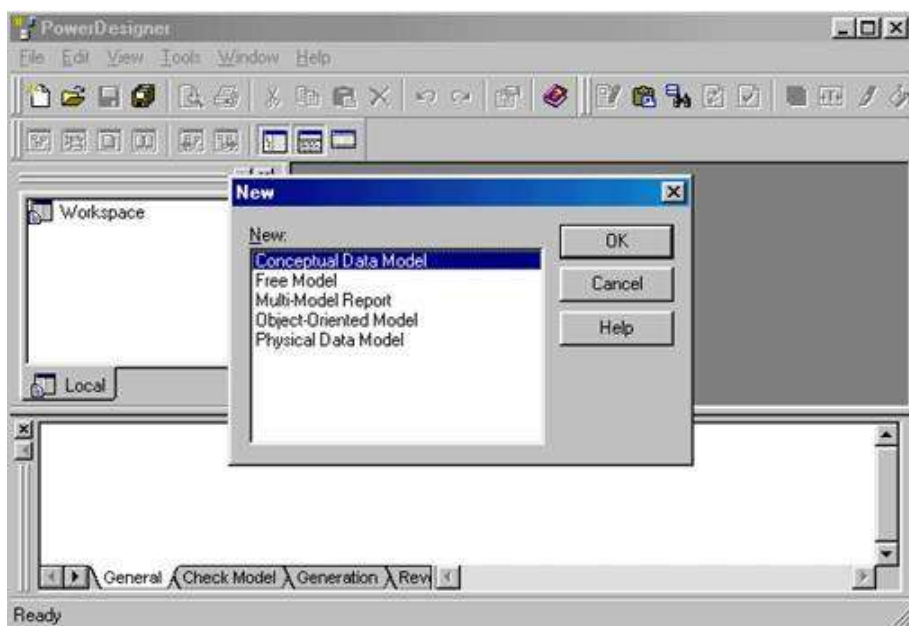


Modelo Entidade Relacionamento

Ao iniciar um novo projeto no PowerDesigner, será apresentada uma caixa de diálogo. Esta possibilita a escolha de vários tipos de modelos a serem criados dentre eles: o modelo conceitual (ver Figura 1), modelo físico, modelo orientado a objetos utilizando a linguagem UML, relatórios e modelos livres (não padronizados). Como neste exemplo estamos projetando um banco de dados e estamos na etapa de criação do modelo conceitual, escolhemos a opção Conceptual Data Model.

A partir daí será elaborado o MER (Modelo Entidade-Relacionamento). Este é formado por um conjunto de entidades que se relacionam. O MER é representado graficamente através do DER (Diagrama de Entidade e Relacionamento). Para a construção do DER existe uma barra de ferramentas (palette) contendo todos os objetos necessários (ver Figura 2). Caso a palette não esteja visível na sua área de trabalho, abra o menu Tools -> Customize Toolbars. . . -> e selecione Palette.



O PowerDesigner

O PowerDesigner é uma ferramenta da Sybase para modelagem e soluções de projeto para empresas que necessitam construir ou refazer aplicações de forma rápida e consistente.

Para isto, o PowerDesigner possui componentes para diversas atividades: modelagem de processos de negócios, modelagem de dados, modelagem de sistemas orientados a objetos, armazenamento e controle dos modelos desenvolvidos. Para cada um desses componentes, existe ainda uma série de funcionalidades agregadas. Entretanto, nos restringiremos a discutir as características envolvidas com a modelagem e projeto de bancos de dados.

Sabe-se que um projeto de banco de dados é composto de três fases: modelo conceitual, modelo físico e a criação do script (código com os comandos DDL) responsável por criar toda a estrutura necessária no sistema gerenciador de banco de dados (SGBD). Agora apresentaremos como o PowerDesigner pode auxiliar o DBA durante estas etapas.

Você pode criar um novo projeto de banco de dados e importar o esquema de banco de dados de um banco de dados existente, um arquivo de script. sql ou um aplicativo da camada de dados (dcpac).

Você poderá invocar então as mesmas ferramentas de designer visual (Editor Transact-SQL, Designer de Tabela) disponíveis para o desenvolvimento de bancos de dados conectados para fazer alterações no projeto de banco de dados offline e para publicar as alterações no banco de dados de produção.

As alterações também podem ser salvas como um script a ser publicado posteriormente. Usando o painel Propriedades do Projeto, você pode alterar a plataforma de destino para versões diferentes do SQL Server (incluindo o SQL Azure).

Os dois procedimentos a seguir obtêm essencialmente a mesma meta criando um novo projeto de banco de dados e importando esquema de um banco de dados existente. Cada objeto de banco de dados será representado como um arquivo de script do SQL (.sql) no Gerenciador de Soluções. Para obter mais informações sobre como importar esquema de banco de dados de um instantâneo

Para criar um novo projeto de banco de dados fora de um banco de dados conectado

Clique com o botão direito no nó TradeDev no Pesquisador de Objetos do SQL Server e selecione Criar Novo Projeto.

Na caixa de diálogo Importar Banco de Dados, observe que as configurações de Conexão de banco de dados de origem foram predefinidas pelo banco de dados que você selecionou no Pesquisador de Objetos do SQL Server. Na configuração Projeto de destino, altere o nome do projeto para TradeDev.

Na seção Importar Configurações, observe as opções para importar objetos e configurações específicas, e criar pastas para cada esquema e/ou tipo de objeto. Para obter uma hierarquia organizada de todos os seus objetos de banco de dados, aceite todas as configurações padrão e clique em Iniciar.

A caixa de diálogo Importar Banco de Dados mostra uma barra de progresso e exibe uma lista de objetos que o SSDT está importando. Quando a operação de importação tiver sido concluída, clique em Concluir para sair da tela final.

Examine a hierarquia no Gerenciador de Soluções. Expanda a pasta dbo e você localizará pastas Funções, Tabelas e Exibições separadas. Observe que as tabelas e a função são agrupadas nas suas pastas de esquema.

Clique duas vezes em Products.sql em Tabelas. O Designer de Tabela abre, mostrando a interpretação visual da tabela na Grade de Colunas e a definição de script da tabela no Painel de Script. Isto é idêntico ao que nós vemos na seção Desenvolvimento de banco de dados conectado.

Desmarque a caixa Permitir Valores Nulos para a coluna CustomerId. Pressione CTRL + S para salvar o arquivo.

Clique com o botão direito no projeto TradeDev no Gerenciador de Soluções e selecione Build para criar o projeto de banco de dados.

Os resultados da operação Compilar podem ser vistos na Janela de Saída

Para criar um novo projeto e importar esquema de banco de dados existente

Clique em Arquivo, Novo e Projeto. Na caixa de diálogo Novo Projeto, selecione SQL Server no painel esquerdo. Observe que há somente um tipo de projeto de banco de dados: o Projeto de Banco de Dados do SQL Server. Não há nenhum projeto específico de plataforma, como em versões anteriores do Visual Studio. Você poderá definir sua plataforma de destino na caixa de diálogo Configurações do Projeto depois que o projeto tiver sido criado. Esse tipo de tarefa será abordado no tópico Como: Alterar a plataforma de destino e publicar um projeto de banco de dados.

Altere o nome do projeto para TradeDev e clique em OK para criar o novo projeto.

Clique com o botão direito no projeto TradeDev criado no Gerenciador de Soluções, selecione Importar e Banco de Dados.

A caixa de diálogo Importar Banco de Dados é aberta. Na seção Conexão de banco de dados de origem, clique em Escolher um banco de dados e selecione TradeDev. Se TradeDev estiver ausente da lista suspensa, use o botão Nova Conexão para editar as Propriedades da Conexão.

Na seção Importar Configurações, observe as opções para importar objetos e configurações específicas, e criar pastas para cada esquema e/ou tipo de objeto. Para obter uma hierarquia organizada de todos os seus objetos de banco de dados, aceite todas as configurações padrão e clique em Iniciar.

A caixa de diálogo Importar Banco de Dados mostra uma barra de progresso e exibe uma lista de objetos que o SSDT está importando. Quando a operação de importação tiver sido concluída, clique em Concluir para sair da tela final.

Examine a hierarquia no Gerenciador de Soluções. Expanda a pasta dbo e você localizará pastas Funções, Tabelas e Exibições separadas. Observe que as tabelas e a função são agrupadas nas suas pastas de esquema.

Clique duas vezes em Products. sql em Tabelas. O Designer de Tabela abre, mostrando a interpretação visual da tabela na Grade de Colunas e a definição de script da tabela no Painel de Script. Isto é idêntico ao que nós vemos na seção Desenvolvimento de banco de dados conectado.

Desmarque a caixa Permitir Valores Nulos para a coluna CustomerId. Pressione CTRL + S para salvar o arquivo.

Clique com o botão direito no projeto TradeDev no Gerenciador de Soluções e selecione Build para criar o projeto de banco de dados

O banco de dados, muitas vezes, é a parte mais importante do sistema, pois é onde fica a informação. Por consequência, um projeto de banco de dados é essencial para o desenvolvimento de sistemas de informação. Um projeto de banco de dados bem ou mal realizado pode determinar o sucesso ou insucesso do sistema. Muitas vezes a falta de clareza ao entender a natureza exata do ambiente onde o banco de dados será aplicado, leva à criação de bancos de dados ruins, que não alcançam o objetivo esperado pelo cliente. Neste artigo vou apresentar as etapas de um projeto de banco de dados.

As Etapas de Um Projeto de Banco de Dados

Um projeto de banco de dados é sub-dividido em etapas onde o objetivo é a criação de um banco de dados otimizado que atenda as expectativas do cliente. E nesse contexto, os modelos de dados são muito importantes para a transmissão de ideias entre o cliente e o projetista, bem como facilitar a manutenção do banco de dados no futuro. O projeto de banco de dados é basicamente, dividido em Projeto Conceitual, Projeto Lógico e Projeto Físico. Porém antes de começar a projetar o banco de dados é necessário a realização de análise de requisitos junto ao cliente. Essa talvez seja a fase mais importante do projeto, pois é nessa hora que as necessidades e expectativas do cliente são transmitidas para o projetista

Projeto Conceitual

O Projeto Conceitual se baseia na especificação de requisitos criada na etapa anterior. A partir deste insumo de informações é gerado um esquema conceitual do banco de dados. Esquema conceitual é uma visão macro do banco de dados, uma descrição de alto nível da estrutura. Os modelos de Entidade-Relacionamento são muito utilizados para descrever os esquemas conceituais. É importante frisar que nesta fase descreve-se o conteúdo de informação e não a estrutura onde elas serão armazenadas (tabelas, colunas, visões, etc...)

A modelagem de dados é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de um banco de dados. Ela faz parte do ciclo de desenvolvimento de um sistema de informação e é de vital importância para o bom resultado do projeto. Modelar dados consiste em desenhar o sistema de informações, concentrando-se nas entidades lógicas e nas dependências lógicas entre essas entidades.

Modelagem de dados ou modelagem de banco de dados envolve uma série de aplicações teóricas e práticas, visando construir um modelo de dados consistente, não redundante e perfeitamente aplicável em qualquer SGBD moderno.

Modelo Conceitual

A modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica.

O diagrama de dados que deve ser construído aqui é o Diagrama de Entidade e Relacionamento, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.

Modelo Lógico

O modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

Modelo Físico

No modelo físico fazemos a modelagem física do modelo de banco de dados. Neste caso leva-se em conta as limitações impostas pelo SGBD escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico.

O que é um Modelo de Dados?

Os modelos de dados são artefatos que representam de forma estática, sob o ponto de vista dos dados, a captura dos requisitos de informação e as regras de negócio através das entidades (ou classes de negócio), atributos, relacionamentos e demais regras representadas em formas gráficas e textuais.

A modelagem de dados é o processo que engloba o entendimento, a especificação e a validação do modelo de dados proposto, seguindo as diretrizes, padrões e melhores práticas recomendadas pela Governança de Dados de cada empresa.

O modelo de dados é um dos principais artefatos utilizados na Gestão dos Dados. Podem ser utilizados em vários assuntos ligados à Gestão de Dados. Como exemplo:

Arquitetura de Dados: Onde os modelos de dados corporativos são representados, fornecendo informações das principais entidades de negócio corporativas da empresa.

Governança de Dados: Os modelos de dados exercem um importante papel nesta função. Através deles podemos identificar quais os gestores das informações, quais pessoas ou aplicações têm acesso aos dados representados em cada modelo.

Modelagem de Dados: Função específica para análise, especificação e implementação dos modelos de dados. É nesta função onde os modelos de dados são criados.

Integração das informações: Os mecanismos de integração de informação podem ser representados nos modelos de dados. Várias ferramentas para modelagem de dados fornecem recursos para criação de propriedades específicas, onde as informações sobre os mecanismos de integração utilizados podem ser representadas facilmente.

Gestão de Metadados: Boa parte dos metadados das empresas é representada através dos modelos de dados das aplicações. A gestão de todos esses modelos incluindo a sua guarda e disponibilidade é feita através da função Gestão de Metadados.

Vale ressaltar que, atualmente, no mundo da Gestão de Dados, a Modelagem de Dados não é encarada como uma função específica dentro da versão atual do guia DAMA-DMBOK® - Data Management Body of Knowledge. No guia, as atividades de modelagem de dados estão contempladas na função “Desenvolvimento dos Dados”, junto com outras atividades focadas na criação de estruturas de dados dentro do ciclo de vida do desenvolvimento das aplicações, tais como: levantamento e análise dos requisitos de dados, implementação dos modelos de dados nos SGBDs e disponibilização de dados armazenados nos SGBDs.

As técnicas de construção e representação acerca dos modelos de dados são muitas. Os modelos de dados são criados e ajustados dentro de várias fases do ciclo de vida do sistema. Para cada fase são recomendados modelos com diferentes graus de abstração, representação e detalhamento. Esta abordagem resulta em uma divisão onde três tipos de modelos são adotados. São eles:

- Modelo Conceitual de Dados;
- Modelo Lógico de Dados;
- Modelo Físico de Dados.

O Modelo Conceitual de Dados

Um modelo conceitual de dados é um modelo de dados de alto nível. Sua principal finalidade é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio. Ou seja, é um modelo que não sofre interferência de fatores tecnológicos e fatores de projeto em sua construção. É um modelo não tecnológico e não implementável.

No desenvolvimento de soluções é o primeiro modelo que deve ser desenvolvido, já na fase de levantamento de requisitos preferencialmente pelo Gestor de Dados de Negócio ou outro profissional acompanhado de sua supervisão/orientação.

Como um importante instrumento utilizado para representar o negócio da empresa, a construção de um Modelo Conceitual de Dados não deve-se limitar somente à necessidade de representar um novo sistema/aplicação que será desenvolvida na empresa, mas sim, desenvolver visões globais de todo o negócio, bem como integrar essas visões com demais artefatos, geralmente criados por outras Arquiteturas Especialistas, como por exemplo: modelos canônicos, modelos de processos e demais modelos de alto nível.

A técnica para construção de um Modelo Conceitual de Dados é simples. De forma geral, o profissional deve conhecer os mecanismos de abstração utilizados e ter habilidade específica para extrair e entender a visão de negócio do assunto em questão. Entre os componentes de um modelo conceitual, podemos relacionar:

- Entidades;
- Atributos;
- Relacionamentos;

A utilização e representação destes componentes será tratada no próximo artigo.

Recomendações para Modelagem Conceitual de Dados

Somente o domínio da técnica não é necessário para construir modelos conceituais de qualidade. Relaciono a seguir algumas dicas que devem ser consideradas quando o profissional implementar a prática de modelagem conceitual de dados na empresa. São elas:

- Estabeleça um processo formal de trabalho e padrão para modelagem conceitual. O conteúdo deste e do próximo artigo é uma boa fonte para elaborar este padrão.
- Defina um papel responsável pela construção do modelo conceitual de dados. De forma geral, analistas de negócio ou gestores de dados de negócio são os profissionais mais indicados para realizar esta construção.
- O Modelo conceitual deve ser elaborado sempre com a participação dos profissionais de negócio.
- A validação do modelo conceitual deve ser feita por profissionais da área de negócio. Nunca por profissionais de TI.
- Tenha sempre em mente o objetivo do modelo conceitual (representar o negócio), portanto ignore questões técnicas como normalização e performance.
- Modelos conceituais de dados validados devem ser armazenados e seu acesso deve ser concedido para todos os profissionais envolvidos (TI e Negócio).
- Todo modelo é considerado “vivo”, portanto a representação do negócio de hoje poderá não ser a de amanhã. Estabeleça um mecanismo para avaliar o quanto o modelo é atual.

São vários os tipos de banco de dados existentes e, com o passar dos anos e a evolução dos computadores, tornou-se possível o armazenamento de dados no formato digital.

No entanto, a expressão “banco de dados” não é tão simples assim, uma vez que possui duas aplicações distintas e muitos consideram que “dados” e “informações” são palavras sinônimas, o que não é verdade.

Dados e Informações

Antes de entender o que é um banco de dados, é importante saber a diferença entre as palavras “dados” e “informações”. Os dados são os fatos brutos, em sua forma primária, e podem não fazer nenhum sentido quando estão isolados; já as informações são o agrupamento de dados organizados, de forma que façam sentido e gerem algum conhecimento.



O que é um 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.

Quais os tipos de bancos de dados existentes e suas diferenças?

Primeiramente, é preciso dividi-los em duas categorias: os Modelos Relacionais e os Não-Relacionais. Vejamos:

1. Banco de Dados Relacional

Os bancos de dados relacionais são fundamentados no paradigma da orientação a conjuntos, uma vez que sua base é construída em cima da teoria dos conjuntos.

Esses bancos armazenam dados em estruturas chamadas tabelas, compostas por colunas — atributos e linhas —, tuplas ou registros. Sua linguagem é a SQL (Structured Query Language).

Eles são usados para dados tabulares, de fácil inserção e recuperação. Dominam atualmente a maior fatia do mercado de banco de dados, devido à sua aplicabilidade. Seus principais representantes são o Oracle, SQL Server, MySQL e PostgreSQL.

Sistemas ERP, CRM e similares utilizam bancos relacionais, cuja característica principal é a preocupação com a consistência de seus dados, garantida pelo princípio conhecido como ACID:

Atomicidade: Numa transação, ou todos os registros são alterados ou tudo é restaurado à condição original, garantindo que nenhuma alteração fique pela metade;

Consistência: Assegura que os dados sejam consistentes antes e depois de uma alteração. Por exemplo, não se pode vender um item cuja quantidade na compra é maior do que a disponível em estoque;

Isolamento: É encarregado de isolar as transações, de forma que elas sejam visíveis ao resto da aplicação somente depois de concluídas.

Um exemplo é o ajuste do salário de um funcionário. Enquanto o Departamento de Pessoal não termina de atualizar o cadastro do funcionário, os demais setores continuam a ver o salário antigo. Depois, então, de concluída a transação, uma nova consulta no setor financeiro, por exemplo, mostrará o novo salário;

Durabilidade: Toda informação do banco de dados precisa ser durável, somente podendo ser alterada pela aplicação através de comandos DML (Data Manipulation Language), que fazem "inserts", "updates" ou "deletes".

Banco de Dados Não-Relacional

Esse tipo de Banco de Dados surge como solução para situações nas quais os bancos relacionais não atendem de forma satisfatória. Ambientes com dados mistos — como imagens, mapas e tabelas — que não podem ser facilmente tabulados em linhas e colunas necessitam de uma solução não-relacional.

Surgem aí bancos conhecidos como NoSQL (Do inglês, Not Only SQL). Esses bancos dão vazão a demandas de gigantes como Google, por exemplo, que oferecem, no seu portfólio, as mais diversas soluções, desde contas de e-mail, dados espaciais e armazenamento de imagens e Cloud Computing. Podemos citar como exemplos de bancos NoSQL, o MongoDB, Redis e Cassandra.

Enquanto bancos relacionais usam o conceito de ACID como vimos, os bancos NoSQL usam o paradigma conhecido como CAP:

Consistency: Consistência nas informações armazenadas;

Availability: Disponibilidade do banco de dados;

Partition Tolerance: Tolerância ao particionamento das informações.

Sua natureza faz com que o responsável precise avaliar qual dos atributos mencionados é o mais importante para seu negócio, pois o NoSQL não garante totalmente os três.

Qual a Solução mais Adequada a Escolher?

A escolha entre bancos relacionais e não relacionais está intimamente ligada à aplicação:

Em se tratando de sistemas desenvolvidos para criar, alterar, excluir e consultar informações que tenham um padrão de formato regular, o banco de dados mais adequado é o Relacional. Os sistemas de gerenciamento financeiro, administração de empresas, contábeis etc., são os mais indicados para bancos relacionais.

Em contrapartida, grandes soluções baseadas em nuvem e dados não tabulados com características heterogêneas demandam a utilização de um banco Não-relacional.

Esperamos, com este post, ter contribuído para enriquecer seu conhecimento sobre os tipos de bancos de dados. Convidamos você a compartilhar este texto nas redes sociais, de forma a difundir ainda mais o conhecimento sobre o assunto.

Modelagem Relacional

O conceito principal vem da teoria de conjuntos (álgebra relacional) atrelado a ideia de que não é relevante ao usuário saber onde os dados estão ou como eles se encontram, representado por uma coleção de tabelas (entidade/relação) é um conjunto de linhas (tuplas) uma lista de valores de atributos.

Modelo entidade relacionamento proposto por Peter P. Chen pode ser melhor compreendido por uma teoria chamada de “A lei do Mundo”, teoria essa, que conceitua que o mundo está cheio de coisas que possuem características próprias e que se relacionam entre si. Sua análise da teoria pode ser dividida em três partes.

“O Mundo Está Cheio De Coisas”

Tudo que possa ser caracterizado, conceituado, real ou imaginário, no nosso Universo (Mundo), é definido como coisa, que futuramente, dependendo da abordagem, poderá ser definido como uma entidade.

“Que Possuem Características Próprias”

Características comuns percebidas entre as coisas de modo que haja a possibilidade de enquadramento dessas coisas em conjuntos particulares. Exemplo: “conselho de economia, conselho de medicina, conselho de odontologia” todos podem ser enquadrados em um mesmo conjunto, denominado como Órgão normalizador.

“E Que Se Relacionam Entre Si”

São as relações entre as coisas. Como as mesmas irão relaciona-se entre elementos individualizados de diferentes conjuntos ou entre elementos de um mesmo conjunto. A forma de comunicação entre as coisas ou um conjunto delas, Exemplos: “Adail é credenciado pelo conselho de economia” é um relacionamento entre elementos de diferentes conjuntos. “Adail é substituto de Caio” é um relacionamento entre elementos do mesmo conjunto.

Objeto De Dados Ou Entidade

É a representação genérica de um componente do mundo real, sobre o qual desejamos armazenar informações, uma representação de quase todas as informações com várias propriedades que devem ser compreendidas pelo sistema de informação, qualquer coisa que produza ou consuma informações. Entidade são coisas significativas sobre a qual a organização deseja guarda, ou seja, (coletar, manter e etc) dados podendo ser algo tangível ou intangível. Ex.: Cliente; Produto; Contrato de Operação

Vários autores defendem formas de identificar e classificar as entidades, onde suas tipificações mais comuns são:

Coisas tangíveis: todos os elementos que tenham existência concreta. Fisicamente existente. Ex: Produto, animal, carro

Funções: todo o tipo de papel, atribuição, classificação, capacitação, ou outra característica que especifique atuação. Ex: Cliente, professor, departamento

Eventos ou ocorrências: só conseguem ser percebidos ou caracterizados, enquanto uma certa ação se desenrola. Ex: Lançamento em conta corrente

Nomenclatura e Dicionarização

Como objeto de comunicação, um modelo de dados, deve ter a capacidade de informar sobre o que representa de forma clara, sendo uma unificação do diagrama com informações textuais, sua representação gráfica por si só normalmente não é suficiente para gerar entendimento dos conceitos representados, entretanto um modelo deve ser auto-explicativo, fundamental, é necessário para gerar a interpretação correta, onde a definição do nome do objeto bem como a sua conceituação tem o papel fundamental para gerar esse entendimento, desta forma, nomes e definições não podem gerar ambiguidade, isto é devem ser precisa, de forma que não gere dúvida, incerteza, interpretação de conceitos distintos.

Cada um dos elementos identificados e representados deverá ser definido claramente para que, associando-se seu nome, sua representação e sua definição, sejamos capazes de ter o completo entendimento do conceito que estes procuram transmitir. Ex: Cobertura: o que serve para cobrir, para seguros.

A nomenclatura de objetos deve prever nomes breves e objetivos, que identifiquem facilmente o conteúdo da entidade. Estar no singular, pois a pluralidade decorre, naturalmente da cardinalidade. Ex: PESSOA, CLIENTE, CONTRATO

A definição do objeto deve ser uma definição formal dos elementos, o que irá gerar o dicionário de dados, que deverá trazer a conhecimento público a toda e qualquer informação útil para o processo de compreensão e unificação de conceitos, que possam parecer triviais para quem está modelando, não serão do mesmo modo triviais para outras pessoas que não tenham conhecimento prévio.

Atributo

Atributo é tudo o que se pode relacionar como próprio da entidade (propriedade) que de alguma maneira a qualifique e a distinga de outras, estes podem ser classificados e identificados como:

Atributos descritivos: atributo que seja capaz de demonstrar, ou representar, características formadoras, ou pertencentes, a um objeto. Ex: Data de nascimento, idade, sexo.

Atributos Nominativos: atributo que além de cumprirem a função de descritivos, também servem como definidores de nomes ou rótulos de identificação aos objetos aos quais pertencem. Ex: código do. . . , matrícula, número...

Atributos Referenciais: atributo que não pertencem propriamente a entidade onde estão, mas fazem algum tipo de referência dessa entidade com outra entidade.

Relacionamento

Relacionamento é a relação existente entre entidades, isto é a ligação lógica entre duas entidades que representa uma regra ou restrição de negócio, possibilitando entender como uma entidade se comporta em relação às demais, qual o seu grau de dependência de outras entidades e qual a associação de dados existentes entre elas.

Representação de Relacionamentos de Modelo

Podem ser estabelecidos mais de um relacionamento entre entidades, de acordo com a regra de negócio a ser representada onde cada entidade pode participar de vários relacionamentos.

O verb phrase é a denominação do relacionamento, devendo representar o conceito observado. Existem várias necessidades que justifiquem a existência desta denominação, destacam-se as situações que podem gerar ambiguidade de compreensão, por exemplo quando existe dois relacionamentos entre as mesmas entidades representando conceitos distintos, em auto-relacionamentos ou quando existem várias possibilidades claras de relacionamentos e será representado determinado relacionamento. O verbo deve ser na forma verbal ativa ou na forma verbal passiva. Ex: Pessoa / pedido = possui ou pertence a.

Os relacionamentos possuem características que os tipificam.

Cardinalidade: Indica a quantidade de ocorrências de determinado relacionamento, Sempre a maior possibilidade, sua representação é variável de acordo com a notação, por exemplo N para Perter Chen e Para James Martin

N : várias vezes

1 : apenas uma vez

0: não acontece

Opcionalidade: Analisa os relacionamentos pelo lado da obrigatoriedade das ocorrências de uma entidade se ligarem às ocorrências das outras. Podem ser de 3 tipos:

Opcional: quando as ocorrências das entidades que se relacionam são independentes das outras.

Contingente: a obrigatoriedade só acontece por um lado do relacionamento e somente uma entidade possui independência com relação a outra.

Mandatário: As ocorrências das entidades somente podem existir se ambas (dominante e dependente) existirem.

Em uma série de quatro artigos, apresentarei alguns conceitos básicos sobre Bancos de Dados, mais especificamente sobre o Modelo Relacional de Dados. Para a melhor utilização, ou seja, para uma utilização eficiente de bancos de dados como o Microsoft Access, SQL Server, ORACLE, DB2 ou qualquer outro banco de dados relacional, é importante o conhecimento e correto entendimento dos conceitos apresentados nesta série de artigos. Vou abordar os seguintes Conceitos:

Entidades e atributos

Chave primária

Relacionamentos entre entidades (tabelas)

Integridade Referencial

Normalização de tabelas

Um Problema Proposto

Arquitetura do Microsoft Access.

Nota: Os exemplos apresentados utilizarão telas do Microsoft Access. Porém os princípios básicos do modelo relacional aplicam-se a qualquer banco de dados baseado no modelo relacional de dados. Estes bancos de dados são algumas vezes denominados: SGBDR - Sistemas Gerenciadores de Banco de Dados Relacionais.

Entidades e Atributos:

Toda a Informação de um banco de dados relacional é armazenada em Tabelas, que na linguagem do modelo relaciona, também são chamadas de Entidades. Por exemplo, posso ter uma Tabela "Clientes", onde seriam armazenadas informações sobre os diversos clientes.

Sobre cada um dos clientes podem ser armazenadas diversas informações tais como:

Nome

RG

CPF

Rua

Bairro

Telefone

CEP

Data de Nascimento

Essas diversas características de cada Cliente são os "Atributos" da entidade Cliente, também chamados de campos da tabela Cliente.

"O Conjunto de todos os Atributos de um cliente e os valores dos atributos é o que forma o Registro do Cliente".

Com isso temos uma Tabela que é constituída por um conjunto de Registros (uma linha completa com informações sobre o cliente) e cada Registro formado por um conjunto de atributos (Nome, Endereço, etc).

Resumindo: Entidade ou Tabela: Um conjunto de Registros.

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais

Quando se inicia o desenvolvimento de um novo sistema, ou mesmo de uma nova funcionalidade para um sistema existente, um dos primeiros passos a ser executado é o estudo e levantamento dos requisitos necessários para a construção do produto final. Durante essa análise, identifica-se as principais partes e objetos envolvidos, suas possíveis ações e responsabilidades, suas características e como elas interagem entre si.

A partir das informações obtidas, pode-se desenvolver um modelo conceitual que será utilizado para orientar o desenvolvimento propriamente dito, fornecendo informações sobre os aspectos relacionados ao domínio do projeto em questão.

Quando estudamos os conceitos introdutórios de banco de dados sempre falando do modelo conceitual entidade relacionamento (ER) e do modelo lógico relacional. Explicamos que durante um projeto de banco de dados a evolução natural leva você a construir os dois modelos a partir dos requisitos do cliente.

Legal, então você começou pelo mini mundo, juntou os requisitos e construiu seu modelo ER! Agora você está com dificuldade de passar esse modelo para o modelo relacional. Seus problema acabaram! Nas próximas linhas vamos explicar o passo a passo da transformação do seu diagrama ER em tabelas.

O livro do Navathe sugere um algoritmo que nos ajuda no processo de transformar um esquema ER e um mapeamento relacional. Vamos utilizar o esquema abaixo para ilustrar o passo-a-passo do mapeamento.
