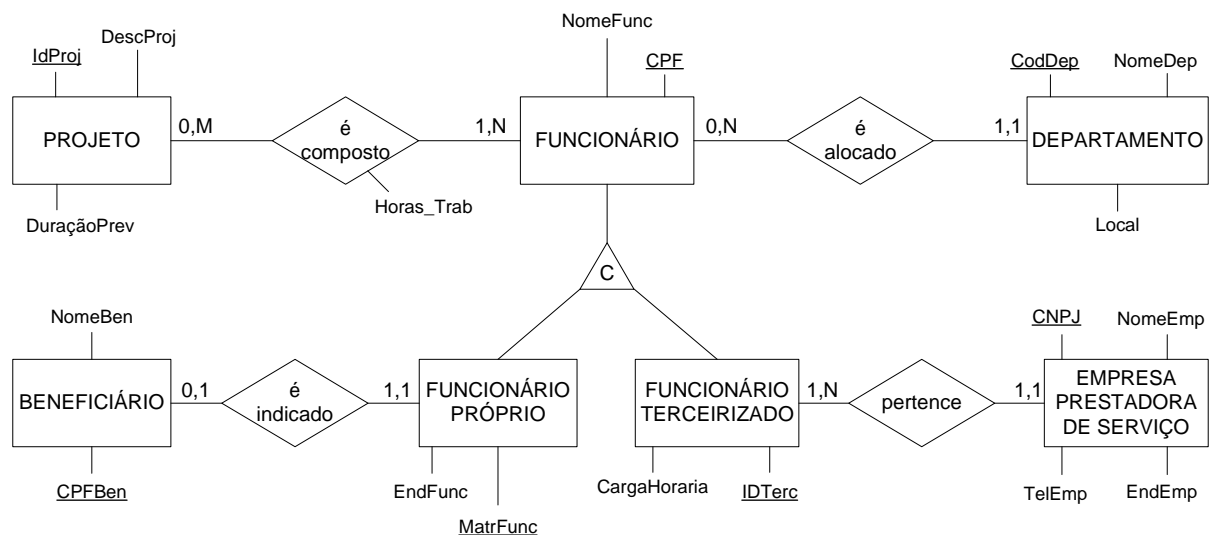




DBDesigner 4

O **DBDesigner 4** é uma ferramenta CASE (*Computer-Aided Software Engineering*) voltada para a modelagem de dados dos sistemas.

Para utilizá-lo é necessário, primeiramente, criar o Modelo Conceitual dos dados do sistema. Como exemplo, será utilizado o seguinte Modelo Conceitual:



Funcionamento do DBDesigner 4

Antes de modelar os dados utilizando o DBDesigner 4 é importante conhecer como ele funciona, em termos de derivação do Modelo Conceitual para o Modelo Lógico.

Como já vimos, existem diversas opções para derivar um modelo para o outro. O DBDesigner 4 tem as suas opções bem definidas e que devemos conhecer.

1. O Modelo Conceitual já deve estar normalizado, pois o DBDesigner 4 apenas modela e não normaliza as tabelas.
2. Derivação de relacionamentos 1:1
O DBDesigner 4 cria uma cópia da chave primária de uma das tabelas como chave estrangeira da outra tabela. Você deve decidir qual das tabelas deve receber a chave estrangeira antes de utilizar a ferramenta.
3. Derivação de relacionamentos 1:N
O DBDesigner 4 cria uma cópia da chave primária da tabela com cardinalidade 1 como chave estrangeira da tabela com cardinalidade N.
4. Derivação de relacionamentos M:N
O DBDesigner 4 cria uma terceira tabela que "liga" as tabelas com cardinalidade M e N. Nessa tabela serão usadas cópias das chaves primárias das duas tabelas, usadas como

chaves primárias e chaves estrangeiras. A cardinalidade entre as tabelas iniciais e a tabela criada são 1:N.

5. Derivação de relacionamentos generalização/especialização

O DBDesigner cria uma cópia da chave primária da tabela generalizada como chave estrangeira das tabelas especializadas. O relacionamento resultante é do tipo 1:N.

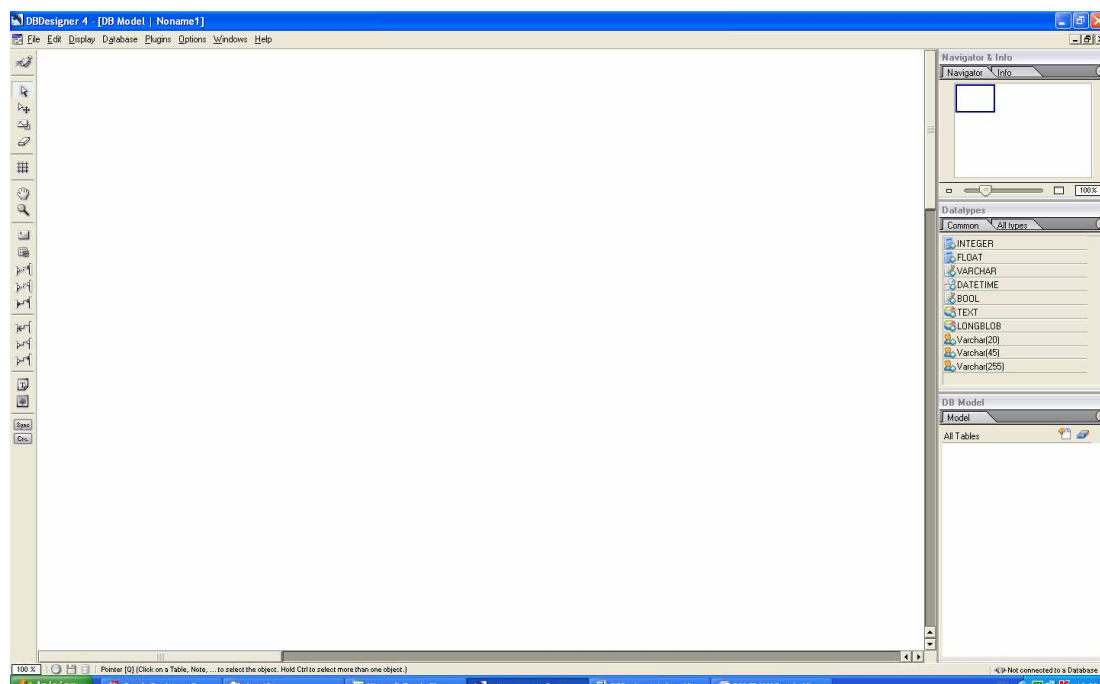
6. Derivação de relacionamentos com atributos

Nos relacionamentos com cardinalidade 1:1 ou 1:N, que possuam atributos, devemos decidir como deve ser feita a derivação desses atributos (criar uma terceira tabela, passar os atributos para uma das tabelas, etc. como vimos na teoria) ANTES de utilizar o DBDesigner 4. Ou seja, o nosso Modelo Conceitual deve apresentar os relacionamentos com cardinalidade 1:1 e 1:N SEM qualquer atributo.

No relacionamento M:N, não devemos nos preocupar, pois o DBDesigner 4 permite a inclusão de atributos no relacionamento (atributo fica na tabela criada para "ligar" as duas tabelas).

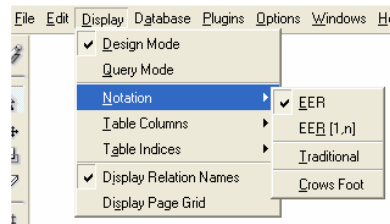
Criando o modelo utilizando o DBDesigner 4

1. Abrir o DBDesigner 4. Surgirá a seguinte janela, com uma área de trabalho em branco, onde será criado o modelo:



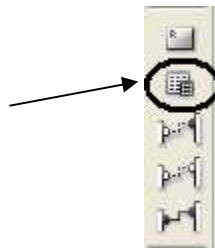
2. Para criar o modelo vamos utilizar a Barra de Ferramentas vertical, localizada à esquerda da janela.
3. Primeiramente vamos configurar o DBDesigner 4 para que apresente o modelo na forma esperada.

A principal modificação é utilizar a notação “Pé-de-Galinha” (Crow’s Foot) para representar o relacionamento entre as tabelas. Para isso, clicar no menu **Display** e escolher a opção **Notation**.



Feito isso, escolha a opção **Crows Foot** (ela ficará marcada).

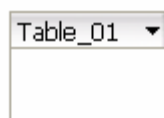
4. Feito isso, você deve criar as tabelas correspondentes às entidades do Modelo Conceitual. Para isso basta clicar no botão **New Table** (barra de ferramentas à esquerda).





Surgirá um cursor em forma de mão com um dedo apontando. Basta clicar no local onde você deseja que a tabela seja criada.

Uma dica é criar as tabelas nas mesmas posições onde você criou as entidades no Modelo Conceitual (ajuda a organizar o modelo resultante).

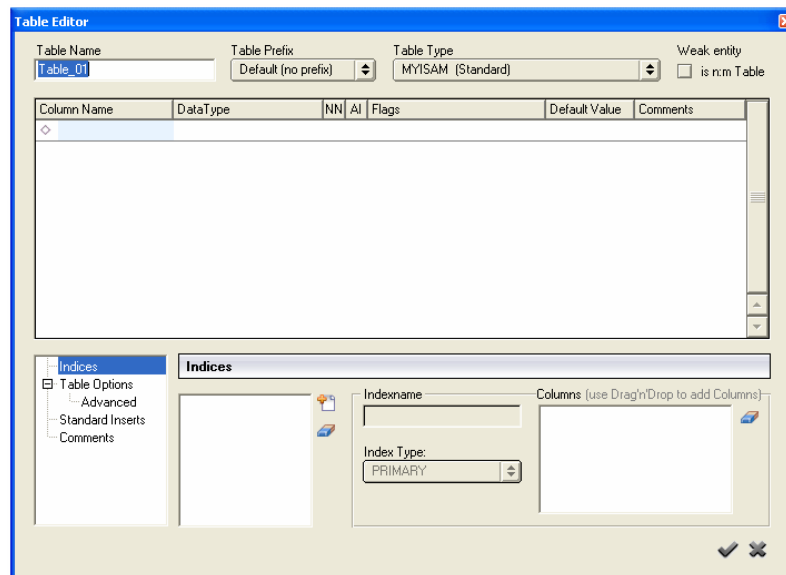
5. Na área de trabalho da ferramenta aparecerá o seguinte:



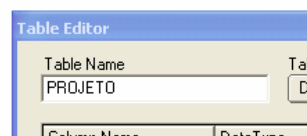
Isso indica que a tabela foi criada. Mas, temos que associar as características das entidades do Modelo Conceitual a cada uma das tabelas criadas.

OBS: Caso deseje excluir alguma tabela criada, você deve clicar no botão **Delete**  (barra de ferramentas vertical). O cursor ficará com a forma de uma borracha e então basta clicar na tabela que você deseja excluir do modelo. Para voltar à operação normal (cursor normal) clicar no botão **Pointer**  (na mesma barra de ferramentas).

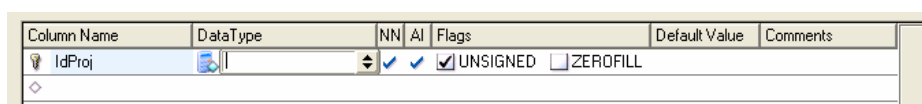
6. Para configurar uma tabela, basta dar duplo clique sobre ela (na área de trabalho da ferramenta). Surgirá a janela **Table Editor** (abaixo):



7. A primeira operação a ser realizada é atribuir um nome para a tabela. Observe que no campo **Table Name** aparece um nome padrão (*Table-01*), que deverá ser substituído pelo nome da entidade. No nosso exemplo, a primeira tabela a ser criada será PROJETO. Basta digitar o novo nome (se o campo **Table Name** não estiver destacado, clicar no campo e digitar).



8. Depois disso devemos determinar as colunas da tabela. Elas correspondem aos atributos da entidade. Para fazer isso, basta clicar no campo destacado abaixo de **Column Name**. Inicie a determinação das colunas pelo atributo chave (chave primária da tabela), pois o DBDesigner 4 atribui a primeira coluna criada como sendo a chave primária da tabela. No nosso exemplo, é o atributo IDProj.
9. Ao digitar o nome da coluna e pressionar ENTER para confirmar, o foco passará para o campo **Data Type**.



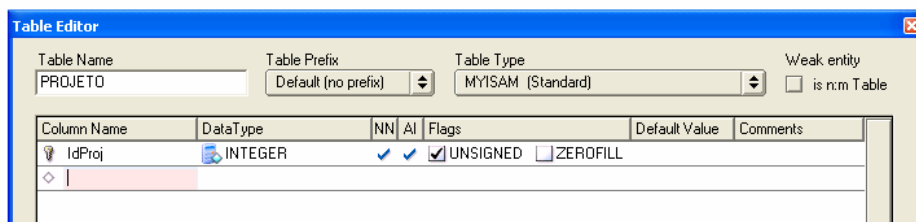
10. Nesse campo, que é uma caixa drop-down, basta clicar no botão à direita com seta dupla e escolher o tipo de dado que será armazenado nesse campo.

OBS: Para efeito de modelagem, utilizaremos as seguintes opções:

- **INTEGER** – para valores numéricos inteiros.
- **FLOAT** – para valores numéricos reais.
- **VARCHAR** – para caracteres de texto.

O tipo **VARCHAR** permite que qualquer quantidade de caracteres seja colocada no campo. Ele ajusta o tamanho do campo dinamicamente ao que for digitado. É possível determinar uma quantidade máxima de caracteres para esse tipo de campo (o que é interessante quando soubermos quantos caracteres serão usados, como por exemplo, em Unidade da Federação que sempre utiliza 2 caracteres). Para isso a ferramenta oferece as opções **Varchar (20)**, **Varchar (45)** e **Varchar (255)**. Caso nenhum desses valores seja o desejado, escolher qualquer uma das opções e depois clicar sobre a escolha e editar o valor dentro dos parênteses para o valor desejado.

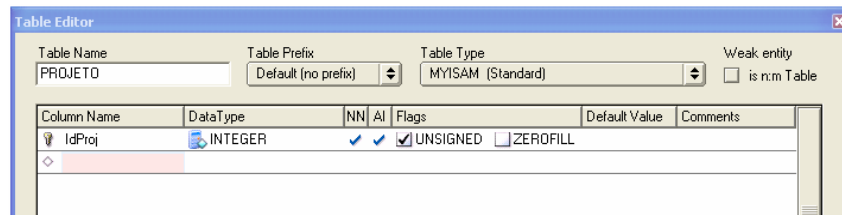
No nosso exemplo, escolheremos INTEGER para IDProj. Terminada a escolha, basta pressionar ENTER para confirmar.



11. Além do nome e do tipo da coluna outras características podem ser configuradas:

- **NN (Not Null)** – Quando a opção estiver selecionada (sinal ☒ aparecendo) significa que não é permitido deixar a coluna vazia (sem digitação). Para alterar a opção, basta clicar no sinal (ele é do tipo "liga/desliga"). **Obs:** a coluna chave primária SEMPRE terá essa opção selecionada.
- **AI (Auto Increment)** – Quando essa opção estiver selecionada significa que os valores da coluna serão criados automaticamente pelo banco de dados. Utilizada somente para colunas chave primária, determina que ela será automaticamente numerada de um em um, a partir do número 1.
- **Flags** – essas opções serão alteradas automaticamente de acordo com o tipo de dado escolhido.
- **Default Value** – deve ser usado quando a coluna tiver um valor padrão que deve ser utilizado, mesmo que não seja entrado dado nenhum nela. Como exemplo, podemos imaginar uma tabela onde os dados sobre o estado de origem da pessoa devem ser armazenados. Imaginemos que 99% das pessoas são do estado de SP, nesse caso seria interessante determinar SP como valor padrão.
- **Comments** – esse campo serve para que possamos colocar comentários sobre as colunas criadas.

Terminada a configuração da primeira coluna, o foco passa para a linha de baixo, para que possamos adicionar novos campos/colunas à tabela criada. A aparência será a seguinte:

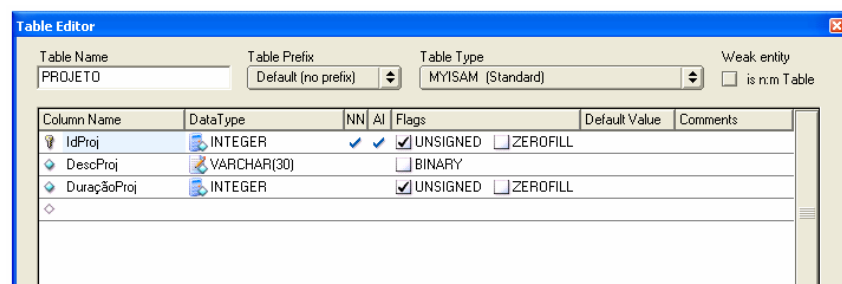




Observe que surgiu o símbolo de uma chave na esquerda da primeira linha (coluna IdProj). Isso indica que esse atributo é chave (ele cria isso automaticamente). Para modificar isso, ou indicar que outra coluna também é uma chave primária, basta clicar na posição à esquerda do nome da coluna (funciona no modo "liga/desliga").

12. Depois disso é só acrescentar as colunas que faltam na tabela.

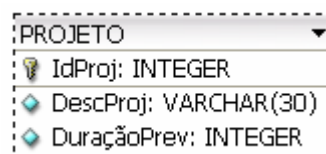
Column Name	Data Type
DescProj	Varchar(30)
DuraçãoPrev	INTEGER

Terminada a criação das colunas, pressionar ESC. A janela deve ter a seguinte aparência:



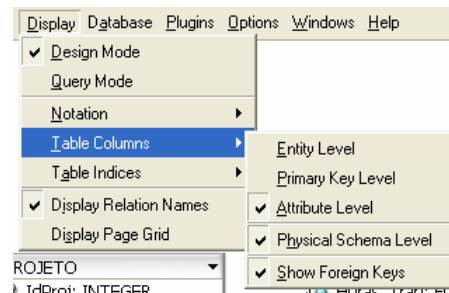
13. Clicar no botão de confirmação  para que a configuração seja complementada. Para cancelar as alterações realizadas clicar no botão de cancelamento .

14. Terminada a criação da tabela, ela terá a seguinte aparência na área de trabalho (agora com as suas novas características).



OBS: O DBDesigner 4 tem como padrão apresentar os nomes das colunas da tabela junto com os tipos de dados escolhidos para elas. Se você achar que isso pode deixar o seu modelo com informações demais e atrapalhar a compreensão do mesmo, é possível ocultar a informação do tipo de dados.

Para isso escolher o menu **Display**, acessar a opção **Table Columns** e desmarcar a opção **Physical Schema Level**.



15. Para salvar o trabalho, o procedimento é o mesmo de qualquer aplicativo comum. Na primeira vez utilizar a opção **Save As**, onde será designado um nome para o arquivo e depois a opção **Save** para salvar os resultados intermediários do trabalho.

OBS: O arquivo é salvo com a extensão **.XML**.

16. Depois disso, é necessário criar as outras tabelas. Para isso, repetir as etapas de 4 a 13 para cada uma das tabelas. As tabelas, suas colunas e tipos de dados são as seguintes:

Tabela: FUNCIONÁRIO	
CPF (PK)	Varchar(11)
NomeFunc	Varchar(45)

Tabela: DEPARTAMENTO	
CodDep (PK)	Integer
NomeDep	Varchar(45)
Local	Varchar(20)

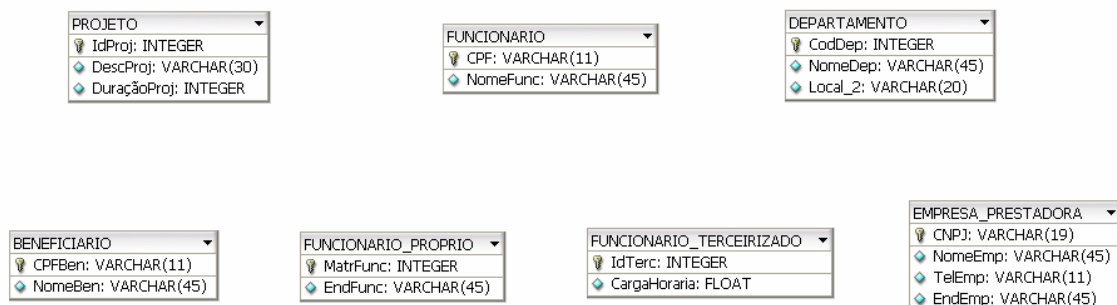
Tabela: BENEFICIÁRIO	
CPFBen (PK)	Varchar(11)
NomeBen	Varchar(45)

Tabela: FUNCIONÁRIO_PRÓPRIO	
MatrFunc	Integer
EndFunc	Varchar(45)

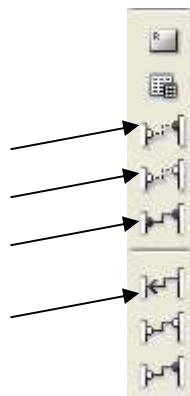
Tabela: FUNCIONÁRIO_TERCEIRIZADO	
IdTerc	Integer
CargaHorária	Float

Tabela: EMPRESA_PRESTADORA	
CNPJ (PK)	Varchar(19)
NomeEmp	Varchar(45)
TelEmp	Varchar(11)
EndEmp	Varchar(45)

17. Terminado o processo de criação das tabelas, a área de trabalho da ferramenta apresentará a seguinte aparência:




18. A próxima etapa é criar os relacionamentos entre as tabelas. Para isso utilizaremos os botões correspondentes na barra de ferramentas vertical (esquerda).



19. Para exemplificar cada um dos tipos de relacionamento, vamos acompanhar o Modelo Conceitual.

20. O primeiro relacionamento a ser criado é aquele entre as entidades PROJETO e FUNCIONÁRIO (relacionamento chamado "é composto").

Observe que o relacionamento tem duas características importantes: ele é um relacionamento M:N e possui atributos (*Horas_Trab*).

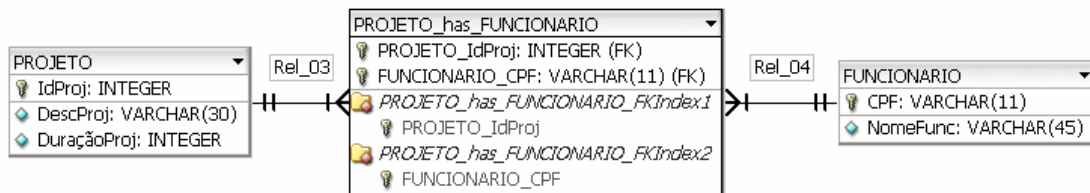
21. Para criar um relacionamento M:N basta clicar no botão **New n:m Relation** . O cursor mudará de forma, indicando que estamos no modo de criação de

relacionamentos. Clicar na primeira tabela do relacionamento e depois clicar na segunda.

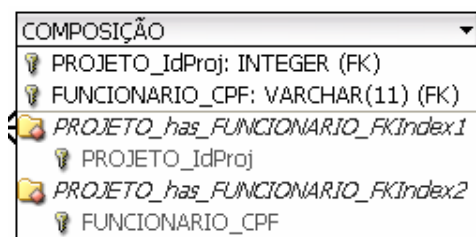
O relacionamento será criado na forma de uma nova tabela (como vimos na derivação de modelos), formada pela cópia das duas chaves primárias das tabelas representantes do relacionamento, como chaves estrangeiras.

OBS: Note que o layout do modelo não ficou muito bom. Para alterar a posição da tabela, basta clicar nela e arrastar para uma posição melhor.

O resultado será o seguinte:



A nova tabela se chama PROJETO_has_FUNCIONÁRIO (ou seja, PROJETO tem FUNCIONÁRIO). É possível alterar o nome dessa tabela. Vamos alterar o nome para COMPOSIÇÃO (dar duplo clique na tabela, clicar no campo **Table Name** e digitar o novo nome). O resultado é:



22. A próxima etapa é acrescentar o atributo (coluna) no relacionamento, ou seja, na nova tabela criada. Para isso, o processo é o mesmo que vimos para alterar as configurações das tabelas.

Duplo clique na tabela, clicar na linha em branco abaixo da última coluna existente e acrescentar o atributo. No nosso exemplo, *Horas_Trab* do tipo FLOAT. O resultado é:



23. Os relacionamentos resultantes também possuem nomes padrão que podem ser alterados (duplo clique sobre o relacionamento e alterar o nome). O DBDesigner 4


obriga que o relacionamento tenha um nome (apesar de não utilizarmos isso na teoria), então vamos utilizar os nomes "cria" e "possui" para os relacionamentos resultantes.



24. O próximo relacionamento a ser criado é aquele entre as entidades FUNCIONÁRIO e DEPARTAMENTO (relacionamento denominado "é alocado"). Este é um relacionamento 1:N sem atributos.

OBS: Como mencionado anteriormente, se o relacionamento for 1:N e tiver atributos, é necessário tomar a decisão de como esses atributos serão derivados (conforme vimos na derivação dos modelos) e representar isso logo na criação das tabelas.

25. Para criar um relacionamento 1:N basta clicar no botão **New 1:n Non-Identifying-**

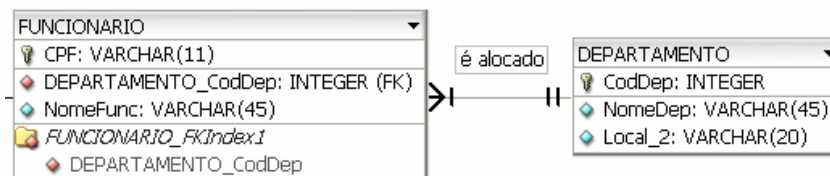
Relation . O cursor mudará de forma, indicando que estamos no modo de criação de relacionamentos. Clicar **OBRIGATORIAMENTE** em primeiro lugar na tabela com cardinalidade 1 do relacionamento e **DEPOIS** clicar na tabela com cardinalidade N.

OBS: Se isso não for feito, o relacionamento será criado com a cardinalidade "invertida".

O relacionamento será criado utilizando o campo chave primária da tabela com cardinalidade 1 como chave estrangeira na tabela com cardinalidade N (como vimos na derivação de modelos).

Podemos atribuir para o relacionamento o mesmo nome do nosso Modelo Conceitual ("é alocado").


O resultado será o seguinte:



26. O próximo relacionamento a ser criado é aquele entre as entidades BENEFICIÁRIO e FUNCIONÁRIO_PROPRIO (relacionamento "é indicado"). Este é um relacionamento 1:1 sem atributos.

OBS: Como mencionado anteriormente, se o relacionamento for 1:1 e tiver atributos, é necessário tomar a decisão de como esses atributos serão derivados (conforme vimos na derivação dos modelos) e representar isso logo na criação das tabelas.

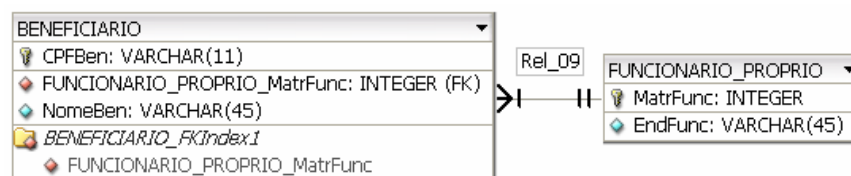
27. Para criar um relacionamento 1:1 basta clicar no botão **New 1:1 Non-Identifying-**

Relation . O cursor mudará de forma, indicando que estamos no modo de criação de relacionamentos.

Para esse tipo de cardinalidade é necessário tomar outra decisão. A criação do relacionamento é feita colocando uma cópia da chave primária de uma das tabelas como chave estrangeira da outra (como vimos na derivação de modelos). Portanto, devemos determinar em qual tabela será criada a chave estrangeira.

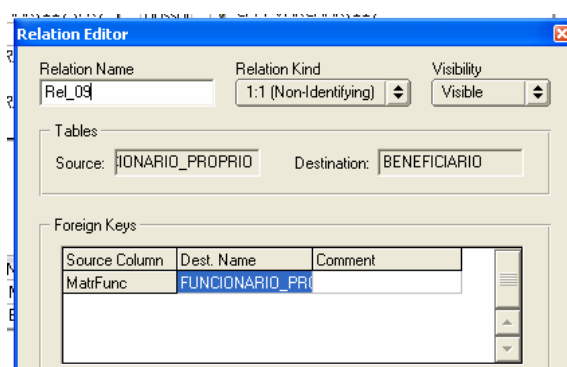
Tomada essa decisão, clicar na tabela que cederá a cópia da chave primária (no nosso exemplo escolheremos FUNCIONARIO_PRÓPRIO) e depois clicar na tabela que receberá a chave estrangeira (no nosso exemplo BENEFICIÁRIO).

O resultado será o seguinte:




Observe que algo de estranho aconteceu. O relacionamento resultante é 1:N e não 1:1! Isso acontece devido ao DBDesigner 4 entender que usar uma chave primária de uma tabela, como chave estrangeira de outra é característica de relacionamento 1:N.

Para resolver esse problema, dar duplo clique sobre o relacionamento. Surgirá a janela **Relation Editor**.

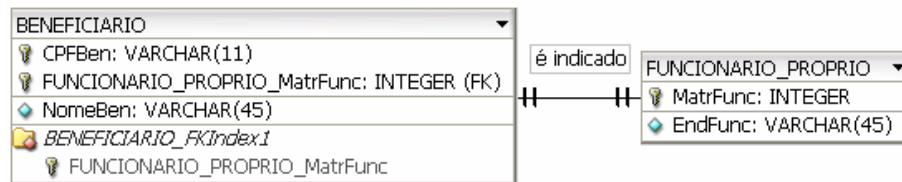


Nessa janela clicar na seta dupla do campo **Relation Kind** e modificar para a opção **1:1**

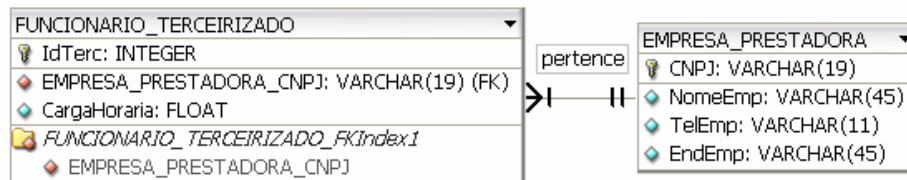
(sem nenhum outro detalhe no nome) e confirmar .

Na mesma janela podemos atribuir para o relacionamento o mesmo nome do nosso Modelo Conceitual ("é indicado").


O resultado deve ser:



28. Agora devemos criar o outro relacionamento 1:N (entre as tabelas FUNCIONÁRIO_TERCEIRIZADO e EMPRESA_PRESTADORA) conforme as etapas 24 e 25. O relacionamento se chama "pertence". O resultado será:

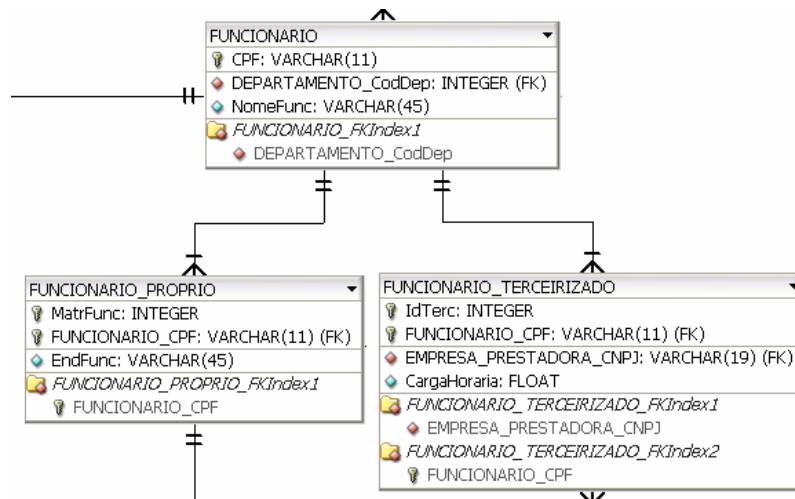


29. A próxima etapa é criar os relacionamentos de generalização/especialização do nosso Modelo Conceitual. O relacionamento acontece entre as entidades FUNCIONÁRIO (generalizada) e as entidades FUNCIONÁRIO_PRÓPRIO e FUNCIONÁRIO_TERCEIRIZADO (especializadas).

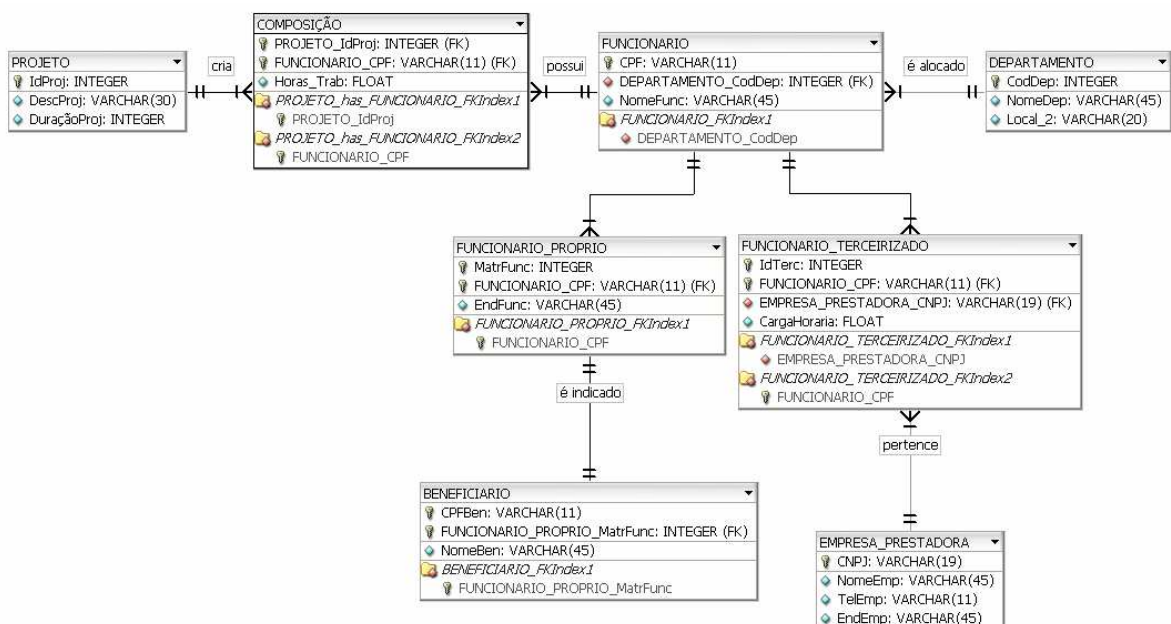
30. Para criar esse relacionamento clicar no botão **New Generalization** . O cursor mudará de forma, indicando que estamos no modo de criação de relacionamentos. Clicar na tabela generalizada (FUNCIONÁRIO) e depois clicar na tabela a ser especializada (FUNCIONÁRIO_TERCEIRIZADO). O relacionamento será criado colocando a chave primária (ou chaves primárias) da tabela generalizada como chave estrangeira da tabela especializada. Temos que repetir o processo para cada uma das tabelas especializadas.

OBS: Se alguma das tabelas especializadas possuir o relacionamento 1:1 com alguma outra tabela, sempre deixar essa tabela como última a ser generalizada/especializada. Se isso não for feito o DBDesigner 4 cria uma chave estrangeira a mais no relacionamento 1:1, sem necessidade.

O resultado será o seguinte:



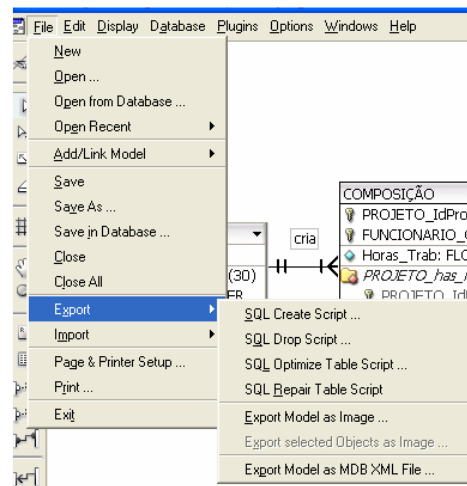
Com isso o modelo está terminado. O resultado será esse (depois de um rearranjo no layout):



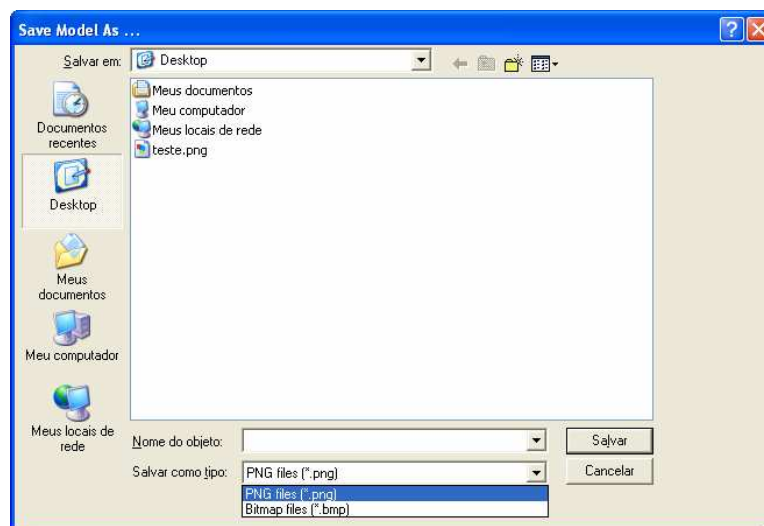
31. Terminado o trabalho de modelagem e salvo o arquivo é possível utilizar esse modelo de diversas formas. Vejamos algumas delas:

I. Criar um arquivo de imagem com o modelo

Para criar um arquivo de imagem a partir do modelo acessar o menu **File**, acessar a opção **Export** e clique na opção **Export Model as Imagem...**



Escolhida essa opção surgirá a janela **Save Model As...** (figura abaixo). Nessa janela é necessário escolher o local onde o arquivo de imagem será gravado, um nome para a imagem e o tipo de arquivo de imagem que será gerado (arquivo **PNG** ou arquivo **Bitmap**). Basta clicar em **Salvar**.

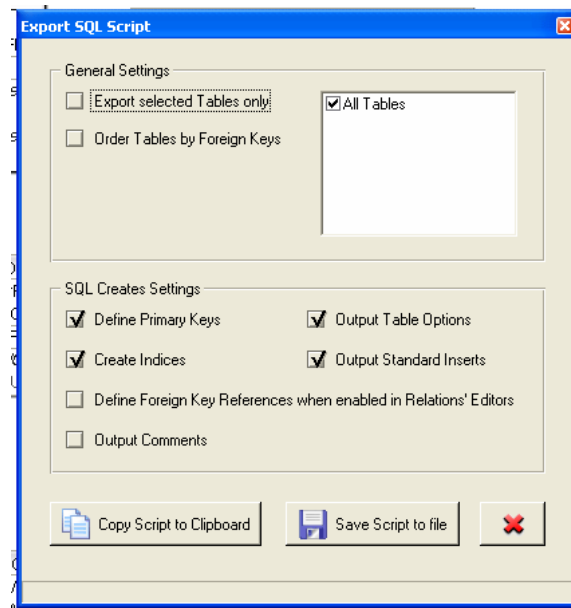


OBS: Apenas para efeito de comparação e para auxiliar na escolha do tipo de arquivo e levando em consideração o processo realizado nesse exemplo, é possível afirmar que a qualidade dos dois formatos é igual (a resolução é exatamente a mesma). Mas, quando comparamos o tamanho dos arquivos, temos que o arquivo gravado no formato BMP ficou com aproximadamente 2 MB de tamanho e o arquivo gravado no formato PNG ficou com aproximadamente 40 kB !!!

II. Criar um script SQL para que seja gerado o banco de dados

É possível exportar o arquivo diretamente para um formato de script SQL que pode ser utilizado no gerenciador de banco de dados MySQL para criação das tabelas. O processo é semelhante ao anterior, clicando no menu **File**, acessando a opção **Export** e clicando na opção **SQL Create Script...**

Escolhida essa opção surgirá a janela **Export SQL Script** (figura abaixo).



Nessa janela é possível escolher entre copiar o script para a Área de Transferência (e depois Colar em algum editor de textos), usando o botão **Copy Script to Clipboard**, ou salvar o script em um arquivo SQL (que pode ser usado no gerenciador de banco de dados), usando o botão **Save Script to File**.

Abaixo o script resultante da modelagem do exemplo utilizado até aqui:

```
CREATE TABLE BENEFICIARIO (
  CPFBen VARCHAR(11) NOT NULL AUTO_INCREMENT,
  FUNCIONARIO_PROPRIO_MatrFunc INTEGER UNSIGNED NOT NULL,
  NomeBen VARCHAR(45) NULL,
  PRIMARY KEY(CPFBen, FUNCIONARIO_PROPRIO_MatrFunc),
  INDEX BENEFICIARIO_FKIndex1 (FUNCIONARIO_PROPRIO_MatrFunc)
);

CREATE TABLE COMPOSIÇÃO (
  PROJETO_IdProj INTEGER UNSIGNED NOT NULL,
  FUNCIONARIO_CPF VARCHAR(11) NOT NULL,
  Horas_Trab FLOAT NULL,
  PRIMARY KEY(PROJETO_IdProj, FUNCIONARIO_CPF),
  INDEX PROJETO_has_FUNCIONARIO_FKIndex1 (PROJETO_IdProj),
  INDEX PROJETO_has_FUNCIONARIO_FKIndex2 (FUNCIONARIO_CPF)
);

CREATE TABLE DEPARTAMENTO (
  CodDep INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  NomeDep VARCHAR(45) NULL,
  Local_2 VARCHAR(20) NULL,
  PRIMARY KEY(CodDep)
);

CREATE TABLE EMPRESA_PRESTADORA (
  CNPJ VARCHAR(19) NOT NULL AUTO_INCREMENT,
```



```
NomeEmp VARCHAR(45) NULL,  
TelEmp VARCHAR(11) NULL,  
EndEmp VARCHAR(45) NULL,  
PRIMARY KEY(CNPJ)  
);  
  
CREATE TABLE FUNCIONARIO (  
    CPF VARCHAR(11) NOT NULL AUTO_INCREMENT,  
    DEPARTAMENTO_CodDep INTEGER UNSIGNED NOT NULL,  
    NomeFunc VARCHAR(45) NULL,  
    PRIMARY KEY(CPF),  
    INDEX FUNCIONARIO_FKIndex1(DEPARTAMENTO_CodDep)  
);  
  
CREATE TABLE FUNCIONARIO_PROPRIO (  
    MatrFunc INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    FUNCIONARIO_CPF VARCHAR(11) NOT NULL,  
    EndFunc VARCHAR(45) NULL,  
    PRIMARY KEY(MatrFunc, FUNCIONARIO_CPF),  
    INDEX FUNCIONARIO_PROPRIO_FKIndex1(FUNCIONARIO_CPF)  
);  
  
CREATE TABLE FUNCIONARIO_TERCEIRIZADO (  
    IdTerc INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    FUNCIONARIO_CPF VARCHAR(11) NOT NULL,  
    EMPRESA_PRESTADORA_CNPJ VARCHAR(19) NOT NULL,  
    CargaHoraria FLOAT NULL,  
    PRIMARY KEY(IdTerc, FUNCIONARIO_CPF),  
    INDEX FUNCIONARIO_TERCEIRIZADO_FKIndex1(EMPRESA_PRESTADORA_CNPJ),  
    INDEX FUNCIONARIO_TERCEIRIZADO_FKIndex2(FUNCIONARIO_CPF)  
);  
  
CREATE TABLE PROJETO (  
    IdProj INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    DescProj VARCHAR(30) NULL,  
    DuraçãoProj INTEGER UNSIGNED NULL,  
    PRIMARY KEY(IdProj)  
);
```