628225	((sensitivity)='CONFIDENTIAL' O...	2005-10-14 01:5

Query executed successfu... FABIODNB-PC\SQLEXPRESS (9.0 SP3) FabioDNB-PC\Fabio DNB (52) msdb 00:00:00 8 rows

FIGURA 2.7. CONSULTA À VISÃO DAS RESTRIÇÕES DO TIPO CHECK DO BANCO DE DADOS MSDB

Esse formato de organizar internamente os dados com uma visão mais abrangente (SYS.OBJECTS) e outras menores, por exemplo SYS.CHECK_CONSTRAINTS, é muito eficaz e, por isso, foi adotado para organizar o banco de dados da ferramenta desenvolvida neste trabalho. Isso será mais detalhado no capítulo 3.

2.5 TRABALHOS CORRELATOS

Dos poucos trabalhos encontrados semelhantes a este, o que mais se assemelha é o de autoria de Robson Leonardo Ferreira Cordeiro, que, em seu

mestrado na UFRGS (CORDEIRO, SANTOS, 2004) trabalhou com restrições de banco de dados, mencionando a importância de existir um histórico dos dados. Porém o foco de seu trabalho, diferente deste, contempla apenas tempo e versão. Em seu trabalho é afirmado que “restrições que consideram a gerência de tempo e de versões sobre dados ainda representam uma área de pesquisa praticamente inexplorada”, o que é uma realidade e uma das razões para surgirem pesquisas nesse contexto (CAMOLESI JR., 2004; CAMOLESI JR., 2006). O primeiro objetivo do trabalho de Cordeiro consiste em “definir uma classificação de restrições de integridade para bases de dados com suporte a tempo e versões, a fim de prover uma base para o desenvolvimento de pesquisas relacionadas à sua especificação e manutenção”. O segundo objetivo consiste em agregar ao Modelo Temporal de Versões (TVM) uma linguagem que permita a especificação de restrições de integridade. Com base nessas descrições dos objetivos, podemos notar a diferença do trabalho dele e do apresentado nesta dissertação.

Outro trabalho relacionado envolve condições temporais e restrições de integridade em bancos de dados ativos (SISTLA and WOLFSON, 1995). Neste artigo é apresentado um formalismo unificado baseado em PTL (*Past Temporal Logic*) para especificar eventos e condições em sistemas de banco de dados ativos. O modelo proposto trabalha com regras no formato C-A (condição-ação), diferente, por exemplo, do modelo utilizado pelas *triggers* que é E-C-A (evento-condição-ação). Alguns operadores PTL são, por exemplo, *Since*, *Lasttime* e *Previously*. Além dessa linguagem apresentada para trabalhar com lógica temporal, é apresentado um algoritmo incremental para processar as condições especificadas pela linguagem.

Uma terceira ferramenta relacionada a este trabalho é intitulada X-Tool de Igor Fabiano Nazar da UFPR (NAZAR and VERGILIO, 2007). Este trabalho foi mencionado porque a ferramenta X-tool possui um procedimento que faz a extração de elementos e das restrições de um esquema e gera um modelo formal. O fato de extrair restrição foi a razão da semelhança.

Outra ferramenta semelhante vem da UFSC e possui o nome FERIX (MADUELL and MELLO, 2007), a razão da semelhança é porque esta ferramenta possibilita a extração de restrições de integridade de fontes de dados XML, o restante da ferramenta FERIX está bem distante deste trabalho.

2.6 CONSIDERAÇÕES FINAIS

As restrições semânticas sempre estiveram presentes nos bancos de dados, mas nunca tiveram essa capacidade e flexibilidade de variar de acordo com o tempo. Por esta razão, Camolesi (CAMOLESI JR., 2004) as batizou de variantes.

As seções expostas neste capítulo estão intrinsecamente ligadas. A capacidade de modelar o tempo utilizando cenários de sobrevivência para esculpir um escopo de vida para as restrições e também a criação de cenários de aplicabilidade para informar o momento ou período de tempo exato em que as restrições devem ser ativadas ou desativadas jamais foi visto. Este trabalho apresenta uma maneira díspar de qualquer outra forma de trabalhar com restrições, além de eficiente e seguro, pois o histórico das restrições está armazenado e uma restrição pode ser facilmente reativada sem maiores problemas.

3 MODELAGEM DA FERRAMENTA

3.1 CONSIDERAÇÕES INICIAIS

Neste capítulo é apresentada a proposta do trabalho, seus benefícios, as razões de sua existência e os objetivos da ferramenta. Também são apresentados diagramas de fácil compreensão (diagramas de classes) da ferramenta.

3.2 REQUISITOS

A tarefa de um banco de dados é formar um repositório de dados para uma organização no qual informações são extraídas por diversas razões. Para isso ter validade, o banco de dados necessita de uma fiel representação do mundo real e não somente no momento em que foi criado. A qualquer momento, o banco de dados precisa representar os acontecimentos do mundo real. É lógico pensar que a existência de um banco de dados não teria razão caso estivesse representando uma atividade que não existisse no mundo real (ELMASRI and NAVATHE, 2006).

Para satisfazer as mudanças do mundo real, o banco de dados tem que ser readaptado, tanto com uma simples modificação em uma restrição, como também em uma mudança na própria estrutura do banco de dados, podendo exigir até uma nova reengenharia.

Existe a impressão que as restrições criadas para um banco de dados raramente são modificadas, porém é uma impressão falsa, pois as alterações nas restrições de banco de dados são tão frequentes quanto qualquer outra mudança estrutural (CAMOLESI JR., 2004), sendo que, em muitos casos, as alterações nas restrições podem ser a maior razão de mudanças no banco dados.

Por exemplo, uma loja qualquer com foco em apenas um nicho do mercado, que possui um banco de dados cujo levantamento de requisitos foi bem feito, assim como a engenharia do projeto, poderá não precisar de uma nova tabela no banco de dados ou de alterar algo em sua estrutura que não sejam as restrições. Eventuais promoções aos seus clientes, descontos em certos períodos, mudar o tempo necessário para efetivar um funcionário, etc., são mudanças que envolvem apenas as restrições do banco de dados. Por exemplo, do dia primeiro de março ao dia primeiro de abril conceda desconto de 10% a todos os clientes com mais de um “ano de casa”.

Como consequência deste trabalho, uma ferramenta foi desenvolvida com a capacidade de auxiliar e melhorar a maneira em que o DBA gerencia as restrições semânticas que existem nos bancos de dados. Esta ferramenta também provê ao DBA a capacidade de, grosseiramente falando, temporizar restrições, de modo que ele consiga, de uma maneira mais eficiente, estabelecer o período de tempo em que uma restrição é válida. Esta capacidade de estabelecer prazos para as restrições, consequentemente automatiza a forma em que o DBA trabalha com as restrições, pois uma vez que uma restrição é definida na ferramenta, ela possuirá o seu período de atuação no banco. O DBA não precisa se preocupar com a restrição, pois, quando o momento de início do período de atuação da restrição for atingido, a própria ferramenta irá inseri-la no banco de dados. Da mesma forma, quando o momento final do período de validade da restrição for atingido, a ferramenta irá removê-la do banco de dados.



FIGURA 3.1. INTERAÇÃO ENTRE O DBA (USUÁRIO) E AS RESTRIÇÕES DO BANCO.

Como mostra a Figura 3.1, a ferramenta é, essencialmente, uma interface que faz a interação entre o DBA e as restrições do banco de dados,

facilitando assim o gerenciamento das mesmas. A ferramenta trabalha em conjunto com o SGBD escolhido para auxiliar o DBA a manipular e controlar restrições, principalmente as semânticas. O objetivo é prover uma maior capacidade de gerenciamento do banco de dados com as novas facilidades que a ferramenta propõe como, por exemplo, o versionamento de restrições, histórico de restrições, automatização de restrições, empacotamento de restrições.

A seguir são apresentadas as funcionalidades da ferramenta:

- **Histórico de restrições:** A capacidade de armazenar o histórico das restrições é um aspecto fundamental que a ferramenta propicia ao DBA. Os bancos de dados convencionais podem ser vistos como *snapshot databases* (que em português seria algo parecido com banco de dados instantâneos), uma vez que representam o estado de uma empresa naquele exato momento de tempo. Isto pode causar uma má interpretação da informação contida no banco de dados, já que muitas vezes a informação pode estar discrepante se o histórico não for levado em conta. Por exemplo, imagine a situação de um banco de dados que há cinco anos criou a restrição dizendo que o salário dos funcionários não pode exceder o valor de U\$ 20.000, porém antes disso, alguns funcionários ultrapassaram este valor por bonificações e mérito. Analisando este caso no momento, sem maiores informações de, por exemplo, quando a restrição foi criada, isto caracterizaria uma discrepância no banco de dados ou até a suspeita de que o DBA está privilegiando alguns funcionários. O simples fato de se possuir um histórico das restrições resolveria este problema.
- **Empacotamento de restrições:** Pacote é a funcionalidade da ferramenta que tornará possível trabalhar com um conjunto de restrições que estão interligadas por um mesmo contexto. Por exemplo, se é necessário ativar 20 restrições durante uma data festiva (natal, por exemplo), será muito mais fácil trabalhar com elas agrupadas, já que isto torna possível a ativação e desativação do pacote como um todo, ao invés de programar cada restrição

isoladamente. Ao agrupá-las em pacotes (empacotamento), a manipulação destas será mais fácil, rápida e organizada.

- **Versionamento de restrições:** A possibilidade de se criar versões de restrições habilita a ferramenta a clonar as restrições desejadas, isso será importante para trabalhar com o histórico das restrições e também porque, caso se deseja mudar o valor de uma restrição, a maneira mais fácil e correta de fazer isso é criando uma versão desta restrição. É como se a restrição evoluísse mudando de valor para se adequar às necessidades do mundo real.
- **Variantes:** A associação dos conceitos explanados na Seção 2 propicia a criação das variantes. Em seu âmago, uma variante é uma restrição de banco de dados cuja existência depende do tempo. É uma nova maneira de se enxergar as invariantes, que é o nome usado para referenciar as restrições dos bancos de dados (DATE, 2003; ELMASRI and NAVATHE, 2006; RAMAKRISHNAN and GEHRKE, 2002). As variantes são o ápice da ferramenta, pois além de ser uma inovação, provê aos DBAs uma capacidade de gerenciamento das restrições muito mais flexível e eficiente. Com a inserção das variantes, conseqüentemente facilitando o gerenciamento das restrições, espera-se que os projetistas de banco de dados dêem mais atenção à modelagem das restrições enquanto o projeto está em fase inicial. É sabido que esta é uma tarefa pouco executada (CURRIM and RAM, 2008), espera-se também inspirar os DBAs a explorarem mais o uso das restrições dos bancos de dados, pois ativar e desativar 50 restrições em momentos distintos já não será mais uma tarefa tão desgastante.

A ferramenta é constituída de cinco casos de uso, sendo que em todos eles acontece a interação do usuário (DBA) e estão representados pela Figura 3.2. Como o objetivo desta Figura é demonstrar apenas uma visão geral da ferramenta, as funcionalidades estão encapsuladas e cada uma será explicada e expandida detalhadamente mais adiante. Por enquanto, um resumo dessas funcionalidades é descrito a seguir:

- **Funções Essenciais:** Funções vitais à ferramenta, por exemplo, registrar um servidor, extrair um banco de dados, atualizar restrições;
- **Manipular Restrições:** Trabalhar com restrições removendo-as, inserindo-as e modificando-as;
- **Manipular Variantes e cenários de sobrevivência:** Trabalhar com as variantes associadas aos cenários de sobrevivência;
- **Manipular Versões:** Trabalhar com versões das restrições, podendo também criar versões de versões;
- **Manipular Pacotes:** Trabalhar com pacotes que irão agrupar as restrições para que possam ser manipuladas em conjunto.

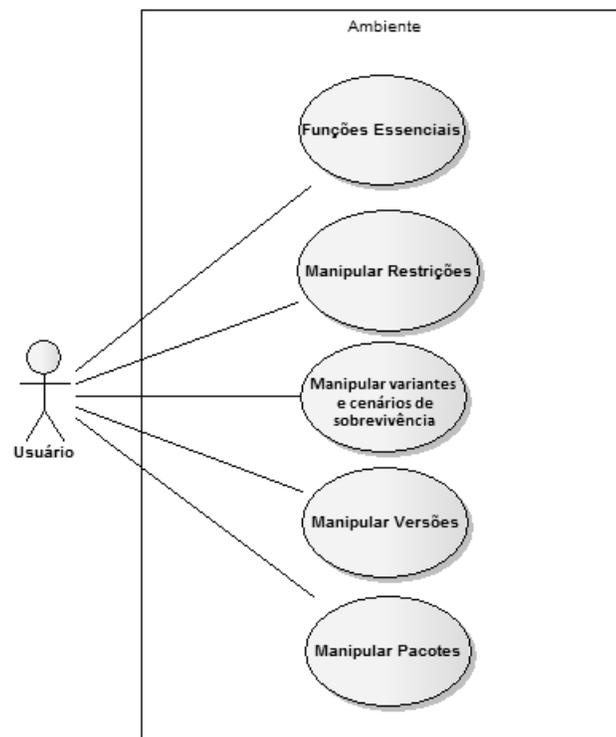


FIGURA 3.2. CASO DE USO: VISÃO GERAL DA FERRAMENTA.

3.3 MODELAGEM

Aprofundando com a modelagem da ferramenta, nesta seção será apresentado o detalhamento das funcionalidades, onde cada funcionalidade é estendida para outro caso de uso específico.

3.3.1 FUNÇÕES ESSENCIAIS

A ferramenta é constituída essencialmente de seis funcionalidades, que estão representadas na Figura 3.3 e são: Registrar servidor, Excluir servidor, Extrair banco de dados, Atualizar restrições, Executar mudanças, Excluir banco de dados. Três atores existem para interagir com essas funcionalidades, que são: o DBA (usuário), o banco de dados alvo de onde as restrições serão consultadas e “extraídas” e o Timer, que pode ser visto como a própria ferramenta que efetuará atualizações nos bancos de dados alvo automaticamente, tal como: remover uma restrição que chegou ao final de seu “prazo de validade”.

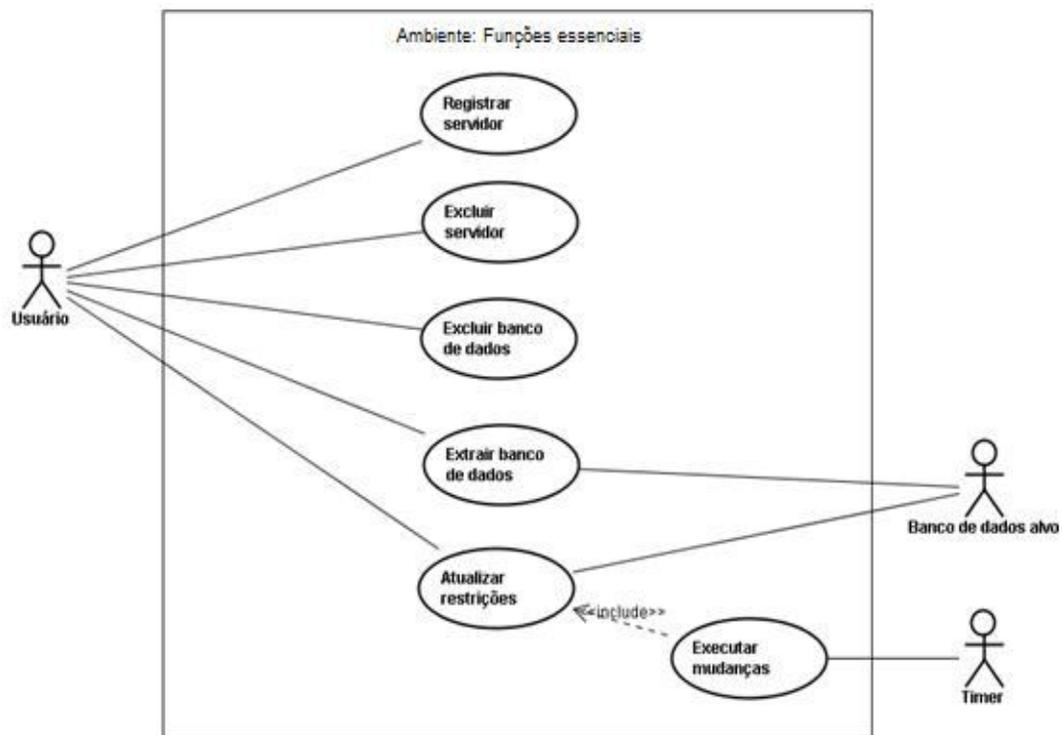


FIGURA 3.3. CASO DE USO: FUNÇÕES ESSENCIAIS.

A funcionalidade registrar servidor tem por objetivo registrar um servidor à ferramenta. O servidor servirá como um container para os esquemas/bancos de dados que serão extraídos adiante, o nome do servidor é

escolhido pelo DBA conforme desejado e deve-se informar qual o SGBD correspondente. Além disso, é necessário informar um usuário e sua senha do SGBD (com privilégios de administrador) para tornar possível o acesso aos metadados. A funcionalidade Excluir servidor fará a remoção de um servidor do banco de dados da ferramenta e irá excluir todos os registros referentes aos esquemas/bancos de dados contidos no servidor que será eliminado. A funcionalidade Extrair banco de dados faz a extração de um esquema/banco de dados de um SGBD específico. Quando um servidor é registrado, ele disponibiliza ao DBA todos os bancos contidos nesse SGBD e com isso o DBA pode escolher de qual ele deseja fazer a extração. Um servidor pode conter nenhum ou quantos esquemas/bancos de dados forem desejados. A funcionalidade Excluir Banco de Dados irá excluir um esquema/banco de dados que foi previamente extraído do SGBD. Todos os registros referentes a este esquema serão apagados do banco de dados da ferramenta. A funcionalidade atualizar restrições faz a ativação ou desativação de uma restrição. Como pode ser observado na Figura 3.3, os atores usuário (DBA) e Timer podem ativar esta funcionalidade informando que a ativação ou desativação de uma restrição pode ser feita tanto pelo DBA, portanto manualmente, quanto pela ferramenta sem a intervenção do DBA, indicando uma atualização automática. A funcionalidade executar mudanças indica a habilidade da ferramenta de ativar ou desativar as restrições do banco de dados.

Para facilitar ainda mais a compreensão dessas funcionalidades essenciais da ferramenta, a seguir estão alguns quadros explicando-as detalhadamente (Quadro 3.4.1 até 3.4.6):

Quadro 3.4.1 – Caso de uso: Registrar Servidor

Característica	Descrição
Resumo/Objetivo	Registrar servidor remoto no banco de dados da ferramenta
Atores	Usuário
Pré-condição	-
Descrição	O Registro do servidor é a etapa inicial das possíveis atividades dentro da ferramenta.
Cenários Exceções	Erro na tentativa de conexão, não registra servidor
Pós-Condição	Servidor registrado no banco de dados da ferramenta

Quadro 3.4.2 - Caso de uso: Excluir Servidor

Característica	Descrição
Resumo/Objetivo	Excluir servidor do banco de dados da ferramenta
Atores	Usuário
Pré-condição	Servidor precisa estar registrado no banco de dados da ferramenta
Descrição	Exclui o servidor e todos seus dependentes do banco de dados da ferramenta, por exemplo, os esquemas do servidor excluído também serão removidos
Cenários Exceções	-
Pós-Condição	Registro do servidor e todos seus dependentes excluídos

Quadro 3.4.3 - Caso de uso: Excluir Banco de Dados

Característica	Descrição
Resumo/Objetivo	Excluir banco de dados extraído de um servidor da ferramenta
Atores	Usuário
Pré-condição	Banco de dados deve ter sido registrado (extraído)
Descrição	Exclui o banco de dados extraído e todos seus dependentes do banco de dados da ferramenta
Cenários Exceções	-
Pós-Condição	Registro do banco de dados extraído e todos seus dependentes excluídos

Quadro 3.4.4 – Caso de uso: Extrair Banco de Dados

Característica	Descrição
Resumo/Objetivo	Extrai os meta-dados do servidor registrado
Atores	Usuário, Banco de dados alvo
Pré-condição	Servidor deve estar registrado no sistema
Descrição	Extrai a estrutura do banco de dados como, por exemplo, esquemas, tabelas, colunas, restrições e etc...
Cenários Exceções	1 – Erro de conexão com o servidor, cancelado a extração. 2 – Problema com a extração. A extração é feita através de transação (atomicidade), portanto, qualquer problema que ocorra a extração não será feita.
Pós-Condição	Estrutura do banco de dados registrado no banco de dados da ferramenta

Quadro 3.4.5 – Caso de uso: Atualizar Restrições

Característica	Descrição
Resumo/Objetivo	Atualizar as alterações feitas com as restrições no banco de dados local para o banco de dados alvo
Atores	Usuário, Banco de dados alvo
Pré-condição	Banco de dados deve ter sido extraído e conter restrições
Descrição	Adiciona/Altera/Remove as restrições que foram alteradas do banco de dados local, no banco de dados alvo
Cenários Exceções	1 – Erro de conexão com o banco de dados local ou com o banco de dados alvo. Atualização cancelada.
Pós-Condição	Restrições sincronizadas entre o banco de dados da ferramenta e o banco de dados alvo, as antigas restrições do banco de dados alvo serão apagadas, porém não serão perdidas, pois a ferramenta mantém um histórico das restrições

Quadro 3.4.6 - Caso de uso: Executar mudanças

Característica	Descrição
Resumo/Objetivo	Ativa as restrições que estão esperando para serem ativadas e desativa as restrições que possuem uma data para serem desativadas.
Atores	Timer
Pré-condição	Banco de dados deve ter sido extraído e conter restrições
Descrição	Pesquisa as restrições dos bancos de dados registrados e verifica se alguma mudança precisa ser feita. Caso sim, a ferramenta irá fazer essas modificações disparando o caso de uso Atualizar Restrições. Isso ocorre sem a interação do usuário. Por exemplo, quando uma restrição chega ao momento em que deve ser ativada
Cenários Exceções	-
Pós-Condição	Se alguma mudança é necessária, então será disparado o caso de uso Atualizar Restrições.

3.3.2 MANIPULAR RESTRIÇÕES

As restrições, como já amplamente descritas na Seção 2, possuem grande importância neste trabalho. O diagrama representado pela Figura 3.4, apresenta quais são as opções do usuário quando o mesmo deseja trabalhar com as restrições. Após alguns estudos, foi verificado que o Microsoft SQL Server não está mais trabalhando com restrições do tipo *rule*. No site (MICROSOFT-DEVELOPER-NETWORK) que pode ser encontrado nas referências bibliográficas, é informado que se for desejado trabalhar com *rules*, é melhor utilizar uma restrição do tipo *check* que o efeito será o mesmo. Por esta razão, essa funcionalidade de manipular e criar restrição do tipo *rule* não foi implementada na ferramenta.

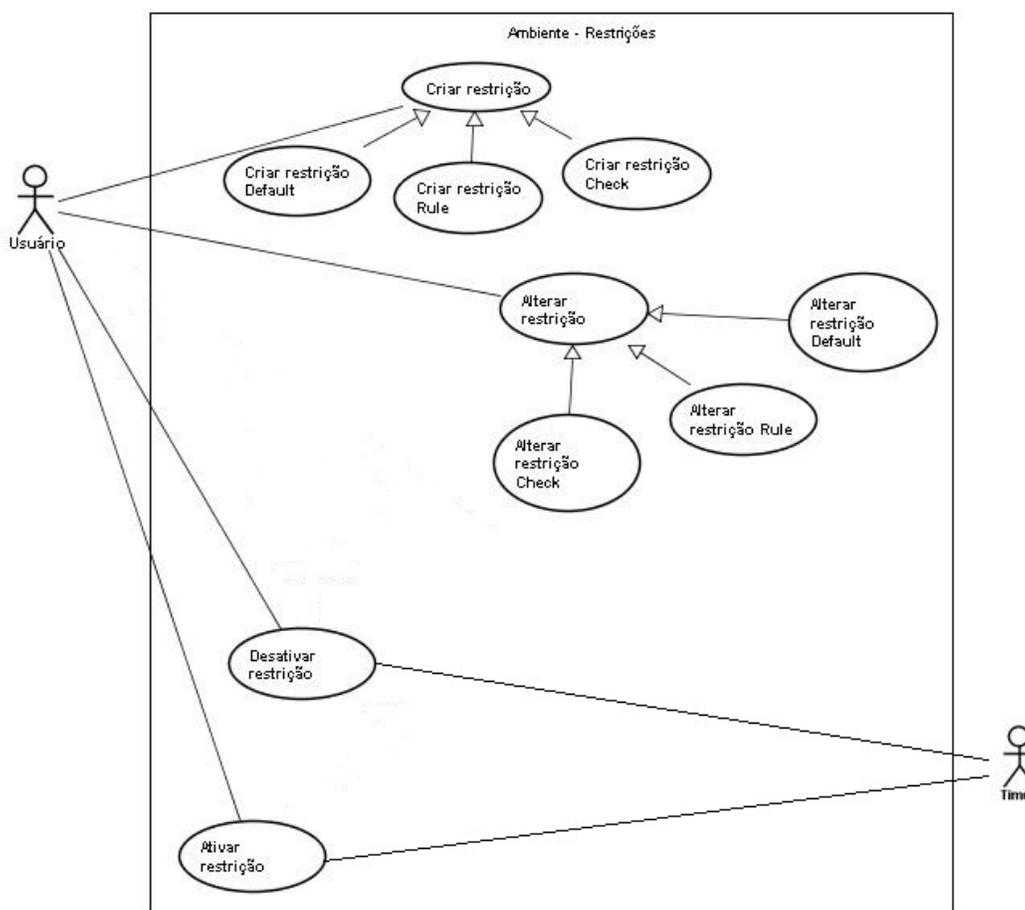


FIGURA 3.4. CASO DE USO: RESTRIÇÕES

O usuário (DBA) pode criar restrições, vale salientar que as restrições referenciadas na Figura 3.4 são apenas as invariantes. As restrições possuem três tipos: *Default*, *Rule* e *Check*, o usuário deverá escolher qual a restrição que ele deseja criar. A funcionalidade alterar restrição, internamente, irá criar uma nova versão da restrição escolhida, pois como já mencionado anteriormente, é mantido o histórico de todas as restrições. Portanto, quando se deseja alterar uma restrição, esta será desativada e uma nova restrição assumirá seu posto. O usuário pode desativar ou ativar uma restrição, estas duas funcionalidades são duas especificações da funcionalidade executar mudanças da Figura 3.3. A seguir, serão apresentados alguns quadros com a finalidade de explicar melhor as funcionalidades desta Figura (Quadros 3.4.7 até 3.4.10):

Quadro 3.4.7 – Caso de uso: Criar restrição

Característica	Descrição
Resumo/Objetivo	Adicionar restrição na estrutura de um banco de dados
Atores	Usuário
Pré-condição	Banco de dados deve ter sido extraído
Descrição	Adiciona restrição no banco de dados local para que a atualização possa ser feita imediatamente ou possa ser postergada.
Cenários Exceções	1 – Dependendo da restrição, apenas uma instância dela existe. Caso isso aconteça, não será possível atualizar o BD alvo
Pós-Condição	Restrição registrada

Quadro 3.4.8 – Caso de uso: Alterar restrição

Característica	Descrição
Resumo/Objetivo	Alterar restrição na estrutura de um banco de dados
Atores	Usuário
Pré-condição	Restrição deve existir
Descrição	Altera a restrição no banco de dados local para imediatamente ou posteriormente atualizar o BD alvo. Essa alteração é feita através da criação de uma nova versão.
Cenários Exceções	-
Pós-Condição	Restrição alterada

Quadro 3.4.9 - Caso de uso: Desativar restrição

Característica	Descrição
Resumo/Objetivo	Desativar restrição na estrutura de um banco de dados
Atores	Usuário
Pré-condição	Restrição deve existir e estar ativada
Descrição	Desativa a restrição no banco de dados local para ser imediatamente ou posteriormente atualizada com o banco de dados alvo
Cenários Exceções	-
Pós-Condição	Restrição desativada

Quadro 3.4.10 - Caso de uso: Ativar restrição

Característica	Descrição
Resumo/Objetivo	Ativar restrição na estrutura de um banco de dados
Atores	Usuário
Pré-condição	Restrição deve existir e estar desativada
Descrição	Ativa a restrição
Cenários Exceções	-
Pós-Condição	Restrição ativada

3.3.3 MANIPULAR VARIANTES E CENÁRIOS DE SOBREVIVÊNCIA

Com a criação desta ferramenta e a implementação de cenários de sobrevivência e aplicabilidade, é possível agora criar variantes. As variantes são semelhantes às invariantes, porém estão ligadas aos cenários de sobrevivência e aplicabilidade, tornando-as extremamente dependentes do tempo. A possibilidade de criar cenários que tornarão essas variantes vivas ou ativas, é essencial para que a ferramenta possa ativá-las ou desativá-las, inserindo-as ou removendo-as do

banco de dados alvo. O cenário de aplicabilidade é criado quando se cria uma variante. O caso de uso das variantes e cenários de sobrevivência encontra-se na Figura 3.5:

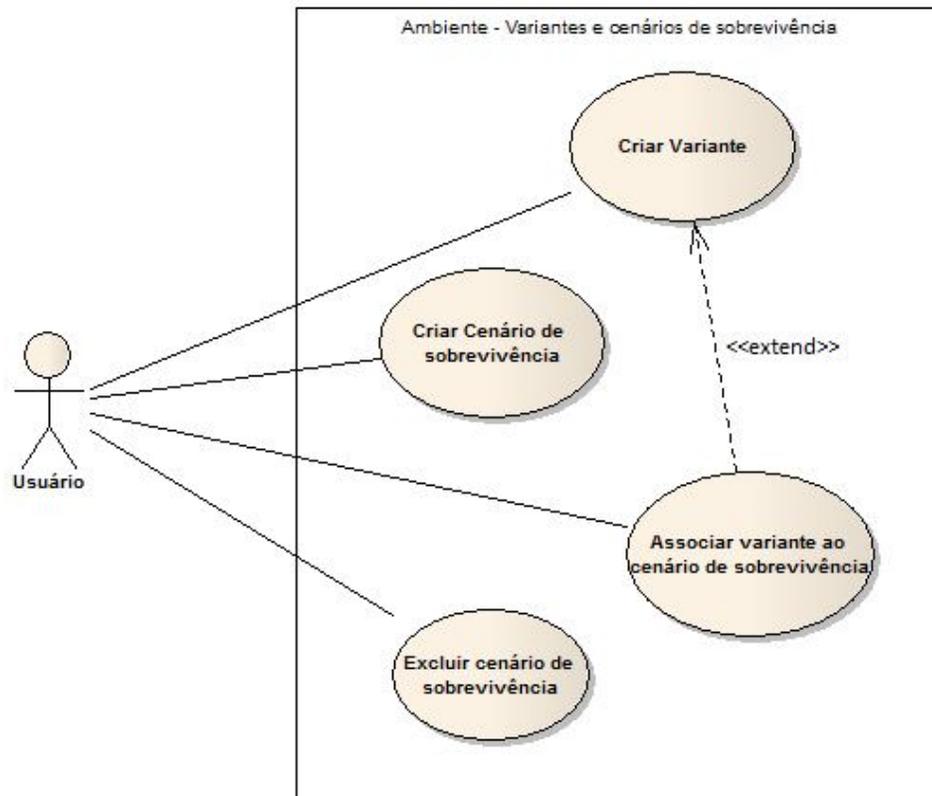


FIGURA 3.5. CASO DE USO: VARIANTES E CENÁRIOS DE SOBREVIVÊNCIA

Observe que não existe a funcionalidade excluir variante ou desativar variante, isso porque a ativação e desativação das variantes é extremamente dependente do cenário de sobrevivência ao qual ela pertence, é ele quem irá “acordá-la” e deixá-la pronta para ser ativada pela ferramenta. Caso seja desejado desativar uma variante, basta associá-la ao cenário de sobrevivência global.

Como apresentado na Figura 3.5, o DBA pode criar uma variante. A variante tem as mesmas propriedades das invariantes, portanto, pode ser uma restrição *check*, *default* ou *rule*, além de outras relativas aos cenários. As propriedades relativas aos cenários, as invariantes não possuem. Para a variante ser ativada pela ferramenta, ela precisa estar associada a um cenário de sobrevivência

que não seja o global, portanto o DBA precisa criar um cenário de sobrevivência para que seja possível associar uma variante a este cenário de sobrevivência. O DBA também pode excluir um cenário de sobrevivência, isto não irá apagar as variantes associadas a este cenário, apenas as desativará caso alguma esteja ativa. A seguir, serão apresentados alguns quadros com a finalidade de explicar melhor as funcionalidades da Figura 3.5 (Quadros 3.4.11 até 3.4.14):

Quadro 3.4.11 – Caso de uso: Criar variante

Característica	Descrição
Resumo/Objetivo	Criar variante
Atores	Usuário
Pré-condição	-
Descrição	Cria uma variante que, ao se criada, ficará associada ao cenário global e nunca será ativada. Ela só será ativada quando for associada a algum cenário de sobrevivência, ele é quem dá vida as variantes. O cenário de aplicabilidade é um dos atributos das variantes. Opcional (<i>extend</i>): Associar esta variante a um cenário de sobrevivência
Cenários Exceções	-
Pós-Condição	Variante criada e associada ao cenário global ou a algum cenário de sobrevivência

Quadro 3.4.12 - Caso de uso: Criar cenário de sobrevivência

Característica	Descrição
Resumo/Objetivo	Criar cenário de sobrevivência
Atores	Usuário
Pré-condição	-
Descrição	Cria um cenário de sobrevivência que irá ser o escopo de vida para as variantes que forem associadas a ele. Existem três tipos de cenários de sobrevivência, que são; sem abertura, sem fechamento e o normal que possui os momentos de abertura e fechamento
Cenários Exceções	-
Pós-Condição	Cenário de sobrevivência criado

Quadro 3.4.13 - Caso de uso: Excluir cenário de sobrevivência

Característica	Descrição
Resumo/Objetivo	Excluir cenário de sobrevivência
Atores	Usuário
Pré-condição	Existência do cenário de sobrevivência
Descrição	Exclui um cenário de sobrevivência do BD local. Irá desativar todas as variantes que estão ativadas, porém as variantes não serão excluídas do BD local
Cenários Exceções	-
Pós-Condição	Cenário de sobrevivência excluído

Quadro 3.4.14 - Caso de uso: Associar variante ao cenário de sobrevivência

Característica	Descrição
Resumo/Objetivo	Associar variantes a um cenário de sobrevivência
Atores	Usuário
Pré-condição	Existência de variantes e de cenários de sobrevivência
Descrição	Associa uma variante a um cenário de sobrevivência, podendo assim, a variante ser ativada pela ferramenta
Cenários Exceções	-
Pós-Condição	Variante associada a um cenário de sobrevivência

3.3.4 MANIPULAR VERSÕES

A possibilidade de se criar versões de restrições, variantes e pacotes, como mostra a Figura 3.6, faz com que a ferramenta possa, por exemplo, copiar as restrições desejadas e atribuir valores diferentes a elas. Com isso, as restrições poderão ser utilizadas em um mesmo local do banco de dados, porém em diferentes períodos de tempo. A criação de versões possibilita também uma melhor organização do histórico das restrições.

Assim que uma versão é criada, podendo ser de uma restrição simples ou de uma variante ou mesmo de um pacote, essa versão terá as mesmas

funcionalidades e irá se comportar como se fosse um dado criado da forma normal. Por exemplo, ao criarmos uma versão de uma variante, essa nova versão pode possuir um nome diferente, um valor diferente e até um período de aplicabilidade diferente, o que muda é que existe uma coluna no banco de dados local, chamada de *version*, para as tabelas das invariantes, variantes e pacotes. O valor para essa coluna será calculado assim que uma nova versão for inserida, informando assim a versão da restrição ou do pacote. A funcionalidade criar versão de pacote irá copiar um pacote inteiro e possibilitar que o pacote receba novas configurações, por exemplo, quando deve ser ativado ou desativado.

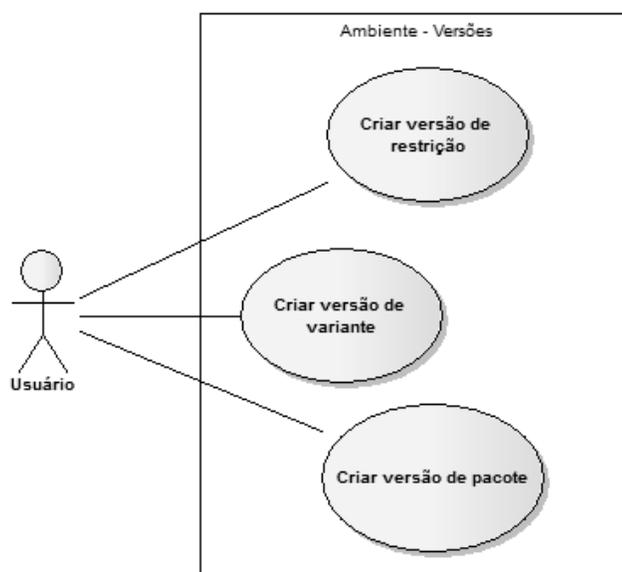


FIGURA 3.6. CASO DE USO: VERSÕES

A seguir, são apresentados alguns quadros com a finalidade de explicar melhor as funcionalidades desta Figura (Quadros 3.4.15 até 3.4.17):

Quadro 3.4.15 – Caso de uso: Criar versão de restrição

Característica	Descrição
Resumo/Objetivo	Criar versão de uma restrição simples
Atores	Usuário
Pré-condição	Existência da restrição
Descrição	Assim que a versão é criada, elas estarão ligadas permanentemente, a versão pode ser uma cópia idêntica da restrição ou não
Cenários Exceções	-
Pós-Condição	Versão criada

Quadro 3.4.16 - Caso de uso: Criar versão de variante

Característica	Descrição
Resumo/Objetivo	Criar versão de uma variante
Atores	Usuário
Pré-condição	Existência da variante
Descrição	Assim que a versão é criada, elas estarão ligadas permanentemente, a versão pode ser tanto uma cópia idêntica da variante ou não
Cenários Exceções	-
Pós-Condição	Versão criada

Quadro 3.4.17 - Caso de uso: Criar versão de pacote

Característica	Descrição
Resumo/Objetivo	Criar versão de um pacote
Atores	Usuário
Pré-condição	Existência do pacote
Descrição	Assim que a versão é criada, os pacotes estarão ligados permanentemente, a versão pode ser tanto uma cópia idêntica do pacote ou não
Cenários Exceções	-
Pós-Condição	Versão criada

3.3.5 MANIPULAR PACOTES

Pacote é uma funcionalidade da ferramenta que tornará possível trabalhar com um conjunto de regras que estão interligadas. Ao agrupá-las em pacotes, a manipulação destas será mais fácil, rápida e organizada. O caso de uso desta funcionalidade encontra-se representado pela Figura 3.7.

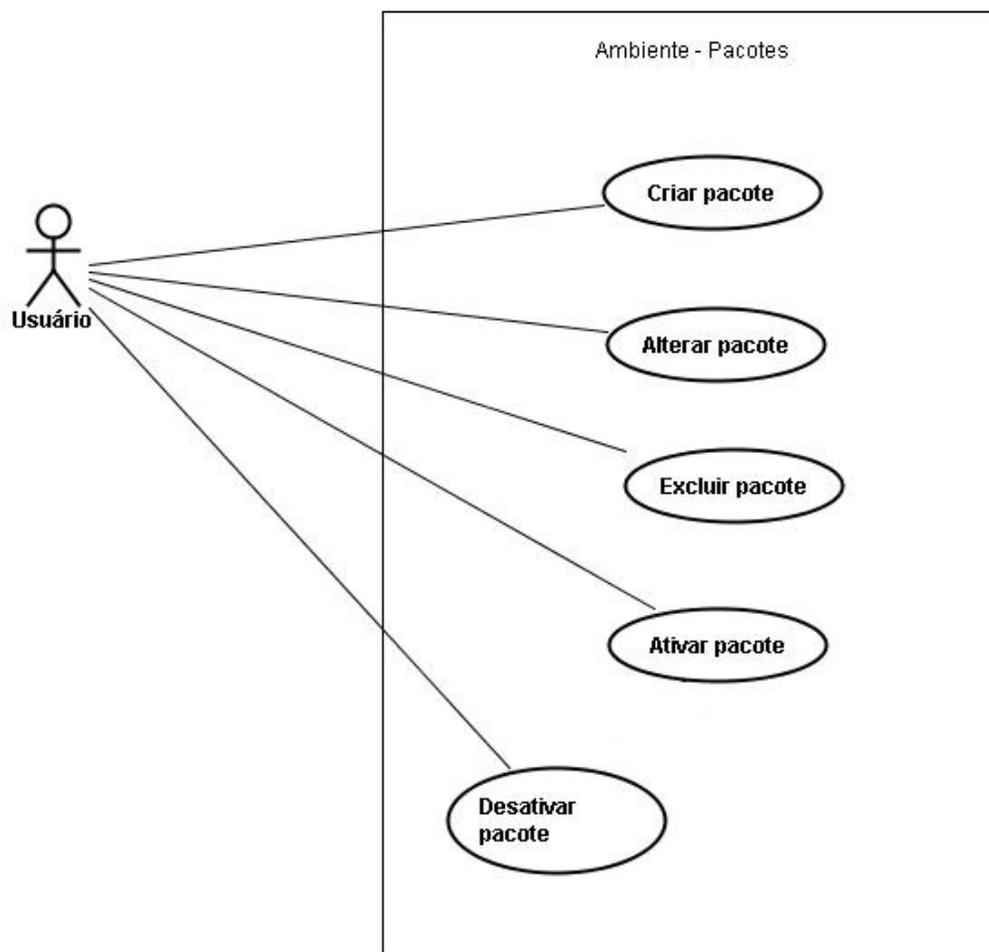


FIGURA 3.7. CASO DE USO: PACOTES

A funcionalidade criar pacote permite ao DBA que um pacote seja criado com um nome escolhido por ele. É possível informar detalhes sobre o pacote, para que se entenda, por exemplo, a razão de o pacote existir. Também é possível especificar uma data para ativação e desativação e, o mais importante, associar as restrições aos pacotes. A funcionalidade alterar pacote permite que as restrições contidas no pacote sejam removidas ou novas restrições sejam inseridas. É possível excluir um pacote e nesse caso, as restrições não serão afetadas. Também é possível desativar ou ativar um pacote manualmente ou, caso o pacote possua uma data de ativação e desativação, este processo será automático. A seguir são apresentados alguns quadros com a finalidade de explicar melhor as funcionalidades desta Figura (Quadros 3.4.18 até 3.4.22):

Quadro 3.4.18 - Caso de Uso : Criar Pacote

Característica	Descrição
Resumo/Objetivo	Criar um pacote
Atores	Usuário
Pré-condição	-
Descrição	Consiste em agrupar restrições simples, elas poderão, com a criação deste pacote, serem controladas por ele, sendo ativadas ou desativadas
Cenários Exceções	-
Pós-Condição	Pacote criado

Quadro 3.4.19 - Caso de Uso : Alterar Pacote

Característica	Descrição
Resumo/Objetivo	Alterar restrições que estão contidas no pacote
Atores	Usuário
Pré-condição	Pacote deve existir
Descrição	Adicionar/Remover restrições do pacote
Cenários Exceções	-
Pós-Condição	Pacote modificado

Quadro 3.4.20 - Caso de Uso : Excluir Pacote

Característica	Descrição
Resumo/Objetivo	Remover o agrupamento de restrições
Atores	Usuário
Pré-condição	Pacote deve existir
Descrição	Remove o agrupamento das restrições, o pacote será apagado
Cenários Exceções	-
Pós-Condição	Pacote excluído

Quadro 3.4.21 - Caso de Uso : Ativar Pacote

Característica	Descrição
Resumo/Objetivo	Ativar todas as restrições contidas no pacote
Atores	Usuário
Pré-condição	Pacote deve existir e estar desativado
Descrição	Ativa todas as restrições do pacote no banco de dados alvo
Cenários Exceções	-
Pós-Condicion	Pacote ativo

Quadro 3.4.22 - Caso de Uso – Desativar Pacote

Característica	Descrição
Resumo/Objetivo	Desativar todas as restrições contidas no pacote
Atores	Usuário
Pré-condição	Pacote deve existir e estar ativado
Descrição	Desativa todas as restrições do pacote no banco de dados alvo
Cenários Exceções	-
Pós-Condicion	Pacote desativado

3.4 PROJETO DO BANCO DE DADOS DA FERRAMENTA

Como comentado na subseção anterior, a ferramenta extrai os metadados de um banco de dados alvo. Essa extração precisa ser armazenada em algum lugar e para isso criou-se um banco de dados que possui a estrutura conforme a Figura 3.8. Nesta Figura, pode-se observar uma tabela chamada *objects* que é a tabela responsável por armazenar todos os dados referentes às restrições, inclusive tabelas e atributos. Ela funciona como uma tabela-mãe, de onde as restrições podem ser especificadas para, por exemplo, uma variante ou para uma invariante. Toda restrição inserida no banco de dados da ferramenta estará contida

nesta tabela. Ela é responsável, por exemplo, para informar o nome do objeto, e seu tipo.

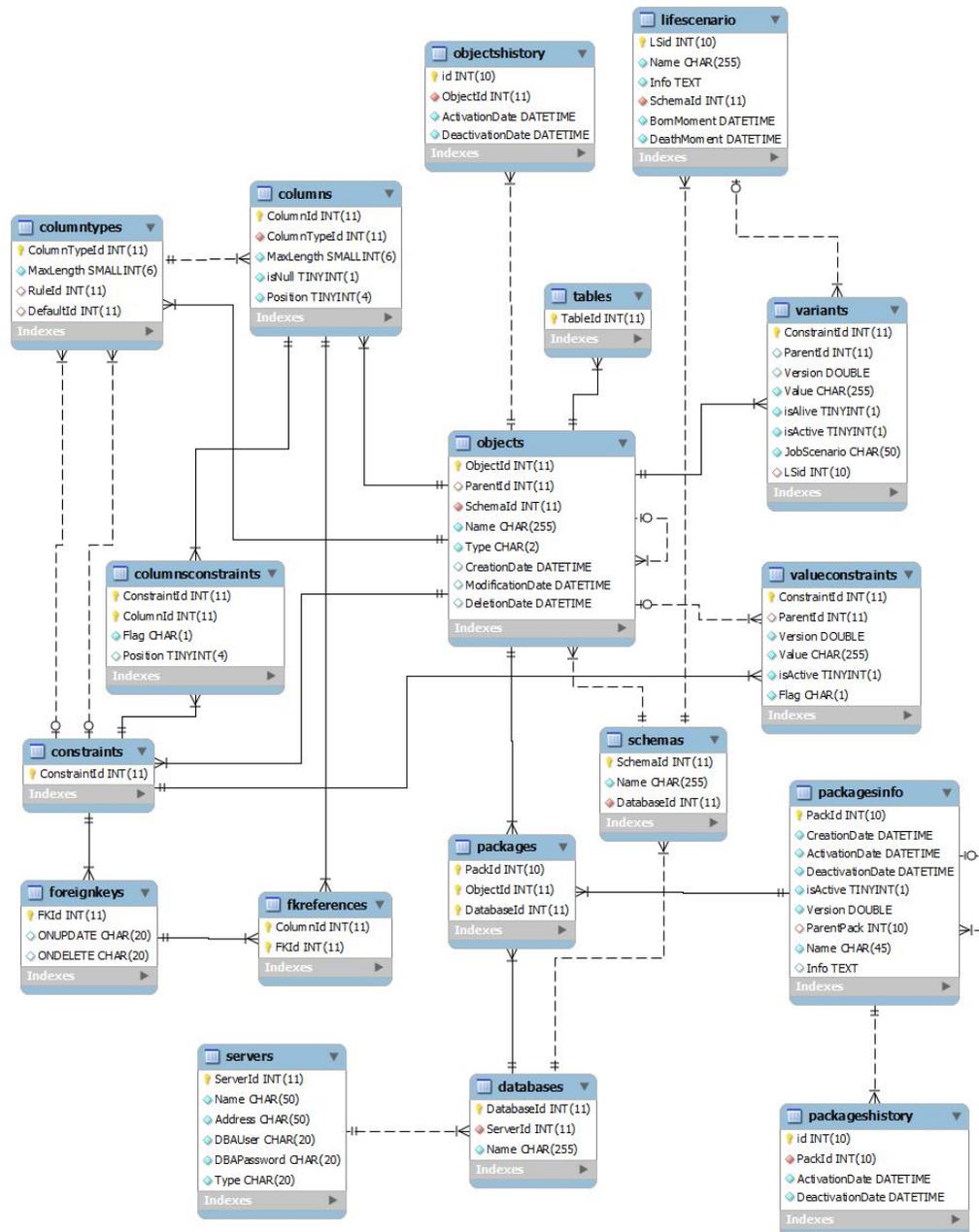


FIGURA 3.8. DIAGRAMA DE CLASSES DO BANCO DE DADOS DA FERRAMENTA

Existe uma pequena tabela chamada *tables*, ela irá conter os objetos que são do tipo tabela. Existe também a tabela *columns* que contém informações

sobre as colunas (atributos) de uma tabela e outras tabelas que especificam a tabela *objects*. A tabela *servers* armazena as informações pertinentes aos servidores criados pelo DBA. Nela serão inseridos, por exemplo, o nome do usuário e a senha para acessar os metadados do SGBD. A tabela *databases* informa os esquemas/bancos de dados que um servidor possui. A tabela *schemas*, apesar de parecer redundante à tabela *databases*, informa se o objeto contido na tabela *objects* pertence a um objeto do sistema (sys) ou do usuário (dbo). A tabela *valueconstraints* é responsável por armazenar as informações das invariantes, por exemplo, o seu valor, se ela está ativa ou qual é a versão da invariante. A tabela *variants*, é semelhante à tabela *valueconstraints*, porém armazena os dados referentes às variantes, por exemplo, o cenário de sobrevivência em que pertence, seu cenário de aplicabilidade e se está ativa ou viva. A tabela *lifescenario* armazena os dados dos cenários de sobrevivência que serão associados a uma variante. O pacote é armazenado na tabela *packagesinfo*, que possui diversas informações, por exemplo, versão, nome, informações extras, etc. A associação das restrições aos pacotes é feita através da tabela *packages* que possui três atributos que constituem a chave primária da tabela. Os três atributos são chaves estrangeiras: *PackId* que indica o pacote envolvido na associação, *ObjectId* que indica o objeto envolvido na associação e o *Databaseld* que indica o banco de dados envolvido na associação. As tabelas *packageshistory* e *objectshistory* são responsáveis por armazenar os dados referentes ao histórico de ativação e desativação de um pacote ou de um objeto. Os dicionários de dados de todas as classes da Figura 3.8 demonstram as suas estruturas, facilitando assim a sua compreensão (Quadros 3.4.23 até 3.4.40):

Quadro 3.4.23 - Dicionário da Classe Servers

Servers				
Campo	Descrição	Tipo	Tamanho	Restrições
ServerId	Identificador do servidor	Integer	-	PK, auto_increment, not null
Name	Nome do servidor	Char	50	Not null, unique
Address	Endereço do servidor	Char	50	Not null
DBAUser	Login do administrador	Char	20	Not null
DBAPassword	Senha do administrador	Char	20	Not null
Type	Tipo do servidor	Char	20	('SqlServer2005' ou 'Oracle10g')

Quadro 3.4.24 - Dicionário da Classe Databases

Databases				
Campo	Descrição	Tipo	Tamanho	Restrições
Databaseld	Identificador do banco de dados	Integer	-	PK, Auto_increment, not null
ServerId	Identificador para o servidor ao qual o banco pertence	Integer	-	FK(Servers), Not null
Name	Nome do banco de dados	Char	255	Not null

Quadro 3.4.25 - Dicionário da Classe Schemas

Schemas				
Campo	Descrição	Tipo	Tamanho	Restrições
Schemald	Identificador do schema	Integer	-	PK, Auto_increment, not null
Name	Nome do schema	Char	255	Not null
Databaseld	Identificador para o banco de dados ao qual o schema pertence	Integer	-	FK(Databases), Not null

Quadro 3.4.26 - Dicionário da Classe Objects

Objects				
Campo	Descrição	Tipo	Tamanho	Restrições
ObjectId	Identificador do objeto	Integer	-	PK, Auto_increment, not null
ParentId	Identificador para objeto pai	Integer	-	FK (Objects)
Schemald	Identificador para o schema ao qual object pertence	Integer	-	FK(Schemas), not null
Name	Nome do objeto	Char	255	Not null
Type	Tipo do objeto	Char	2	('TB','CL','CT','PK','FK','UQ','CK','DF','RL')
CreationDate	Data da criação do objeto	DateTime	-	Null
ModificationDate	Última data em que este objeto foi modificado	DateTime	-	Null
DeletionDate	Data em que este objeto está planejado para ser desativado	DateTime	-	Null

Quadro 3.4.27 - Dicionário da Classe ObjectsHistory

Objects				
Campo	Descrição	Tipo	Tamanho	Restrições
id	Chave primária da tabela	Integer	-	PK, Auto_increment, not null
ObjectId	Identificador para o objeto	Integer	-	FK (PackagesInfo)
ActivationDate	Data em que o objeto foi ativado	DateTime	-	Not Null
DeactivationDate	Data em que o objeto foi desativado	DateTime	-	Not Null

Quadro 3.4.28 - Dicionário da Classe Tables

Tables				
Campo	Descrição	Tipo	Tamanho	Restrições
TableId	Identificador da tabela	Integer	-	PK, FK(Objects), not null

Quadro 3.4.29- Dicionário da Classe Columns

Columns				
Campo	Descrição	Tipo	Tamanho	Restrições
ColumnId	Identificador da coluna	Integer	-	PK, FK(Objects), not null
ColumnTypeId	Identificador do tipo da coluna	Integer	-	FK(ColumnTypes), not null
MaxLength	Tamanho coluna	SmallInt	-	= 0
isNull	Coluna é nula?	Bool	-	Not null
Position	Posição da coluna dentro da tabela	Tinyint	-	Not null

Quadro 3.4.30 - Dicionário da Classe ColumnTypes

ColumnTypes				
Campo	Descrição	Tipo	Tamanho	Restrições
ColumnTypeId	Identificador do tipo da coluna	Integer	-	PK, FK(Objects), not null
MaxLength	Tamanho máximo do tipo	Smallint	-	=0, Not null
RuleId	Identificador da restrição de regra	Integer	-	FK(Constraints)
DefaultId	Identificador da restrição de valor padrão	Integer	-	FK(Constraints)

Quadro 3.4.31 - Dicionário da Classe Constraints

Constraints				
Campo	Descrição	Tipo	Tamanho	Restrições
ConstraintId	Identificador da restrição	Integer	-	PK, FK(Objects), not null

Quadro 3.4.32 - Dicionário da Classe ForeignKeys

ForeignKeys				
Campo	Descrição	Tipo	Tamanho	Restrições
FKId	Identificador da chave estrangeira	Integer	-	PK, FK(Constraints), not null
ONUPDATE	Ação ao atualizar	Char	20	-
ONDELETE	Ação ao deletar	Char	20	-

Quadro 3.4.33 - Dicionário da Classe ValueConstraints

ValueConstraints				
Campo	Descrição	Tipo	Tamanho	Restrições
ConstraintId	Identificador da constraint de valor	Integer	-	PK, FK(Constraints), not null
ParentId	Identificador da versão pai	Integer	-	FK(ValueConstraints)
Version	Numero da versão da restrição	Double	-	=0.0, Not null
Value	Valor da constraint	Char	255	Not null
isActive	Define se constraint está ativo ou não	Bool	-	=1, Not null
Flag	Define estado para ação de atualização. (Deprecado)	Char	1	=N, Not null, ('N','C','U','D')

Quadro 3.4.34 - Dicionário da Classe FKReferences

FKReferences				
Campo	Descrição	Tipo	Tamanho	Restrições
FKId	Identificador da restrição de FK	Integer	-	PK, FK(ForeignKeys), not null
ColumnId	Identificador da coluna	Integer	-	PK, FK(Columns), not null

Quadro 3.4.35 - Dicionário da Classe ColumnsConstraints

ColumnsConstraints				
Campo	Descrição	Tipo	Tamanho	Restrições
ConstraintId	Identificador da restrição	Integer	-	PK, FK(Constraints), not null
ColumnId	Identificador da coluna	Integer	-	PK, FK(Columns), not null
Position	Ordem da restrição em relação à coluna			
Flag	Define estado para ação	Char	1	=N, Not null, ('N','C','U','D')

Quadro 3.4.36 - Dicionário da Classe Packages

Packages				
Campo	Descrição	Tipo	Tamanho	Restrições
PackId	Identificador do pacote	Integer	-	PK, FK(PackagesInfo), not null
ObjectId	Identificador do pacote	Integer	-	PK, FK(Objects), not null
DatabaseId	Identificador do pacote	Integer	-	PK, FK(Databases), not null

Quadro 3.4.37 - Dicionário da Classe PackagesInfo

Objects				
Campo	Descrição	Tipo	Tamanho	Restrições
PackId	Identificador do pacote	Integer	-	PK, Auto_increment, not null
ParentPack	Identificador para o pacote pai	Integer	-	FK (PackagesInfo)
Version	Versão do pacote	Double	-	= 0.0, Not null
isActive	Define se o pacote está ativo ou não	Bool	-	Not null
Name	Nome do pacote	Char	255	Not null
Info	Informações sobre o pacote	Text	-	Null
CreationDate	Data da criação do pacote	DateTime	-	Not Null
ActivationDate	Data em que o pacote está planejado para ser ativado	DateTime	-	Not Null
DeactivationDate	Data em que o pacote está planejado para ser desativado	DateTime	-	Not Null

Quadro 3.4.38 - Dicionário da Classe LifeScenario

Objects				
Campo	Descrição	Tipo	Tamanho	Restrições
LSId	Identificador do cenário de sobrevivência	Integer	-	PK, Auto_increment, not null
Name	Nome do cenário de sobrevivência	Char	255	Not null
Schemald	Identificador para o esquema que o cenário pertence	Integer	-	FK (Schemas)
Info	Informações sobre o cenário	Text	-	Null
BornMoment	Momento em que o cenário é aberto	DateTime	-	Not Null
DeathMoment	Momento em que o cenário é fechado	DateTime	-	Not Null

Quadro 3.4.39 - Dicionário da Classe Variants

Variants				
Campo	Descrição	Tipo	Tamanho	Restrições
ConstraintId	Identificador da variante	Integer	-	PK, auto_increment, not null
ParentId	Identificador da variante pai	Integer	-	FK(Variants)
Version	Número da versão da variante	Double	-	=0.0, Not null
Value	Valor da restrição	Char	255	Not null
isAlive	Define se a variante está viva ou não	Bool	-	=0, Not null
isActive	Define se a variante está ativa ou não	Bool	-	=0, Not null
JobScenario	Define o cenário de aplicabilidade da variante	Char	50	Not Null
LSid	Identificador do cenário de sobrevivência ao qual a variante pertence	Integer	-	FK(lifeScenario)

Quadro 3.4.40 - Dicionário da Classe PackagesHistory

Objects				
Campo	Descrição	Tipo	Tamanho	Restrições
id	Chave primária da tabela	Integer	-	PK, Auto_increment, not null
PackId	Identificador para o pacote	Integer	-	FK (PackagesInfo)
ActivationDate	Data em que o pacote foi ativado	DateTime	-	Not Null
DeactivationDate	Data em que o pacote foi desativado	DateTime	-	Not Null

3.5 PROJETO DA INTERFACE

Nesta seção são apresentadas algumas interfaces da ferramenta e suas partes serão descritas para que se possa compreendê-la.

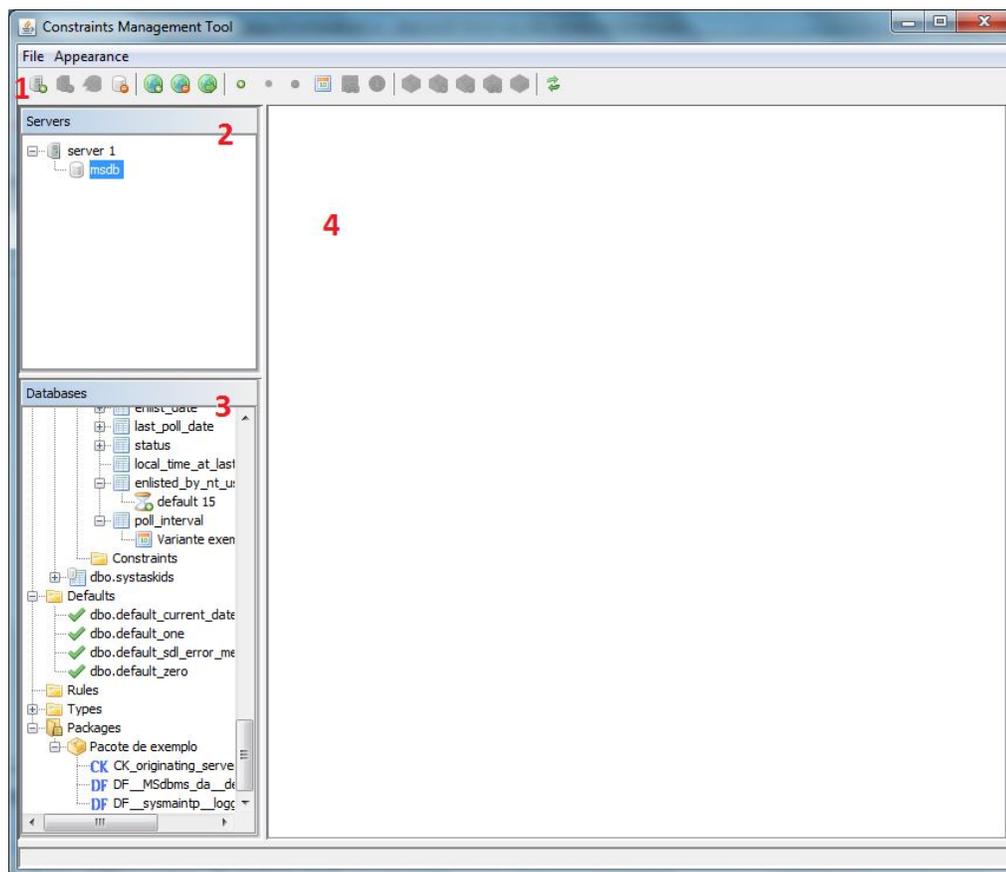


FIGURA 3.9. INTERFACE GRÁFICA DO SISTEMA

Para a construção da interface gráfica da ferramenta alguns requisitos foram pré-definidos, como a facilidade de uso, a possibilidade de extensão e que fosse intuitiva para o usuário. A interface gráfica representada na Figura 3.9 pode ser dividida basicamente em quatro áreas e a parte inferior da tela apresenta ao usuário as informações relevantes sobre os botões.

A primeira área é a barra de atividades onde estão os botões que acionam as funcionalidades do sistema, os ícones dos botões são intuitivos. A Figura 3.10 representa esta área em uma escala maior.



FIGURA 3.10. ÁREA 1: BARRA DE FUNCIONALIDADES

Os botões da esquerda para direita são:

- Registrar Servidor
- Excluir Servidor
- Extrair Banco de Dados
- Excluir Banco de Dados
- Adicionar cenário de sobrevivência
- Excluir cenário de sobrevivência
- Associar um cenário de sobrevivência às variantes
- Adicionar Restrição
- Alterar Restrição (criará uma nova versão)
- Desativar a restrição
- Criar variante
- Criar uma versão de uma variante
- Exibir informações da restrição ou da variante selecionada
- Criar um novo pacote
- Ativar pacote
- Desativar pacote
- Criar versão de pacote
- Exibir informações do pacote selecionado
- Atualizar interface

A segunda área, representada pela Figura 3.11, apresenta os servidores registrados no sistema e seus bancos de dados extraídos. Empregou-se uma distribuição na forma de árvore para facilitar a visualização e sua hierarquia. Ao selecionar um nó específico desta área, os botões da área 1 serão habilitados ou desabilitados.



FIGURA 3.11. ÁREA 2: ÁRVORE DE SERVIDORES E BANCO DE DADOS

A terceira área, conforme mostra a Figura 3.12, apresenta a estrutura de um banco de dados extraído, uma vez que o mesmo tenha sido selecionado na área 2. Também foi empregada uma árvore para facilitar a visualização da estrutura do banco, pois existem muitos dados a serem mostrados. Cada nó dentro da árvore tem um ícone específico para facilitar na identificação de seus tipos. Todo o banco de dados extraído possuirá as cinco seguintes pastas:

- *Tables*: Onde são mostradas todas as tabelas e suas colunas que estão contidas no banco de dados extraído, assim como as suas restrições e variantes;
- *Defaults*: Aonde são mostradas as restrições do tipo *default* do banco de dados;
- *Rules*: Onde seriam mostradas as restrições *rule*, porém, ela não foi implementada, vide subseção 3.4;
- *Types*: Onde são mostrados os tipos de dados existentes no banco de dados;
- *Packages*: Onde são mostrados os pacotes existentes no banco de dados e as restrições contidas em cada pacote.

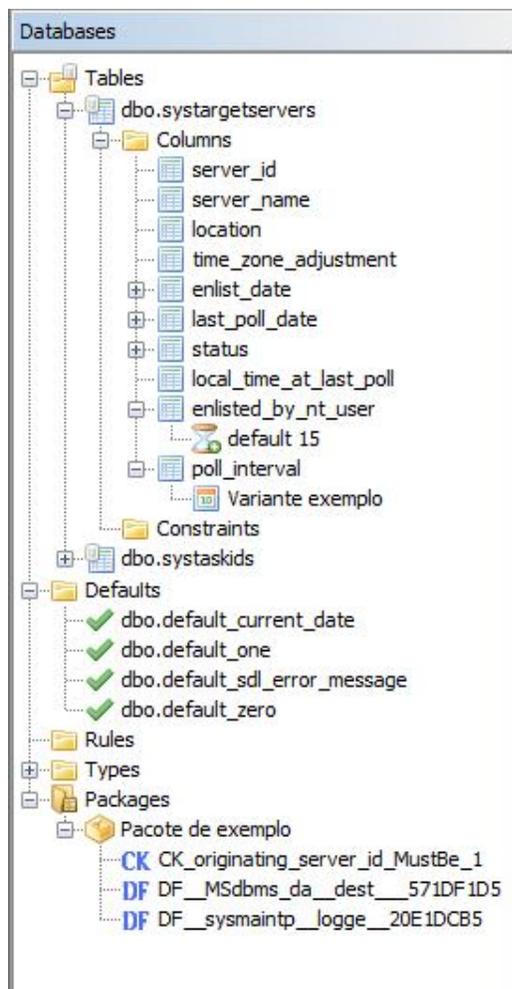


FIGURA 3.12. ÁREA 3: ÁRVORE DE ELEMENTOS DO BANCO DE DADOS

A quarta área é uma área auxiliar da ferramenta e é utilizada para apresentar as janelas de funcionalidades, por exemplo, a janela de seleção de um banco de dados a ser extraído, conforme mostra a Figura 3.13.

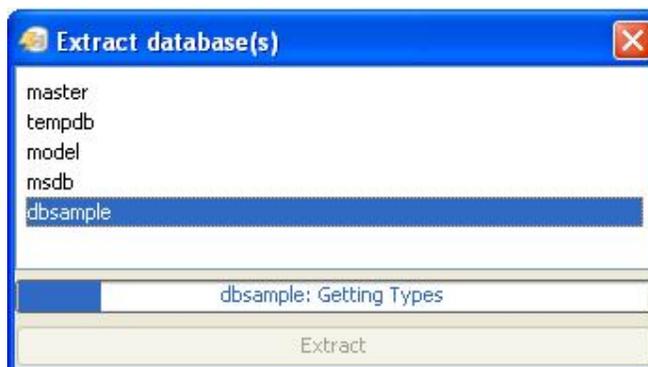


FIGURA 3.13. ÁREA 4: JANELA DE EXTRAÇÃO DE BANCO DE DADOS

3.6 CONSIDERAÇÕES FINAIS

Associar o tempo a uma restrição invariante aparenta ser uma tarefa simples, mas é uma tarefa ligeiramente complexa e trabalhosa. Baseando-se neste trabalho, além do estudo feito para a sua conclusão, foi necessário criar um banco de dados com diversas tabelas para armazenar as mais variadas informações referentes a uma restrição, tais como, o esquema em que a restrição pertence, o seu servidor, qual o tipo da restrição, se está associada a algum pacote, além de, é claro, informar os momentos em que a ativação e desativação devem ser feitas.

A construção dessa ferramenta é um primeiro passo na tentativa de criar uma ferramenta capaz de facilitar as tarefas do DBA com relação ao gerenciamento de restrições. O projeto não tem o propósito de apenas contemplar o Microsoft SQL Server, espera-se que futuramente outros SGBDs possam ser incluídos na ferramenta.

4 GERENCIAMENTO DE CENÁRIOS E VARIANTES

4.1 CONSIDERAÇÕES INICIAIS

A tarefa de associar um momento de ativação e desativação de uma invariante é simples, afinal é necessário apenas informar dois momentos de tempo, um para a ativação e outro para a desativação. Com isso, uma *thread* da ferramenta fica em *loop* infinito verificando a hora atual do sistema. Se o momento de ativação ou desativação for alcançado, a ferramenta irá ativar ou desativar a restrição.

Essa associação de tempo básica, descrita anteriormente com apenas dois momentos, é um diferencial somente para as restrições invariantes com o propósito de fazer com que a ferramenta, além de lidar com o tempo das restrições variantes, consiga também ativar ou desativar automaticamente uma invariante. Ao decorrer desta seção será demonstrado, na prática, como foi modelado o tempo para as restrições variantes.

4.2 CRIAÇÃO DE CENÁRIOS DE SOBREVIVÊNCIA

Como já amplamente discutido no capítulo dois, os cenários de sobrevivência irão mapear o intervalo de tempo em que as restrições variantes poderão ser ativadas e desativadas. Com isso, as restrições que estão contidas neles ganharão vida, uma vez que para as variantes serem ativadas elas precisam de um cenário de sobrevivência. Se uma variante não for associada a um cenário de sobrevivência criado pelo DBA que está usando a ferramenta, esta variante nunca será ativada. O cenário de sobrevivência é o responsável por “acordar” e tornar as variantes “vivas”.

The image shows a software dialog box titled "Add a Life Scenario". It is divided into several sections. At the top, under "Life Scenario Info", there is a "Name" text box containing "Exemplar para tese" and an "Info" text area containing the text "Este cenário de sobrevivência foi criado para ser apresentado na dissertação.". Below this, the "Scenario Type" section has three radio buttons: "Normal" (which is selected), "Without opening", and "Without closing". To the right of these are two columns for "Opening Moment" and "Closing Moment", each containing "Date" and "Time" input fields with placeholders "yyyy/mm/dd" and "hh:mm:ss" respectively. At the bottom of the dialog is a "Create Life Scenario" button.

FIGURA 4.1. INTERFACE DA FERRAMENTA PARA CRIAR UM CENÁRIO DE SOBREVIVÊNCIA

Como pode ser observado na Figura 4.1, para se criar um cenário são necessárias três informações:

- Informar o nome que será atribuído ao cenário;
- Informar detalhes sobre o cenário. Esta é uma tarefa opcional, porém fica mais claro para distinguir entre um cenário e outro;
- Informar o tipo do cenário. Na figura fica claro que existem três tipos de cenários de sobrevivência; normais; sem abertura; sem fechamento.

Assim que um cenário de sobrevivência é criado, ele ficará disponível para que variantes sejam associadas a ele. Os três tipos de cenários de sobrevivência existentes são detalhados nas próximas Seções.

4.2.1 CENÁRIO DE SOBREVIVÊNCIA NORMAL

O cenário de sobrevivência normal possui dois momentos de tempo bem definidos, um para o seu momento de abertura, isto é, quando irá “acordar” e o outro para o seu momento de fechamento, quando irá morrer e com isso desativar todas as variantes contidas nele.

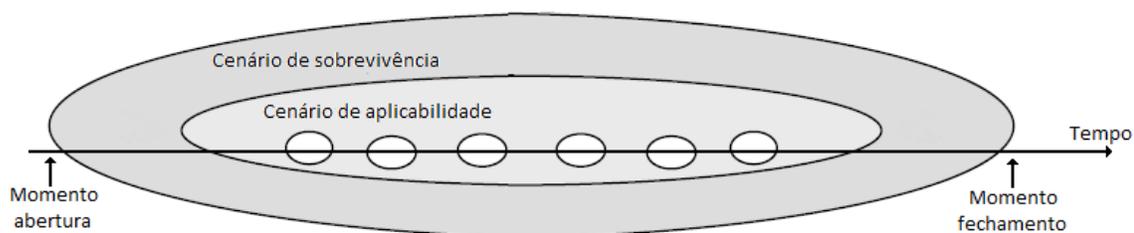


FIGURA 4.2. CENÁRIO DE SOBREVIVÊNCIA NORMAL

O uso deste tipo de cenário é bem aplicado quando se tem a certeza do momento em que ele deve ser ativado, e também o momento em que ele deve ser fechado. Como exemplo, pode-se citar uma promoção que dure o mês de junho inteiro e dará desconto de 10% em todas as compras.

4.2.2 CENÁRIO DE SOBREVIVÊNCIA SEM ABERTURA

O cenário de sobrevivência sem abertura define apenas um momento de fechamento para o cenário, portanto o cenário estará sempre vivo enquanto o seu momento de fechamento for menor que o momento atual do sistema. Por exemplo, se são meio dia do dia 19 de fevereiro de 2010 e o momento de fechamento de um cenário de sobrevivência sem abertura for apenas no natal de 2020, este cenário de sobrevivência estará ativo e suas variantes estarão vivas.

Infelizmente não podemos informar para o banco de dados que este é um cenário sem momento de abertura, ele não iria entender, portanto informamos uma data bem antiga que funcionará como um identificador de um cenário de

sobrevivência sem abertura. Com isso, o momento de abertura que representará um cenário de sobrevivência sem abertura é o dia 01/01/1000 e o horário é 01h01m01s.

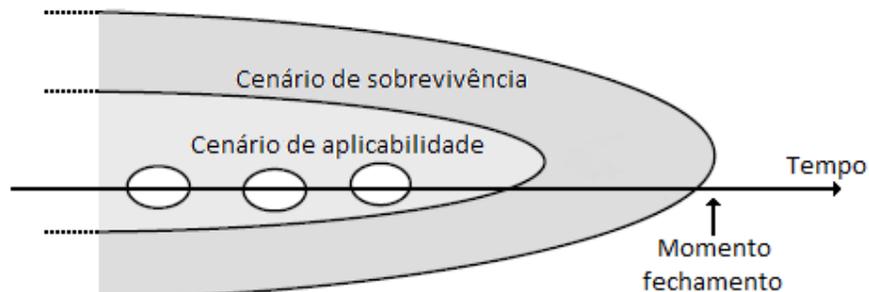


FIGURA 4.3. CENÁRIO DE SOBREVIVÊNCIA SEM ABERTURA

O uso deste tipo de cenário é bem aplicado quando se sabe que ele deve estar ativo até um momento específico, não importa quando ele foi ativado, só se tem a certeza do momento em que ele deve ser desativado.

4.2.3 CENÁRIO DE SOBREVIVÊNCIA SEM FECHAMENTO

Este cenário de sobrevivência, ao contrário do anterior, não possui um momento de fechamento, portanto ele estará ativo sempre que o momento de abertura dele for menor ou igual ao que o momento atual do sistema. Por exemplo, se for considerado este exato momento de tempo e o momento de abertura de um cenário de sobrevivência sem fechamento for o réveillon de 2008, este cenário de vida estará ativo e todas as restrições variantes contidas nele estarão vivas.

O mesmo problema do cenário de sobrevivência sem abertura acontece aqui, portanto uma data bem distante para identificar um cenário de sobrevivência sem fechamento precisou ser criada. O momento escolhido foi 31/12/9999 às 12h12m12s. Esta não seria a solução ideal, pois abre uma brecha para o “bug do milênio” ocorrer, porém, isso só irá acontecer daqui a praticamente oito mil anos.

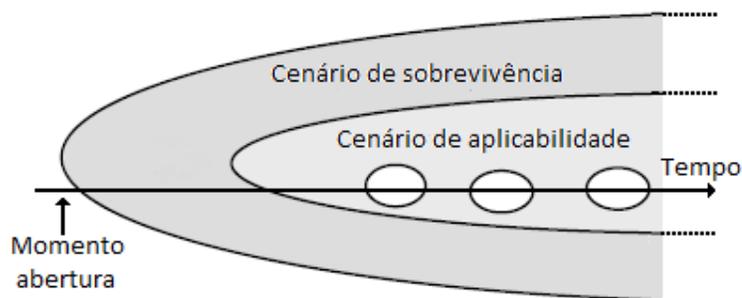


FIGURA 4.4. CENÁRIO DE SOBREVIVÊNCIA SEM FECHAMENTO

O uso deste tipo de cenário é bem aplicado para quando se tem a certeza do momento em que ele deve ser ativado, mas não se sabe ao certo quando ele deve ser fechado ou que pelo menos deve perdurar por um bom tempo. A única maneira de desativar um cenário de sobrevivência deste tipo no futuro, será apagando o cenário do banco de dados. A ferramenta possui um ícone para apagar cenários.

4.2.4 CENÁRIO DE SOBREVIVÊNCIA GLOBAL

O cenário de sobrevivência global é um cenário que não é criado pelo DBA e pode ser compreendido como uma junção de dois cenários anteriores, o cenário sem abertura e o cenário sem fechamento. É como se a ferramenta considerasse este cenário nativo. Na teoria, o cenário de sobrevivência global armazena todas as variantes criadas pela ferramenta e que ainda não foram associadas a algum cenário criado pelo DBA, isso ficará claro na Figura 4.16 que mostra três variantes; uma foi associada a um cenário e as outras duas, que ainda não foram associadas a um cenário, estão associadas ao cenário global. O cenário global, no caso desta ferramenta, é o responsável por alocar todas as restrições ociosas, em outras palavras: todas as restrições que não estão sendo utilizadas e, portanto, “mortas”.

A questão de cenários globais foi pouco explorada pela ferramenta, já que não era necessário para o propósito deste trabalho. No futuro poderá ser feito um estudo mais detalhado sobre cenários globais para que eles possam ter mais

funcionalidades na ferramenta, como, por exemplo, criar dois cenários globais, um que contém variantes que devem estar “vivas” e outro que deve conter variantes que estão “mortas”. Este último é o que está sendo feito com a ferramenta.

Internamente, como se faz uso de apenas um cenário global, quando a coluna *LSid* da tabela *variants*, que é uma chave estrangeira indicando o cenário de sobrevivência ao qual a variante pertence, for nulo, significa que a variante está associada ao cenário global, caso contrário, a variante está associada a algum cenário de sobrevivência criado pelo DBA.

4.3 CRIAÇÃO DE VARIANTES

Como previamente discutido no capítulo 2, os cenários de sobrevivência podem ser considerados o escopo de vida das restrições variantes. Se não estiverem associadas a algum cenário, as variantes apenas existirão, elas não terão um propósito e não serão ativadas.

As variantes nada mais são do que as famosas restrições invariantes, bastante conhecidas na área de banco de dados, porém com o detalhe de que agora, estão ligadas ao tempo (CAMOLESI JR., 2006). O fato de estarem ligadas ao tempo permite que elas variem, o que soaria um pouco estranho “invariantes que variam”. Com isso, Camolesi resolveu modificar a nomenclatura para variantes.

No caso do Microsoft SQL Server, as variantes possuem dois tipos, ou são uma restrição *CHECK* ou são do tipo *DEFAULT*. Essencialmente, uma variante do tipo *check* pode estar associada a uma tabela inteira, checando diversos atributos (colunas) dessa tabela, ou estar associada diretamente a um atributo. Portanto, a variante do tipo *check* verifica situações, tais como: um indivíduo vai se cadastrar em um site do governo para tirar seu passaporte via internet, seu cadastro só será efetivado no banco de dados e posteriormente enviado para avaliação, caso ele possua nacionalidade brasileira. As variantes do tipo *default* também podem ser associadas a uma coluna, informando qual é o valor padrão para esta coluna (atributo) da tabela, ou diretamente a uma tabela. Ao associar este tipo de variante a uma tabela deve-se informar um valor padrão e um nome. Por exemplo, é desejado

criar um valor padrão para o piso salarial de uma empresa, portanto sempre que algum funcionário for cadastrado e não for repassado qual o valor de seu salário, o piso salarial será informado. Isso pode ser feito associando o valor do piso salarial a uma restrição *default*, supondo que o valor do piso salarial seja R\$ 1000,00, então a restrição chamada de “pisoSalarial” será agora associada ao valor padrão 1000. Após a criação desta restrição do tipo *default* para a tabela, basta associá-la ao campo do salário do empregado. Maiores detalhes sobre restrições podem ser obtidos na seção 2 ou nas referências bibliográficas.

A ferramenta criada neste trabalho possibilita duas formas de criação de variantes, a criação de uma variante “normal” e a criação de uma versão de uma variante.

The image shows a software interface window titled "Add New Variant". It is divided into two main sections: "STEP 1 - Variant's Info" and "STEP 2 - Variant's Moments".

STEP 1 - Variant's Info:

- Contains two text input fields: "Name" and "Value".
- Has a "Constraint type:" label with two radio buttons: "Check" and "Default". The "Default" radio button is selected.
- Includes a button labeled "Activate Step 2".

STEP 2 - Variant's Moments:

- Has a "Start Moment:" label.
- Below it are six checkboxes: "Year", "Month", "Day", "Hours", "Minutes", and "Seconds". Each checkbox is followed by a text input field.
- Below the checkboxes is the question "Is this a variant with a duration time?" with two radio buttons: "Yes" and "No". The "No" radio button is selected.
- Below the question is a "Finish Moment:" label.
- Below the label are six checkboxes: "Year", "Month", "Day", "Hours", "Minutes", and "Seconds". Each checkbox is followed by a text input field.
- At the bottom of the window is a button labeled "Create Variant".

FIGURA 4.5. INTERFACE DA FERRAMENTA PARA ADICIONAR NOVA VARIANTE

Após a escolha da coluna ou da tabela em que a variante deve ser criada, a interface da ferramenta disponibiliza quatro itens que devem ser preenchidos; o nome da variante; o valor da variante; o tipo da variante e, o mais importante, a associação do tempo à variante.

4.3.1 ASSOCIANDO O TEMPO À VARIANTE

Como opção do projeto, foi adotado um padrão de representação de tempo baseado no padrão ISO (ISO, 2000), conforme apresentado no capítulo 2. Esta escolha se deve à preocupação por uma representação mais uniforme do tempo, independente de SGBD. Assim, houve a necessidade de dar capacidade à ferramenta de reconhecer uma data nos formatos apresentados na subseção 2.3. Por exemplo, se é desejado informar que uma variante qualquer deve ser ativada do dia 1 ao dia 10 de cada mês, o cenário de aplicabilidade desta variante será: [01,10,P2D].

Inicialmente foi criada uma expressão regular para fazer a verificação do que o DBA digitava, portanto se o DBA digitasse [102010,P2M2DT2H] a ferramenta iria validar essa expressão e entender que ele estava referenciando o momento de tempo referente ao dia 20 do mês de outubro às 10 horas. No entanto, após alguns testes, foi decidido abolir a expressão regular e deixar esta tarefa, de gerar o momento de tempo seguindo este modelo, para a ferramenta. Repare que na Figura 4.6, não existe espaço para entrar com a expressão, o usuário seleciona se deseja inserir um ano, um dia, um segundo, etc e depois digita o valor de cada campo.

Existem dois tipos de cenários de aplicabilidade que podem ser gerados por esta ferramenta, um modelo mais simples, que informa um momento apenas, por exemplo [10,P2M] que indica o mês 10, e um modelo mais avançado, que informa dois momentos de tempo, por exemplo [10,11,P2M] que indica que este cenário de aplicabilidade deve ser iniciado no mês 10 e fechado no mês 11. Perceba que o ano é opcional, quem define o ano ou os anos em que esta variante será ativada/desativada é o cenário de sobrevivência que for associado a esta variante.

O DBA também pode escolher se deseja inserir um tempo de duração para a variante. Esse tempo de duração é obtido informando o momento para desativar essa restrição, porém esse momento não é a desativação definitiva, é apenas um momento em que a restrição será desativada. Dependendo da sua definição, a variante poderá ser ativada novamente ou não. Por exemplo, a expressão [06,18,T2H] informa que o momento de início da variante será às 06

horas e o seu momento final será às 18 horas, porém os dias que isto será feito depende do cenário de sobrevivência desta variante. Se a variante pertencer a um cenário que tem como escopo de vida o ano de 2010, do começo ao fim, então esta restrição será ativada todo dia a partir das 6 horas e desativada no mesmo dia às 18 horas por 364 dias.

The screenshot shows a software interface for creating a new variant. It is titled 'Add New Variant' and contains two main sections:

- STEP 1 - Variant's Info:**
 - Name: Maioridade
 - Value: (18)
 - Constraint type: Check Default
 - Button: Activate Step 2
- STEP 2 - Variant's Moments:**
 - Start Moment:
 - Year: Month: Day: Hours: Minutes: Seconds:
 - Input fields: [] 10 [] 20 [] 07 [] []
 - Is this a variant with a duration time? Yes No
 - Finish Moment:
 - Year: [] Month: [] Day: [] Hours: [] Minutes: [] Seconds: []
 - Button: Create Variant

FIGURA 4.6. CRIAÇÃO DA VARIANTE CHAMADA DE MAIORIDADE

4.3.2 ATIVAÇÃO E DESATIVAÇÃO DE UMA VARIANTE

Quando uma variante é criada, ela fica disponível para ser associada a algum cenário, assim que ela é associada a ele, se este cenário estiver vivo, a variante será prontamente acordada e irá esperar o seu cenário de aplicabilidade chegar para ser ativada. O mesmo acontece para desativar a variante, caso ela esteja ativa e possua um momento de desativação, assim que este momento chegar, ela será desativada.

Considere, como ilustração, a criação de uma variante chamada “Variante de teste dissertação” com o seguinte cenário de aplicabilidade [44,46,T2M], portanto o momento de ativação é aos 44 minutos e o momento de desativação é aos 46 minutos. Com isso, essa variante será ativada aos 44 minutos de cada hora e desativada aos 46 minutos de cada hora, caso pertença a algum cenário de sobrevivência que esteja vivo. Esta é uma variante do tipo *check*, e faz a verificação se a coluna (atributo) id da tabela é maior que 15.

A Figura 4.7 demonstra essa variante sendo ativada e a Figura 4.8 demonstra a mesma chegando ao seu momento final, aos 46 minutos. Neste exemplo, como pode ser observado no horário que está na barra de tarefas, ela foi ativada às 15:44 e desativada às 15:46, portanto, o próximo momento em que esta restrição estará ativa no banco de dados será das 16:44 às 16:46. Ela será ativada novamente neste horário, pois o cenário de sobrevivência a qual ela pertence só será fechado em 2011.

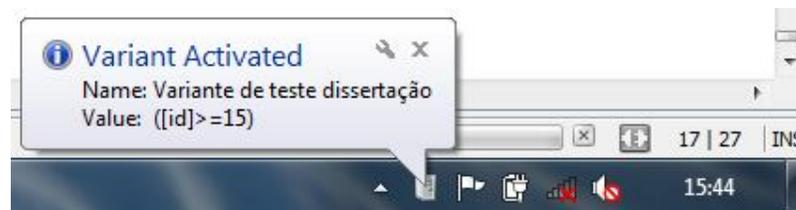


FIGURA 4.7. ATIVANDO UMA VARIANTE

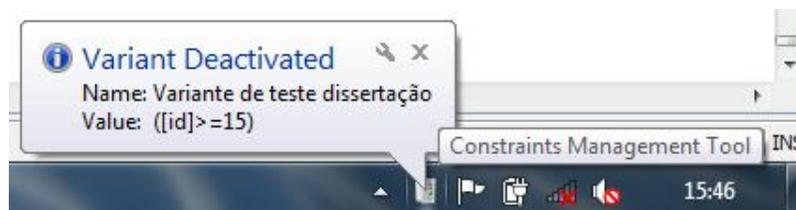


FIGURA 4.8. DESATIVANDO UMA VARIANTE

Essa é uma das vantagens de se utilizar a ferramenta, pois uma vez que a variante foi bem definida, bem como o cenário de sobrevivência em que ela pertence, o DBA não tem a necessidade de se lembrar de ativar ou desativar as restrições, a ferramenta será a responsável por esta tarefa.

4.4 CENÁRIOS DE SOBREVIVÊNCIA SOBREPOSTOS E CONFLITOS DE VARIANTES

Mais do que a responsabilidade de acordar e tornar as variantes vivas, possibilitando que elas sejam ativadas, os cenários de sobrevivência permitem, de uma forma extremamente mais clara, observar quais as restrições que podem ser associadas a um determinado cenário de sobrevivência ou não. Isso ocorre porque se consegue enxergar claramente o período de tempo que cada cenário possui, com o seu momento de abertura e momento de fechamento, ao passo que analisar o cenário de aplicabilidade, que é o período de tempo que a variante será ativada, seria uma tarefa bem mais trabalhosa, mais detalhes sobre o cenário de aplicabilidade serão expostos na subseção 4.3.

Não há problema algum de existir cenários de sobrevivência sobrepostos. A criação deles é permitida, pois pode existir no mundo real. Por exemplo, se um cenário de sobrevivência dura o ano de 2010 inteiro e se a criação de cenários de sobrevivência sobrepostos não fosse permitida, o ano de 2010 possuiria apenas um cenário de sobrevivência, o que não faz sentido. É preciso criar variantes com diferentes escopos, e a impossibilidade de criar cenários sobrepostos seria um grande limitador ao uso das variantes.

O problema com cenários de sobrevivência sobrepostos reside na associação das mesmas variantes ou de suas versões por estes cenários sobrepostos e isso não pode acontecer. Portanto, se uma variante C está sendo utilizada pelo cenário de sobrevivência B, que por sua vez é sobreposto ao cenário de sobrevivência A, não poderia ser permitido que a variante C seja associada ao cenário A e ao cenário B ao mesmo tempo. Se existisse alguma versão da variante C, por exemplo, C1 e C2, essas duas versões também não poderiam ser associadas aos dois cenários simultaneamente. O fato de associar uma dessas variantes, pegando como exemplo C1, ao cenário de sobrevivência B, irá bloquear a possibilidade de associar as variantes C, C1 e C2 ao cenário de sobrevivência A. A razão desta restrição existir será elucidada com o seguinte exemplo: Imagine que o cenário B anterior, o cenário sobreposto ao cenário A, exista do dia primeiro de julho 2010 até o dia 31 de julho de 2010. A criação deste cenário foi feita especialmente para ativar a variante C1, que é uma versão da variante C. A variante C por sua vez é uma restrição do tipo *default* e está associada ao campo desconto de uma tabela

chamada Compra. Portanto, a variante C é responsável por atribuir uma porcentagem de desconto a uma compra. A variante C1 dará 20% de desconto para as compras e está associada ao cenário B, com isso todas as compras feitas no mês de julho terão 20% de desconto.

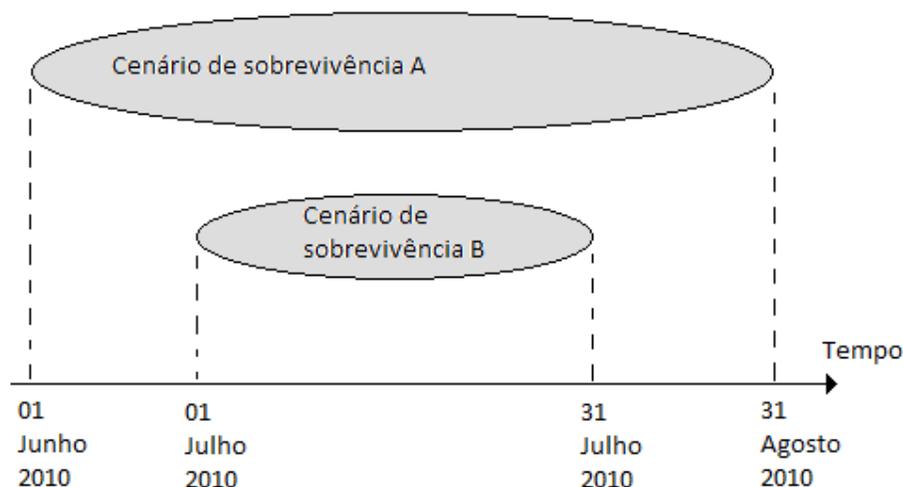


FIGURA 4.9. EXEMPLO DE CENÁRIOS SOBREPOSTOS

Esse desconto de 20% será efetivado sem problemas, pois até então nenhuma restrição variante está fazendo o uso desta variante. Agora imagine o que aconteceria se fosse permitido associar a mesma variante ou uma versão dela a um cenário sobreposto, por exemplo, associar a variante C2, que dá um desconto de 10% nas compras, ao cenário de sobrevivência A. Existiria neste momento um conflito de restrições, pois sobre um mesmo período de tempo (01 de julho até 31 de julho) duas variantes, C1 e C2, estarão tentando ser ativadas e como são referentes a uma mesma restrição, mesmo sendo versões diferentes, demonstra um erro lógico. Para ilustrar este exemplo na realidade, aqui será apresentado em forma de figuras o exemplo previamente discutido.

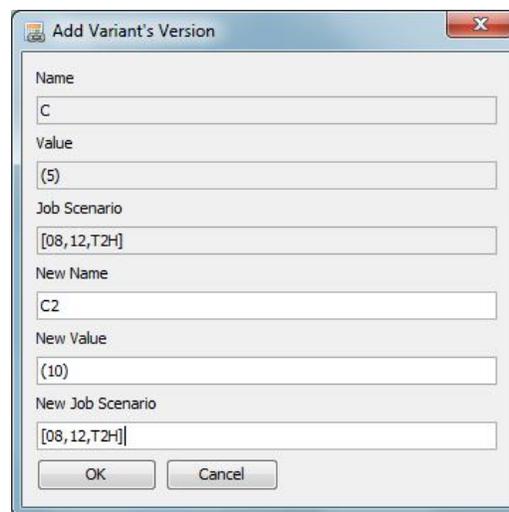
O primeiro passo foi criar os cenários de sobrevivência A e B, representado na Figura 4.10. Com a criação destes dois cenários feita, o próximo passo é criar as variantes, porém nada impede das variantes serem criadas antes e os cenários de sobrevivência depois. Como já comentado, as variantes só terão efeito quando estiverem associadas a um cenário, e a função dos cenários é, basicamente, fornecer um escopo de vida para as variantes contidas nele.

FIGURA 4.10. CRIAÇÃO DOS CENÁRIOS DE SOBREVIVÊNCIA B E A

No exemplo anterior, a variante C1 era para ser ativada do dia 01 de julho ao dia 31 de julho, porém aqui neste exemplo, ela foi ligeiramente modificada, o desconto será de 20%, mas só das 08 horas da manhã até as 12 horas. A variante C1 é filha da variante C, portanto se faz necessária a criação de variante C antes da variante C1. Para ilustração, a variante C dará um desconto de 5%, por isso seu valor padrão é 5, isso pode ser observado na Figura 4.11.

FIGURA 4.11. CRIAÇÃO DA VARIANTE C

O *job scenario* (cenário de aplicabilidade) da variante possui um momento de início, que são às 8 horas, e um momento de término, que são às 12 horas. Como pode ser observado, o cenário de aplicabilidade provê uma maior variância da restrição, já que apenas inserindo esses dois momentos, a variante será sempre ativada às 8 horas e desativada às 12 horas. Porém, só enquanto perdurar seu escopo de vida, isto é, seu cenário de sobrevivência. Como pode se observar nas Figuras 4.12 e 4.13, os três primeiros campos são referentes à variante pai, que é a variante C e os três restantes e últimos campos são referentes à nova versão que será criada, portanto C2 e C1.

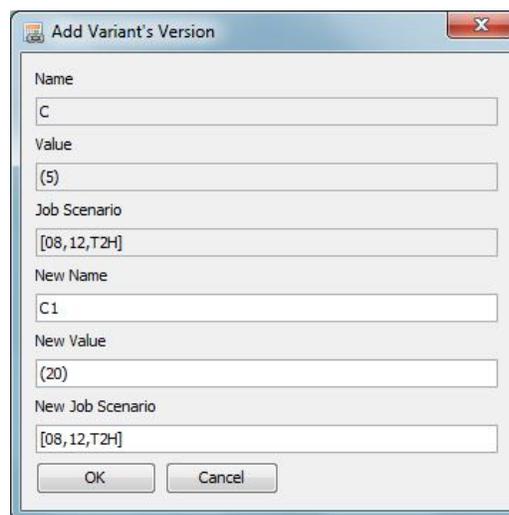


The screenshot shows a dialog box titled "Add Variant's Version" with a close button (X) in the top right corner. The dialog contains the following fields:

- Name: C
- Value: (5)
- Job Scenario: [08,12,T2H]
- New Name: C2
- New Value: (10)
- New Job Scenario: [08,12,T2H]

At the bottom of the dialog are two buttons: "OK" and "Cancel".

FIGURA 4.12. CRIANDO A VARIANTE C2 COMO UMA VERSÃO DA VARIANTE C



The screenshot shows a dialog box titled "Add Variant's Version" with a close button (X) in the top right corner. The dialog contains the following fields:

- Name: C
- Value: (5)
- Job Scenario: [08,12,T2H]
- New Name: C1
- New Value: (20)
- New Job Scenario: [08,12,T2H]

At the bottom of the dialog are two buttons: "OK" and "Cancel".

FIGURA 4.13. CRIANDO A VARIANTE C1 COMO UMA VERSÃO DA VARIANTE C

Depois da criação das variantes C, C1 e C2, a ferramenta possibilita observar a estrutura da árvore de versão criada, que é a demonstrada na Figura 4.14. O formato “-v0.0” indica a versão, o primeiro número é o nível da árvore, e o segundo é o filho, portanto v0.0 é a raiz, v1.0 é o primeiro filho no nível 1 e assim por diante. O “X” em vermelho indica que nenhuma das variantes está ativada no SGBD. A razão é óbvia: nenhuma delas, por enquanto, foi associada a algum cenário de sobrevivência criado pelo DBA.

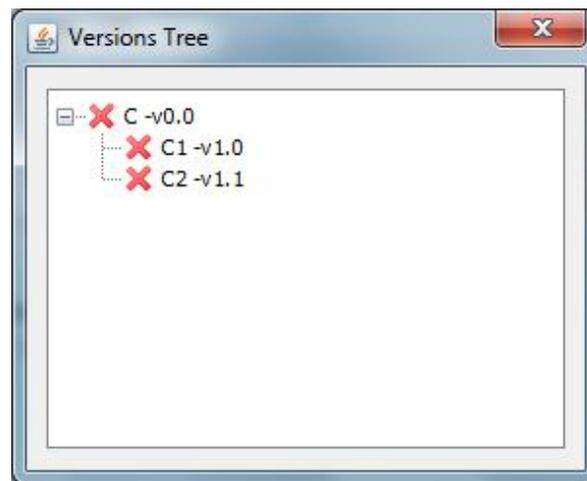


FIGURA 4.14. ÁRVORE DE VERSÃO DAS VARIANTES

Assim que a variante é associada a um cenário de sobrevivência, todos os outros cenários que forem sobrepostos ao cenário associado não poderão utilizar a variante associada nem as suas versões, devido aos problemas já comentados. Optamos por restringir a associação das variantes ao invés de dar a possibilidade do DBA ir associando todas as variantes que ele desejar e ao final receber uma mensagem de erro informando quais as variantes que não podem ser associadas devido aos cenários sobrepostos. Esta demonstração está na Figura 4.15 que faz a associação da variante C1 ao cenário B. A Figura 4.16 mostra que o DBA não consegue selecionar as variantes C, C1 e C2 para associar ao cenário sobreposto A e a Figura 4.17 mostra que um cenário não sobreposto ao cenário B pode associar qualquer uma das três variáveis.

Na Figura 4.15, só existem três variantes disponíveis, pois quando foi criado este exemplo só existiam essas três variantes cadastradas no banco de dados. As variantes disponíveis são todas as variantes contidas no esquema do banco de dados que está sendo utilizado.

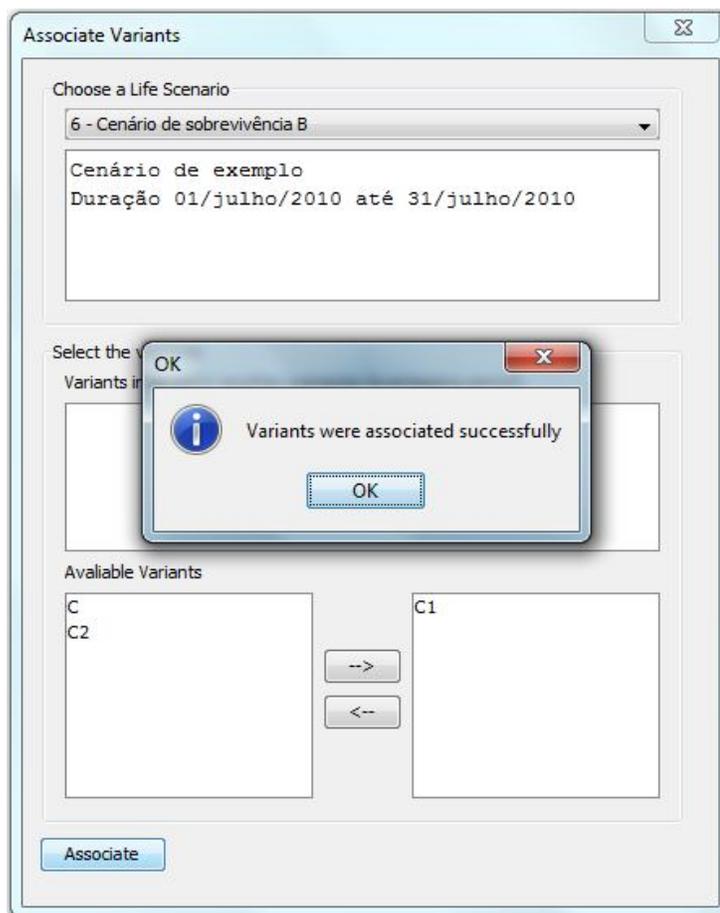


FIGURA 4.15. ASSOCIANDO A VARIANTE C1 AO CENÁRIO DE SOBREVIVÊNCIA B

Como comentado, a variante C1 associada ao cenário B irá bloquear a possibilidade de sua associação, de C e C2 ao cenário A. Isso está demonstrado na Figura 4.16.

A Figura 4.16 mostra ao DBA que o cenário A está sobreposto ao cenário B, portanto bloqueando a associação da variante C1, ainda assim, C e C2 não estão associadas a nenhum cenário, elas pertencem ao cenário global, portanto quando uma variante é bloqueada, a ferramenta precisa fazer uma busca em árvore nas versões até a raiz. Quando a raiz da árvore é encontrada, é feita uma busca por todos os nós existentes nesta árvore e estes são inseridos em uma lista de variantes contidas em um cenário sobreposto.

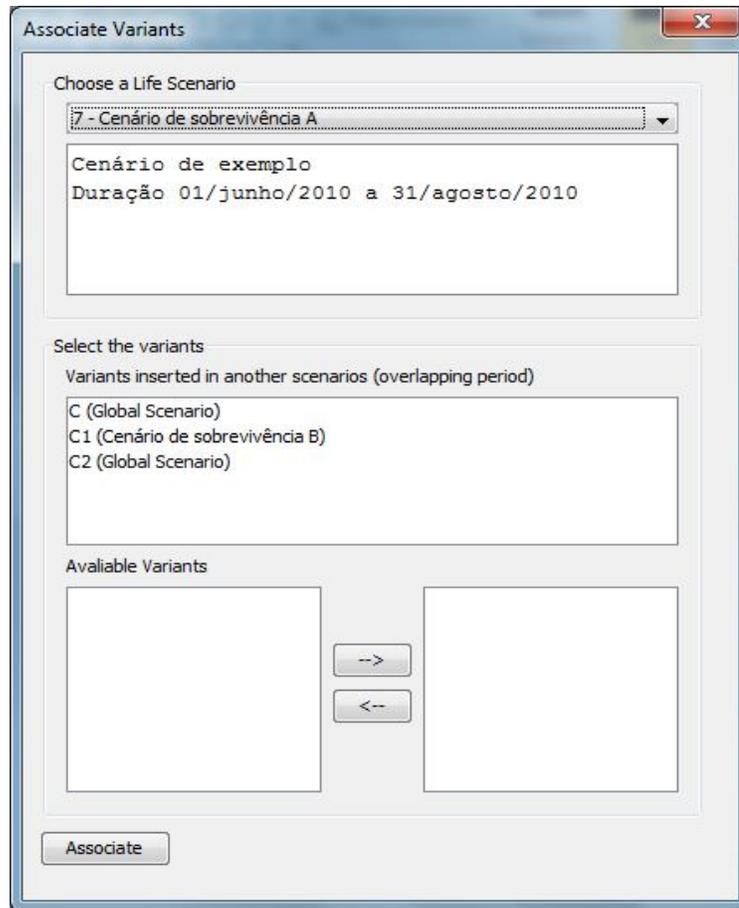


FIGURA 4.16. TENTATIVA DE ASSOCIAÇÃO DAS VARIANTES C, C1 E C2 NO CENÁRIO A

A questão de variantes conflitantes e cenários sobrepostos foi bastante discutida e no futuro espera-se uma melhoria neste formato de busca para identificar variantes que estão em conflito ou não, já que é possível existir uma variante X que possui um cenário de aplicabilidade que não é sobreposto a um período de aplicabilidade de uma versão filha X1, isso permitiria a associação de X em um cenário de sobrevivência A e X1 em um cenário de sobrevivência B, mesmo que os cenários A e B estejam sobrepostos.

A solução adotada, como mostra a Figura 4.17, é não apresentar as variantes na lista de variantes disponíveis que o DBA pode selecionar para fazer a associação ao cenário. Como isso é sempre verificado antes da associação de uma variante a um cenário, o DBA não precisa se preocupar com cenários sobrepostos.

Para finalizar o exemplo, foi criado um cenário de sobrevivência, chamado de “Cenário 1”, que não era sobreposto ao cenário de sobrevivência B.

Como pode se observar na Figura 4.17, agora as variantes C, C1 e C2 estão disponíveis e podem ser associadas ao cenário não sobreposto.

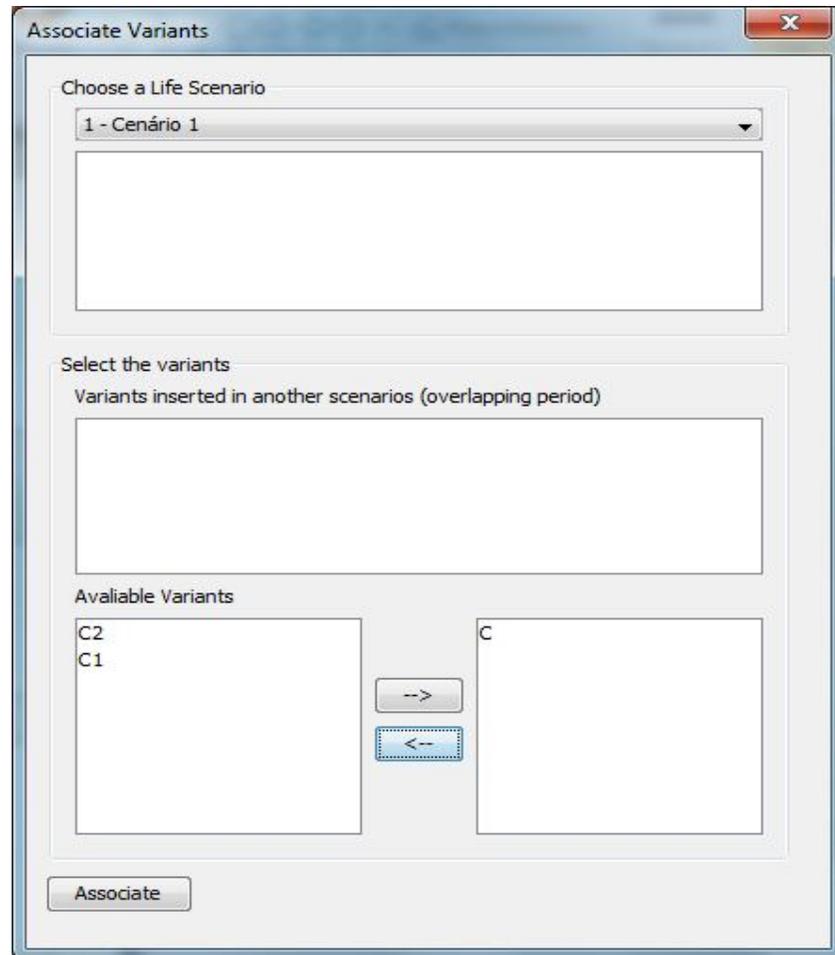


FIGURA 4.17. ASSOCIAÇÃO DA VARIANTE C A UM CENÁRIO NÃO SOBREPOSTO AO CENÁRIO B

Assim que um cenário de sobrevivência entra em seu período de ativação, a ferramenta o detecta e acorda todas as variantes contidas nele. Essa informação é disposta no *tray icon* da barra de tarefas, como mostra a Figura 4.18. Uma mensagem também é disposta quando um cenário é fechado.

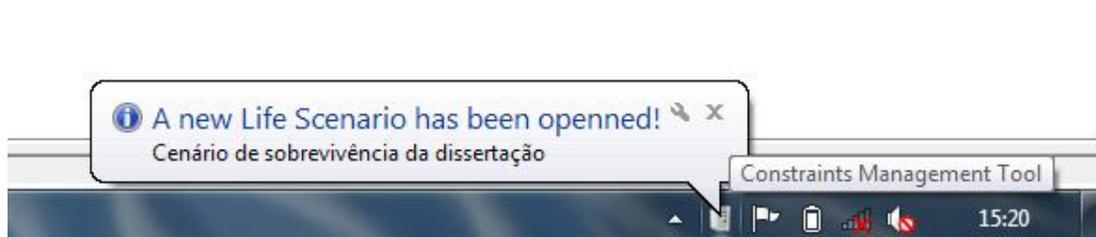


FIGURA 4.18. AVISO DE ABERTURA DE CENÁRIO NA BARRA DE TAREFAS

4.5 CONSIDERAÇÕES FINAIS

A criação de cenários e variantes é um conceito totalmente novo de se trabalhar com restrições em bancos de dados e por isso é esperado que haja certa resistência para mudar do paradigma das restrições invariantes para o das variantes. As variantes possibilitam uma flexibilidade muito maior ao DBA, mais segurança ao se gerenciar restrições, já que a ferramenta armazena o histórico, além de deixar o SGBD sempre com as restrições que realmente importam em um determinado momento, com menos restrições a serem verificadas pelo SGBD, ganhasse em desempenho.

5 CONCLUSÃO

5.1 CONSIDERAÇÕES INICIAIS

O objetivo deste trabalho foi criar uma ferramenta capaz de auxiliar no gerenciamento das restrições dos bancos de dados.

Até o momento da conclusão desta dissertação, nenhuma outra ferramenta apresentando os mesmos propósitos foi encontrada. Algumas justificativas para a criação desta ferramenta são: o fato de ser uma ferramenta inovadora; implementar um novo paradigma para tratar as restrições invariantes; facilitar o gerenciamento das restrições; servir de base para novas pesquisas e implementações.

Com essas justificativas expostas, espera-se que o uso das restrições se torne mais constante pelos DBAs e também pelos projetistas de bancos de dados durante a concepção de um banco.

5.2 DIFICULDADES ENCONTRADAS

O desenvolvimento de uma ferramenta com o propósito apresentado por esta dissertação não é trivial. A ferramenta foi feita em Java, utilizando a IDE do NetBeans. Até o final de sua implementação, existiam cerca de dez mil linhas de código e cerca de vinte e cinco classes. Por se tratar de uma linguagem orientada a objetos, seus conceitos podem e devem ser aproveitados, como polimorfismo e fácil reutilização de código (SIERRA and BATES, 2009). Por exemplo, para adicionar um SGBD novo à ferramenta, basta estender a classe *Database* e implementar a classe *ITargetDatabase* com os comandos em SQL correspondentes aos comandos já criados para o Microsoft SQL Server.

Inicialmente foi escolhida a IDE do eclipse, que por sua vez não possuía um facilitador para a construção de interfaces gráficas, portanto fez-se necessário o emprego de um outro programa para criar as interfaces gráficas, o JFormDesigner, que é uma ferramenta não gratuita e de utilização não muito simples, além de ser complicado manusear as interfaces já criadas e adicionar novos componentes. Depois de tantos problemas com essas duas ferramentas, resolvemos parar o uso de ambas e migrar tudo para a IDE do NetBeans que possui todas as funcionalidades da IDE do eclipse, além de outras funcionalidades.

Para o versionamento das restrições, novos atributos foram inseridos a algumas tabelas já existentes modificando a estrutura no banco de dados para suportar a implementação de versões. A maioria dos dados que são inseridos no banco de dados da ferramenta, são feitos através de *stored procedures*, que são procedimentos armazenados no próprio BD. Para criar uma versão de uma restrição não foi diferente, fez-se necessário o uso de uma *stored procedure*, porém um pouco mais complexa, que irá analisar toda a procedência da restrição. Essa análise retorna um resultado numérico informando a versão correta desta restrição, por exemplo, se temos as restrições, representadas pela Figura 5.1, inseridas no banco de dados da ferramenta, onde A é a versão raiz, B, C e D são filhas de A, E e F são filhas de D. Com isso temos que a versão de A é 0.0, a versão de B é 1.0, C é 1.1, D é 1.2, E é 2.0 e F é 2.1. Se for desejado inserir uma nova versão da restrição A, vamos chamar de G, essa versão deverá ser, portanto, a versão 1.3, que é filha de A e irmã de B, C e D.

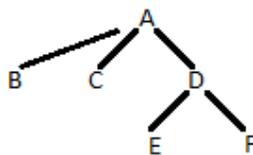


FIGURA 5.1. EXEMPLO DE ESTRUTURA EM ÁRVORE DE VERSÕES DE UMA RESTRIÇÃO

O uso deste esquema de organização em árvore das restrições foi empregado para as restrições invariantes, para as restrições variantes e também para os pacotes.

A implementação de cenários de sobrevivência e das variantes requerem um estudo profundo, principalmente das pesquisas de Camolesi que associam os limites temporais para cenários (CAMOLESI JR., 2004; CAMOLESI JR.,

2006), além dos estudos de Allen (ALLEN, 1983). Como já explicado na Seção 4 que uma variante ou uma de suas versões não podem ser associadas simultaneamente a dois cenários de sobrevivência sobrepostos, precisamos identificar quando um cenário está sobreposto a outro. A maneira que identificamos para fazer isso está apresentada na Figura 5.2.

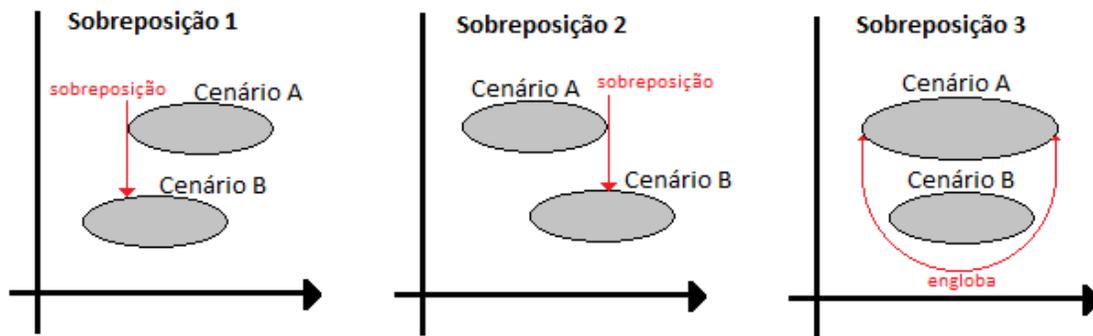


FIGURA 5.2. TIPOS DE SOBREPOSIÇÃO DE CENÁRIOS

Esta Figura apresenta os três tipos de sobreposição necessários para identificar uma sobreposição. Na realidade existem quatro tipos, o quarto tipo seria o oposto da sobreposição 3, mas ele não é necessário já que seria uma união da sobreposição 1 e da sobreposição 2, onde o momento de abertura do cenário de A e o momento de fechamento de A estão contidos no cenário B.

Na sobreposição 1, o cenário A é sobreposto ao cenário B pois o momento de abertura do cenário A está contido no cenário B. Na sobreposição 2, o cenário A é sobreposto ao cenário B pois o momento de fechamento do cenário A está contido no cenário B. O terceiro e último tipo é uma sobreposição, pois o cenário A, apesar de não ter nem o momento de abertura e nem o fechamento contido no cenário B, o seu momento de abertura é menor do que o momento de abertura do cenário B e o seu momento de fechamento é maior do que o momento de fechamento do cenário B.

Com esses três tipos de sobreposições definidos, é possível gerar uma regra para identificar cenários sobrepostos, o momento de abertura do cenário A será chamado de Aa , o seu momento de fechamento de Af , o momento de abertura do cenário B de Ba e o seu momento de fechamento de Bf . Com isso, temos a seguinte regra: Se $(Ba \leq Aa \geq Bf)$ **OU** $(Ba \leq Af \geq Bf)$ **OU** $(Aa \leq Ba \text{ E } Af \geq Bf)$

então é um cenário sobreposto, se nenhuma das três regras forem satisfeitas então não temos um cenário sobreposto.

Após a identificação dos cenários sobrepostos, as variantes desses cenários sobrepostos são inseridas em uma lista de variantes não disponíveis para serem associadas ao cenário escolhido. Ao final, cada variante inserida nesta lista, passará por uma busca para identificar todas as suas versões (caso possua), já que as versões também não podem ser associadas ao cenário escolhido.

5.3 CONTRIBUIÇÕES

Esta ferramenta, além de todas as facilidades que está provendo aos DBAs para trabalhar com restrições de bancos de dados, está mudando um conceito há muito tempo disseminado na área, que é chamar as restrições dos bancos de dados de invariantes. Atualmente existe a visão de que restrições de banco de dados são regras fixas, que não variam e ficam armazenadas até alguém removê-las do banco. Com o emprego desta ferramenta acredita-se que se pode modificar essa visão e que este conceito de variantes realmente funciona e pode facilitar o uso das mesmas, além de fornecer mais controle aos DBAs.

Esta ferramenta foca em dois propósitos, prover um controle mais amplo aos DBAs e uma facilidade maior para se trabalhar com regras. Apenas para fornecer um exemplo das diversas funcionalidades que esta ferramenta está agregando à área, será comentado adiante sobre o histórico das restrições. Os outros exemplos podem ser encontrados nas subseções 3.2 e 3.3. Atualmente as restrições armazenadas nos bancos de dados não possuem um histórico, o que acaba sendo um grande problema, pois caso alguém apague alguma restrição do banco de dados, ela será perdida para sempre. A ferramenta armazena o histórico todo das restrições, com isso, fornece uma segurança maior ao DBA quando for manipular as restrições, já que se por um equívoco a restrição errada é apagada, ela pode ser recuperada. Sem mencionar que armazenando o histórico das restrições, fica mais fácil para o DBA reutilizar alguma restrição antiga que não está mais ativada no banco de dados.

5.4 TRABALHOS FUTUROS

A ferramenta possibilita a incorporação de novas funcionalidades que ajudarão no trabalho do DBA. Alguns exemplos de novas funcionalidades que podem ser implementadas no futuro e são:

- Tornar a ferramenta diferente para os diversos perfis de usuários (normalmente DBAs). Se existem usuários com privilégios diferentes, cada um possuirá mais ou menos acesso ao BD;
- Tratamento de variantes e cenários baseados em perfil do usuário, possibilitando que cenários e variantes possam ser definidos distintamente para cada perfil de usuários do BD;
- Implementar lógica nebulosa para tratar as datas e os momentos em que um cenário é aberto e fechado, assim como os momentos das variantes;
- Implementar a extração e inserção de regras para outros SGBDs, o que não é uma tarefa difícil, pois toda a lógica de comunicação com o banco, tanto para inserção como para extração de restrições, foi feita com programação orientada a objetos utilizando os recursos de polimorfismo. Basta criar uma classe com o nome desejado, informar que ela é uma classe filha (*extends*) da classe que existe na ferramenta, chamada de "Database", e implementar os métodos existentes na interface "ITargetDatabase". Feito isso, deve-se saber onde estão armazenadas as visões dos metadados do SGBD para que se possa fazer a extração e inserção das restrições;
- Refinar e manter atualizada algumas tarefas feitas por esta dissertação, como por exemplo, um estudo mais profundo da sobreposição de cenários, analisando não somente os cenários de sobrevivência, mas também os cenários de aplicabilidade e com isso fazer uma projeção dos cenários de

aplicabilidade sobre o cenário de sobrevivência e observar se as variantes irão estar ativas nos dois cenários ao mesmo tempo e dessa forma, entrando em um conflito que não deve ser permitido.

5.5 CONSIDERAÇÕES FINAIS

Participar de um projeto inovador é muito gratificante. O desenvolvimento desta ferramenta pode ser considerado um sucesso porque o objetivo, que era gerenciar restrições temporais em banco de dados, foi alcançado e atende as motivações do trabalho. As experiências feitas mostraram isso. O DBA que fizer uso desta ferramenta irá facilitar as suas tarefas com relação à manipulação de restrições, além de ganhar uma flexibilidade maior, mais tempo livre para outras atividades, já que as tarefas podem ser executadas automaticamente, e mais segurança, já que a ferramenta armazena todo o histórico das restrições.

O mais importante é que com a criação desta ferramenta, com a implementação das funcionalidades descritas neste trabalho e outras que serão provenientes de estudos, esta ferramenta consiga alçar voo e continue a implementar novas funcionalidades com o âmbito de facilitar e melhorar a manipulação e o gerenciamento das restrições dos bancos de dados.

Referências Bibliográficas

Allen, J.F., *Maintaining knowledge about temporal intervals*. Commun. ACM, 1983. **26**(11): p. 832-843.

Amyot, D. and Yan, J.B., *Flexible verification of user-defined semantic constraints in modelling tools*, In: *Proceedings of the 2008 conference of the center for advanced studies on collaborative research: meeting of minds*. 2008, ACM: Ontario, Canada.

Autili, M., Inverardi, P., and Pelliccione, P., *A scenario based notation for specifying temporal properties*, In: *Proceedings of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools*. 2006, ACM: Shanghai, China.

Beighley, L., *Head First SQL: Your Brain on SQL -- A Learner's Guide*. 2007: O'Reilly Media.

Belussi, A., Negri, M., and Pelagatti, G., *An integrity constraints driven system for updating spatial databases*, In: *Proceedings of the 8th ACM international symposium on Advances in geographic information systems*. 2000, ACM: Washington, D.C., United States.

Buneman, P., Davidson, S.B., and Kosky, A., *Semantics of Database Transformations*, In: *Selected Papers from a Workshop on Semantics in Databases*. 1998, Springer-Verlag.

Camolesi Jr., L., *Survivability and applicability in database constraints: temporal boundary to data integrity scenarios*, In: *Proceedings of V IEEE International Conference on Information Technology: Coding and Computing*, Press, I., Editor. 2004, IEEE International Conference on Information Technology: Coding and Computing (ITCC): Las Vegas. p. 518-522.

Camolesi Jr., L., *Temporal Scenarios for Database Constraints: Data Integrity Modeling based on Contextual Predicates*, In: *Proceedings of XXXII Latin-American Conference on Informatics*. 2006, Latin-American Conference on Informatics (CLEI): Santiago (Chile). p. 11.

Cordeiro, R.L.F., Santos, C.S.d., and Edelweiss, N., *Classificação de Restrições de Integridade em Bancos de Dados Temporais de Versões* In: *XIX Simpósio Brasileiro de Banco de Dados (SBBD)*. 2004: Brasília, Brasil. p. 363-377.

Coronel, C. and Rob, P., *Database Systems: Design, Implementation, and Management*. 8th ed. 2007: Course Technology.

Currim, F. and Ram, S., *Conceptually modeling windows and bounds for space and time in database constraints*. Commun. ACM, 2008. **51**(11): p. 125-129.

Date, C.J., *An Introduction to Database Systems*. 8th ed. 2003: Addison Wesley.

Delaney, K., Randal, P.S., Tripp, K.L., Cunningham, C., and Machanic, A., *Microsoft SQL Server 2008 Internals (Pro - Developer)* 2009: Microsoft Press.

Dey, D., Barron, T.M., and Storey, V.C., *A complete temporal relational algebra*. The VLDB Journal, 1996. **5**(3): p. 167-180.

Díaz, O. and Piattini, M., *Metrics for Active Database Maintainability*, In: *Proceedings of the 11th International Conference on Advanced Information Systems Engineering*. 1999, Springer-Verlag.

Doucet, A. and Monties, S., *Versions of Integrity Constraints in Multiversion Databases*, In: *Proceedings of the 8th International Conference on Database and Expert Systems Applications*. 1997, Springer-Verlag.

Elmasri, R. and Navathe, S.B., *Fundamentals of Database Systems*. 5th ed. 2006: Addison Wesley.

Etalle, S. and Winsborough, W.H., *Integrity constraints in trust management*, In: *Proceedings of the tenth ACM symposium on Access control models and technologies*. 2005, ACM: Stockholm, Sweden.

Fernandez-Medina, E., Piattini, M., and Serrano, M.A., *Specification of security constraint in UML*, In: *Proceedings of the 35th International Carnahan Conference on Security Technology*. 2001: London, UK. p. 163-171.

Flesca, S., Greco, S., and Zumpano, E., *Active integrity constraints*, In: *Proceedings of the 6th ACM SIGPLAN international conference on Principles and practice of declarative programming*. 2004, ACM: Verona, Italy.

Fontaine, D. and Ramaux, N., *An Approach by Graphs for the Recognition of Temporal Scenarios*. IEEE Transactions on Systems, Man, and Cybernetics, 1998. **28**(3): p. 387-403.

Goralwalla, I.A., Tansel, A.U., \, M.T., \#214, and zsu, *Experimenting with temporal relational databases*, In: *Proceedings of the fourth international conference on Information and knowledge management*. 1995, ACM: Baltimore, Maryland, United States.

Grishman, R. and Sterling, J., *Information extraction and semantic constraints*, In: *Proceedings of the 13th conference on Computational linguistics - Volume 3*. 1990, Association for Computational Linguistics: Helsinki, Finland.

Hertzum, M., *Making use of scenarios: a field study of conceptual design*. Int. J. Hum.-Comput. Stud., 2003. **58**(2): p. 215-239.

ISO, *Data elements and interchange formats -- Information interchange -- Representation of dates and times*, In: *Technical Committee ISO/TC 154*. 2000.

Kriegel, A., *SQL Bible*. 2nd ed. 2008: Wiley.

Lin, Y., Kemme, B., Jiménez-Peris, R., Patiño-Martínez, M., and Armendáriz-Iñigo, J.E., *Snapshot isolation and integrity constraints in replicated databases*. ACM Trans. Database Syst., 2009. **34**(2): p. 1-49.

Maduell, M. and Mello, R., *FERIX: Uma Ferramenta para Extração de Restrições de Integridade de Fontes de Dados XML*, In: *Simpósio Brasileiro de Banco de Dados (SBBD)*. 2007: João Pessoa, Brasil.

Microsoft-Developer-Network. *Instrução Create Rule*. 2009 10/03/2009]; Disponível em: msdn.microsoft.com/pt-br/library/ms188064.aspx. Último acesso: 2 de março de 2010 às 19:30.

Mok, A.K., Lee, C.-G., Woo, H., and Konana, P., *The Monitoring of Timing Constraints on Time Intervals*, In: *Proceedings of the 23rd IEEE Real-Time Systems Symposium*. 2002, IEEE Computer Society.

Muniswamy-Reddy, K.-K. and Holland, D.A., *Causality-based versioning*. Trans. Storage, 2009. **5**(4): p. 1-28.

Nazar, I.F. and Vergilio, S.R., *X-TOOL: uma ferramenta de teste de esquemas para estrutura de dados*. 2007, Universidade Federal do Paraná. Setor de Ciências Exatas. Programa de Pós-Graduação em Informática: Paraná.

Oakasha, H., Conrad, S., and Saake, G., *Consistency Management in Object-Oriented Databases*, In: *Proceedings of the Workshop on Object-Oriented Technology*. 1999, Springer-Verlag.

Ramakrishnan, R. and Gehrke, J., *Database Management Systems*. 3rd ed. 2002: McGraw Hill Higher Education.

Rankins, R., Bertucci, P., Gallelli, C., and Silverstein, A.T., *Microsoft SQL Server 2005 Unleashed*. 2006: Sams.

Revesz, P., *A comparison of abstract data type and constraint database approaches to GIS query languages*, In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2009, ACM: Seattle, Washington.

Riordan, R., *Designing Relational Database Systems* ed. Press, M. 1999.

Sierra, K. and Bates, B., *Use a cabeça! Java*. 2nd ed. 2009: O'Reilly Media.

Silberschatz, A., Korth, H.F., and Sudarshan, S., *Database System Concepts*. 6 ed, ed. McGraw-Hill Science/Engineering/Math. 2010.

Sistla, A.P. and Wolfson, O., *Temporal Conditions and Integrity Constraints in Active Database Systems*, In: *SIGMOD Conference*. 1995. p. 269-280.