

RAFAEL BEVILACQUA MELLO

MIGRAÇÃO ENTRE SISTEMAS GERENCIADORES DE BANCO DE
DADOS

ASSIS
2009

MIGRAÇÃO ENTRE SISTEMAS GERENCIADORES DE BANCO DE DADOS

RAFAEL BEVILACQUA MELLO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Analisador: Diomara Martins Reigato Barros

ASSIS
2009

RAFAEL BEVILACQUA MELLO

MIGRAÇÃO ENTRE SISTEMAS GERENCIADORES DE BANCO DE DADOS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Alex Sandro Romeo de Souza Poletto

Área de Concentração: Sistema Gerenciador de Banco de Dados. Migração entre Sistemas de Banco de Dados.

ASSIS
2009

DEDICATÓRIA

A Deus por me dar forças nesta jornada.

*A meus pais Célio e Sonia, minha irmã
Graciele e minha namorada Viviane por
estarem sempre ao meu lado, me ajudando
sempre a alcançar meus objetivos.*

*A todos os amigos presentes em todos os
momentos.*

AGRADECIMENTOS

Ao Prof. Dr. Alex Sandro Romeo de Souza Poletto, pela compreensão e auxílio na orientação, transmitindo-me conhecimento de forma clara e objetiva, com muito profissionalismo e inteligência, tornando sua ajuda indispensável para a conclusão desse trabalho.

Aos meus grandes amigos que fiz nesta instituição Larissa Piovezani, Rodrigo Merlin, Simone Cardoso e Vinicius Oliveira, que me ajudaram diretamente ou indiretamente para a conclusão deste trabalho. Onde os levarei para sempre em minha memória.

A todos os amigos do CEPEIN (Centro de Pesquisa em Informática) que foram muito importantes todos esses anos, ajudando-me sempre que possível.

RESUMO

Este trabalho tem por objetivo mostrar como é feita a migração de dados entre Sistemas Gerenciadores de Banco de Dados (SGBD), bem como oferecer um modelo para a migração de dados, apoiada por uma ferramenta desenvolvida no transcorrer dessa pesquisa, com base nos Sistemas Gerenciadores de Banco de Dados Oracle, MySQL e PostgreSQL.

Palavras-chave: Sistema Gerenciador de Banco de Dados, Migração.

ABSTRACT

This project aims on describing how data migration between Database Management Systems (DMS) works, and also offering a model for data migration, demonstrated by a tool developed during this research, based on Oracle, MySQL and PostgreSQL Database Management Systems.

Keywords: Database Management Systems, Migration.

LISTA DE ILUSTRAÇÕES

Figura 1. Representação de um Sistema de Banco de Dados	17
Figura 2. Exemplo de um Sistema Gerenciador de Banco de Dados	18
Figura 3. Exemplo de Chave Primária.....	22
Figura 4. Exemplo de Chave Primária Composta	23
Figura 5. Exemplo de Chave Estrangeira.....	24
Figura 6. Nível 1	25
Figura 7. Nível 2	26
Figura 8. Nível 3	27
Figura 9. Proposta do Trabalho.....	31
Figura 10. Etapa 1 da Proposta do Trabalho	32
Figura 11. Etapa 2 da Proposta do Trabalho	33
Figura 12. Tela de Conexão com os Bancos de Dados	34
Figura 13. Tela 1 da Criação dos Bancos de Dados	34
Figura 14. Tela 2 da Criação dos Bancos de Dados	35
Figura 15. Tela da Criação dos Arquivos Textos.....	35
Figura 16. Tela do Código da Criação dos Arquivos Textos	36
Figura 17. Tela de Migração das Tabelas	36
Figura 18. Tela de Migração dos Dados	37
Figura 19. Tela do Código que lê os Arquivos Textos.....	37
Figura 20. Tela de Conclusão da Migração.....	38
Figura 21. Modelo de Dados do Sistema de Consultorio	39
Figura 22. Estrutura das Tabelas	40
Figura 23. Arquivo Texto das Tabelas.....	41
Figura 24. Arquivo Texto dos Dados	42
Figura 25. Select da Conclusão da Migração.....	43

LISTA DE TABELAS

Tabela 1. Tipos de dados suportados pelo Oracle	18
Tabela 2. Tipos de dados suportados pelo MySQL.....	19
Tabela 3. Tipos de dados suportados pelo PostgreSQL	21
Tabela 4. Comparação do banco de dados Oracle e MySQL	27
Tabela 5. Comparação do banco de dados Oracle e PostgreSQL.....	28
Tabela 6. Comparação do banco de dados MySQL e PostgreSQL	30

LISTA DE ABREVIATURAS E SIGLAS

SGBD	Sistema Gerenciador de Banco de Dados
SBD	Sistema de Banco de Dados
DDL	Data Definition Language
DML	Data Manipulation Language
DBA	Administrador de Banco de Dados
SQL	Structure Query Language
DER	Diagrama de Entidade Relacionamento

SUMÁRIO

1. INTRODUÇÃO	13
1.1. OBJETIVOS	14
1.2. JUSTIFICATIVAS	14
1.3. MOTIVAÇÃO	14
1.4. ESTRUTURA DO TRABALHO	14
2. SISTEMA DE BANCO DE DADOS	16
2.1. CONCEITOS GERAIS.....	16
2.2. SISTEMAS GERENCIADOR DE BANCO DE DADOS	17
2.3. TIPOS DE DADOS DOS SGBD	18
2.3.1 Oracle	18
2.3.2 MySQL	19
2.3.3 PostgreSQL.....	21
2.4. INTEGRIDADE REFERENCIAL	22
2.4.1 Chave Primária (Primary Key)	22
2.4.2 Chave Estrangeira (Foreign Key)	23
3. MIGRAÇÃO ENTRE SISTEMAS DE BANCO DE DADOS	25
3.1. NÍVEIS DE MIGRAÇÃO	25
3.2. COMPATIBILIDADE DE TIPOS DE DADOS	27
3.2.1 Oracle x MySQL	27
3.2.2 Oracle x PostgreSQL.....	28
3.2.3 MySQL x PostgreSQL	29
4. PROPOSTA DO TRABALHO	31
4.1. ETAPAS DA PROPOSTA.....	31
4.1.1 Etapa 1.....	32
4.1.2 Etapa 2.....	32
4.2. FERRAMENTA DESENVOLVIDA PARA AS MIGRAÇÕES	33
4.2.1 Conexão com os Bancos de Dados.....	33

4.2.2 Criando a nova base de dados no SGBD Destino	34
4.2.3 Gerando os Arquivos Textos das Tabelas e Dados	35
4.2.4 Lendo os Arquivos Textos e Migrando	36
4.2.5 Conclusão da Migração entre os SGBD	38
5. ESTUDO DE CASO	39
5.1. APLICAÇÃO DA ETAPA 1 DA PROPOSTA DO TRABALHO.....	41
5.2. APLICAÇÃO DA ETAPA 2 DA PROPOSTA DO TRABALHO.....	41
5.3. CONCLUSÃO DA MIGRAÇÃO.....	43
6. CONSIDERAÇÕES FINAIS E PROJEÇÕES FUTURAS	44
REFERÊNCIAS.....	45
ANEXO	46

1. INTRODUÇÃO

Nas últimas décadas, o aumento do uso de computadores e aplicativos, fez com que fosse criado um grande número de diferentes tecnologias de gerenciamento de dados, e o que na época era a melhor solução, hoje está desatualizado, ocasionando uma grande necessidade em se trocar seu Sistema Gerenciador de Banco de Dados (SGBD).

Atualmente, há uma grande importância em se migrar um SGBD, dentre os motivos estão o melhor desempenho, o menor custo e a usabilidade de acesso aos dados por mais usuários.

Uma migração de base de dados não é tão fácil de ser realizada, ela embute certo nível de risco, no qual, se o projeto de migração dentro de uma empresa não for bem planejado, os riscos podem ser fatais, portanto antes de realizar a migração, essa empresa deve escolher cuidadosamente uma base de dados que atenda suas necessidades.

Podem ocorrer muitos problemas no meio do processo de migração, sendo que um dos principais problemas é o corrompimento da base de dados original ou até mesmo a perda dessa base. Esses problemas podem ser ocasionados devido a fatores não esperados tais como a falta de energia, dentre outros. Portanto, no processo de migração deve ser feito uma cópia da base de dados original, e por intermédio desta cópia realizar a migração. Já a base original ficará intacta perante a migração, já que caso aconteça qualquer um dos problemas levantados, ou até mesmo outros problemas não apontados, a empresa em questão não perderá suas informações.

1.1. OBJETIVOS

Este trabalho tem por objetivo mostrar o que é uma migração entre SGBD, como esta é realizada e os requisitos necessários para que este possa acontecer.

Além disso, como segundo objetivo será a preparação de um modelo de migração, bem como o desenvolvimento de uma ferramenta que auxilie esse modelo durante o processo de migração entre os SGBD.

1.2. JUSTIFICATIVAS

Diante dos problemas encontrados nas bases de dados de uma empresa, devido a fatores de custo, desempenho e usabilidades, surge então a necessidade das empresas trocarem seus SGBD. A partir desta necessidade que a migração de SGBD vem se tornando essencial.

1.3. MOTIVAÇÃO

Os motivos pelo qual este trabalho foi proposto vão ao encontro das necessidades de oferecer um modelo, apoiado por uma ferramenta que auxilie no processo de migração de bases de dados. Outra motivação, é a disponibilização desse trabalho para que interessados na área possam ter um suporte sobre migração de dados entre SGBD, já que existe uma relativa falta de material e de ferramentas com relação ao assunto em questão.

1.4. ESTRUTURA DO TRABALHO

Este trabalho foi organizado em seis capítulos, sendo o primeiro esta introdução.

No segundo capítulo, serão apresentados os conceitos gerais dos SBD, uma introdução sobre SGBD, os tipos de dados dos SGBD e os conceitos de chaves, ou seja, Integridade Referencial.

No terceiro capítulo, serão apresentados os níveis existentes para se fazer uma migração, as incompatibilidades e compatibilidades dos tipos de dados dos Sistemas Gerenciadores de Banco de Dados.

No quarto capítulo, será apresentada a proposta de trabalho e como a mesma deve ser aplicada.

No quinto capítulo, será apresentado um Estudo de Caso.

No sexto capítulo, serão apresentadas as considerações finais e as projeções futuras.

2. SISTEMAS DE BANCO DE DADOS

2.1. CONCEITOS GERAIS

Um Banco de Dados é uma coleção de dados armazenados e inter-relacionados, que atende às necessidades de vários usuários dentro de uma ou mais organizações, ou seja, coleções inter-relacionadas de muitos tipos diferentes de tabelas (TEOREY, 2007, p. 2).

As informações (dados) de uma organização são armazenadas em um Banco de Dados, este é responsável pelo controle sobre as informações armazenadas, no qual são considerados muito valiosos.

Um Banco de Dados é um sistema cujo objetivo é registrar, atualizar, manter e disponibilizar informações relevantes para a atividade de uma organização.

Os sistemas de banco de dados são projetados para gerenciar grandes blocos de informações. Esses grandes blocos de informações não existem isolados. Eles são parte da operação de alguma empresa cujo produto final pode ser informações do banco de dados ou pode ser algum dispositivo ou serviço para o qual o banco de dados desempenha apenas um papel de apoio (SILBERSCHATZ, 2006, p. 9).

Segundo Chu (1943): “Banco de Dados (BD) é um conjunto de dados estruturados de maneira adequada de forma que pode ser utilizado com eficiência por uma diversidade de aplicações dentro de uma organização”.

O SBD em uma organização corresponde a um conjunto de arquivos de dados de toda organização, armazenadas em algum tipo de dispositivo magnético, sendo manipulado por um conjunto de programas, representados na Figura 1.

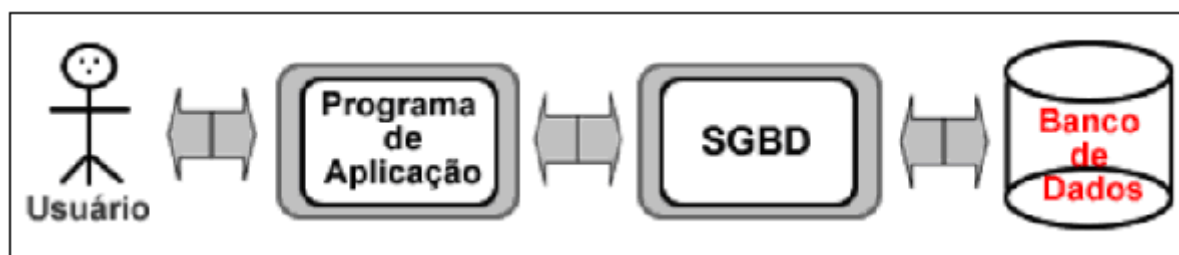


Figura 1. Representação de um Sistema de Banco de Dados

2.2. SISTEMA GERENCIADOR DE BANCO DE DADOS

Um Sistema Gerenciador de Banco de Dados (SGBD) é um sistema de software genérico para manipular bancos de dados. Um SGBD admite uma visão lógica (esquema e subesquema); visão física (métodos de acesso, *clustering* de dados); linguagem de definição de dados (SQL-DDL); linguagem de manipulação de dados (SQL-DML) e utilitários importantes, como gerenciamento de transação e controle de concorrência, integridade de dados, recuperação de falhas e segurança (TEOREY, 2007, p. 2).

O SGBD é o software responsável pela criação e manutenção de BD, ou seja, ele se comunica com o BD para que possam ser realizadas as consultadas e manipulações das informações pelos usuários, conforme ilustrado na Figura 2. Um bom SGBD deve oferecer:

- Garantia de Integridade: As restrições de consistência dos dados devem ser feitas pelo DBA.
- Restrição de Acesso não autorizado: Restringir o acesso indevido aos dados para que cada usuário tenha acesso apenas aos dados necessários.
- Recuperação e Backup: Garantir a recuperação dos dados quando houver falhas.
- Controle de Concorrência: Realizar uma atualização por vez quando o BD for acessado por mais de um usuário, para não ocorrer inconsistência de dados.
- Múltiplas Interfaces: Fornecer diversas ferramentas de acesso ao BD.

- Representação de Relações Complexas entre os Dados: Fornecer variedades de relacionamentos complexos entre os dados, recuperando e atualizando dados relacionados.

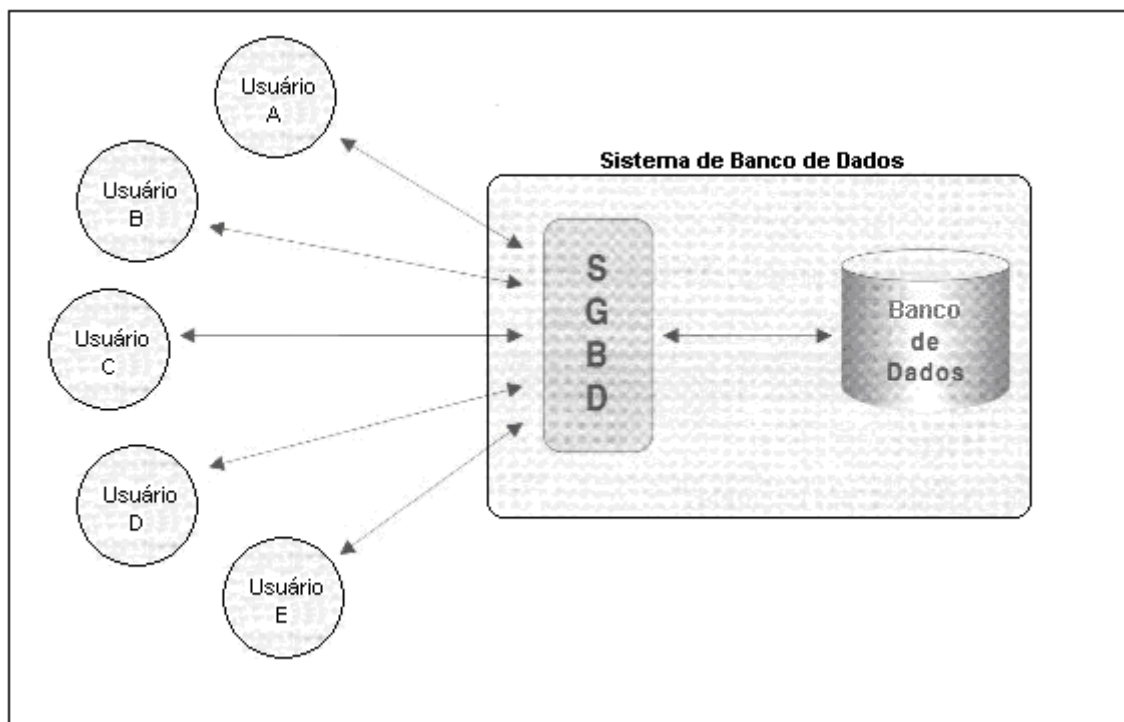


Figura 2. Exemplo de um Sistema Gerenciador de Banco de Dados

2.3. TIPOS DE DADOS DOS SGBD

2.3.1. Oracle

Tabela 1. Tipos de dados suportados pelo Oracle

ATRIBUTO	DESCRIÇÃO
BFILE	REFERÊNCIA DE VALOR BINÁRIO DE TAMANHO 4 GIGABYTES
BLOB	VALOR BINÁRIO DE COMPRIMENTO 4 GIGABYTES
CHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 250 DIGITOS

CLOB	VALOR ALFANUMÉRICO DE COMPRIMENTO 4 GIGABYTES
DATE	VALOR DE DATA
FLOAT	VALOR NUMÉRICO REAL
INT	VALOR NUMÉRICO INTEIRO
LONG	VALOR ALFANUMÉRICO DE COMPRIMENTO 2 GIGABYTES
LONG RAW	DADOS BINARIOS DE COMPRIMENTO MAXIMO DE 2 GIGABYTES
NCHAR	VALOR ALFANUMÉRICO QUE OCUPA ESPAÇO DE ARMAZENAMENTO
NCLOB	TIPO OBJETO DE CARACTERES UNICOD DE TAMANHA MAXIMO 4 GB
NUMBER	VALOR NUMÉRICO REAL OU INTEIRO
NVARCHAR2	VALOR ALFANUMERICO DE TAMANHO MAXIMO 4000 BYTES
RAW	DADOS BINARIOS DE COMPRIMENTO MAXIMO 200 BYTES
REAL	VALOR NUMÉRICO REAL
ROWID	TIPO STRING DE BASE 64 QUE REPRESENTA UMA LINHA DA TABELA
TIMESTAMP	TIPO DE DATA QUE INCLUI SEGUNDOS FRACIONADOS
UROWID	TIPO STRING DE BASE 64 QUE REPRESENTA UMA LISTA DA TABELA
VARCHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 2000 DIGITOS
VARCHAR2	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 4000 DIGITOS

2.3.2. MySQL

Tabela 2. Tipos de dados suportados pelo MySQL

ATRIBUTO	DESCRIÇÃO
TINYINT	VALOR NUMÉRICO INTEIRO DE -128 a 127
BOOLEAN	POSSUI DOIS ESTADOS: “VERDADE” OU “FALSO”
SMALLINT	VALOR NUMÉRICO INTEIRO DE -32768 a 32767
MEDIUMINT	VALOR NUMÉRICO INTEIRO DE -8388608 a 8388607
INT / INTEGER	VALOR NUMÉRICO INTEIRO DE -2147483648 a 2147483647

BIGINT	VALOR NUMÉRICO INTEIRO DE -9223372036854775808 a 9223372036854775807
FLOAT	VALOR NUMÉRICO REAL DE -3.402823466E+38 a -1.175494351E-38, 0, e 1.175494351E-38 a 3.402823466E+38
DOUBLE / REAL	VALOR NUMÉRICO REAL DE -1.7976931348623157E+308 a -2.2250738585072014E-308
DECIMAL	VALOR NUMÉRICO REAL DE -999.99 a 999.99
NUMERIC	VALOR NUMÉRICO REAL DE -9999.99 a 9999.99
DATE	VALOR DE DATA FORMATO YYYY-MM-DD
DATETIME	VALOR DE DATA E HORA FORMATO YYYY-MM-DD HH:MM:SS
TIMESTAMP	VALOR DE DATA E HORA FORMATO YYYYMMDDHHMMSS, YYMMDDHHMMSS, YYYYMMDD, ou YYMMDD
TIME	VALOR DE HORA
YEAR	VALOR DE ANO FORMATO YYYY
CHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 255 CARACTERES
VARCHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 65535 CARACTERES
TINYTEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 255 CARACTERES
TEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 65535 CARACTERES
MEDIUMTEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 16777215 CARACTERES
LONGTEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO MÁXIMO DE 4294967295 CARACTERES
ENUM	OBJETO STRING QUE PODE TER UM VALOR
SET	OBJETO STRING QUE PODE TER ZERO OU MAIS VALORES

2.3.3. PostgreSQL

Tabela 3. Tipos de dados suportados pelo PostgreSQL

ATRIBUTO	DESCRIÇÃO
SMALLINT	VALOR NUMÉRICO DE 2 BYTES
INTEGER	VALOR NUMÉRICO DE 4 BYTES
BIGINT	VALOR NUMÉRICO DE 8 BYTES
DECIMAL	TIPO NUMERICO COM PRECISAO SELECIONAVEL
NUMERIC	TIPO NUMERICO COM PRECISAO SELECIONAVEL
REAL	VALOR REAL DE 4 BYTES
DOUBLE PRECISION	VALOR REAL DE 8 BYTES
SERIAL	TIPO INTEIRO DE 4 BYTES COM AUTO-INCREMENTO
BIGSERIAL	TIPO INTEIRO DE 8 BYTES COM AUTO-INCREMENTO
MONEY	VALOR MONETARIO
CHARACTER VARYING	COMPRIMENTO VARIÁVEL COM LIMITE
VARCHAR	COMPRIMENTO VARIÁVEL COM LIMITE
CHARACTER	COMPRIMENTO FIXO, COMPLETADO COM BRANCOS
CHAR	COMPRIMENTO FIXO, COMPLETADO COM BRANCOS
TEXT	COMPRIMENTO VARIÁVEL NÃO LIMITADO
BYTEA	TAMANHO DE TEXTO BINARIO
TIMESTAMP	TANTO DATA QUANTO HORA
INTERVAL	INTERVALOS DE TEMPO
DATE	SOMENTE DATAS
TIME	SOMENTE A HORA
BOOLEAN	POSSUI DOIS ESTADOS: “VERDADE” OU “FALSO”

2.4. INTEGRIDADE REFERENCIAL

A integridade referencial pode ser especificada durante a construção das estruturas de dados de uma tabela, por meio da instrução “CREATE TABLE”, onde são acrescentadas as cláusulas “PRIMARY KEY” (chave primária), “FOREIGN KEY” (chave estrangeira), e “UNIQUE KEY” (chave candidata). No qual, modificações no banco de dados por meio de inserções, atualizações e remoções, podem causar violação da integridade referencial, e conseqüentemente inconsistência dos dados.

A chave é o conceito básico para identificar entidades e estabelecer relacionamentos entre essas entidades de um banco de dados. Existem dois tipos de chaves que os bancos de dados dão possibilidades de uso, que são a chave primária e a chave estrangeira.

2.4.1. Chave Primária (Primary Key)

Segundo SILBERSCHATZ (2006), chave primária é um conjunto de um ou mais atributos que, tomados coletivamente, nos permitem identificar de maneira exclusiva, de maneira única, uma entidade em um conjunto de entidades.

CODIGO	DESCRIÇÃO	VALOR
1	Teclado	R\$ 25,00
2	Mouse	R\$ 12,00
3	Monitor	R\$ 195,00

Figura 3. Exemplo de Chave Primária

Na Figura 3, o campo “codigo” é uma chave primária, no qual nenhum registro dele possa ser repetido. Dessa maneira através desse campo teremos o valor do registro inteiro, no qual possibilita a identificação unívoca (exclusiva) de cada uma das

entidades. Exemplo: Quando o código for '1', saberemos que a entidade correspondente é a descrição "Teclado" e o valor 'R\$ 25,00'.

DOMICILIO	PESSOA	NOME
1	25	RAFAEL BEVILACQUA MELLO
1	53	VIVIANE APARECIDA DE SOUZA
2	13	LUCAS ÁUREO GUIDASTRE

Figura 4. Exemplo de Chave Primária Composta

Na Figura 4 é mostrado um exemplo de uma tabela que possui mais que um atributo como chave primária, ou seja, tendo os campos "domicilio" e "pessoa" como chave primária, eles não podem se repetir simultaneamente. Portanto, não tem como uma tabela possuir duas ou mais chaves primárias, sendo que em alguns casos há necessidade de se combinar mais que um atributo para compor a chave primária da tabela, dando-se o nome de chave primária composta.

2.4.2. Chave Estrangeira (Foreign Key)

Uma chave estrangeira nada mais é que um atributo no qual é chave primária de outra tabela. Conforme HEUSER (1999, p. 80), uma chave estrangeira é uma coluna ou uma combinação de colunas, cujos valores aparecem necessariamente na chave primária de uma tabela. A chave estrangeira é o mecanismo que permite a implementação de relacionamentos em um banco de dados relacional.

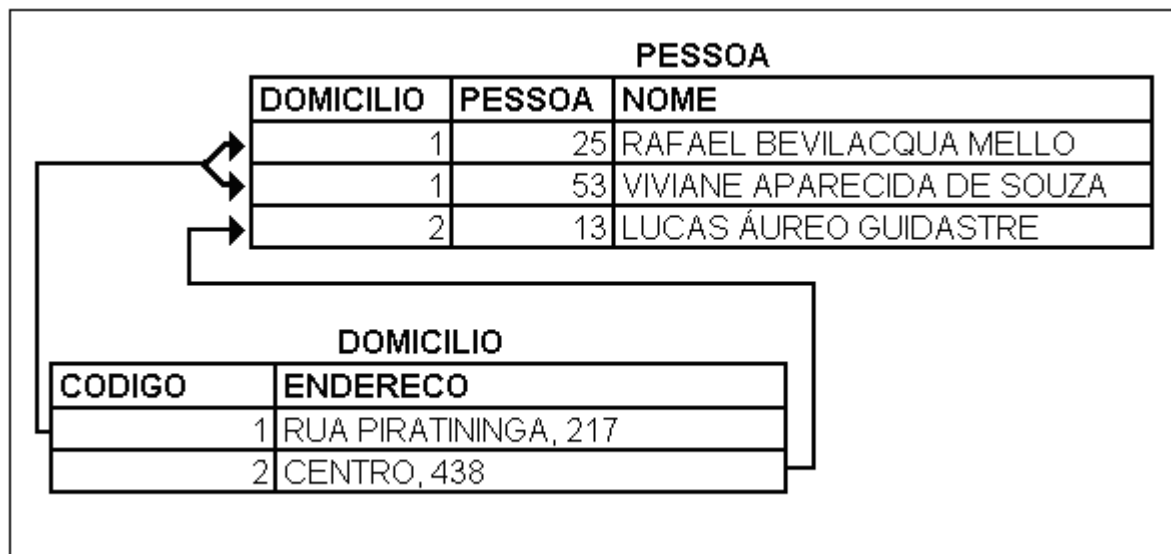


Figura 5. Exemplo de Chave Estrangeira

Na Figura 5, na tabela “pessoa” o campo “domicilio” é uma chave estrangeira que se refere à tabela “domicilio” onde esta é uma chave primária. Também podem acontecer casos em que uma chave estrangeira pode fazer parte de uma chave primária composta, como mostra também a figura acima onde a tabela “pessoa” possui uma chave primária composta por “domicilio” e “pessoa”, no qual o campo “domicilio” é uma chave estrangeira.

3. MIGRAÇÃO ENTRE SISTEMAS DE BANCO DE DADOS

Para realizar uma migração entre tecnologias diferentes de SGBD, vários fatores devem ser levados em consideração. Entre eles, o que motivou a realização da migração e se a migração será a melhor solução.

Sabe-se também que os riscos são bem elevados, portanto, é necessário fazer uma análise completa tanto da tecnologia antiga quanto da tecnologia pela qual se optou (nova), para saber se os riscos são favoráveis para que ocorra a migração.

3.1. NÍVEIS DE MIGRAÇÃO

Uma migração de base de dados é classificada em 3 (três) níveis, no qual estes determinam os elementos envolvidos na migração.

A migração de Nível 1 (um) inclui a migração dos esquemas (definições de dados) e os dados em si, ou seja, a migração ocorre apenas entre as tabelas do banco de dados e seus respectivos dados. Isso faz com que a nova base de dados não possua alterações no código fonte, que são causadas por funções específicas de cada ferramenta (PL/SQL, Triggers, Funções), conforme ilustrado na Figura 6.

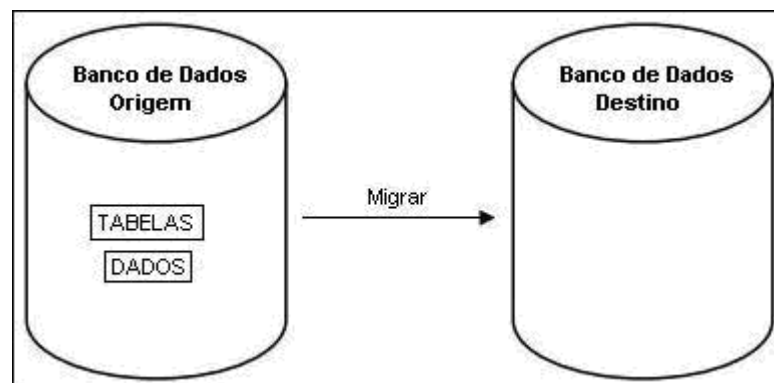


Figura 6. Nível 1

A migração de Nível 2 (dois) também conhecido como extensão do Nível 1, inclui a migração dos esquemas, dos dados em si e algumas alterações no código fonte (Triggers, Funções, PL/SQL, entre outras).

Essas alterações são realizadas quando há diferenças de versões, o que ocasiona uma necessidade de se alterar o código fonte do aplicativo para que este possa funcionar corretamente com a nova base de dados, e essas alterações que diferenciam a migração de Nível 1 para a migração de Nível 2, conforme ilustrado na Figura 7.

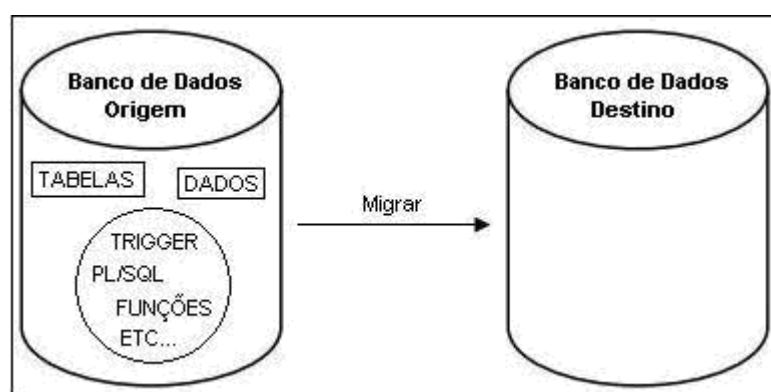


Figura 7. Nível 2

A migração de Nível 3 (três) é totalmente diferente dos tipos anteriores de migração, no qual o único objetivo dos níveis anteriores é alterar o sistema de armazenamento de dados, ou seja, fazer com que o aplicativo funcione com a nova base de dados sem aproveitar recursos específicos oferecidos pela nova base de dados. A migração de Nível 3 possui dois objetivos, um deles é o mesmo dos níveis anteriores, ou seja, alterar o sistema de armazenamento de dados sem usar recursos específicos da nova versão de banco de dados. Já o segundo objetivo implica um risco maior, já que a camada de acesso a dados é alterada para que o aplicativo use os recursos específicos da nova base de dados, gerando um maior risco que os outros níveis de migração. Entretanto, o melhor desempenho e a facilidade de administração do aplicativo são os principais benefícios que este nível oferece, fazendo com que este seja mais utilizado, conforme ilustrado na Figura 8.

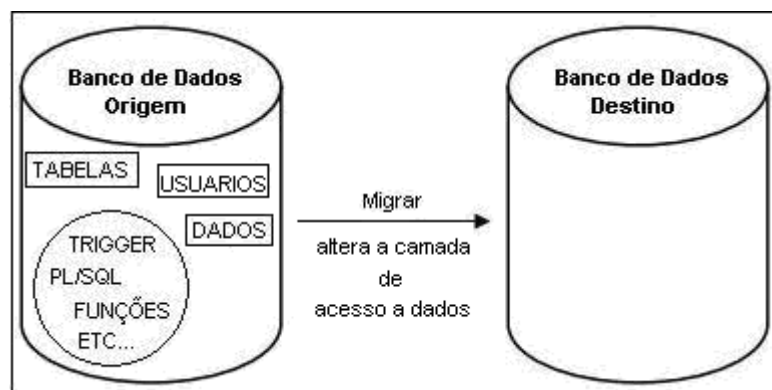


Figura 8. Nível 3

3.2. COMPATIBILIDADE DE TIPOS DE DADOS

Nesta seção serão apresentadas as compatibilidades de tipos de dados entre os sistemas gerenciadores de banco de dados Oracle e MySQL, Oracle e PostgreSQL e MySQL e PostgreSQL.

3.2.1. Oracle x MySQL

Tabela 4. Comparação do banco de dados Oracle e MySQL

Oracle	MySQL	Descrição
CHAR	CHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 250 DIGITOS
	TINYTEXT	
CLOB	LONGTEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO 4 GIGABYTES
NCLOB		
DATE	DATE	VALOR DE DATA
	DATETIME	

FLOAT REAL	FLOAT	VALOR NUMÉRICO REAL
	DOUBLE / REAL	
	DECIMAL	
	NUMERIC	
INT	INT / INTEGER	VALOR NUMÉRICO INTEIRO
	TINYINT	
	SMALLINT	
	MEDIUMINT	
	BIGINT	
VARCHAR	VARCHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 2000 DIGITOS
	TEXT	
TIMESTAMP	TIMESTAMP	TIPO DE DATA QUE INCLUI SEGUNDOS
LONG	MEDIUMTEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 4000 DIGITOS
NVARCHAR2		
VARCHAR2		

Na tabela 4, é mostrada a compatibilidade dos atributos Oracle com MySQL, este será utilizado para migração, onde os atributos Oracle serão substituídos com os atributos compatíveis do MySQL para criação das tabelas no BD Destino.

3.2.2. Oracle x PostgreSQL

Tabela 5. Comparação do banco de dados Oracle e PostgreSQL

Oracle	PostgreSQL	Descrição
BLOB	BOOLEAN	VALOR BINÁRIO
CHAR	CHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 250 DIGITOS
	CHARACTER	
DATE	DATE	VALOR DE DATA

INT	SMALLINT	VALOR NUMÉRICO INTEIRO
	INTEGER	
	BIGINT	
NUMBER	DECIMAL	VALOR NUMÉRICO REAL OU INTEIRO
	NUMERIC	
REAL FLOAT	REAL	VALOR NUMÉRICO REAL
	DOUBLE PRECISION	
TIMESTAMP	TIMESTAMP	TIPO DE DATA QUE INCLUI SEGUNDOS
VARCHAR	VARCHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 2000 DIGITOS
	CHARACTER VARYING	
VARCHAR2	TEXT	VALOR ALFANUMÉRICO DE COMPRIMENTO ATÉ 4000 DIGITOS
NVARCHAR2		

Na tabela 5, é mostrada a compatibilidade dos atributos Oracle com PostgreSQL, este será utilizado para migração, onde os atributos Oracle serão substituídos com os atributos compatíveis do PostgreSQL para criação das tabelas no BD Destino.

3.2.3. MySQL x PostgreSQL

Tabela 6. Comparação do banco de dados MySQL e PostgreSQL

MySQL	PostgreSQL	Descrição
SMALLINT	SMALLINT	VALOR NUMÉRICO DE 2 BYTES
INT / INTEGER	INTEGER	VALOR NUMÉRICO DE 4 BYTES
BIGINT	BIGINT	VALOR NUMÉRICO DE 8 BYTES
FLOAT DOUBLE / REAL DECIMAL NUMERIC	DOUBLE PRECISION	VALOR NUMÉRICO REAL
	REAL	
	DECIMAL	
	NUMERIC	

DATE	DATE	VALOR DE DATA
TIMESTAMP	TIMESTAMP	TIPO DE DATA QUE INCLUI SEGUNDOS
TIME	TIME	VALOR DE HORA
CHAR	CHAR	VALOR ALFANUMÉRICO DE COMPRIMENTO FIXO, COMPLETADO COM BRANCOS
	CHARACTER	
VARCHAR	VARCHAR	COMPRIMENTO VARIÁVEL COM LIMITE
TEXT	TEXT	COMPRIMENTO VARIÁVEL NÃO LIMITADO
MEDIUMTEXT		
LONGTEXT		

Na tabela 6, é mostrada a compatibilidade dos atributos MySQL e com PostgreSQL, este será utilizado para migração, onde os atributos MySQL serão substituídos com os atributos compatíveis do PostgreSQL para criação das tabelas no BD Destino.

A compatibilidade de tipos de dados é muito importante em uma migração entre SGBD, pois é através desta que um SGBD pode ser migrado para outro. Como a incompatibilidade dos atributos dos SGBD, foi feita tabelas para associar os atributos compatíveis e através destas, é possível fazer uma migração bem sucedida, para que os dados possam ser migradas também sem ocorrência de erros.

4. PROPOSTA DO TRABALHO

Neste capítulo, será mostrado como é feita a migração entre Sistemas Gerenciadores de Banco de Dados.

O tipo de migração a ser realizada é a de Nível 1, ou seja, migrar apenas as tabelas e seus dados correspondentes. Para demonstrar como é feita tal migração, através do banco de dados de origem é feita a criação de arquivos no formato “texto”, nos quais estão contidos as estruturas das tabelas do banco de dados de origem na linguagem DDL e seus correspondentes dados na linguagem DML (Inserts), no qual o banco de dados de destino executa esses arquivos textos para criação das estruturas das tabelas e inclusão dos dados correspondentes, conforme ilustrado na Figura 9.

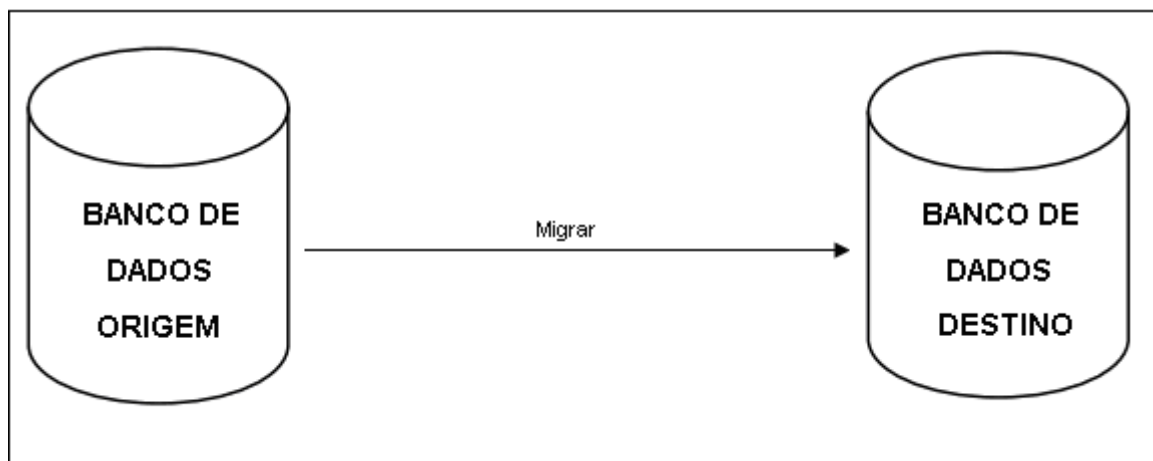


Figura 9. Proposta do Trabalho

4.1. ETAPAS DA PROPOSTA

Para que possa ser realizada a migração entre os Sistemas Gerenciadores de Banco de Dados, serão necessárias duas etapas conforme apresentada a seguir.

4.1.1. Etapa 1

Na primeira etapa da proposta, deve ser feito o levantamento das estruturas das tabelas, os tipos de atributos e suas restrições do banco de dados de origem.

Após todo o levantamento, deve-se criar um arquivo texto contendo instruções SQL onde será inserida toda a estrutura das tabelas, com os tipos de atributos compatíveis com o banco de dados de destino e as restrições do mesmo. Esse arquivo texto deve ser executado pelo banco de dados de destino. A seguir, é ilustrado como é feita a primeira etapa da proposta, conforme ilustrado na Figura 10.

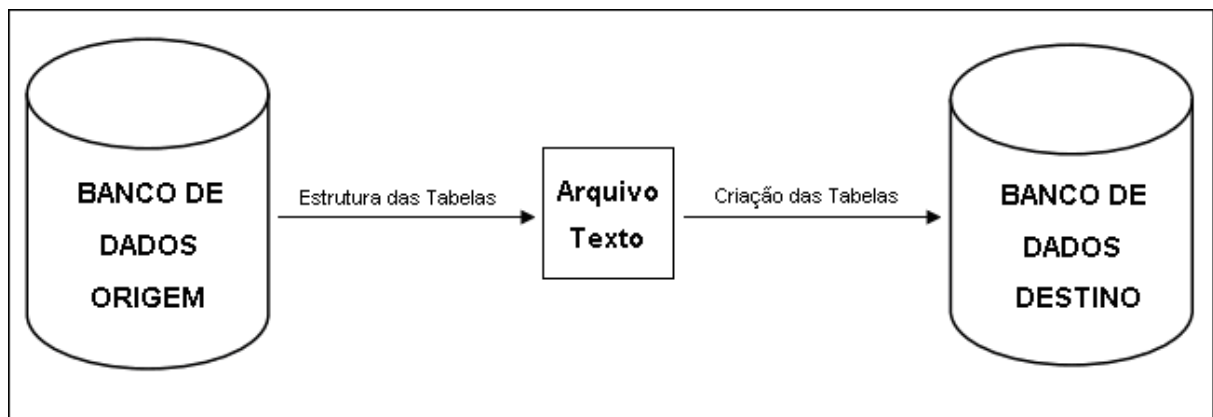


Figura 10. Etapa 1 da Proposta do Trabalho

4.1.2. Etapa 2

Na segunda etapa da proposta, deve-se preparar o banco de dados de destino para que possa receber os dados contidos no banco de dados de origem, ou seja, a Etapa 1 deste capítulo deve estar finalizada.

Para que possa ser feita esta segunda etapa, primeiramente as duas bases de dados devem estar ligadas para que os dados sejam inseridos corretamente.

Após a ligação das bases de dados é feita a criação de um arquivo texto contendo instruções SQL, onde a mesma realiza uma busca nos dados das tabelas do banco

de dados de origem, inserindo-as num arquivo texto também em instruções SQL, onde esta é executada pelo banco de dados de destino, fazendo com que os dados sejam inseridos em suas tabelas correspondentes, conforme ilustrado na Figura 11.

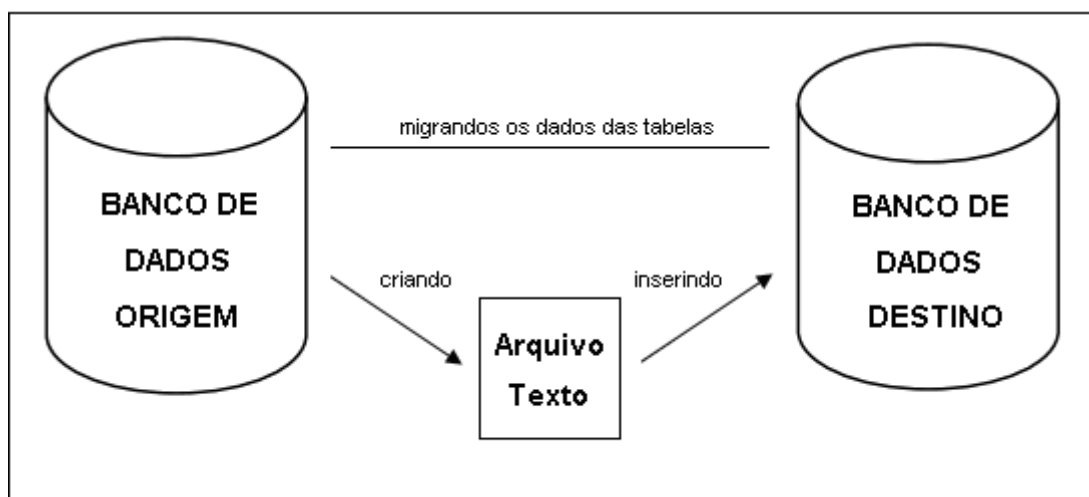


Figura 11. Etapa 2 da Proposta do Trabalho

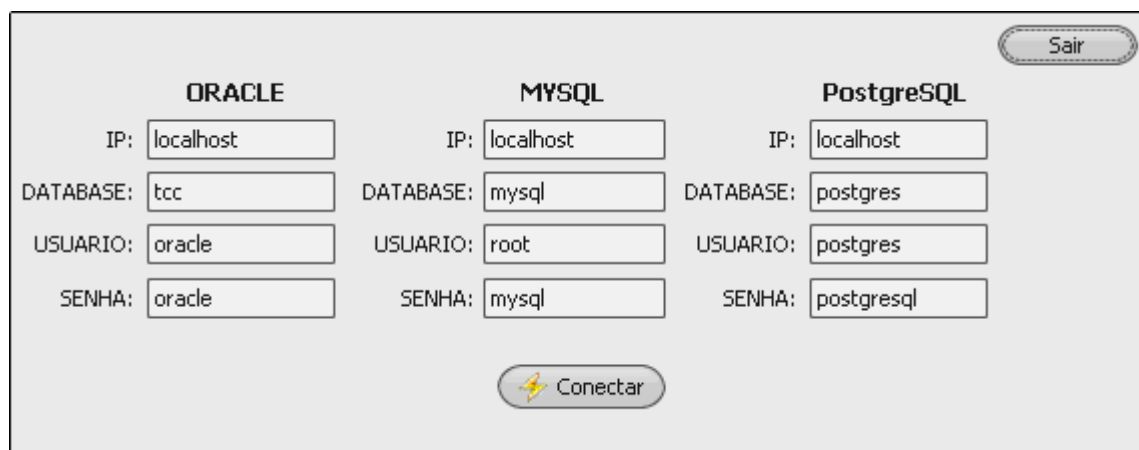
4.2. FERRAMENTA DESENVOLVIDA PARA AS MIGRAÇÕES

Conforme apresentado nos objetivos, outra contribuição desse trabalho é a elaboração de uma ferramenta para apoiar o processo de migração entre os Sistemas Gerenciadores de Banco de Dados. Em particular essa ferramenta fará as migrações entre os Bancos de Dados Oracle 11g, MySQL 5.0, PostgreSQL 8.4, e será desenvolvida utilizando a linguagem Java usando a IDE NetBeans 6.0.1.

4.2.1. Conexão com os Bancos de Dados

A primeira atividade para a realização da migração por meio da ferramenta desenvolvida, é o estabelecimento da conexão entre os bancos de dados, na qual o banco de dados Oracle será base origem e o MySQL ou PostgreSQL serão a base destino.

Na tela de conexão, deve-se preencher os campos corretamente de cada um dos bancos de dados. Esses campos são os “IPs” utilizados para buscar o local onde os bancos estão armazenados. No campo “DATABASE” deve ser informado o nome do banco, conseqüentemente devem ser informados o “USUARIO” e a “SENHA”, conforme ilustrado na Figura 12.



A interface de conexão com bancos de dados, organizada em três colunas para ORACLE, MYSQL e PostgreSQL. Cada coluna possui campos para IP, DATABASE, USUARIO e SENHA. Um botão 'Sair' está no canto superior direito e um botão 'Conectar' com um ícone de relâmpago está no centro inferior.

ORACLE	MYSQL	PostgreSQL
IP: localhost	IP: localhost	IP: localhost
DATABASE: tcc	DATABASE: mysql	DATABASE: postgres
USUARIO: oracle	USUARIO: root	USUARIO: postgres
SENHA: oracle	SENHA: mysql	SENHA: postgresql

Conectar

Figura 12. Tela de Conexão com os Bancos de Dados

4.2.2. Criando a nova base de dados no SGBD Destino

Dando continuidade, após a atividade de conexão, deve ser acionado o botão “Visualizar Tabelas”, para visualizar todas as tabelas da base de dados Origem (Oracle), como ilustrado na Figura 13.



A interface para a criação e migração de bancos de dados. À esquerda, uma seção para o banco de origem 'Oracle' com um botão 'Visualizar Tabelas'. À direita, seções para os bancos de destino 'MySQL' e 'PostgreSQL', cada uma com um botão 'Criar Banco' e quatro botões de ação: 'Gerar arquivo das Tabelas', 'Migrar Tabelas', 'Gerar arquivo dos Dados' e 'Migrar Dados'. Um botão 'Sair' está no canto superior direito.

Oracle	MySQL	PostgreSQL
Visualizar Tabelas	Criar Banco	Criar Banco
	Gerar arquivo das Tabelas	Gerar arquivo das Tabelas
	Migrar Tabelas	Migrar Tabelas
	Gerar arquivo dos Dados	Gerar arquivo dos Dados
	Migrar Dados	Migrar Dados

Figura 13. Tela 1 da Criação dos Bancos de Dados

Após ser visualizadas as tabelas da base de dados Origem, os botões para a criação dos bancos serão habilitados para que as novas bases de dados sejam criadas com o mesmo nome da base de dados origem, conforme ilustrado na Figura 14.

Oracle		Visualizar Tabelas
Tabelas		
1	AGENDAMENTOS	
2	ASS_ATEND_EXAMES	
3	ATENDIMENTOS	
4	BAIRROS	
5	CIDADES	
6	CONVENIOS	
7	ESPECIALIZACAO	
8	EXAMES	
9	PACIENTES	
10	PROFISSIONAIS	
11	USUARIOS	

MySQL
Criar Banco

Gerar arquivo das Tabelas
Migrar Tabelas

Gerar arquivo dos Dados
Migrar Dados

PostgreSQL
Criar Banco

Gerar arquivo das Tabelas
Migrar Tabelas

Gerar arquivo dos Dados
Migrar Dados

Sair

Figura 14. Tela 2 da Criação dos Bancos de Dados

4.2.3. Gerando os Arquivos Textos das Tabelas e Dados

Após a criação da Base de Dados Destino, serão habilitados os botões para a criação dos arquivos “textos” das estruturas das tabelas e dos dados, “Gerar arquivos das Tabelas” e “Gerar arquivos dos Dados”, como ilustrado na Figura 15.

Oracle		Visualizar Tabelas
Tabelas		
1	AGENDAMENTOS	
2	ASS_ATEND_EXAMES	
3	ATENDIMENTOS	
4	BAIRROS	
5	CIDADES	
6	CONVENIOS	
7	ESPECIALIZACAO	
8	EXAMES	
9	PACIENTES	
10	PROFISSIONAIS	
11	USUARIOS	

MySQL
Criar Banco

⚡ Gerar arquivo das Tabelas
Migrar Tabelas

⚡ Gerar arquivo dos Dados
Migrar Dados

PostgreSQL
Criar Banco

⚡ Gerar arquivo das Tabelas
Migrar Tabelas

⚡ Gerar arquivo dos Dados
Migrar Dados

Sair

Figura 15. Tela da Criação dos Arquivos Textos

Para a criação dos arquivos “textos”, foi desenvolvida uma função, para receber o conteúdo que será gravado no arquivo, bem como o nome que deverá ser utilizado para o salvamento, conforme ilustrado na Figura 16.

```
public void gravaArquivo(String conteudo, String tipo) {
    try {
        FileWriter x = new FileWriter(System.getProperty("user.dir") + "\\\" +
            "arquivo" + tipo + ".txt");
        x.write(conteudo);
        x.close();
        JOptionPane.showMessageDialog(null, "Arquivo Criado", "MigraSGBD",
            JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Erro ao criar o arquivo", "Atenção",
            JOptionPane.WARNING_MESSAGE);
    }
}
```

Figura 16. Tela do Código da Criação dos Arquivos Textos

4.2.4. Lendo os Arquivos Textos e Migrando

Após serem criados os arquivos texto, será habilitado o botão “Migrar Tabelas”, responsável por ler o arquivo texto das tabelas a serem migradas e executá-lo no banco de dados Destino, como ilustrado na Figura 17.



Figura 17. Tela de Migração das Tabelas

Na Figura 18, após a criação das tabelas no banco de dados Destino, será habilitado o botão de “Migrar Dados” que como as tabelas, irá ler o arquivo texto dos dados criado e executá-lo no banco de dados Destino.



Figura 18. Tela de Migração dos Dados

Para a leitura dos arquivos textos, foi desenvolvido uma função que recebe o nome do arquivo texto, joga todo o conteúdo em uma “ArrayList” do tipo “String”, onde a mesma será retornada, conforme ilustrado na Figura 19.

```
public ArrayList<String> leArquivo(String tipo) {
    ArrayList<String> conteudo = new ArrayList<String>();
    try {
        FileReader reader = new FileReader(System.getProperty("user.dir") +
            "\\\" + "arquivo" + tipo + ".txt");
        BufferedReader buffReader = new BufferedReader(reader);

        String linha;
        while ((linha = buffReader.readLine()) != null) {
            conteudo.add(linha);
        }
        reader.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Erro ao ler o arquivo", "Atenção",
            JOptionPane.WARNING_MESSAGE);
    }
    return conteudo;
}
```

Figura 19. Tela do Código que lê os Arquivos Textos

4.2.5. Conclusão da Migração entre os SGBD

Migradas as estruturas das tabelas, bem como seus dados, será mostrado em uma mensagem de conclusão da migração, conforme ilustrado na Figura 20.



Figura 20. Tela de Conclusão da Migração

5. ESTUDO DE CASO

Neste capítulo será realizado um Estudo de Caso para a validação da proposta de trabalho presente. Para melhor entendimento, esse estudo de caso será desenvolvido com base em um modelo de dados de um sistema de consultório médico. A migração será de uma bases de dados Oracle para MySQL.

Na Figura 21, é ilustrada parte do Modelo de Dados para o qual será feito o processo de migração de dados.

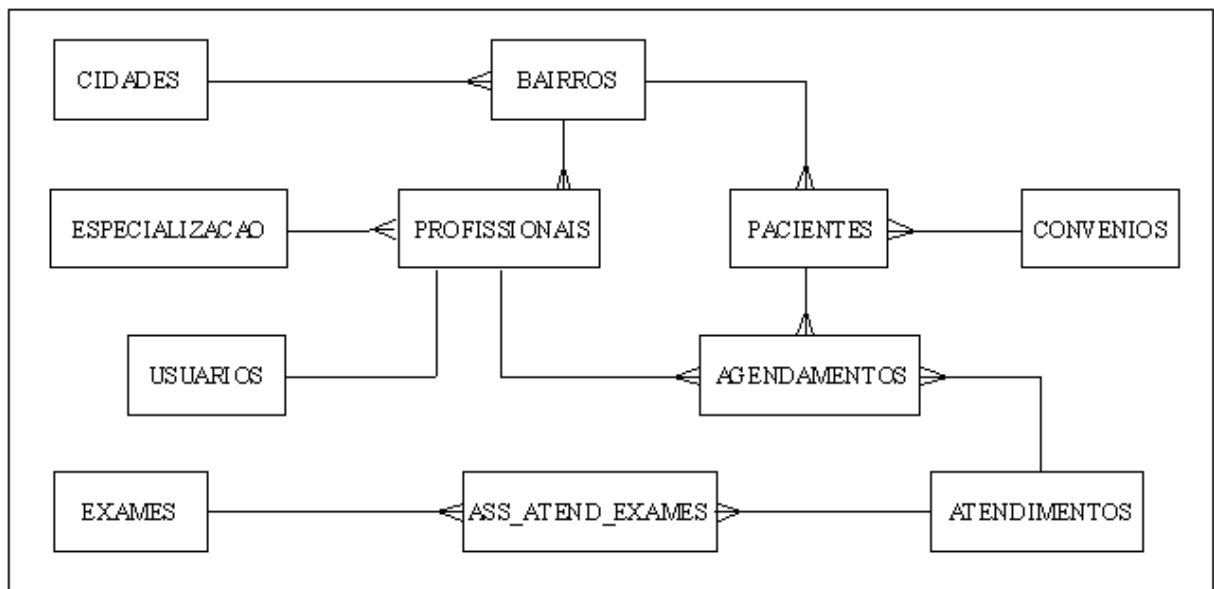


Figura 21. Modelo de Dados do Sistema de Consultório

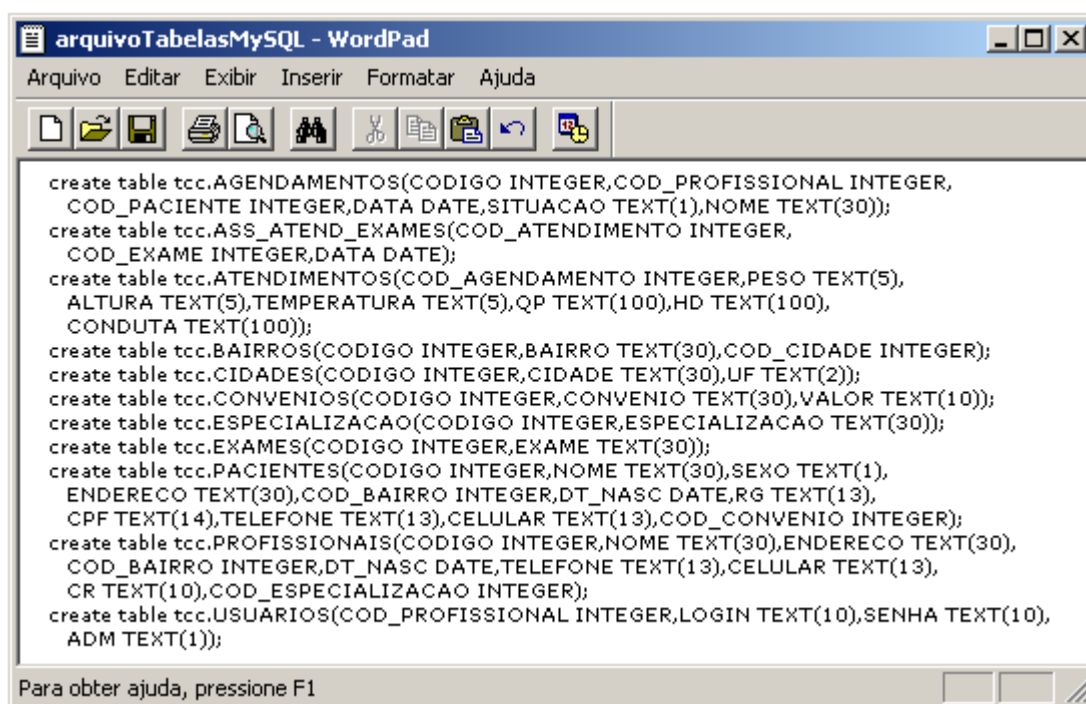
Após ser visualizados todas as tabelas e suas relações será feito um levantamento dos tipos dos dados das tabelas, conforme ilustrado na Figura 22.

AGENDAMENTOS		ESPECIALIZACAO	
CODIGO	NUMBER(6)	CODIGO	NUMBER(3)
COD_PROFISSIONAL	NUMBER(3)	ESPECIALIZACAO	VARCHAR2(30)
COD_PACIENTE	NUMBER(3)		
DATA	DATE	EXAMES	
SITUACAO	VARCHAR2(1)	CODIGO	NUMBER(3)
NOME	VARCHAR2(30)	EXAME	VARCHAR2(30)
ASS_ATEND_EXAMES			
COD_ATENDIMENTO	NUMBER(6)	ATENDIMENTOS	
COD_EXAME	NUMBER(3)	CODIGO	NUMBER(3)
DATA	DATE	NOME	VARCHAR2(30)
		SEXO	VARCHAR2(1)
ATENDIMENTOS		ENDereco	VARCHAR2(30)
COD_AGENDAMENTO	NUMBER(6)	COD_BAIRRO	NUMBER(3)
PESO	VARCHAR2(5)	DT_NASC	DATE
ALTURA	VARCHAR2(5)	RG	VARCHAR2(13)
TEMPERATURA	VARCHAR2(5)	CPF	VARCHAR2(14)
QP	VARCHAR2(100)	TELEFONE	VARCHAR2(13)
HD	VARCHAR2(100)	CELULAR	VARCHAR2(13)
CONDUTA	VARCHAR2(100)	COD_CONVENIO	NUMBER(3)
BAIRROS		PROFISSIONAIS	
CODIGO	NUMBER(3)	CODIGO	NUMBER(3)
BAIRRO	VARCHAR2(30)	NOME	VARCHAR2(30)
COD_CIDADE	NUMBER(3)	ENDereco	VARCHAR2(30)
		COD_BAIRRO	NUMBER(3)
CIDADES		DT_NASC	DATE
CODIGO	NUMBER(3)	TELEFONE	VARCHAR2(13)
CIDADE	VARCHAR2(30)	CELULAR	VARCHAR2(13)
UF	VARCHAR2(2)	CR	VARCHAR2(10)
		COD_ESPECIALIZACAO	NUMBER(3)
CONVENIOS			
CODIGO	NUMBER(3)	USUARIOS	
CONVENIO	VARCHAR2(30)	COD_PROFISSIONAL	NUMBER(3)
VALOR	NUMBER(6,2)	LOGIN	VARCHAR2(10)
		SENHA	VARCHAR2(10)
		ADM	VARCHAR2(1)

Figura 22. Estrutura das Tabelas

5.1. APLICAÇÃO DA ETAPA 1 DA PROPOSTA DE TRABALHO

Nesta sessão será criado o arquivo texto referente as tabelas. Para isto, é feita uma busca no banco de dados origem dos atributos e tipos, onde será criado um arquivo texto contendo-as em instruções SQL do tipo DDL, conforme ilustrado pela Figura 23.



```
create table tcc.AGENDAMENTOS(CODIGO INTEGER,COD_PROFISSIONAL INTEGER,
  COD_PACIENTE INTEGER,DATA DATE,SITUACAO TEXT(1),NOME TEXT(30));
create table tcc.ASS_ATEND_EXAMES(COD_ATENDIMENTO INTEGER,
  COD_EXAME INTEGER,DATA DATE);
create table tcc.ATENDIMENTOS(COD_AGENDAMENTO INTEGER,PESO TEXT(5),
  ALTURA TEXT(5),TEMPERATURA TEXT(5),QP TEXT(100),HD TEXT(100),
  CONDUTA TEXT(100));
create table tcc.BAIRROS(CODIGO INTEGER,BAIRRO TEXT(30),COD_CIDADE INTEGER);
create table tcc.CIDADES(CODIGO INTEGER,CIDADE TEXT(30),UF TEXT(2));
create table tcc.CONVENIOS(CODIGO INTEGER,CONVENIO TEXT(30),VALOR TEXT(10));
create table tcc.ESPECIALIZACAO(CODIGO INTEGER,ESPECIALIZACAO TEXT(30));
create table tcc.EXAMES(CODIGO INTEGER,EXAME TEXT(30));
create table tcc.PACIENTES(CODIGO INTEGER,NOME TEXT(30),SEXO TEXT(1),
  ENDereco TEXT(30),COD_BAIRRO INTEGER,DT_NASC DATE,RG TEXT(13),
  CPF TEXT(14),TELEFONE TEXT(13),CELULAR TEXT(13),COD_CONVENIO INTEGER);
create table tcc.PROFISSIONAIS(CODIGO INTEGER,NOME TEXT(30),ENDERECO TEXT(30),
  COD_BAIRRO INTEGER,DT_NASC DATE,TELEFONE TEXT(13),CELULAR TEXT(13),
  CR TEXT(10),COD_ESPECIALIZACAO INTEGER);
create table tcc.USUARIOS(COD_PROFISSIONAL INTEGER,LOGIN TEXT(10),SENHA TEXT(10),
  ADM TEXT(1));
```

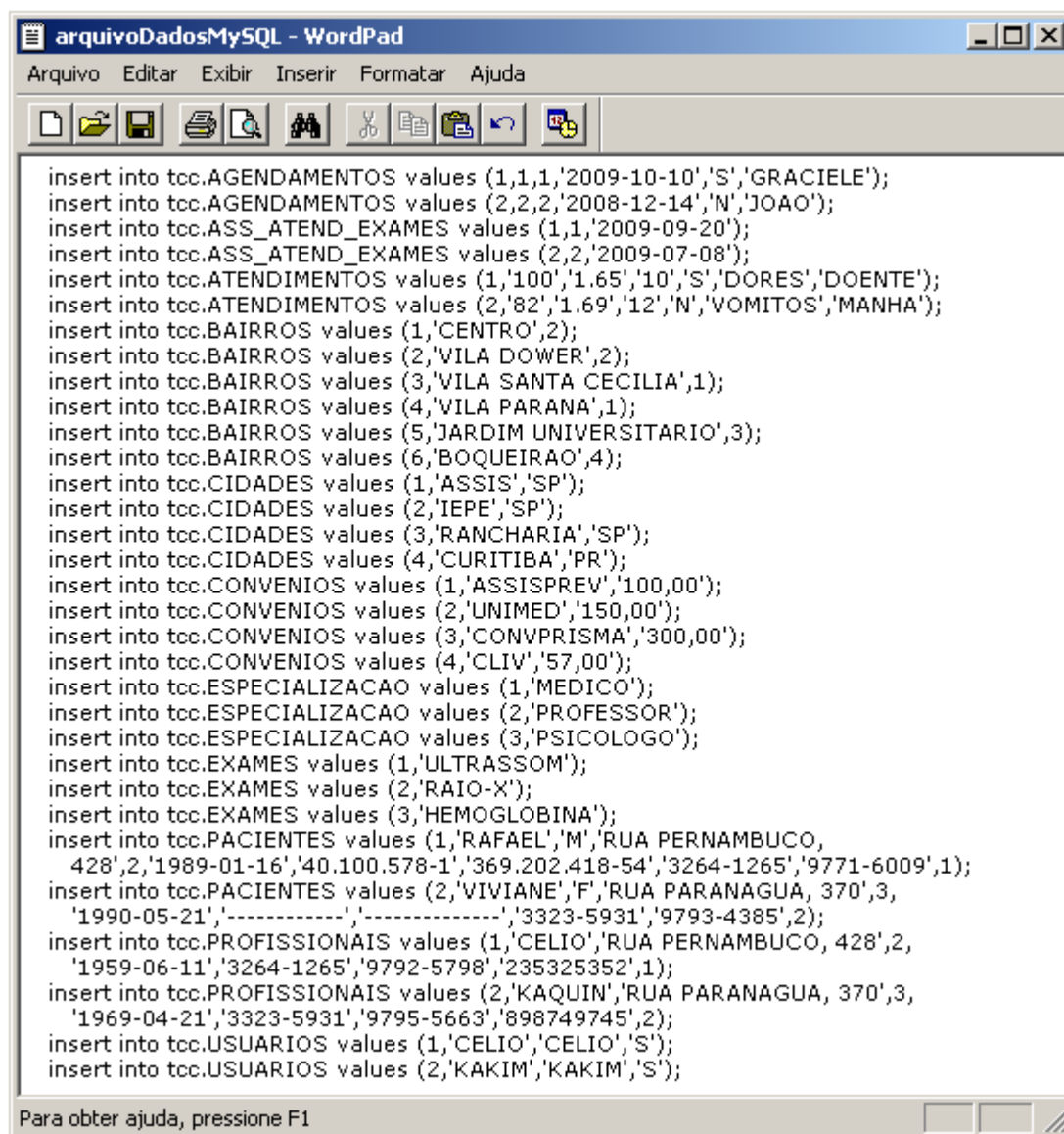
Figura 23. Arquivo Texto das Tabelas

Após a criação do arquivo texto das tabelas, este deve ser executado pelo banco de dados destino para que as tabelas sejam criadas.

5.2. APLICAÇÃO DA ETAPA 2 DA PROPOSTA DE TRABALHO

Nesta sessão será criado o arquivo texto referente aos Dados. Para isto, é feita uma busca no banco de dados origem dos dados, onde será criado um arquivo texto

contendo-as em instruções SQL do tipo DML. Para tal criação, primeiramente a “aplicação da etapa 1” deve ter sido realizada, conforme ilustrado na Figura 24.



```

insert into tcc.AGENDAMENTOS values (1,1,1,'2009-10-10','S','GRACIELE');
insert into tcc.AGENDAMENTOS values (2,2,2,'2008-12-14','N','JOAO');
insert into tcc.ASS_ATEND_EXAMES values (1,1,'2009-09-20');
insert into tcc.ASS_ATEND_EXAMES values (2,2,'2009-07-08');
insert into tcc.ATENDIMENTOS values (1,'100','1.65','10','S','DORES','DOENTE');
insert into tcc.ATENDIMENTOS values (2,'82','1.69','12','N','VOMITOS','MANHA');
insert into tcc.BAIRROS values (1,'CENTRO',2);
insert into tcc.BAIRROS values (2,'VILA DOWER',2);
insert into tcc.BAIRROS values (3,'VILA SANTA CECILIA',1);
insert into tcc.BAIRROS values (4,'VILA PARANA',1);
insert into tcc.BAIRROS values (5,'JARDIM UNIVERSITARIO',3);
insert into tcc.BAIRROS values (6,'BOQUEIRAO',4);
insert into tcc.CIDADES values (1,'ASSIS','SP');
insert into tcc.CIDADES values (2,'IEPE','SP');
insert into tcc.CIDADES values (3,'RANCHARIA','SP');
insert into tcc.CIDADES values (4,'CURITIBA','PR');
insert into tcc.CONVENIOS values (1,'ASSISPREV','100,00');
insert into tcc.CONVENIOS values (2,'UNIMED','150,00');
insert into tcc.CONVENIOS values (3,'CONVPRISMA','300,00');
insert into tcc.CONVENIOS values (4,'CLIV','57,00');
insert into tcc.ESPECIALIZACAO values (1,'MEDICO');
insert into tcc.ESPECIALIZACAO values (2,'PROFESSOR');
insert into tcc.ESPECIALIZACAO values (3,'PSICOLOGO');
insert into tcc.EXAMES values (1,'ULTRASSOM');
insert into tcc.EXAMES values (2,'RAIO-X');
insert into tcc.EXAMES values (3,'HEMOGLOBINA');
insert into tcc.PACIENTES values (1,'RAFAEL','M','RUA PERNAMBUCO,
428',2,'1989-01-16','40.100.578-1','369.202.418-54','3264-1265','9771-6009',1);
insert into tcc.PACIENTES values (2,'VIVIANE','F','RUA PARANAGUA, 370',3,
'1990-05-21','-----','-----','3323-5931','9793-4385',2);
insert into tcc.PROFISSIONAIS values (1,'CELIO','RUA PERNAMBUCO, 428',2,
'1959-06-11','3264-1265','9792-5798','235325352',1);
insert into tcc.PROFISSIONAIS values (2,'KAQUIN','RUA PARANAGUA, 370',3,
'1969-04-21','3323-5931','9795-5663','898749745',2);
insert into tcc.USUARIOS values (1,'CELIO','CELIO','S');
insert into tcc.USUARIOS values (2,'KAKIM','KAKIM','S');

```

Figura 24. Arquivo Texto dos Dados

Após a criação do arquivo texto dos dados, este deve ser executado pelo banco de dados destino para que os dados sejam inseridos em suas tabelas correspondentes.

5.3. CONCLUSÃO DA MIGRAÇÃO

Após serem aplicadas as etapas 1 e 2 em ordem, a migração do consultório de Oracle para MySQL será concluída. Para conferir se o resultado foi o esperado, aconselha-se fazer uma “SELECT” nas tabelas do banco de dados destino.

Na Figura 25, é ilustrada uma “SELECT” na tabela “CIDADE” do banco de dados destino.

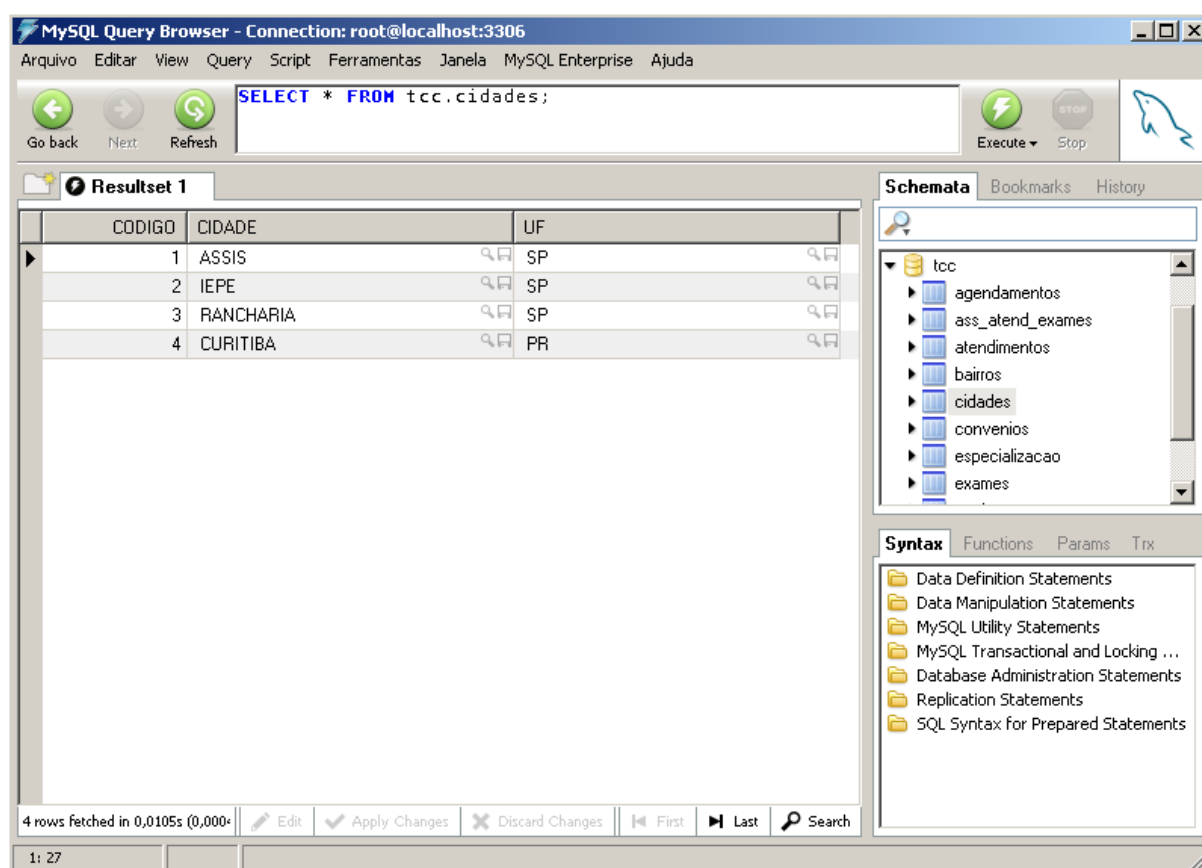


Figura 25. Select da Conclusão da Migração

6. CONSIDERAÇÕES FINAIS E PROJEÇÕES FUTURAS

Este trabalho teve por objetivo principal mostrar de que forma é realizada a migração entre SGBD e os níveis existentes de migrações, pois como a falta de materiais que abordam tal tema é muito grande, pretende-se que este trabalho possa auxiliar pessoas que queiram estudar tal assunto ou até mesmo dar continuidade.

Com este trabalho foi possível mostrar as funções de criação e leitura de arquivos texto, usados para armazenar as SQL de criação e inserção de tabelas e registros para executá-las no BD destino. Também foi desenvolvida uma ferramenta que auxilie pessoas que queiram migrar suas bases de dados de Oracle para MySQL e PostgreSQL..

Uma das principais vantagens da migração é sem dúvida o “custo”, pois é possível migrar um banco de dados pago, como o Oracle, em um banco de dados gratuito, como o MySQL e PostgreSQL.

Durante o desenvolvimento deste trabalho foram encontrados vários problemas, entre eles estão às incompatibilidades dos tipos de dados, que no momento da migração de um SGBD para outro, os atributos do SGBD origem eram de tipos incompatíveis com o SGBD destino. Outro problema foi à criação de arquivos texto, que no momento da criação os arquivos não estavam sendo gravados corretamente.

Como proposta de trabalho futuro, espera-se migrar as tabelas com restrições, como por exemplo, chaves primárias, chaves estrangeiras, entre outras. Espera-se migrar alterações no código, como funções, triggers, entre outras. Poderá também haver uma variedade maior de Sistema Gerenciador de Banco de Dados para migração.

REFERÊNCIAS

DEITEL, Harvey M.; DEITEL, Paul J. **Java: Como Programar**. 6ª edição – Nacional: Prentice-Hall, 2005.

Documentação do PostgreSQL 8.0.0. Disponível em : <http://pgdocptbr.sourceforge.net/pg80/index.html>. Acesso em 02 Outubro 2009.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 3. ed. Porto Alegre: Editora Sagra Luzzato (Instituto de Informática da UFRGS), 1999.

Manual de Referência do MySQL 4.1. Disponível em : <http://dev.mysql.com/doc/refman/4.1/pt/index.html>. Acesso em 02 Outubro 2009.

NAUGHTON, Patrick. **Dominando Java**. Editora Makron Books, 1996.

POLETTTO, Alex Sandro Romeo de Souza. **Projeto de Banco de Dados – Modelando e Construindo Bancos de Dados em Oracle e MySQL**, (Centro de Pesquisa em Informática da FEMA), 2009.

_____. **Banco de Dados I**. Fundação Educacional do Município de Assis, 2008.

PRICE, Jason. **Oracle Database 11g SQL**. Tradução de João Eduardo Nóbrega Tortello. Porto Alegre; Bookmark, 2009.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. Tradução de Daniel Vieira. 5ª edição – Rio de Janeiro: Elsevier, 2006.

TEOREY, Toby; LIGHTSTONE, Sam; NADEAU, Tom. **Projeto e Modelagem de Banco de Dados**. 4. ed. Tradução de Daniel Vieira. Rio de Janeiro: Editora Elsevier, 2007.

YONG, Chu Shao. **Banco de Dados: organização, sistema e administração**. São Paulo: Atlas, 1990.

ANEXO

A seguir é mostrado a tela de conexão aos bancos de dados Oracle, MySQL e PostgreSQL.

```
package view;
```

```
import conexao.ConnectionMySQL;
```

```
import conexao.ConnectionOracle;
```

```
import conexao.ConnectionPostgreSQL;
```

```
import javax.swing.JOptionPane;
```

```
public class Tela extends javax.swing.JFrame {
```

```
    private ConnectionOracle connectionOracle;
```

```
    private ConnectionMySQL connectionMySQL;
```

```
    private ConnectionPostgreSQL connectionPostgreSQL;
```

```
    private boolean ora,my,post;
```

```
    public Tela() {
```

```
        initComponents();
```

```
        setLocationRelativeTo(null);
```

```
        connectionOracle = ConnectionOracle.getInstance();
```

```
        connectionMySQL = ConnectionMySQL.getInstance();
```

```
        connectionPostgreSQL = ConnectionPostgreSQL.getInstance();
```

```
    }
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
```

```
private void initComponents() {
```

```
    jPanel3 = new javax.swing.JPanel();
```

```
    jButton1 = new javax.swing.JButton();
```

```
    jTextField1 = new javax.swing.JTextField();
```

```
    jTextField2 = new javax.swing.JTextField();
```

```
    jTextField3 = new javax.swing.JTextField();
```

```
    jLabel1 = new javax.swing.JLabel();
```

```
    jLabel2 = new javax.swing.JLabel();
```

```
    jLabel3 = new javax.swing.JLabel();
```

```
    jLabel4 = new javax.swing.JLabel();
```

```
    jLabel8 = new javax.swing.JLabel();
```

```
    jTextField4 = new javax.swing.JTextField();
```

```
    jLabel9 = new javax.swing.JLabel();
```

```
    jLabel5 = new javax.swing.JLabel();
```

```
    jTextField5 = new javax.swing.JTextField();
```

```
    jTextField6 = new javax.swing.JTextField();
```

```
    jLabel6 = new javax.swing.JLabel();
```

```
    jLabel7 = new javax.swing.JLabel();
```

```
    jLabel10 = new javax.swing.JLabel();
```

```
    jTextField7 = new javax.swing.JTextField();
```

```
    jTextField8 = new javax.swing.JTextField();
```

```

jButton2 = new javax.swing.JButton();

jTextField10 = new javax.swing.JTextField();

jLabel14 = new javax.swing.JLabel();

jLabel15 = new javax.swing.JLabel();

jTextField11 = new javax.swing.JTextField();

jLabel11 = new javax.swing.JLabel();

jTextField9 = new javax.swing.JTextField();

jLabel13 = new javax.swing.JLabel();

jTextField12 = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("MigraÃ§Ã£o de SGBD");

setResizable(false);

setUndecorated(true);


jPanel3.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,
0), 1, true));


jButton1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N

jButton1.setText("Conectar");

jButton1.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        jButton1MouseClicked(evt);
    }
});

```

```

    }
});

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12));
jLabel1.setText("ORACLE");
jLabel2.setText("IP:");
jLabel3.setText("USUARIO:");
jLabel4.setText("DATABASE:");
jLabel8.setFont(new java.awt.Font("Tahoma", 1, 12));
jLabel8.setText("PostgreSQL");
jLabel9.setText("SENHA:");
jLabel5.setText("IP:");
jLabel6.setText("DATABASE:");
jLabel7.setText("USUARIO:");
jLabel10.setText("SENHA:");
jButton2.setText("Sair");
jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton2MouseClicked(evt);
    }
});

jLabel14.setText("USUARIO:");
jLabel15.setText("SENHA:");
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 12));
jLabel11.setText("MYSQL");

```

```

jLabel13.setText("DATABASE:");

jLabel12.setText("IP:");


    javax.swing.GroupLayout jPanel3Layout = new
    javax.swing.GroupLayout(jPanel3);

    jPanel3.setLayout(jPanel3Layout);

    jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel3Layout.createSequentialGroup()

            .addGap(10, 10, 10)

            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addGroup(jPanel3Layout.createSequentialGroup()

                    .addGap(10, 10, 10)

                    .addComponent(jLabel1)

                    .addGap(10, 10, 10)

                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                        .addGroup(jPanel3Layout.createSequentialGroup()

                            .addGap(10, 10, 10)

                            .addComponent(jLabel9)

                            .addGap(10, 10, 10)

                            .addComponent(jLabel4)

                        )

                    )

                )

            )

        )

    )

    )

    )

```

`.addComponent(jLabel2)`

`.addComponent(jLabel3))`

`.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)`

`.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)`

`.addComponent(jTextField4)`

`.addComponent(jTextField3)`

`.addComponent(jTextField2,
javax.swing.GroupLayout.DEFAULT_SIZE, 94, Short.MAX_VALUE)`

`.addComponent(jTextField1))))`

`.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)`

`.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)`

`.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)`

`.addGroup(jPanel3Layout.createSequentialGroup()`

`.addGap(14, 14, 14)`

`.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)`

`.addGroup(jPanel3Layout.createSequentialGroup()`

`.addGap(6, 6, 6)`

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addComponent(jLabel5)
```

```
.addComponent(jLabel10)
```

```
.addComponent(jLabel7)))
```

```
.addComponent(jLabel6))
```

```
.addGap(4, 4, 4)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
.addComponent(jTextField6)
```

```
.addComponent(jTextField7)
```

```
.addComponent(jTextField8,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addComponent(jTextField5)))
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
jPanel3Layout.createSequentialGroup())
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addComponent(jButton1)))
```

```
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,  
jPanel3Layout.createSequentialGroup())
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addComponent(jLabel11)
```

```
.addGap(30, 30, 30)))
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
.addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addGap(10, 10, 10)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
```

```
.addComponent(jLabel15)
```

```
.addComponent(jLabel13)
```

```
.addComponent(jLabel12)
```

```
.addComponent(jLabel14))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
.addComponent(jTextField12)
```

```
.addComponent(jTextField11)
```

```
.addComponent(jTextField10)
```

```
.addComponent(jTextField9,  
javax.swing.GroupLayout.DEFAULT_SIZE, 89, Short.MAX_VALUE)))
```

```
.addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addGap(79, 79, 79)
```

```
.addComponent(jLabel8)))
```

```
.addContainerGap(81, Short.MAX_VALUE))
```

```

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

        .addComponent(jButton2)

        .addContainerGap()))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel3Layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(jButton2)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
CENTER)

        .addComponent(jLabel8)

        .addComponent(jLabel11)

        .addComponent(jLabel1))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
CENTER)

        .addComponent(jTextField9,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```
.addComponent(jLabel12)

.addComponent(jTextField5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel5)

.addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
CENTER)

.addComponent(jLabel4)

.addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel6)

.addComponent(jTextField6,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addComponent(jLabel13)

.addComponent(jTextField10,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
CENTER)
```

```
        .addComponent(jTextField11,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel14)
```

```
        .addComponent(jTextField7,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel7)
```

```
        .addComponent(jTextField3,  
        javax.swing.GroupLayout.PREFERRED_SIZE,  
        javax.swing.GroupLayout.DEFAULT_SIZE,  
        javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addComponent(jLabel3))
```

```
.addGap(7, 7, 7)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
CENTER)
```

```
        .addComponent(jTextField12,  
        javax.swing.GroupLayout.PREFERRED_SIZE,
```

```

    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel15)

        .addComponent(jTextField8,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel10)

        .addComponent(jTextField4,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(jLabel9))

    .addGap(23, 23, 23)

    .addComponent(jButton1)

    .addContainerGap(24, Short.MAX_VALUE))

);

```

```

    javax.swing.GroupLayout layout = new
    javax.swing.GroupLayout(getContentPane());

    getContentPane().setLayout(layout);

    layout.setHorizontalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
    javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

        .addContainerGap()

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addComponent(jPanel3,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

    );

```

```

    pack();

} // </editor-fold> //GEN-END: initComponents

```

```

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton1MouseClicked

```

```

    // Conecta Banco ORACLE

```

```

    try {

        connectionOracle.conectar(jTextField1.getText(), jTextField2.getText(),
            jTextField3.getText(), jTextField4.getText());

        ora = true;

    } catch (Exception ex) {

```

```

        JOptionPane.showMessageDialog(null, "Erro ao Conectar Banco Oracle",
"MigraSGBD",

        JOptionPane.INFORMATION_MESSAGE);

        ora = false;
    }

    // Conecta Banco MySQL

    try {

        connectionMySQL.conectar(jTextField5.getText(), jTextField6.getText(),

            jTextField7.getText(),jTextField8.getText());

        my = true;
    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Erro ao Conectar Banco MySQL",
"MigraSGBD",

        JOptionPane.INFORMATION_MESSAGE);

        my = false;
    }

    // Conecta Banco PostgreSQL

    try {

        connectionPostgreSQL.conectar(jTextField9.getText(), jTextField10.getText(),

            jTextField11.getText(),jTextField12.getText());

        post = true;
    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Erro ao Conectar Banco
PostgreSQL", "MigraSGBD",

        JOptionPane.INFORMATION_MESSAGE);

        post = false;
    }

```

```
}
```

```
if (ora) {
```

```
    if (my == false && post == false) {
```

```
        JOptionPane.showMessageDialog(null, "Apenas banco Oracle conectado",  
"MigraSGBD",
```

```
        JOptionPane.INFORMATION_MESSAGE);
```

```
    }else if(my == true && post == false) {
```

```
        JOptionPane.showMessageDialog(null, "Banco Oracle e MySQL  
conectado com sucesso", "MigraSGBD",
```

```
        JOptionPane.INFORMATION_MESSAGE);
```

```
    } else if(my == false && post == true) {
```

```
        JOptionPane.showMessageDialog(null, "Banco Oracle e PostgreSQL  
conectado com sucesso", "MigraSGBD",
```

```
        JOptionPane.INFORMATION_MESSAGE);
```

```
    } else if(my == true && post == true) {
```

```
        JOptionPane.showMessageDialog(null, "Banco Oracle, PostgreSQL e  
MySQL conectado com sucesso", "MigraSGBD",
```

```
        JOptionPane.INFORMATION_MESSAGE);
```

```
    }
```

```
setVisible(false);
```

```
Inicial start = new Inicial(jTextField2.getText(),my,post);
```

```
start.setVisible(true);
```

```
}else
```

```
JOptionPane.showMessageDialog(null, "Banco Oracle não Conectado",  
"MigraSGBD",
```

```
JOptionPane.INFORMATION_MESSAGE);
```

```
//GEN-LAST:event_jButton1MouseClicked
```

```
private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {//GEN-  
FIRST:event_jButton2MouseClicked
```

```
System.exit(0);
```

```
//GEN-LAST:event_jButton2MouseClicked
```

```
// Variables declaration - do not modify//GEN-BEGIN:variables
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JButton jButton2;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel10;
```

```
private javax.swing.JLabel jLabel11;
```

```
private javax.swing.JLabel jLabel12;
```

```
private javax.swing.JLabel jLabel13;
```

```
private javax.swing.JLabel jLabel14;
```

```
private javax.swing.JLabel jLabel15;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JLabel jLabel5;
```

```
private javax.swing.JLabel jLabel6;
```

```
private javax.swing.JLabel jLabel7;  
private javax.swing.JLabel jLabel8;  
private javax.swing.JLabel jLabel9;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JTextField jTextField1;  
private javax.swing.JTextField jTextField10;  
private javax.swing.JTextField jTextField11;  
private javax.swing.JTextField jTextField12;  
private javax.swing.JTextField jTextField2;  
private javax.swing.JTextField jTextField3;  
private javax.swing.JTextField jTextField4;  
private javax.swing.JTextField jTextField5;  
private javax.swing.JTextField jTextField6;  
private javax.swing.JTextField jTextField7;  
private javax.swing.JTextField jTextField8;  
private javax.swing.JTextField jTextField9;  
// End of variables declaration//GEN-END:variables
```

```
}
```

A seguir é mostrado a tela de migração das tabelas e dados dos bancos de dados Oracle para MySQL e PostgreSQL.

```
package view;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import conexao.ConnectionMySQL;
```

```
import conexao.ConnectionOracle;
```

```
import conexao.ConnectionPostgreSQL;
```

```
import java.io.BufferedReader;
```

```
import java.io.FileReader;
```

```
import java.io.FileWriter;
```

```
import java.sql.SQLException;
```

```
import java.util.ArrayList;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.SwingConstants;
```

```
import javax.swing.table.DefaultTableCellRenderer;
```

```
import javax.swing.table.DefaultTableModel;
```

```
/**
```

```
*
```

```
* @author Rafael Bevilacqua
```

```
*/
```

```
public class Inicial extends javax.swing.JFrame {
```

```
private boolean tabelasm = false, tabelasp = false;

private boolean dadosm = false, dadosp = false;

private boolean my, post;

private String bancoOracle;

private PreparedStatement stmt;

private ConnectionOracle connectionOracle;

private ConnectionMySQL connectionMySQL;

private ConnectionPostgreSQL connectionPostgreSQL;

private DefaultTableModel modelo = new DefaultTableModel();

private String[] titulos = {"", "Tabelas"};

DefaultTableCellRenderer rendererCentro = new DefaultTableCellRenderer();


public Inicial(String bancoOracle, boolean my, boolean post) {

    initComponents();

    setLocationRelativeTo(null);


    this.my = my;

    this.post = post;

    this.bancoOracle = bancoOracle;

    connectionOracle = ConnectionOracle.getInstance();

    connectionMySQL = ConnectionMySQL.getInstance();

    connectionPostgreSQL = ConnectionPostgreSQL.getInstance();

    rendererCentro.setHorizontalAlignment(SwingConstants.CENTER);

}
```

```
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
```

```
private void initComponents() {
```

```
    jPanel1 = new javax.swing.JPanel();
```

```
    jPanel3 = new javax.swing.JPanel();
```

```
    jButton1 = new javax.swing.JButton();
```

```
    jButton2 = new javax.swing.JButton();
```

```
    jButton5 = new javax.swing.JButton();
```

```
    jButton6 = new javax.swing.JButton();
```

```
    jButton4 = new javax.swing.JButton();
```

```
    jPanel4 = new javax.swing.JPanel();
```

```
    jButton7 = new javax.swing.JButton();
```

```
    jButton8 = new javax.swing.JButton();
```

```
    jButton9 = new javax.swing.JButton();
```

```
    jButton10 = new javax.swing.JButton();
```

```
    jLabel2 = new javax.swing.JLabel();
```

```
    jLabel3 = new javax.swing.JLabel();
```

```
    jPanel5 = new javax.swing.JPanel();
```

```
    jScrollPane1 = new javax.swing.JScrollPane();
```

```
    jTable1 = new javax.swing.JTable();
```

```
    jLabel1 = new javax.swing.JLabel();
```

```
    jButton3 = new javax.swing.JButton();
```

```
    jButton11 = new javax.swing.JButton();
```

```
    jButton12 = new javax.swing.JButton();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);  
setTitle("Migração de SGBD");  
setResizable(false);  
setUndecorated(true);
```

```
jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,  
0), 1, true));
```

```
jPanel3.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,  
0), 1, true));
```

```
jButton1.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N
```

```
jButton1.setText("Gerar arquivo das Tabelas");
```

```
jButton1.setEnabled(false);
```

```
jButton1.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        jButton1MouseClicked(evt);
```

```
    }
```

```
});
```

```
jButton2.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N
```

```
jButton2.setText("Gerar arquivo dos Dados");
```

```
jButton2.setEnabled(false);
```

```
jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        jButton2MouseClicked(evt);  
    }  
});
```

```
        jButton5.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N  
        jButton5.setText("Migrar Tabelas");  
        jButton5.setEnabled(false);  
        jButton5.addMouseListener(new java.awt.event.MouseAdapter() {  
            public void mouseClicked(java.awt.event.MouseEvent evt) {  
                jButton5MouseClicked(evt);  
            }  
        });
```

```
        jButton6.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N  
        jButton6.setText("Migrar Dados");  
        jButton6.setEnabled(false);  
        jButton6.addMouseListener(new java.awt.event.MouseAdapter() {  
            public void mouseClicked(java.awt.event.MouseEvent evt) {  
                jButton6MouseClicked(evt);  
            }  
        });
```

```
        javax.swing.GroupLayout jPanel3Layout = new  
javax.swing.GroupLayout(jPanel3);  
        jPanel3.setLayout(jPanel3Layout);
```

```

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel3Layout.createSequentialGroup()

        .addContainerGap()

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
        RAILING, false)

        .addComponent(jButton2,
            javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jButton1,
            javax.swing.GroupLayout.Alignment.LEADING,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
        EADING)

        .addComponent(jButton6, javax.swing.GroupLayout.DEFAULT_SIZE,
            205, Short.MAX_VALUE)

        .addComponent(jButton5, javax.swing.GroupLayout.DEFAULT_SIZE,
            205, Short.MAX_VALUE))

        .addContainerGap())

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(jPanel3Layout.createSequentialGroup()

        .addContainerGap()

```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
BASELINE)
```

```
    .addComponent(jButton1)
```

```
    .addComponent(jButton5))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
BASELINE)
```

```
    .addComponent(jButton2)
```

```
    .addComponent(jButton6))
```

```
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE))
```

```
);
```

```
jButton4.setText("Sair");
```

```
jButton4.setMargin(new java.awt.Insets(2, 5, 2, 5));
```

```
jButton4.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        jButton4MouseClicked(evt);
```

```
    }
```

```
});
```

```
jPanel4.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,  
0), 1, true));
```

```
jButton7.setIcon(new  
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N
```

```
jButton7.setText("Gerar arquivo das Tabelas");
```

```
jButton7.setEnabled(false);

jButton7.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        jButton7MouseClicked(evt);

    }

});
```

```
jButton8.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N

jButton8.setText("Gerar arquivo dos Dados");

jButton8.setEnabled(false);

jButton8.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        jButton8MouseClicked(evt);

    }

});
```

```
jButton9.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N

jButton9.setText("Migrar Tabelas");

jButton9.setEnabled(false);

jButton9.addMouseListener(new java.awt.event.MouseAdapter() {

    public void mouseClicked(java.awt.event.MouseEvent evt) {

        jButton9MouseClicked(evt);

    }

});
```

```

        jButton10.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/imagens/raio.png"))); // NOI18N

        jButton10.setText("Migrar Dados");

        jButton10.setEnabled(false);

        jButton10.addMouseListener(new java.awt.event.MouseAdapter() {

            public void mouseClicked(java.awt.event.MouseEvent evt) {

                jButton10MouseClicked(evt);

            }

        });

```

```

        javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);

```

```

        jPanel4.setLayout(jPanel4Layout);

        jPanel4Layout.setHorizontalGroup(

```

```

        jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel4Layout.createSequentialGroup()

                .add(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addGroup(jPanel4Layout.createSequentialGroup()

                        .add(jPanel4Layout.createSequentialGroup()

                            .addContainerGap()

```

```

                            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

```

```

                                .addComponent(jButton8,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

```

```

                                .addComponent(jButton7,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(jButton10, javax.swing.GroupLayout.DEFAULT_SIZE, 205, Short.MAX_VALUE)
```

```
    .addComponent(jButton9, javax.swing.GroupLayout.DEFAULT_SIZE, 205, Short.MAX_VALUE))
```

```
    .addContainerGap())
```

```
);
```

```
jPanel4Layout.setVerticalGroup(
```

```
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(jPanel4Layout.createSequentialGroup())
```

```
    .addContainerGap()
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jButton7)
```

```
    .addComponent(jButton9))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
    .addComponent(jButton8)
```

```
    .addComponent(jButton10))
```

```
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```
);
```

```
jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14));
```

```
jLabel2.setText("MySQL");
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14));
```

```
jLabel3.setText("PostgreSQL");
```

```
jPanel5.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0), 1, true));
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(  
    new Object [][] {
```

```
    },
```

```
    },
```

```
    new String [] {
```

```
    }  
));
```

```
jScrollPane1.setViewportView(jTable1);
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14));
```

```
jLabel1.setText("Oracle");
```

```
jButton3.setText("Visualizar Tabelas");
```

```
jButton3.setMargin(new java.awt.Insets(0, 0, 0, 0));
```

```
jButton3.addMouseListener(new java.awt.event.MouseAdapter() {
```

```
    public void mouseClicked(java.awt.event.MouseEvent evt) {
```

```
        jButton3MouseClicked(evt);
```

```

    }

});

    javax.swing.GroupLayout jPanel5Layout = new
    javax.swing.GroupLayout(jPanel5);

    jPanel5.setLayout(jPanel5Layout);

    jPanel5Layout.setHorizontalGroup(

        jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                jPanel5Layout.createSequentialGroup()

                    .addGap()

                    .addComponent(jLabel1)

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
                        javax.swing.GroupLayout.DEFAULT_SIZE,
                        Short.MAX_VALUE)

                    .addComponent(jButton3)

                    .addGap()

                    .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE,
                        278, Short.MAX_VALUE)

                )
            );

    jPanel5Layout.setVerticalGroup(

        jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel5Layout.createSequentialGroup()

                .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
                    CENTER)

```

```
        .addComponent(jLabel1)
        .addComponent(jButton3))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

        .addComponent(jScrollPane1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 229,
    javax.swing.GroupLayout.PREFERRED_SIZE))

    );
```

```
jButton11.setText("Criar Banco");
jButton11.setEnabled(false);
jButton11.setMargin(new java.awt.Insets(0, 0, 0, 0));
jButton11.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton11MouseClicked(evt);
    }
});
```

```
jButton12.setText("Criar Banco");
jButton12.setEnabled(false);
jButton12.setMargin(new java.awt.Insets(0, 0, 0, 0));
jButton12.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jButton12MouseClicked(evt);
    }
});
```

```

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);

        jPanel1.setLayout(jPanel1Layout);

        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

                .addGap(10, 10, 10)

                .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                    .addComponent(jLabel2)

                    .addComponent(jLabel3)

                    .addGap(84, 84, 84)

                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
CENTER)

                        .addComponent(jButton11)

                        .addComponent(jButton12))

                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 141,
Short.MAX_VALUE)

                    .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE,
48, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addGap(10, 10, 10)

                    .addGroup(jPanel1Layout.createSequentialGroup()

```

```

        .addGap(296, 296, 296)

        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addContainerGap(10, Short.MAX_VALUE))

    .addGroup(jPanel1Layout.createSequentialGroup())

        .addGap(296, 296, 296)

        .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addContainerGap(10, Short.MAX_VALUE))

    );

    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())

        .addContainerGap()

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)

        .addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
270, Short.MAX_VALUE)

        .addGroup(jPanel1Layout.createSequentialGroup())

    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)

        .addGroup(jPanel1Layout.createSequentialGroup())

            .addComponent(jButton4)

            .addGap(14, 14, 14))

        .addGroup(jPanel1Layout.createSequentialGroup())

```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
BASELINE)
```

```
    .addComponent(jLabel2)
```

```
    .addComponent(jButton11))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
```

```
    .addComponent(jPanel3,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 23,  
Short.MAX_VALUE)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.  
BASELINE)
```

```
    .addComponent(jLabel3)
```

```
    .addComponent(jButton12))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```

```
    .addComponent(jPanel4,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE)))
```

```
    .addContainerGap())
```

```
);
```

```
    javax.swing.GroupLayout layout = new  
    javax.swing.GroupLayout(getContentPane());
```

```
    getContentPane().setLayout(layout);
```

```
    layout.setHorizontalGroup(
```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

        );

        layout.setVerticalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            );

        pack();

} // </editor-fold> // GEN-END: initComponents

private void jButton3MouseClicked(java.awt.event.MouseEvent evt) { // GEN-
FIRST:event_jButton3MouseClicked

    CarregaTabela();

    jTable1.setEnabled(false);

    jButton3.setEnabled(false);

    if (my) {

        jButton11.setEnabled(true);

    }

    if (post) {

        jButton12.setEnabled(true);

    }

} // GEN-LAST:event_jButton3MouseClicked

```

```
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event_jButton1MouseClicked
```

```
String texto = "";
```

```
try {
```

```
for (int cont = 0; cont < jTable1.getRowCount(); cont++) {
```

```
    stmt = connectionOracle.getPreparedStatement(  
        "select column_name as col, data_type as tp, data_length as tm1,  
        data_precision as tm2, data_scale as tm3 " +  
        "from user_tab_columns where table_name = '" +  
        jTable1.getValueAt(cont, 1) + "'";  
    
```

```
    ResultSet rs = stmt.executeQuery();
```

```
    texto += "create table " + bancoOracle + "." + jTable1.getValueAt(cont, 1) +  
    "(";
```

```
    while (rs.next()) {
```

```
        texto += rs.getString("col") + " ";
```

```
        texto += testaTipoMySQL(rs.getString("tp"), rs.getString("tm1"),  
rs.getString("tm2"), rs.getString("tm3"));
```

```
        texto += ",";
```

```
    }
```

```
    //tira um caracter do texto (,)...e add " ); "
```

```
    texto = texto.substring(0, texto.length() - 1) + ");";
```

```
    if (cont < jTable1.getRowCount() - 1) {
```

```
        texto += "\n";
```

```

    }

    connectionOracle.desconectar();
}

gravaArquivo(texto, "TabelasMySQL");

jButton1.setEnabled(false);

jButton5.setEnabled(true);

} catch (SQLException ex) {

    JOptionPane.showMessageDialog(null, ex.getMessage(),
        "Atenção", JOptionPane.WARNING_MESSAGE);

}

} //GEN-LAST:event_jButton1MouseClicked

private void jButton4MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jButton4MouseClicked

    System.exit(0);

} //GEN-LAST:event_jButton4MouseClicked

private void jButton5MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jButton5MouseClicked

    try {

        ArrayList<String> conteudo = new ArrayList<String>();

        conteudo = leArquivo("TabelasMySQL");

        for (int z = 0; z < conteudo.size(); z++) {

            stmt = connectionMySQL.getPreparedStatement(conteudo.get(z));

```

```

        stmt.execute();

        connectionMySQL.desconectar();
    }

    tabelasm = true;

    JOptionPane.showMessageDialog(null, "Tabelas Migradas", "MigraSGBD",
        JOptionPane.INFORMATION_MESSAGE);

    jButton5.setEnabled(false);

    if (dadosm) {

        jButton6.setEnabled(true);

    }

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "Erro ao gravar o arquivo",
        "AtenÃ§Ã£o",
        JOptionPane.WARNING_MESSAGE);

}

} //GEN-LAST:event_jButton5MouseClicked

private void jButton6MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jButton6MouseClicked

    try {

        ArrayList<String> conteudo = new ArrayList<String>();

        conteudo = leArquivo("DadosMySQL");

        for (int z = 0; z < conteudo.size(); z++) {

            stmt = connectionMySQL.prepareStatement(conteudo.get(z));

            stmt.execute();

```

```

        connectionMySQL.desconectar();
    }

    JOptionPane.showMessageDialog(null, "Dados Migrados", "MigraSGBD",
        JOptionPane.INFORMATION_MESSAGE);

    jButton6.setEnabled(false);

    JOptionPane.showMessageDialog(null, "Banco Oracle -> MySQL\nMigrado
com Sucesso", "MigraSGBD",
        JOptionPane.INFORMATION_MESSAGE);
} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "Erro ao gravar o arquivo",
        "AtenÃ§Ã£o",
        JOptionPane.WARNING_MESSAGE);
}

} //GEN-LAST:event_jButton6MouseClicked

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jButton2MouseClicked

    String texto = "";

    try {

        for (int cont = 0; cont < jTable1.getRowCount(); cont++) {

            stmt = connectionOracle.prepareStatement(

                "select data_type as tp, column_name as nm from user_tab_columns

" +

                "where table_name = " + jTable1.getValueAt(cont, 1) + """);

            ResultSet rs = stmt.executeQuery();

            ArrayList<String> tipTab = new ArrayList<String>();

            ArrayList<String> nmTab = new ArrayList<String>();

```

```

while (rs.next()) {
    tipTab.add(rs.getString("tp"));
    nmTab.add(rs.getString("nm"));
}
connectionOracle.desconectar();

stmt = connectionOracle.getPreparedStatement(
    "select * from " + jTable1.getValueAt(cont, 1));
ResultSet rs2 = stmt.executeQuery();
while (rs2.next()) {
    texto += "insert into " + bancoOracle + "." + jTable1.getValueAt(cont, 1)
+ " values (";
    for (int z = 0; z < tipTab.size(); z++) {
        //testa se Ã© string
        if ("VARCHAR2".equals(tipTab.get(z)) ||
            "VARCHAR".equals(tipTab.get(z)) ||
            "CHAR".equals(tipTab.get(z))) {
            texto += "'" + rs2.getString(nmTab.get(z)) + "'";
        } //testa se Ã© inteiro
        else if ("NUMBER".equals(tipTab.get(z)) ||
            "INT".equals(tipTab.get(z))) {
            texto += rs2.getInt(nmTab.get(z));
        } //testa se Ã© data
        else if ("DATE".equals(tipTab.get(z))) {
            texto += "'" + rs2.getDate(nmTab.get(z)) + "'";
        }
    }
}

```

```

        texto += ",";
    }

    //tira um caracter do texto (,)...e add "; "
    texto = texto.substring(0, texto.length() - 1) + ";\n";
}

connectionOracle.desconectar();
}

texto = texto.substring(0, texto.length() - 1);
dadosm = true;

gravaArquivo(texto, "DadosMySQL");

if (tabelasm) {
    jButton6.setEnabled(true);
}

jButton2.setEnabled(false);
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null, ex.getMessage(),
        "Atenção", JOptionPane.WARNING_MESSAGE);
}

}

} //GEN-LAST:event_jButton2MouseClicked

```

```

private void jButton7MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton7MouseClicked

```

```

    String texto = "";

```

```

    try {

```

```

        for (int cont = 0; cont < jTable1.getRowCount(); cont++) {

```

```

        stmt = connectionOracle.getPreparedStatement(
            "select column_name as col, data_type as tp, data_length as tm1,
            data_precision as tm2, data_scale as tm3 " +
            "from user_tab_columns where table_name = '" +
            jTable1.getValueAt(cont, 1) + "'");

        ResultSet rs = stmt.executeQuery();

        texto += "create table " + jTable1.getValueAt(cont, 1) + "(";
        while (rs.next()) {
            texto += rs.getString("col") + " ";
            texto += testaTipoPostgreSQL(rs.getString("tp"), rs.getString("tm1"),
            rs.getString("tm2"), rs.getString("tm3"));

            texto += ",";
        }
        //tira um caracter do texto (,)...e add " ); "
        texto = texto.substring(0, texto.length() - 1) + ");";

        if (cont < jTable1.getRowCount() - 1) {
            texto += "\n";
        }
        connectionOracle.desconectar();
    }

    gravaArquivo(texto, "TabelasPostgreSQL");
    jButton7.setEnabled(false);

```

```

        jButton9.setEnabled(true);

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, ex.getMessage(),
            "Atenção", JOptionPane.WARNING_MESSAGE);

    }

} //GEN-LAST:event_jButton7MouseClicked

private void jButton8MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton8MouseClicked

    String texto = "";

    try {

        for (int cont = 0; cont < jTable1.getRowCount(); cont++) {

            stmt = connectionOracle.getPreparedStatement(

                "select data_type as tp, column_name as nm from user_tab_columns

" +

                "where table_name = " + jTable1.getValueAt(cont, 1) + " ";

            ResultSet rs = stmt.executeQuery();

            ArrayList<String> tipTab = new ArrayList<String>();

            ArrayList<String> nmTab = new ArrayList<String>();

            while (rs.next()) {

                tipTab.add(rs.getString("tp"));

                nmTab.add(rs.getString("nm"));

            }

            connectionOracle.desconectar();

            stmt = connectionOracle.getPreparedStatement(

```

```

        "select * from " + jTable1.getValueAt(cont, 1));

ResultSet rs2 = stmt.executeQuery();

while (rs2.next()) {

    texto += "insert into " + jTable1.getValueAt(cont, 1) + " values (";

    for (int z = 0; z < tipTab.size(); z++) {

        //testa se Ã© string
        if ("VARCHAR2".equals(tipTab.get(z)) ||
            "VARCHAR".equals(tipTab.get(z)) ||
            "CHAR".equals(tipTab.get(z))) {
            texto += "'" + rs2.getString(nmTab.get(z)) + "'";
        } //testa se Ã© inteiro
        else if ("NUMBER".equals(tipTab.get(z)) ||
            "INT".equals(tipTab.get(z))) {
            texto += rs2.getInt(nmTab.get(z));
        } //testa se Ã© data
        else if ("DATE".equals(tipTab.get(z))) {
            texto += "'" + rs2.getDate(nmTab.get(z)) + "'";
        }

        texto += ",";

    }

    //tira um caracter do texto (,)...e add "; "
    texto = texto.substring(0, texto.length() - 1) + ");\n";

}

connectionOracle.desconectar();

}

```

```

        texto = texto.substring(0, texto.length() - 1);

        dadosp = true;

        gravaArquivo(texto, "DadosPostgreSQL");

        if (tabelasp) {

            jButton10.setEnabled(true);

        }

        jButton8.setEnabled(false);

    } catch (SQLException ex) {

        JOptionPane.showMessageDialog(null, ex.getMessage(),

            "AtenÃ§Ã£o", JOptionPane.WARNING_MESSAGE);

    }

} //GEN-LAST:event_jButton8MouseClicked

```

```

private void jButton9MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton9MouseClicked

```

```

    try {

        ArrayList<String> conteudo = new ArrayList<String>();

        conteudo = leArquivo("TabelasPostgreSQL");

        for (int z = 0; z < conteudo.size(); z++) {

            stmt = connectionPostgreSQL.prepareStatement(conteudo.get(z));

            stmt.execute();

            connectionPostgreSQL.desconectar();

        }

    }

```

```

        tabelasm = true;

```

```

JOptionPane.showMessageDialog(null, "Tabelas Migradas", "MigraSGBD",
    JOptionPane.INFORMATION_MESSAGE);

jButton9.setEnabled(false);

if (dadosp) {
    jButton10.setEnabled(true);
}

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, "Erro ao gravar o arquivo",
        "AtenÃ§Ã£o",
        JOptionPane.WARNING_MESSAGE);

}

} //GEN-LAST:event_jButton9MouseClicked

private void jButton10MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton10MouseClicked
    try {
        ArrayList<String> conteudo = new ArrayList<String>();
        conteudo = leArquivo("DadosPostgreSQL");

        for (int z = 0; z < conteudo.size(); z++) {
            stmt = connectionPostgreSQL.prepareStatement(conteudo.get(z));
            stmt.execute();
            connectionPostgreSQL.desconectar();
        }

        JOptionPane.showMessageDialog(null, "Dados Migrados", "MigraSGBD",
            JOptionPane.INFORMATION_MESSAGE);

        jButton10.setEnabled(false);
    }
}

```

```

        JOptionPane.showMessageDialog(null, "Banco Oracle ->
PostgreSQL\nMigrado com Sucesso", "MigraSGBD",

        JOptionPane.INFORMATION_MESSAGE);

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "Erro ao gravar o arquivo",
"AtenÃ§Ã£o",

        JOptionPane.WARNING_MESSAGE);

    }

} //GEN-LAST:event_jButton10MouseClicked

private void jButton11MouseClicked(java.awt.event.MouseEvent evt) { //GEN-
FIRST:event_jButton11MouseClicked

    try {

        stmt = connectionMySQL.getPreparedStatement("create database " +
bancoOracle);

        stmt.execute();

        connectionMySQL.desconectar();

        connectionMySQL.conectar(bancoOracle);

        JOptionPane.showMessageDialog(null, "Base de Dados Criada",
"AtenÃ§Ã£o",

        JOptionPane.INFORMATION_MESSAGE);

        jButton11.setEnabled(false);

        jButton1.setEnabled(true);

        jButton2.setEnabled(true);

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "Erro ao criar a base de dados",
"AtenÃ§Ã£o",

```

```

        JOptionPane.WARNING_MESSAGE);

    }

    }//GEN-LAST:event_jButton11MouseClicked

    private void jButton12MouseClicked(java.awt.event.MouseEvent evt) { //GEN-FIRST:event_jButton12MouseClicked

        try {

            stmt = connectionPostgreSQL.getPreparedStatement("create database " +
bancoOracle);

            stmt.execute();

            connectionPostgreSQL.desconectar();

            connectionPostgreSQL.conectar(bancoOracle);

            JOptionPane.showMessageDialog(null, "Base de Dados Criada",
"AtenÃ§Ã£o",

                JOptionPane.INFORMATION_MESSAGE);

            jButton12.setEnabled(false);

            jButton7.setEnabled(true);

            jButton8.setEnabled(true);

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, "Erro ao criar a base de dados",
"AtenÃ§Ã£o",

                JOptionPane.WARNING_MESSAGE);

        }

    }//GEN-LAST:event_jButton12MouseClicked

    public String testaTipoMySQL(String tipo, String tamanho1, String tamanho2,
String tamanho3) {

```

```

String retorno = "";

if (tipo.equals("VARCHAR2")) {
    retorno = "TEXT" + "(" + tamanho1 + ")";
} else if (tipo.equals("VARCHAR")) {
    retorno = "VARCHAR" + "(" + tamanho1 + ")";
} else if (tipo.equals("INT") || (tipo.equals("NUMBER") &&
tamanho3.equals("0"))) {
    retorno = "INTEGER";
} else if (tipo.equals("NUMBER") && !tamanho3.equals("0")) {
    retorno = "FLOAT" + "(" + tamanho2 + "," + tamanho3 + ")";
} else if (tipo.equals("FLOAT") || tipo.equals("REAL")) {
    retorno = "FLOAT" + "(" + tamanho2 + "," + tamanho3 + ")";
} else if (tipo.equals("DATE")) {
    retorno = "DATE";
} else if (tipo.equals("CHAR")) {
    retorno = "CHAR" + "(" + tamanho1 + ")";
}

return retorno;
}

public String testaTipoPostgreSQL(String tipo, String tamanho1, String tamanho2,
String tamanho3) {

    String retorno = "";

    if (tipo.equals("VARCHAR2")) {

```

```

        retorno = "VARCHAR" + "(" + tamanho1 + ")";
    } else if (tipo.equals("VARCHAR")) {
        retorno = "VARCHAR" + "(" + tamanho1 + ")";
    } else if (tipo.equals("INT") || (tipo.equals("NUMBER") &&
tamanho3.equals("0"))) {
        retorno = "INTEGER";
    } else if (tipo.equals("NUMBER") && !tamanho3.equals("0")) {
        retorno = "REAL";
    } else if (tipo.equals("FLOAT") || tipo.equals("REAL")) {
        retorno = "REAL";
    } else if (tipo.equals("DATE")) {
        retorno = "DATE";
    } else if (tipo.equals("CHAR")) {
        retorno = "CHAR" + "(" + tamanho1 + ")";
    }

    return retorno;
}

```

```

public ArrayList<String> leArquivo(String tipo) {
    ArrayList<String> conteudo = new ArrayList<String>();
    try {
        FileReader reader = new FileReader(System.getProperty("user.dir") + "\\ " +
"arquivo" + tipo + ".txt");
        BufferedReader buffReader = new BufferedReader(reader);

        String linha;

```

```

        while ((linha = buffReader.readLine()) != null) {
            conteudo.add(linha);
        }

        reader.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Erro ao ler o arquivo", "Atenção",
            JOptionPane.WARNING_MESSAGE);
    }

    return conteudo;
}

public void gravaArquivo(String conteudo, String tipo) {
    try {
        FileWriter x = new FileWriter(System.getProperty("user.dir") + "\\ " + "arquivo"
+ tipo + ".txt");

        x.write(conteudo);

        x.close();

        JOptionPane.showMessageDialog(null, "Arquivo 'arquivo' + tipo + '.text'
Criado", "MigraSGBD",
            JOptionPane.INFORMATION_MESSAGE);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Erro ao criar o arquivo",
            "Atenção",
            JOptionPane.WARNING_MESSAGE);
    }
}

```

```

public void CarregaTabela() {
    modelo.getDataVector().clear();
    modelo.setColumnCount(0);
    String lst[] = new String[2];
    try {
        stmt = connectionOracle.getPreparedStatement(
            "select table_name as tabelas from user_tables order by tabelas");
        ResultSet rs = stmt.executeQuery();
        //TITULO
        for (int i = 0; i < titulos.length; i++) {
            modelo.addColumn(titulos[i]);
        }

        int x = 1;
        while (rs.next()) {
            lst[0] = String.valueOf(x);
            lst[1] = rs.getString("tabelas");
            modelo.addRow(lst);
            x++;
        }

        connectionOracle.desconectar();
        stmt.close();
        jTable1.setModel(modelo);

        //Ajusta o tamanho das colunas e centraliza as desejadas
    }
}

```

```

        jTable1.getColumnModel().getColumn(0).setResizable(false);
        jTable1.getColumnModel().getColumn(0).setCellRenderer(rendererCentro);
        jTable1.getColumnModel().getColumn(0).setPreferredWidth(2);
        jTable1.getColumnModel().getColumn(1).setResizable(false);
        jTable1.getColumnModel().getColumn(1).setPreferredWidth(200);
    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, "Erro ao Carregar Tabela",
"MigraSGBD",

        JOptionPane.INFORMATION_MESSAGE);
    }
}

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton10;
private javax.swing.JButton jButton11;
private javax.swing.JButton jButton12;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JButton jButton9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;

```

```
private javax.swing.JLabel jLabel3;  
private javax.swing.JPanel jPanel1;  
private javax.swing.JPanel jPanel3;  
private javax.swing.JPanel jPanel4;  
private javax.swing.JPanel jPanel5;  
private javax.swing.JScrollPane jScrollPane1;  
public javax.swing.JTable jTable1;  
// End of variables declaration//GEN-END:variables  
}
```