

Banco de Dados

Uma das definições de banco de dados afirma que se trata de uma coleção de informações que se relacionam de modo que criem algum sentido, isto é, é uma estrutura bem organizada de dados que permite a extração de informações. Assim, são muito importantes para empresas e tornaram-se a principal peça dos sistemas de informação.

Além dos dados, um banco de dados também é formado pelos metadados. Um metadado é todo dado relativo a outro dado, sem o qual não seria possível organizar e retirar as informações de um banco de dados.

Alguns afirmam que a expressão é sinônimo de SGBD (Sistema Gerenciador de Banco de Dados), que é um programa de gerenciamento de dados. O termo “banco de dados” também é usado para definir uma base de dados, que é um grupo de dados agrupados por um SGBD.

O SGBD usa uma linguagem para criar a base de dados, sendo que, atualmente, a mais usada é a SQL (Structured Query Language). São vários os SGBDs disponíveis no mercado; alguns são pagos e outros gratuitos.

Alguns dos tipos de SGBD existentes no mercado:

- **SQLServer:** Um dos maiores do mundo, sob licença da Microsoft;
- **MySQL:** Trata-se de um software livre, com código fonte aberto;
- **FirebirdSQL:** Possui código fonte aberto e roda na maioria dos sistemas Unix;
- **Microsoft Access:** É um Sistema Gerenciador de Banco de Dados que acompanha o pacote Office da Microsoft. Este SGBD tem poucas atribuições profissionais, sendo mais usado para aprendizagem, devido à sua interface amigável;
- **mSQL:** Sistema pequeno e que trabalha mais com o uso eficiente da memória. Foi criado pela Hughes Technologies Pty Ltd.

No armazenamento de um dado, é necessário criar tabelas, dentro das quais são criadas colunas, onde serão guardadas as informações. Para que os dados presentes na base de dados fiquem bem organizados, as tabelas devem ser criadas para que não misturem as informações.

Tipos de Bancos de Dados

Os bancos de dados atendem a praticamente qualquer tipo de aplicação. É por esse motivo que não existe só um tipo de banco de dados. O padrão mais conhecido é chamado de banco de dados relacional.

Esse tipo é muito utilizado em sistemas de ERP (gestão de empresas), CRM (relacionamento com o cliente), controle financeiro e tantas outras aplicações porque os dados são estruturados em tabelas cujas colunas e linhas se relacionam. Uma loja, por exemplo, pode ter uma tabela de clientes, outra para controle de estoque, uma terceira para fluxo de caixa e assim por diante.

Via de regra, bancos de dados relacionais são baseados na linguagem SQL. Mas a gente também vai encontrar bancos de dados não relacionais que, como tal, são frequentemente chamados de noSQL. Eles possuem estruturas que diferem do SQL e, portanto, são direcionados a aplicações que não podem ter os dados completamente organizados em tabelas, como serviços que armazenam imagens na web, por exemplo.

Não pense que termina aí. Há outras soluções no mercado, inclusive dotadas de muita tecnologia. É o caso da Oracle, que lançou, no ano passado, o primeiro banco de dados autônomo nas nuvens. Essa solução é capaz de ajustar, reparar e atualizar o banco de dados de modo automatizado, dispensando interferência humana.

Abstração de Dados

O sistema de banco de dados deve garantir uma visão totalmente abstrata do banco de dados para o usuário, ou seja, para o usuário do banco de dados pouco importa qual unidade de armazenamento está sendo usada para guardar seus dados, contanto que os mesmos estejam disponíveis no momento necessário.

Esta abstração se dá em três níveis (Figura 2):

- Nível de visão do usuário: as partes do banco de dados que o usuário tem acesso de acordo com a necessidade individual de cada usuário ou grupo de usuários;
- Nível conceitual: define quais os dados que estão armazenados e qual o relacionamento entre eles;
- Nível físico: é o nível mais baixo de abstração, em que define efetivamente de que maneira os dados estão armazenados.

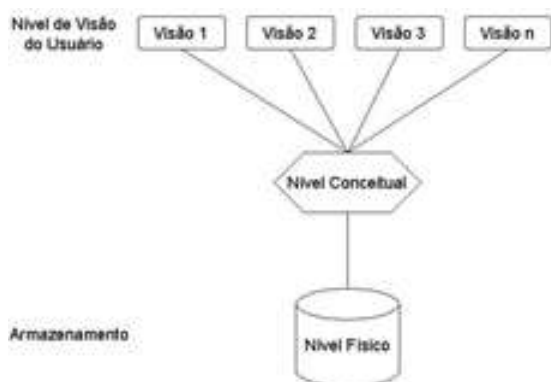


Figura 2. Níveis de abstração

Projeto de Banco de Dados

Todo bom sistema de banco de dados deve apresentar um projeto, que visa a organização das informações e utilização de técnicas para que o futuro sistema obtenha boa performance e também facilite infinitamente as manutenções que venham a acontecer.

O projeto de banco de dados se dá em duas fases:

- Modelagem conceitual;
- Projeto lógico.

Estas duas etapas se referem a um sistema de banco de dados ainda não implementado, ou seja, que ainda não exista, um novo projeto.

Para os casos em que o banco de dados já exista, mas é um sistema legado, por exemplo, ou um sistema muito antigo sem documentação, o processo de projeto de banco de dados se dará através da utilização de uma técnica chamada de Engenharia Reversa, que será visto em outra oportunidade.

Funcionamento

O Gerenciamento e a Aplicação do Banco de Dados

A operação desses bancos se dá por meio de Sistemas Gerenciadores de Bancos de Dados (SGBD), que nada mais são do que conjuntos de programas de computador, responsáveis pelo gerenciamento dessas informações. Pode parecer complexo agora, porém, mais adiante, explicaremos melhor como tudo isso funciona e quais são os gerenciadores mais famosos do mercado.

Apesar da complexidade técnica envolvida, a aplicação prática é muito simples e está presente em nosso cotidiano. Por exemplo, quando acessamos uma loja virtual e compramos um produto, por mais simples que ela seja, nós navegamos por centenas de milhares de dados e, graças a um gerenciamento correto das informações, recebemos o que compramos em nossas casas.

Isso ocorre porque nosso cadastro - incluindo nome, endereço e dados de pagamento - e as informações sobre o produto estão num banco de dados. Tudo o que fazemos depende de uma central de informações para funcionar adequadamente.

As Vantagens de um SGBD

Maior disponibilidade: Uma das principais vantagens de um SGBD é que a mesma informação pode ser disponibilizada a utilizadores diferentes, ou seja, compartilhamento de dados.

Redundância minimizada: Os dados de um SGBD são mais concisos, porque, como regra geral, a informação nela aparece apenas uma vez. Isto reduz a redundância de dados, ou em outras palavras, a necessidade de repetir os mesmos dados uma e outra vez. Minimizando a redundância pode, portanto, reduzir significativamente o custo de armazenamento de informações em discos rígidos e outros dispositivos de armazenamento.

Precisão: dados precisos, consistentes são um sinal de integridade dos dados. SGBDs fomentam a integridade dos dados, porque as atualizações e alterações dos dados só tem que ser feitas em um só lugar. As chances de se cometer um erro são maiores se você é obrigado a alterar os mesmos dados em vários lugares diferentes do que se você só tem que fazer a mudança em um só lugar.

Programa e arquivo de consistência: Usando um sistema de gerenciamento de banco de dados, formatos de tabelas e programas do sistema são padronizados. Isso faz com que os tabelas de dados sejam mais fáceis de manter, porque as mesmas regras e diretrizes se aplicam a todos os tipos de dados. O nível de consistência entre os tabelas e programas também torna mais fácil de gerenciar dados quando vários programadores estão envolvidos.

User-friendly: Os dados são é mais fáceis de acessar e manipular com um SGBD do que sem ele. Na maioria dos casos, SGBDs também reduzem a dependência de usuários individuais à especialistas em computação para atender às necessidades de seus dados.

Maior segurança: Como afirmado anteriormente, SGBDs permitem que múltiplos usuários acessem os recursos dos mesmos dados. Esta capacidade é geralmente vista como um benefício, mas há riscos potenciais para a organização. Algumas fontes de informação devem ser protegidas ou garantida e vista apenas por indivíduos selecionados. Através do uso de senhas, sistemas de gerenciamento de banco de dados podem ser usado para restringir o acesso aos dados a apenas aqueles que devem vê-lo.

Outros: Tempo de desenvolvimento de aplicações é reduzido, Maior flexibilidade para realizar alterações (independência de dados) e Maior economia, informações atualizadas, menor volume de papel.

Barra de Ferramentas Padrão



A barra de ferramentas padrão do QlikView contém botões para as funções usadas com mais frequência. Para ativar ou desativar a barra de ferramentas padrão, escolha Barra de Ferramentas Padrão em Barras de Ferramentas no menu Exibir. A figura acima e o texto abaixo referem-se ao conteúdo padrão da barra de ferramentas padrão.

Novo Arquivo	Abre uma nova janela do QlikView e permite criar um novo arquivo QlikView. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+N.
Abrir Arquivo	Permite a abertura de um arquivo QlikView ou arquivo de tabela em uma nova janela do QlikView. Abrir um arquivo de tabela inicia automaticamente o assistente de arquivo. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+O.
Atualizar	Este comando só fica disponível com documentos abertos no QlikView Server e quando há uma versão mais nova do documento disponível no servidor.

	Quando faz uma atualização, você obtém acesso aos últimos dados gravados enquanto mantém a sessão, incluindo seleções e layout.
Salvar	Salva o documento ativo como um arquivo. O formato padrão de arquivo é definido em Preferências do Usuário.
Imprimir...	Ao clicar na ferramenta imprimir, será efetuada imediatamente a impressão do objeto de pasta selecionado de acordo com as configurações padrão da impressora, ignorando a página de propriedades gerais de impressão. A ferramenta ficará inativa se não houver nenhum objeto imprimível selecionado. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+P.
Imprimir como PDF	Abre a caixa de diálogo Imprimir com a impressora PDF-XChange 3.0 pré-selecionada. Após clicar no botão Imprimir, você deverá digitar um nome para o arquivo PDF gerado. Esse botão só estará disponível se houver uma impressora PDF no sistema.
Editar Script	Abre a caixa de diálogo Editar Script, permitindo gravar e executar scripts que abrem bancos de dados e importam dados para o QlikView. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+E.
Executar Script	Executa novamente o script atual, atualizando a base de dados associativa do QlikView para incluir as alterações feitas nos dados de origem desde a execução anterior. A hora da última recarga é mostrada como um carimbo de data/hora na barra de status.
Desfazer Alteração de Layout	Desfaz a última alteração de layout, inclusive a movimentação, o ajuste e a exclusão de objetos de pasta, assim como as alterações das propriedades de objetos de pasta. A exclusão de pastas e a edição das propriedades da pasta ou do documento também podem ser desfeitas. O QlikView mantém uma lista das últimas alterações de layout. Cada comando Desfazer Alteração de Layout o levará um passo atrás na lista. Algumas operações, por exemplo, Executar Script e Reduzir Dados, esvaziarão o buffer de Desfazer/Refazer. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+Z.
Refazer Alteração de Layout	Refaz a última ação desfeita no layout. Cada comando Refazer Alteração de Layout o levará um passo para frente na lista, enquanto houver ações desfeitas para serem refeitas. Algumas operações, por exemplo, Executar Script e Reduzir Dados, esvaziarão o buffer de Desfazer/Refazer. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+Y.
Pesquisar	Abre a caixa de Pesquisa referente ao objetivo ativo. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+F.
Seleções Atuais	Abre a caixa de diálogo Seleções Atuais, que permite visualizar as seleções ativas. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+O.
Assistente de Gráfico Rápido	Abre o Assistente de Gráfico Rápido, que permite criar um gráfico, de forma rápida e fácil, sem se preocupar com a grande quantidade de configurações e opções diferentes à disposição.
Incluir Marcador	Abre a caixa de diálogo Adicionar Marcador, que permite editar o nome do marcador. Esse comando também pode ser chamado pelo seguinte atalho de teclado: Ctrl+Shift+B.
Mostrar/Ocultar Notas	Mostra ou oculta as notas criadas pelo usuário e anexadas a objetos da pasta.
Tópicos da Ajuda	Abre a ajuda do QlikView.

Ajuda de Contexto	Exibe ajuda específica relativa ao objeto escolhido, por exemplo, um comando de menu. Depois de clicar nesse botão, basta mover o sinal de interrogação para o objeto sobre o qual precisa de ajuda.
-------------------	---

Tabelas



Uma tabela de banco de dados tem uma aparência semelhante à de uma planilha, já que os dados são armazenados em linhas e colunas. Como resultado, normalmente é bem fácil importar uma planilha para uma tabela de banco de dados. A principal diferença entre armazenar seus dados em uma planilha e armazená-los em um banco de dados é a forma como os dados são organizados.

Para obter o máximo de flexibilidade de um banco de dados, os dados precisam estar organizados em tabelas para que não ocorram redundâncias. Por exemplo, se você estiver armazenando informações sobre funcionários, cada funcionário só precisará ser inserido uma vez em uma tabela configurada somente para armazenar dados de funcionários. Os dados sobre produtos serão armazenados em sua própria tabela, e os dados sobre filiais serão armazenados em outra tabela. Esse processo é chamado de normalização.

Cada linha de uma tabela é conhecida como registro. Os registros são o local onde as informações individuais são armazenadas. Cada registro consiste em um ou mais campos. Os campos correspondem às colunas da tabela.

Por exemplo, você poderia ter uma tabela chamada "Funcionários", onde cada registro (linha) contém informações sobre um funcionário diferente, e cada campo (coluna) contém um tipo de informação diferente, como nome, sobrenome, endereço e assim por diante. Os campos devem ser designados como um determinado tipo de dados, seja ele texto, data ou hora, número ou outro tipo.

Outra maneira de descrever registros e campos é visualizar um antigo catálogo de cartões de biblioteca. Cada cartão do arquivo corresponde a um registro do banco de dados. Cada informação em um cartão individual (autor, título e assim por diante) corresponde a um campo do banco de dados.

Para mais informações sobre tabelas, confira Introdução às tabelas.

Formulários



Os formulários permitem que você crie uma interface do usuário na qual é possível inserir e editar seus dados. Com frequência, os formulários contêm botões de comando e outros controles que executam diversas tarefas.

Você pode criar um banco de dados sem usar formulários, simplesmente editando seus dados nas folhas de dados de tabela. Entretanto, a maioria dos usuários de banco de dados prefere usar formulários para exibir, inserir e editar dados nas tabelas.

Você pode programar botões de comando para determinar quais dados aparecerão no formulário, para abrir outros formulários ou relatórios ou para executar uma variedade de outras tarefas.

Por exemplo, talvez você tenha um formulário chamado "Formulário de Cliente" no qual você trabalha com dados do cliente. O formulário de cliente poderia ter um botão para abrir um formulário de pedido a partir do qual é possível inserir um novo pedido para o cliente.

Os formulários também permitem controlar como outros usuários interagem com os dados no banco de dados. Por exemplo, você pode criar um formulário que mostre apenas determinados campos e

permita que somente determinadas operações sejam executadas. Isso ajuda a proteger os dados e a garantir que eles sejam inseridos adequadamente.

Para mais informações sobre formulários, confira Introdução aos formulários.

Relatórios



Os relatórios são o que você usa para formatar, resumir e apresentar dados. Um relatório normalmente responde a uma pergunta específica, como "Quanto recebemos de cada cliente este ano?" ou "Em que cidades estão localizados nossos clientes?". Cada relatório pode ser formatado para apresentar as informações da forma mais legível possível.

Um relatório pode ser executado em qualquer ocasião e sempre refletirá os dados atuais do banco de dados. Geralmente, os relatórios são formatados para serem impressos, mas também podem ser exibidos na tela, exportados para outro programa ou enviados como um anexo de uma mensagem de email.

Para obter mais informações sobre relatórios, consulte o artigo Introdução aos relatórios no Access.

Consultas



As consultas podem executar várias funções diferentes em um banco de dados. Sua função mais comum é recuperar dados específicos das tabelas. Os dados que você deseja ver geralmente estão espalhados por várias tabelas, e as consultas permitem que você os exiba em uma única folha de dados. Além disso, como normalmente você não deseja ver todos os registros de uma vez, as consultas permitem adicionar critérios para "filtrar" os dados de modo a exibir somente os registros desejados.

Determinadas consultas são "atualizáveis", o que significa que você pode editar os dados nas tabelas subjacentes por meio da folha de dados da consulta. Se você estiver trabalhando em uma consulta atualizável, lembre-se de que suas alterações na verdade estão sendo feitas nas tabelas e não somente na folha de dados da consulta.

Há dois tipos de consulta básicos: consultas seleção e consultas ação. Uma consulta seleção simplesmente recupera os dados e os disponibiliza para uso. Você pode exibir os resultados da consulta na tela, imprimi-los ou copiá-los para a área de transferência. Ou você pode usar a saída da consulta como a fonte do registro para um formulário ou relatório.

Uma consulta ação, como o nome implica, executa uma tarefa com os dados. As consultas ação podem ser usadas para criar novas tabelas, adicionar dados a tabelas existentes, atualizar dados ou excluir dados.

Para mais informações sobre consultas, confira Introdução às consultas.

Modelagem de Dados

Modelagem de dados é o ato de explorar estruturas orientadas a dados. Como outros artefatos de modelagem, modelos de dados podem ser usados para uma variedade de propósitos, desde modelos conceituais de alto nível até modelos físicos de dados. Do ponto de vista de um desenvolvedor atuando no paradigma orientado a objetos, modelagem de dados é conceitualmente similar à modelagem de classes.

Com a modelagem de dados identificamos tipos de entidades da mesma forma que na modelagem de classes identificamos classes. Atributos de dados são associados a tipos de entidades exatamente como associados atributos e operações às classes. Existem associações entre entidades, similar às

associações entre classes – relacionamento, herança, composição e agregação são todos conceitos aplicáveis em modelagem de dados.

Modelagem de dados tradicional é diferente da modelagem de classes porque o seu foco é totalmente nos dados – modelos de classes permitem explorar os aspectos comportamentais e de dados em um domínio de aplicação, já com o modelo de dados podemos apenas explorar o aspecto dado. Por causa deste foco, projetistas de dados tendem a serem melhores em identificar os dados “corretos” em uma aplicação do que modeladores de objetos. No entanto, algumas pessoas modelam métodos de banco de dados (stored procedures, stored functions e triggers) quando estão realizando a modelagem física dos dados.

Apesar de o foco deste artigo ser modelagem de dados, existem normalmente alternativas para artefatos orientados a dados. Por exemplo, quando estamos na modelagem conceitual, os diagramas ORM (Object Role Model) não são a única opção. Além do Modelo Lógico de Dados, é comum a criação de diagramas de classes da UML.

Como modelos de dados são usados na prática?

Embora as questões de metodologias sejam abordadas depois, precisamos discutir como modelos de dados podem ser usados na prática para melhor entendê-los. Provavelmente, iremos nos deparar a três estilos básicos de modelos de dados:

- Modelos de dados conceituais: esses modelos, algumas vezes chamados modelos de domínio, são tipicamente usados para explorar conceitos do domínio com os envolvidos no projeto. Em equipes ágeis, modelos conceituais de alto nível são normalmente criados como parte do esforço inicial do entendimento dos requisitos do sistema, pois eles são usados para explorar as estruturas e conceitos de negócio estáticos de alto nível. Em equipes tradicionais (não ágeis), modelos de dados conceituais são normalmente criados como precursores aos modelos lógicos de dados (MLD) ou suas alternativas.
- Modelos Lógico de Dados (MLDs): MLDs são usados para explorar os conceitos do domínio e seus relacionados. Isso pode ser feito para o escopo de um simples projeto ou para uma empresa inteira. MLDs descrevem os tipos de entidades lógicas, tipicamente referenciadas simplesmente como tipos de entidades, os atributos de dados que descrevem essas entidades e os relacionamentos entre as entidades. MLDs são raramente usados em projetos ágeis apesar de normalmente estarem presentes em projetos tradicionais (onde eles raramente adicionam muito valor na prática).
- Modelos Físicos de Dados (MFDs): MFDs são usados para projetar o esquema interno de um banco de dados, descrevendo as tabelas de dados, as colunas de dados das tabelas e o relacionamento entre as tabelas. MFDs normalmente são bastante úteis em projetos ágeis e tradicionais, por isso este será o foco deste artigo: modelagem física dos dados.

Embora MLDs e MFDs parecerem similares, e eles de fato são, o nível de detalhes que eles modelam pode ser significativamente diferente. Isso porque o objetivo de cada diagrama é diferente – podemos usar um MLD para explorar conceitos do domínio com os envolvidos no projeto e MFD para definir o projeto do banco de dados.

A Figura 1 apresenta um simples MLD e a Figura 2 um simples MFD, ambos modelando o conceito de clientes e endereços, assim como o relacionamento entre eles. Ambos os diagramas seguem a notação de Barker, que será descrita a seguir. Note como o MFD mostra mais detalhes, incluindo uma tabela associativa necessária para implementar a associação, assim como as chaves necessárias para manter os relacionamentos. Mais detalhes sobre esses conceitos serão descritos a seguir.

MFDs devem também refletir os padrões de nomenclatura de banco de dados da organização. Neste caso, uma abreviação do nome da entidade é colocado para cada nome de coluna e uma abreviação para “número” foi consistentemente introduzida. Um MFD deve também indicar os tipos de dados das colunas, tais como integer e char(5). Apesar de a Figura 2 não mostrá-las, tabelas de referência como para o endereço é usado, assim como para estados e países estão implícitos pelos atributos.

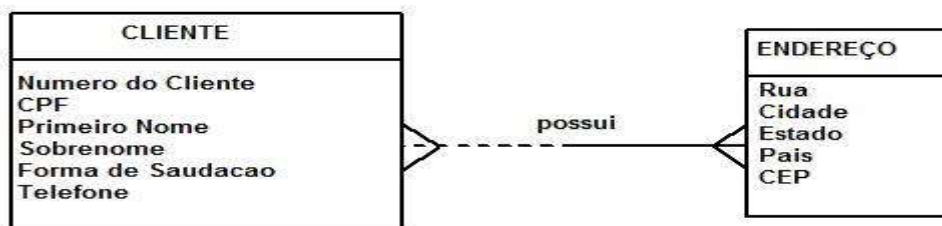


Figura 1. Um simples modelo lógico de dados

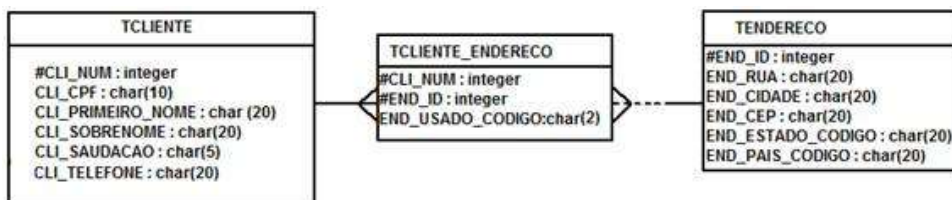


Figura 2. Um simples modelo físico de dados

Modelos de dados podem ser usados efetivamente tanto no nível da empresa como de projetos. Os arquitetos da empresa normalmente criarão um ou mais MLDs de alto nível que descreve as estruturas de dados que apoiam toda a empresa, normalmente chamados de modelos de dados da empresa ou modelos de informação da empresa.

Um modelo de dados da empresa é uma das várias visões que os arquitetos da empresa podem escolher para manter e apoiar – outras visões podem explorar a infraestrutura de rede/hardware, a estrutura da organização, infraestrutura de softwares o processo de negócios, dentre outros. Esses modelos provêm informações que uma equipe de projeto pode usar como conjunto de restrições e também como descrição da estrutura do sistema.

Equipes de projeto tipicamente criarão MLDs como um dos principais artefatos de análise quando seu ambiente de implementação é predominantemente procedural por natureza, por exemplo quando estão usando COBOL estruturado como linguagem de implementação. MLDs são também boas escolhas quando um projeto é orientado a dados, como um data warehouse ou sistema de relatório. No entanto, MLDs são normalmente escolhas ruins quando uma equipe de projeto está usando tecnologias orientadas a objeto ou baseadas em componentes porque os desenvolvedores trabalhariam melhor com diagramas UML ou quando o projeto não é orientado a dados. Como uma dica de modelagem, aplique os artefatos corretos para aquele trabalho a ser desenvolvido.

Quando um banco de dados relacional é usado para armazenar dados, equipes de projeto são aconselhadas a criar um MFD para modelar um esquema interno. MFD normalmente é apenas um dos artefatos de projeto críticos para projetos de desenvolvimento de aplicações de negócio.
