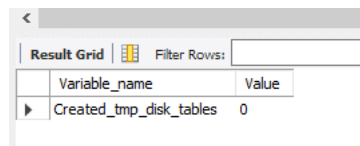


Consolidando o seu conhecimento

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

- 1) As variáveis que estão declaradas no **C:\ProgramData\MySQL\MySQL Server 8.0\my.ini** serão inicializadas com os valores declarados no arquivo sempre que o MySQL for inicializado.
- 2) [Aqui \(https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html\)](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html), você pode ver a documentação de inúmeras variáveis de ambiente.
- 3) O valor das variáveis durante a sessão pode ser vista pelo Workbench. Entre no Workbench e, na base de dados **sakila**, digite no editor de comandos SQL:

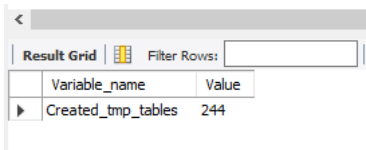
```
SHOW GLOBAL STATUS LIKE 'Created_tmp_disk_tables';
```



Variable_name	Value
Created_tmp_disk_tables	0

- 4) Ainda na base de dados **sakila**, outra variável pode ser observada:

```
SHOW GLOBAL STATUS LIKE 'Created_tmp_tables';
```



Variable_name	Value
Created_tmp_tables	244

Estas duas variáveis estão relacionadas com o número de tabelas temporárias que podem ser abertas durante uma sessão em memória e em disco. Claro que isso influencia na performance do banco, caso seja necessário usar o HD para armazenar tabelas temporárias criadas pelo MySQL durante os comandos SQL.

- 5) A variável `tmp_table_size`, que foi inicializada pelo **my.ini**, tem o valor de 103 e ele pode ser visto pelo comando do WorkBench:

```
SHOW GLOBAL VARIABLES LIKE 'tmp_table_size';
```

- 6) A variável de ambiente pode ser modificada pelo usuário que tenha privilégios para isso. Para isso, novamente na base de dados **sakila**, digite o seguinte comando:

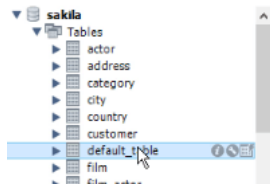
```
SET GLOBAL tmp_table_size = 208003328;
```

- 7) Assim, é possível modificar o valor desta variável e ignorar o que estava, inicialmente, especificado no **my.ini**.

8) Já sobre mecanismos de armazenamentos, durante a criação da tabela, é possível determinar qual mecanismo a mesma irá utilizar. Crie uma tabela, na base de dados **sakila**, conforme o comando abaixo:

```
CREATE TABLE DEFAULT_TABLE (ID INTEGER, NOME VARCHAR(100));
```

9) Se você for na tabela, na árvore de objetos do Workbench e clicar sobre o ícone de informações, verá as características de armazenamento desta tabela que foi criada:



10) Você pode ver que, por padrão, as tabelas são criadas com o mecanismo de armazenamento **InnoDB**:

Local instance MySQL80 sakila.default_table	
Table Details	
Engine:	InnoDB
Row format:	Dynamic
Column count:	2
Table rows:	0
AVG row length:	0
Data length:	16.0 KiB
Index length:	0.0 bytes
Max data length:	0.0 bytes

11) É possível alterar a propriedade do mecanismo de armazenamento da tabela, com o comando:

```
ALTER TABLE DEFAULT_TABLE ENGINE = MyISAM;
```

12) Além disso, você pode definir o tipo de mecanismo de armazenamento que será usado na tabela no momento de sua criação. Para isso, digite:

```
CREATE TABLE DEFAULT_TABLE2 (ID INTEGER, NOME VARCHAR(100)) ENGINE = MEMORY;
```

13) Quando você cria uma tabela pelo assistente do Workbench, você pode ver a opção de seleção dos mecanismos de armazenamento, sempre apresentando o **InnoDB** como padrão:

Table Name:

Schemas: **sakila**

Charset/Collation:

Default Charset

Default Collation

Engine:

InnoDB

Comments:

int:

text:

Default:

Storage:

☐ Virtual

☐ Primary

☐ Binary

☐ Auto

Indexes

Foreign Keys

Triggers

Partitioning

Options

MyISAM

ndbcluster

MEMORY

EXAMPLE

FEDERATED

ARCHIVE

CSV

BLACKHOLE

innoDB

IBMDB2

BrightHouse

KFDB

ScaleDB

TokuDB

XtraDB

Spider

Mroonga

MRG_MyISAM

Aria

Percona

Save

Cancel

OK

Apply

Revert