

cronapi.database.Operations.executeProcedure

Executar procedure retornando dados

Executa os comandos de uma *procedure* previamente criada no banco de dados e retorna uma lista de dados da mesma.

Parâmetros de Entrada

Nome	Tipo	Doc
Parâmetro 1	String	Informa o nome do pacote (<i>namespace</i>) que contém a entidade
Parâmetro 2	String	Nome da procedure no banco de dados
Parâmetro 3	Objeto	(Opcional) Campo para informar os dados dos parâmetros da procedure
Parâmetro 4	String	(Opcional) Informe o nome da classe Java para retornar a lista de objetos com os campos definidos ou deixe em branco para retornar listas com os valores dos registros.

Sobre procedure

Procedure é um conjunto de comandos em **SQL** que podem ser executados de uma só vez, como em uma função. Ele armazena tarefas repetitivas e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

Nesta página

- [Executar procedure retornando dados](#)
 - [Parâmetros de Entrada](#)
 - [Retorno](#)
 - [Compatibilidade](#)
 - [Diagrama de dados dos exemplos](#)
 - [Exemplo 1](#)
 - [Exemplo 2](#)

Retorno

Retorna uma lista de dados.

Compatibilidade



Servidor

Diagrama de dados dos exemplos

Neste tutorial estamos utilizando o diagrama abaixo (Figura 1).

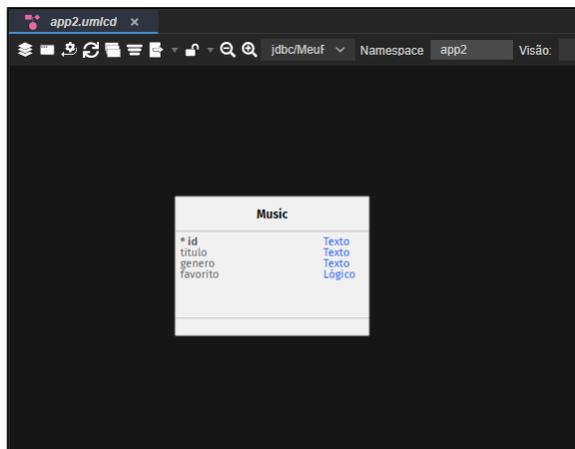


Figura 1 - Diagrama de dados

Exemplo 1

Ao criar uma procedure, como mostrado na Figura 1.1, a função **Executar procedure retornando dados** (Figura 1.2) faz a sua chamada, passando o *namespace* que se encontra a entidade, o nome da *procedure* e o valor do parâmetro através de uma lista. Ao final, ele irá retornar um *array* com os dados do comando executado (nesse caso, os dados referentes aos campos *titulo* e *genero* da entidade *Music*).

```
Query 1
1
2 • use IBase64;
3
4 DELIMITER $$
5 • create procedure minhas_musicas (in favorita boolean)
6 BEGIN
7   select titulo, genero from MUSIC
8   where favorito = favorita;
9 END $$
10 DELIMITER ;
11
12
```

Figura 1.1 - Procedure criada

Na figura abaixo o bloco retorna uma lista com várias sub-listas que contém os valores dos campos requisitado na *procedure*, por padrão é utilizado a classe gerada pelo diagrama gerada dentro de *entity*.

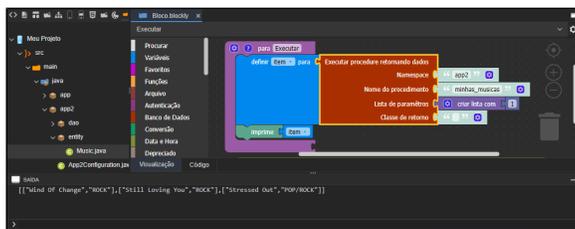


Figura 1.2 - Executando a procedure

Exemplo 2

Nesse exemplo, ao informar a classe de retorno da *procedure*, ele irá retornar uma lista de objetos, com os respectivos campos (definidos na classe) e seus valores (Figura 2.1). A classe pode ser gerada pelo diagrama ou criada manualmente, mas ela deve conter os campos da procedure na mesma ordem - ou seja, a ordem dos campos definidos na classe Java devem ser iguais aos campos retornados pela procedure. No exemplo da figura 2.1, foi retornado uma lista de objetos com os atributos *titulo* e *genero* contidos, e na mesma ordem, na classe (*Music2.java* - Figura 2) e também na procedure (*minhas_musicas* - Figura 1.1).

No nosso exemplo, criamos a classe *Music2.java* dentro do namespace *app2*.

```
Music2.java
package app2;

/**
 * Classe que representa ...
 * @author Wesley Miranda de Oliveira
 * @version 1.0
 * @since 2022-08-24
 */
public class Music2 {
    // ...
    // Construtor
    // ...
    // Método
    public Music2 () {
        // ...
        public String titulo;
        public String genero;
    }
}
```

Figura 2 - Classe Java utilizada no segundo exemplo

Classe Java Music

```
package app2;

/**
 * Classe que representa ...
 *
 * @author Wesley Miranda De Oliveira
 * @version 1.0
 * @since 2022-08-24
 */

public class Music2 {

    /**
     * Construtor
     */
    public Music2 () {

        public String titulo;
        public String genero;

    }
}
```

Quando informado uma classe no último parâmetro, o retorno da procedure será convertido em uma lista de objetos (Figura 2.1). O resultado final obtido será um **array de objetos** com os títulos e gêneros das músicas que estão cadastradas como favoritas.

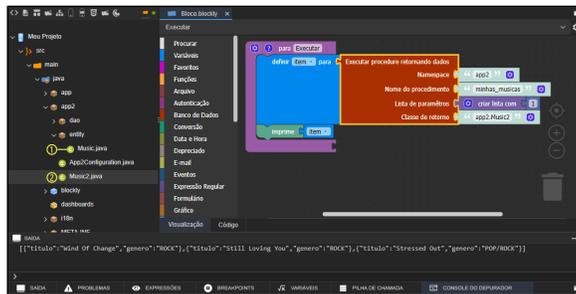


Figura 2.1 - Executando a procedure informando a classe de retorno