

Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



**SGB\_MANUAL\_WorkBench**

**Versão 1.0**



## Histórico de Revisões

Nome	Alterações	Data	Versão
Allan Vieira Ribeiro	Criação do Documento	22/10/2012	0.1
Inael Rodrigues	Criação do conteúdo	01/11/2012	0.2
Bruno Marquete	Ajustes finais do documento	01/11/2012	1.0

Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



# Sumário

1 Introdução	4
2 Entendendo as edições do MySQL Workbench	4
3 Instalando o MySQL Workbench (MW) no Windows	5
4 Modelando um sistema exemplo com o MySQL Workbench	6
5 Especificando os relacionamentos entre as entidades	11
6 Sincronização e engenharia reversa da base de dados	14
7 Criando objetos avançados e inserindo registros em tabelas através do MySQL Workbench	20
8 Referência	25

Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



## 1 Introdução

Este manual descreve a ferramenta de modelagem de dados em MySQL, chamada MySQL Workbench, apontando suas principais funcionalidades. Essa ferramenta fornece mecanismos simples para modelar, construir, gerenciar e manter base de dados e simplifica a execução de tarefas complexas de administração a ser exercida pelo DBA.

## 2 Entendendo as edições do MySQL Workbench

O MySQL Workbench é oferecido em duas edições, a OSS (Community Edition – Edição da comunidade) e a SE (Standard Edition – Edição padrão). A OSS é a edição disponível para a comunidade, e também é a base da SE, e de todas as possíveis edições que eventualmente forem disponibilizadas no futuro. A MySQL/Sun explica no site da ferramenta que todas as melhorias feitas no framework básico do MySQL Workbench serão aplicadas na edição OSS, já que, como dito anteriormente, esta é a base de todos os outros produtos. Vale destacar que a ferramenta possui um mecanismo de plugins que possibilita a criação de novas extensões para o produto. Desta forma, é possível incluir recursos novos a ela de acordo com as necessidades específicas de cada usuário ou organização.

A edição SE é a versão comercial do produto que inclui módulos e plugins extras que disponibilizam recursos avançados para seus usuários. Alguns dos diferenciais que merecem destaque são engenharia reversa e sincronização de dados, que permitem a sincronização do



modelo diretamente com o banco MySQL e vice-versa. Esta mesma operação utilizando a edição OSS só pode ser feita através da geração de scripts SQL.

Outro recurso importante presente na versão SE é a validação dos modelos de dados, o que é de extrema importância para gerenciar a migração de bases entre versões diferentes do MySQL. Além disto, nesta versão há recursos para a geração e impressão de documentação do modelo tanto em HTML quanto em formato textual. No endereço [http://dev.mysql.com/workbench/?page\\_id=11](http://dev.mysql.com/workbench/?page_id=11) encontra-se um quadro comparativo entre estas duas edições salientando as diferenças de recursos existente entre elas.

É importante destacar que a edição OSS é disponível sob a licença GPL e pode ser utilizada gratuitamente, estando disponível para download a partir do endereço <http://dev.mysql.com/downloads/workbench/5.0.html>. Por outro lado, a edição SE está disponível sob uma licença comercial, e o seu uso está condicionado ao pagamento de US\$ 99,00 por cópia do produto, por um período de 1 ano de utilização. Este custo é bastante razoável se formos considerar os grandes benefícios agregados à ferramenta.

Vale destacar que todas as ilustrações e exemplos presentes neste artigo foram desenvolvidos utilizando a versão GPL da ferramenta, ou seja, a edição OSS. Portanto, determinados recursos não poderão ser apresentados em função de não estarem disponíveis nesta versão.

### 3 Instalando o MySQL Workbench (MW) no Windows

O MW está disponível em formato binário para o sistema operacional Windows, e também pode ser obtido o seu código fonte a partir do site do produto. O procedimento de instalação é bastante simples. Há uma interface que conduz o usuário através das etapas de configuração dos parâmetros básicos para o funcionamento do sistema. Na Figura 1 encontra-se a tela para a instalação detalhada (custom) do produto, onde podem ser escolhidos os módulos a serem instalados, além de definir o local de instalação do mesmo.

Figura 1. Interface para a instalação completa do MySQL Workbench.



Como se pode observar na Figura 1, existe a opção de instalação da base do sistema (MySQL Workbench core), que contém a base obrigatória do sistema. Há também a interface de linha de comando para Windows (MySQL GRT Shell), utilizada para a edição de scripts para o desenvolvimento de plugins. Finalmente, pode-se optar pela instalação do recurso para atualização do sistema (Update feature), que auxiliará a utilizar as novas releases do sistema. Além disso, é possível alterar o diretório padrão de instalação do sistema, para colocá-lo em um diretório mais adequado ao esquema de diretórios do usuário. Feito isto, basta acionar o botão Next e aguardar a cópia dos dados para o seu disco local, finalizando o processo de instalação.

#### 4 Modelando um sistema exemplo com o MySQL Workbench

Para ilustrar os principais recursos disponíveis no MW, será considerada a modelagem de uma base de dados para armazenar os dados de uma olimpíada. Neste caso, temos um sistema composto de três entidades básicas que são os atletas, as modalidades esportivas, e os times. Além disso, é preciso manter registros do calendário de jogos, bem como os confrontos entre equipes e os resultados obtidos por elas. Lembrando que os atletas podem se organizar em times que jogam entre si, mas que esta não é uma condição obrigatória. Além disso, um atleta necessariamente tem que possuir uma modalidade esportiva, tais como, natação, futebol, etc.

A Figura 2 ilustra a tela principal do MW que permite a criação do modelo lógico de dados, bem como os objetos físicos do banco de dados, como tabelas, stored procedures e views.

"

Figura 2. Tela principal do MySQL Workbench.

Para criar o modelo de dados, basta acionar a opção Add Diagram (adicionar diagrama), existente na área identificada como EER Diagrams (Diagramas EER) dentro da área MySQL Model (Modelo MySQL). Vale destacar que todos os modelos existentes estarão disponíveis para uso nesta mesma área do sistema. Ao acionar a opção de criação de modelos, o sistema apresentará a tela exibida na Figura 3.

Figura 3. Interface para a edição dos modelos de dados.



Na tela de edição de modelos encontra-se à esquerda a paleta com os principais elementos para a criação de um diagrama ER (Entidade-Relacionamento – ver Nota DevMan 1). Estes itens são as tabelas e os relacionamentos (1 para 1, 1 para muitos e muitos para muitos).

Iniciando o modelo do sistema proposto para controlar uma olimpíada, considere inicialmente a criação da tabela de atletas. Para isto, deve-se acionar o ícone da tabela na paleta à esquerda ou pressionar a tecla T. Feito isto, basta clicar novamente na área de desenho e uma tabela será adicionada, conforme ilustra a Figura 5.

"

Figura 5. Adicionando uma nova tabela ao diagrama ER.

Uma vez inserida a tabela, o próximo passo é definir a sua estrutura de colunas. Para a manipulação da tabela, basta clicar sobre a mesma (table1) usando o botão direito do mouse e clicar na opção Edit table (editar tabela). Neste caso, aparecerá na parte inferior da janela uma tela contendo os principais elementos da tabela a serem definidos, tais como colunas, índices e chaves estrangeiras.

A Figura 6 ilustra a interface para a definição do nome da tabela, o Storage Engine a ser utilizado (MyISAM, InnoDB, dentre outros), o conjunto de caracteres e um comentário para a documentação da mesma.

Figura 6. Definindo os o nome da tabela e seus atributos gerais.

Concluída a primeira etapa, podemos definir os campos ou atributos da tabela acionando a aba Columns (colunas) na parte inferior da Figura 6. Cada coluna de uma tabela é composta de um nome, um tipo de dado (inteiro, textos, data, etc), as restrições quanto à inserção de valores (NULL, NOT NULL) e ainda conter números formando uma sequência incrementada automaticamente (Auto-Increment). Vale ressaltar que a ferramenta disponibiliza uma lista com todos os tipos de dados do MySQL, o que facilita a definição dos atributos da tabela em questão.

A Figura 7 apresenta a inclusão de três campos: o nome do atleta que é um texto de 45 caracteres, a data de nascimento que é do tipo data e o código de identificação do atleta nas olimpíadas que é um código seqüencial inteiro representando a chave primária da tabela, e é usado para a identificação única de cada atleta cadastrado no sistema.



"

Figura 7. Definindo os campos da tabela Atletas.

Considerando que o sistema permitirá a pesquisa de atletas por nome, é importante a inclusão de um índice para este atributo a fim de agilizar esta busca. Para isto, basta acionar a aba Indexes (Índices) e adicionar o novo índice conforme ilustra a Figura 8.

"

Figura 8. Criando um índice na coluna nome do atleta.

Concluída a criação da primeira entidade do modelo, os passos descritos anteriormente podem ser repetidos para criar as tabelas de modalidades esportivas e times. Ao final de todo o processo, haverá um cenário como o exibido na Figura 9.

"

Figura 9. Modelo contendo as entidades Atletas, times e Modalidades esportivas.

## 5 Especificando os relacionamentos entre as entidades

Na seção anterior foi concluída a primeira etapa da modelagem, que é a definição das entidades ou tabelas e seus atributos.

No MW existem quatro opções disponíveis para definição de relacionamentos, que podem ser encontradas na barra à esquerda da tela principal. O primeiro deles é o relacionamento 1:N não identificado (1:N Non-Identifying Relationship), utilizado para uma relação 1 para muitos, onde pode haver registros apenas no lado 1 da relação. Utilizando uma outra terminologia, este relacionamento é conhecido também como um relacionamento fraco. Logo abaixo existe o relacionamento 1 para 1 não identificado, que segue a mesma lógica do anterior, porém há apenas 1 registro em cada tabela participando da relação. Em seguida há o relacionamento 1:N identificado (1:N Identifying Relationship), onde neste caso devem existir registros em ambas as tabelas que participam da relação, conhecido também como um relacionamento forte. O mesmo se aplica ao próximo relacionamento que é o 1 para 1 identificado (1:1 Identifying). A última opção é o relacionamento de muitos para muitos (N:M), onde neste caso é necessário criar uma tabela intermediária contendo as chaves primárias das tabelas envolvidas no relacionamento, e eventualmente, outros atributos da relação. A Tabela 1 ilustra a paleta com as opções de criação



dos relacionamentos apresentados anteriormente.

Ícone	Significado
	Relacionamento do tipo 1:N não identificado
	Relacionamento do tipo 1:1 não identificado
	Relacionamento do tipo 1:N identificado
	Relacionamento do tipo 1:1 identificado
	Relacionamento do tipo N:M

**Tabela 1.** Ícones para a definição dos relacionamentos entre tabelas.

Ícone	Significado
	Relacionamento do tipo 1:N não identificado
	Relacionamento do tipo 1:1 não identificado
	Relacionamento do tipo 1:N identificado
	Relacionamento do tipo 1:1 identificado
	Relacionamento do tipo N:M

**Tabela 1.** Ícones para a definição dos relacionamentos entre tabelas.

Neste momento é possível definir os relacionamentos entre as tabelas propostas no modelo. Na definição inicial, um atleta pode ou não estar associado a um time, e os times são compostos de vários atletas. Portanto, temos um relacionamento 1 para muitos fraco, pois podem haver atletas que não participam de times (modalidades individuais como natação, por exemplo). Assim, para criá-lo devemos usar o primeiro botão apresentado na Tabela 1 ou pressionar a tecla 1. Feito



isto, basta clicar na tabela com várias ocorrências – lado muitos da relação - neste caso seria a tabela Atletas, e em seguida no lado 1, neste caso a tabela Times. Assim, a ferramenta criará automaticamente a chave estrangeira e colocará uma ligação entre as tabelas, identificando o relacionamento entre elas. A Figura 10 ilustra este cenário.

Figura 10. Criando um relacionamento 1:N fraco entre atletas e times.

Como pôde ser observado a partir da figura anterior, a linha de ligação é exibida em linhas pontilhadas, o que identifica um relacionamento fraco, como proposto anteriormente.

Avançando na definição do modelo, é preciso criar a relação entre os atletas e as modalidades esportivas. Neste caso, todo atleta deve obrigatoriamente ter uma, e somente uma, modalidade associada. Assim, podem existir vários atletas em uma mesma modalidade, mas não há atletas sem modalidades, o que indica uma relação forte entre estas entidades. Desta forma, deve-se utilizar o relacionamento 1:N Identifying Relationship (Relacionamento 1:N identificado). Para criar esta relação, devemos escolher o terceiro botão apresentado na Tabela 1 ou apertar a tecla 3. Enfim, basta clicar primeiramente na tabela com vários registros e a seguir na tabela correspondente ao lado 1 da relação. Assim, o relacionamento será criado como ilustra a Figura 11.

"

Figura 11. Criando um relacionamento 1:N forte entre atletas e modalidades esportivas.

Como se vê na figura anterior, foi utilizada uma linha preenchida para ligar as entidades, indicando um relacionamento obrigatório (forte) entre elas.

Finalmente, é preciso registrar as partidas ou disputas entre os times, sendo que cada time joga com outros vários times. Neste contexto, temos um relacionamento do tipo muitos para muitos da entidade Times com ela mesma, ou seja, um auto-relacionamento. Para conceber este cenário, aciona-se a opção n:m Identifying (n:m Identificado) que é a quinta opção na Tabela 1 ou a pressiona-se a tecla 5. Feito isto, basta dar dois cliques (pausadamente) sobre a tabela Times para que a tabela exibida na Figura 12 seja construída automaticamente. O nome desta nova tabela é atribuído pela ferramenta, e o mesmo foi alterado para Partidas através da edição da tabela, que pode ser acionada com um clique com o botão direito sobre a tabela, e então acionando o botão Edit table, como ilustra a Figura 6.

"



Figura 12. Criando um relacionamento muitos para muitos entre os times.

Como se pode notar, a ferramenta fez todo o trabalho de criar a tabela que identifica o relacionamento (Partidas), bem como as chaves estrangeiras, facilitando bastante na construção da base de dados. Após isso, foram adicionados outros atributos a esta tabela que são importantes para armazenar informações sobre as partidas realizadas em uma olimpíada. Como dito anteriormente, esta alteração foi feita manualmente, ou seja, pelo usuário da ferramenta, através do editor de tabela exibido na Figura 7.

Um detalhe importante que deve ser considerado é que a ferramenta executou automaticamente várias tarefas ao longo da construção da base de dados que manualmente seriam onerosas e susceptíveis a erros. Por exemplo, todas as chaves estrangeiras e a tabela de relacionamento foram criadas pelo próprio MW sem nenhuma necessidade do usuário escrever códigos SQL para isto. Vale destacar que à direita há a exibição do catálogo da base de dados contendo todos os objetos definidos no modelo lógico, facilitando assim a navegação entre eles onde se tem modelos complexos contendo muitos objetos.

Neste ponto, o modelo está definido e já pode ser usado para construir e povoar uma base de dados no MySQL. Os detalhes desta operação são apresentados na seção seguinte.

## 6 Sincronização e engenharia reversa da base de dados

Uma das grandes vantagens e necessidades de uma ferramenta de modelagem de dados é criar um mecanismo simples que permita manter o modelo de dados lógico sincronizado com a base de dados física. Desta forma, toda alteração no modelo pode ser imediatamente aplicada à base de dados armazenada no MySQL. Desta forma, minimiza-se as chances de erros de operação, que poderiam acarretar a perda de sincronismo entre a base de dados física e o modelo lógico.

No MW edição SE, este sincronismo entre modelo lógico e com os bancos no MySQL e a geração de um modelo lógico a partir de uma base de dados existente (que é chamada de engenharia reversa) podem ser feitos através de uma conexão direta com o servidor MySQL. Esta operação facilita consideravelmente o trabalho, uma vez que nenhum passo intermediário se faz necessário. Na versão OSS, utilizada na elaboração deste artigo, estas opções estão desabilitadas como ilustra a Figura 13.

Figura 13. Opções de sincronização e engenharia reversa não disponíveis na versão OSS.

Nesta versão OSS, apesar de não ser possível a conexão direta com a base de dados, é possível gerar um script SQL contendo os comandos DDL para a definição dos objetos do banco. Este



arquivo então pode ser executado no MySQL para a criação desta base de dados. Para a geração deste script, basta acionar a opção de exportação de dados, conforme ilustra a Figura 14.

"

Figura 14. Geração de script SQL a partir do modelo lógico.

Neste momento, a ferramenta solicitará que o usuário informe o arquivo onde serão gravados os comandos DDL. Após esta etapa, o sistema apresentará a tela exibida na Figura 15 para que se possa informar quais objetos devem ser exportados, ou seja, quais tabelas, usuários, views, stored procedures, funções e triggers.

Figura 15. Seleção dos objetos a serem exportados para o arquivo SQL.

A opção Detailed Selection (Seleção detalhada) permite selecionar individualmente os objetos a serem exportados. Por exemplo, pode-se exportar apenas uma tabela em vez de todas as tabelas existentes no modelo.

A Listagem 1 contém o script SQL que representam os objetos definidos no modelo criado anteriormente para o sistema de gerenciamento dos dados de olimpíadas.

Listagem 1. Listagem dos comandos SQL DDL para a criação da base de dados do sistema de olimpíadas.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
```

USE `mydb`;

-----  
-- Table `mydb`.`Times`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`Times` (  
  `idTimes` INT NOT NULL AUTO_INCREMENT ,  
  `nome` VARCHAR(45) NOT NULL ,  
  PRIMARY KEY (`idTimes`))  
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`Modalidades`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`Modalidades` (  
  `idModalidades` INT NOT NULL AUTO_INCREMENT ,  
  `nome` VARCHAR(45) NOT NULL ,  
  `sexo` ENUM('Masculino','Feminino') NOT NULL ,  
  PRIMARY KEY (`idModalidades`))  
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`Atletas`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`Atletas` (  
  `idAtletas` INT NOT NULL AUTO_INCREMENT ,  
  `nome` VARCHAR(45) NOT NULL ,  
  `dataNascimento` DATE NOT NULL ,  
  `Times_idTimes` INT NULL ,  
  `Modalidades_idModalidades` INT NOT NULL ,  
  PRIMARY KEY (`idAtletas`, `Modalidades_idModalidades`),  
  INDEX nome_idx (`nome` ASC) ,  
  INDEX fk_Atletas_Times (`Times_idTimes` ASC) ,  
  INDEX fk_Atletas_Modalidades (`Modalidades_idModalidades` ASC) ,  
  CONSTRAINT `fk_Atletas_Times`  
    FOREIGN KEY (`Times_idTimes` )  
    REFERENCES `mydb`.`Times` (`idTimes` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Atletas_Modalidades`  
    FOREIGN KEY (`Modalidades_idModalidades` )  
    REFERENCES `mydb`.`Modalidades` (`idModalidades` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `mydb`.`Partidas`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`Partidas` (
```

```
`Times_idTimes` INT NOT NULL ,  
`Times_idTimes1` INT NOT NULL ,  
`data` DATE NOT NULL ,  
`pontos_time1` FLOAT NOT NULL ,  
`pontos_time2` FLOAT NOT NULL ,  
PRIMARY KEY (`Times_idTimes`, `Times_idTimes1`),  
INDEX fk_Times_has_Times_Times (`Times_idTimes` ASC) ,  
INDEX fk_Times_has_Times_Times1 (`Times_idTimes1` ASC) ,  
CONSTRAINT `fk_Times_has_Times_Times`  
FOREIGN KEY (`Times_idTimes` )  
REFERENCES `mydb`.`Times` (`idTimes` )  
ON DELETE NO ACTION  
ON UPDATE NO ACTION,  
CONSTRAINT `fk_Times_has_Times_Times1`  
FOREIGN KEY (`Times_idTimes1` )  
REFERENCES `mydb`.`Times` (`idTimes` )  
ON DELETE NO ACTION  
ON UPDATE NO ACTION);
```

Este script possui todas as instruções SQL para a definição das tabelas, relacionamentos e chaves estrangeiras previstas no modelo. Ele pode ser aplicado imediatamente no MySQL para a criação da base, pois os comandos SQL já estão prontos e isso elimina as chances de erros de sintaxe.

## 7 Criando objetos avançados e inserindo registros em tabelas através do MySQL Workbench

Além das opções para a edição gráfica de tabelas e relacionamentos, o MW fornece uma interface amigável para a inserção de dados nas tabelas do sistema. Isto pode ser feito através de uma interface simples para a manipulação dos registros, como aquela mostrada na Figura 6 para a



criação de modelos ER. Além disto, há também um editor para a criação de rotinas (procedures, funções e triggers) e views. A grande vantagem deste editor é a geração do template com os comandos básicos para a criação destas estruturas, e o corretor de sintaxe, que facilita a identificação de erros na digitação dos comandos.

Para a inserção de dados em uma tabela, basta clicar sobre a mesma e acionar o botão Open Editor (Abrir Editor), e então realizar a entrada de dados conforme ilustra a Figura 16.

Figura 16. Inserção de modalidades esportivas utilizando o editor de registros.

Ao concluir a entrada de dados, a ferramenta apresentará os comandos INSERTs para a inserção dos dados na tabela. Vale ressaltar que na interface ambos os registros ficaram com o ID igual a zero para que o código auto\_increment seja gerado automaticamente pelo MySQL. Este é um comportamento específico deste SGBD. A Figura 17 ilustra esta situação, onde os comandos são exibidos na parte inferior da tela.

Figura 17. Comandos INSERTs gerados automaticamente pelo MySQL Workbench.

Para criar uma view, basta acionar o botão Add view (Adicionar view – ver Figura 17) e inserir o comando SELECT para a geração da mesma. A Figura 18 fornece a tela com a criação de uma view para listar o nome dos atletas e suas modalidades, respectivamente. Vale destacar que apenas o comando SELECT teve que ser digitado, já que a ferramenta já disponibilizou o template com o comando CREATE VIEW.

Figura 18. View para listar o nome dos atletas e suas modalidades.

É importante perceber que é possível visualizar todas as views existentes através da seção dedicada a estes objetos. Ela está situada na parte central da tela, como mostra a Figura 18. Além disto, pode-se ver no catálogo exibido à direita da tela todos os objetos definidos até o momento.





Da mesma forma que as views, uma rotina pode ser criada a partir do botão Add Routine (Incluir Rotina). Vale destacar que uma rotina pode ser tanto uma stored procedure, uma função ou trigger. Além disto, o editor aparecerá com todos os comandos básicos para a definição desta rotina, e o usuário terá apenas que informar o código que define a sua lógica. Na Figura 19 está sendo criada uma stored procedure que exibe o nome de todos os atletas cadastrados. Lembrando que apenas o comando SELECT foi digitado, já que o os demais comandos foram gerados pela própria ferramenta.

Figura 19. Stored Procedure para exibir o nome de todos os atletas cadastrados.

Da mesma forma, todas as rotinas existentes ficam disponíveis na área central dedicada a estes objetos, e no catálogo de dados é possível visualizá-las da mesma forma.

Para ilustrar a geração dos comandos DDL para estes objetos, se for utilizada a opção de exportação ilustrada na seção anterior, obteremos o código exibido na Listagem 2. Vale destacar que na Listagem 2 estão apenas os novos objetos construídos após a criação das tabelas e relacionamentos.

Listagem 2. Listagem dos comandos SQL DDL para a insercao de dados, criação de view e stored procedure.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';


USE `mydb`;


-- -----
-- Data for table `mydb`.`Modalidades`
-- -----
```

```
SET AUTOCOMMIT=0;

INSERT INTO `Modalidades` (`idModalidades`, `nome`, `sexo`) VALUES (0, 'Futebol', 'Masculino');
INSERT INTO `Modalidades` (`idModalidades`, `nome`, `sexo`) VALUES (0, 'Volei', 'Feminino');

COMMIT;
```

```
-- -----
-- View `mydb`.`atletaModalidade`
-- -----

DROP TABLE IF EXISTS `mydb`.`atletaModalidade`;
CREATE OR REPLACE VIEW `mydb`.`atletaModalidade` AS
SELECT a.nome, m.nome FROM Atletas INNER JOIN Modalidades USING (idModalidade);

DELIMITER //
//
CREATE PROCEDURE `mydb`.`listaNomeAtletas` ()
BEGIN
SELECT nome FROM mydb.Atletas;
END//

DELIMITER ;
```

Pode-se observar que na listagem estão os INSERTs das modalidades esportivas, a view e a procedure definidas anteriormente. Este script pode ser utilizado juntamente com aquela exibido na Listagem 1 para a geração desta base de dados completa no MySQL.

Finalmente, para os usuários acostumados a utilizar a ferramenta de modelagem DBDesigner4, que foi descontinuada, o MW possui suporte para a importação dos modelos gerados naquela ferramenta. Assim, torna-se fácil o aproveitamento de modelos já existentes, evitando o

Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



Fábrica de Software – Instituto de Informática – Universidade Federal de Goiás  
Sistema de Gestão Bibliográfica  
SGB\_MANUAL\_Workbench Versão 1.0



retrabalho. A Figura 20 ilustra esta opção para importação de modelos do DBDesigner4.

Figura 20. Opção para importação de dados do DBDesigner4.

## 8 Referência

- Artigo SQL Magazine 58 - Modelagem de dados com o MySQL Workbench <http://www.devmedia.com.br/artigo-sql-magazine-58-modelagem-de-dados-com-o-mysql-workbench/10936#ixzz2B0uYB4sH>