

Módulo 7

Bancos de Dados

INTRODUÇÃO

O Delphi é a melhor ferramenta para desenvolvimento de sistemas comerciais que lidam com dados. No entanto o Delphi não armazena os dados. Para isso você precisará utilizar um Banco de Dados. Um banco de dados é um sistema capaz de armazenar dados de forma eficiente. Existem diversos deles. Na página seguinte você poderá observar uma tabela com alguns deles incluindo algumas vantagens e desvantagens. É evidente que a escolha do Banco de Dados vai depender inteiramente do seu cliente.

SGDB - Sistemas Gerenciadores de Banco de Dados

Essa lição apresenta os sistemas gerenciadores de banco de dados, seus serviços, recursos e modelos.

Os SGBDs são sistemas responsáveis em armazenar os dados e fornecer serviços capazes de manipulá-los. Para armazenar os dados em um disco, os SGBDs possuem uma forma estruturada padrão de representar os registros e campos, fornecendo serviços que permitem a criação e alteração dessas definições de dados. Fornecem ainda, um mecanismo interno para manter os dados no disco e para saber onde está cada elemento em particular. Também são responsabilidades dos SGBDs disponibilizar serviços para incluir, classificar, atualizar e eliminar os dados armazenados.

Pela forma com que os dados são armazenados e de como eles se relacionam podemos classificar os SGBDs através de vários modelos até chegarmos aos bancos de dados relacionais.

MODELOS DE BANCO DE DADOS

Analisando os diversos modelos de banco de dados, pode-se notar uma evolução no relacionamento entre os dados, eliminando cada vez mais a redundância e proporcionando mais flexibilidade e, portanto uma maior facilidade de manutenção das definições de dados. Para melhor entendermos os modelos vamos utilizar os dados das três guias de pedidos da figura 7.1.

| Pedido: 1 Data: 20/10/96 | | |
|--------------------------|----------------|--------------------|
| Cliente | Ana Maria Lima | CPF 999.876.555-22 |
| Itens | | |
| Produto | Quantidade | Valor |
| Caneta | 10 | 10,00 |
| Lápis | 5 | 5,00 |
| | | 15,00 |

| Pedido: 2 Data: 21/10/96 | | |
|--------------------------|------------|--------------------|
| Cliente | Maria José | CPF 111.111.111-22 |
| Itens | | |
| Produto | Quantidade | Valor |
| Caneta | 15 | 15,00 |
| Caderno | 5 | 50,00 |
| | | 65,00 |

| Pedido: 3 Data: 22/10/96 | | |
|--------------------------|------------|--------------------|
| Cliente | Maria José | CPF 111.111.111-22 |
| Itens | | |
| Produto | Quantidade | Valor |
| Lápis | 15 | 15,00 |
| Caderno | 1 | 10,00 |
| | | 25,00 |

Fig. 7.1 – Guias de pedido.

| Banco de Dados | Prós | Contras |
|----------------|---|---|
| Oracle | Um dos melhores BD's do mundo, sendo muito rápido e confiável. Roda em diversas plataformas (Windows, Unix, Linux...). Já possui uma versão gratuita. A Oracle 10g Express Edition. | Necessita de um servidor dedicado. O preço é muito alto. É de difícil manutenção. Necessita de DBA (Administrado de Banco de Dados), um profissional treinado apenas para lidar com BD's. |
| MS-SQL | BD da Microsoft. Possui uma versão gratuita. O SQL Server 2005 Express Edition. | Preço alto. Necessita DBA. Não funciona em Unix's e Linux's. |
| DB2 | BD da IBM, confiável e também muito rápido. | Preço alto e difícil manutenção. Necessita de DBA. |
| Interbase | BD da Borland. Foi lançada uma versão gratuita: a 6.0. Rápido. Tem vários componentes no Delphi que se comunicam diretamente com o BD, dispensando o BDE. Roda em diversas plataformas. | Difícil manutenção. Corrompe com facilidade o BD e às vezes é irrecuperável. Coisas simples para outros BD's tornam-se complexas para o Interbase, como alterar o tamanho de um campo sem perda de dados. |
| Paradox | Vem com o Delphi. Muito simples a utilização desse BD. Fácil manutenção. | Corrompe índices com facilidade. Torna-se muito lento para trabalhar em rede. Roda apenas em Windows. |
| MS-Access | Muito fácil a utilização desse BD. O Delphi traz componentes para acessá-lo diretamente sem uso da BDE. É ideal para um pequeno programa, que rode numa rede pequena e não tenha um fluxo de dados muito grande. | É proprietário e o usuário terá de pagar para usá-lo, embora não seja tão caro. Fica lento depois de muitos registros cadastrados. Sobrecarrega a rede, visto que executa suas consulta na máquina cliente e não no servidor. Roda apenas no Windows. |
| PostgreSQL | É dito o Oracle do mundo do software livre. Tem praticamente todos os recursos do Oracle, a vantagem de rodar em diversas plataformas e ser gratuito. | Por ter muitos recursos, acaba se tornando complicado para quem desenvolve sistemas médios, mas com certeza é ideal para grandes sistemas. |
| MySQL | O BD Relacional mais rápido. Confiável. Fácil manutenção. Roda em diversas plataformas. Existem componentes para Delphi que acessam diretamente o BD, sem precisar de BDE ou ODBC. Já existe uma versão com todos os recursos dos BDs maiores. Um dos BDs mais utilizados do mundo. | Quando se fala em MySQL costuma-se relacioná-lo com Web. Mas também é um excelente banco para aplicativos Desktop. |
| Firebird | Quando a Borland abriu o código fonte do Interbase a comunidade criou o Firebird. BD pequeno, rápido e open source. | Poucas funções. Necessário acrescentar Funções Definidas pelo Usuário: UDF's. |

SISTEMA DE GERENCIAMENTO DE ARQUIVOS

Esse é o único sistema que descreve como os dados são armazenados. Nesse modelo cada campo ou item de dado é armazenado seqüencialmente no disco em um único e grande arquivo. Para encontrar um item de dado é necessário verificar cada elemento desde o início do arquivo. A única vantagem desse método é sua simplicidade, já que seu formato muito se parece com o de um arquivo texto onde as palavras se encontram escritas em uma determinada seqüência. Entretanto, por causa dessa simplicidade, muitos dados têm que ser escritos repetidamente gerando uma enorme redundância que dificulta a manutenção e a integridade dos dados. Não há também nenhuma

indicação de relacionamento entre os dados, e o programador precisa saber exatamente como os dados estão armazenados para poder acessá-los de forma consistente. Além disso, sua estrutura física é rígida, o que dificulta alterações na definição dos dados, gerando a necessidade de reconstruir todo o arquivo.

BANCO DE DADOS HIERÁRQUICO

Nesse modelo, os dados são organizados em uma estrutura de árvore que se origina a partir de uma raiz. Essa estrutura identifica as relações “pai-filho” entre os vários itens do banco de dados, mostrando assim suas vantagens sobre o modelo de gerenciamento de arquivos. No modelo hierárquico é possível definir relações “de-um-para-muitos” que facilita e acelera o processo de pesquisa dos dados. Para encontrar uma informação, não é mais necessário percorrer o arquivo inteiro. Basta examinar o item pedido, decompô-lo em componentes e descer pelos ramos necessários até encontrá-lo. Esse método, também facilita a inserção de novos dados, devido aos relacionamentos entre os campos serem feitos através de ponteiros. Para inserir um novo elemento basta alterar os ponteiros dos relacionamentos entre pais e filhos.

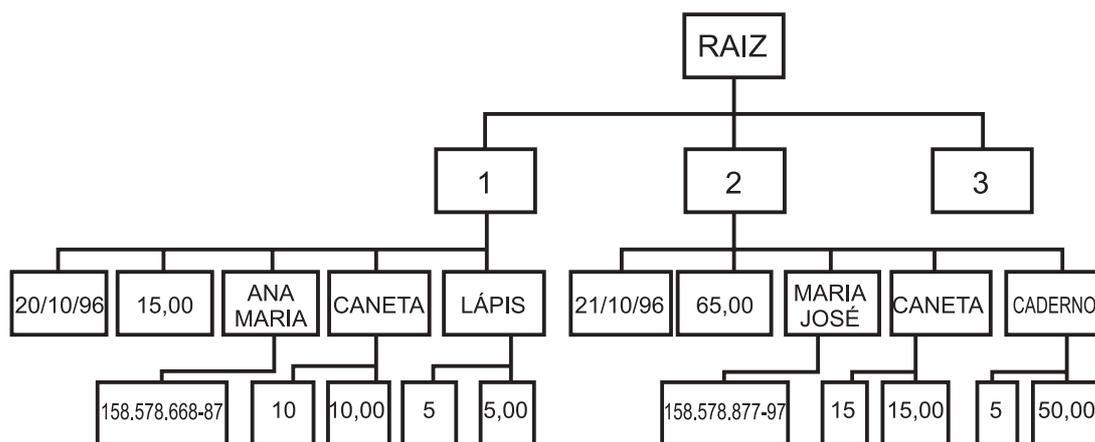


Fig. 7.2 – Modelo de Banco de Dados hierárquico.

Entretanto, esse método ainda possui alguns problemas. Cada nível da árvore é inicialmente definido e qualquer alteração na estrutura desses níveis é uma tarefa difícil. Além disso, o problema da redundância de dados ainda não foi resolvido, já que esse modelo não implementa relações de “muitos-para-muitos”. No exemplo, podemos notar a repetição de algumas informações, o que dificulta a integridade e manutenção dos dados.

BANCO DE DADOS DE REDE

O modelo de rede descreve, conceitualmente, os banco de dados nos quais permitem relações de “muitos-para-muitos” entre os elementos de dados. Desta forma cada item possui um ponteiro para os itens com os quais se relaciona, eliminando assim a necessidade de qualquer tipo de redundância de dados. O grande problema desse modelo é a sua complexidade devido a flexibilidade existente em suas relações. Quando o volume de dados começa a crescer, os relacionamentos entre os itens de dados ficam cada vez mais complexos, tornando sua visualização e entendimento cada vez mais difícil.

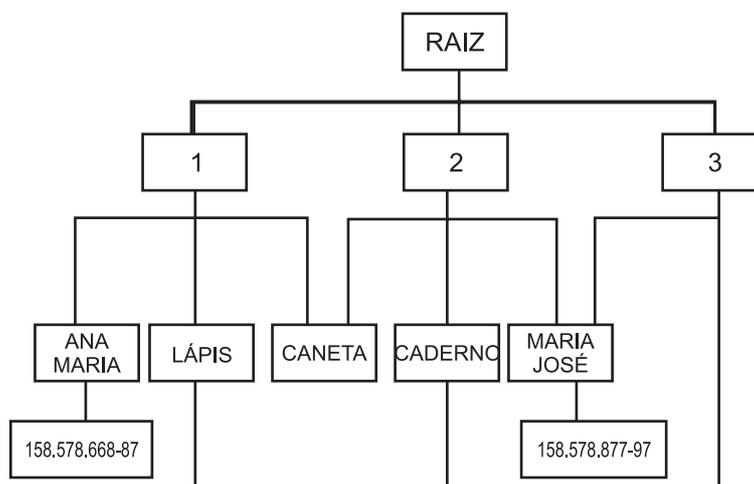


Fig. 7.3 – Banco de Dados de Rede.

BANCO DE DADOS RELACIONAL

Em 1969, foi publicado pelo Dr. E. F. Codd, o primeiro artigo que definia um modelo com base no conceito matemático dos conjuntos relacionais. A partir desse artigo, o modelo relacional tem sido refinado, até que 1985 o próprio Dr. Codd lançou as “12 regras” que definiam um banco de dados relacional. Em 1990, foram publicadas as 333 regras que são subconjuntos e expansões das 12 regras originais.

O modelo relacional abandona o conceito de relações “pai-filho” feitas diretamente entre os dados e os organiza em conjuntos matemáticos lógicos de estrutura tabular. Nesse modelo, cada item de dado pertence a uma coluna da tabela e uma linha da tabela é composta por vários elementos diretamente relacionados.

As tabelas também se relacionam através de funções matemáticas como JOINS e UNIONS. Para fazer esse relacionamento parte dos dados, que identificam unicamente o registro da tabela, são repetidos dentro da outra tabela.

As vantagens desse método são sua simplicidade e flexibilidade nas definições das relações entre os vários itens de dados, já que não são feitos diretamente entre os dados e sim entre as tabelas. Entretanto, esse método não elimina por completo a redundância de dados, já que no mínimo os relacionamentos entre as tabelas são feitos através da repetição de parte dos dados. Além dessa redundância, fica a cargo do projetista do banco de dados se mais repetições de dados irão ou não fazer parte do modelo. O processo de fragmentação dos dados, a fim de serem organizados em subconjuntos (tabelas), é conhecido como normalização.

| Pedido | | | |
|--------|----------|-------|---------|
| NumPed | Data | Valor | Cliente |
| 1 | 20/10/96 | 15,00 | 1 |
| 2 | 21/10/96 | 65,00 | 2 |
| 3 | 22/10/96 | 25,00 | 2 |

| Itens | | | |
|--------|---------|----------|-------|
| NumPed | Produto | Quantid. | Valor |
| 1 | Caneta | 10 | 10,00 |
| 1 | Lápis | 5 | 5,00 |
| 2 | Caneta | 15 | 15,00 |
| 2 | Caderno | 5 | 50,00 |
| 3 | Lápis | 15 | 15,00 |
| 3 | Caderno | 1 | 10,00 |

| Cliente | | |
|---------|----------------|----------------|
| Codigo | Nome | CPF |
| 1 | Ana Maria Lima | 999.876.555-22 |
| 2 | Maria José | 111.111.111-22 |

Fig. 7.4 – Banco de Dados Relacional.

BANCOS DE DADOS RELACIONAIS

Essa lição apresenta algumas características dos bancos de dados relacionais.

Os bancos de dados relacionais vêm se tornando um padrão no mercado, servindo como base de dados para a maioria dos sistemas das empresas. A cada dia, mais ferramentas são construídas para tirar proveito dessa tecnologia, fazendo surgir um número crescente de recursos e produtos a serem oferecidos para os usuários. Pode-se dizer então, que esta é uma tecnologia sólida e consistente e que irá acompanhar o mercado por um longo período de tempo. No entanto, é uma tecnologia que continua evoluindo com o objetivo de disponibilizar as informações para os usuários de maneira eficiente, viabilizando o negócio da corporação e descentralizando as tomadas de decisão.

CLASSIFICAÇÃO

Quanto a capacidade, recursos e facilidade de uso, os banco de dados relacionais podem ser divididos em três categorias: corporativos, departamentais e locais. Em uma mesma empresa todas as categorias podem coexistir e cooperar entre si para juntas formarem uma poderosa base de dados distribuída.

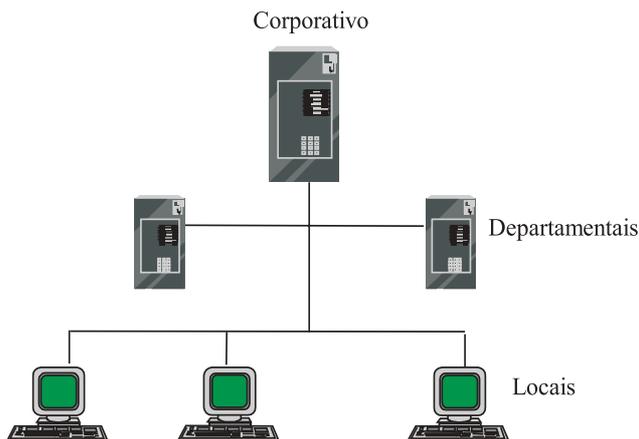


Fig. 7.5 – Classificação dos bancos de dados.

CORPORATIVOS

Sistema de banco de dados para atender toda a corporação. Os principais sistemas dessa categoria são: DB2, Oracle, MS-SQL, Sybase, Informix, Ingres e PostgreSQL. Estes bancos devem ser capazes de armazenar e manipular de maneira eficiente grandes volumes de dados e permitir acesso rápido a um número elevado de usuários conectados ao mesmo tempo. Devido a sua enorme capacidade, os preços desses bancos são também mais elevados e normalmente necessitam de profissionais especializados para configurá-los e monitorá-los no dia a dia.

DEPARTAMENTAIS

Sistemas de banco de dados capazes de atender as requisições de dados ao nível de um departamento. Nesse patamar, existem bancos de dados mais baratos (ou gratuitos) e mais fáceis de configurar, entre eles estão: MySQL, Interbase, SQLBase.

A maioria dos fornecedores de bancos de dados da categoria corporativa está também disponibilizando versões departamentais para disputar esse mercado, trazendo como vantagem a possibilidade de permanecer com o mesmo fornecedor em todos os níveis, tornando o ambiente mais integrado. Por outro lado, as principais vantagens dos bancos desenvolvidos especialmente para essa plataforma são: a facilidade de configuração, o preço dos produtos e a menor requisição de recursos de hardware.

LOCAIS OU MÓVEIS

Sistemas de banco de dados capazes de rodar na mesma máquina que a aplicação. Nessa categoria os principais requisitos são: ser um banco de dados bastante leve que gaste poucos recursos do sistema operacional e da máquina (memória, processamento, etc) e que não necessite de gerenciamento podendo rodar basicamente com sua configuração original.

Nessa categoria podemos citar os bancos MySQL, Interbase, SQLBase, SQLAnywhere, Paradox e MS-Access. Os fornecedores corporativos também estão descendo para esse nível. Porém, para manter os bancos de dados com os mesmos recursos tem sido difícil torná-los leves e fáceis suficientemente. Alguns fornecedores possuem até sistemas de banco de dados diferentes para cada plataforma.

Como esses bancos são instalados na mesma máquina é possível sua utilização em "notebooks". Isso permite as empresas manterem a mesma tecnologia de banco de dados relacional em aplicações móveis, facilitando a integração com a base corporativa da empresa.

Deve-se notar, entretanto, que não está se fazendo uso da arquitetura cliente/servidor nesse caso, já que a mesma máquina é utilizada para rodar o aplicativo e o sistema gerenciador de banco de dados. A principal vantagem dessa utilização sobre a utilização de base de dados de arquivos, é que a última não possui recursos suficientes para manter a integridade dos dados e das transações. Os sistemas gerenciadores de banco de dados são também menos factíveis a corrupção de dados e falhas do que os sistemas baseados em arquivos onde o gerenciamento é feito por cada aplicação. Outra vantagem é prover mais facilidade de integração com as bases departamentais e corporativas, por utilizarem a mesma tecnologia. Entretanto, os SGBDs possuem um custo adicional e exigem uma configuração de máquina melhor devido a um número maior de camadas que terão que ser percorridas para se chegar ao dado.

MODELAGEM DE DADOS

Modelagem de dados é o processo utilizado para definir a estrutura do banco de dados através da distribuição dos dados nas tabelas. Existem maneiras diferentes de se definir o mesmo conjunto de dados, e uma boa modelagem de dados pode facilitar bastante o desenvolvimento das aplicações. Vamos ressaltar aqui, dois pontos que devem ser observados na construção do modelo: o processo de normalização e a propagação de chaves primárias.

NORMALIZAÇÃO

Técnica de análise e organização de dados que visa determinar a melhor composição para uma estrutura de dados. Os principais objetivos dessa técnica são:

- * Eliminar anomalias que dificultam as operações sobre os dados;
- * Minimizar as redundâncias e os conseqüentes riscos de inconsistências;
- * Reduzir e facilitar as manutenções.

Como exemplo, vamos utilizar um modelo, no qual todos os dados foram colocados em uma única tabela através dos seguintes campos:

NumPed, DataPed, ValorPed, NomeCliente, CPFCliente, NomeProduto1, PreçoProduto1, Quantidade1, ValorItem1, NomeProduto2, PreçoProduto2, Quantidade2, ValorItem2, ..., NomeProduto5, PreçoProduto5, Quantidade5, ValorItem5

DER – DIAGRAMA ENTIDADE RELACIONAMENTO – É o diagrama que você vai criar para definir as tabelas e chaves de seu banco de dados, normalizando o mesmo de tal forma que não exista redundância de dados, ou que a redundância seja mínima.

PRIMEIRA FORMA NORMAL

Uma tabela está na primeira forma normal se não possuir grupos de atributos repetitivos.

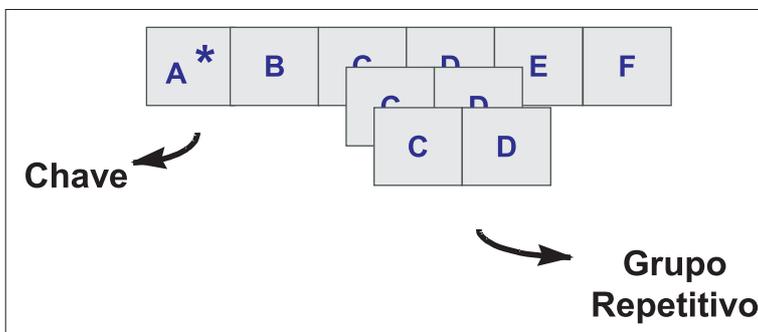


Fig. 7.6 – Presença de grupos repetitivos em uma tabela relacional.

Para resolver esse problema, deve-se remover os grupos repetitivos para uma outra tabela. A chave primária dessa nova tabela pode ser formada através da chave primária da tabela original mais um conjunto de novos atributos.

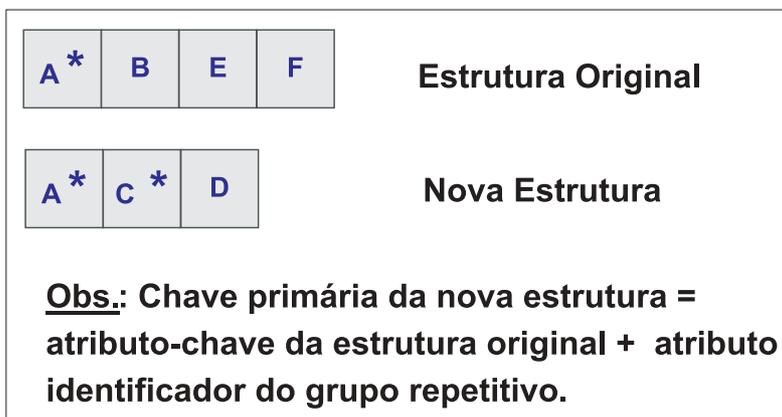


Fig. 7.7 – Eliminação de grupos repetitivos em uma tabela.

Desta forma, elimina-se a deficiência do primeiro modelo que limitava o número de itens de produto que um pedido podia ter. Nessa nova estrutura não há limite para os itens, mas continua-se mantendo a relação que existia entre os itens e o pedido através da inclusão do número do pedido na nova tabela.

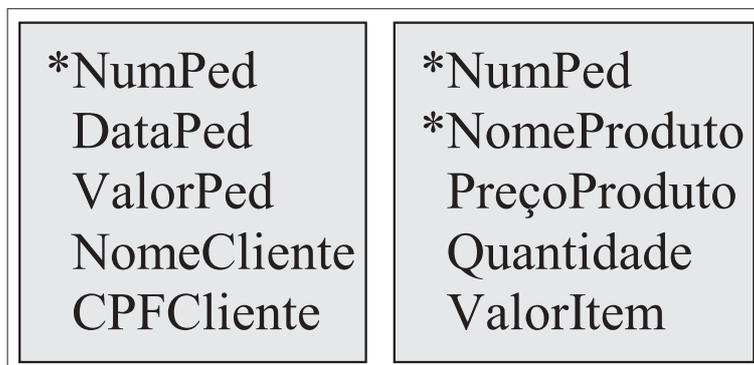


Fig. 7.8 – Exemplo na primeira forma normal.

SEGUNDA FORMA NORMAL

Uma tabela está na segunda forma normal quando sua chave primária não for composta ou quando todos os atributos “não chaves” forem funcionalmente dependentes da chave inteira e esta estiver na primeira forma normal. Um atributo “b” é dependente funcionalmente de “a” se dado o valor de “a”, o valor de “b” puder ser determinado.

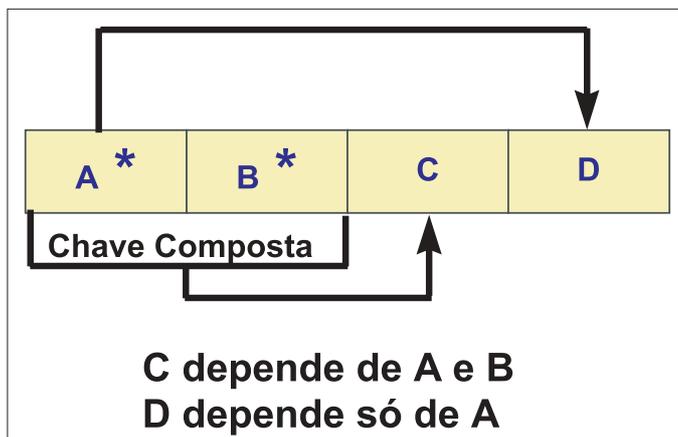


Fig. 7.9 – Modelo na 1FN, mas não em 2FN. Presença de atributos dependentes funcionalmente de apenas parte da chave primária.

Para resolver esse problema, basta remover os atributos que dependem apenas de parte da chave primária para uma nova tabela. A chave primária da nova tabela passa a ser então essa parte da chave primária da tabela original da qual o atributo dependia.

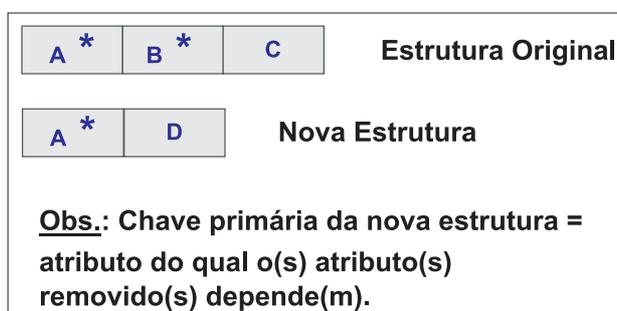


Fig. 7.10 – Reestruturação do modelo na segunda forma normal.

No exemplo de pedido, isso ocorre com a coluna PRECOPRODUTO, que depende apenas do NOMEPRODUTO e não depende de NUMPED. Dado o nome do produto é possível saber seu preço não importando o pedido que o produto pertença. Assim o preço do produto seria repetido em vários registros da estrutura desnecessariamente e dificultando a manutenção e consistência dos dados. Para evitar essa redundância, deve-se criar uma nova estrutura e a chave primária dessa nova tabela seria o NOMEPRODUTO, porém por questões de eficiência pode-se criar uma nova coluna CODPRODUTO de tamanho menor para fazer a ligação entre as tabelas. Além da eficiência, esse procedimento possibilita alterações futuras nos nomes dos produtos, já que esses não são mais usados como chave primária.

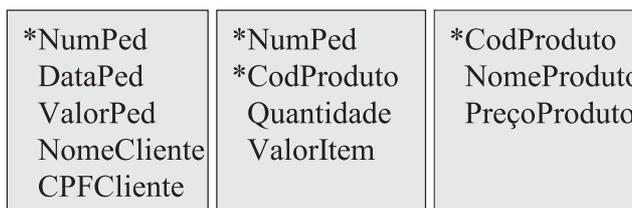


Fig. 7.11 – Exemplo na segunda forma normal.

TERCEIRA FORMA NORMAL

Uma tabela está na terceira forma normal quando está na segunda forma normal e não possui nenhum atributo “não chave” dependendo funcionalmente de outro atributo “não chave”.

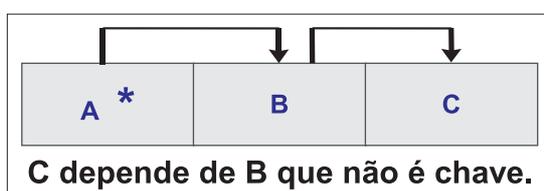


Fig. 7.12 – Presença de atributos dependentes funcionalmente de outros atributos não-chave.

Para resolver esse problema, basta remover os atributos que dependem de outros atributos não chave para uma nova tabela. A chave primária da nova tabela passa a ser então o atributo não chave que possui dependentes.

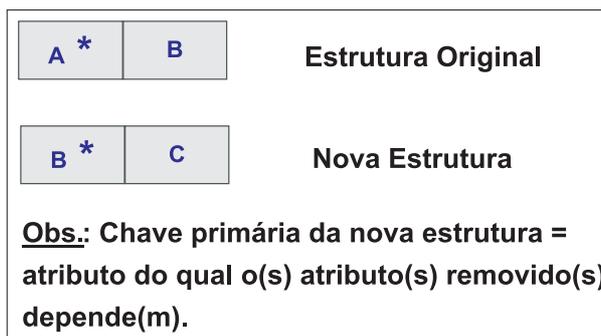


Fig. 7.13 – Reestruturação do modelo na terceira forma normal.

No exemplo apresentado, há uma dependência funcional do CPFCLIENTE com o NOMECLIENTE, ou vice-versa já que ambos são candidatos a definir unicamente um cliente. Dado o valor de NOMECLIENTE é possível determinar o valor de CPFCLIENTE, portanto esse não depende da chave primária NUMPED. Para eliminar tal redundância, deve-se criar uma outra estrutura e colocar o modelo na terceira forma normal.

| | |
|---|--|
| Pedido | Cliente |
| *NumPed DataPed ValorPed CodCliente | *CodCliente NomeCliente CPFCliente |
| ItemPedido | Produto |
| *NumPed *CodProduto Quantidade ValorItem | *CodProduto NomeProduto PreçoProduto |

Fig. 7.14 – Exemplo na terceira forma normal.

Existem outras formas normais, entretanto, colocando-se o modelo na terceira forma normal já é possível tratar os dados de forma consistente e segura provendo uma certa facilidade no desenvolvimento.

PROPAGAÇÃO DE CHAVES PRIMÁRIAS

Outro processo que deve ser bem definido é quanto a forma de propagação de chaves entre as tabelas. No exemplo anterior pode-se observar que a chave primária da tabela Item foi formada pela propagação da chave primária das tabelas Pedido e Produto.

| |
|---|
| ItemPedido |
| *NumPed *CodProduto Quantidade ValorItem |

Fig. 7.15 – Tabela de Itens. Chave primária composta pela propagação das chaves primárias de outras tabelas.

Essa técnica de propagar as chaves primárias tornando-as também chaves primárias na tabela destino pode facilitar consultas diminuindo o número de tabelas utilizadas no comando. Imagine uma tabela de históricos de itens que armazenasse cada atualização que ocorresse em cada item. Seria fácil então consultar os históricos de um determinado pedido ou produto, sem a necessidade de utilizar uma outra tabela, já que essa tabela possui os campos NUMPED e CODPRODUTO.

Entretanto, esse método pode tornar mais difícil a construção das aplicações de inclusão e manutenção dos dados (Fig. 7.16).

Uma outra alternativa para montar a chave primária da tabela de itens é criar uma outra coluna com um número seqüencial do sistema e substituir a coluna CODPRODUTO. Desta forma o campo CODPRODUTO seria utilizado apenas para se fazer o relacionamento entre as tabelas (Fig. 7.17).

| |
|---------------------------------|
| Hist Itens |
| *NumPed *CodProduto *Data |

Fig. 7.16 – Tabela de Hist Itens.

Assim, definindo novamente a tabela de histórico de itens ficaria como na figura 7.18.

Percebe-se, agora, que não há informação na tabela de histórico de itens sobre o produto. Para consultar os registros de um determinado produto é necessário fazer um "join" (junção) com a tabela de Itens, dificultando a consulta, mas facilitando a construção das telas de manutenção. Além disso, esta alternativa permite que o usuário altere o produto de um item já que esse não é mais parte da chave primária.

Concluindo, não existe uma melhor forma de se definir as chaves primárias. Existe a melhor alternativa para um determinado caso.

Fig. 7.17 – Tabela de Itens com chave primária alternativa

Hist Itens
*NumPed
*NumSeq
*Data

Fig. 4.19 – Tabela de Hist. de Itens.

EXERCÍCIOS

01) Controle de Cinemas - Estudo de Caso

Você deve projetar um modelo de dados que atenda às necessidades de controle dos cinemas e filmes em uma determinada empresa de distribuição de filmes.

Visão do negócio:

- Área de Negócio: Departamento de Programação;
- Função Gerencial: Administração de Cinemas.

Após várias reuniões com os futuros usuários do sistema, relacionamos uma série de regras do negócio e que serão a base para o desenvolvimento do diagrama ER.

- A empresa de distribuição possui vários cinemas, em diversas localidades;
- Cada cinema possui uma identificação única, um nome fantasia, um endereço completo, incluindo rua, avenida, bairro, município, estado e sua capacidade de lotação;
- Os filmes podem ser dos mais variados tipos e gêneros;
- Cada filme é registrado com um título original, e se for filme estrangeiro, possuirá também o título em português, o gênero, sua duração e seu país de origem, informações sobre os atores que compõem seu elenco, e o seu diretor. Existirá um único diretor para cada filme;
- Alguns cinemas apresentam mais de um filme em cartaz, sendo nestes casos, sessões alternadas com um filme e outro;
- As sessões possuem horários que variam de acordo com a duração do filme, havendo sempre um intervalo de aproximadamente 15 minutos entre elas;
- Os atores de um filme podem, obviamente, atuar em diversos filmes, assim como o diretor de um filme pode também ser ator neste filme, ou ainda mais, ser ator em outro filme. Um ator possui as seguintes características: um número de identificação, um nome, uma nacionalidade e uma idade;
- As sessões de cinema devem ter seu público registrado diariamente, para que se permita a totalização dos assistentes quando o filme sair de cartaz, ou a qualquer instante;

Nas reuniões de levantamento, os usuários nos passaram as seguintes necessidades de informação:

- Apuração do público por município, por cinema, e por sessão de cada cinema;
- Permitir uma forma de consulta, que dado um determinado ator, sejam localizados os cinemas onde estão em cartaz os filmes em que este ator atua;
- Em quais cinemas está sendo exibido um determinado gênero de filme;
- Em quais cinemas estão sendo exibidos filmes nacionais.

Veja o Conteúdo Exclusivo do Curso Delphi Avançado



1 CD - 130 Minutos

- * Compreender o modelo Entidade-Relacionamento e Lógico-Relacional;
- * Saber definir e identificar: Entidades, Relacionamentos, Atributos e tudo mais que tenha a ver com o Modelo Entidade-Relacionamento;
- * Conhecer os tipos de relacionamentos: Condicionais e Incondicionais;
- * Saber os graus de relacionamento: Um-Para-Um, Um-Para-Muitos, Muitos-Para-Muitos;
- * Compreender uma realidade e a partir dessa compreensão criar um modelo de dados e logo depois o banco de dados em si;
- * Dominar o conceito de chaves, índices e integridade referencial;
- * Saber normalizar um modelo através das Formas Normais (1FN a 5FN);
- * Usar programas para a criação do modelo de dados;
- * Utilização do DBDesigner.



1 CD - 159 Minutos

- * Definição e História da Linguagem SQL;
- * Entendendo as partes da Linguagem: DDL, DML, DCL e Transactions Control;
- * Instrução Select (alias, order by, like, asc, desc, group by, having, where, not, in, exists, sum, max, min, avg, count, between, distinct, etc);
- * Instrução Insert Into;
- * Instrução Update;
- * Instrução Delete;
- * Consultas Encadeadas (SubQueries);
- * Junções (inner join, left join, right join e full outer join);
- * Criação, alteração e deleção de tabelas e índices;
- * Controle de Transações;
- * Diversos exemplos feitos no programa SQL Explorer;
- * Criação de um sistema no Delphi implementando os comandos SQL juntamente com o componente Query. Cadastro de Clientes, Cadastro de

Usuarios e Formulário Mestre Detalhe com duas tabelas (CVenda e DVenda).



2 CDs - 340 Minutos

- * Introdução: Definição, História, Características, Licenças e Versões;
- * Instalação da versão 3.23 no Windows 98, tipos de arquivos, estrutura de diretórios do MySQL, uso do Console (prompt), etc;
- * Utilização da ferramenta gráfica MySQLFront 2.3, criação de bancos de dados, tabelas, explicação dos tipos de campos, importação de dados do paradox, atributos especiais de campos do MySQL, uso do Manual de Referência, uso das funções do MySQL;
- * Instalação da Versão 4 do MySQL no Windows XP, uso do MySQLFront 3.2, importação de arquivos do Excel, Exportação de dados para o formato HTML, etc;
- * Instalação do MySQL no Linux Slacware, uso do console no Linux, uso da ferramenta gráfica PHPMyAdmin;
- * Instalação e configuração da versão 5 no Windows XP;
- * Uso da ferramenta gráfica DBManager, tipos de tabelas, foreign keys, constraints, views, backup e restauração de dados, controle de transações;

- * Uso da ferramenta gráfica MySQL Administrator, controle de usuários, backup e restauração de dados, replicação;
- * Uso da ferramenta gráfica MySQL Query Browser;
- * Uso da ferramenta gráfica MySQL Migration Toolkit, migração de um banco de dados do MS-Access;
- * DBDesigner - Sincronização de dados e engenharia reversa;
- * Apresentação dos sites oficiais do MySQL - Internacional e Nacional;
- * Stored Procedures e Triggers.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 254 Minutos

- * Introdução: Definição, História, Características, Licenças e Especificações Firebird;
- * Comparativo das arquiteturas Classic x SuperServer;
- * Descrição dos tipos de dados suportados pelo Firebird;
- * Dialeto;
- * Tipos de Transações;
- * Instalação passo a passo da versão 1.5 do Firebird no Windows;
- * Explicação da estrutura de diretórios;
- * Serviços executados no Windows e opções no Painel de Controle;
- * Utilização das seguintes ferramentas gráficas: IBExpert, IBEasy+, FlameRobin, IBAccess, SQLHammer, DBak Win, IB First AID, Interbase Grant Manager;
- * Criação de tabelas, triggers, generatos, domains, views, queries;
- * Controle de acesso para usuários;
- * Backup do BD;

- * Constraints: Check, PK, FK, Unique;
- * Corrigindo BDs corrompidos;
- * Criação de um projeto de BD;
- * Utilização do ISQL (console);
- * Criação de triggers para controle de estoque;
- * Exceptions;
- * Criação de Stored Procedures;
- * UDFs - Conceito; Criação de uma UDF no Delphi e uso no Firebird;
- * Corrupção de Dados: Conhecendo, evitando, corrigindo;
- * Utilização do Gfix (console);
- * Comparativo: MySQL x Firebird;
- * Apresentação dos sites importantes internacionais e nacionais;
- * Instalação do Firebird Cliente numa máquina virtual Windows 98;
- * Conexão com o banco de dados através da rede;
- * Apresentação da documentação.



1 CD - 116 Minutos

- * Definição e estrutura das empresas;
- * Níveis de decisão das empresas;
- * O Analista de Sistemas;
- * Engenharia de Software;
- * Ciclo de vida de um sistema;
- * Coleta de Informações: Entrevistas e pesquisas;
- * Entrevistas: planejamento, preparação, comportamento, linguagem, fatos x opiniões, desejos dos usuários, observações gerais;
- * Contratos e lei: análise das principais cláusulas de um contrato (identificação das partes, documentação, prazo de entrega, códigos fontes, confiabilidade);
- * Lei do Software - comentários sobre os principais artigos;
- * O que é melhor: autônomo ou empresa;
- * Estudo de caso: Sistema para Hotel - apresentação de um estudo de caso mostrando desde o início um problema enfrentado por um hotel e a

solução adotada por um analista de sistemas. Neste estudo são mostrados e comentados a proposta e o contrato utilizado pelo analista;

- * Apresentação de recibos: simples e RPA;
- * Vantagens do aluguel de software;
- * A carreira do consultor independente;
- * Comentários acerca do Guia do Empreendedor e do manual A Pequena Empresa e o Novo Código Civil;
- * Prosperidade - Mapas Mentais - explanação a apresentação do programa freemind.

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 208 Minutos

- * Compreensão da realidade do SisCom;
- * Criação das tabelas (Produto - Cliente - Fornecedor - Funcionario - Banco - Departamento - Cartao - Unidade - Plano_Conta - Tipo_Pgto - CFOP - Pagamento - Recebimento - C_NFE - D_NFE - C_Venda - D_Venda - C_Requisicao - D_Requisicao - C_Cotacao - D_Cotacao - C_Pedido - D_Pedido - Configuracao - Nivel_Acesso);
- * Comentário detalhado sobre cada tabela e seus campos;
- * Apresentação do DER do SisCom e comentários sobre cada relacionamento;
- * Criação das tabelas e dos relacionamentos no DBDesigner;
- * Sincronização do DBDesigner com o MySQL;
- * Migrando o BD do MySQL para o Firebird;
- * Conectando o DBDesigner com o Firebird através de ODBC;
- * Conectando o DBDesigner com o Firebird diretamente;
- * Fazendo Engenharia Reversa de tabelas do Firebird no DBDesigner;

- * Visualizando a estrutura de arquivos do DBDesigner para fazê-lo conectar-se ao Firebird;
- * Visualizando o Código Fonte do DBDesigner dentro do Delphi;
- * Criação passo a passo do BD no Firebird utilizando o IBExpert;
- * Padronização de campos e tabelas;
- * Erros encontrados por falta de padronização.



2 CDs - 416 Minutos

- * Apresentação da Suite de Componentes ZeosDBO;
- * Vantagens de usar o ZeosDBO;
- * Instalação do ZeosDBO;
- * Explicação sobre cada componente do ZeosDBO;
- * Comparação dos componentes do ZeosDBO com os seus equivalentes no BDE;
- * Diferença entre sistemas MDI e SDI e explicação sobre qual o melhor;
- * Início do desenvolvimento do sistema SisCom;
- * Criação do Formulário Principal: Menu, Barra de Ferramentas, Barra de Status. Todos os hints do sistema serão enviados para a barra de status, possibilitando uma ajuda online através de dicas rápidas para os usuários;
- * Carregando um logotipo para a tela principal do sistema;
- * Uso de Application.MessageBox;
- * Componentes de acesso ao banco em DataModule;
- * Criação dos cadastros utilizando ZTable (equivalente a TTable). Para

quem vem do paradox ficará muito fácil desenvolver dessa forma;

- * Criação dos cadastros utilizando ZQuery (Equivalente a TQuery). Utilização de Edits Simples no lugar dos DBEdits, com procedimentos específicos para o controle de edição e gravação dos campos;
- * Utilização de comandos dos tipos TTable (Append, Edit, Post, Delete) e cadastros usando apenas SQL puro (Insert Into, Update Set, Delete From);
- * O sistema roda AUTOMATICAMENTE em Firebird e MySQL. O usuário escolhe qual banco usar. Não existe alteração nenhuma no código;
- * Utilização dos Generators (firebird) para autonumerar campos no sistema;
- * Implementação de uma rotina onde o usuário procura por qualquer informação em qualquer campo do sistema, bastando para isso clicar no título do DBGrid;
- * Criação do nível de acesso para controle de acessos dos usuários aos módulos do sistema;
- * Verificação e análise de várias mensagens de erro;
- * Implementação de um controle de auditoria, onde o sistema armazena qual o usuário que inseriu ou alterou registros, juntamente com a data e hora.

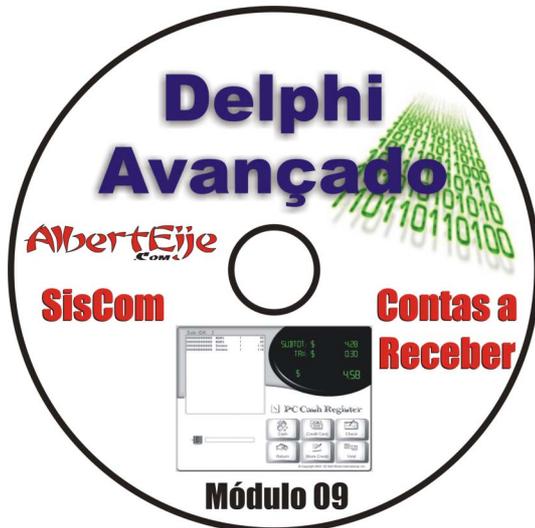
Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 236 Minutos

- * Introdução a tecnologia dbExpress;
- * Vantagens e desvantagens do dbExpress;
- * Comparativo dbExpress x BDE;
- * Apresentação de todos os componentes da paleta dbExpress;
- * Utilização do SQLQuery + DataSetProvider + ClientDataSet para criar o Contas a Pagar;
- * Desenvolvimento do Contas a Pagar passo a passo;
- * Acessando Firebird e MySQL ao mesmo tempo através do SQLConnection;
- * Utilizando um método de procura ao clicar no título do DBGrid, tendo por base um período passado por 2 MaskEdits;
- * Pegando o código AutoIncrement no MySQL para o ClientDataSet;
- * Pegando o código do Generator no Firebird para o ClientDataSet (Stored Procedures);
- * Explicação do uso de ActionLists para otimizar o trabalho;

* O SisCom funciona automaticamente com MySQL e Firebird. Os módulos de cadastro estão utilizando a tecnologia Zeos. O módulo de Contas a Pagar está usando a tecnologia dbExpress + ClientDataSet.



1 CD - 140 Minutos

- * Desenvolvimento do Contas a Receber do SisCom utilizando a tecnologia dbExpress + ClientDataSet;
- * Implementação do conceito de reconciliação de erros no SisCom;
- * Tradução do Formulário de Reconciliação de Erros para o Português;
- * Utilização apenas do SQLConnection para realizar consultas SQL;
- * Explicação detalhada do uso do SQLDataSet (Table, Query, StoredProc);
- * Utilização de parâmetros para realizar consultas SQL;
- * Manipulação dos dados do ClientDataSet antes de serem enviados para o Banco de Dados - ainda na memória do micro cliente (utilização da propriedade Delta do ClientDataSet);
- * Criação de consultas Mestre/Detalhe. (Dados mostrados numa mesma DBGrid e posteriormente em duas DBGrids);
- * Criação de bancos de dados locais com o ClientDataSet (utilização de arquivos XML);

- * Criação de Campos Agregados;
- * Utilização do componente SimpleDataSet;
- * Utilização do componente SQLMonitor para mapear o que está ocorrendo entre a aplicação e o banco de dados (tudo que está ocorrendo em baixo nível é mostrado num Memo);
- * Implementando o Controle de Transações com o dbExpress;
- * Explicação sobre o nível de isolamento (TransIsolationLevel) das transações do dbExpress.

Após assistir às vídeo aulas do curso você estará apto para desenvolver qualquer sistema comercial. Este conteúdo você não encontra em nenhum curso presencial ou online. Conteúdo único e exclusivo.

Curso Delphi Avançado - 15 Módulos (19 CDs)



2 CDs - 447 Minutos

- * Apresentação de cada componente da paleta ADO - principais propriedades;
- * Criando uma conexão no SisCom para Firebird e MySQL (alternando entre um e outro de acordo com a escolha do usuário);
- * Instalação dos Drivers ODBC para MySQL e Firebird e criação de DSN para acesso ao banco de dados do SisCom;
- * Criação do Módulo Requisição de Compras;
- * Utilização do ADOConnection, ADOQuery, ADOTable e ADOSToredProc;
- * Análise de erros ao construir o módulo;
- * Construção do formulário para importar itens para a requisição;
- * Trabalhando com passagem de parâmetros;
- * Localizando e eliminando erros do projeto do BD;
- * Criação do Módulo Cotação de Compras;
- * Utilização do repositório de formulários do Delphi;
- * Utilização de um componente permite abrir um menu em um botão;

- * Construção do formulário para importar os itens das requisições que vão compor a cotação;
- * Construção do Módulo Confirmação da Cotação (neste formulário o usuário insere os preços e condições enviadas pelos fornecedores);
- * Construção do Módulo Pedido de Compras (neste formulário o usuário visualiza as cotações e pode fazer pedidos aos fornecedores baseado nas informações prestadas pelos mesmos. É permitido realizar vários pedidos através de uma cotação);
- * Construção do formulário para importar os itens das cotações.



1 CD - 165 Minutos

- * Construção do Sistema de Vendas utilizando os componentes da paleta ADO (ADOConnection, ADOQuery, ADOTable, ADOSToredProc);
- * Chamadas a Stored Procedures armazenadas no BD com passagem e retorno de parâmetros;
- * Importação dos dados da tabela oficial CFOP de um BD diferente;
- * Análise de alguns erros que ocorrem com ADO;
- * Usando ADO com ClientDataSet;
- * Acessando o BD sem o uso do componente ADOConnection;
- * Construindo consultas Mestre-Detalhe com ADO (3 tabelas no exemplo);
- * ADO e Controle de Transações;
- * Enviando informações para o BD sem Result Set com o componente ADOCommand.



1 CD - 188 Minutos

- * Descrição detalhada dos 13 Componentes da paleta IBX (Interbase Express);
- * Comentários sobre suas propriedades;
- * Construção do Formulário de Entrada de Notas Fiscais (IBDataBase, IBTransaction, IBDataSet, IBQuery, IBUpdateSQL);
- * Passagem de parâmetros;
- * Atualização do Estoque via código feito no Delphi;
- * Atualização do Estoque via Triggers criadas no Firebird (AfterInsert, AfterUpdate, AfterDelete);
- * Construção do Formulário para Ajuste de Preços;
- * Código para ajustar preços na tabela de produtos utilizando uma taxa (aumentando e diminuindo valores).

Curso Delphi Avançado - 15 Módulos (19 CDs)



1 CD - 198 Minutos

- * Instalação do Quick Report no Delphi 7;
- * Descrição detalhada dos componentes da Suite Quick Report;
- * Desenvolvimento dos relatórios de cadastro do SisCom utilizando o Quick Report;
- * Criação de relatórios Mestre-Detalhe com o Quick Report;
- * Exportando relatórios para TXT e HTML através do Quick Report;
- * Descrição detalhada dos componentes da Suite Rave Reports;
- * Desenvolvimento dos relatórios de Contas a Pagar e a Receber utilizando o Rave Reports;
- * Criação de relatórios Mestre-Detalhe com o Rave Reports;
- * Utilização de imagens e códigos de barras usando o Rave Reports;
- * Exportando relatórios para PDF através do Rave Reports;
- * Apresentação do Componente PrintFast (componente que envia o relatório diretamente para a impressora - excelente para impressoras matriciais e Clippeiros de plantão);
- * Instalação do PrintFast no Delphi 7;

* Desenvolvimento dos seguintes relatórios usando o PrintFast: Requisição de Compras, Mapa Comparativo de Preços e Pedido de Compras;

* Apresentação dos Geradores de Relatórios: FreeReport e Fortes Reports.



1 CD - 168 Minutos

- * Análise de 10 Manuais em formato PDF;
- * Criação do Manual do SisCom em formato PDF;
- * Criação de um arquivo HLP (ajuda) utilizando ferramentas gratuitas;
- * Chamando a ajuda de dentro do SisCom ao pressionar a tecla F1 (ajuda sensível ao contexto);
- * Desenvolvimento de um manual HTML para o SisCom;
- * Convertendo arquivos HTML para CHM (HTML Help);
- * Convertendo arquivos CHM para 42 outros formatos (PDF, HLP, RTF, TXT, XLS, DOC e muitos outros);
- * Utilização de uma ferramenta que cria o Help e já salva em diversos formatos diferentes (HLP, CHM, PDF e outros). Ferramenta muito fácil de utilizar;
- * Chamando a ajuda CHM de dentro de uma aplicação Delphi ao pressionar F1 (ajuda sensível ao contexto);
- * Utilização de uma ferramenta que cria o arquivo de ajuda baseado nas

imagens capturadas das telas do SisCom. O usuário clica em qualquer parte das janelas capturadas e a ajuda referente àquela parte da janela é apresentada.



1 CD - 175 Minutos

- * Modificando o código do SisCom para que o mesmo funcione em rede;
- * Testando o funcionamento do SisCom em rede (máquina virtual);
- * Identificação das bibliotecas necessárias para a instalação numa máquina cliente;
- * Trabalhando com arquivos Ini (configuração do acesso aos dados através desses arquivos);
- * Desenvolvimento de um instalador personalizado no Delphi (o instalador se encarrega de copiar todos os arquivos necessários para o funcionamento do SisCom e instala o MySQL e o Firebird);
- * Utilização de ferramentas para criação de instaladores: Setup2Go, CreateInstall, InnoSetup);
- * Compactação de executáveis;
- * Criando uma biblioteca de funções e integrando a mesma ao SisCom;
- * Aperfeiçoamento da janela de Login;
- * Desenvolvimento de uma Splash Screen para o SisCom utilizando uma

imagem feita no CorelDraw e exportada para o SisCom. Utilização de uma barra de progresso.

Curso Delphi Avançado - Módulos Plus



2 CDs - 224 Minutos

- * Introdução a Orientação a Objetos
- * Histórico do Paradigma de Orientação a Objetos
- * Explicações sobre: Abstração; Objeto; Encapsulamento; Mensagem; Classe; Herança; Polimorfismo; Classificação; Associação; Agregação; Generalização; Especialização
- * Introdução a UML - Unified Modeling Language
- * Histórico da notação UML
- * Fases do desenvolvimento de um sistema em UML: Análise de requisitos; Análise; Design (projeto); Programação; Testes
- * Visões: Visão de Componentes; Visão Lógica; Visão de Use-Case; Visão de Organização; Visão de Concorrência
- * Modelos de Elementos: Classes; Objetos; Estado; Pacote; Componente; Relacionamentos (associação, generalização, dependência e refinamento)
- * Diagramas: Use-Case; Classes; Objetos; Estado; Seqüência;

Colaboração; Atividade; Componente; Execução

- * Estudos de caso: conta corrente e aplicações financeiras; locadora; sistema de matrícula; posto de gasolina; biblioteca.
- * Cada estudo de caso acima é feito em uma ferramenta diferente.

Obs: Este Módulo é pré-requisito para o Módulo Delphi OOP



1 CD - 220 Minutos

- * Introdução à Linguagem Object Pascal (Delphi Language);
- * Variáveis - declaração e exemplos; Variáveis Locais e Globais;
- * Constantes - declaração e exemplos;
- * Tipos de Dados - Inteiros, Reais, Texto, Ordinais;
- * Rotinas de conversão de tipos; TypeCasting;
- * Operadores - unários, multiplicativos, direção de bits, aditivos, relacionais;
- * Estruturas de Repetição - While, For, Repeat; Quebras de Laço (continue, break, exit, halt);
- * Tipos Definidos pelo Usuários - SubRange, Enumerações, Ponteiros, Records, Arrays, Sets;
- * Procedimentos, Funções e Métodos;
- * Passagem de parâmetros - explicações sobre passagem por valor e por referência; Uso do With;
- * Desenvolvimento de diversas rotinas para praticar os conceitos acima;
- * Explicações sobre a estrutura da Unit: Áreas: Unit, Interface,

Implementation, Var, Const, Uses, Referência Circular;

- * Sobrecarga de métodos: Overload - Exemplo de vários métodos de mesmo nome para mostrar o funcionamento;
- * Blocos protegidos - explicações sobre exceções (try, except, finally, raise);
- * Explicações sobre parâmetros formais e parâmetros reais;
- * Classes e Objetos; Desenvolvimento de diversas classes no Delphi;
- * Métodos construtores e destrutores; Métodos Getters e Setters;
- * Instanciação das classes - Desenvolvimento de uma aplicação Console para analisar o funcionamento;
- * Encapsulamento - Operadores de visibilidade (private, public, protected, published) - explicações e prática de cada um deles;
- * Herança entre classes; * Utilização de Property (propriedades);
- * Sobreposição de Métodos - Virtual/Override - Sobrescrever métodos em classes descendentes - Conceito de Polimorfismo;
- * Desenvolvimento do SisCom OO - explicações sobre o objetivo do SisCom (Sistema Comercial);
- * Exibição do DER e da UML do SisCom. Diferenças entre os dois diagramas;
- * Apresentação do SisCom desenvolvido durante o Curso Delphi Avançado;
- * Desenvolvimento passo a passo do Formulário Padrão de Cadastro. Através desse formulário, todos os outros forms de cadastro serão criados. Utilização do conceito de Herança Visual do Delphi;
- * Integração Delphi / ModelMaker; * Engenharia reversa do SisCom para o ModelMaker;
- * Alterações em modelo no ModelMaker e atualização em Tempo Real na aplicação no Delphi;
- * Explicações para desenvolvimento dos diagramas UML do SisCom no ModelMaker;

Obs: O Módulo Plus Entendendo UML é pré-requisito para este módulo

Curso Delphi Avançado - Módulos Plus



2 CDs - 339 Minutos

- * Explicações sobre ECF (Emissor de Cupom Fiscal) e TEF (Transferência Eletrônica de Fundos);
- * Instalação e utilização de Emulador de Impressora Fiscal para utilização junto com o sistema de Frente de Caixa;
- * Análise dos manuais das Impressoras Bematech, Yanco, Daruma e Zanthus;
- * Criação passo a passo da Interface do Sistema de Frente de Caixa;
- * Implementação dos comandos para utilização com a impressora fiscal: Leitura X, Redução Z, Relatórios de Memória Fiscal; Abertura e Fechamento de Cupons; Envio de Formas de Pagamento; Envio e Cancelamento de Itens; Etc;
- * Criação de um formulário de configurações onde o próprio usuário define as formas de pagamentos que vai utilizar, em quais ocasiões vai solicitar senhas, qual impressora está utilizando e o caminho onde as fotos dos produtos estão armazenadas (as fotos dos produtos são armazenadas

fora do banco de dados);

- * Criação de um formulário para localização de produtos, caso o sistema não consiga ler o código ou o produto não tenha código de barras;
- * Criação de um formulário para armazenar os dados da venda;
- * Estudo e implementação da Solução TEF;



2 CDs - 275 Minutos

- * SQL Server - Apresentação - Características;
- * Instalação passo a passo do SQL Server e Ferramentas Avançadas - Instalação do .NET Framework;
- * Apresentação da ferramenta de configuração (SQL Server Configuration Manager);
- * Apresentação e configuração do SQL Browser - ferramenta para aceitar conexões remotas;
- * Explicação e configuração dos tipos de conexões (Shared Memory, Named Pipes, TCP/IP, VIA);
- * Criação de aliases para o Banco de Dados;
- * Apresentação da ferramenta de configuração (SQL Server Surface Area Configuration);
- * Configuração e uso da ferramenta para gerenciamento de bancos de dados feitos em SQL Server (Microsoft SQL Server Management Studio Express);

- * Explicação sobre Collations; Detalhamento sobre os tipos de dados;
- * Criação e configuração de um banco de dados;
- * Criação das tabelas, índices, relacionamentos, constraints, etc do banco de dados utilizando ferramenta para modelagem;
- * Criação do campo Auto-Increment; Criação de Views atualizáveis;
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Manager 2005 for SQL Server);
- * Criação de Checks Constraints; Criação de Usuários/Roles para acesso ao banco;
- * Confecção de relatórios utilizando a ferramenta;
- * Exportando dados (Excel, Access, Word, RTF, HTML, PDF, TXT, CSV, DBF, XML);
- * Importando dados (Excel, Access, DBF, XML, TXT, CSV);
- * Explicação sobre Stored Procedures e Triggers;
- * Operações com o Banco (extração de dados, backup e restore);
- * Apresentação de ferramenta gráfica adicional para gerenciamento do banco de dados (SQL Lite);
- * Conhecendo as funções de sistema (agregação, matemáticas, data e hora, string, diversas...);
- * Desenvolvimento de uma aplicação do Delphi para acesso ao banco de dados (cadastros, vendas, etc). Utilização das tecnologias de acesso ADO e dbExpress;
- * Formulário mestre-detelhe e campos de lookup; Explicação e criação de Trigger para atualização do estoque;
- * Utilização de transações (início, cancelamento e confirmação);
- * Funcionamento da aplicação em rede através de máquina virtual W98;

Curso Delphi Avançado - Módulos Plus

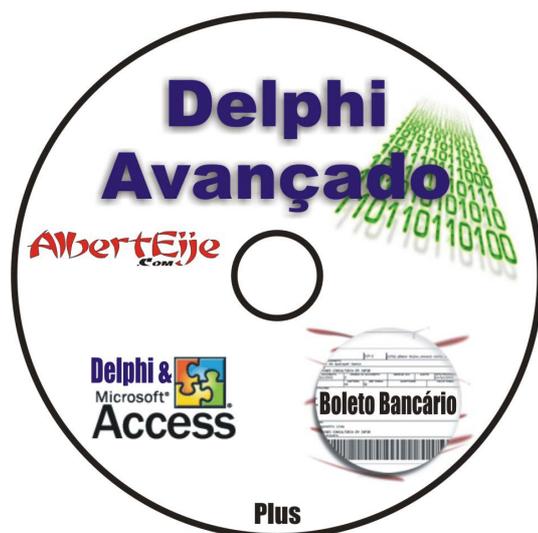


1 CD - 129 Minutos

- * Introdução a Criptologia (Criptografia e Criptoanálise);
- * Utilizando na prática a Esteganografia (esconder arquivos em imagens);
- * Análise de vários componentes que implementam a Criptografia no Delphi;
- * Apresentação de vários métodos para Proteção de Programas (Nag-Screen, Período, Limitação de Registros, HardLocks, Identidade Única);
- * Implementando na prática várias técnicas para proteção de programas (Por Arquivo, Por Diretório, Por Registro, etc);
- * Desenvolvimento de um sistema que bloqueia o uso do sistema no usuário. O sistema grava as informações no registro do Windows. Pega informações da máquina do usuário (HD, BIOS) e combina tais informações com Constantes implementadas pelo programador (Phi e Pi). O desbloqueio do sistema se dá ao informar a Contra-Chave (Demonstrado passo a passo);
- * Apresentação de vários componentes para Delphi que implementam a

proteção de programas;

- * Dicas importantes para colocar em prática no momento da criação de uma proteção;
- * Apresentação de Engenharia Reversa (Descompiladores);
- * Utilização na prática de alguns descompiladores (DeDe, Source Rescuer, ResHacker);
- * Demonstração de técnica para evitar a Engenharia Reversa de programas feitos em Delphi;
- * Demonstração de como Crackear um programa feito em Delphi utilizando um Editor HexaDecimal (passo a passo);



1 CD - 229 Minutos

- * MSAccess - Apresentação;
- * Criação de Tabelas (explicação dos tipos de dados);
- * Especificações do Access - limitações do banco de dados;
- * Definindo relacionamentos e criando integridade referencial. Explicações sobre cardinalidade;
- * Criação de consultas (Visual e SQL);
- * Criação de formulários. Formulários prontos para impressão. Formulários vinculados a código em VBA. Subformulários;
- * Criação de uma consulta vinculada a dados de um formulário. Visualização dos dados de uma consulta em um subformulário;
- * Desenvolvimento de relatórios;
- * Desenvolvimento de uma página WEB que acessa e altera os dados da tabela;
- * Introdução a VBA (Visual Basic for Applications);
- * Acessando o banco de dados através do Delphi (tecnologia DAO);
- * Criação de um pequeno aplicativo no Delphi acessando o BD Access.

Utilização de Acion Lists;

- * Demonstração do Assistente de tabelas do Access que ajuda a normalizar o Banco de Dados;
- * Demonstração do Analisador de Desempenho do Access;
- * Definindo e alterando senhas no Access;
- * Acessando os relatórios desenvolvidos no Access através do Delphi: visualização em tela e envio para impressão sem que o usuário perceba que o relatório foi feito no Access;
- * Criação de índices primários e secundários;
- * Introdução ao conceito de boletos bancários;
- * Apresentação do Manual da Federação Brasileira dos Bancos;
- * Criação passo a passo de um registro de 240 posições definido no manual da Febraban;
- * Apresentação de uma solução pronta (componente open source) para a impressão de boletos bancários em Delphi, emissão do arquivo de remessa que é enviado pelas empresas aos bancos e leitura do arquivo de retorno que é enviado pelos bancos para as empresas;
- * Instalação e utilização do referido componente: impressão de boletos bancários em QuickReport (visualização personalizada e RaveReports, exportação do boleto para PDF, geração do arquivo de remessa);
- * Demonstrando a impressão de boletos bancários com dados vindos de uma tabela do banco de dados Access;
- * Orientações para migrar o componente para uma versão posterior do Delphi;