



WWW.UNICARIOCA.EDU.BR

Banco de Dados II

Tema 10

Profª Giselle Batalha



MELHOR CENTRO UNIVERSITÁRIO DO RIO DE JANEIRO!

Stored procedures (*procedimentos armazenados*) são rotinas definidas no banco de dados, identificadas por um nome pelo qual podem ser invocadas.

Um procedimento pode executar uma série de instruções, receber parâmetros e retornar valores.

Principais vantagens e desvantagens do uso de Stored Procedure:



Pontos positivos:

- Simplificação da execução de instruções **SQL** pela aplicação;
- Transferência de parte da responsabilidade de processamento para o servidor.
- Facilidade na manutenção, reduzindo a quantidade de alterações na aplicação.



Pontos negativos:

- Necessidade de maior conhecimento da sintaxe do banco de dados para escrita de rotinas em SQL;
- As rotinas ficam mais facilmente acessíveis. Alguém que tenha acesso ao banco poderá visualizar e alterar o código.

Sintaxe

```
CREATE PROCEDURE proc_name([parameters, ...])  
[BEGIN]
```

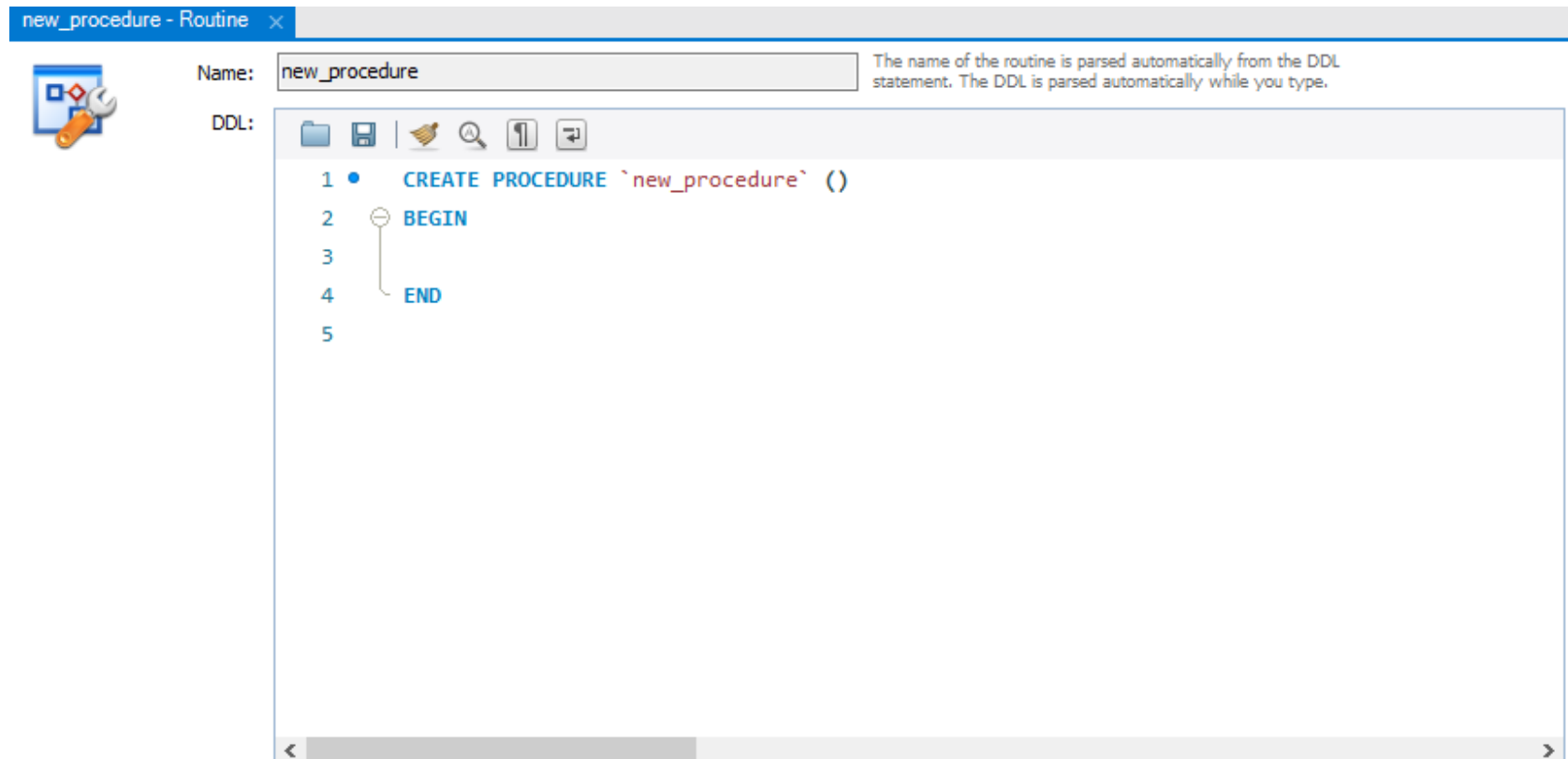
```
    corpo_da_rotina;
```

```
[END]
```

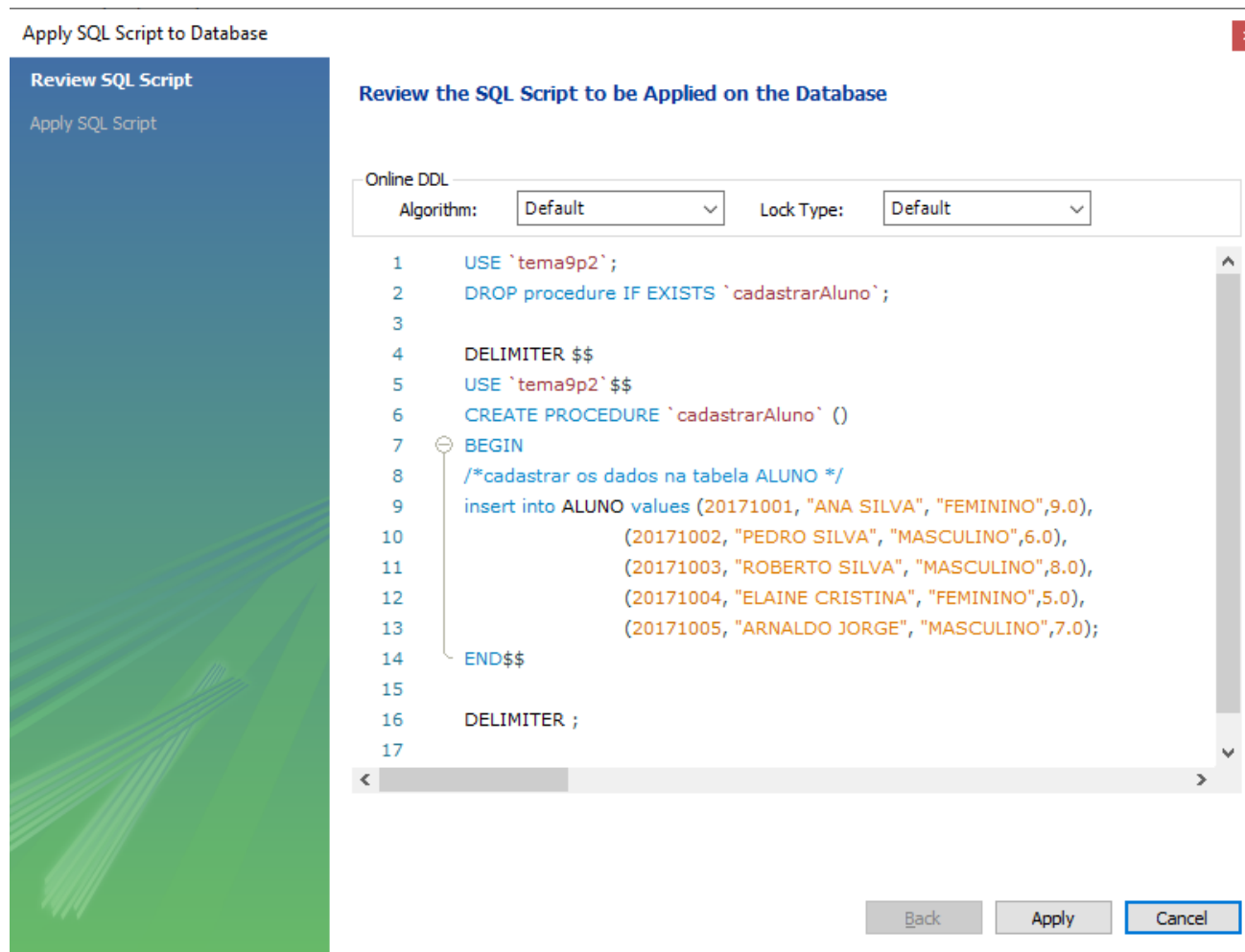
Tipos de Parâmetros

- IN => de entrada, ou seja, um parâmetro cujo seu valor será utilizado no interior do procedimento para produzir algum resultado;
- OUT => retorna algo de dentro do procedimento para o lado externo, colocando os valores manipulados disponíveis na memória ou no conjunto de resultados;
- INOUT => entrada e saída

Criando Stored Procedure



Exemplo 1



Para executar o procedimento: `call cadastrarAluno();`

Exemplo 2

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Default Lock Type: Default

```
1  USE `tema9p2` ;
2  DROP procedure IF EXISTS `cadastrarCurso` ;
3
4  DELIMITER $$
5  USE `tema9p2` $$
6  CREATE PROCEDURE `cadastrarCurso` ()
7  BEGIN
8      /*cadastrar os dados na tabela CURSO */
9      insert into CURSO values (1, "BD MODELAGEM", "MANHÃ",1000),
10                               (2, "BD SQL", "TARDE",2000),
11                               (3, "ALGORITMOS", "TARDE",3000),
12                               (4, "COBOL", "NOITE",4000),
13                               (5, "PROGRAMAÇÃO", "NOITE",3000);
14
15  END$$
16
17  DELIMITER ;
```

Back Apply Cancel

Para executar o procedimento: `call cadastrarCurso();`

Exemplo 3

Apply SQL Script to Database ✕

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Lock Type:

```
1  USE `tema9p2` ;
2  DROP procedure IF EXISTS `Verificar_Quantidade_Cursos` ;
3
4  DELIMITER $$
5  USE `tema9p2` $$
6  CREATE PROCEDURE `Verificar_Quantidade_Cursos` (OUT quantidade INT)
7  BEGIN
8    SELECT COUNT(*) INTO quantidade FROM CURSO;
9  END$$
10
11 DELIMITER ;
12
13
```

< >

Para executar o procedimento:

```
call Verificar_Quantidade_Cursos (@total) ;
SELECT @total;
```

Acessar o código de uma Stored Procedure

SHOW CREATE PROCEDURE NOMEPROCEDURE;

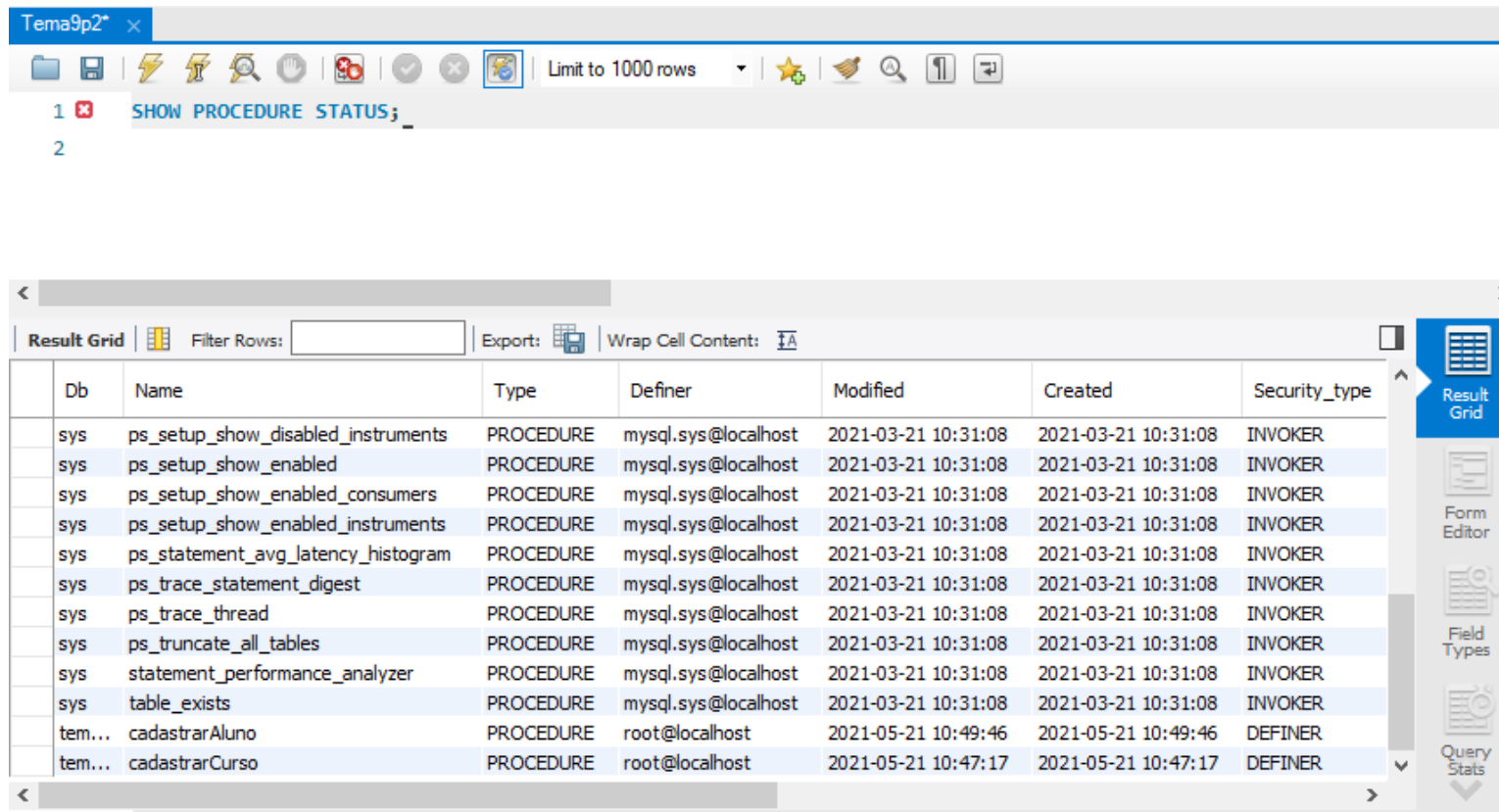
The screenshot shows a database client interface. The top window displays the SQL command `show create procedure cadastrarAluno;`. The bottom window shows the result of this command as a table with 5 columns: Procedure, sql_mode, Create Procedure, character_set_client, and collation_cor. The first row of data shows the procedure `cadastrarAluno` with its full create statement, which includes a `CREATE DEFINER='root'@'localhost' PROCEDURE` clause. A third window, titled "Edit Data for Create Procedure (VARCHAR)", shows the full SQL code for the procedure, including the `CREATE DEFINER` clause and the `BEGIN` block with `INSERT` statements.

Procedure	sql_mode	Create Procedure	character_set_client	collation_cor
cadastrarAluno	STRICT_TRANS_TABLES,NO_ENGINE_SUBSTIT...	CREATE DEFINER='root'@'localhost' PROCEDURE `cadastrarAluno`()	utf8mb4	utf8mb4_090

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `cadastrarAluno`()
2 BEGIN
3 /*cadastrar os dados na tabela ALUNO */
4 insert into ALUNO values (20171001, "ANA SILVA", "FEMININO",9.0),
5                           (20171002, "PEDRO SILVA", "MASCULINO",6.0),
6                           (20171003, "ROBERTO SILVA", "MASCULINO",8.0),
7                           (20171004, "ELAINE CRISTINA", "FEMININO",5.0),
8                           (20171005, "ARNALDO JORGE", "MASCULINO",7.0);
9 END
```

Exibindo procedures

SHOW PROCEDURE STATUS;

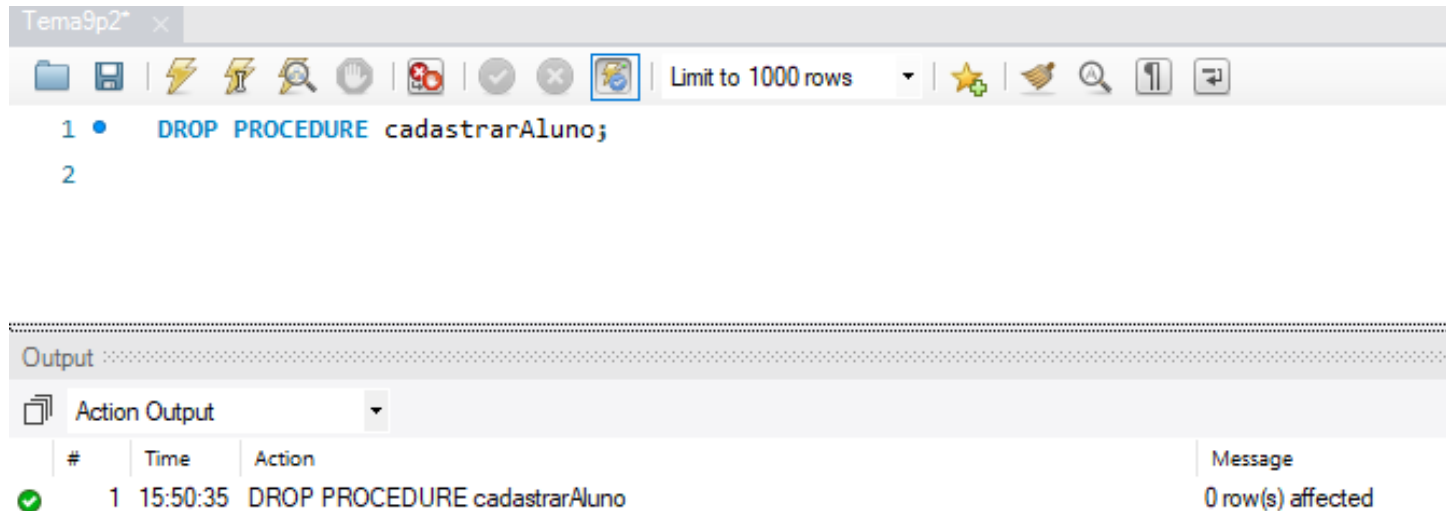


The screenshot shows a MySQL database client interface. The top toolbar includes icons for file operations, execution, and search. The command line shows the execution of 'SHOW PROCEDURE STATUS;'. Below the command line, a table displays the results of the query. The table has columns for Db, Name, Type, Definer, Modified, Created, and Security_type. The results list various system procedures and two user-defined procedures in the 'tem...' database.

Db	Name	Type	Definer	Modified	Created	Security_type
sys	ps_setup_show_disabled_instruments	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_setup_show_enabled	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_setup_show_enabled_consumers	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_setup_show_enabled_instruments	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_statement_avg_latency_histogram	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_trace_statement_digest	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_trace_thread	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	ps_truncate_all_tables	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	statement_performance_analyzer	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
sys	table_exists	PROCEDURE	mysql.sys@localhost	2021-03-21 10:31:08	2021-03-21 10:31:08	INVOKER
tem...	cadastrarAluno	PROCEDURE	root@localhost	2021-05-21 10:49:46	2021-05-21 10:49:46	DEFINER
tem...	cadastrarCurso	PROCEDURE	root@localhost	2021-05-21 10:47:17	2021-05-21 10:47:17	DEFINER

Excluindo procedures

```
DROP PROCEDURE NOMEPROCEDURE;
```



The screenshot shows a database management interface with a toolbar and a query editor. The toolbar includes icons for file operations, execution, search, and other database functions. The query editor contains the following SQL statement:

```
1 DROP PROCEDURE cadastrarAluno;  
2
```

Below the query editor is an "Output" section with a dropdown menu set to "Action Output". The output table shows the execution results:

#	Time	Action	Message
1	15:50:35	DROP PROCEDURE cadastrarAluno	0 row(s) affected