Banco de Dados

Aula 3 - MySQL Workbench, SELECT e WHERE

MySQL Workbench

O MySQL Workbench é uma ferramenta gráfica de usuário que permite projetar, criar e explorar seus esquemas de bancos de dados. Além disso, ela permite editar/executar queries SQL, trabalhar com objetos de bancos de dados e fazer migração de esquemas de outros bancos.

O MySQL Workbench é mantido pela Oracle e está disponível no link abaixo para baixar de forma gratuita.

https://www.mysql.com/products/workbench/

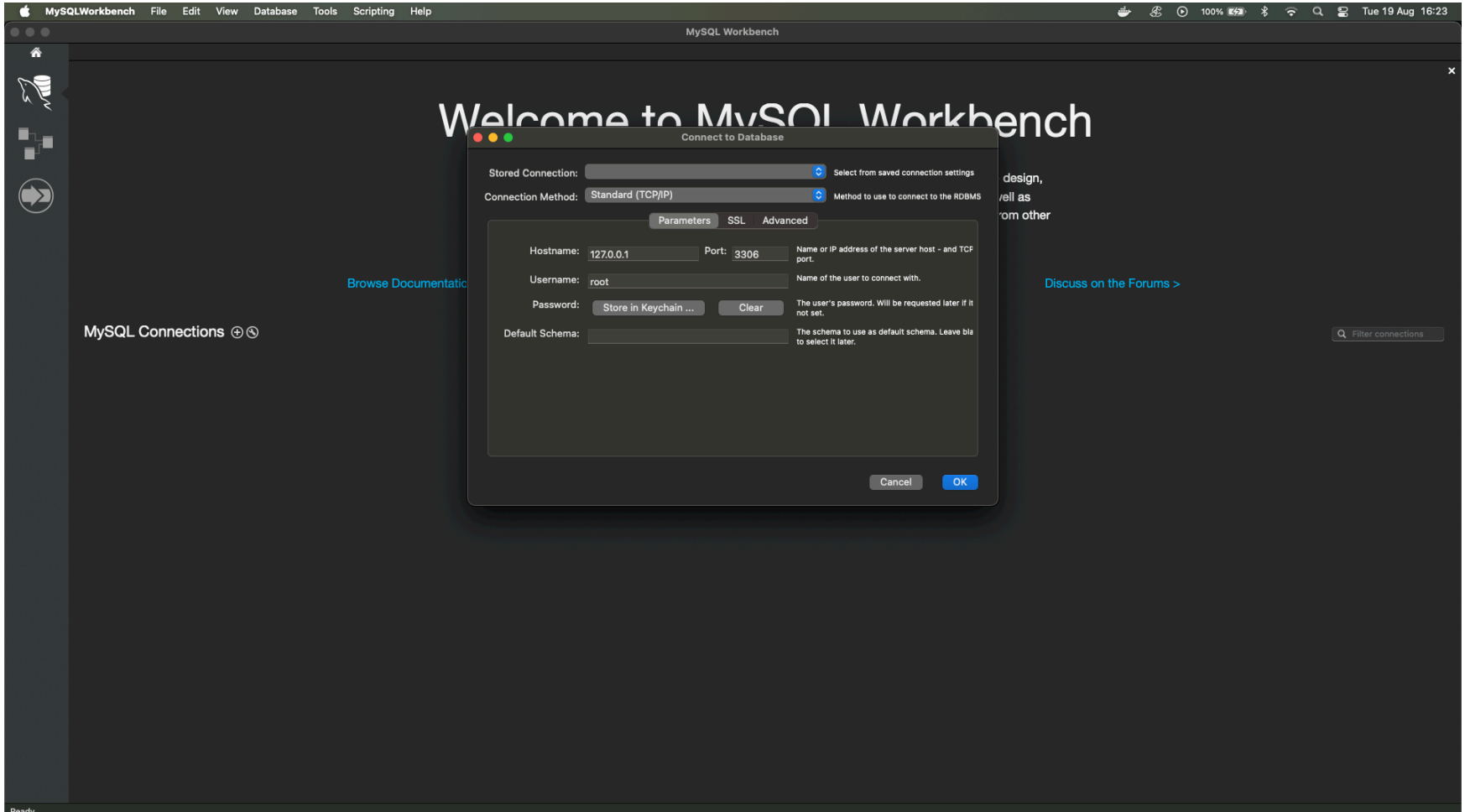
Conexão com o Banco de Dados

Antes de qualquer coisa precisamos iniciar uma conexão com o banco de dados. No repositório da disciplina no Github encontramos um arquivo docker-compose.yml configurado para rodar o MySQL 8. Então, após clonar o repositório na nossa máquina, acesse o diretório do projeto e inicie o banco de dados utilizando o Docker.

```
docker compose up -d
```

Em seguida inicie a conexão do Workbench com o banco acessando o menu `Database > Connect to Database`. Preencha o campo "Password" com o valor configurado na arquivo `docker-compose.yml` na variável de ambiente `MYSQL_ROOT_PASSWORD`. No campo "Port" devemos manter 3306 (padrão pro banco MySQL). Os campos o User e Host deve ser `root` e 127.0.0.1, respectivamente, pois estamos conectando com um banco local.

É importante lembrar que esses valores foram configurados desta maneira por questão de aprendizado, mas informações como senhas não devem ser compartilhadas e nem configuradas no docker-compose.yml da maneira como foi apresentado. Pois dessa maneira os dados ficam expostos no Github, e quem tiver acesso ao seu repositório terá acesso de administrador ao seu banco de dados.


MySQLWorkbench File Edit View Database Tools Scripting Help
Tue 19 Aug 16:23
MySQL Workbench
Welcome to MySQL Workbench
design,
ell as
om other
Browse Documentatic
Discuss on the Forums >
MySQL Connections
Filter connections
Connect to Database
Stored Connection:
Select from saved connection settings
Connection Method:
Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters
SSL
Advanced
Hostname:
127.0.0.1
Port:
3306
Name or IP address of the server host - and TCP port.
Username:
root
Name of the user to connect with.
Password:
Store in Keychain ...
Clear
The user's password. Will be requested later if it not set.
Default Schema:
The schema to use as default schema. Leave bla to select it later.
Cancel
OK
Ready.

Populando o banco

Após criar a conexão com o banco, já podemos executar nossas primeiras queries. Inicialmente, nós devemos popular o banco com o que chamamos de semente (seeds). São as primeiras informações necesssárias para um sistema funcionar, como, por exmeplo, contas administrativas.

Dentro do repositório da disciplina nós encontramos o arquivo `lectures/lecture-3/criar-tabela.sql`. Nós podemos abrir esse arquivo com o Workbench, assim ele já carrega todo o código na seção de edição de query.

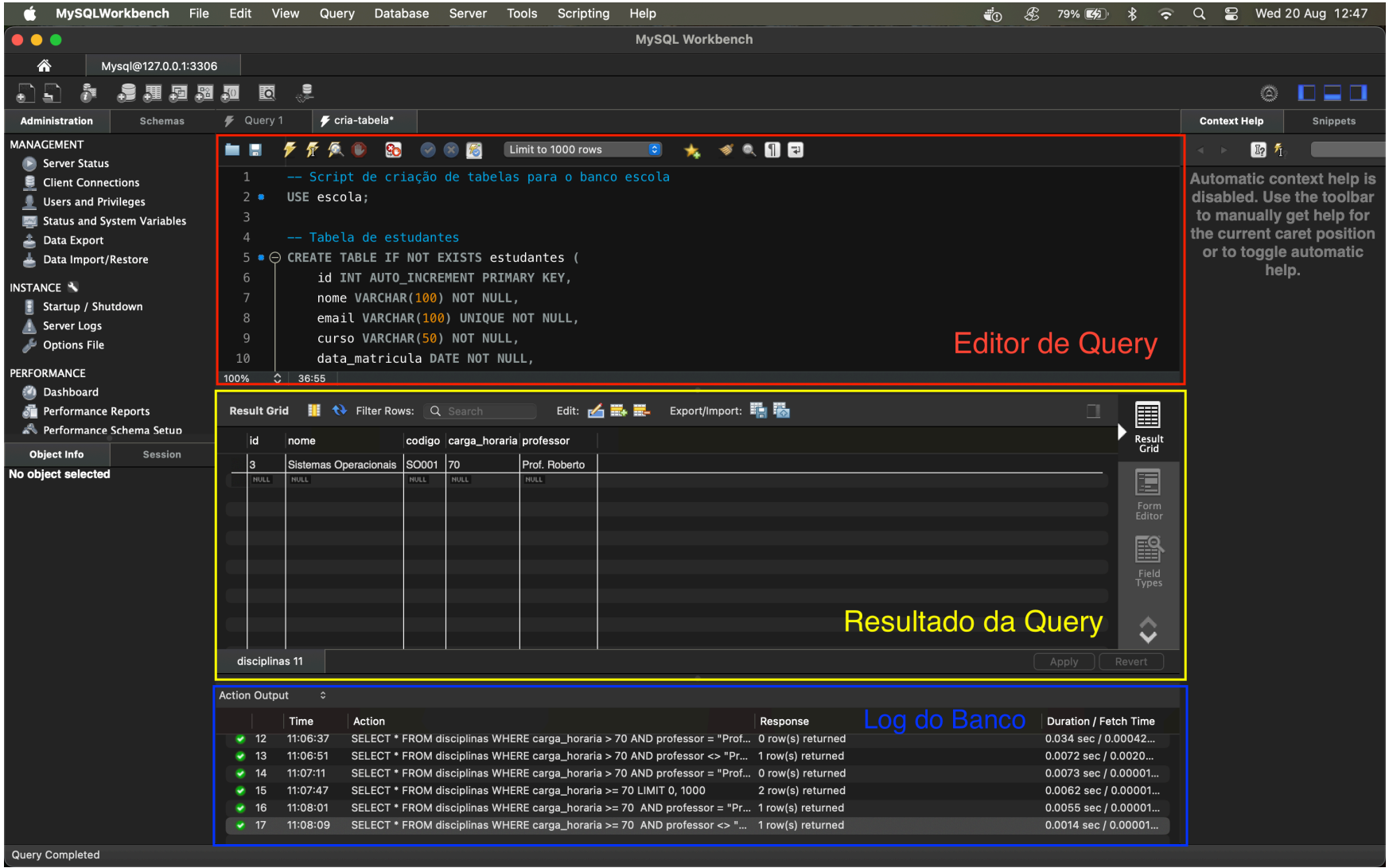
Após carregar as queries, basta clicar no ícone com o raio amarelo, mais a esquerda, para executar todas as queries de uma vez. Este botão serve para executar as queries selecionadas com o cursor. Porém, se nenhuma linha estiver selecionada o Workbench executa todas as queries do editor, linha por linha.

O ícone do meio é utilizado para executar apenas a query onde se encontra o cursor do teclado. O ícone mais a direita é utilizado pra ajudar a entender quais modificações a query está executando. Esta última opção é utilizada para queries muito complexas e veremos mais pra frente sua utilidade.

Populando o banco

Após executar todas as queries podemos obrsevar alterações nas seções logo abaixo do editor de query. Na seção do meio se encontra o resultado da execução das queries. Como neste exemplo ainda não fizemos nenhuma ação importante, vemos apenas "Dados inseridos com sucesso!", pois é o que o resultado esperado da última query de SELECT, na linha 52 do nosso arquivo SQL.

Na seção na parte inferior da tela, nos vemos um log de ações executadas na comunicação com o banco de dados, como data e hora, informações da query, a resposta do banco e quanto tempo levou pro banco de dados processar a query. Essas informações são importantes quando estamos tentando debugar um conjunto de queries muito complexas ou que levam muito tempo para executar.


MySQLWorkbench File Edit View Query Database Server Tools Scripting Help
Wed 20 Aug 12:47
MySQL Workbench
Mysql@127.0.0.1:3306
Administration
Schemas
Query 1
cria-tabela*
Context Help
Snippets
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Object Info
Session
No object selected
Limit to 1000 rows
1 -- Script de criação de tabelas para o banco escola
2 USE escola;
3
4 -- Tabela de estudantes
5 CREATE TABLE IF NOT EXISTS estudantes (
6 id INT AUTO_INCREMENT PRIMARY KEY,
7 nome VARCHAR(100) NOT NULL,
8 email VARCHAR(100) UNIQUE NOT NULL,
9 curso VARCHAR(50) NOT NULL,
10 data_matricula DATE NOT NULL,
Editor de Query
100%
36:55
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
id nome codigo carga_horaria professor
3 Sistemas Operacionais SO001 70 Prof. Roberto
Result Grid
Form Editor
Field Types
Resultado da Query
disciplinas 11
Apply
Revert
Action Output
Time Action Response Duration / Fetch Time
12 11:06:37 SELECT * FROM disciplinas WHERE carga_horaria > 70 AND professor = "Prof... 0 row(s) returned 0.034 sec / 0.00042...
13 11:06:51 SELECT * FROM disciplinas WHERE carga_horaria > 70 AND professor <> "Pr... 1 row(s) returned 0.0072 sec / 0.0020...
14 11:07:11 SELECT * FROM disciplinas WHERE carga_horaria > 70 AND professor = "Prof... 0 row(s) returned 0.0073 sec / 0.00001...
15 11:07:47 SELECT * FROM disciplinas WHERE carga_horaria >= 70 LIMIT 0, 1000 2 row(s) returned 0.0062 sec / 0.00001...
16 11:08:01 SELECT * FROM disciplinas WHERE carga_horaria >= 70 AND professor = "Pr... 1 row(s) returned 0.0055 sec / 0.00001...
17 11:08:09 SELECT * FROM disciplinas WHERE carga_horaria >= 70 AND professor <> "... 1 row(s) returned 0.0014 sec / 0.00001...
Log do Banco
Query Completed

Executando o primeiro SELECT

Agora já podmeos iniciar as primeiras ações de busca de informações no banco. O primeiro comando importante é o SELECT, o qual indica ao banco de dados em quais tabelas buscar informações e quais critérios a busca deve atender. Vejamos um primeiro exmeplo:

```
SELECT * FROM estudantes;
```

Neste exemplo estamos pedindo ao banco que "selecione todas as colunas da tabela estudantes". O asterísco indica que é pra buscar todas as colunas disponíveis. Se quisermos ver apenas os nomes dos estudantes podemos fazer o seguinte:

```
SELECT nome FROM estudantes;
```

Assim, o banco de dados entende que deve retornar apenas a coluna com os nomes dos estudantes.

Tornando o SELECT mais interessante

Vamos olhar agora para a tabela notas e tentar arredondar os valores das notas de cada estudante.

```
SELECT * FROM notas;
SELECT id, estudante_id, disciplina_id, nota as "nota inicial", round(nota) as "nota arredondada" FROM notas;
```

Ainda podemos executar operações matemáticas com os dados númericos. Imagina que precisamos multiplicar por um valor como 1,52.

```
SELECT id, estudante_id, disciplina_id, nota as "nota inicial", nota * 1.52 as "nota arredondada" FROM notas;
```

Operações com números

Algumas das operações matemáticas permitidas com SQL são:

Operador	Descrição	Exemplo
+	Soma os valores de duas colunas	prduto_A + produto_B
-	Subtrai os valores de duas colunas	prduto_C - desconto
*	Multiplica os valores de duas colunas	prduto_D * quantidade
/	Divide os valores de duas colunas	preço_total / preço_unitario
%	Divide os valores de duas colunas e retorna o resto da divisão	preço_total / preço_unitario

WHERE

Em bancos de dados com muitas informações, buscar todos os dados da tabela pode não ser muito útil. Nestes casos é importante saber criar filtros para que o banco de dados retorne informações selecionadas dentro das tabelas.

Imagina que no nosso banco de dados queremos selecionar apenas estudantes inscritos no curso de Informática. Então podemos executar a seguinte query:

```
SELECT * FROM estudantes WHERE curso = "informática";
```

Agora, se queremos buscar apenas as notas maiores que 8, podemos usar a seguinte query:

```
SELECT * FROM notas WHERE nota > 8;
```

Ainda podemos fazer combinações de filtros utilizando operadores `AND` e `OR`. Por exemplo, considere a query a seguir, onde filtramos por diciplinas com carga horária maior ou igual a 70 horas e com professor diferente do Professor Carlos

```
SELECT * FROM disciplinas WHERE carga_horaria >= 70 AND professor <> "Prof. Carlos";
```

Operadores com WHERE

Alguns dos operadores comumente utilizados com WHERE na linguagem SQL são:

Operadores	Descrição	Exemplo
>, >=, <, <=, =, != ou <>	Comparação de valores	nota >= 7
AND, OR	Agrupamento de condições	prduto_C - desconto
IN	Usado para verificar se o valor se encontra dentro do vetor	nota IN [9.0, 8.0, 10.0]
IS	Usado para verificar se o valor é do mesmo tipo de NULL	IS NULL
NOT	Operador de negação para inverter o resultado de uma comparação	IS NOT NULL