

Nesta seção você encontra artigos sobre projeto, análise ou modelagem de dados



Modelagem de dados para um sistema de controle de orçamento

O objetivo deste artigo é demonstrar o processo de construção do modelo conceitual e lógico de um banco de dados relacional. Para isso, utilizaremos um estudo de caso hipotético, que consiste na criação de um sistema de controle de orçamentos. Este sistema permite a inclusão de um orçamento mensal que é composto por contas de receita e despesa. Estas contas são definidas pelo usuário e classificadas em grupos e subgrupos que geram o plano de contas do sistema. Além do orçamento mensal, todos os recebimentos e pagamentos são lançados diariamente e vinculados à conta cadastrada no plano de contas. Desta forma, têm-se a possibilidade de análise do previsto através do orçamento mensal e da análise do que efetivamente foi realizado através dos lançamentos diários nas contas.

Para a construção do modelo de dados, seguiremos uma metodologia de desenvolvimento composta pelas etapas de: levantamento de dados, definição de entidades e associações, definição

De que se trata o artigo?

Apresenta todos os passos para a modelagem de dados de um sistema de controle de orçamento, desde a elaboração do modelo conceitual até chegarmos ao modelo físico.

Para que serve?

A metodologia descrita no artigo pode ser seguida por outros desenvolvedores para a modelagem de dados para qualquer tipo de aplicação. Além disso, apresenta um estudo de caso como base que pode ser consultado em caso de dúvidas.

Em que situação o tema é útil?

A modelagem de dados é uma atividade essencial no desenvolvimento de software, e se ela for feita de forma inadequada pode resultar em problemas graves e impossibilidade de evolução no modelo após a construção do software.



Moacir Solano Kichel

moacir.kichel@gmail.com

Atua no ramo de tecnologia da informação há mais de 19 anos, presta consultoria a empresas, é coordenador do núcleo de tecnologia da informação, professor da cadeira de Banco de Dados e pós-graduação da UnC - Universidade do Contestado - campus Concórdia/SC, possui graduação em gestão da tecnologia da informação, pós-graduação em administração da informação e mestrado em ciência da computação.

de atributos e identificadores, criação do modelo conceitual, definição das cardinalidades e, por fim, a criação do modelo lógico de dados.

Modelo Conceitual

Um modelo conceitual descreve a estrutura do banco de dados sem preocupar-se com o Sistema Gerenciador de Banco de Dados (SGBD) que será utilizado. Este modelo expressa quais dados aparecerão em um Banco de Dados (BD), e não como estes estão armazenados. Normalmente ele é usado para se representar todo o conjunto de informações disponíveis em um BD a ser projetado.

Considerado como um padrão para a modelagem conceitual, o modelo Entidade-Relacionamento (comumente chamado de **modelo ER**) foi criado por Peter Chen em 1976 e é ainda hoje serve como base para vários modelos, inclusive modelos orientado a objetos. Ele percebe o mundo como sendo um conjunto de entidades e relações entre estas. Uma entidade nada mais é do que a representação de algum fato ou componente de um sistema. Usualmente, suas propriedades são representadas graficamente através de um Diagrama Entidade-Relacionamento (DER), onde os **retângulos** representam as entidades e os **losangos** interligados por **linhas** representam as associações. O relacionamento entre as entidades a partir destas associações caracteriza uma instância ou ocorrência de relacionamento. A identificação de uma instância de relacionamento entre as entidades é gerada a partir da utilização do **atributo identificador**. A **Figura 1** apresenta todos os elementos e símbolos que compõem um diagrama de ER.

O número de ocorrências de determinada entidade que podem estar associadas a ocorrências de outra entidade chama-se cardinalidade. Devemos considerar dois tipos de cardinalidade: a mínima e a máxima. Conforme representado na **Figura 1**, a cardinalidade mínima é "0" (zero) e a cardinalidade máxima é "n".

Após a definição das cardinalidades, temos como identificar se a entidade é forte ou fraca. Quando esta for fraca, a linha de ligação aparece em destaque (mais forte) conforme representado na **Figura 1**. Para melhor compreender o uso de cardinalidades e da definição de entidade forte ou fraca, vejamos o exemplo a seguir:

O domínio deste exemplo é a relação entre entidades *Cliente* e *Compra*, onde

a cardinalidade mínima entre elas é "0" (zero) e a máxima "n". Pela cardinalidade mínima, podemos afirmar que uma ocorrência de *Cliente* não obrigatoriamente precisa estar associada a uma ocorrência da entidade *Compra*. Pela cardinalidade máxima, podemos afirmar que várias ocorrências (um número máximo, porém não conhecido de ocorrências) da entidade *Cliente* podem estar relacionadas à entidade *Compra*.

Simplificando, podemos dizer que podem ser cadastrados clientes sem a obrigatoriedade destes terem registro de compra – também conhecida como associação opcional da entidade *Cliente*. Por outro lado, podemos dizer que cada compra possuirá no máximo um cliente. Como a entidade *Compra* depende de obrigatoriamente de uma ocorrência na entidade *Cliente*, dizemos que a entidade *Compra* é a entidade fraca, como observado na **Figura 2**.

Ao desenvolvermos o DER, temos uma melhor visão das relações existentes entre as entidades e desta forma facilita-se a construção do modelo lógico. Entretanto, para melhor compreendermos esta afirmação, vejamos a seguir algumas definições e características de qualquer modelo lógico de BD.

Modelo Lógico

O modelo lógico se preocupa em representar a estrutura dos dados e suas relações em um BD. Ele consiste em criar uma abstração da realidade do BD, enfatizando os objetos que dele pertencem, as relações entre estes objetos e os dados que serão armazenados.

Diferente do modelo conceitual que cria uma abstração da realidade voltada ao projeto do BD, o modelo lógico possui uma representação mais próxima da utilizada pelos SGBDs e, portanto, mais conhecida. Desta forma, muitos

desenvolvedores criam os modelos lógicos de dados diretamente nestes. Entretanto, focar apenas no SGBD poderá limitar consideravelmente a percepção das relações e dados que deverão fazer parte do BD, e por isso o desenvolvimento do modelo conceitual é tão importante, pois foca apenas no propósito do BD sem considerar o SGBD que será utilizado.

A maioria dos SGBDs utiliza-se de arquiteturas de Banco de Dados Relacionais (BDR), mas uma outra possibilidade seria o uso de Banco de Dados Orientados a Objetos (ver **Nota DevMan 1**). A arquitetura de um banco de dados relacional é constituída por objetos chamados de **tabela** e suas **relações** com as demais em um mesmo BD, constituindo o modelo lógico chamado de **Modelo Relacional (MR)**. Aqui, percebemos claramente a relação do modelo lógico com o modelo conceitual, pois as **entidades** nada mais são do que as possíveis tabelas de um MR, e suas associações, as associações do MC.

As tabelas são compostas por **campos** que armazenam os dados destas tabelas. Ao conjunto dos dados, oriundos de

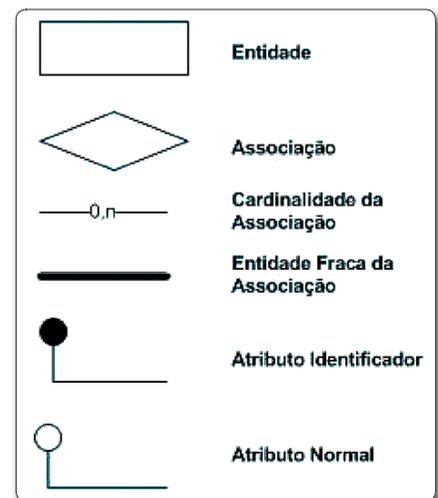


Figura 1. Símbolos Utilizados no DER

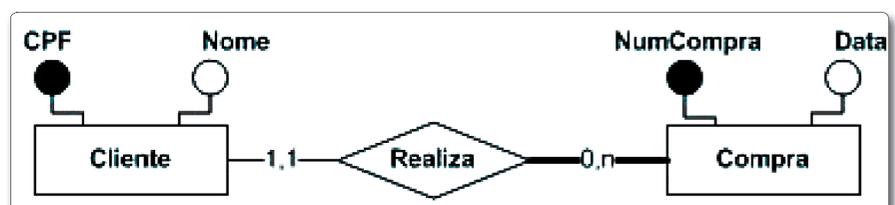


Figura 2. Exemplo de Diagrama ER

todos os campos de uma tabela, dá-se o nome de **registro**. A **chave primária** é constituída de um ou mais campos que identificam uma única ocorrência de registro. Uma **chave estrangeira** é um campo originado da necessidade da relação entre as tabelas. A chave estrangeira é a chave primária da tabela origem de uma determinada relação.

Após a construção do modelo conceitual, é feita a conversão para o MR. As entidades são convertidas em **tabelas**, os atributos em **campos** e os identificadores em **chaves-primárias**.

As chaves-estrangeiras são geradas a partir do identificador da entidade forte ao qual uma entidade fraca se relaciona. No caso do exemplo da **Figura 2**, o identificador de *Cliente* (CPF) que seria chave-primária da tabela *Cliente*, quando

da conversão para modelo lógico seria repassada à tabela *Compra* como chave-estrangeira.

Conforme representado na **Figura 3**, temos um modelo conceitual com duas entidades (*Acadêmico* e *Avaliação*) e uma associação com cardinalidade do tipo n:n entre elas. Neste tipo de relação (n:n), temos o que chamamos de **entidade associativa** gerada a partir da associação (como indica seu nome) entre as entidades. Quando não existir uma entidade forte, nesta relação ambas entidades irão ceder seus identificadores para a entidade associativa. Na conversão do modelo, a entidade associativa passa a ser uma tabela contendo uma chave primária composta por dois campos: os identificadores de cada entidade, que são "ID" para a entidade *Avaliação* e "Matrícula" para a

entidade *Acadêmico*. Nesta representação de modelo lógico, a parte superior da figura utilizada (retângulo) separa a chave primária dos demais campos (ver **Figura 3**). As letras "FK" significam *Foreign Key* (chave estrangeira).

Enfim, iremos partir para a construção dos modelos conceitual e lógico do sistema de orçamentos proposto neste artigo. Desta forma, acreditamos que a construção do modelo lógico será beneficiada, pois o esforço de análise e definição da estrutura do BD propriamente dito será despendido durante a criação do modelo conceitual. Entretanto, criar um modelo conceitual e convertê-lo em um modelo lógico ao contrário do que parece não é uma tarefa simples. Para auxiliar neste processo, iremos seguir uma proposta de metodologia de desenvolvimento para esta prática.



Nota do DevMan

Banco de Dados Orientado a Objetos

Definição

Um banco de dados orientado a objetos é um banco de dados em que cada informação é armazenada na forma de objetos. O gerenciador de banco de dados que considera a orientado a objeto é referenciado por vários como ODBMS ou OODBMS.

Existem dois fatores principais que levam à adoção da tecnologia de banco de dados orientados a objetos. A primeira, é que em um banco de dados relacional a manipulação de dados complexos é difícil. Segundo, os dados são geralmente manipulados pela aplicação escrita usando linguagens de programação orientada a objetos, como C++, C#, Java, e o código precisa ser traduzido entre a representação do dado e as linhas da tabela relacional, o que além de ser uma operação tediosa de ser escrita, consome tempo.

Recursos Técnicos

Num banco de dados orientado a objetos puro, os dados são armazenados como objetos onde só po-

dem ser manipulados pelos métodos definidos pela classe que estes objetos pertencem. Os objetos são organizados numa hierarquia de tipos e subtipos que recebem as características de seus supertipos. Os objetos podem conter referências para outros objetos, e as aplicações podem conseqüentemente acessar os dados requeridos usando um estilo de navegação de programação.

A maioria dos bancos de dados também oferece algum tipo de linguagem de consulta, permitindo que os objetos sejam localizados por uma programação declarativa mais próxima. Uma tentativa de padronização foi feita pela ODMG (Object Data Management Group) com a OQL (Object Query Language).

O acesso aos dados pode ser rápido porque as junções não são geralmente necessárias (como numa implementação tabular de uma base de dados relacional). Isto porque um objeto pode ser obtido diretamente sem busca, seguindo os ponteiros.

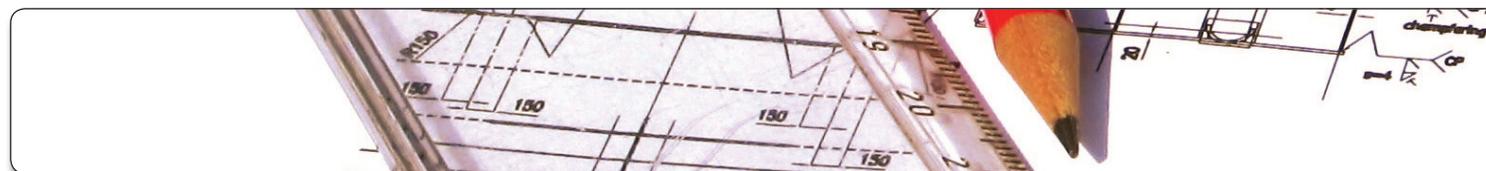
Desenvolvendo o Modelo de Dados para o Sistema de Orçamentos

A partir de agora iremos descrever as etapas para a construção de um modelo de dados para o nosso estudo de caso que será um sistema de orçamento. Ao longo deste artigo iremos conhecer melhor sobre este sistema.

Para o desenvolvimento de seu modelo dados, seguiremos uma metodologia composta por 7 etapas: levantamento de dados, definição de entidades, definição das associações, definição de atributos e identificadores, criação do modelo conceitual, a definição das cardinalidades e por fim a geração do modelo lógico. A **Figura 4** demonstra as etapas que serão detalhadas a seguir.

Etapa 1 – Levantamento dos Dados

O primeiro passo para a construção de qualquer modelo conceitual é a realização do levantamento dos dados. Esta etapa consiste em descrever textualmente as características e necessidades do sistema. É como se o responsável por este



levantamento escrevesse uma pequena história enfatizando os aspectos relacionados aos processos e necessidades de dados do sistema que pretende modelar. Além do texto, poderão ser adicionadas figuras, fluxogramas ou diagramas com a representação dos processos e dados, ou qualquer recurso que possa ilustrar o funcionamento do sistema, o que é uma prática muito comum em qualquer processo de análise de sistema. A descrição textual poderá ser armazenada e usada como documento do sistema, servindo inclusive como registro de autoria. Esta prática não é muito comum entre os desenvolvedores que usam SGBDs, visto que estes tendem a fazer este processo diretamente no SGBD criando diretamente o modelo lógico. Entretanto, uma análise limitada dos requisitos de dados do sistema poderá produzir a médio e longo prazo prejuízos à continuidade ou evolução do BD.

Na modelagem dos dados para o sistema de orçamentos iremos de considerar o seguinte cenário:

Cenário 1: O Sr. X, desenvolve mensalmente uma planilha com a previsão de suas contas de despesa e receita no intuito de melhor organizar sua gestão orçamentária. Para melhor compreender a necessidade do Sr. X, analisemos a **Figura 5**.

- Na **Figura 5**, constatamos as seguintes características presentes na planilha:

1 - Ela possui contas de Receitas e Despesas, que são subdivididas em despesas fixas e variáveis. As Receitas consistem em todos os valores que podem ser recebidos pelo Sr. X, as Despesas Fixas são aquelas que irão acontecer todo mês e com valor pré-determinado e as Despesas Variáveis não têm um valor fixo e os valores podem oscilar de um mês para outro;

2 - Apresenta duas colunas todo mês, uma de valor Previsto e outra de Realizado. O valor previsto consiste no orçamento atribuído ou previsão de valor

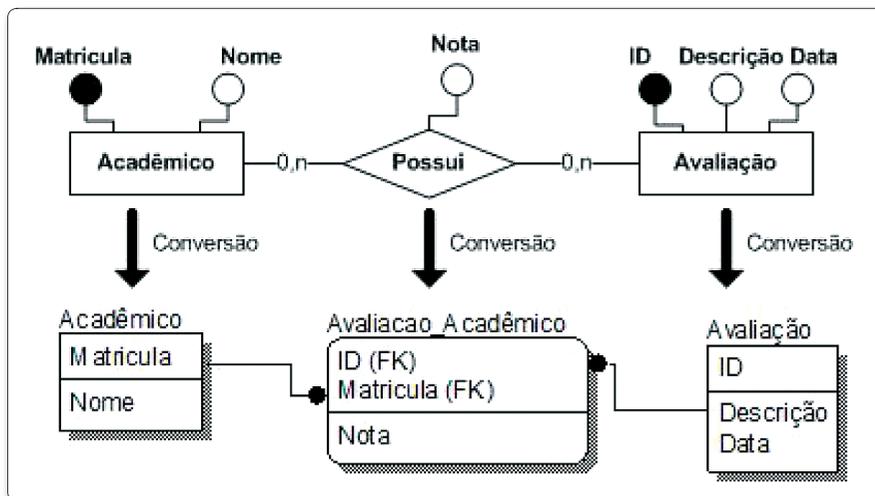


Figura 3. Conversão do Modelo Conceitual para o Lógico

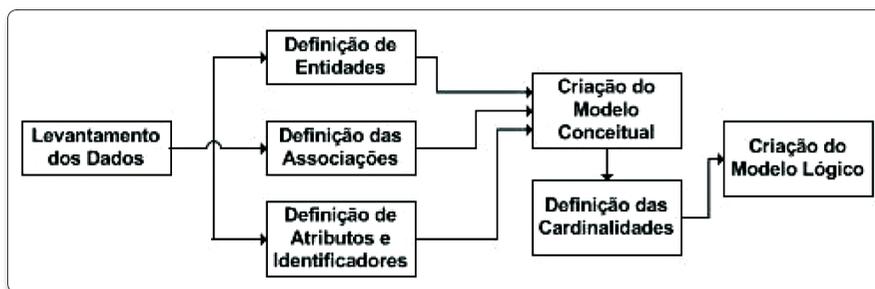


Figura 4. Etapas para construção do modelo lógico dos dados

para a conta, já o Realizado consiste no valor que efetivamente foi lançado (gasto ou recebido) em determinada conta;

3 - Os valores previstos e realizados são separados por mês. O valor previsto é lançado uma única vez no mês para cada conta orçada. Já o valor realizado pode receber mais de um lançamento (ver coluna de receita C6 – Outros recebimentos do mês set/2008 na **Figura 5**);

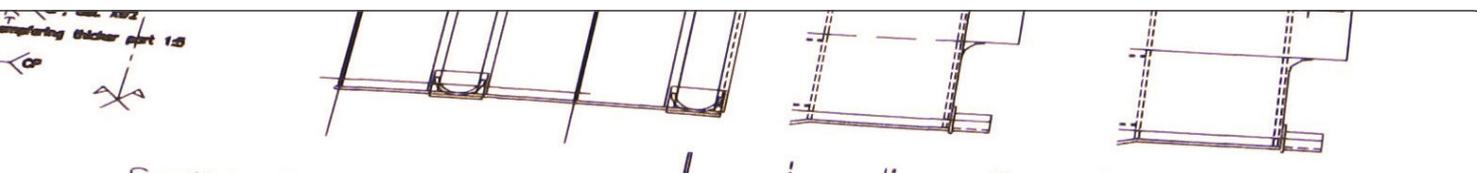
4 - A linha Saldo nada mais é do que um cálculo do que foi recebido (Receita) subtraindo o que foi pago (Despesa).

5 - Cada grupo de contas de Despesa e Receita receberá um subtotal.

Apesar de não estar explícito na planilha, é importante considerar ainda o que segue:

- 1 - Em todo mês poderão surgir novas contas, devendo portanto prever a possibilidade de inserção destas novas contas;
- 2 - Eventualmente contas não orçadas irão acontecer, ou seja, poderão receber lançamentos mesmo sem previsão.
- 3 - Contas orçadas podem também não receber lançamentos em determinado mês;
- 4 - Mesmo que as contas de Receita não apresentem divisões, não significa que estas não os terão futuramente.

Apresentadas as informações pertinentes ao sistema que se propõe desenvolver, seguiremos para a próxima etapa que consiste na definição das entidades aplicadas ao modelo.



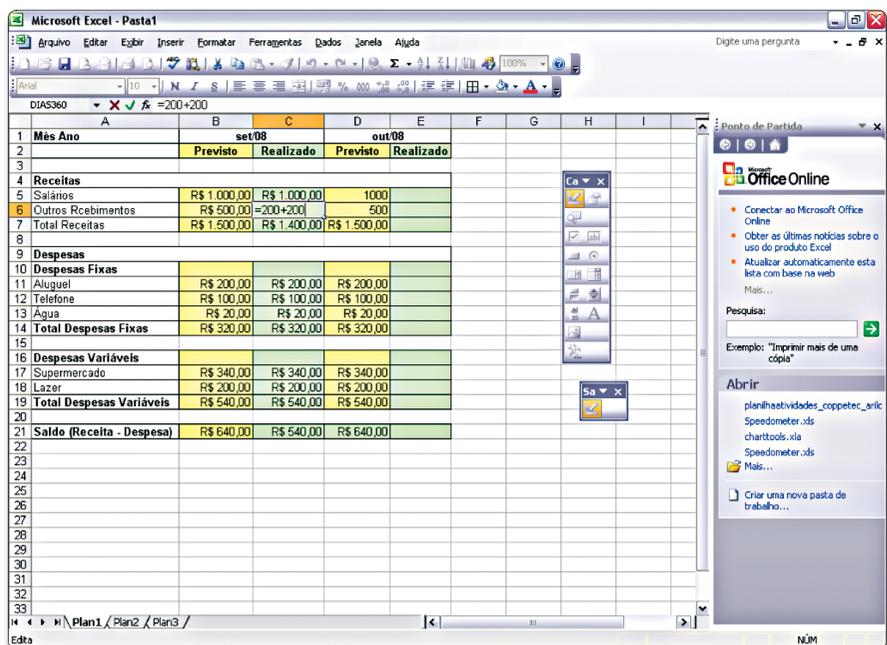


Figura 5. Planilha de Orçamento Mensal

Etapa 2 – Definição das Entidades

O segundo passo para criação do modelo conceitual é a definição das entidades, que é dependente da etapa levantamento de dados. Esse passo consiste basicamente em identificar os recipientes de dados, ou seja, as tabelas que irão compor nosso modelo. Analisando o levantamento de dados realizado, podemos perceber que:

Existe um grupo de contas principal que são as contas de Receitas e Despesas. Estas contas é que irão permitir a separação dos saldos referentes ao que o Sr. X tem a receber e pagar. Torna-se necessário, portanto, criar uma separação lógica destes grupos de contas. A esta separação daremos o nome de **Grupo**;

A subdivisão de contas é ressaltada nas contas de despesas, que é subdividida em Fixas e Variáveis. Da mesma forma, estas subdivisões poderão acontecer também em contas de receita. Para possibilitarmos esta subdivisão, iremos criar uma entidade chamada **Subgrupo**;

Os valores que serão lançados no sistema dividem-se em previstos e realizados.

O valor previsto é lançado uma única vez no mês, já o valor realizado pode receber vários lançados no decorrer deste. Desta forma, torna-se necessário separar o valor previsto que será tratado como uma entidade chamada **Orçamento** e o valor realizado, cuja entidade irá se chamar **Realizado**;

Os valores precisam estar alocados em contas que deverão estar vinculadas aos respectivos grupos e subgrupos de origem (despesa e receita). Os lançamentos oriundos dos valores previstos ou realizados não podem ser armazenados no grupo ou subgrupo em virtude da necessidade de detalhamento. Para possibilitar este controle, é necessário criar uma entidade chamada **Contas** que irá receber o cadastro das possíveis contas usadas para os lançamentos previstos ou realizados. O plano de contas do Sr. X será a representação (relatório) das contas vinculadas, cada qual vinculada a um respectivo subgrupo e grupo de origem, não sendo portanto necessária a criação de uma entidade para tal fim.

É necessário enfatizar que a construção de entidades leva em consideração a necessidade do armazenamento de dados. A consolidação de informações numéricas (totais e subtotais) não motiva a criação de entidades.

Entretanto, durante o desenvolvimento da interface do sistema ou em nível de aplicação, poderão ser criadas entidades não associadas ao modelo com o intuito de obter-se um melhor desempenho no banco de dados, o que chamamos de operações de *tunning* (ajustes). Estes aspectos não serão tratados neste artigo.

Vejam os seguintes resumos das entidades encontradas:

GRUPO: cadastro dos grupos de Receita e Despesa;

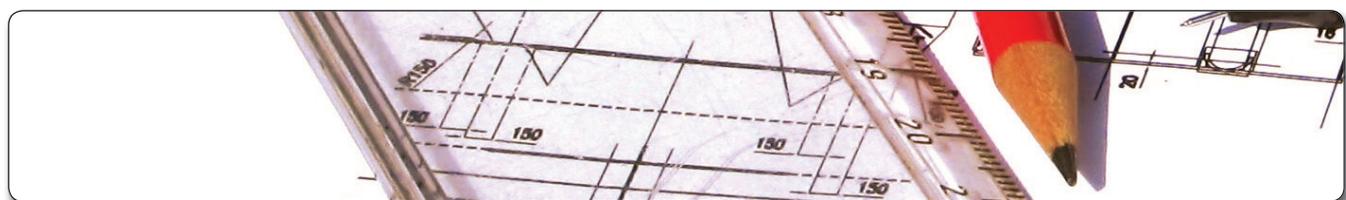
SUBGRUPO: onde serão criadas as divisões lógicas dos Grupos.

CONTAS: é onde serão vinculados os valores de lançamento orçados e realizados. É vinculada ao Subgrupo. O plano de contas é a representação das contas separadas nos respectivos subgrupos e grupos;

ORÇAMENTO: com base no período (Mês/Ano), permite a previsão de valores (D ou C) para cada Conta cadastrada.

REALIZADO: com base no cadastro das Contas, gera o movimento ou histórico dos lançamentos na data (Dia/Mês/Ano) em que ocorrem.

Podemos perceber que as entidades resultantes são oriundas das principais características apresentadas. As subdivisões ou características específicas tratadas no levantamento de dados devem ser absorvidas pelas entidades. Para isso, não existe uma fórmula mágica a não ser a leitura, releitura, análise e interpretação do que foi observado durante o levantamento dos dados. Quanto maior for o detalhamento encontrado, mais fácil será a conclusão desta etapa. Desta forma, qualquer recurso gráfico



Entidade 1	Entidade 2	Possui relação direta?	Justificativa
Grupo	Subgrupo	Sim	O Grupo utiliza-se do Subgrupo para criar as subdivisões de seu conteúdo
Grupo	Contas	Não	Relação indireta, pois não existe uma relação direta de Grupo com Contas pelo fato de Subgrupo ser a subdivisão utilizada por grupo. Se criarmos uma relação direta entre Grupos e Contas, o Subgrupo perderá seu sentido.
Grupo	Orçamento	Não	Relação indireta, pois Orçamento é gerado a partir das Contas, que não são vinculadas diretamente com Grupo.
Grupo	Realizado	Não	Relação indireta, pois Orçamento é gerado a partir das Contas que não são vinculadas diretamente com Grupo.
Subgrupo	Contas	Sim	Relação direta, pois Contas consistem no detalhamento de Subgrupo.
Subgrupo	Orçamento	Não	Relação indireta, pois Subgrupo está vinculado diretamente com a entidade Contas, e o Orçamento mensal utiliza-se da relação com Contas para ser gerado.
Subgrupo	Realizado	Não	Relação indireta, pois Subgrupo está vinculado diretamente com a entidade Contas. A entidade Realizado utiliza-se da relação com Contas para ser gerada.
Contas	Orçamento	Sim	Relação é Direta, pois Orçamento é gerado a partir de Contas.
Contas	Realizado	Sim	Relação Direta, pois a entidade Realizado é gerada a partir de Contas.
Orçamento	Realizado	Não	Relação indireta, pois a entidade Realizado é gerada apenas a partir de Contas. Se definíssemos que Realizado fosse originada a partir do Orçamento, ou seja, com relação direta, todos os lançamentos gerados na coluna Previsto da nossa planilha, teriam de ter um lançamento de origem na entidade Orçamento.

Tabela 1. Associações das Entidades do Sistema de Orçamentos

ou detalhamento textual é sempre bem vindo.

Etapa 3 – Definição das Associações

A etapa de Definição das Associações consiste em fazer uma análise das possíveis associações existentes entre as entidades elencadas na etapa anterior. Ela deve ser realizada combinando uma entidade com as demais – cada entidade verifica a possibilidade de associação direta com as demais. É a etapa mais crítica, pois é o momento onde é mais exigida uma análise ampla do contexto do problema apresentado na etapa de levantamento dos dados.

Para verificar as associações, deve ser feita uma combinação das entidades não importando a ordem de verificação, ou seja, **Entidade A** combinada com **Entidade B** é o mesmo que **Entidade B** combinada com a **Entidade A**.

A associação só será gerada se a relação entre as entidades for **DIRETA**. Para verificar se existe uma relação direta, ao combinarmos duas entidades verificamos se estas de alguma maneira dependem dos atributos da outra para representar as suas informações.

Considerando o sistema de orçamentos, as entidades elencadas na etapa anterior e o levantamento de dados, verificamos na Tabela 1 as associações entre as entidades identificadas na etapa 2.

Percebemos nesta etapa a importância

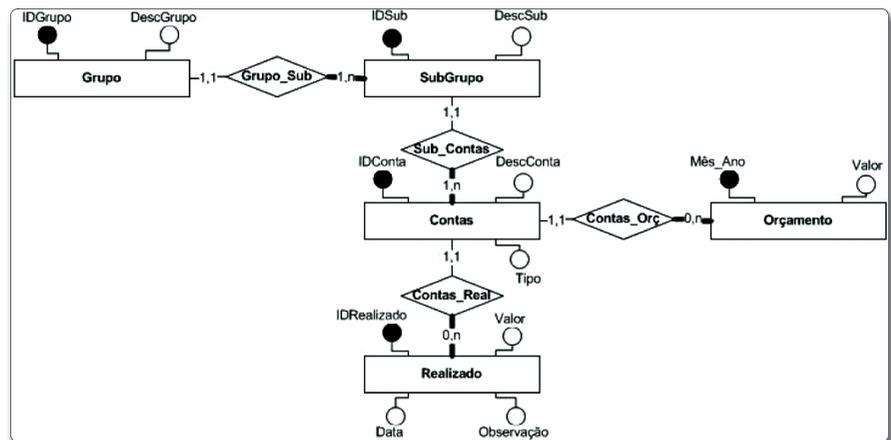


Figura 6. Modelo Conceitual do Sistema de Orçamentos

e necessidade de utilizarmos as informações obtidas nas etapas anteriores para a geração das associações. Na etapa a seguir, iremos definir os atributos e identificadores das entidades.

Etapa 4 – Definição de Atributos e Identificadores

É aqui que devemos definir as propriedades das entidades do modelo conceitual. Nesse momento, devemos definir os atributos de cada entidade e o identificador que será responsável pelo vínculo do relacionamento gerado pela associação. Os atributos identificadores aparecem em destaque na representação das entidades. Um atributo nada mais é do que uma característica ou propriedade pertinente a entidade que necessita ser armazenado. O número

de atributos pode ser elevado quando da necessidade de representar as mais diversas características das entidades. No caso do sistema de orçamentos, não encontramos esta situação.

Uma dica importante é que os atributos e identificadores das entidades devem ser analisados individualmente, contemplando o nível de detalhamento exigido por cada entidade. Vejamos os atributos encontrados nas entidades do sistema de orçamentos. Em negrito estão destacados os identificadores de cada entidade:

- GRUPO (**IDGrupo**, DescGrupo)
- SUBGRUPO (**IDSub**, DescSub)
- CONTAS (**IDConta**, DescConta, Tipo)
- ORÇAMENTO (**Mês_Ano**, Valor)
- REALIZADO (**IDRealizado**, Data, Valor, Observação)

Associação	Cardinalidade	Considerações
Grupo e Subgrupo	Mínima: 1 Máxima: n	Uma ocorrência de Grupo deve obrigatoriamente ter no mínimo uma ocorrência em Subgrupo e no máximo quantas necessitar (n ocorrências). Desta forma, sempre que uma ocorrência de Grupo for gerada, uma de Subgrupo precisará também ser criada e vinculada. Já a máxima indica que podem ser criados quantos Subgrupos forem necessários para um determinado Grupo.
Subgrupo e Grupo	Mínima: 1 Máxima: 1	Uma ocorrência de Subgrupo poderá estar vinculada a no mínimo e no máximo uma ocorrência de Grupo. Isto significa dizer que a relação de Subgrupo com Grupo é obrigatória (cardinalidade mínima: 1), sendo que cada Subgrupo poderá ter no máximo um Grupo vinculado. Subgrupo é a entidade fraca da relação, pois depende necessariamente do identificador de Grupo em todas as ocorrências de registro geradas.
Subgrupo e Contas	Mínima: 1 Máxima: n	Indica que para um Subgrupo existir ele precisa ter obrigatoriamente uma ocorrência de registro na tabela Contas (cardinalidade mínima: 1) e no máximo quantas ocorrências de Contas precisar (cardinalidade máxima: n).
Contas e Subgrupo	Mínima: 1 Máxima: 1	Indica que cada ocorrência de conta deverá estar vinculada obrigatoriamente a no mínimo uma ocorrência de Subgrupo. É a entidade fraca da relação. Nenhum registro de Conta poderá ser gerado sem estar vinculado a um e somente um Subgrupo (cardinalidade mínima e máxima = 1).
Contas e Orçamento	Mínima: 0 Máxima: n	Indica uma relação opcional entre Contas e Orçamento (cardinalidade mínima: 0), onde não obrigatoriamente uma conta cadastrada precisa ter ocorrências relacionadas na entidade Orçamento. No entanto, poderá ter várias ocorrências de registros relacionadas (cardinalidade máxima: n). A relação opcional foi usada neste caso para flexibilizar o lançamento ou não de orçamento, caso alguma conta não prevista durante o mês de competência for gerada.
Orçamento e Contas	Mínima: 1 Máxima: 1	Indica uma relação obrigatória entre Orçamento e Contas, onde obrigatoriamente uma ocorrência de registro de Orçamento precisa ter uma e somente uma ocorrência relacionada na entidade Contas (cardinalidade mínima e máxima = 1). É a entidade fraca da relação por depender do identificador de Contas para gerar suas ocorrências de registros.
Contas e Realizado	Mínima: 0 Máxima: n	Indica uma relação opcional entre Contas e Realizado (cardinalidade mínima: 0), onde não obrigatoriamente uma Conta precisa ter ocorrências relacionadas na entidade Realizado. No entanto, poderá ter várias ocorrências de registros relacionadas (cardinalidade máxima: n). A relação opcional foi usada neste caso para flexibilizar o lançamento ou não de valor pago ou recebido (Realizado) caso alguma conta não tenha movimentação durante o mês de competência da análise.
Realizado e Contas	Mínima: 1 Máxima: 1	Indica uma relação obrigatória entre Realizado e Contas, onde obrigatoriamente um registro de Realizado precisa ter uma e somente uma ocorrência relacionada na entidade Contas (cardinalidade mínima e máxima = 1). É a entidade fraca da relação por depender do identificador de Contas para gerar suas ocorrências de registros.

Tabela 2. Cardinalidades das Entidades do Sistema de Orçamentos

É importante ressaltar que os atributos que serão apresentados devem ser somente os da entidade analisada. Os atributos de relação serão subentendidos na apresentação do modelo conceitual e apresentados somente no modelo lógico que será gerado posteriormente.

Perceba que a entidade CONTAS possui um atributo chamado **Tipo**. Este atributo permite que através da conta seja atribuído os valores “D” ou “C” automaticamente ao registro. Desta forma, não há a necessidade de informar o tipo quando da geração do lançamento, o que de certa forma agiliza o processo de lançamento de uma receita ou despesa.

Podemos considerar esta como sendo uma das etapas mais simples do desenvolvimento do modelo. Tendo cumprido a etapa 4, teremos todas as informações necessárias para desenvolver o modelo conceitual, como será visto na etapa a seguir.

Etapa 5 – Criação do Modelo Conceitual

Com base no levantamento, nós geramos até o momento o descritivo das

entidades, das associações e dos atributos/identificadores. A partir de agora já podemos pensar na representação dessas informações a partir de um modelo conceitual. Para isso, utilizaremos um DER que nos permite uma visão gráfica das entidades e suas relações.

O modelo descrito na **Figura 6** é apresentado juntamente com as cardinalidades, que serão alvo de análise na próxima etapa.

Conforme definido nas etapas anteriores, a entidade *Grupo* está associada à entidade *Subgrupo*, onde a entidade *Subgrupo* é a entidade fraca da relação por depender do identificador de *Grupo* no momento da geração de qualquer ocorrência de registro.

Continuando, a entidade *Subgrupo* está vinculada à entidade *Contas*, que consiste no detalhamento das contas que serão usadas no controle do sistema de orçamentos e é a entidade fraca da relação.

Em seguida, a entidade *Contas* está vinculada às entidades *Orçamento* e *Realizado*. As entidades *Orçamento* e *Realizado* serão as entidades fracas da relação e receberão o identificador de *Contas* em cada

ocorrência de registro. A entidade *Orçamento* será utilizada para armazenar a previsão de gasto mensal de cada conta e a entidade *Realizado* irá armazenar um ou mais lançamentos que possibilitarão informar o que efetivamente foi movimento em cada conta.

As cardinalidades mínimas e máximas estão apresentadas no modelo conceitual da **Figura 6**, apesar deste ser o tópico da etapa seguinte.

Etapa 6 – Definindo as Cardinalidades

Para gerar a cardinalidade, devemos fazer a seguinte pergunta: quantas ocorrências no mínimo e no máximo poderão existir entre a entidade A e entidade B em uma associação? A representação da cardinalidade encontrada é repassada então para o lado em que se encontra a entidade B. A **Tabela 2** apresenta a análise da cardinalidade das associações presentes no modelo conceitual do sistema de orçamento que estamos desenvolvendo.

Uma dica interessante para definir as cardinalidades é identificar quais são as entidades de cadastro e movimento.

As entidades de cadastro servem para padronizar a entrada de registros e definir o nível de detalhamento que vai ser usado. As entidades de movimento armazenam os fatos que ocorrem no sistema, vinculando estes às entidades de cadastro. A entidade de cadastro é a entidade forte de qualquer relação. No sistema de orçamentos, as entidades de cadastro seriam *Grupos*, *Subgrupos* e *Contas*, e as de movimento seriam *Orçamento* e *Realizado*.

Neste sentido, perceba na **Tabela 2** que sempre que temos uma relação entre entidade de movimento com uma de cadastro a cardinalidade mínima e máxima será 1. Como exemplo, podemos citar a relação entre *Orçamento* e *Contas* ou entre *Realizado* e *Contas*.

Quando a verificação é feita entre as entidades de cadastro, devemos perceber a hierarquia de representação, ou seja, qual entidade de cadastro depende da outra. No caso da relação entre *Subgrupo* e *Grupo*, a entidade *Subgrupo* depende de *Grupo* para ser gerada, e desta forma recebe a cardinalidade mínima e máxima como sendo 1. O mesmo acontece na relação entre *Contas* e *Subgrupo*, onde a *Conta* depende de um *Subgrupo* para existir.

Identificadas as cardinalidades, o modelo conceitual está pronto, e agora já podemos criar o modelo lógico. Na etapa 7 faremos as considerações sobre o processo de conversão ou criação do modelo lógico de dados.

Etapa 7 – Criação do Modelo Lógico de Dados

Também conhecido como estrutura **data-lógica**, o modelo lógico de dados, neste caso relacional, nos permite visualizar a estrutura das tabelas, campos e associações existentes. O processo de conversão de um modelo conceitual para lógico se dá através dos seguintes passos:

Tabelas: São oriundas das entidades geradas pelo modelo conceitual. Podem também ser definidas após a verificação da existência de **entidades associativas**, que podem ser encontradas no modelo conceitual identificadas pela relação N:N. No exemplo tratado, não existem ocorrências deste tipo.

Campos: transcrição dos campos

Entidade e Atributos (Modelo Conceitual)	Tabela e campos (Modelo Lógico)
Grupo • IDGrupo (identificador) • DescGrupo	Grupo • IDGrupo (chave primária) • DescGrupo
Subgrupo • IDSub (identificador) • DescSub	Subgrupo • IDSub (chave primária) • DescSub • IDGrupo (chave Estrangeira)
Contas • IDConta (identificador) • DescConta	Contas • IDConta (chave primária) • DescConta • IDSub (chave estrangeira)
Orçamento • Mês_ano (identificador) • Valor	Orçamento • Mês_ano (chave primária) • Valor • IDConta (chave primária e estrangeira)
Realizado • IDRealizado (identificador) • Data • Valor • Observação	Realizado • IDRealizado (chave primária) • Data • Valor • Observação • IDConta (chave estrangeira)

Tabela 3. Conversão das Entidades do Modelo conceitual para Tabelas

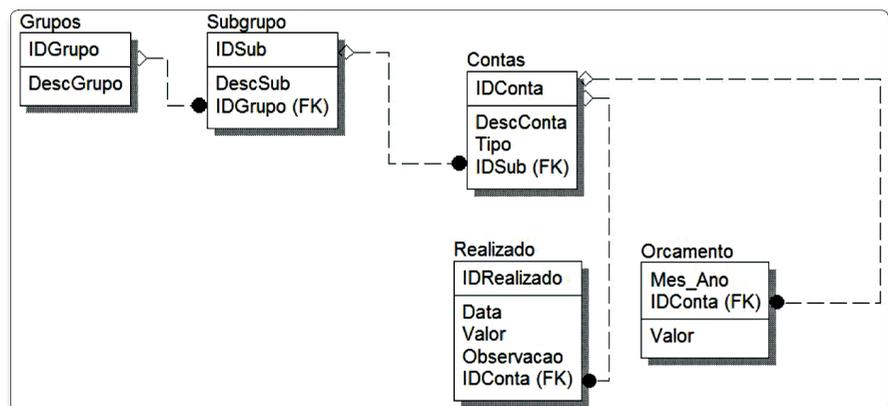


Figura 7. Modelo lógico do Sistema de Orçamentos

apresentados pelas entidades no modelo conceitual, sendo o identificador da tabela caracterizado como **chave-primária**.

Associações: verificando a entidade fraca através do modelo conceitual, pode-se gerar a associação entre as entidades a partir do identificador da entidade forte. Desta forma, no modelo lógico o identificador da entidade forte passa a ser uma **chave-estrangeira** do modelo relacional.

Desta forma, foram geradas as seguintes tabelas e atributos, descritas na **Tabela 3**.

Assim sendo, será possível a criação do modelo lógico. Perceba que no

processo de conversão os atributos antes omitidos pelas associações são representados na tabela.

No caso da entidade *Orçamento*, convém identificar o atributo *IDConta* da entidade *Contas* também como chave primária no modelo lógico, no intuito de não permitir o cadastro em duplicidade. Ao tentar inserir o *Orçamento* de uma mesma *Conta* no mês, o próprio SGBD fará o bloqueio de gravação deste registro.

Já na tabela *Realizado* o *IDConta* da entidade *Contas* é apenas chave estrangeira, tendo como base o que foi mencionado no levantamento de dados quanto à possibilidade de uma determinada *Conta* possuir

vários lançamentos ou valores em um mesmo mês. Vejamos na **Figura 7** como o modelo lógico ficará representado.

O modelo lógico apresenta as definições contidas no modelo conceitual. Podemos perceber que as tabelas recebem o identificador da entidade forte como chave-estrangeira (FK ou *Foreign Key*). A única exceção é a chave-estrangeira *IDConta* na tabela *Orçamento* que também é definida como primária na entidade, conforme já relatado anteriormente.

Conclusões

Construir um modelo conceitual nem sempre é uma tarefa atrativa para um desenvolvedor. Entretanto, ao aplicar a metodologia sugerida quando do desenvolvimento de qualquer sistema, obtemos uma análise mais aprofundada das características do BD que irá armazenar as informações pertinentes ao propósito do mesmo.

O processo vivenciado durante modelagem conceitual gera uma maior maturidade com relação aos propósitos do sistema, permitindo inclusive que sejam discutidas características que durante a criação apenas do modelo lógico poderiam não ser observadas. A análise das associações e das cardinalidades são exemplos claros da necessidade de atenção que a modelagem conceitual exige e se reverte em benefícios para o desenvolvedor durante a criação do modelo lógico. Este último terá facilitado o processo de desenvolvimento do sistema, tanto pela documentação quanto pela facilidade de desenvolvimento do processo efetivo de modelagem de dados.

A modelagem conceitual não necessita de ferramentas específicas. Pode ser gerada inclusive no rascunho usado para anotações do sistema. Porém, armazenar a documentação é bastante interessante no desenvolvimento de qualquer sistema, até mesmo para comprovar a autoria ou os direitos sobre o mesmo. ODER e os demais diagramas apresentados neste artigo foram desenvolvidos usando a ferramenta MS-Microsoft Visio 2003. Já o modelo lógico resultante foi desenvolvido usando a ferramenta CA-ERWin Versão 4.0.

A conversão do modelo conceitual para o lógico é muito natural, pois as estruturas que utilizam são muito parecidas. Entretanto, a modelagem conceitual tem a vantagem de manter o foco no contexto do problema enquanto que o modelo lógico tende a se preocupar mais com a estrutura de armazenamento dos dados.

Ao analisar o modelo lógico gerado para o sistema de orçamentos, podemos perceber que este atende aos requisitos especificados pelo levantamento de dados. Ele permitirá que sejam criados novos *Grupos*, *Subgrupos* e *Contas* sem prejuízos àquelas já existentes, da mesma forma que valores não orçados poderão compor os valores de contas

realizadas no mês. Muitos relatórios poderão ser gerados tais como: plano de contas (combinação de *Grupos*, *Subgrupos* e *Contas*), *Orçamento* e *Realizado* do mês ou período, em separado ou em conjunto, com os respectivos totais e subtotais, atendendo plenamente as características do cenário apresentado. ●

Dê seu feedback sobre esta edição!

A SQL Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/sqlmagazine/feedback



PENSE...

QUANTO TEMPO
VOCÊ GASTARIA
PARA DESENVOLVER
COBRANÇA COM BOLETOS
BANCÁRIOS PARA
APENAS UM BANCO
NO SEU SOFTWARE

COBREBEMX

-  56 BANCOS E MAIS DE 430 CARTEIRAS DE COBRANÇA PARA IMPRESSÃO E/OU ENVIO DE BOLETO BANCÁRIO POR EMAIL;
-  GERAÇÃO DE BOLETOS ON LINE;
-  GERAÇÃO E LEITURA DE ARQUIVOS (REMESSA/RETORNO) NOS PADRÕES FEBRABAN E CNAB;
-  MAIS DE 40 EXEMPLOS EM DIVERSAS LINGUAGENS DE PROGRAMAÇÃO

DOWNLOADS E INFORMAÇÕES EM WWW.COBREBEM.COM

