

Modelagem de Dados



Material Teórico



Ferramenta para Modelagem ER

Responsável pelo Conteúdo:

Prof. Esp. Hugo Fernandes

Revisão Textual:

Profa. Dra. Geovana Gentili Santos

UNIDADE

Ferramenta para Modelagem ER



- Introdução
- BrModelo
- Criando um Diagrama de Entidade-Relacionamento (DER)
- MySQL Workbench



OBJETIVO DE APRENDIZADO

- Estudar um pouco sobre duas ferramentas CASE que auxiliam no desenvolvimento do projeto de Modelagem de dados: BrModelo e MySQL Workbench.



Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

Introdução

Nessa unidade, estudaremos sobre duas ferramentas CASE que auxiliam no desenvolvimento do projeto de Modelagem de Dados: BrModelo e MySQL Workbench. Mas, afinal, o que são ferramentas CASE?

O termo CASE vem do acrônimo *Computer-aided software engineering*, ou traduzindo, Engenharia de Software Auxiliada por Computador. A utilização de ferramentas CASE permite que os projetistas e outros atores envolvidos no processo de desenvolvimento do projeto de banco de dados compartilhem uma visão comum de cada estágio de desenvolvimento. Ferramentas CASE ajudam a garantir um processo disciplinado além de retratar o progresso (ou falta dele) graficamente.

Alguns dos benefícios que as ferramentas CASE e as abordagens semelhantes trazem, é a possibilidade de fazer com que o cliente faça parte do processo (através de diagramas ou análise de regras de negócios) de uma forma mais dinâmica e assertiva. Tornando, assim, o objeto modelado o mais próximo à necessidade e à realidade do cliente.

Como esse tipo de abordagem enfatiza o teste e o redesenho, o custo de manutenção de um produto ao longo de sua vida útil pode ser reduzido consideravelmente. Uma abordagem organizada para o desenvolvimento incentiva a reutilização de código e design, reduzindo custos e melhorando a qualidade.

No estudo sobre Modelagem de Dados, estudaremos, especificamente, ferramentas que auxiliam na concepção, desenvolvimento e implementação de sistemas de banco de dados.

Uma das grandes vantagens oferecidas pelas ferramentas CASE de banco de dados é a possibilidade de criar *scripts SQL* a partir de um diagrama de entidade-relacionamento (DER).

O SQL (*Structure Query Language*) é uma linguagem de programação usada para armazenar e gerenciar dados no SGBD (MySQL, Oracle, SqlServer). SQL foi a primeira linguagem comercial introduzida para o modelo relacional do E. Codd. Hoje, quase todo o SGBD usa o SQL como a linguagem de banco de dados padrão. SQL é usado para executar todo o tipo de operações de dados, como criar e excluir tabelas; inserir, atualizar e excluir dados, e selecionar e exibir dados em um SGBD.

Nos exemplos práticos a seguir, demonstraremos como criar diagramas de entidade-relacionamento (DER) e, em seguida, exportar esse esquema em formato SQL, pronto para ser utilizado na criação de tabelas em um banco de dados.



O que é SQL e por que você deve estudá-lo?

Autor: Paulo Oliveira

<https://goo.gl/AfszrlU>

BrModelo

A ferramenta **BrModelo** é uma ferramenta *freeware* que possibilita o desenvolvimento de diagramas DER, modelo lógico, e exportar esses artefatos em formato de *script SQL* para criação de tabelas no banco de dados (CÂNDIDO, 2007).



O software **BrModelo 3.0** pode ser baixado e executado apenas nos sistemas operacionais Windows: <https://goo.gl/fxFg1>

Visão Geral da Ferramenta

A ferramenta **BrModelo** utiliza o padrão de desenvolvimento *drag and drop*, ou seja, você arrasta e solta objetos para dentro do palco do software. Abaixo, temos a visão geral da ferramenta.

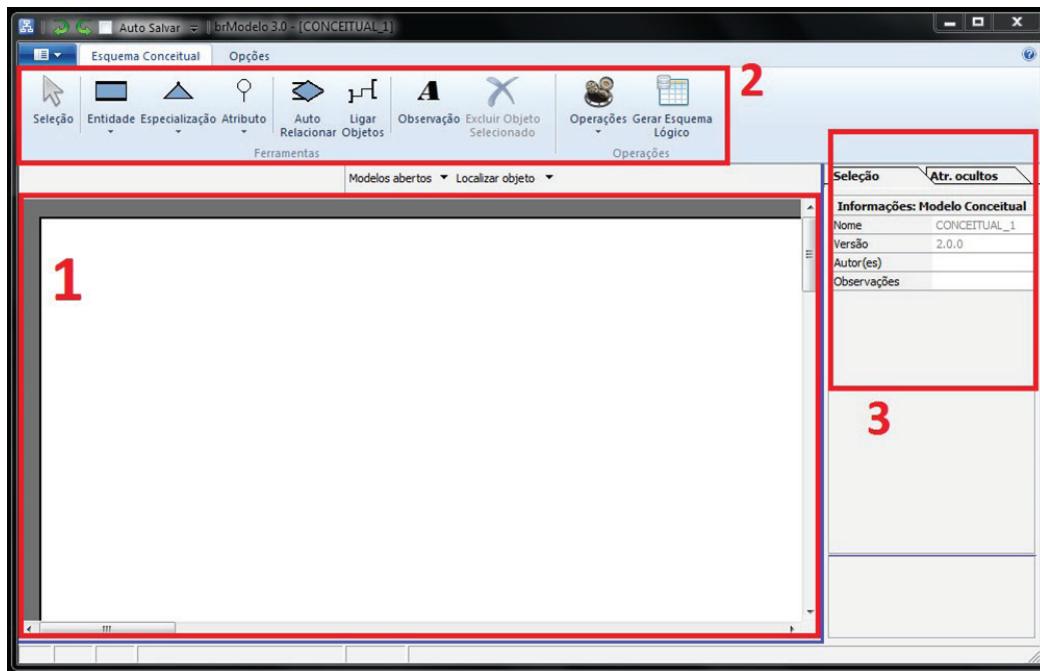


Figura 1

Em destaque, temos três áreas da ferramenta.

- **Área 1.** É o palco do software. É, nesse espaço, que é organizado os objetos do DER.
- **Área 2.** Aqui, estão dispostos os principais objetos do DER (entidade, relacionamento, atributos), esses objetos são arrastados para o palco do software.
- **Área 3.** Refere-se à tela de propriedades de objeto, ao selecionar algum objeto no palco, essa tela apresenta as propriedades do objeto.

Criando um Diagrama de Entidade-Relacionamento (DER)

Nessa sessão, criaremos um DER a partir do seguinte cenário:

- Um funcionário deve possuir no mínimo zero dependente e no máximo muitos dependentes.
- Um dependente deve pertencer, no mínimo, a 1 funcionário e, no máximo, a 1 funcionário.

A partir dessas assertivas, verificamos que existe um relacionamento entre duas entidades: “Funcionário” e “Dependente”, bem como sua cardinalidade.

Nesse contexto, criaremos o DER nos passos a seguir.

Passo 1 – Clique no objeto “Entidade” e, em seguida, selecione **Entidade**.

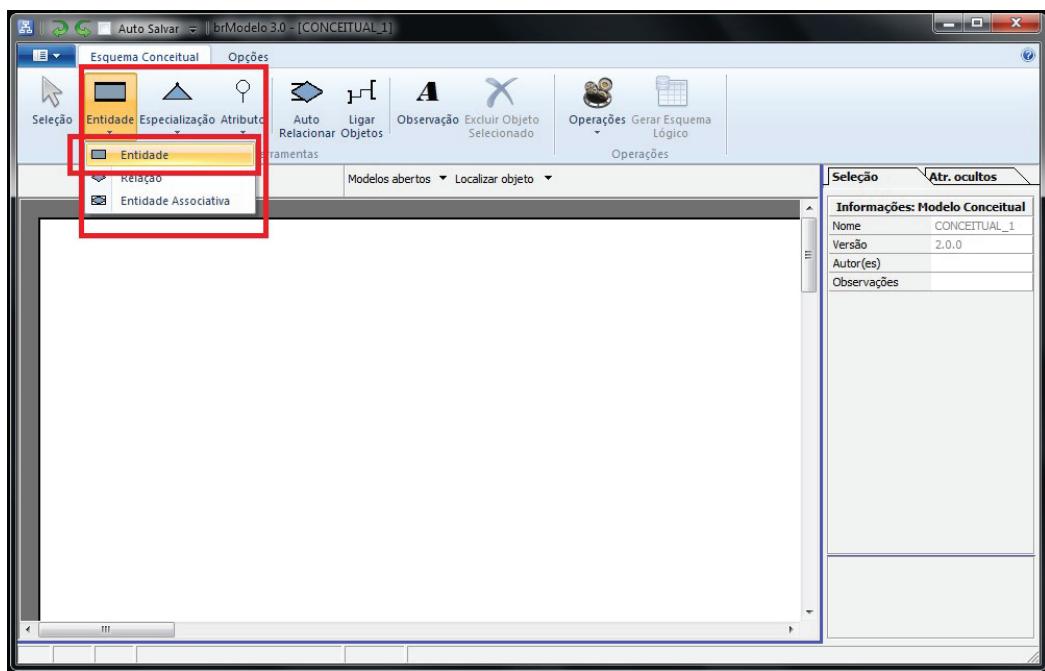


Figura 2

Passo 2 – Arraste o objeto para o palco e com o objeto selecionado, altere a propriedade nome para: **Funcionario** (sem acento) e, em seguida, pressione “Enter” no teclado.

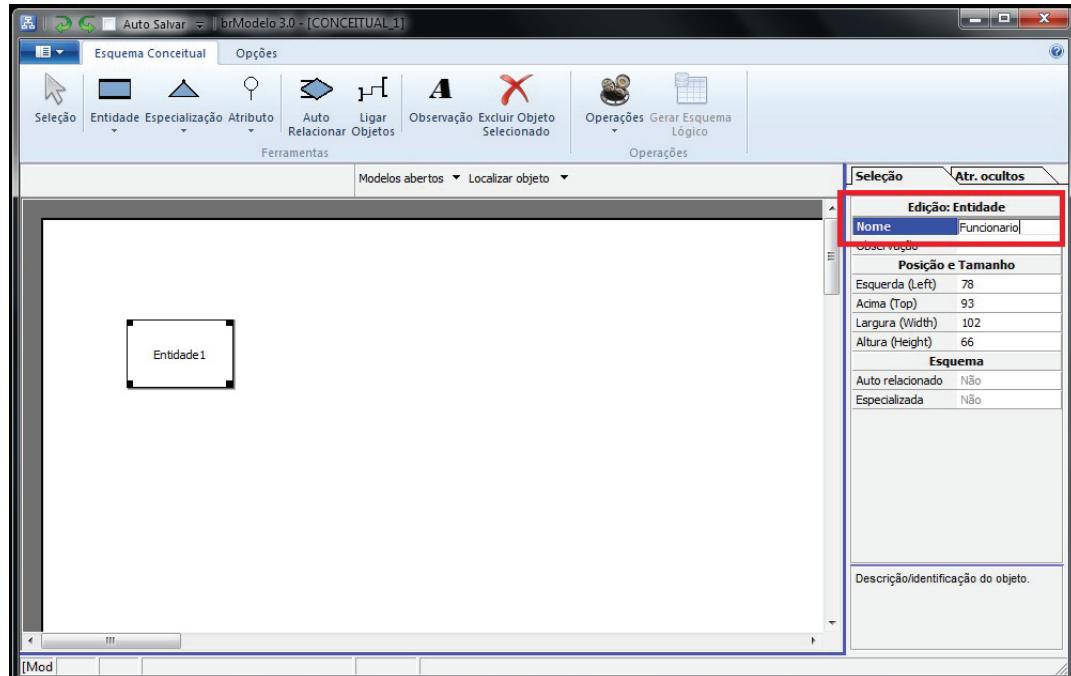


Figura 3

Passo 3 – Clique no objeto “Entidade” e, em seguida, selecione **Relação**.

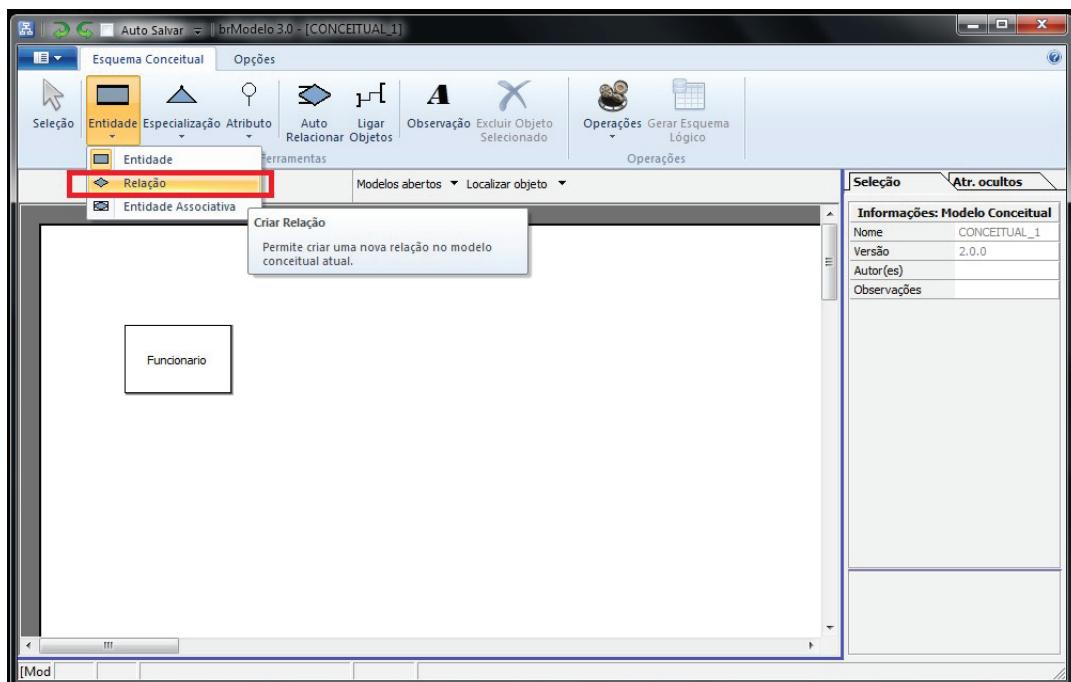


Figura 4

Passo 4 – Arraste o objeto para o palco e, com o objeto selecionado, altere a propriedade nome para: *possui* e, em seguida, pressione “Enter” no teclado.

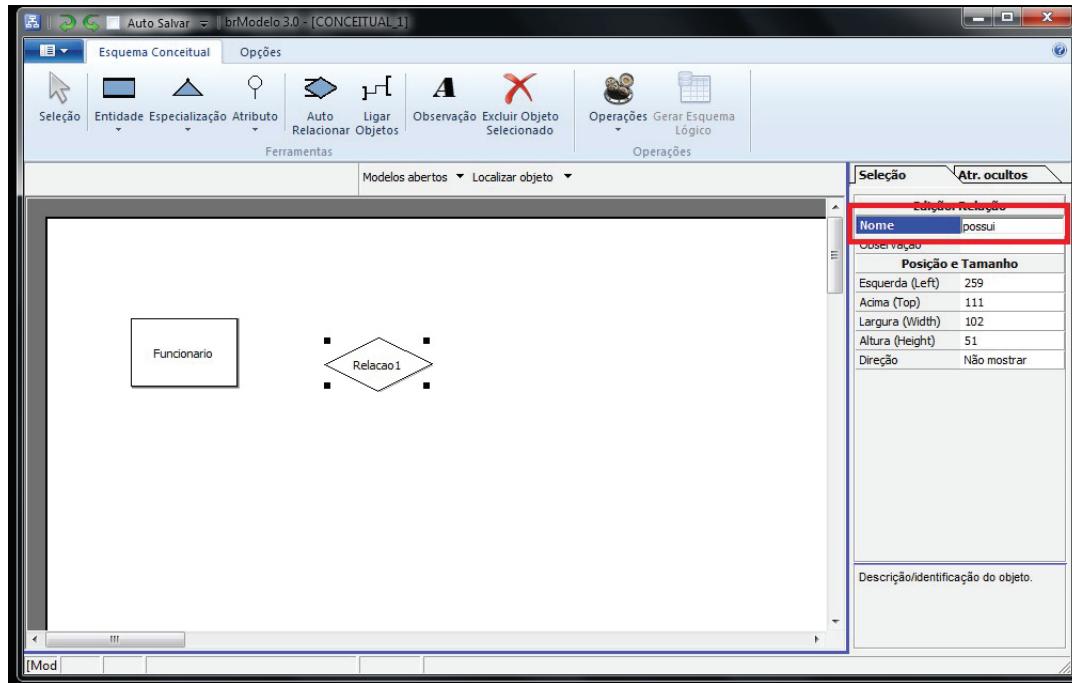


Figura 5

Passo 5 – Adicione um objeto “Entidade” ao palco e altere a propriedade nome para: **Dependente** e, em seguida, pressione “Enter” no teclado.

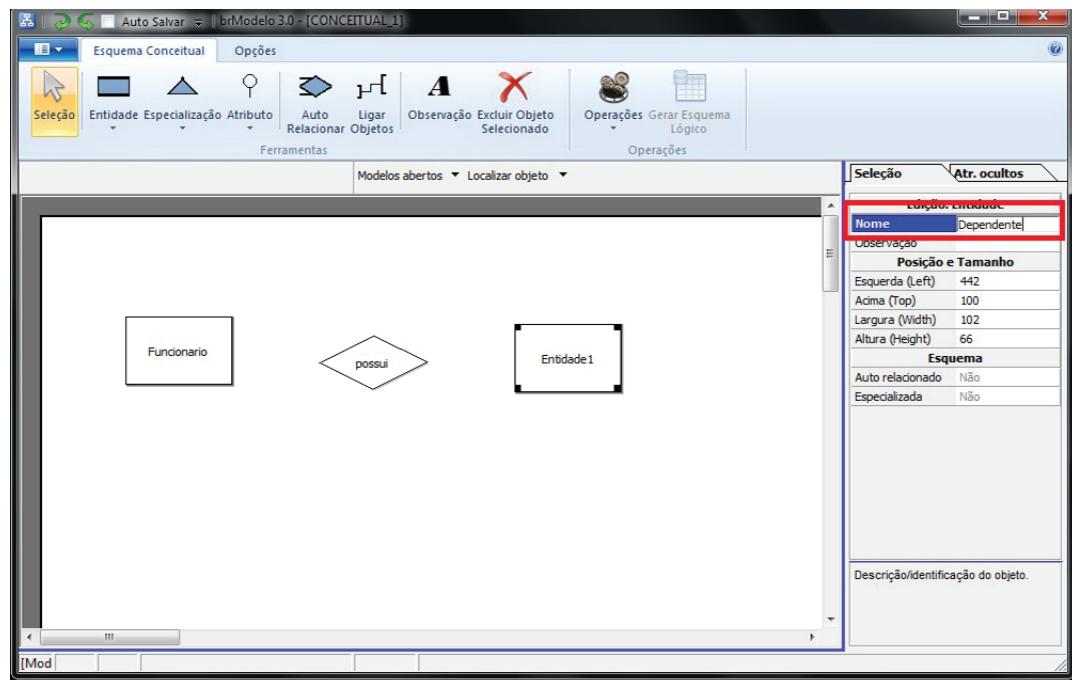


Figura 6

Para representar os graus de relacionamento e sua cardinalidades (mínimas e máximas), utilizamos o objeto “Ligar Objetos”, destacado na imagem abaixo.



Figura 7

Nos próximos passos, ligaremos os objetos e indicaremos seu grau de relacionamento e suas cardinalidades.

Passo 6 – Selecione o objeto “Ligar Objetos” e, com o objeto selecionado, clique com o botão esquerdo do mouse primeiro no objeto “**Funcionário**” e, depois, no objeto “**possui**”. Essa ação irá gerar uma “linha” interligando os dois objetos.

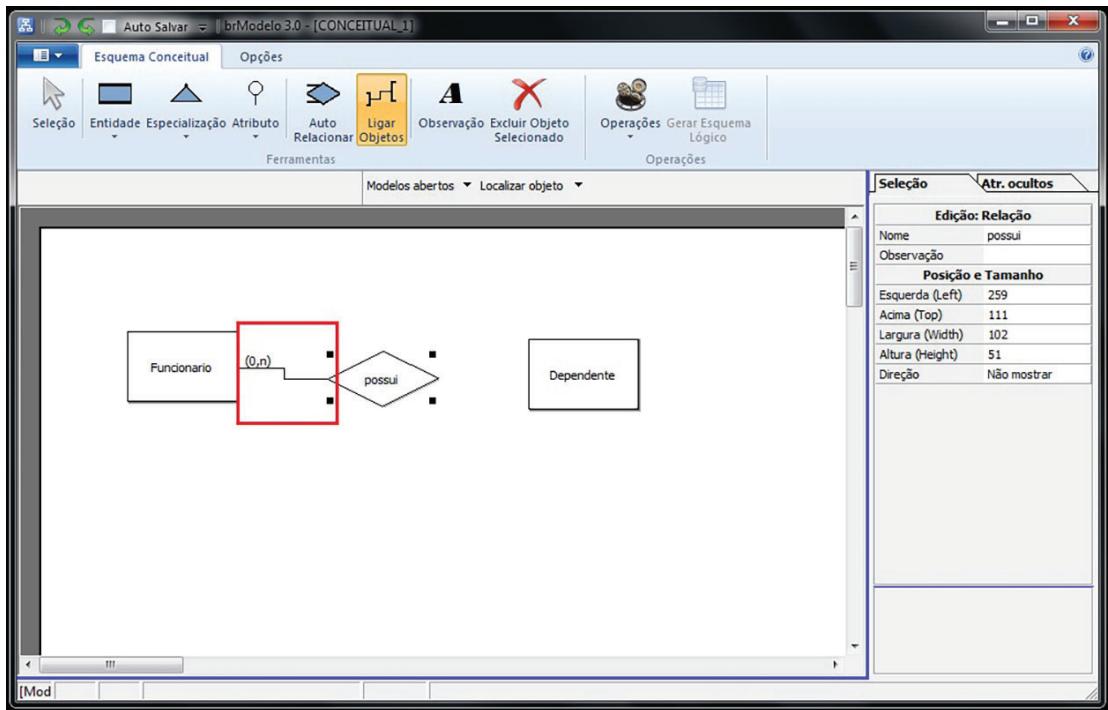


Figura 8

Antes de configurar o grau e a cardinalidade do relacionamento. Faça a ligação entre os objetos **Dependente** e **possui**.

Passo 7 – Selecione o objeto “Ligar Objetos”, com o objeto selecionado, clique com o botão esquerdo do mouse primeiro no objeto “**Dependente**” e, depois, no objeto “**possui**”. Essa ação irá gerar uma “linha” interligando os dois objetos.

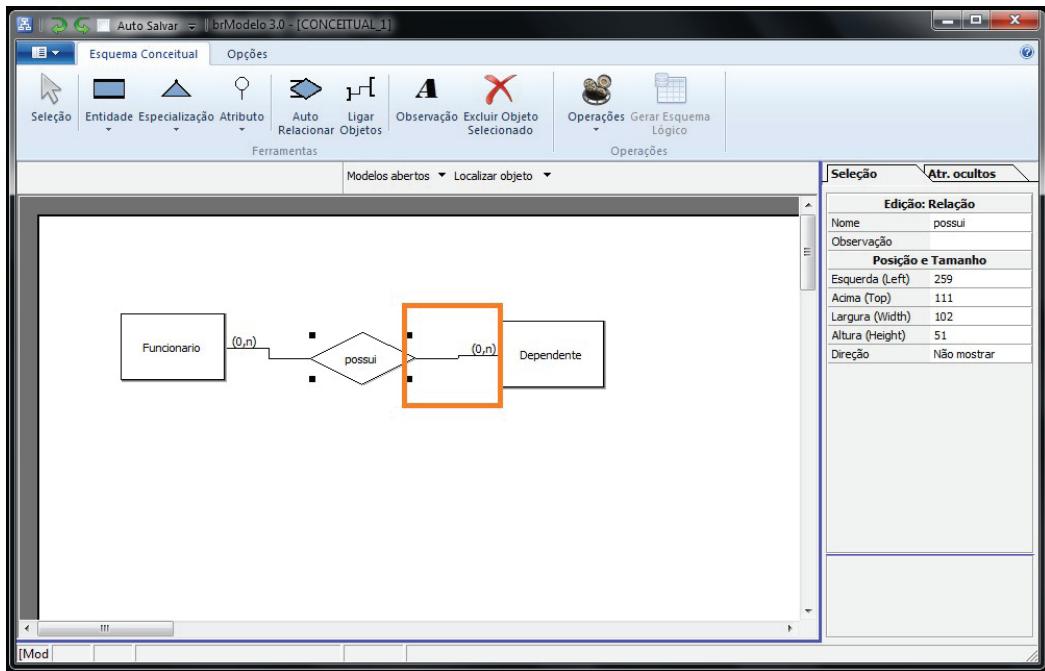


Figura 9

O próximo passo será configurar o grau e cardinalidade do relacionamento entre as duas entidades.

Relembrando o cenário:

- Um funcionário deve possuir, no mínimo, zero dependente e, no máximo, muitos dependentes.
- Um dependente deve pertencer, no mínimo, a 1 funcionário e, no máximo, a 1 funcionário.

Devemos também lembrar que a representação das cardinalidades deve ser informada no lado oposto ao da entidade.

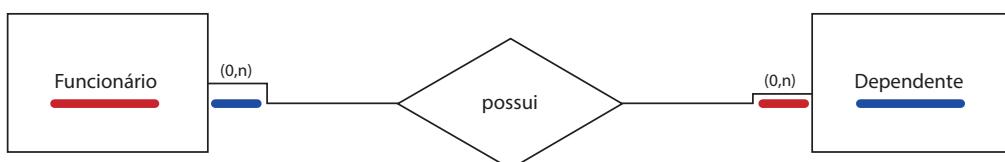


Figura 10

Passo 8 – Para configurar o grau e a cardinalidade da entidade FUNCIONARIO, selecione a cardinalidade ao lado da entidade DEPENDENTE, com a cardinalidade selecionada, altere a propriedade “Cardinalidade” para “0,n”.

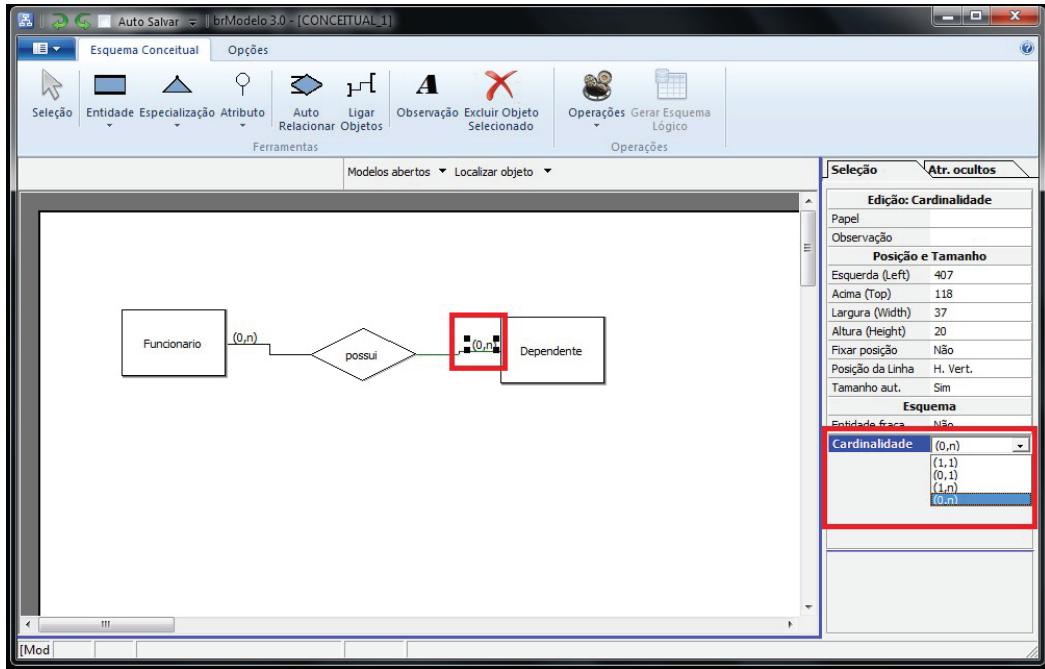


Figura 11

Passo 9 – Para configurar o grau e a cardinalidade da entidade DEPENDENTE, selecione a cardinalidade ao lado da entidade FUNCIONARIO, com a cardinalidade selecionada, altere a propriedade “Cardinalidade” para “1,1”.

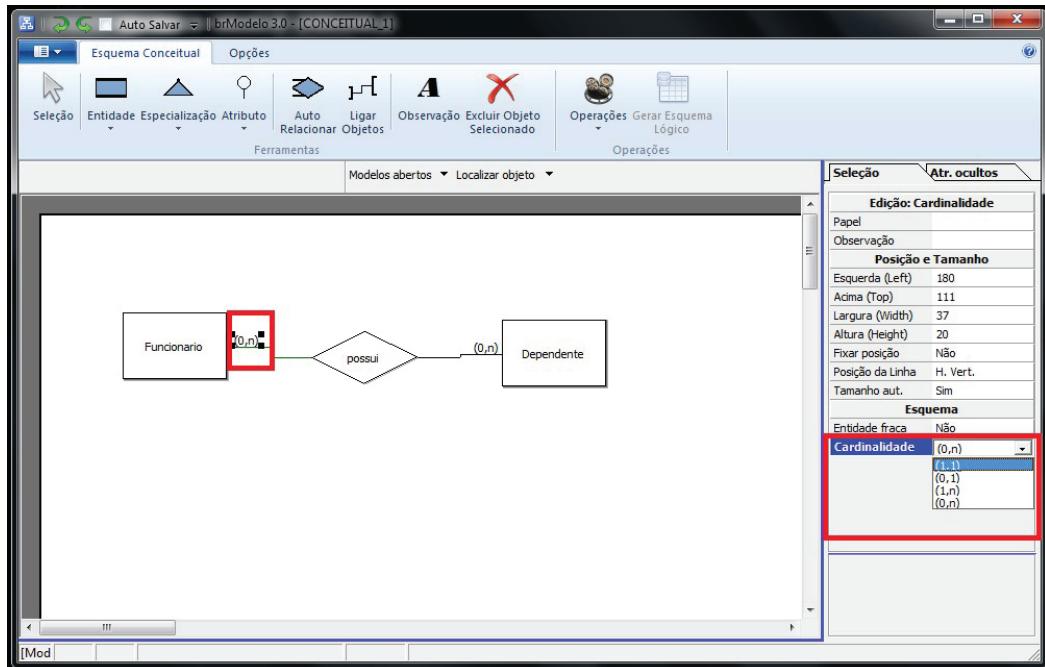


Figura 12

Podemos observar o resultado da implementação do cenário de exemplo no DER abaixo.

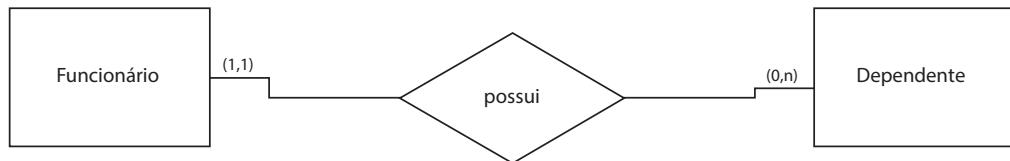


Figura 13

Onde:

- Um funcionário possui no mínimo zero e no máximo vários dependentes e;
- Um dependente está ligado, no mínimo, a um e, no máximo, a um funcionário.

Antes de criarmos o modelo lógico, devemos adicionar atributos às entidades. Para adicionar atributos, basta clicar no objeto “Atributo”, selecionar o atributo desejado e, em seguida, clicar no objeto no qual queremos adicionar o atributo. Em nosso exemplo, utilizaremos os atributos do tipo “Atributo” e “Atributo Identificador”.

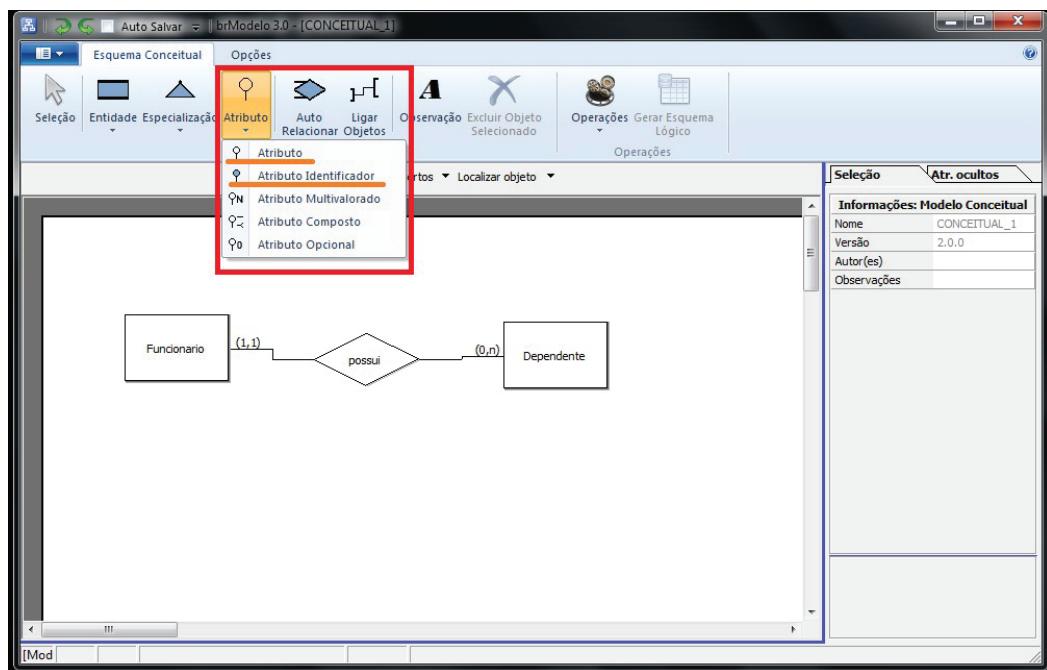


Figura 14

Passo 10 – Clique no objeto “Atributo”, selecione o objeto “Atributo” e, em seguida, clique na entidade FUNCIONARIO. Com o objeto “Atributo” selecionado, altere a propriedade nome para: **Nome** e pressione “Enter” no teclado.

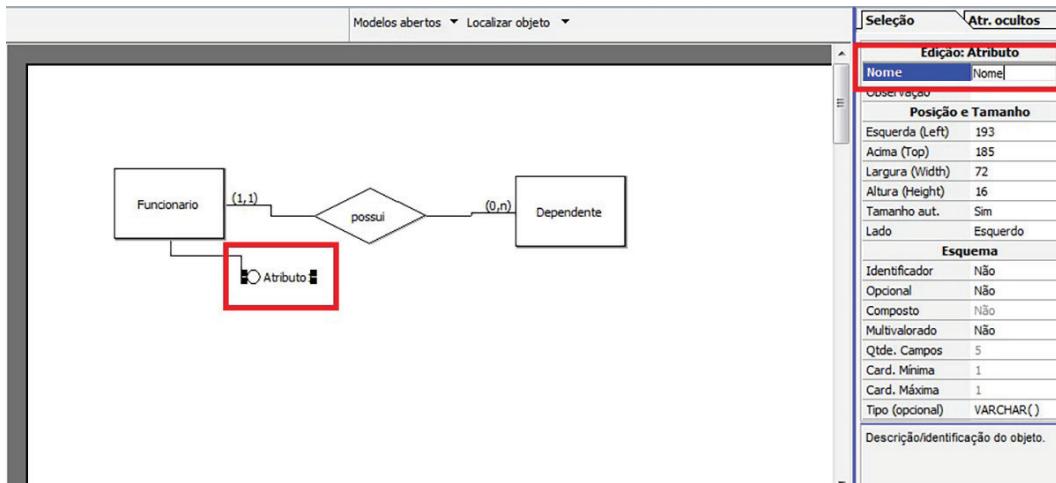


Figura 15

Passo 11 – Clique no objeto “Atributo”, selecione o objeto “Atributo Identificado” e, em seguida, clique na entidade FUNCIONARIO. Com o objeto “Atributo” selecionado, altere a propriedade nome para: **Id_Funcionario** e pressione “Enter” no teclado.

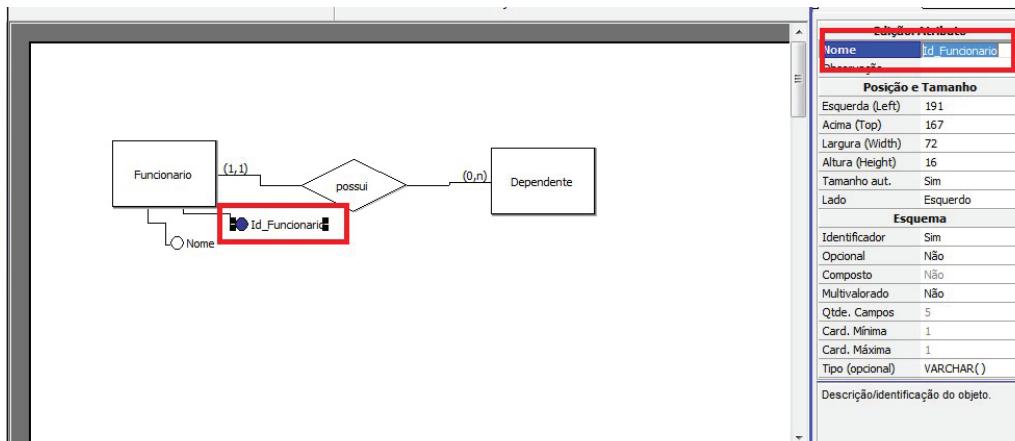


Figura 16

Passo 12 – Adicione dois atributos à entidade DEPENDENTE, um atributo (não-Identificador) com o nome: **Nome_Dependente** e um atributo Identificador com o nome: **Id_Dependente**.

O resultado esperado é o DER exibido abaixo.

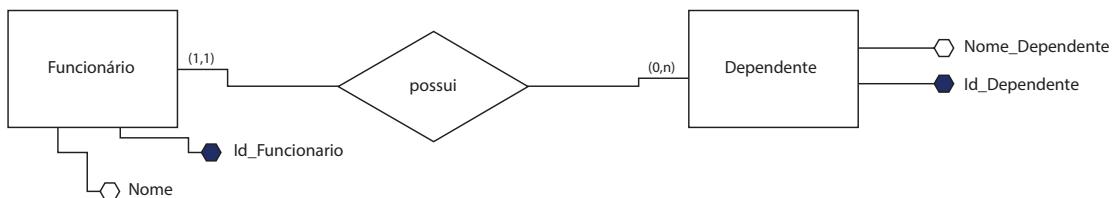


Figura 17

Gerando o esquema Lógico a partir do DER

Para gerar o esquema lógico (modelo lógico) a partir de um DER no **BrModelo**, basta clicar com o botão direito do mouse sobre um local vazio no palco. Em seguida, clique na opção “Gerar esquema lógico”.

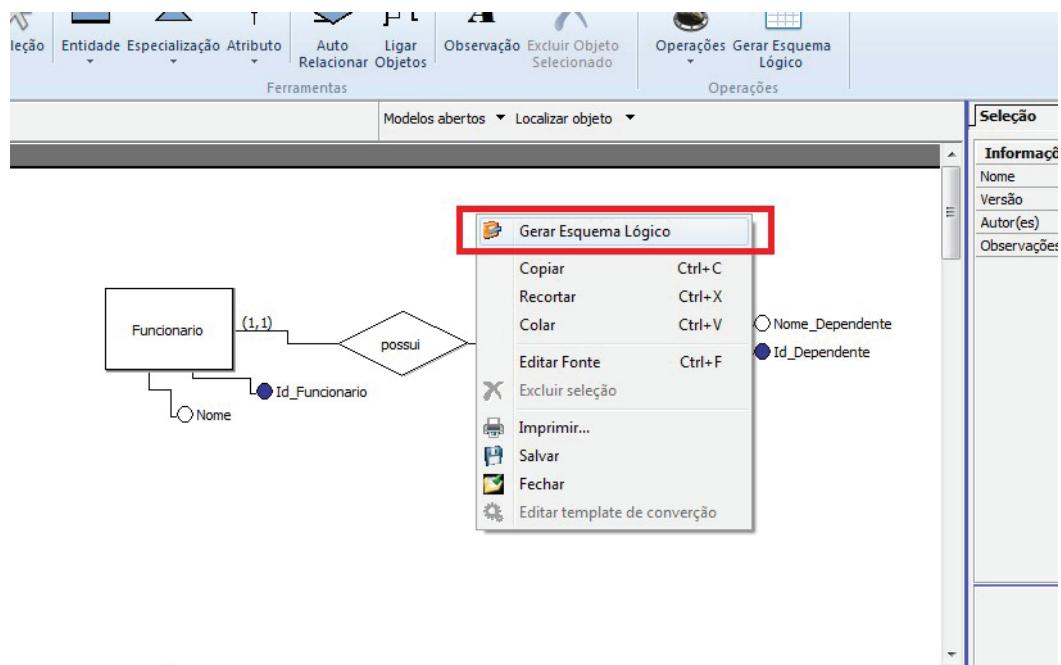


Figura 18

Essa ação irá gerar um esquema lógico que podemos observar na imagem abaixo.

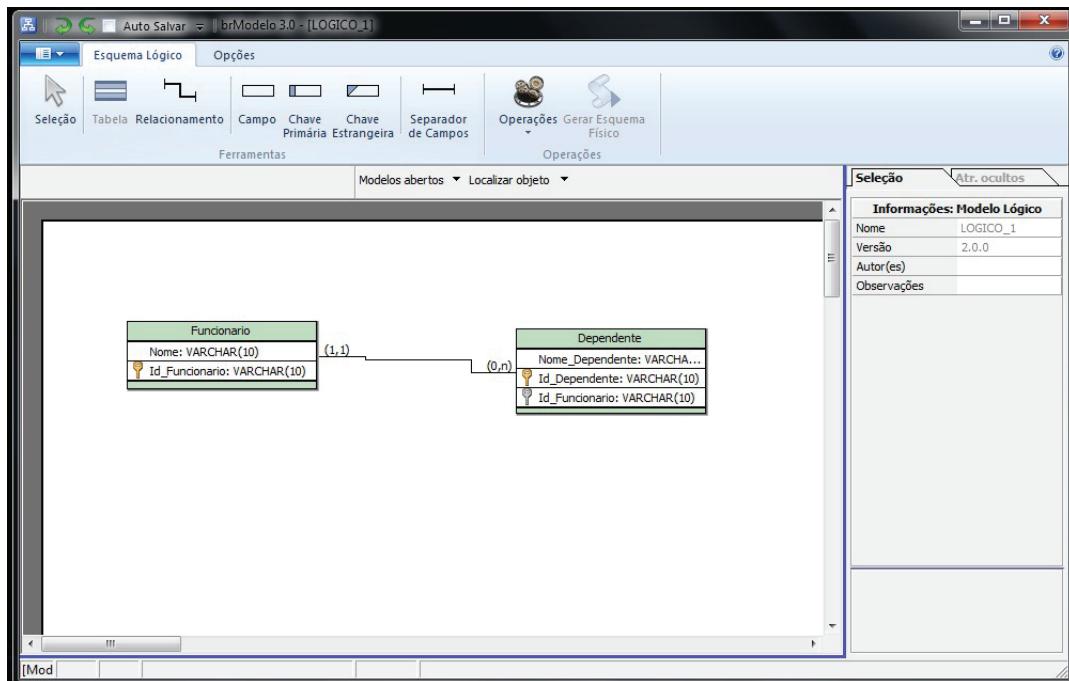


Figura 19

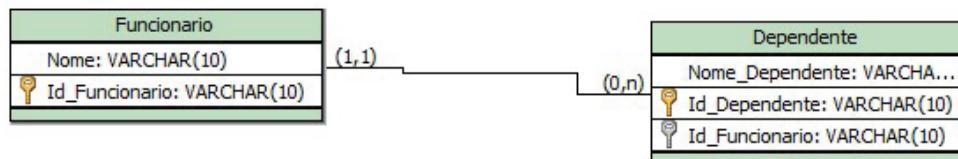


Figura 20

Percebam que, ao criar o modelo lógico, o software **BrModelo** gerou automaticamente a chave estrangeira “***Id_Funcionario***” na entidade **DEPENDENTE**, bem como toda a estrutura de dados do modelo físico (nomes das tabelas, nome de campos e tipo de dados).

É, nessa etapa, que no **BrModelo**, temos a possibilidade de alterar o tipo de dado para determinado Banco de dados.



Importante!

Caso queira gerar um *script* para banco de dados *Oracle*, deve-se alterar os tipos de dados especificamente para *Oracle* e, assim, por diante.

Para nosso estudo, iremos gerar o *script SQL* com os tipos de dados da configuração padrão do **BrModelo**. Esse padrão gera *scripts* com a sintaxe do banco de dados SQLite.



SQLite é o SGBD de código aberto e o mais utilizado nos ambientes *mobile*.
<https://goo.gl/h1lcMc>

Gerando o script SQL

Para gerar o *script SQL* referente ao DER que criamos e o modelo lógico que criamos, até agora, no **BrModelo**, basta clicar com o botão direito do mouse sobre um local vazio no palco. Em seguida, clique na opção “Gerar esquema Físico”.

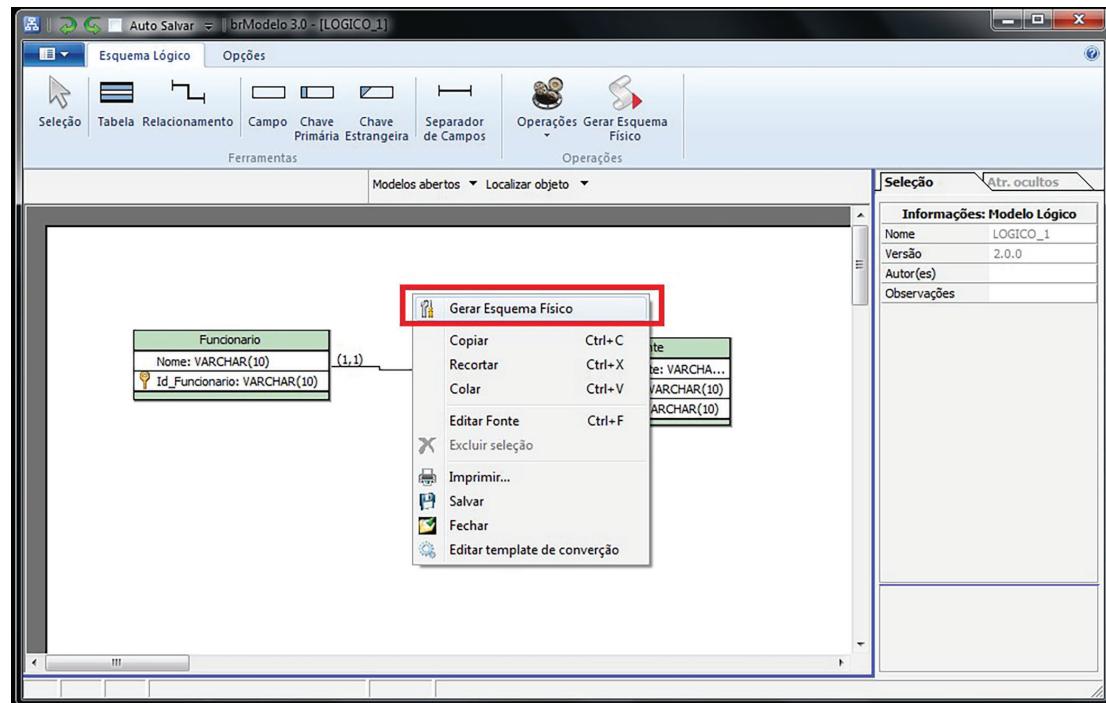


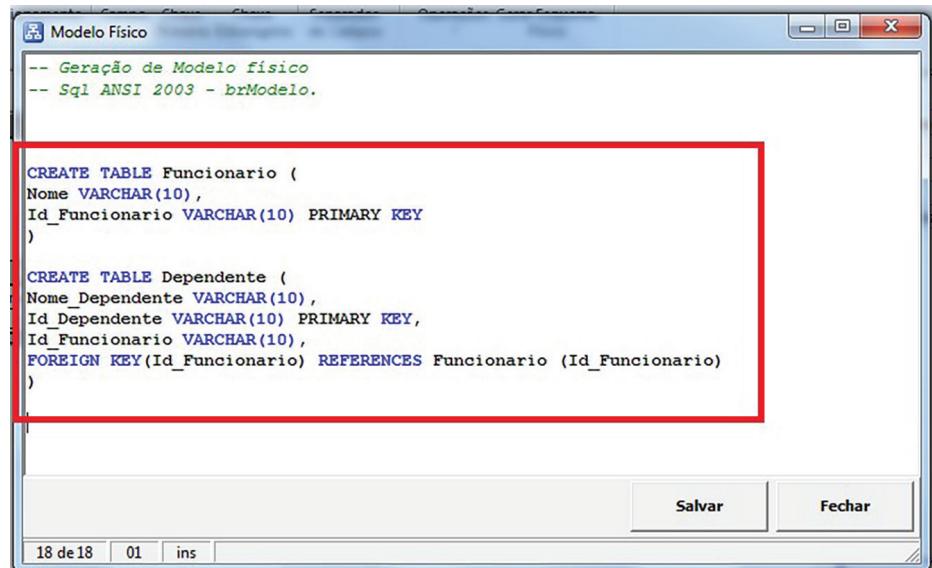
Figura 21

Na tela a seguir, clique no botão “Gerar Modelo Físico”.



Figura 22

Essa ação irá gerar um *script* SQL, que poderá ser salvo em um arquivo do formato **.sql**.



```
-- Geração de Modelo físico
-- Sql ANSI 2003 - brModelo.

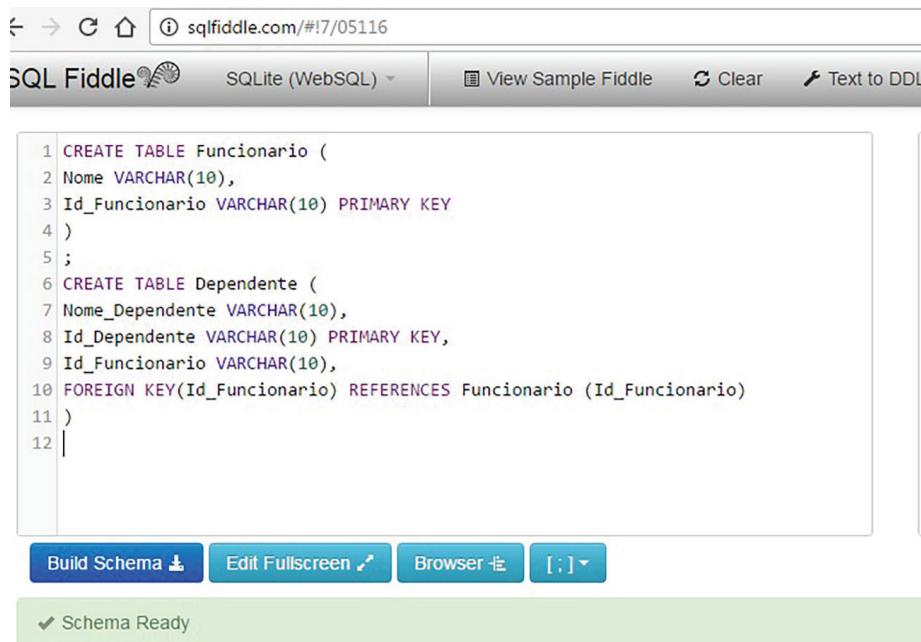
CREATE TABLE Funcionario (
  Nome VARCHAR(10),
  Id_Funcionario VARCHAR(10) PRIMARY KEY
)

CREATE TABLE Dependente (
  Nome_Dependente VARCHAR(10),
  Id_Dependente VARCHAR(10) PRIMARY KEY,
  Id_Funcionario VARCHAR(10),
  FOREIGN KEY(Id_Funcionario) REFERENCES Funcionario (Id_Funcionario)
)
```

Figura 23



Podemos testar o *script* gerado na plataforma SQLFiddle: <http://sqlfiddle.com/>



The screenshot shows the SQL Fiddle interface with the following details:

- URL:** sqlfiddle.com/#!7/05116
- Database:** SQLite (WebSQL)
- Script Content:**

```
1 CREATE TABLE Funcionario (
2   Nome VARCHAR(10),
3   Id_Funcionario VARCHAR(10) PRIMARY KEY
4 )
5 ;
6 CREATE TABLE Dependente (
7   Nome_Dependente VARCHAR(10),
8   Id_Dependente VARCHAR(10) PRIMARY KEY,
9   Id_Funcionario VARCHAR(10),
10  FOREIGN KEY(Id_Funcionario) REFERENCES Funcionario (Id_Funcionario)
11 )
12 |
```
- Buttons:** Build Schema, Edit Fullscreen, Browser, [;]
- Status Bar:** ✓ Schema Ready

Figura 24



O teste do *script* pode ser acessado em: <http://sqlfiddle.com/#!7/05116>

MySQL Workbench

A ferramenta **MySQL Workbench** é uma ferramenta CASE gratuita, que oferece as seguintes funcionalidades:

- **Desenvolvimento SQL:** Funcionalidade para consultas MySQL. Permite que o usuário se conecte a um banco de dados existente, edite e execute consultas SQL.
- **Modelagem de Dados:** Permite a modelagem visual de banco de dados (criação do modelo lógico e físico).
- **Administração de Banco de Dados:** Funcionalidade de administrador do MySQL. Possui uma Interface gráfica para iniciar / parar servidores, criar contas de usuário, editar arquivos de configuração etc.



O MySQL Workbench pode ser baixado pelo link a seguir, a ferramenta pode ser instalada nos sistemas operacionais *Windows*, *Mac OS X*, e nas principais distribuições *Linux*.
<https://goo.gl/YGgkwy>

Para fazer o download da ferramenta, é preciso possuir uma conta *Oracle*, para criar uma conta *Oracle*: <https://goo.gl/s9chx>

No contexto de nossa disciplina, focaremos no desenvolvimento do modelo lógico/físico e, em seguida, geraremos um script SQL.

Visão Geral da Ferramenta

Na tela principal da ferramenta, clique em “File/New Model”.

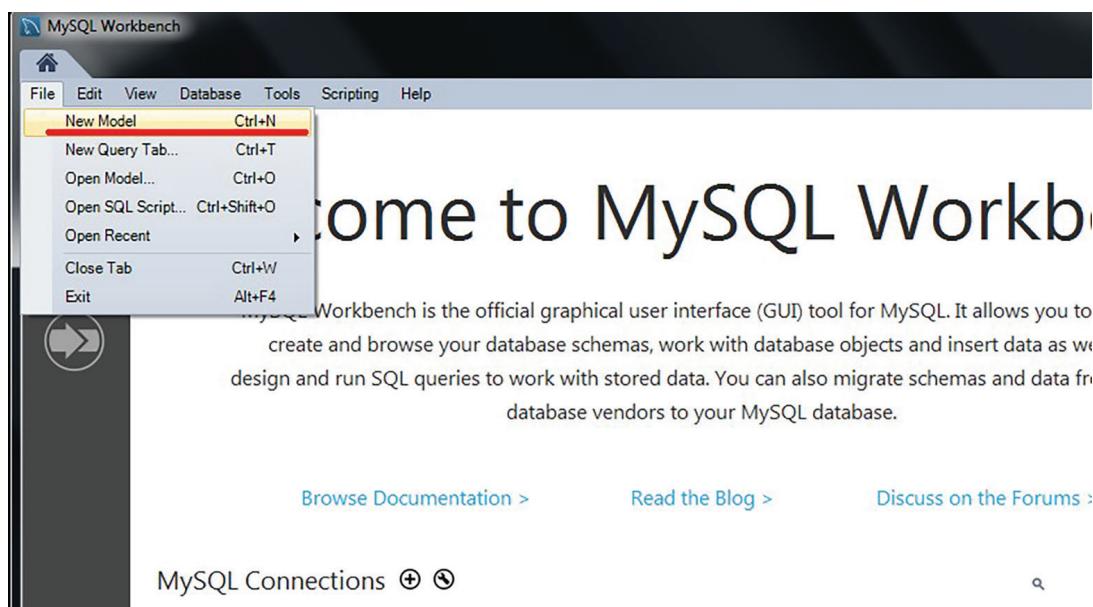


Figura 25

Na tela a seguir, clique no botão “Add Diagram”.

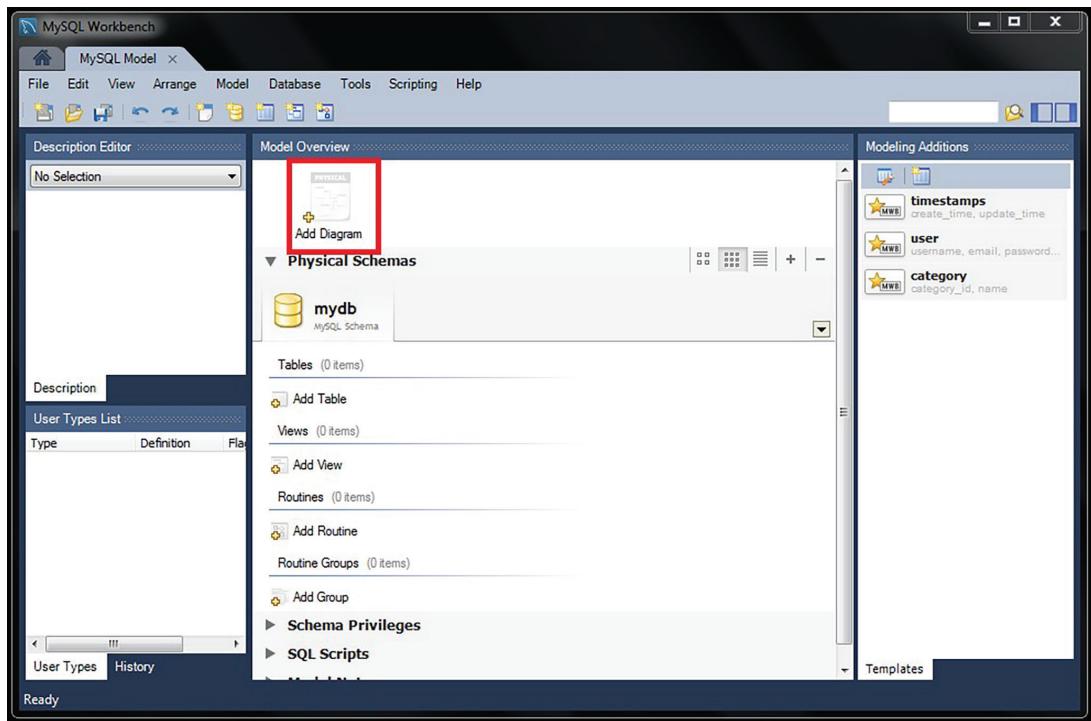


Figura 26

Essa ação leva à tela principal para o desenvolvimento do modelo lógico.

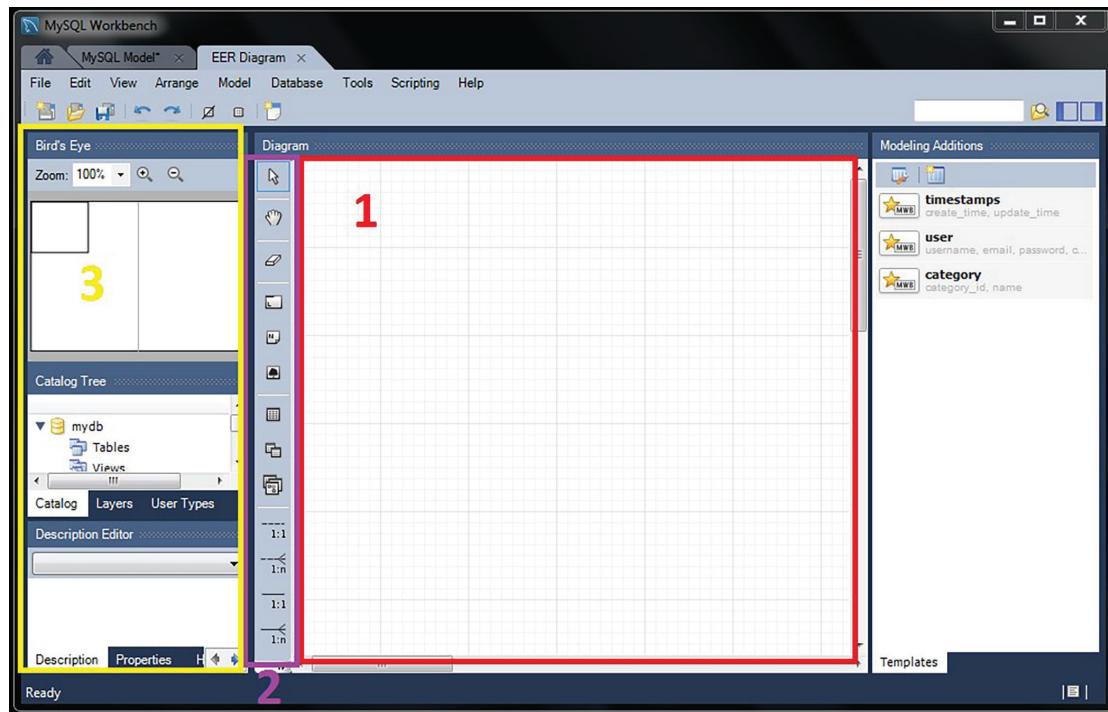


Figura 27

Em destaque, temos três áreas da ferramenta:

- **Área 1.** É o palco do *software*. É, nesse espaço, que é organizado os objetos do DER.
- **Área 2.** Aqui, estão dispostos os principais objetos do DER (entidade, relacionamento, atributos), esses objetos adicionados ao palco do *software*.
- **Área 3.** Refere-se à tela de visualização rápida dos diversos objetos e propriedades do modelo em desenvolvimento.

Criando o Modelo Lógico

Nessa sessão, criaremos um DER levando em consideração o mesmo cenário utilizado anteriormente.

- Um funcionário deve possuir, no mínimo, zero dependente e, no máximo, muitos dependentes.
- Um dependente deve pertencer, no mínimo, a 1 funcionário e, no máximo, 1 funcionário.

Nesse contexto, iremos criar o modelo lógico nos passos a seguir.

Passo 1 – Clique no objeto “New Table” e, em seguida, clique em uma área vazia do palco. Essa ação irá adicionar um objeto “Table”.

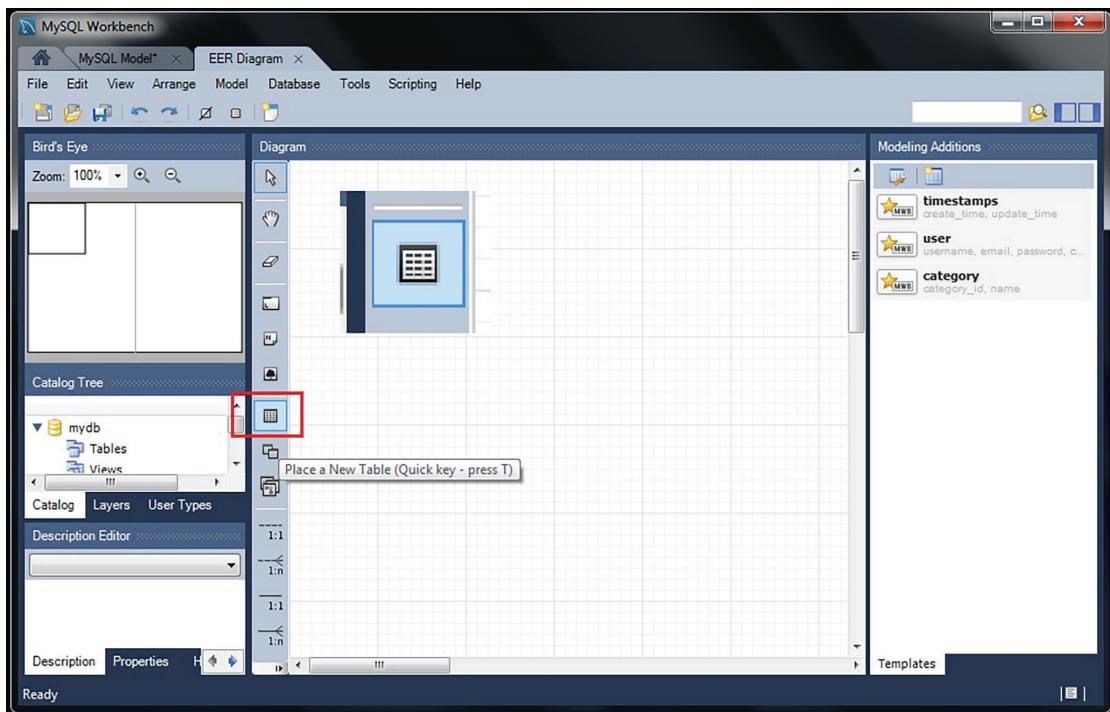


Figura 28

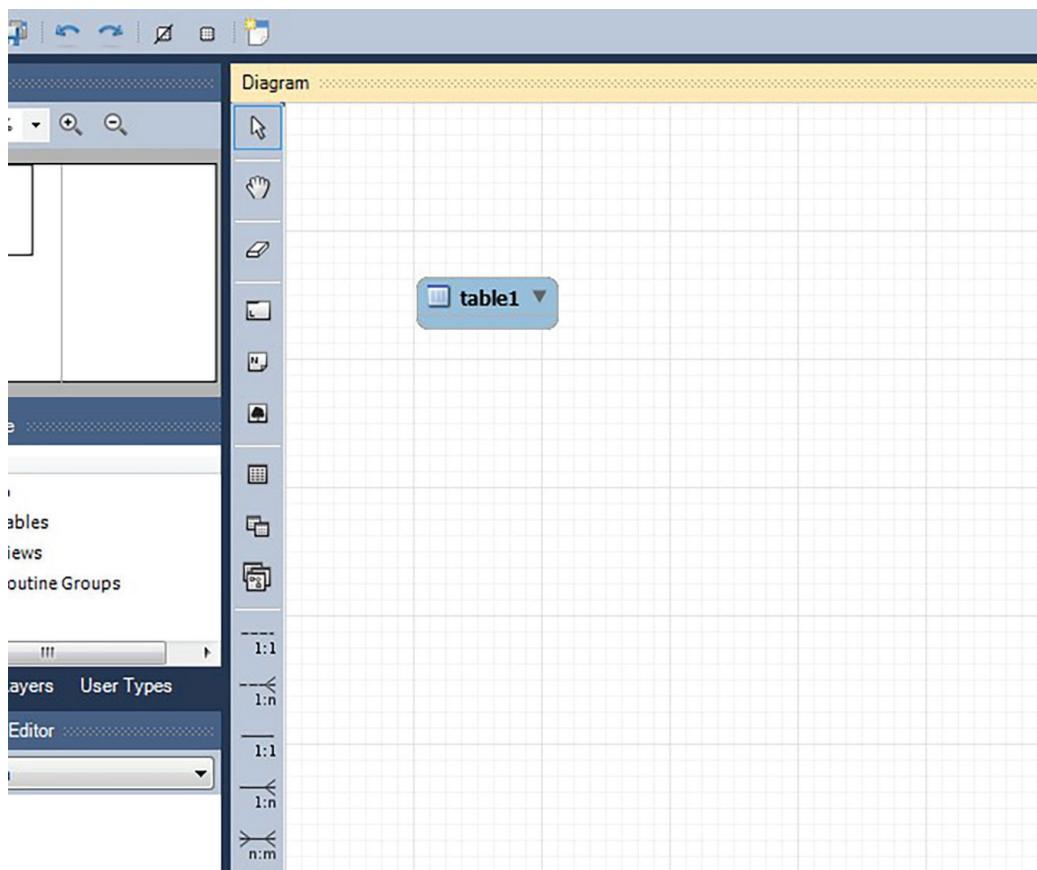


Figura 29

Passo 2 – Clique com o botão esquerdo do mouse duas vezes no objeto “Table1”. Essa ação irá habilitar uma tela de propriedades, nessa tela, altere o nome da propriedade “Table Name” para: **Funcionario**.

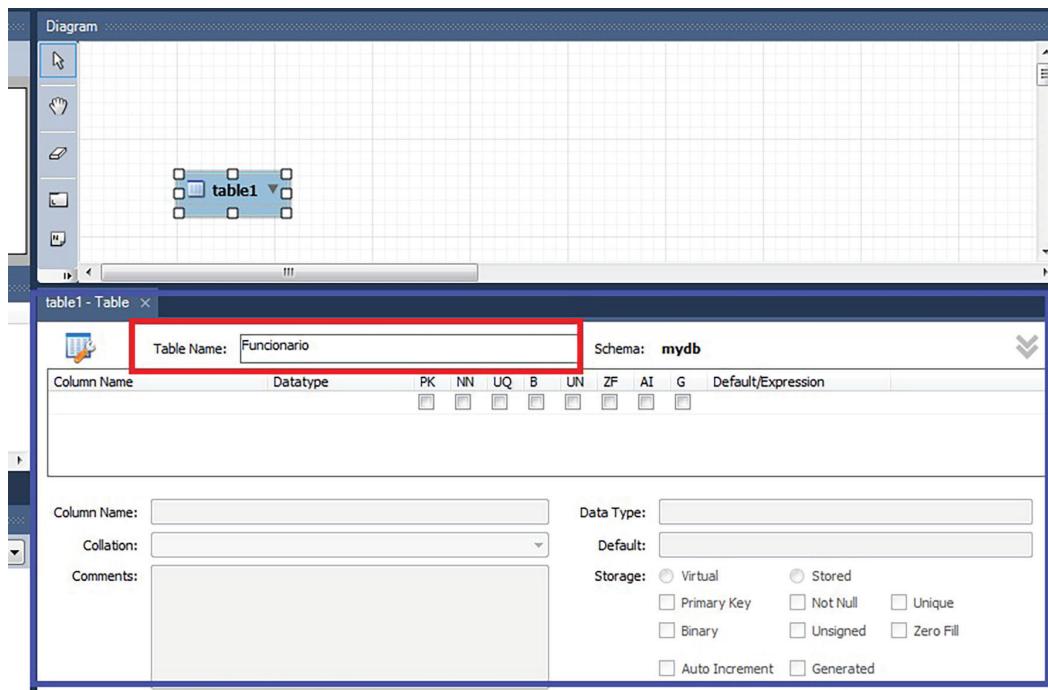


Figura 30

Passo 2 – Adicionando um atributo. Na tela de propriedades, logo abaixo da propriedade “Table Name”, clique duas vezes com o botão esquerdo mouse na célula abaixo da coluna “Column Name”. Em seguida, altere o nome do atributo para: ***id_Funcionario***.

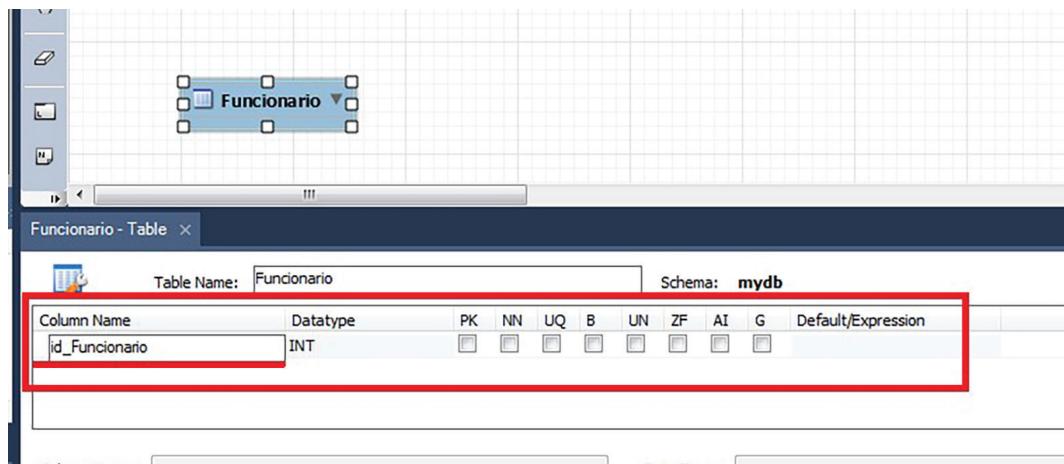


Figura 31

Para cada atributo criado na tabela, existem diversas propriedades, como podemos verificar na imagem abaixo.

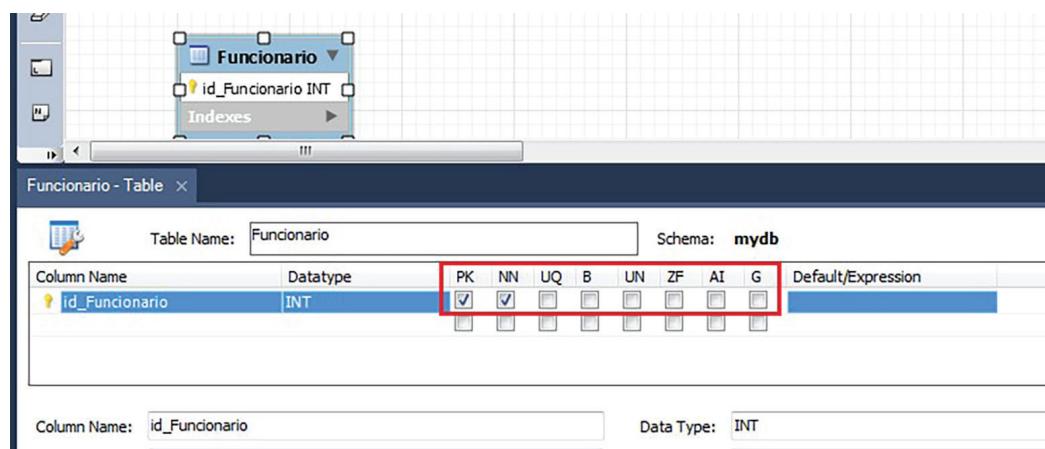


Figura 32

Onde, cada propriedade, quando selecionada, atribui uma funcionalidade ao atributo, essas propriedades são:

- **PK:** Chave primária (O atributo é uma chave primária).
- **NN:** Não nulo (o atributo não pode ficar vazio).
- **UQ:** Único (o valor do atributo deve ser único em toda a tabela)
- **BIN:** Binário (o atributo só aceita valores binários).
- **UN:** Unsigned (não permite valores negativos no atributo).
- **ZF:** Zero fill (preenche os espaços vazios com zero à esquerda até o tamanho padrão do campo).
- **AI:** Autoincremento (o valor do atributo é gerado automaticamente pelo SGBD).

Para concluir, a configuração do atributo “id_Funcionario”, selecione as opções “PK” e “NN”.

Passo 3 – Adicione o atributo “Nome”. Clique duas vezes com o botão esquerdo mouse na célula abaixo do atributo “id_Funcionario”. Em seguida, altere o nome do atributo para: **nome**.

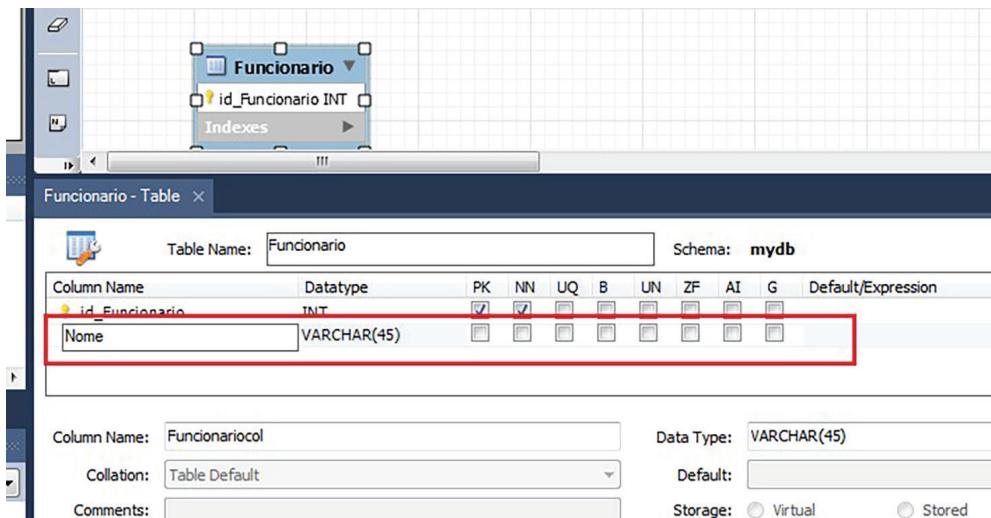


Figura 33

O tipo de dado do campo (coluna *Datatype*) por padrão é VARCHAR(45), para o nosso exemplo não é preciso alterar. Mas, vale salientar, que é nessa propriedade que se pode alterar o tipo e o tamanho do atributo.

Passo 4 – Adicione a tabela **Dependente**. Para isso, clique no objeto “New Table” e, em seguida, clique em uma área vazia do palco. Essa ação irá adicionar um objeto “Table”.

Em seguida, clique com o botão esquerdo do mouse duas vezes no objeto “Table1”. Essa ação irá habilitar uma tela de propriedades, nessa tela, altere o nome da propriedade “Table Name” para: **Dependente**.

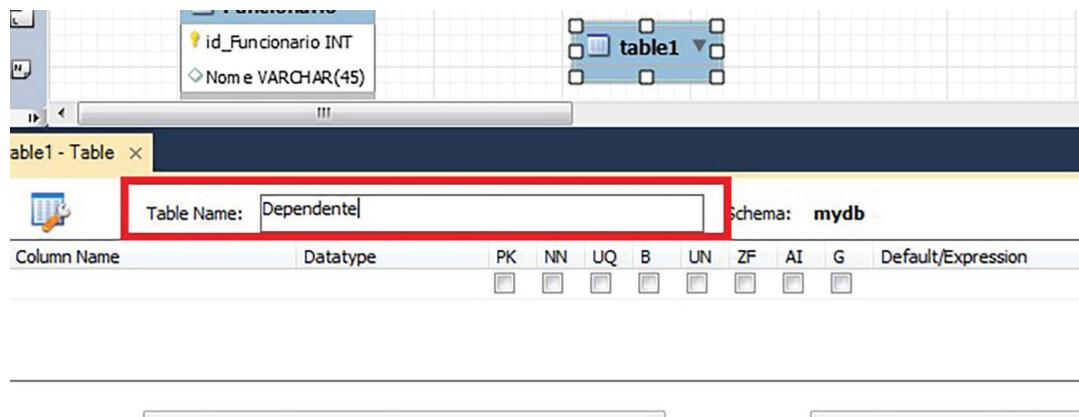


Figura 34

Passo 5 – Adicione o atributo chave. Na tela de propriedades, logo abaixo da propriedade “Table Name”. Clique duas vezes com o botão esquerdo *mouse* na célula abaixo da coluna “Column Name”. Em seguida, altere o nome do atributo para: ***id_Dependente*** e selecione as opções “PK” e “NN”.

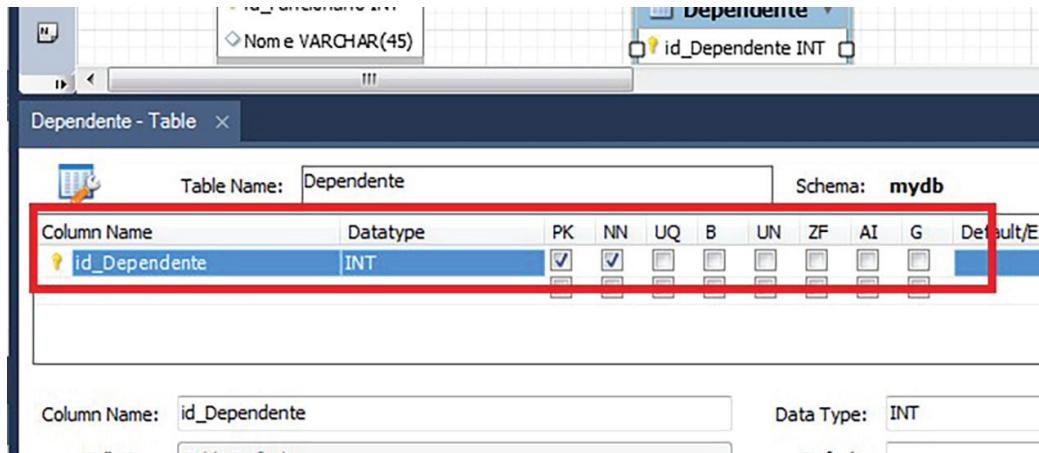


Figura 35

Passo 6 – Adicione o atributo **Nome_Dependente**. Clique duas vezes com o botão esquerdo do *mouse* na célula abaixo do atributo “*id_Funcionario*”. Em seguida, altere o nome do atributo para: ***Nome_Dependente***.

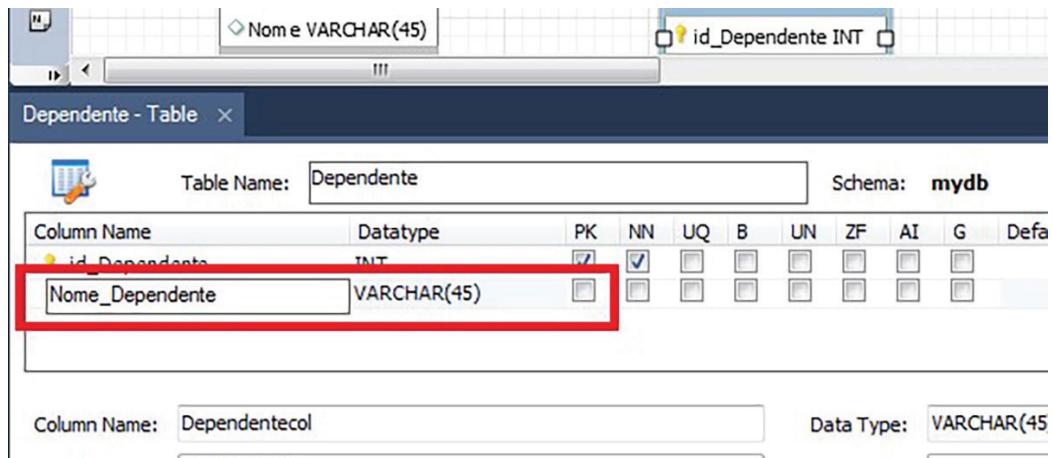


Figura 36

Passo 7 – Adicionando o grau de relacionamento. A ferramenta dispõe de seis (6) possibilidades para configuração do grau de relacionamento entre tabelas, como destacado na imagem abaixo.

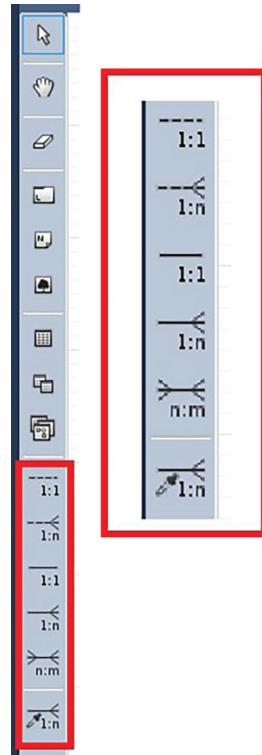


Figura 37

Na sequência (de cima para baixo), esses relacionamentos significam:

- Relacionamento um-para-um (1:1) sem adicionar atributo chave-composto na entidade fraca.
- Relacionamento um-para-muitos (1:n) sem adicionar atributo chave-composto na entidade fraca.
- Relacionamento um-para-um (1:1) adicionando atributo chave-composto na entidade fraca.
- Relacionamento um-para-muitos (1:n) adicionando atributo chave-composto na entidade fraca.
- Relacionamento muito-para-muitos (n:m).

É importante ressaltar que a notação gráfica utilizada pela ferramenta é a notação engenharia de informações ou notação James Martin (MACHADO, 2014).

Recordando o cenário de nosso exemplo, temos um grau de relacionamento (1:n). Onde um funcionário pode possuir diversos dependentes, mas um dependente deve estar vinculado a somente um funcionário.

Nesse contexto, selecione a relação “*one to many identifying relationship*” (Relacionamento um-para-muitos).

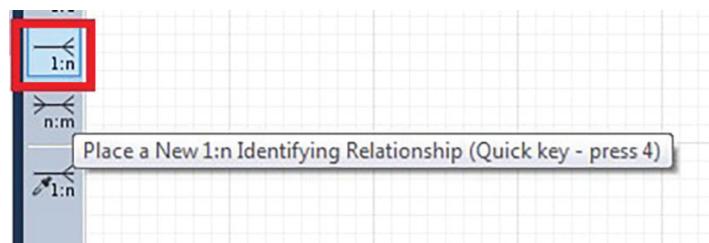


Figura 38

Com a ferramenta selecionada, clique **primeiro** na entidade fraca, nesse caso, clique na tabela DEPENDENTE, em seguida, por último, clique na tabela FUNCIONARIO.

O resultado dessa ação pode ser observado na imagem abaixo.

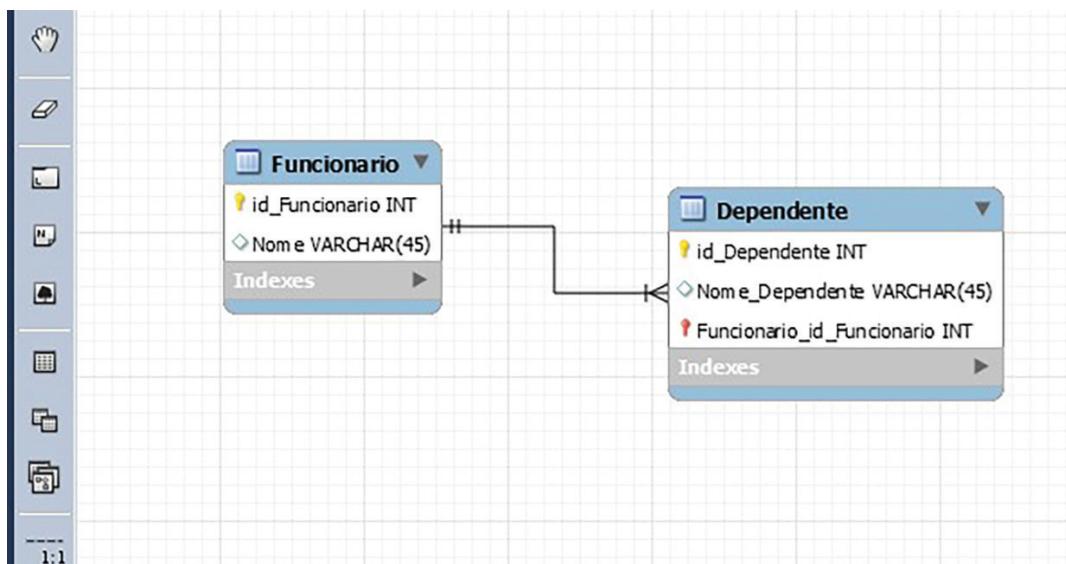


Figura 39

Percebam que, ao configurar o relacionamento, o software criou automaticamente a chave estrangeira “*Funcionario_Id_Funcionario*” na entidade DEPENDENTE.

Gerando o script SQL

Para gerar o script SQL referente ao modelo lógico que criamos, clique no “File\ Export\ Forward Enginner SQL CREATE Script ...”.

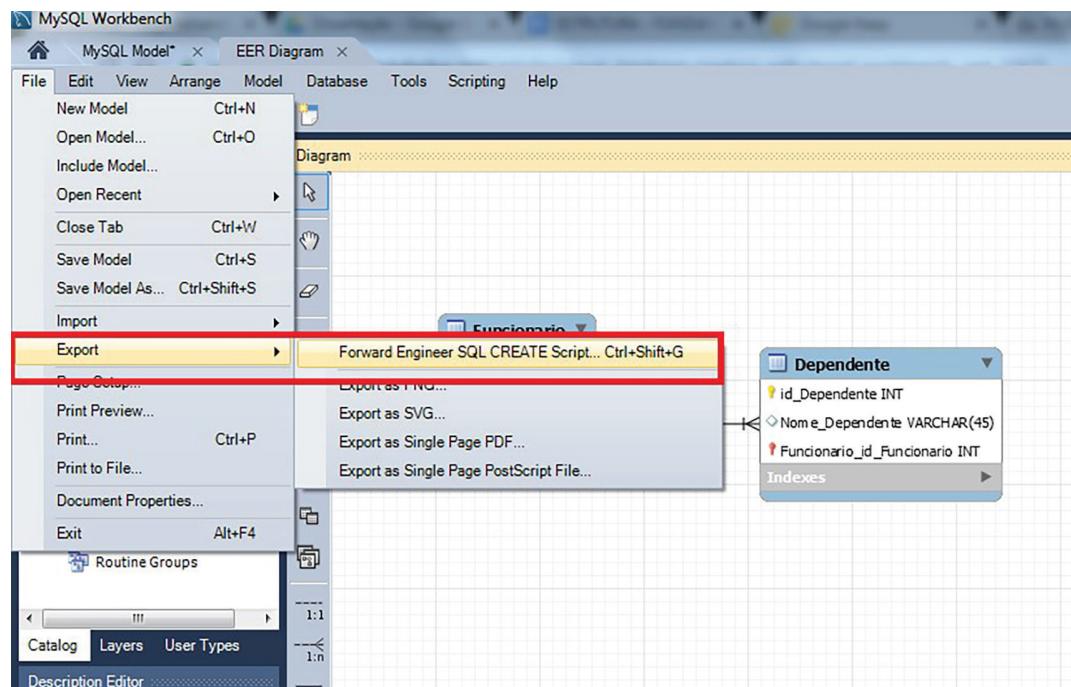


Figura 40

Na tela seguinte, marque as opções: *Omit Schema Qualifier in Object Names*; *Do Not Create Users. Only Export Privileges* e *Don't create view placeholder tables*. Avance, clicando no botão “Next”.

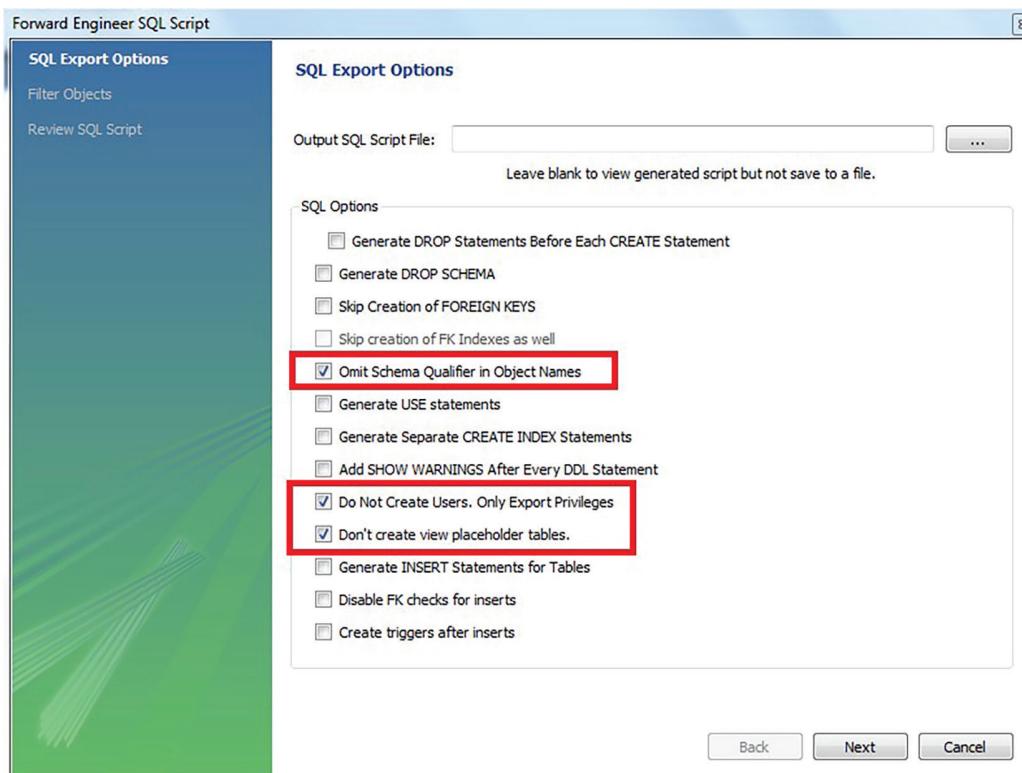


Figura 41

Na tela seguinte, marque a opção “Export MySQL Table Objects” e avance clicando no botão “Next”.

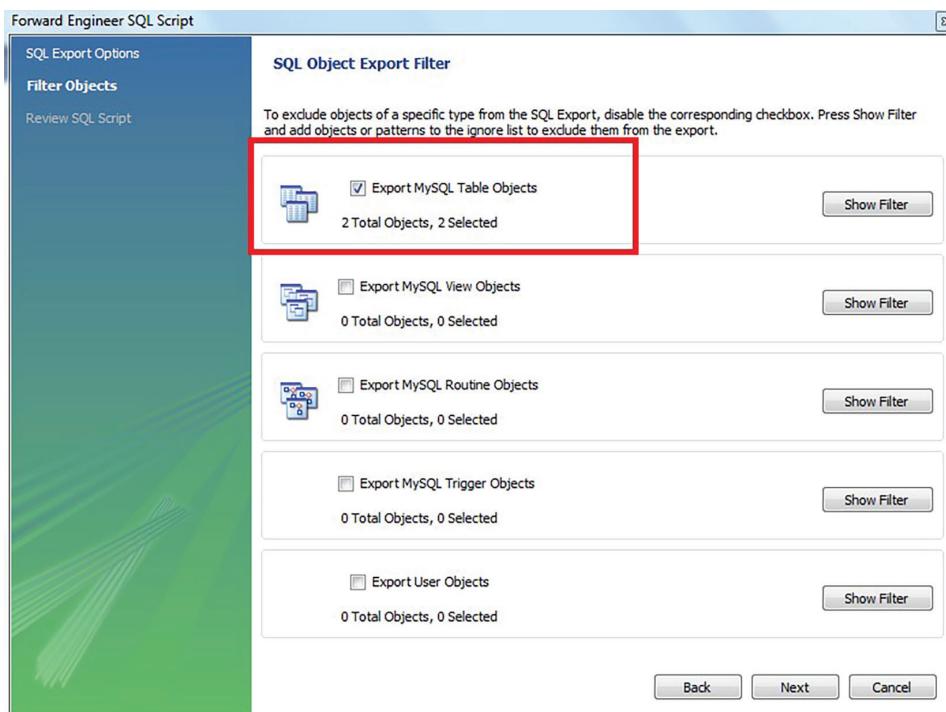


Figura 42

Por fim, é gerado um *script* SQL na sintaxe MySQL para criação das tabelas referente ao modelo lógico criado pela ferramenta. O *script* gerado poderá ser salvo em um arquivo do formato **.sql**.

```

11 -- Schema mydb
12
13
14
15
16
17 CREATE TABLE IF NOT EXISTS `Funcionario` (
18     `id_Funcionario` INT NOT NULL,
19     `Nome` VARCHAR(45) NULL,
20     PRIMARY KEY (`id_Funcionario`)
21 ) ENGINE = InnoDB;
22
23
24
25
26
27 CREATE TABLE IF NOT EXISTS `Dependente` (
28     `id_Dependente` INT NOT NULL,
29     `Nome_Dependente` VARCHAR(45) NULL,
30     `Funcionario_id_Funcionario` INT NOT NULL,
31     PRIMARY KEY (`id_Dependente`, `Funcionario_id_Funcionario`),
32     INDEX `fk_Dependente_Funcionario_idx` (`Funcionario_id_Funcionario` ASC),
33     CONSTRAINT `fk_Dependente_Funcionario`
34         FOREIGN KEY (`Funcionario_id_Funcionario`)
35             REFERENCES `Funcionario` (`id_Funcionario`)
36             ON DELETE NO ACTION
37             ON UPDATE NO ACTION
38 ) ENGINE = InnoDB;
39

```

Figura 43



Podemos testar o *script* gerado na plataforma SQLFiddle <http://sqlfiddle.com>



The screenshot shows the SQL Fiddle interface with the URL sqlfiddle.com/#!9/378711 in the address bar. The database selected is MySQL 5.6. The code area contains the following MySQL DDL script:

```
1 CREATE TABLE IF NOT EXISTS `Funcionario` (
2     `id_Funcionario` INT NOT NULL,
3     `Nome` VARCHAR(45) NULL,
4     PRIMARY KEY (`id_Funcionario`)
5 ) ENGINE = InnoDB;
6
7
8 -----
9 -- Table `Dependente`
10 -----
11 CREATE TABLE IF NOT EXISTS `Dependente` (
12     `id_Dependente` INT NOT NULL,
13     `Nome_Dependente` VARCHAR(45) NULL,
14     `Funcionario_id_Funcionario` INT NOT NULL,
15     PRIMARY KEY (`id_Dependente`, `Funcionario_id_Funcionario`),
```

Below the code are several buttons: Build Schema, Edit Fullscreen, Browser, and [:]. A green bar at the bottom indicates "Schema Ready".

Figura 44



O teste do *script* pode ser acessado em: <http://sqlfiddle.com/#!9/378711>

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Sites

Manual oficial do MySQL Workbench (em Inglês)

<https://goo.gl/oxFJAL>



Livros

Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g

Sandra Puga, Edson França e Milton Goya. Capítulo 7.1 - Introdução à linguagem SQL

Banco de dados: Implementação em SQL, PL/SQL e Oracle 11g

Sandra Puga, Edson França e Milton Goya. Capítulo 4.6 - Notação



Vídeos

Usando o brModelo

<https://youtu.be/dk1-y0PnjuU>

Referências

CÂNDIDO, C. H. **BrModelo 2.0. 2007.** Disponível em: <[http://sis4.com/brModelo/](http://sis4.com.brModelo/)>. Acesso em abr. 2017.

HEUSER, C. A. **Projeto de banco de dados.** 6. ed. Porto Alegre: Bookman, 2010.

MACHADO, F. N. R. **Banco de Dados:** projeto e implementação. 3. ed. São Paulo, SP: Érica, 2014.



Cruzeiro do Sul Virtual
Educação a Distância

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo - SP - Brasil
Tel: (55 11) 3385-3000



Cruzeiro do Sul
Educacional