

Modelagem de dados com o SQL Developer Data Modeler

Este artigo apresenta a aplicação do modelo entidade-relacionamento no projeto de um banco de dados para um módulo de controle bancário.

Ao desenvolvermos um sistema, seja de grande ou pequeno porte, é necessário planejar todas as suas etapas e dedicar uma atenção especial ao projeto do banco de dados. Sem esse planejamento prévio, a manutenção corretiva do sistema, além de constante, será mais complicada. O projeto de banco de dados utiliza uma técnica chamada modelagem de dados. Ela procura transformar uma ideia conceitual em algo que possa ser traduzido em termos computacionais.

Neste artigo, aplicamos o modelo entidade-relacionamento em um estudo de caso projetando e modelando um banco de dados para um módulo de controle bancário com as funcionalidades básicas de cadastramento de clientes, abertura, movimentação e encerramento de contas. Para a modelagem propriamente dita, utilizamos o SQL Developer Data Modeler, uma ferramenta CASE da Oracle disponibilizada de forma gratuita e que cobre todas as fases do projeto de um banco relacional.

O software atualmente é desenvolvido em cenários bastante complexos e com uma gama muito grande de variáveis envolvidas. Nesse contexto, é fundamental que se utilize um modelo de desenvolvimento no planejamento de um sistema. Os paradigmas de engenharia de software propiciam métodos e técnicas que mostram como proceder na construção de software.

A utilização de um método no desenvolvimento de um software traz vantagens que são facilmente verificáveis durante e após a sua construção: trabalho sistematizado, documentos padronizados, melhor controle no desenvolvimento, uniformidade de linguagem e maximização dos resultados.

Uma das primeiras tentativas de criar processos formais para o desenvolvimento de software foi através de uma conferência internacional, em 1968, promovida pelo grupo de estudo em ciência da computação da OTAN (Organização do Tratado do Atlântico Norte). Essa conferência foi uma tentativa de encontrar caminhos para melhorar o desenvolvimento de software que, durante a década de 60, ocorria sem nenhum método, de maneira informal, gerando atrasos e falhas nos projetos, que culminavam em metas não atingidas, custos imprevisíveis e sistemas de difícil manutenção. Tal situação caótica ficou conhecida como a “crise do software”, e a conferência de 1968 da OTAN procurou difundir a necessidade de um software que fosse concebido dentro de práticas disciplinadas. A conferência terminou sem uma proposta concreta, mas fomentou diversas discussões sobre esse assunto. Com o passar do tempo, entre as propostas que surgiram, a primeira a ter maior difusão e prática foi a técnica de análise estruturada.

Essa técnica começou a ser efetivamente utilizada a partir de 1975. Ela envolve a construção de um sistema de forma *top-down* (do todo para as partes) considerando-se os refinamentos sucessivos, produzindo em um primeiro momento uma fotografia global do sistema. Com base nesse esquema global, realiza-se uma decomposição funcional, gerando-se fluxos de processos especializados que são um detalhamento do fluxo macro. Esses detalhamentos começam a dar pistas dos dados requeridos que, posteriormente, serão representados como entidades no modelo entidade-relacionamento.

Dada a simplicidade da diagramação e dos conceitos envolvidos, o modelo teve ampla aceitação e passou a ser referência na modelagem de dados, sendo utilizado até os dias atuais no projeto de bancos relacionais. Ele é baseado em dois grupos de objetos: entidades e relacionamentos (que serão detalhados mais adiante).

Modelo Conceitual

Nessa etapa, as necessidades de informação são representadas através de uma visão genérica dos dados e seus relacionamentos. Um exemplo desse tipo de representação é apresentado a seguir:

1. Clientes pessoa física

Dados necessários: nome, endereço, bairro, cidade, estado, CEP, telefone, e-mail, contato.

2. Clientes pessoa jurídica

Dados necessários: CNPJ, razão social, endereço, bairro, cidade, estado, CEP, e-mail, contato.

3. Produtos

Dados necessários: identificador, nome, modelo, unidade de medida, preço de compra, preço de venda, data da última compra.

4. Operações

a. Cadastro de cliente (pessoa física ou pessoa jurídica)

Quando o cliente que está realizando uma compra não é cadastrado, o funcionário deve realizar o seu registro no sistema.

b. Cadastro de produto

Ao ser adquirido, o produto deve ser cadastrado pelo funcionário. Para isso, ele insere informações como nome, preço de compra e de venda e a data da compra.

Modelo Lógico

Nessa etapa, iremos representar as estruturas utilizadas para o armazenamento no banco de dados utilizando o modelo lógico. Para essa representação gráfica, utiliza-se o diagrama entidade-relacionamento (DER), que permite representar as estruturas de dados referentes a uma parcela do mundo real ("minimundo"). Esse diagrama mapeia os dados que as operações do sistema utilizam e caracteriza-se pela independência de dispositivos ou meios de armazenamento físico.

A modelagem com o DER permite a utilização de diversos elementos de modelagem, alguns mais simplificados e outros mais avançados. Apresentamos, neste artigo, apenas os construtores necessários para a modelagem de nosso estudo de caso. Esses construtores são apresentados na notação Chen (originalmente publicada no trabalho de Peter Chen, em 1976) e, quando necessário, na notação Barker (utilizada pelo SQL Developer Data Modeler):

- **Entidade:** objeto de dado básico do modelo entidade-relacionamento, cujas informações devem ser coletadas. Representa uma pessoa, lugar, coisa ou evento do mundo real, de interesse informativo. Uma ocorrência específica de uma entidade é chamada de *instância da entidade*. Na **Figura 1**, temos uma representação gráfica do construtor para entidades: um retângulo com as bordas retas (Notação Chen) ou arredondadas (Notação Barker), sendo que o nome da entidade é escrito dentro do retângulo.

- **Atributo:** são características das entidades, que oferecem detalhes descritivos sobre elas. Uma ocorrência em particular de um atributo dentro de uma entidade é chamada de *valor de atributo*. Na **Figura 1**, eles são representados de duas formas: na notação Barker, eles aparecem descritos dentro do construtor da entidade, logo abaixo do nome dela; na notação Chen, seus construtores são retângulos com bordas arredondadas, com o nome do atributo em seu interior e uma linha que

o conecta à sua entidade correspondente. Um atributo pode ser caracterizado como descritor ou identificador (chave).

· **Chave:** um identificador (ou chave) é usado para determinar exclusivamente uma instância de uma entidade. Essa chave pode ser composta de um único atributo (chave simples) ou por um conjunto de atributos (chave composta). Uma entidade pode ter mais de uma chave e, nesse caso, esse conjunto é denominado de chaves candidatas. Dentro da relação de chaves candidatas, elege-se uma que será utilizada como indexador exclusivo de cada instância da entidade. Essa chave recebe o nome de chave primária. Na **Figura 1**, as chaves são representadas como um atributo sublinhado (notação Chen) ou como um atributo com o símbolo “#” à sua esquerda (notação Barker).

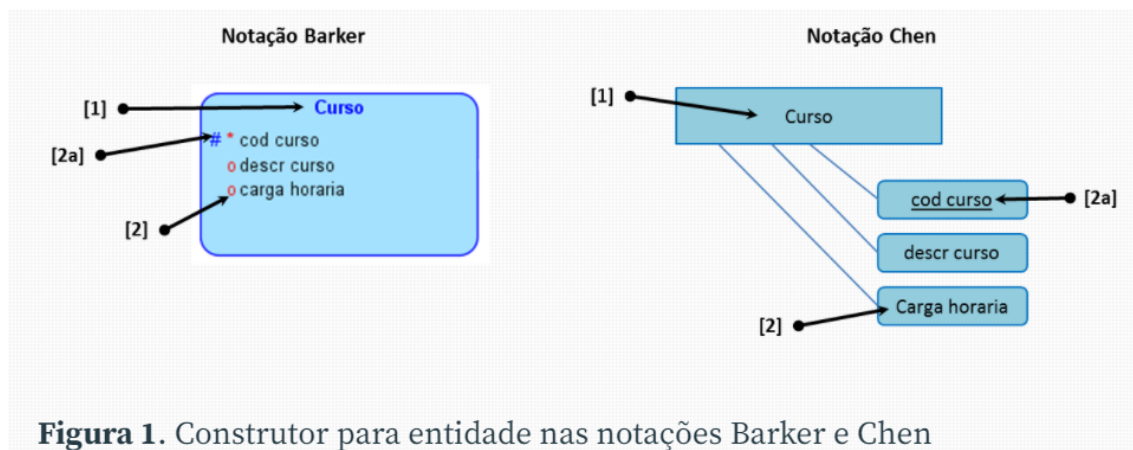


Figura 1. Construtor para entidade nas notações Barker e Chen

Entidade fraca: um tipo de entidade que não possui um identificador próprio, mas deriva sua identidade a partir dos atributos identificadores de uma ou mais “entidades-pai”, de quem é dependente para sua própria existência. Na **Figura 2**, temos a representação gráfica de uma entidade fraca na notação Chen. Ela é representada como um retângulo de borda dupla. Percebam, no exemplo apresentado, que a chave da entidade “sala” (“num_sala” + “id_edificio”) é derivada da chave da entidade “edifício” (“id_edificio”), tendo em vista que uma sala não tem uma identidade separada de seu edifício, pois está contida fisicamente nele.

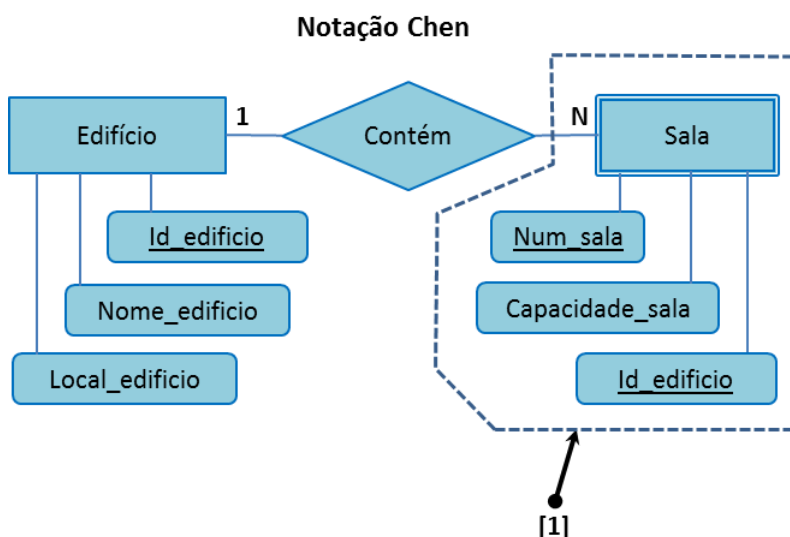


Figura 2. Construtor para entidade fraca na notação Chen

· **Relacionamentos:** representam associações do mundo real entre uma ou mais entidades e, dessa forma, não possuem existência física. Eles apenas conceituam as dependências que existem entre as entidades associadas. Na **Figura 3**, observamos que na notação Barker ele é uma linha que conecta as entidades associadas, com o nome do relacionamento aparecendo logo acima

dessa linha. Na notação Chen, seu construtor é um losango conectando as entidades associadas, com o nome do relacionamento em seu interior.

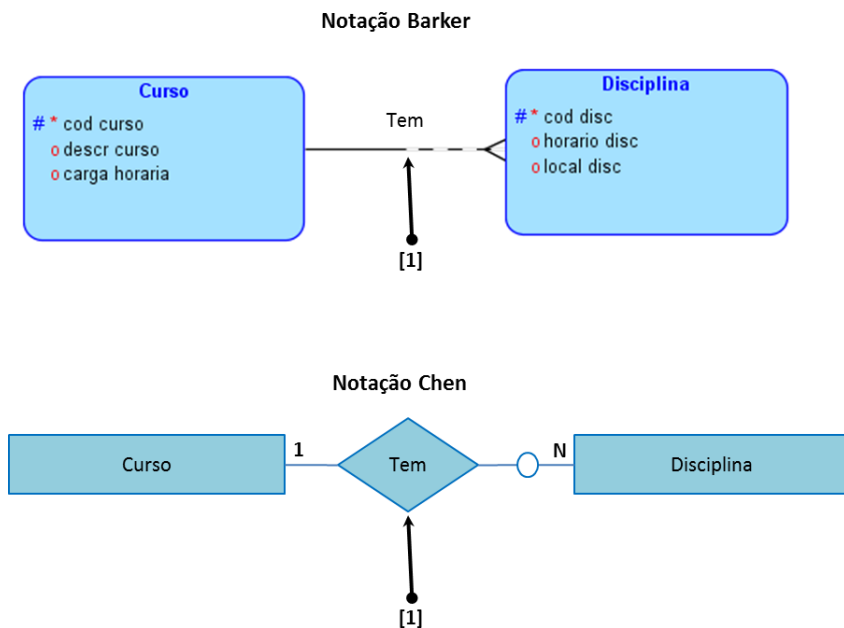


Figura 3. Construtor para relacionamentos nas notações Barker e Chen

Os relacionamentos podem ser descritos em função de sua conectividade e participação:

- **Conectividade:** descreve uma restrição sobre a conexão das ocorrências de entidade associadas no relacionamento. Os valores para conectividade são “um” ou “muitos”. A contagem real de elementos associados à conectividade do relacionamento é chamada de cardinalidade, mas esse conceito é usado com menos frequência do que a conectividade, pois os valores reais variam entre as instâncias do relacionamento.

A **Figura 4** mostra os construtores básicos de conectividade em relacionamentos. Eles podem ser “um-para-muitos”, “um-para-um” e “muitos-para-muitos”. Na notação Barker, o lado “um” é representado por uma extremidade de linha simples conectando-se a uma entidade, enquanto o lado “muitos” aparece como uma extremidade com ramificações conectada à entidade, sendo que essa representação ramificada é conhecida como “pé-de-galinha”. Na notação Chen, o lado “um” possui o número um aparecendo na conexão entre a entidade e o relacionamento, enquanto o lado “muitos” possui a letra “N” na conexão entre o relacionamento e a entidade.

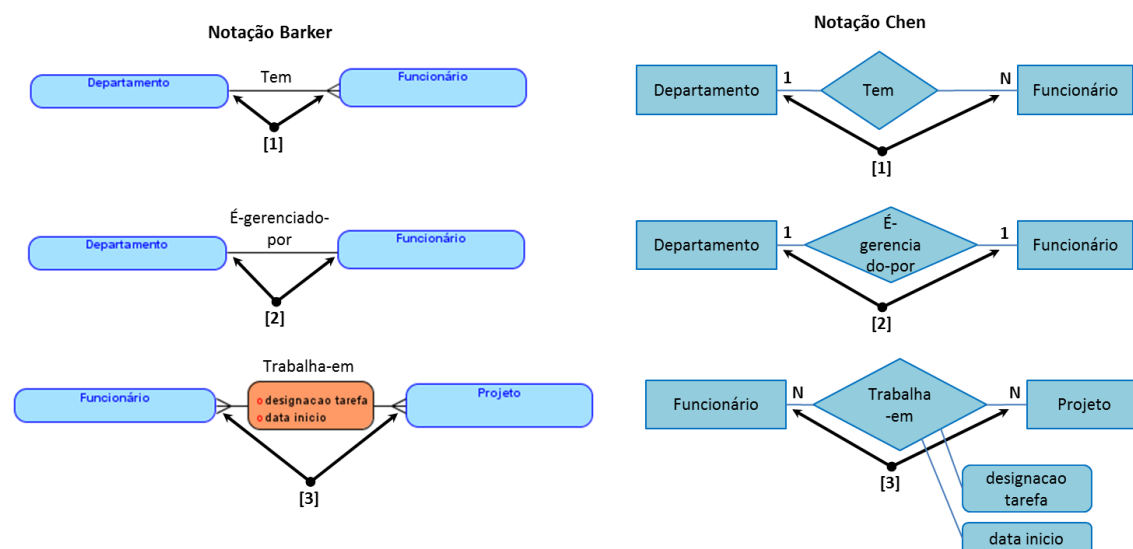


Figura 4. Construtores para conectividade de relacionamentos nas notações Barker e Chen

Analisando os exemplos da **Figura 4**, e com as definições apresentadas até agora para a representação de conectividade, podemos dizer que no exemplo da primeira linha da figura, um “Departamento” possui um ou muitos “Funcionários”, enquanto que um “Funcionário” deve pertencer a pelo menos um e somente um “Departamento”. Na segunda linha da figura, um “Departamento” é gerenciado por pelo menos um e somente um “Funcionário”, enquanto que um “Funcionário” pode gerenciar apenas um “Departamento”. No exemplo da terceira linha da figura, um “Funcionário” pode trabalhar em um ou vários “Projetos”, enquanto que um “Projeto” pode ter um ou muitos “Funcionários” trabalhando nele.

· **Participação:** a participação de uma instância de entidade em um relacionamento é definida como obrigatória ou opcional. Se uma instância de entidade sempre tiver de participar do relacionamento para que ela exista, então sua participação é obrigatória. Se, por outro lado, uma instância dessa entidade nem sempre precisar participar do relacionamento, ela é considerada opcional.

A **Figura 5** exibe exemplos de representação para a participação de uma entidade em relacionamentos. Em [1], temos a representação de uma participação opcional, enquanto que em [2], temos uma participação obrigatória. Na notação Barker, o lado opcional é representado por uma extremidade de linha pontilhada conectando-se a uma entidade, enquanto o lado obrigatório possui uma extremidade com linha contínua. Na notação Chen, o lado opcional possui um círculo vazado na linha de conexão entre a entidade e o relacionamento, enquanto o lado obrigatório possui uma linha contínua na conexão entre o relacionamento e a entidade.

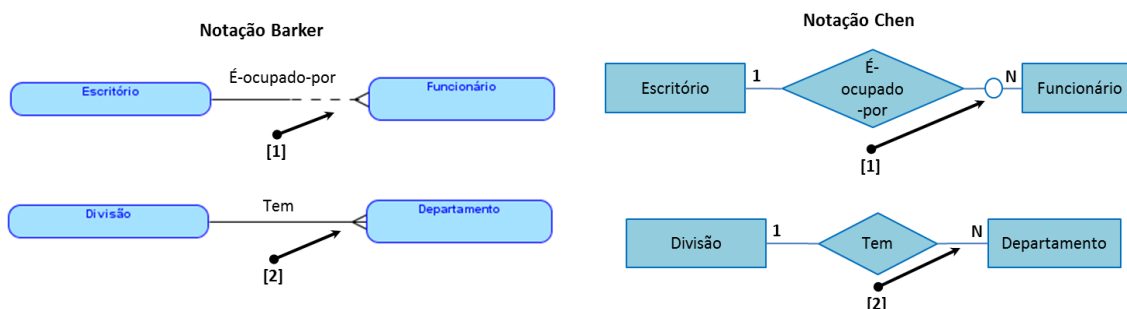
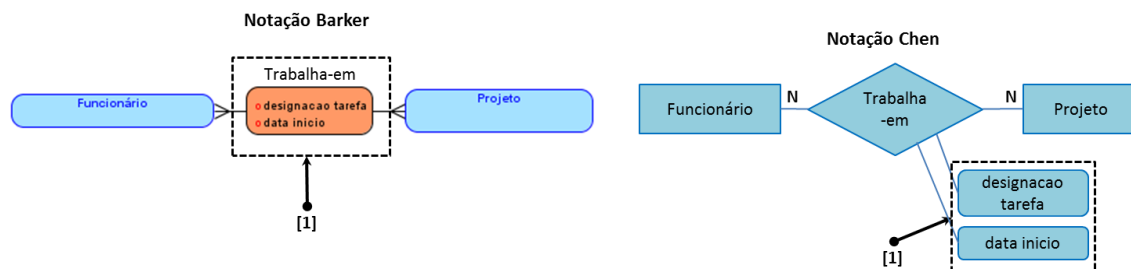


Figura 5. Construtores para participação de relacionamento nas notações Barker e Chen

Com base no que foi exposto, analisemos os exemplos da **Figura 5**. Na primeira linha da figura, um “Escritório” pode ou não ser ocupado por um “Funcionário”, tornando sua ocupação opcional. Na segunda linha da figura, uma “Divisão” tem que possuir pelo menos um “Departamento”, tornando essa relação obrigatória.

· **Atributos de relacionamento:** atributos podem ser associados a certos tipos de relacionamentos, além das entidades. Normalmente, esses atributos associam-se a relacionamentos “muitos-para-muitos”, a fim de identificar uma característica que é comum apenas a uma determinada combinação de instâncias de cada entidade associada. Exemplificamos essa representação na **Figura 6**. Nesse exemplo, os atributos “designação tarefa” e “data início” do relacionamento “Trabalha-em” ([1]) indicam que um determinado “Funcionário” tem uma tarefa designada e uma data de início específica em um determinado “Projeto”. Na notação Barker, esses atributos são encapsulados em um único retângulo com bordas arredondadas em cima da linha de relacionamento das entidades associadas. Na notação Chen, cada atributo é colocado em um retângulo com bordas arredondadas, cujas linhas conectam-se ao losango do relacionamento.



[abrir imagem em nova janela](#)

Figura 6. Construtores para atributos de relacionamento nas notações Barker e Chen

• **Generalização:** o relacionamento de generalização especifica que vários tipos de entidades com atributos comuns podem ser generalizados para um tipo de entidade de nível superior, que é chamada de entidade genérica ou supertipo. As entidades conectadas no nível inferior são conhecidas como entidades subtipo e representam subconjuntos da entidade supertipo. Uma estrutura que é formada pela combinação de relacionamentos supertipo/subtipo é chamada de hierarquia supertipo/subtipo, ou hierarquia de generalização.

A generalização também pode ser descrita em termos de herança, onde os atributos de um supertipo são propagados para baixo na hierarquia até as entidades de um tipo inferior. Quanto ao tipo de restrição, as entidades subtipo em uma hierarquia de generalização podem ser disjuntas ou sobrepostas:

o **Subtipos disjuntos:** esse tipo de restrição exige que as entidades sejam mutuamente exclusivas. Um exemplo de disjunção é mostrado na **Figura 7**. Nesse exemplo, “Pessoa Física” e “Pessoa Jurídica” são subtipos disjuntos ([2]) da entidade supertipo “Cliente” ([1]), pois uma instância de “Cliente” não pode ser simultaneamente uma Pessoa Física e Jurídica. A representação gráfica do relacionamento de disjunção, na notação Barker, é feita colocando-se um arco sobre as linhas de relacionamento vermelhas que ligam os subtipos disjuntos à entidade supertipo ([3]).

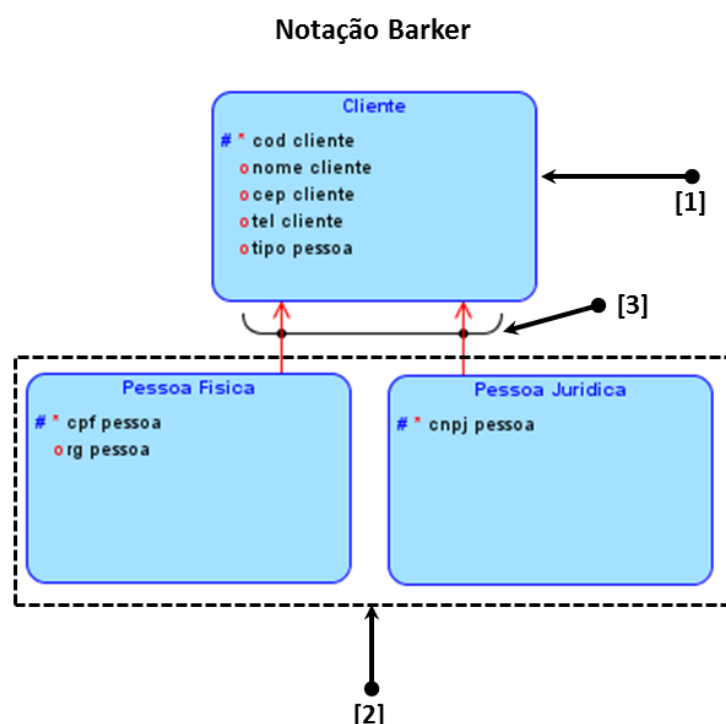


Figura 7. Construtor para generalização com subtipos disjuntos na notação Barker

o **Subtipos sobrepostos**: duas entidades subtipo são consideradas sobrepostas quando podem ocorrer simultaneamente em uma mesma instância da entidade supertipo relacionada. Um exemplo de sobreposição é mostrado na **Figura 8**. Nesse exemplo, “Horista” e “Mensalista” são subtipos sobrepostos ([2]) da entidade supertipo “Funcionário” ([1]), pois uma instância de “Funcionário” pode ser tanto trabalhador horista quanto mensalista. A representação gráfica do relacionamento de sobreposição, na notação Barker, é feita colocando-se linhas de relacionamento vermelhas ligando os subtipos sobrepostos à entidade supertipo ([3]).

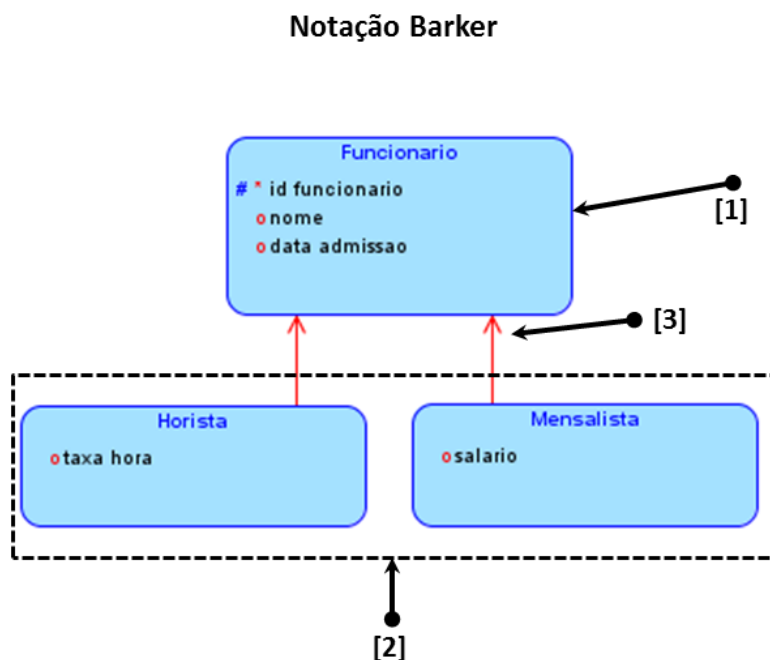


Figura 8. Construtor para generalização com subtipos sobrepostos na notação Barker

Modelo Físico

O modelo físico é derivado a partir do modelo lógico e é formado por detalhes relacionados à estrutura física do banco, como tabelas, tipos, índices etc. Para representá-lo, emprega-se o diagrama de estrutura de dados (DED). Ele possui três notações gráficas: retângulos para representar tabelas, traços indicando relacionamentos e uma representação de conectividade que, no caso da notação utilizada pelo SQL Developer Data Modeler, é a “pé-de-galinha”.

Obtém-se o DED a partir da migração do DER, considerando-se os seguintes aspectos:

1. Cada entidade no DER origina uma tabela no DED (vide exemplo da **Figura 9**);

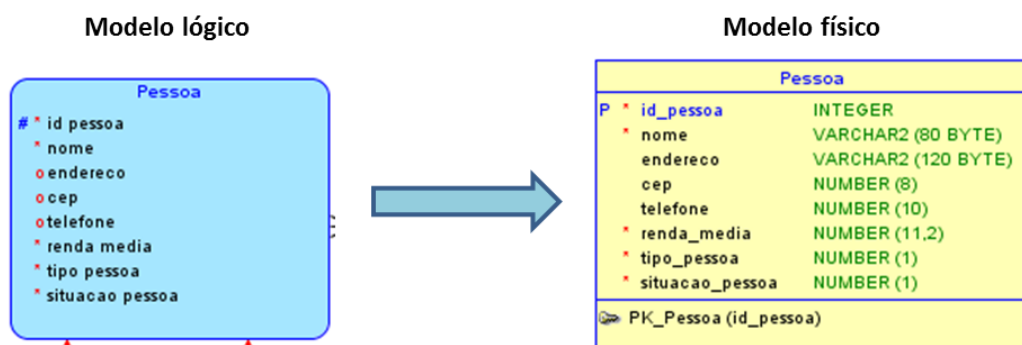


Figura 9. Convertendo entidades (modelo lógico) em tabelas (modelo físico)

2. Todo relacionamento no DER que tenha algum atributo torna-se uma tabela no DED;

3. Toda relação entre uma entidade “A” e outra “B” no DER, em que se verifica uma conectividade de “muitos-para-muitos”, origina uma entidade intermediária “C”, também chamada de entidade associativa. Essa entidade “C” deve estar relacionada com a “A” e a “B”, apresentando conectividade “um-para-muitos” em ambos os casos. Tal entidade “C” também se transforma em uma tabela no DED (vide exemplo da **Figura 10**).

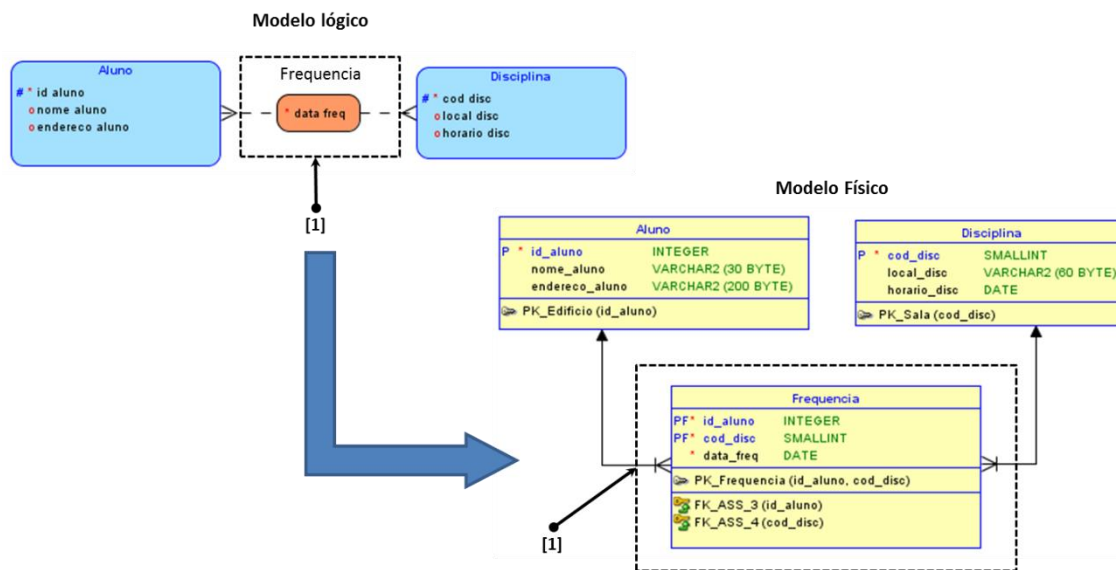


Figura 10. Convertendo relacionamento "muitos-para-muitos" em entidade associativa ([1])

4. Hierarquias de generalização com subtipos sobrepostos no DER são convertidos conforme o exposto na **Figura 11**. A entidade supertipo origina uma tabela no DED ([1]). As entidades subtipo herdam os atributos de chave primária da entidade supertipo e dão origem a tabelas no DED com esses atributos repetidos ([2]). Os relacionamentos de generalização das entidades subtipo com o supertipo tornam-se, cada um deles, relacionamentos “um-para-um” no DED, sendo que esses relacionamentos têm um arco sobreposto a eles ([3]).

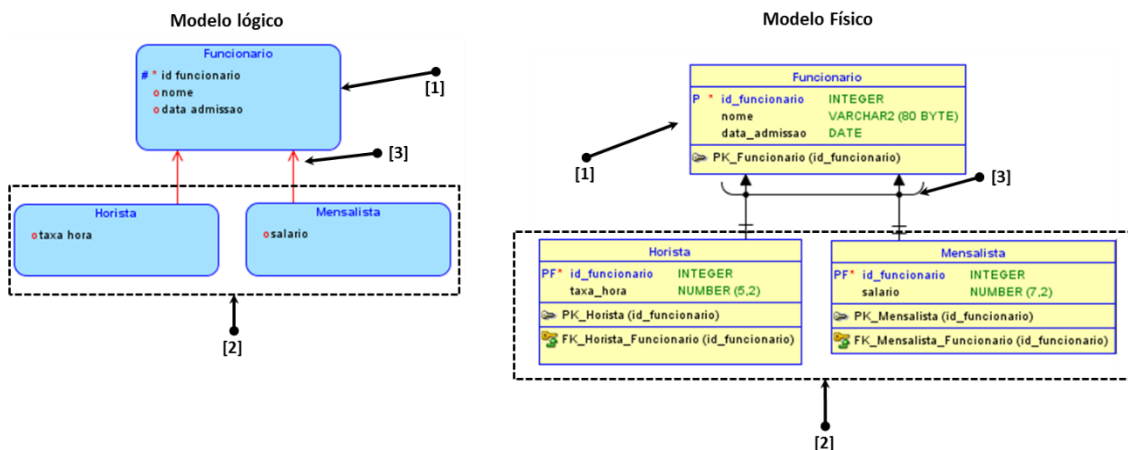


Figura 11. Convertendo hierarquias de generalização com subtipos sobrepostos

5. Hierarquias de generalização com subtipos disjuntos no DER são convertidos conforme o exposto na **Figura 12**. A entidade supertipo origina uma tabela no DED ([1]). As entidades subtipo herdam os atributos de chave primária da entidade supertipo e dão origem a tabelas no DED com esses atributos repetidos ([2]). Os relacionamentos de generalização das entidades subtipo com o supertipo tornam-se, cada um deles, relacionamentos “um-para-um” no DED, sendo que esses relacionamentos têm dois arcos sobrepostos a eles ([3]).

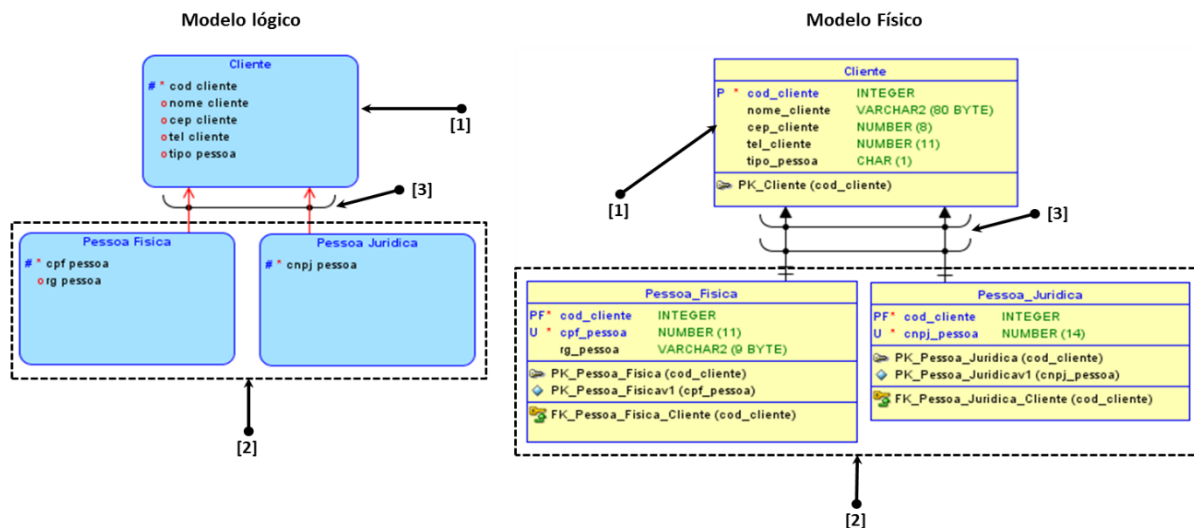


Figura 12. Convertendo hierarquias de generalização com subtipos disjuntos

SQL Developer Data Modeler

O SQL Developer Data Modeler (SDDM) é uma ferramenta moderna de modelagem de dados fornecida gratuitamente pela Oracle. Usando o SDDM, os projetistas de bancos de dados podem criar, navegar e editar modelos lógicos, relacionais, físicos, multidimensionais e de tipos de dados. Ele possui capacidades de conversão automática de modelo lógico para modelo relacional (forward engineering) e de engenharia reversa, além de possibilitar o desenvolvimento colaborativo através de controle de código fonte integrado. O SDDM pode ser utilizado em ambientes tradicionais ou de cloud computing.

Para o estudo de caso deste artigo, modelamos um banco de dados para um módulo de controle bancário. Para essa tarefa, utilizamos o SDDM para gerar o modelo lógico – criando o DER – e o convertimos para o modelo relacional (físico) – obtendo o DED. Ao final, geramos um script DDL (Data Definition Language) contendo os comandos SQL necessários para criar fisicamente os objetos do banco de dados modelado.

Para utilizar o SDDM, basta ir até a página do produto, realizar o download do pacote ZIP, descompactá-lo em seu computador e executá-lo (ver seção **Links**). Nesse artigo, fizemos uso da versão 4.1.1.

Ao iniciar o SDDM e após a carga de todos os seus módulos, seu ambiente de trabalho é exibido, conforme apresentado na **Figura 13**.

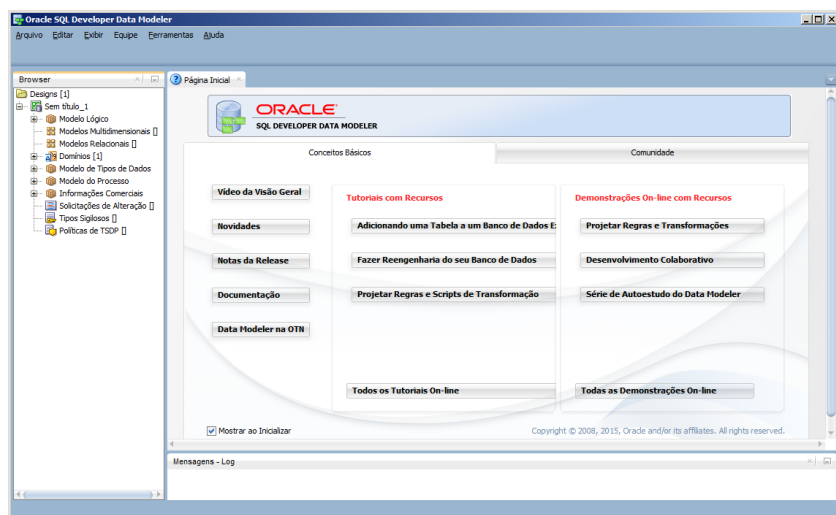


Figura 13. Ambiente de trabalho do SQL Developer Data Modeler

A parte central desse ambiente, onde aparece a Página Inicial, é a área de criação dos diagramas. Na parte superior aparece o menu suspenso e na lateral esquerda temos o Object Browser, cujo detalhamento aparece na **Figura 14**.

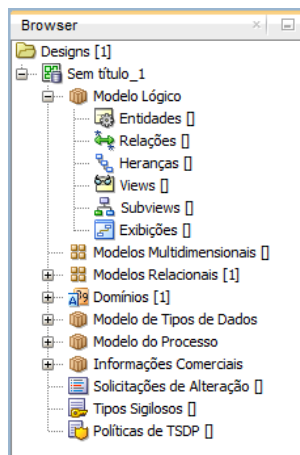


Figura 14. Object Browser

O Object Browser apresenta uma árvore hierárquica com todos os objetos de modelagem de dados disponíveis. Ele permite a seleção rápida desses objetos, expandindo-se o nó de árvore apropriado para visualizá-los. Também é possível executar operações de menu de contexto (botão direito do *mouse*) em seus nós, obtendo-se acesso a diversas configurações do ambiente. Essa característica será útil para nós no tópico seguinte.

Configurando o ambiente do SDDM

Antes de adentrarmos no estudo de caso do módulo de controle bancário e o processo de modelagem propriamente dito, vamos realizar algumas configurações iniciais no ambiente do SDDM para facilitar nosso trabalho.

Inicialmente, criamos um novo projeto (design) associado ao banco de dados, a fim de que todas as configurações de ambiente estejam refletidas nesse projeto criado. Para tanto, abra um menu de contexto em cima do nó Design do Object Browser, conforme demonstrado na **Figura 15**, e selecione a opção Salvar Design.

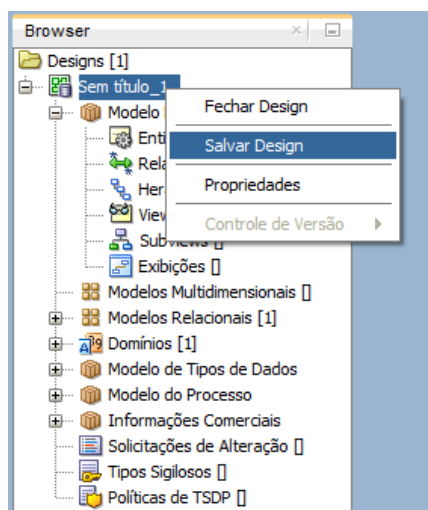


Figura 15. Criando um projeto (design) inicial

Selecione um diretório de sua máquina e crie um projeto chamado “controle_bancario”. Esse procedimento cria um arquivo “controle_bancario.dmd” para abertura do projeto, bem como uma pasta nomeada como “controle_bancario”, que armazena todos os artefatos relacionados com os objetos de modelagem de dados do projeto.

Confirmada a geração do novo projeto, o Object Browser é atualizado (vide **Figura 16**): o nó *Design* agora tem o nome do projeto recém-criado (“controle_bancario”).

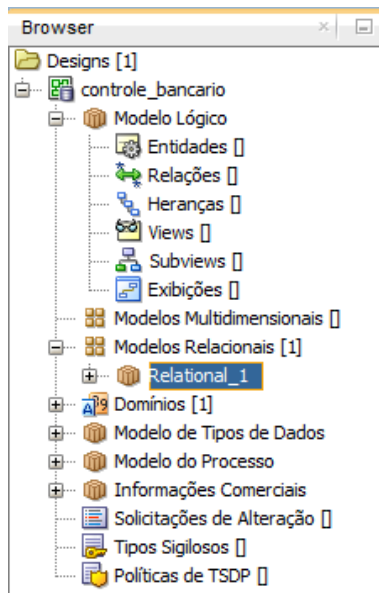


Figura 16. Object Browser atualizado

Vamos agora realizar ajustes no modelo relacional (físico) desse projeto para que ele fique aderente às definições do Sistema Gerenciador de Banco de Dados que iremos utilizar para criar os objetos fisicamente. Para tanto, expanda o nó Modelos Relacionais no Object Browser (vide **Figura 16**) e execute um duplo clique sobre o objeto “Relational_1”. Um formulário de propriedades desse modelo relacional é aberto.

Nesse formulário, preencha o campo “Nome” com o valor “controle_bancario_r” para fazermos referência ao modelo relacional do módulo de controle bancário. Na lista “Tipo RDBMS:”, aparecem os nomes de diversos SGBDs e suas versões específicas. Para o projeto do nosso banco de dados, a título de exemplo, selecionamos a opção Oracle Database 11g.

Após confirmar as alterações e fechar o formulário anterior, retornemos ao Object Browser e selecionemos o nó Design – que agora está com o nome do projeto “controle_bancário”. Após selecionar esse nó, chamamos um menu de contexto e selecionamos a opção Propriedades.

Um formulário de propriedades do Design é aberto. Realizamos adequações nesse formulário para que se reflitam no padrão de nomenclatura dos modelos gerados para o projeto “controle_bancario”.

No menu à esquerda desse formulário, expanda o nó Definições. Em seguida, expanda o nó Padrão de Nomenclatura e, por último, selecione o nó Modelos. No formulário exibido à direita do menu, realize ajustes nos seguintes campos:

- “Chave Primária”: preencher com o valor PK_{table}.
- “Chave Estrangeira”: preencher com o valor FK_{child}_{parent}.
- “Restrição da Verificação da Coluna”: preencher com o valor CK_{table}_{column}.

- “Chave Estrangeira da Coluna”: preencher com o valor {ref column}.
- “Identificador Principal”: preencher com o valor PK {entity}.

Esses ajustes são opcionais, mas são importantes por permitirem definir um padrão para a nomenclatura dos objetos gerados pela ferramenta.

Para verificar o modo de exibição dos padrões de nomenclatura, no frame “Exemplo” desse formulário existe uma lista combo com os tipos de restrição presentes nos campos. Selecione um tipo qualquer dessa lista e veja no campo “Exemplo” como o padrão de nomenclatura é utilizado (na **Figura 17**, o exemplo visualizado foi para o padrão de nomenclatura de chave estrangeira). Confirme as alterações e feche o formulário. Agora já estamos com o ambiente preparado e configurado para o processo de modelagem de nosso estudo de caso.

Propriedades do Design - controle_bancario

Modelos

Restrição de Tabela	Nome	Adicionar Variável
Chave Primária	PK_{table}	Adicionar Variável
Chave Estrangeira	FK_{child}_{parent}	Adicionar Variável
Restrição de Verificação	{table}_{CK}	Adicionar Variável
Restrição Única	{table}_{column}_{UN}	Adicionar Variável
Índice	{table}_{column}_{IDX}	Adicionar Variável
Índice Automático	{table}_{column}_{IDX}	Adicionar Variável
Restrição da Verificação da Coluna	CK_{table}_{column}	Adicionar Variável
Restrição Não Nula	NNC_{table abbr}_{column}	Adicionar Variável
Chave Estrangeira da Coluna	{ref column}	Adicionar Variável
Chave Substituta	{table abbr}_{PK}	Adicionar Variável
Substituir Coluna Chave	{table abbr}_{ID}	Adicionar Variável
Coluna do Discriminador	{table abbr}_{TYPE}	Adicionar Variável

Identificador da Entidade

Identificador Principal	PK {entity}	Adicionar Variável
Relação de Atributo	{ref entity}_{ref attribute}	Adicionar Variável

Exemplo

Exemplo: FK_childTable_parentTable

Chave Estrangeira

OK Aplicar Cancelar Ajuda

Figura 17. Formulário de Propriedades do Design

Estudo de caso: Módulo de Controle Bancário

Esse sistema permite a realização de operações básicas de movimentação bancária de forma que seus clientes abram e encerrem contas, depositem ou saquem valores e emitam saldos ou extratos. Esse módulo também permite manutenções no cadastro de clientes, registrando-os ou alterando seus dados. As funcionalidades descritas podem ser categorizadas pelo tipo de operador que as executa da seguinte forma:

- As funcionalidades de abertura e encerramento de contas, bem como a de manutenção cadastral, só podem ser realizadas por um funcionário do banco;
- As funcionalidades de depósito e saque de valores, além da emissão de saldos ou extratos, podem ser realizadas diretamente pelo cliente, através de um terminal de autoatendimento.

No processo de abertura de conta, o cliente tem a opção de abrir um dos seguintes tipos de conta:

- Conta Comum: conta corrente padrão, que não permite a retirada de mais dinheiro do que está depositado;

- **Conta Especial:** conta que permite o saque extra até um determinado limite;
- **Conta Poupança:** conta que rende juros enquanto o dinheiro depositado permanecer sem ser movimentado.

Para gerarmos a abstração desse sistema, empregamos os três componentes de projeto previstos no modelo entidade-relacionamento: modelo conceitual, modelo lógico e modelo físico. Esses modelos são detalhados nas seções seguintes.

Definindo o modelo conceitual

A seguir, tem-se a lista de entidades e operações que devem constar na definição do módulo de controle bancário:

1. **Pessoa:** armazena informações gerais dos clientes da instituição bancária. Um cliente (pessoa) pode possuir mais de uma conta bancária ativa em seu nome e deve ter em atividade pelo menos uma conta bancária para continuar ativo no sistema.

Dados necessários: nome completo, tipo de pessoa (física ou jurídica), endereço, CEP, telefone, renda média, situação da pessoa (cadastro ativo ou inativo).

2. **Pessoa Física:** especialização (*subtipo*) derivada da entidade Pessoa (*supertipo*), que representa as pessoas físicas que possuem ou possuíram contas ativas na instituição bancária. Ela herda todos os atributos da entidade Pessoa, além de possuir atributos próprios.

Dados necessários: CPF, celular, e-mail, estado civil.

3. **Pessoa Jurídica:** especialização (*subtipo*) derivada da entidade Pessoa (*supertipo*), que representa as pessoas jurídicas que possuem ou possuíram contas ativas na instituição. Ela herda todos os atributos da entidade Pessoa, além de possuir atributos próprios.

Dados necessários: CNPJ, Inscrição Estadual.

4. **Conta Comum:** armazena as contas correntes comuns gerenciadas pelo banco. Nesse tipo de conta, não é possível realizar saques além do valor real depositado. Uma conta corrente pode pertencer a mais de uma pessoa, o que caracteriza uma “conta conjunta”.

Dados necessários: número da conta, código da agência, tipo de conta (além da comum, se é conta especial ou poupança), data de abertura, data de encerramento, situação da conta (ativa ou inativa), senha, saldo.

5. **Conta Especial:** especialização (*subtipo*) derivada da entidade Conta Comum (*supertipo*), que representa as contas que permitem ao correntista sacar valores superiores a seu saldo, até um limite estabelecido. Ela herda todos os atributos da entidade Conta Comum, além de possuir um atributo próprio.

Dados necessários: limite.

6. **Conta Poupança:** especialização (*subtipo*) derivada da entidade Conta Comum (*supertipo*), que representa as contas poupança mantidas pela instituição bancária. Ela herda todos os atributos da entidade Conta Comum, além de possuir um atributo próprio.

Dados necessários: data de aniversário.

7. Movimento: responsável por armazenar as transações ocorridas nas contas (saques e depósitos).

Dados necessários: tipo de movimento (saque ou depósito), data do movimento, hora do movimento, valor movimentado.

8. Operações

a. Abrir Conta Comum: o funcionário do banco consulta o CPF ou o CNPJ do cliente que pretende abrir uma conta e inicia o processo de abertura no sistema. Se o cliente ainda não possui cadastro no banco (primeira conta), realiza-se o cadastramento do cliente. O cliente deve cadastrar uma senha para a conta no sistema. Esse ato confirma a abertura da conta. Em seguida, o cliente deve informar um valor inicial de depósito, que deve ser registrado no sistema.

b. Encerrar Conta: o funcionário solicita ao cliente o número da conta que ele deseja encerrar. Emite-se o saldo da conta em questão para as providências prévias ao encerramento. Se o saldo for positivo, o cliente deve sacar o valor correspondente, caso contrário, ele deve realizar um depósito para quitar o saldo devedor. Efetuada uma das providências anteriores, a conta é encerrada no sistema.

c. Abrir Conta Especial: executam-se os mesmos passos da operação *Abrir conta Comum*, exceto por definir um limite para saque adicional além do valor em saldo do cliente, desde que ele atenda aos requisitos de renda mínima superior a um valor pré-estabelecido pelo banco.

d. Abrir Conta Poupança: executam-se os mesmos passos da operação *Abrir conta Comum*, exceto por definir a data de aniversário da conta, que não é necessariamente a data de abertura, uma vez que o depósito não é obrigatório no momento da abertura de uma conta poupança.

e. Manter Cliente: o funcionário do banco consulta o CPF ou o CNPJ do cliente no sistema. Se o cliente já possui cadastro no sistema, ele exibe seus dados e sua edição é habilitada possibilitando inserir ou alterar dados do cliente. Se as atualizações forem confirmadas, os dados são gravados no sistema.

f. Emitir Saldo: o cliente informa o número da conta que ele deseja emitir saldo no terminal de autoatendimento. O sistema solicita a senha de acesso à conta, que é informada pelo cliente. Após validação da senha, o sistema emite o saldo da conta.

g. Emitir Extrato: o cliente informa o número da conta que ele deseja emitir extrato no terminal de autoatendimento. O sistema solicita a senha de acesso à conta, que é informada pelo cliente. Após validação da senha, o sistema solicita os períodos de movimento a consultar. O cliente informa o período inicial e final do extrato e o sistema, em seguida, lista os movimentos da conta no período informado.

h. Realizar Depósito: o cliente informa o número da conta no terminal de autoatendimento. Após o sistema confirmar a existência da conta, o cliente fornece o valor a ser depositado e realiza a operação. Ao confirmar a transação, o sistema executa a operação *Registrar Movimento* e encerra o processo em seguida.

i. Realizar Saque: o cliente informa o número da conta no terminal de autoatendimento. Após o sistema confirmar a existência da conta, ele solicita a senha de acesso, que é informada pelo cliente. Após validação da senha, o sistema solicita o valor a ser sacado. Se o valor for válido, dependendo do tipo de conta (comum, especial ou poupança), o sistema entrega a importância ao cliente. Ao confirmar a transação, o sistema executa a operação *Registrar Movimento* e encerra o processo em seguida.

j. *Registrar Movimento*: essa operação é acionada de forma indireta pelas operações *Realizar Depósito* e *Realizar Saque*. Ela recebe o número da conta movimentada, o tipo de movimento realizado (saque ou depósito), a data e hora da movimentação e o valor movimentado. De posse desses dados, registra-se o movimento no sistema.

Modelo Lógico: construindo o DER

De posse das definições de dados e operações realizadas pelo sistema, temos condições de identificar as entidades participantes e seus relacionamentos. Elas são modeladas através do DER para construção do modelo lógico do módulo de controle bancário.

Para atingirmos esse objetivo, vamos até o Object Browser na interface do SDDM e selecionamos o nó Modelo Lógico. Escolhido esse nó, acionamos o menu de contexto dele e selecionamos a opção “Mostrar”, conforme indicado na **Figura 18**.

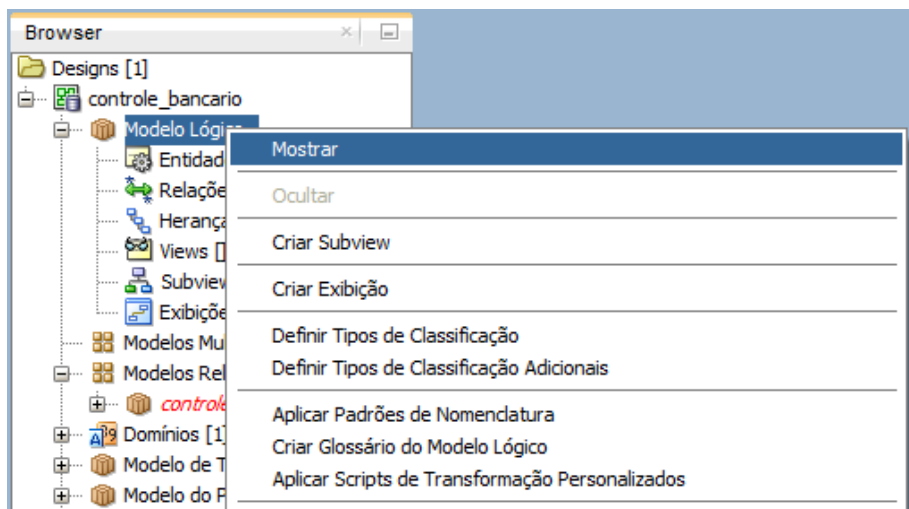


Figura 18. Abrindo o Modelo Lógico no SDDM

Feito isso, um painel se abre na parte central do ambiente de trabalho do SDDM, que possui uma aba na parte superior da área do diagrama com o título “Logical (controle_bancario)”. Acima desse painel e logo abaixo do menu suspenso, visualizamos uma barra de ferramentas, que possui os comandos necessários para criação dos objetos do diagrama escolhido.

Vamos iniciar a criação do DER começando pelas entidades identificadas no modelo conceitual. Para tanto, selecionamos o botão “Nova Entidade” da barra de ferramentas. Pressionado esse botão, o cursor muda sua forma para a de uma cruz. Posicione essa cruz em qualquer região do painel de criação do DER e clique com o botão esquerdo. Uma representação gráfica de uma nova entidade é posicionada na área do diagrama onde estava o cursor e, simultaneamente, é aberto o formulário de Propriedades da Entidade.

No menu à esquerda desse formulário já aparece com o nó “Geral” selecionado. No formulário exibido por esse nó, escreva o valor “Pessoa” no campo “Nome”.

A seguir, selecionamos no menu da esquerda o nó Atributos e obtemos o formulário de cadastramento dos atributos da entidade, conforme indicado na **Figura 19**.

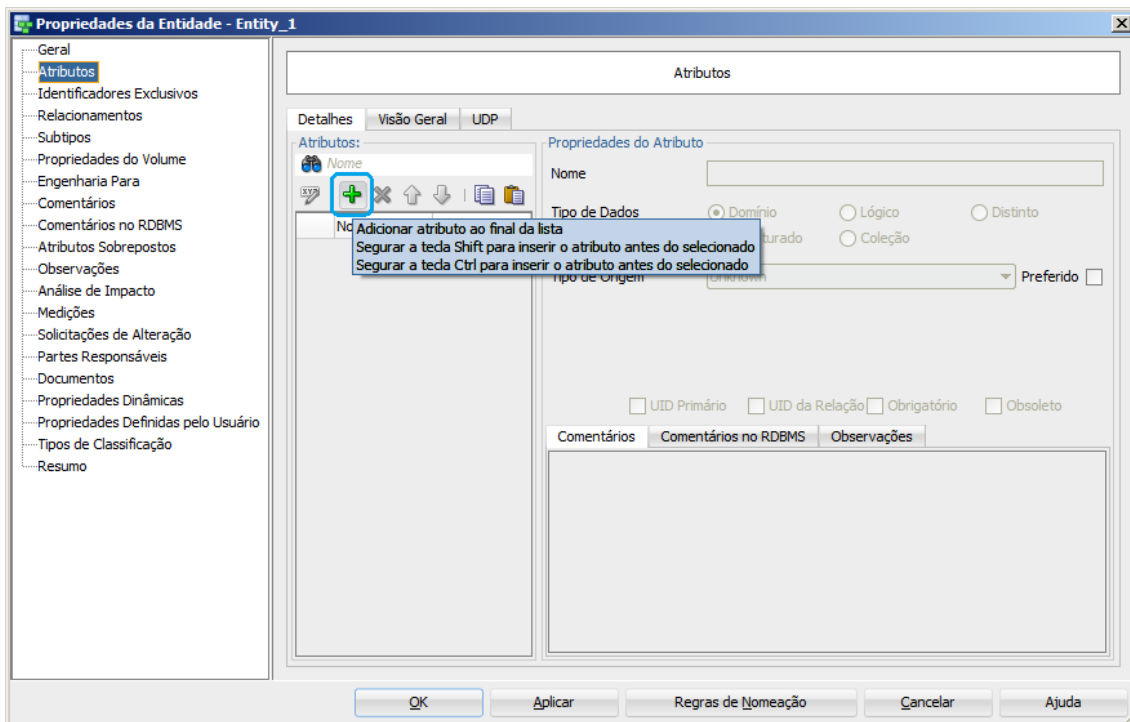


Figura 19. Formulário de cadastramento de atributos da entidade

Para cadastrar um novo atributo nesse formulário, clique no botão “Adicionar atributo ao final da lista” (indicado na **Figura 19**). Essa ação cria uma linha dentro do quadro “Atributos:”, permitindo a criação e edição de um atributo.

Vamos criar o atributo identificador (chave primária) da entidade Pessoa que estamos construindo. Para tanto, preenchemos os seguintes campos desse formulário:

- “Nome”: preencher com o valor “id pessoa”;
- “Tipo de Dados”: selecionar o botão de opção “Lógico”;
- “Tipo de Origem”: selecionar o valor “Integer” da lista suspensa;
- “UID Primário”: marcar a caixa de checagem.

Feitas essas alterações, clique no botão “Aplicar” no rodapé do formulário para confirmá-las (vide **Figura 20**).

Figura 20. Formulário de cadastramento de atributos da entidade (aplicando alterações)

Para cadastrar os atributos restantes da entidade Pessoa, repita o processo clicando no botão da cruz verde (“Adicionar atributo ao final da lista”) e preencha os campos previamente descritos com os valores dos outros atributos.

A **Tabela 1** apresenta os valores que devem ser utilizados para cadastrar os outros atributos da entidade Pessoa. Para facilitar o mapeamento dos valores dessa tabela com os locais que devem ser preenchidos no formulário, o atributo “id pessoa” que acabamos de criar aparece como modelo para preenchimento. Em todos os atributos descritos nessa tabela, devemos selecionar a opção “Lógico” no campo “Tipo de Dados”.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Valor padrão	Restrição
id pessoa	Identificador único do cliente no sistema.	sim	sim	Integer		
nome	Nome completo do cliente.		sim	Varchar (80)		
endereço	Endereço do cliente.			Varchar (120)		
cep	CEP do cliente.			Numeric (8)		
telefone	Telefone de contato do cliente.			Numeric (10)		
renda média	Renda média do cliente.		sim	Numeric (11,2)		
tipo pessoa	Tipo de pessoa associada ao cliente (Física ou Jurídica).		sim	Numeric (1)		“ck tipo pessoa”

situacao pessoa	Valor lógico (binário) que identifica se o cliente está ativo no sistema.		sim	Numeric (1)	1	"ck situacao pessoa"
--------------------	--	--	-----	----------------	---	----------------------------

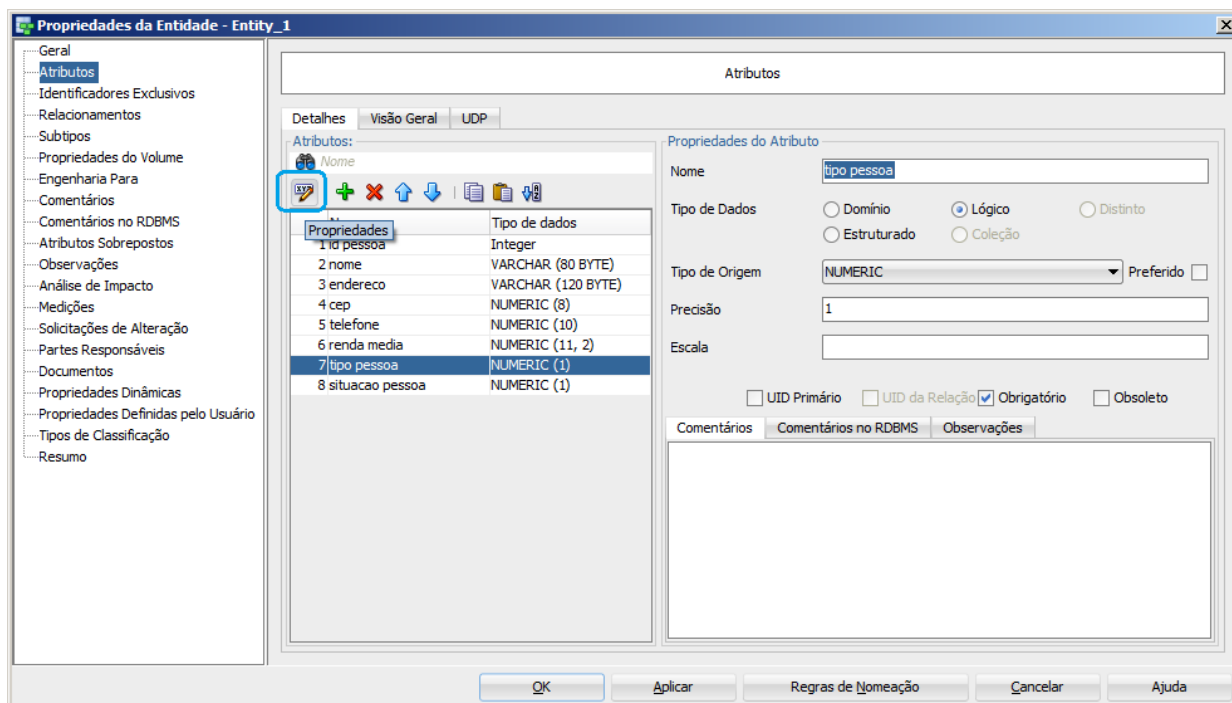
Tabela 1. Valores para atributos da entidade "Pessoa"

Analisemos as colunas Valor padrão e Restrição da **Tabela 1**. Elas indicam, respectivamente, se um atributo deve ser preenchido com um valor padrão, caso o usuário não forneça nenhum durante o preenchimento, e se o atributo possui algum tipo de regra de validação para permitir um conjunto específico de valores durante seu preenchimento.

No caso da entidade Pessoa, repare que existem restrições de preenchimento definidas para os atributos "tipo pessoa" ("ck tipo pessoa") e "situacao pessoa" ("ck situacao pessoa"). Além disso, identificamos que o atributo "situacao pessoa" tem um valor padrão definido como "1" caso nenhum outro valor seja fornecido pelo usuário do sistema.

Essas definições de Valor padrão e Restrição não podem ser configuradas diretamente no formulário de cadastramento de atributos da entidade. Para isso, é preciso editar o atributo em questão, realizando essas configurações em um formulário próprio para ele.

Verifiquemos qual é o procedimento para configurar os valores padrão e restrições relacionados com os atributos "tipo pessoa" e "situacao pessoa". Para isso, no formulário de cadastramento de atributos da entidade, selecionamos o atributo "tipo pessoa" dentro do quadro "Atributos:" e clicamos no botão "Propriedades", conforme demonstrado na **Figura 21**.



[abrir imagem em nova janela](#)

Figura 21. Seleção de propriedades de um atributo.

Um formulário de Propriedades do Atributo é aberto. No menu à esquerda desse formulário, selecione o nó "Padrão e Restrição" e então um formulário específico para essas configurações aparecerá à direita do menu. Nesse formulário, é possível definir um valor padrão para o atributo

selecionado anteriormente, além da possibilidade de definir uma restrição de preenchimento de valores para esse atributo. Entre as opções apresentadas, podemos definir uma “Restrição” codificada de forma genérica, uma “Lista de Faixas” de valores válidos – com valores iniciais e finais da faixa definida – e uma “Lista de Valores” válidos para preenchimento. Clique no botão “Valores” que se encontra à direita do campo “Lista de Valores”, conforme indicado na **Figura 22**.

Figura 22. Formulário de propriedades do atributo.

Um novo formulário é aberto para gerar a lista de valores válidos. Ao clicar no botão “Adicionar”, uma linha é aberta na lista de valores para que seja preenchida com um valor e uma descrição dele. No caso do atributo que estamos configurando, “tipo pessoa”, criamos uma lista de valores conforme descrito na **Figura 23**. Ou seja, quando a entidade Pessoa for de um tipo Pessoa Física, esse atributo recebe o valor “1”. Se for de um tipo Pessoa Jurídica, esse atributo recebe valor “2”. Terminado o cadastramento da lista de valores, pressionamos o botão “OK” (vide **Figura 23**) e retornamos ao formulário de Propriedades do Atributo. Nesse formulário, preencha o campo “Nome da Restrição:” com o valor "ck tipo pessoa", clicando em “OK” para confirmar o cadastro e fechar o formulário.

Figura 23. Cadastro de lista de valores válidos preenchida.

Cadastramos agora uma restrição para o atributo “situacao pessoa”, baseada também em uma lista de valores. No entanto, além da restrição, selecionamos um dos valores dessa lista para que seja o valor padrão do atributo caso nenhum valor seja fornecido durante seu cadastramento no sistema.

Para tanto, seguimos os mesmos passos utilizados para definir uma restrição ao atributo “tipo pessoa”. No formulário de cadastramento de atributos da entidade, selecionamos o atributo “situacao pessoa” dentro do quadro “Atributos:” e clicamos no botão “Propriedades”. Em seguida, no formulário de Propriedades do Atributo aberto, selecionamos o nó Padrão e Restrição no menu à esquerda e, em seguida, voltamos ao formulário e clicamos no botão “Valores” que se encontra à direita do campo “Lista de Valores”.

No formulário de lista de valores que aparece, cadastramos o valor “1” – para indicar uma pessoa “Ativa” na instituição bancária (que ainda possui contas não encerradas) – e o valor “2” – para sinalizar uma pessoa “Inativa” no sistema (que encerrou todas as suas contas). Ao término do cadastramento, temos a lista configurada como aparece na **Figura 24**. Clicamos no botão “OK” para fechar essa lista e retornar ao formulário de Propriedades do Atributo. Nesse formulário, devemos clicar no botão “Aplicar” para habilitar a visualização da lista de valores recém cadastrada.

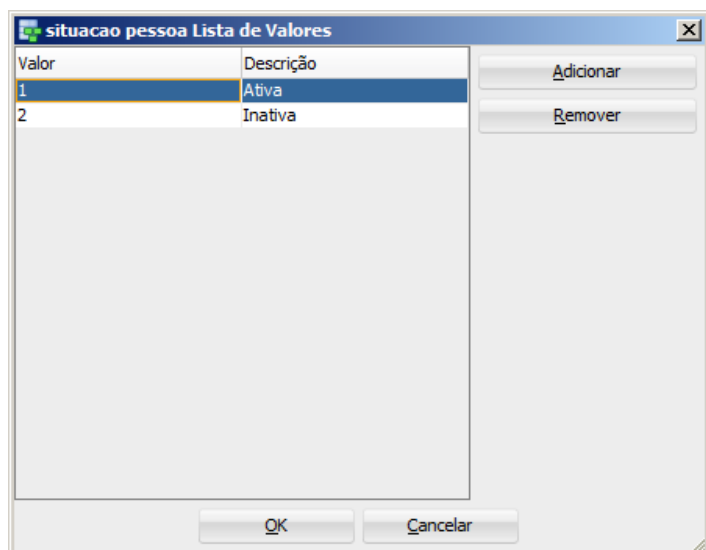
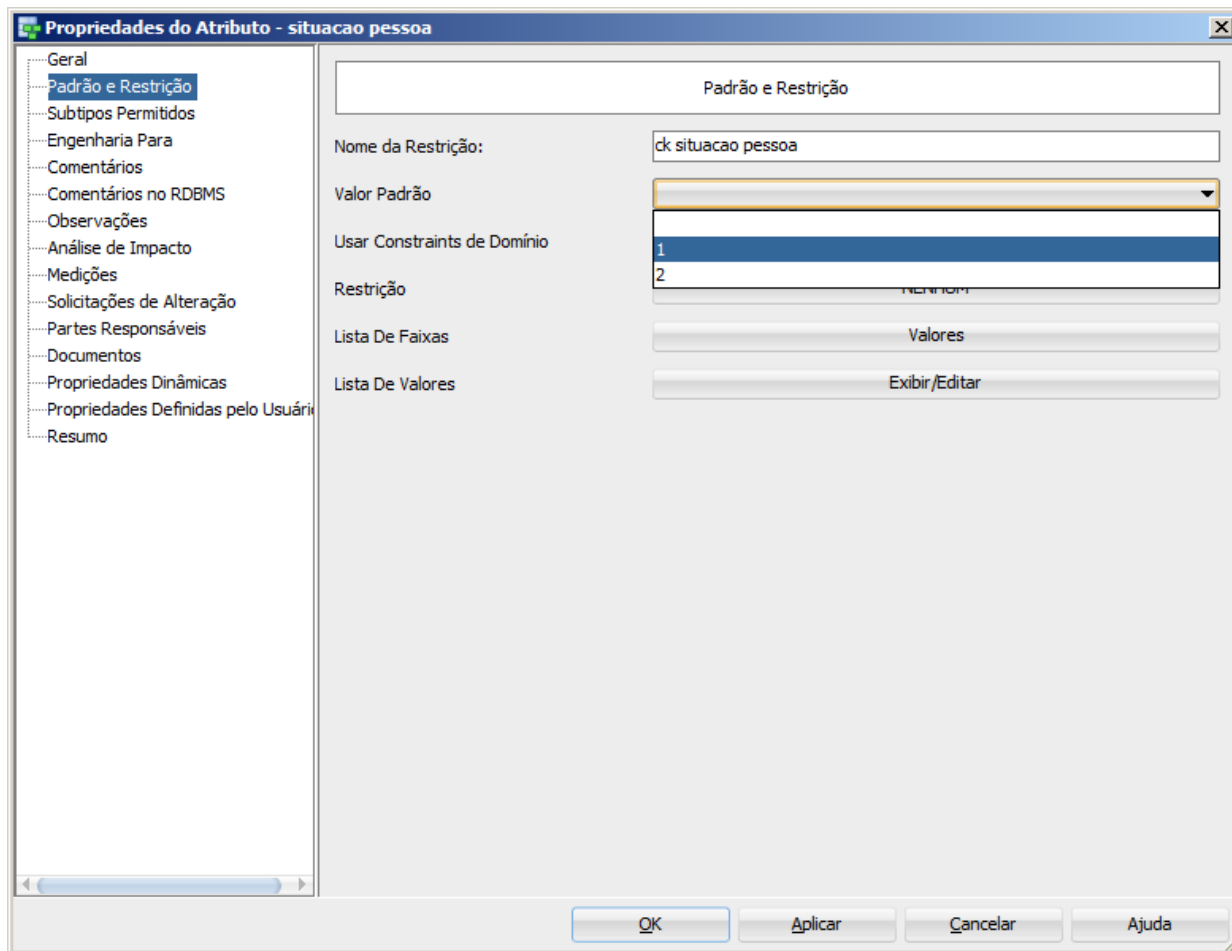


Figura 24. Cadastro de lista de valores válidos (padrão).

Com a lista de valores configurada, selecionamos um deles para que seja o padrão do atributo “situação pessoa”. Ao verificarmos o campo “Valor Padrão” presente no formulário de Propriedades do Atributo – com o nó “Padrão e Restrição” do menu à esquerda selecionado – perceberemos que uma lista combo surgiu ao lado desse campo. Nela estão configurados os itens da lista de valores cadastrada anteriormente (vide **Figura 25**). Selecionamos o item com valor “1” (“Ativa”) a fim de indicar que esse é o valor padrão para o atributo “situação pessoa”. Adotamos essa premissa porque todo cadastro de uma nova pessoa no sistema – que será um cliente bancário – implica na abertura de uma conta, indicando que, nesse momento, a pessoa torna-se automaticamente ativa na instituição bancária. Para finalizar, cadastramos um nome (“ck situacao pessoa”) no campo “Nome da Restrição:”, clicando posteriormente no botão “OK” para fechar o formulário de Propriedades do Atributo. Feito isso, retornamos ao formulário de Propriedades da Entidade, fechamos esse formulário e retornamos ao painel da área do diagrama DER (modelo lógico).



abrir imagem em nova janela

Figura 25. Formulário de Propriedades do Atributo - selecionando valor padrão

Na barra de ferramentas superior à área do diagrama, clicamos no botão “Selecionar” e redimensionamos a entidade Pessoa recém-criada, aumentando seu tamanho para que acomode visualmente todos os seus atributos. Após redimensionar a entidade, ela assume uma identidade visual semelhante à que está apresentada na **Figura 26**.

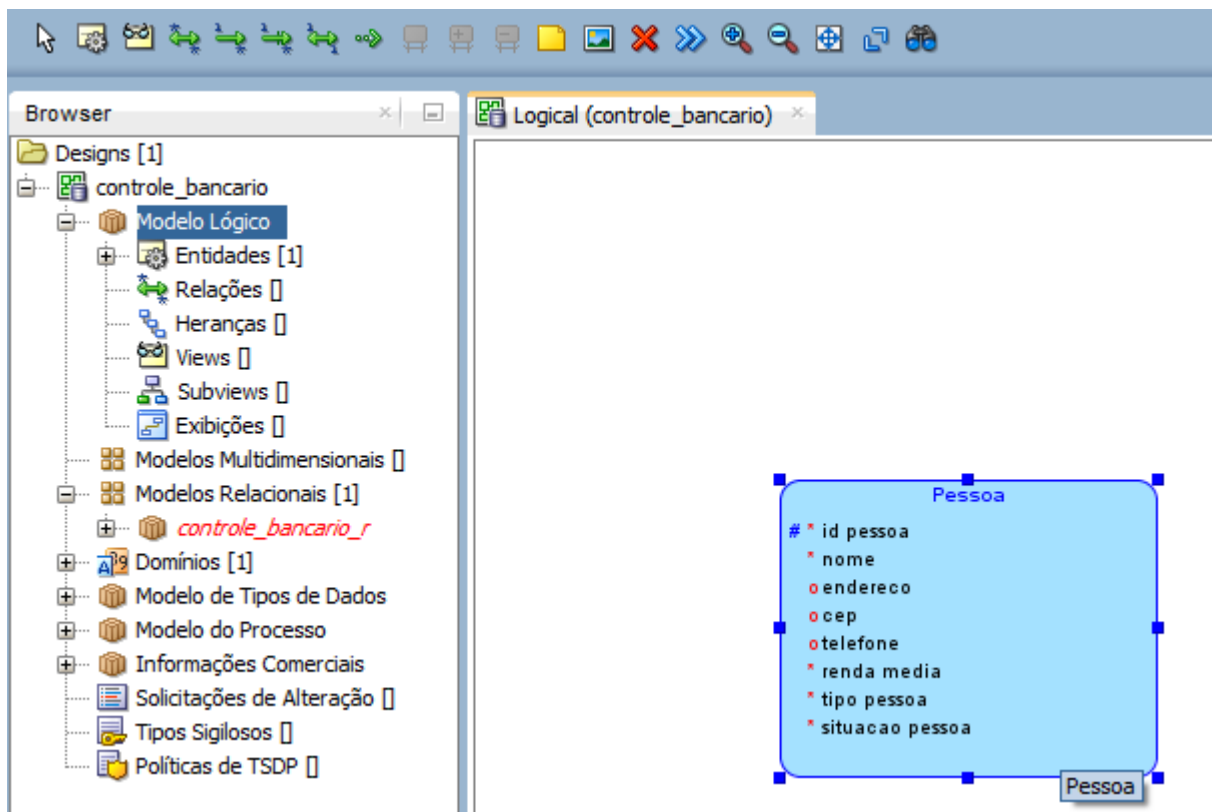


Figura 26. Entidade "Pessoa" redimensionada

Seguindo o exemplo da entidade **Pessoa**, agora cadastramos as outras entidades que compõem o modelo conceitual de nosso módulo de controle bancário. Os mesmos passos explicados até agora devem ser reproduzidos na criação dessas novas entidades. A lista com as entidades, além das características que devem ser cadastradas para cada uma delas (atributos e listas de valores, quando for o caso), é a seguinte:

- **Pessoa Física**, com a lista de atributos descrita na **Tabela 2**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Restrição
cpf	Cadastro de Pessoa Física do cliente	sim	sim	Numeric (11)	
celular	Número do celular do cliente com <u>11</u> posições: · Duas posições iniciais para o código DDD; · Nove posições restantes para o número.			Numeric (11)	
e-mail	Endereço de e-mail do cliente (se houver).			Varchar (100)	
estado civil	Estado civil do cliente.		sim	Char (1)	"ck estado civil"

Tabela 2. Valores para atributos da entidade "Pessoa Física"

o A lista de valores da restrição "ck estado civil" associada ao atributo "estado civil" está descrita na **Tabela 3**.

Atributo	Nome da restrição	Valor	Descrição
estado civil	ck estado civil	C	casado (a)
		D	divorciado (a)
		S	solteiro (a)
		V	viúvo (a)
		X	separado (a)

Tabela 3. Listas de Valores dos atributos da entidade "Pessoa Física"

· **Pessoa Jurídica**, com a lista de atributos descrita na **Tabela 4**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem
cnpj	Cadastro Nacional de Pessoa Jurídica do cliente.	sim	sim	Numeric (14)
inscr estadual	Código de Inscrição Estadual do cliente (se houver)			Numeric (12)

Tabela 4. Valores para atributos da entidade "Pessoa Jurídica"

· **Conta Comum**, com a lista de atributos descrita na **Tabela 5**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Valor padrão	Restrição
nro conta	Número da conta corrente.	sim	sim	Numeric (10)		
cod agencia	Código da agência associada à conta corrente.	sim	sim	Numeric (4)		
tipo conta	Indica o tipo de conta que será criada (comum,		sim	Numeric (1)	1	"ck tipo conta"

	especial ou poupança).					
data abertura	Data de abertura da conta.		sim	Date		
data encerramento	Data de encerramento da conta.			Date		
situacao conta	Indica se a conta está ativa ou não no sistema.		sim	Char (1)	"1"	"ck situacao conta"
senha	Senha para acesso à conta.		sim	Integer		
Saldo	Saldo monetário da conta.		sim	Numeric (11, 2)	0	

Tabela 5. Valores para atributos da entidade "Conta Comum"

A atribuição do valor "0" como "Valor Padrão" para o atributo "saldo" segue o mesmo procedimento que foi utilizado para criar o "Valor Padrão" para o atributo "situação pessoa" da entidade Pessoa. A diferença é que, nesse caso, não será criada uma "Lista de Valores" e o valor "0" será digitado diretamente no campo "Valor Padrão" do Formulário de Propriedades do Atributo;

A lista de valores "ck tipo conta" – associada ao atributo "tipo conta" – e a lista "ck situacao conta" – associada ao atributo "situacao conta" – estão descritas na **Tabela 6**.

Atributo	Nome da restrição	Valor	Descrição
tipo conta	ck tipo conta	1	Conta Comum
		2	Conta Especial
		3	Conta Poupança
situacao conta	ck situacao conta	"0"	Conta inativa
		"1"	Conta ativa

Tabela 6. Listas de Valores dos atributos da entidade "Conta Comum"

• **Conta Especial**, com a lista de atributos descrita na **Tabela 7**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Valor padrão	Restrição
limite	Limite máximo que o correntista pode sacar além do saldo de sua conta.			Numeric (11, 2)		

Tabela 7. Valores para atributos da entidade "Conta Especial"

· **Conta Poupança**, com a lista de atributos descrita na **Tabela 8**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Valor padrão	Restrição
data aniversario	Data em que o valor depositado na conta renderá juros.			Date		

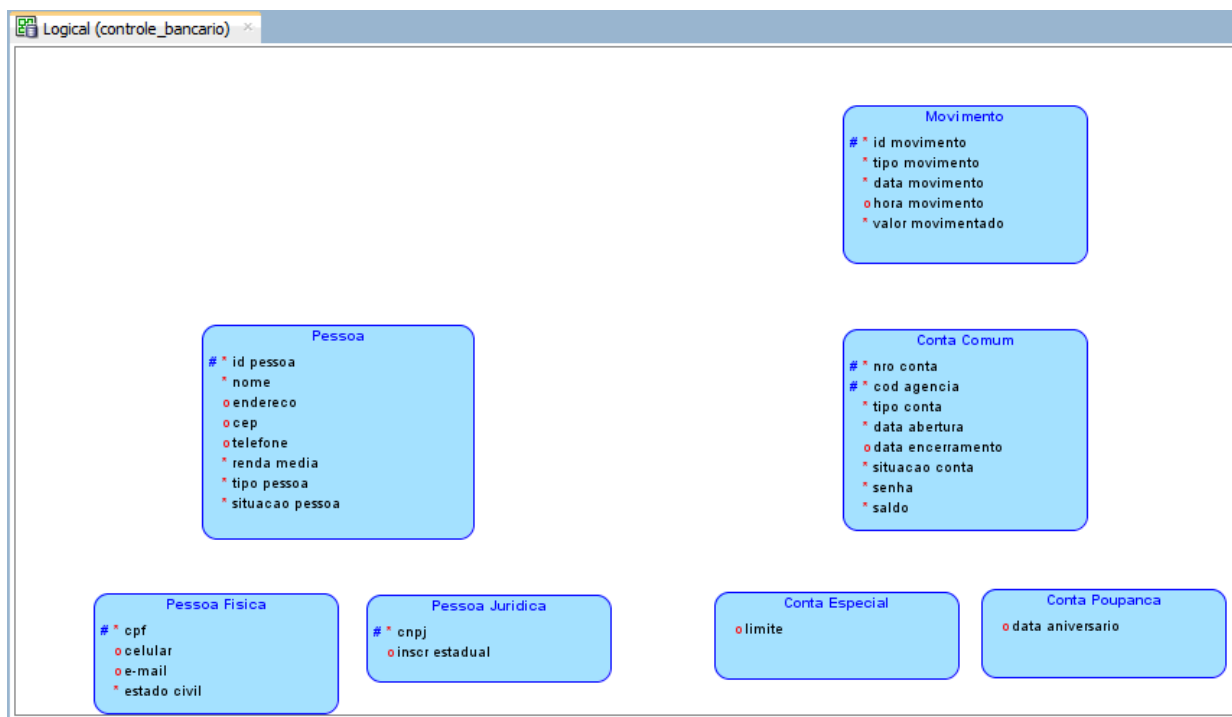
Tabela 8. Valores para atributos da entidade "Conta Poupança"

· **Movimento**, com a lista de atributos descrita na **Tabela 9**.

Nome do atributo	Descrição	UID Primário	Obrigatório	Tipo de Origem	Valor padrão	Restrição
id movimento	Identificador único de uma movimentação bancária realizada no sistema.	ü	ü	Integer		
tipo movimento	Valor que identifica o tipo de movimentação realizada na conta, podendo ser "0" (<i>saque</i>) ou "1" (<i>depósito</i>).		ü	Char (1)		
data movimento	Data do movimento.		ü	Date		
hora movimento	Horário do movimento.			Time		
valor movimentado	Valor movimentado pelo cliente.		ü	Numeric (11, 2)		

Tabela 9. Valores para atributos da entidade "Movimento"

Após a criação de todas as entidades através do SDDM, temos um DER semelhante ao apresentado na **Figura 27**.



[abrir imagem em nova janela](#)

Figura 27. DER com as entidades

Terminada a criação das entidades, identificamos quais são os relacionamentos existentes e os acrescentamos ao diagrama, utilizando os botões construtores de relacionamentos presentes na barra de ferramentas.

Antes de prosseguir com a criação dos relacionamentos, vamos preparar o ambiente gráfico da área do diagrama, a fim de permitir a visualização adequada de alguns elementos do DER. Para esse fim, abrimos um menu de contexto em qualquer área livre do diagrama.

Feito isso, selecionamos a opção “Notação”, que abrirá um novo menu de opções. Selecione a opção “Apresentação Box-in-Box” para que fique desmarcada. Essa configuração faz com que as entidades subtipo em uma hierarquia de generalização apareçam conectadas à sua respectiva entidade supertipo com uma linha de relacionamento vermelha. Caso contrário, um subtipo aparece graficamente inserido dentro de seu supertipo, como se fossem “caixas dentro de caixas”.

Em seguida, abra novamente o mesmo menu de contexto na área do diagrama e, dessa vez, selecione a opção “Mostrar”. Um novo menu de opções é aberto e selecionamos a opção “Atributos de Relacionamento” para que fique marcada. Essa configuração permite visualizar no diagrama os atributos de um relacionamento.

Após a preparação do ambiente do DER, criamos os relacionamentos entre as entidades. Iniciamos com o relacionamento entre as entidades Pessoa e Conta Comum. Nesse relacionamento, de acordo com a regra de negócio definida, uma pessoa pode ter uma ou várias contas abertas em seu nome, assim como uma conta pode ter um ou mais titulares, como no caso de contas conjuntas. Dessa forma, construímos um relacionamento “muitos-para-muitos” entre as entidades Pessoa e Conta Comum.

Na barra de ferramentas, selecionamos o botão “Nova Relação M:N”. Clicamos, em seguida, com o ponteiro do mouse em uma das entidades – Pessoa, por exemplo. Após o clique, uma linha começa a ser desenhada a partir da entidade Pessoa. Arrastamos essa linha até a entidade Conta Comum e, ao alcançarmos essa entidade, clicamos nela (vide **Figura 28**).

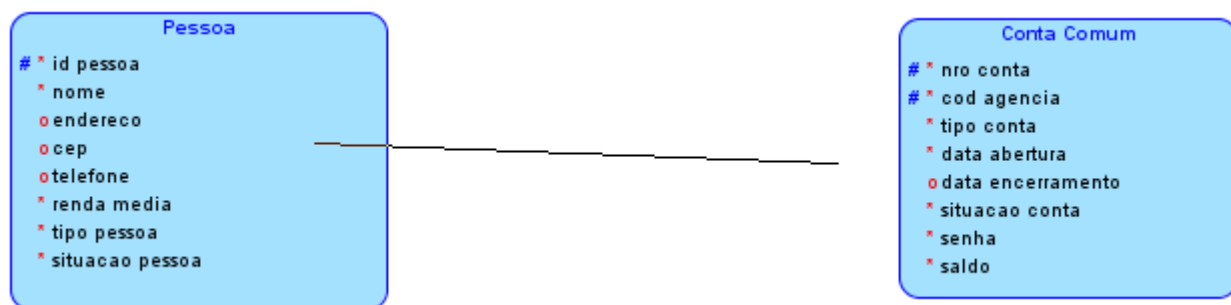


Figura 28. Criando relacionamento "muitos-para-muitos" entre entidades

Um formulário de Propriedades da Relação aparece. Nele, existe um menu hierárquico à esquerda, onde o nó Geral já aparece selecionado. Com esse nó selecionado, configuramos os detalhes principais associados com o relacionamento recém-criado, como nome do relacionamento, cardinalidade e participação. As propriedades que configuramos nessa tela são as que aparecem na **Figura 29**:

Propriedades da Relação - Relation_6

Geral

Nome: Pessoa Conta

Usar chaves substitutas: ☐

Cardinalidade de Origem

Origem: Pessoa

Chave de origem:

Nome na Origem:

Sinônimo da Entidade de Origem: Pessoa

Cardinalidade Origem para Destino: 1..*

Origem Opcional: ☐

Transferível: ☒

Atribuição Dominante: Nenhum

Identificando: ☐

Excluir Regra: NO ACTION

Cardinalidade de Destino

Alvo: Conta Comum

Chave de destino:

Nome no Destino:

Sinônimo da Entidade de Destino: Conta Comum

Cardinalidade de Destino para Orig...: *..1

Destino Opcional: ☐

Transferível: ☒

No Arco: ☐

OK Aplicar Cancelar Ajuda

abrir imagem em nova janela

Figura 29. Formulário de Propriedades da Relação

- “Nome”: preenchamos com o valor “Pessoa Conta”;
- “Cardinalidade Origem para Destino”: deve vir com o tipo correto, mas apenas confirmamos se aparece um símbolo de “pé-de-galinha”;
- “Cardinalidade Destino para Origem”: deve vir com o tipo correto, mas apenas confirmamos se também aparece um símbolo de “pé-de-galinha”;
- “Origem Opcional”: essa caixa de checagem deve estar desmarcada, pois na definição da regra de negócio do sistema, deve existir pelo menos uma Pessoa associada a uma Conta Comum;
- “Destino Opcional”: essa caixa de checagem também deve estar desmarcada, pois na definição da regra de negócio do sistema, deve existir pelo menos uma Conta Comum associada a uma Pessoa.

Terminada a configuração dessas propriedades, devemos agora criar um atributo de “titularidade” associado ao relacionamento “Pessoa Conta” que estamos criando. Esse atributo é necessário, pois no caso de uma conta conjunta – onde existe mais de um indivíduo associado a ela – é preciso definir quem é o Correntista Titular e quem são os Correntistas Adicionais dessa conta. Como essa definição está associada a uma combinação de Pessoa e Conta Comum específica, tal atributo faz parte da relação entre essas entidades.

No formulário de Propriedades da Relação, selecionamos, no menu hierárquico da esquerda, o nó Atributos. No lado direito do formulário, aparece uma tela para cadastramento de atributos do relacionamento. Para cadastrar um atributo nessa tela, seguimos os mesmos passos utilizados para cadastrar atributos no formulário Propriedades da Entidade, visto anteriormente. Vamos cadastrar um atributo nessa tela preenchendo os seguintes campos (vide **Figura 30**):

- “Nome”: preencher com o valor “titularidade”;

- “Tipo de Dados”: selecionar o botão de opção “Lógico”;
- “Tipo de Origem”: selecionar o valor “CHAR” da lista suspensa, com tamanho “1”;
- “Obrigatório”: marcar a caixa de checagem.

O formulário "Propriedades da Relação - Relation_6" apresenta a seguinte configuração para o atributo "titularidade":

Nome	Tipo de dados
1 titularidade	CHAR (1 BYTE)

Propriedades do Atributo:

- Nome: titularidade
- Tipo de Dados: ☒ Lógico, ☐ Domínio, ☐ Distinto, ☐ Estruturado, ☐ Coleção
- Tipo de Origem: CHAR (dropdown), Preferido ☐
- Tamanho: 1
- Unidades: BYTE (dropdown)
- Obrigatório: ☒

Abas de comentários: Comentários, Comentários no RDBMS, Observações.

abrir imagem em nova janela

Figura 30. Formulário de Propriedades da Relação - Tela de Atributos

Após cadastrar o atributo, definimos uma lista de valores válidos que devem ser utilizados no preenchimento dele. Para isso, na tela de atributos do formulário de Propriedades da Relação (vide **Figura 30**), devemos selecionar o atributo “titularidade” dentro do quadro “Atributos:” e clicar no botão “Propriedades” para abrirmos o formulário de Propriedades do Atributo. No menu à esquerda desse formulário, selecionamos o nó Padrão e Restrição para que a tela de configuração respectiva seja exibida. Nessa tela, clicamos no botão “Valores” que se encontra à direita do campo “Lista de Valores” a fim de abrir o formulário para geração de lista de valores. Cadastramos os itens dessa lista conforme os valores apresentados na **Figura 31**.

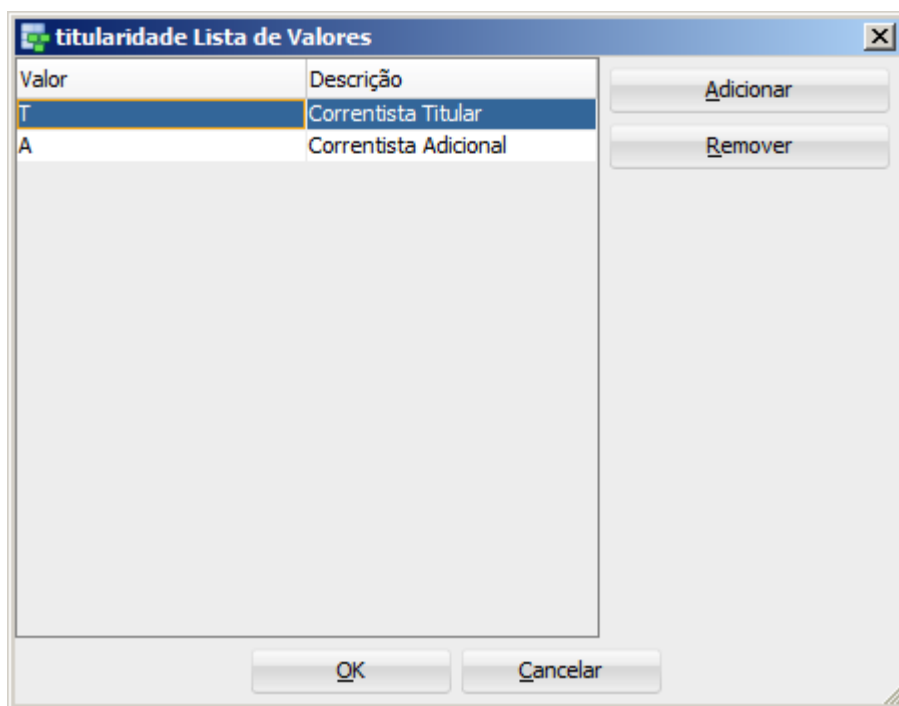


Figura 31. Cadastro de lista de valores válidos para "titularidade"

Retornamos ao formulário de Propriedades do Atributo e clicamos em “Aplicar”. Nessa tela, preenchemos o campo “Nome da Restrição” com o valor “ck titularidade” e selecionamos como “Valor Padrão” o item “T” da lista de valores previamente cadastrada. Clicamos em “OK” para retornar ao formulário de *Propriedades da Relação* e em seguida clicamos “OK” nesse outro formulário para retornarmos à área de edição do **DER**.

De volta ao painel de edição do diagrama, o relacionamento com atributo entre as entidades Pessoa e Conta Comum possui uma aparência semelhante à indicada na **Figura 32**.

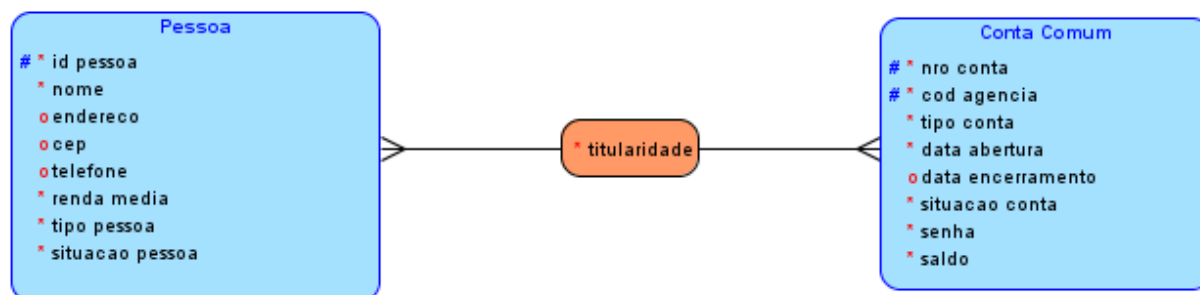


Figura 32. Relacionamento "muitos-para-muitos", com atributo

Agora vamos criar o relacionamento entre as entidades Conta Comum e Movimento. Para essa relação, a regra de negócio define que uma conta pode registrar um ou vários movimentos, enquanto um movimento pode ser registrado somente por uma única conta. Em virtude dessa premissa, devemos construir um relacionamento “um-para-muitos” entre as entidades Conta Comum e Movimento. Na barra de ferramentas, selecionamos o botão “Nova Relação 1:N”.

A seguir, clicamos com o ponteiro do mouse na entidade de origem – a “entidade pai” do relacionamento – que, neste caso, é a entidade Conta Comum. Após o clique, uma linha começa a ser desenhada a partir da entidade Conta Comum. Arrastamos essa linha até a entidade de destino – a “entidade filha” do relacionamento – que, neste caso, é Movimento. Ao alcançarmos essa entidade, clicamos nela.

Como já foi visto anteriormente, um formulário de Propriedades da Relação aparece, com o nó Geral já selecionado. Repare nas propriedades que foram configuradas nessa tela, de acordo com a **Figura 33**.

Figura 33. Formulário de Propriedades da Relação - "um-para-muitos"

O “Nome” atribuído a essa relação foi “Registra”. As caixas de checagem “Origem Opcional” e “Destino Opcional” aparecem desmarcadas, pois em ambas as extremidades do relacionamento devem existir ao menos uma entidade de cada tipo para que a relação seja válida. No campo de “Cardinalidade Origem para Destino” já aparece selecionado um conector com o símbolo do “pé-de-galinha” para o lado “muitos” da relação. A novidade nessa tela é o conector selecionado no campo “Cardinalidade Destino para Origem”, que aparece com o símbolo de um “traço único”, que representa o lado “um” do relacionamento.

Finalizadas as configurações de propriedade desse relacionamento, clicamos no botão “OK” e retornamos à área de edição do DER. Ao retornarmos, encontramos um relacionamento desenhado entre as entidades Conta Comum e Movimento, como o que aparece na **Figura 34**.

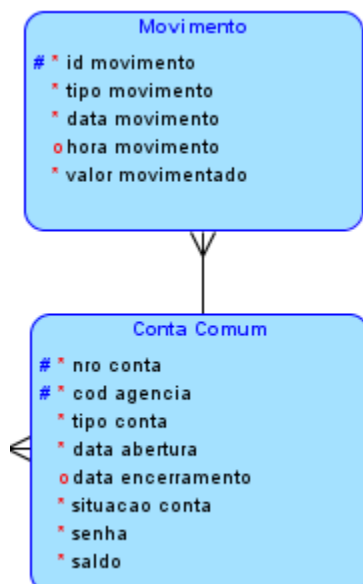


Figura 34. Relacionamento "um-para-muitos"

Para finalizarmos nosso modelo lógico, resta-nos apenas a criação das duas hierarquias de generalização existentes:

- A generalização existente entre o supertipo Pessoa e os subtipos Pessoa Física e Pessoa Jurídica;
- A generalização existente entre o supertipo Conta Comum e os subtipos Conta Especial e Conta Poupança.

Essa definição é realizada através de uma configuração no formulário de Propriedades da Entidade na entidade subtipo, informando-se no campo "Supertipo" quem é a entidade à qual ela se conecta, em esquema de herança. Veja o exemplo da **Figura 35**, onde informamos na entidade Pessoa Física que ela tem como supertipo a entidade Pessoa.

A imagem mostra a janela de configuração 'Propriedades da Entidade - Pessoa Física'. No menu lateral à esquerda, o item 'Supertipo' está selecionado e destacado com um retângulo verde. No painel principal, o campo 'Nome' contém o texto 'Pessoa Física' e também está destacado com um retângulo verde. O campo 'Supertipo' é uma lista suspensa que mostra a entidade 'Pessoa' selecionada, também destacada com um retângulo verde. Outras opções visíveis na lista são 'Conta Corrente', 'Conta Especial', 'Conta Poupança', 'Movimento' e 'Pessoa Jurídica'. O campo 'Nome Longo' contém o texto 'Pessoa Física'. No rodapé da janela, há botões para 'OK', 'Aplicar', 'Regras de Nomeação', 'Cancelar' e 'Ajuda'.

Figura 35. Informando "supertipo" associado a uma entidade "subtipo"

Realizamos o mesmo tipo de configuração na entidade **Pessoa Jurídica**, informando que ela tem como *supertipo* a entidade **Pessoa**. Finalizadas essas configurações, temos uma representação de hierarquia de generalização semelhante à representada na **Figura 36**.

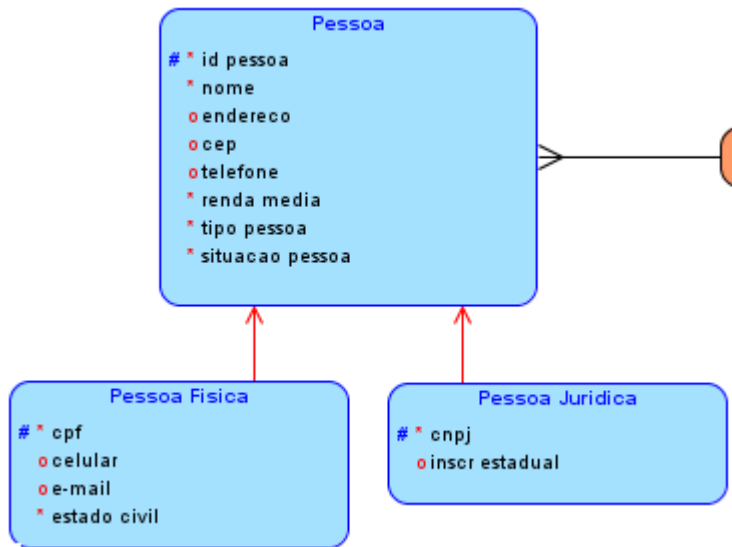


Figura 36. Hierarquia de generalização

No entanto, de acordo com as regras de negócio do módulo de controle bancário, essa hierarquia de generalização é composta por subtipos disjuntos, ou seja, uma Pessoa Física não pode ser ao mesmo tempo uma Pessoa Jurídica e vice-versa. Para demonstrarmos essa regra no diagrama, de acordo com o que foi visto na introdução, devemos colocar um arco sobre as linhas de relacionamento vermelhas que ligam as entidades Pessoa Física e Pessoa Jurídica à entidade Pessoa.

Para realizarmos essa atualização no diagrama, devemos clicar na entidade supertipo Pessoa no DER e, automaticamente, será habilitado o botão “Novo Arco” na barra de ferramentas. Ao clicarmos nesse botão, um arco é colocado sobre as linhas de relacionamentos, conforme demonstrado na **Figura 37**.

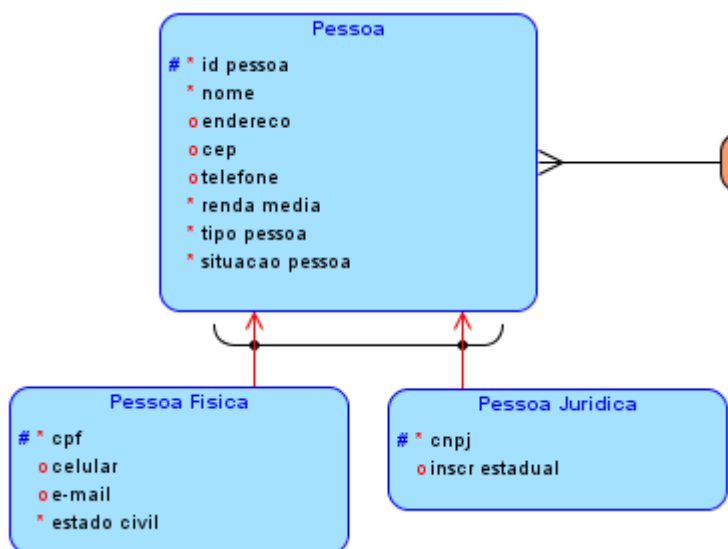


Figura 37. Representação gráfica de uma hierarquia de generalização com subtipos disjuntos.

Todos os passos utilizados para representar graficamente uma hierarquia de generalização com subtipos disjuntos devem ser repetidos para a entidade supertipo Conta Comum e suas respectivas entidades subtipo Conta Especial e Conta Poupança.

Esses últimos passos finalizam o DER do nosso modelo lógico para o módulo de controle bancário. O diagrama final apresenta um aspecto semelhante ao mostrado na **Figura 38**. Na próxima seção, mostramos como obter o DED a partir do DER construído nesta fase.

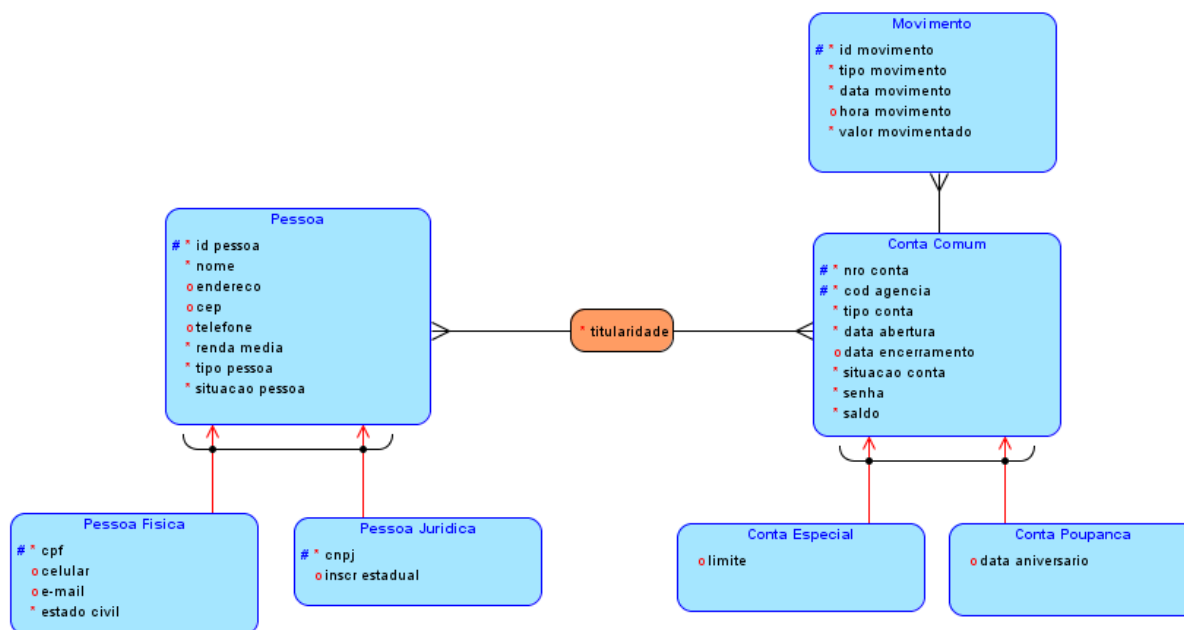


Figura 38. DER final para o "Módulo de Controle Bancário"

Modelo Físico: Derivando o DED

A partir do DER que construímos na seção anterior, obtemos o DED através da migração das definições de entidades, atributos e identificadores de chave do modelo lógico para seus equivalentes relacionais, como tabelas, campos, chaves primárias (primary keys), chaves estrangeiras (foreign keys) e índices. O SDDM tem a capacidade de gerar um modelo relacional (o equivalente do modelo físico nessa ferramenta) a partir de um modelo lógico previamente definido, sendo que após a geração é possível realizar ajustes no DED obtido.

Para gerarmos um DED de forma automática com o auxílio do SDDM, pressionamos o botão "Engenharia para Modelo Relacional" na barra de ferramentas e o respectivo é exibido, conforme a **Figura 39**. Podemos perceber, no topo do formulário, duas janelas com árvores hierárquicas: a da esquerda apresenta grupos de nós com os elementos constituintes do modelo lógico, enquanto que a da direita apresenta grupos de nós com elementos do modelo relacional. Perceba que, para cada nó da árvore da esquerda, o seu elemento correspondente aparece na mesma linha horizontal da árvore da direita, ou seja, as entidades são migradas para tabelas, as hierarquias de entidades são convertidas em tabelas mapeadas para hierarquias, enquanto que as relações são associadas a objetos mapeados para relações (relacionamentos de chave estrangeira ou tabelas associativas). Ao expandirmos cada um desses nós, em ambas as árvores, encontramos os objetos do modelo lógico, que derivam um objeto correspondente no modelo relacional.

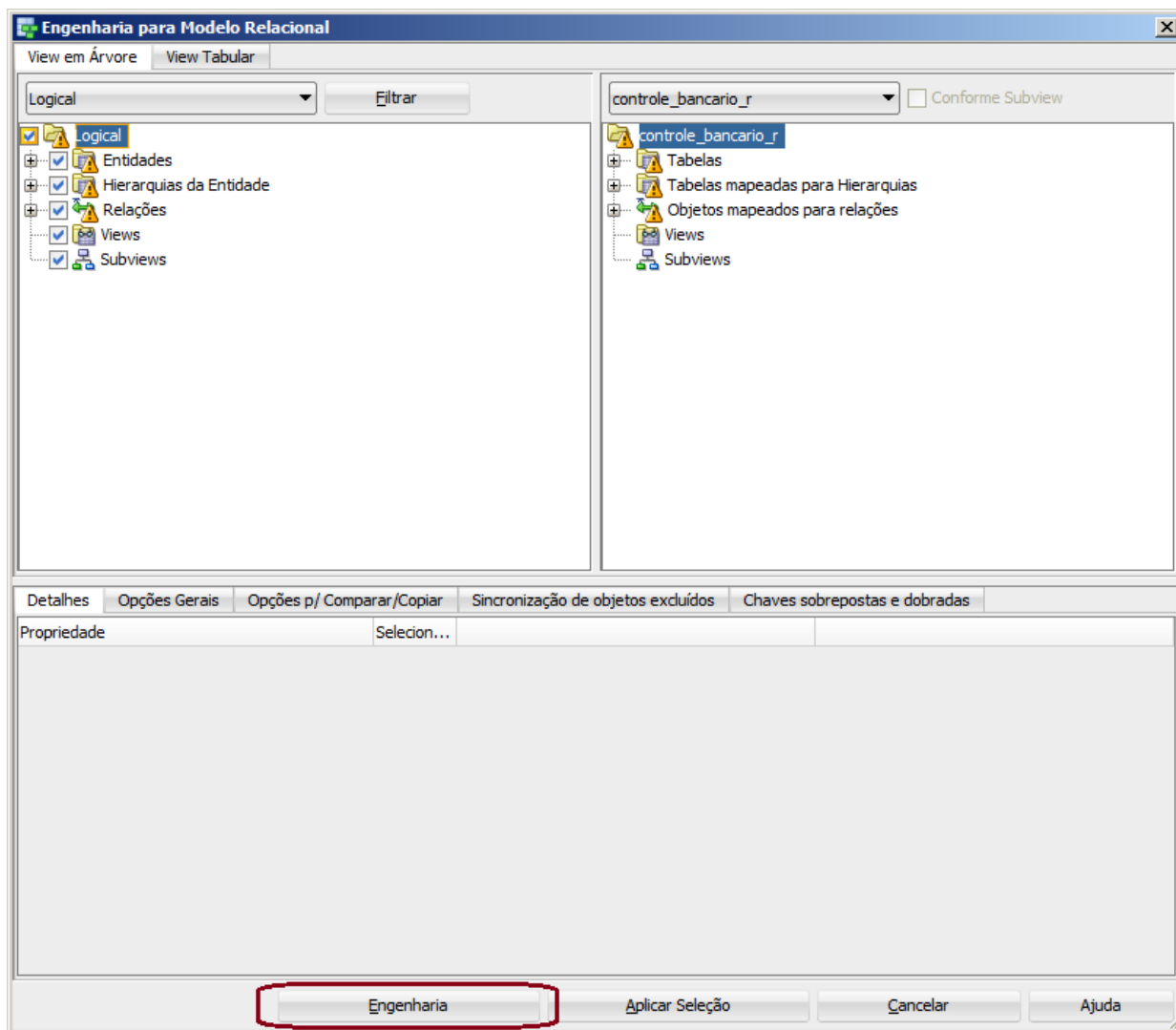


Figura 39. Formulário "Engenharia para Modelo Relacional"

Ao clicarmos no botão “Engenharia”, visualizamos algumas mensagens de processamento do SDDM indicando que ele está gerando o diagrama do modelo relacional. Após esse processamento, uma nova aba que contém o DED que foi gerado a partir do DER previamente desenhado é apresentada (**Figura 40**).

Alguns pontos a observar na migração do DER (**Figura 38**) para o DED (**Figura 40**), a fim de comparar esse resultado com as regras de migração apresentadas na seção Modelo Físico deste artigo, são:

- Todas as entidades do DER tornaram-se tabelas no DED, conforme definido pela regra 1;
- O relacionamento “muitos-para-muitos” contendo o atributo “titularidade” no DER transformou-se na tabela associativa Pessoa_Conta no DED, conforme descrito nas regras 2 e 3;
- A hierarquia de generalização da entidade supertipo Pessoa com os subtipos disjuntos Pessoa Física e Pessoa Jurídica do DER geraram tabelas correspondentes no DED, sendo que as tabelas Pessoa_Física e Pessoa_Jurídica (subtipos) herdaram os atributos de chave primária da tabela Pessoa (supertipo). Além disso, os relacionamentos de generalização do DER tornaram-se relacionamentos “um-para-um” no DED, de forma que apresentam dois arcos sobrepostos a eles, conforme definido pela regra 5 de migração. O mesmo tipo de raciocínio deve ser aplicado para a migração da entidade supertipo Conta Comum e seus respectivos subtipos Conta Especial e Conta Poupança.

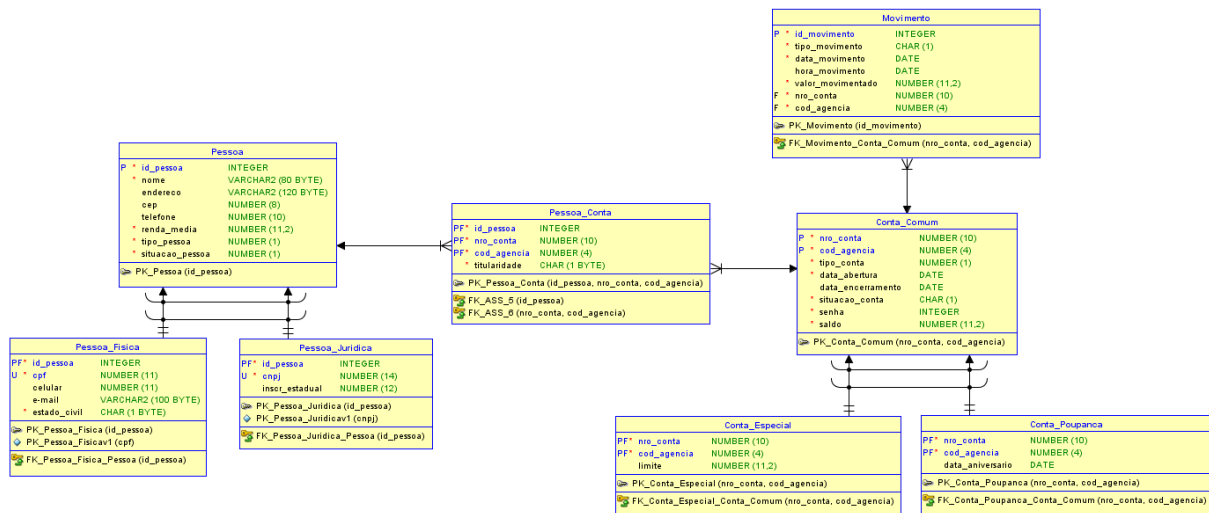
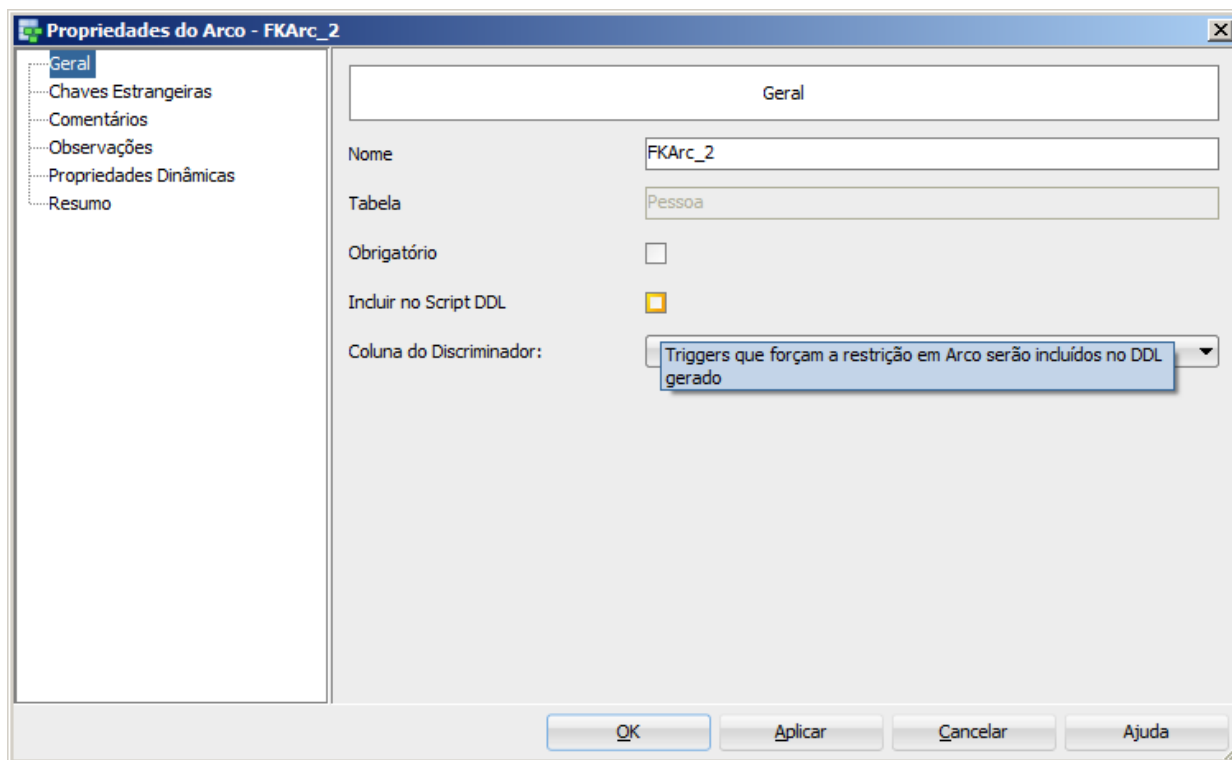


Figura 40. DED resultante para o "Módulo de Controle Bancário"

Um detalhe importante a respeito dos arcos colocados sobre os relacionamentos de hierarquia de generalização com subtipos disjuntos é que eles definem restrições nesse modelo de hierarquia, indicando que os subtipos em questão não podem ocorrer simultaneamente em uma mesma entidade. Citando um exemplo do modelo de dados que definimos, uma Pessoa Física não pode ser uma Pessoa Jurídica ao mesmo tempo e vice-versa. Para impor essa restrição de forma efetiva, o que normalmente se faz é definir triggers associados às tabelas desses arcos para forçar essa validação. O trigger é um código desenvolvido em uma linguagem procedural específica de um SGBD que é associado a uma tabela ou evento do banco de dados. Quando uma tabela tem seus dados alterados ou um evento do banco é deflagrado, seus *triggers* associados são “disparados”, executando o código contido neles de forma automática.

Não é escopo deste artigo tratar da definição e codificação de triggers. Portanto, devemos realizar uma configuração nos quatro arcos do diagrama para inibir a criação desses triggers, ou teremos inconsistências ao gerar o script de criação dos objetos na próxima seção.

Para desativar a inclusão dos triggers, executamos um duplo clique sobre um dos arcos do DED. Um formulário Propriedades do Arco é exibido. Nesse formulário, a caixa de checagem do lado direito da opção “Incluir no Script DDL” deve estar desmarcada (vide **Figura 41**). Isso garante que o SDDM não tentará encontrar triggers associados ao arco e incluí-los no script de criação dos objetos do modelo de dados. Clicamos em “OK” no rodapé desse formulário para confirmarmos a alteração e fechá-lo. Repetimos essa operação nos outros três arcos restantes. Finalizadas as alterações nos arcos, podemos proceder com a geração do script de criação dos objetos do modelo de dados.



abrir imagem em nova janela

Figura 41. Formulário de "Propriedades do Arco"

Gerando script DDL do modelo de dados

Com os ajustes do DED encerrados, vamos gerar um script DDL (Data Definiton Language) para criação dos objetos do modelo de dados no Oracle Database 11g, que foi o SGBD selecionado para o modelo relacional desse sistema.

Com a aba do diagrama DED selecionada, visualizamos na parte superior do SDDM a barra de ferramentas para edição do DED. Selecionamos então o botão “Gerar DDL” e o formulário Editor de Arquivo DDL é aberto.

Nesse formulário, temos uma prévia do script DDL que é gerado para fins de verificação, procedendo posteriormente com sua gravação em um arquivo texto contendo as instruções SQL de criação de tabelas e restrições do script em questão. Percebam que no canto superior esquerdo desse formulário aparecem duas listagens combo com valores pré-selecionados: o primeiro deles refere-se ao SGBD para o qual o script é gerado – que no caso é o Oracle Database 11g – e o segundo informa o nome do modelo relacional que origina as instruções DDL para geração do script – que é o modelo “controle_bancario_r”, cujo nome foi definido no processo de configuração do ambiente do SDDM.

Para iniciarmos o processo de geração do script, devemos pressionar o botão “Gerar” no topo desse formulário. Após pressionar esse botão, é exibido o formulário Opções de Geração de DDL.

Nesse formulário, além de opções específicas de configuração da forma como o script é gerado, também temos a opção de selecionar os objetos que são incluídos nesse script, através da árvore hierárquica apresentada na aba “Seleção ‘Criar’”, cujos nós indicam os tipos de objetos a incluir. Como iremos gerar um script para a criação inicial de nossa base de dados – que deve conter todos os objetos – todas as opções da árvore hierárquica devem permanecer marcadas.

Com todas as opções da árvore selecionadas, pressionamos o botão “OK” no rodapé do formulário “Opções de Geração de DDL”. Após o fechamento do formulário, devemos aguardar um breve

momento de processamento e então retornamos ao formulário Editor de Arquivo DDL, só que desta vez a área do editor contém as instruções SQL do script gerado.

Para salvarmos esse script em um arquivo texto, pressionamos o botão “Salvar” no rodapé desse formulário. Uma caixa de diálogo é exibida para escolhermos um nome para o script e um local em disco para gravá-lo.

Na **Listagem 1**, temos o trecho inicial do script DDL, onde podemos identificar instruções de criação de uma tabela do modelo, bem como a geração das restrições de checagem associadas as listas de valores (definidas no modelo lógico) e a restrição de chave primária associada à tabela previamente criada.

Listagem 1. Script DDL do "Módulo de Controle Bancário" - Trecho inicial

```
-- Gerado por Oracle SQL Developer Data Modeler 4.1.0.881
-- em:      2015-11-04 19:52:36 BRST
-- site:    Oracle Database 11g
-- tipo:    Oracle Database 11g

CREATE TABLE Conta_Comum
(
  nro_conta      NUMBER (10) NOT NULL ,
  cod_agencia    NUMBER (4) NOT NULL ,
  tipo_conta     NUMBER (1) DEFAULT 1 NOT NULL ,
  data_abertura  DATE NOT NULL ,
  data_encerramento DATE ,
  situacao_conta CHAR (1) DEFAULT '1' NOT NULL ,
  senha          INTEGER NOT NULL ,
  saldo          NUMBER (11,2) DEFAULT 0 NOT NULL
);

ALTER TABLE Conta_Comum ADD CONSTRAINT "ck tipo conta" CHECK ( tipo_conta      IN (1,
2, 3));

ALTER TABLE Conta_Comum ADD CONSTRAINT "ck situacao conta" CHECK ( situacao_conta
IN ('0', '1')) ;

ALTER TABLE Conta_Comum ADD CONSTRAINT PK_Conta_Comum PRIMARY KEY (
nro_conta, cod_agencia ) ;

CREATE TABLE Conta_Especial
(
  nro_conta      NUMBER (10) NOT NULL ,
  cod_agencia    NUMBER (4) NOT NULL ,
  limite        NUMBER (11,2)
);
```



```

ALTER TABLE Conta_Especial ADD CONSTRAINT PK_Conta_Especial PRIMARY KEY (
nro_conta, cod_agencia ) ;

CREATE TABLE Conta_Poupanca
(
nro_conta      NUMBER (10) NOT NULL ,
cod_agencia    NUMBER (4) NOT NULL ,
data_aniversario DATE
) ;

ALTER TABLE Conta_Poupanca ADD CONSTRAINT PK_Conta_Poupanca PRIMARY KEY (
nro_conta, cod_agencia ) ;

...

```

Na **Listagem 2**, temos o trecho final do script DDL, onde podemos identificar instruções de criação das restrições de chave estrangeira associadas aos relacionamentos entre as tabelas, além de um relatório de resumo (no estilo de comentários) do SDDM, com uma estatística dos tipos de objetos presentes no script e suas respectivas quantidades.

Listagem 2. Script DDL do "Módulo de Controle Bancário" - Trecho final

```

...

ALTER TABLE Pessoa_Conta ADD CONSTRAINT FK_ASS_5 FOREIGN KEY ( id_pessoa )
REFERENCES Pessoa ( id_pessoa ) ;

ALTER TABLE Pessoa_Conta ADD CONSTRAINT FK_ASS_6 FOREIGN KEY ( nro_conta,
cod_agencia ) REFERENCES Conta_Comum ( nro_conta, cod_agencia ) ;

ALTER TABLE Conta_Especial ADD CONSTRAINT FK_Conta_Especial_Conta_Comum
FOREIGN KEY ( nro_conta, cod_agencia ) REFERENCES Conta_Comum ( nro_conta,
cod_agencia ) ;

ALTER TABLE Conta_Poupanca ADD CONSTRAINT FK_Conta_Poupanca_Conta_Comum
FOREIGN KEY ( nro_conta, cod_agencia ) REFERENCES Conta_Comum ( nro_conta,
cod_agencia ) ;

ALTER TABLE Movimento ADD CONSTRAINT FK_Movimento_Conta_Comum FOREIGN KEY (
nro_conta, cod_agencia ) REFERENCES Conta_Comum ( nro_conta, cod_agencia ) ;

ALTER TABLE Pessoa_Fisica ADD CONSTRAINT FK_Pessoa_Fisica_Pessoa FOREIGN KEY (
id_pessoa ) REFERENCES Pessoa ( id_pessoa ) ;

ALTER TABLE Pessoa_Juridica ADD CONSTRAINT FK_Pessoa_Juridica_Pessoa FOREIGN KEY
( id_pessoa ) REFERENCES Pessoa ( id_pessoa ) ;

```

```
-- Relatório do Resumo do Oracle SQL Developer Data Modeler:
```

```
--
```

```
-- CREATE TABLE          8
-- CREATE INDEX            0
-- ALTER TABLE           23
-- CREATE VIEW             0
-- ALTER VIEW              0
-- CREATE PACKAGE          0
-- CREATE PACKAGE BODY      0
-- CREATE PROCEDURE         0
-- CREATE FUNCTION         0
-- CREATE TRIGGER           0
-- ALTER TRIGGER           0
-- CREATE COLLECTION TYPE   0
-- CREATE STRUCTURED TYPE   0
-- CREATE STRUCTURED TYPE BODY 0
-- CREATE CLUSTER           0
-- CREATE CONTEXT           0
```

```
...
```

O modelo entidade-relacionamento é, até hoje, considerado referência na modelagem de bancos de dados relacionais. Com ele, é possível criar uma visão global do modelo de dados através do modelo conceitual, diagramar essa visão no modelo lógico e definir suas estruturas através do modelo físico. O processo empregado durante a modelagem permite cobrir todo o ciclo de implementação do banco, desde a sua concepção até a criação física de suas estruturas em um SGBD.

Dentro desse contexto, o SQL Developer Data Modeler se mostrou ser uma ferramenta muito versátil durante a elaboração do estudo de caso apresentado nesse artigo. Pudemos gerar um DER com um nível de detalhamento que representou todas as entidades e relacionamentos envolvidos na definição conceitual do sistema modelado. Além disso, a ferramenta foi capaz de derivar um DED consistente a partir do DER modelado. Ao final, o SDDM possibilitou a geração de um script DDL com a sintaxe adequada a um SGBD previamente associado ao nosso modelo relacional.