

MÓDULO

# BASE DE DADOS MYSQL

## PRÁTICA 35

RECUPERAR, MODIFICAR E APAGAR DADOS  
DE UMA BASE DE DADOS MYSQL – PARTE II

## RECUPERAR, MODIFICAR E APAGAR DADOS DE UMA BASE DE DADOS MYSQL – PARTE II

---

### 1.1. DESCRIÇÃO DA PRÁTICA

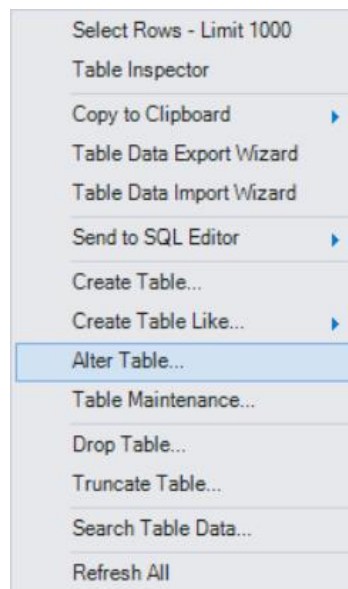
**Tempo estimado:** 1 h.

REQUISITOS
■ Ter concluído a unidade didática “Apagar dados em MySQL”.
OBJETIVOS
■ Consolidar os conhecimentos relativos à modificação de dados das bases de dados.
■ Aprender a fazer essas mesmas modificações no MySQL Workbench.
MATERIAL E FERRAMENTAS NECESSÁRIAS PARA A PRÁTICA
■ Servidor web.
■ MySQL Workbench.

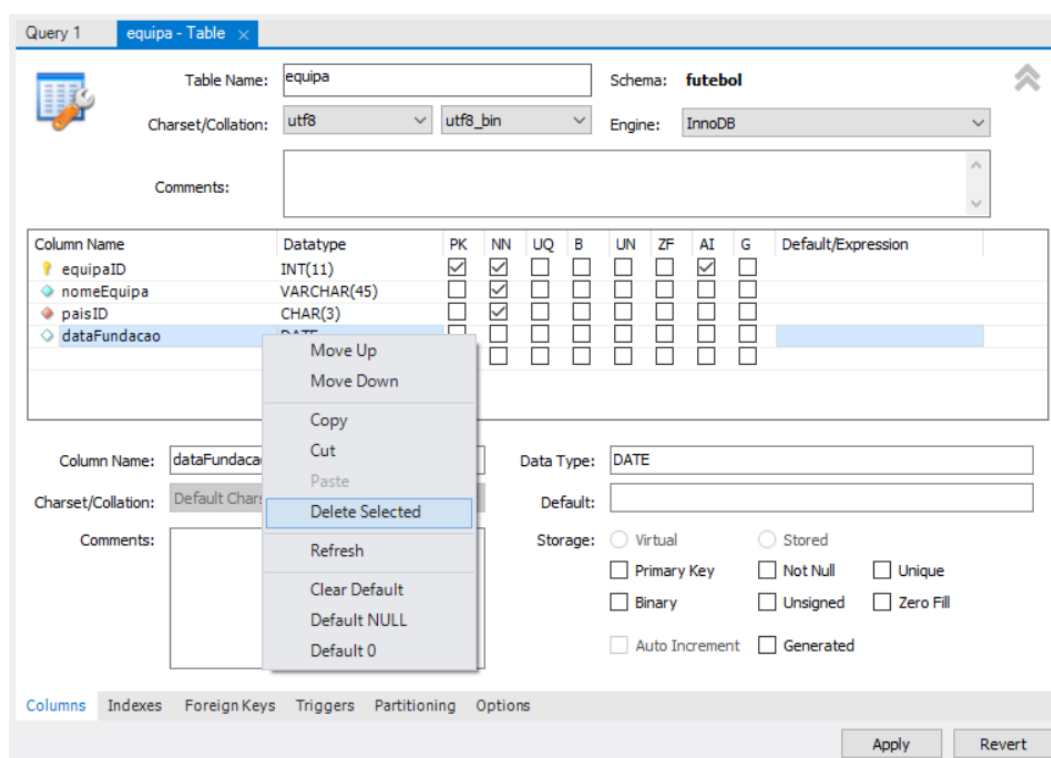
## 1.2. DESENVOLVIMENTO DA PRÁTICA

Nesta prática, à semelhança das anteriores, irá ser abordada a forma mais gráfica de fazer alterações aos dados através do MySQL Workbench.

Primeiramente, abra o programa, indique que a base de dados a ser utilizada é a “futebol” e clique, com o botão do lado direito do rato, na tabela “equipa”. Depois, escolha a opção **Alter Table** do menu.



Uma janela semelhante à utilizada durante a criação da tabela aparecerá em primeiro plano. Nesta janela estão incluídos todos os parâmetros importantes da tabela, como o motor utilizado, a codificação de caracteres, as colunas, os índices, as chaves, entre outros. Selecione a aba **Columns** e elimine a coluna “dataFundacao”, clicando com o botão do lado direito do rato e selecionando a opção **Delete Selected**. De seguida, clique no botão **Apply** para aplicar as alterações feitas. Após esta etapa, aparece uma janela de confirmação, na qual a consulta SQL a ser executada é anexada para que as alterações tenham efeito. Neste caso, a consulta será: **Alter Table** 'futebol'. 'equipa' **Drop Column** 'dataFundacao' (praticamente igual ao que foi executado na linha de comandos na unidade didática “Modificar Dados em MySQL”). Clique novamente no botão **Apply** e, se tudo estiver correto, aparecerá uma janela a confirmar isso mesmo.



Através deste método, pode ser realizado qualquer tipo de modificação na estrutura das tabelas da base de dados. Como não se pretendia apagar a coluna anterior, irá adicionar esta coluna novamente, mas, desta vez, de um modo mais gráfico. Também será adicionada uma coluna chamada “ultimoPresidente” do tipo VARCHAR(100). Ao aplicar estas mudanças, a seguinte consulta será executada, o que mostra que é possível fazer várias modificações na estrutura da tabela com uma única indicação **Alter Table**.

```
ALTER TABLE 'futebol'.'equipa'

ADD COLUMN 'dataFundacao' DATE NULL

AFTER 'paisID',

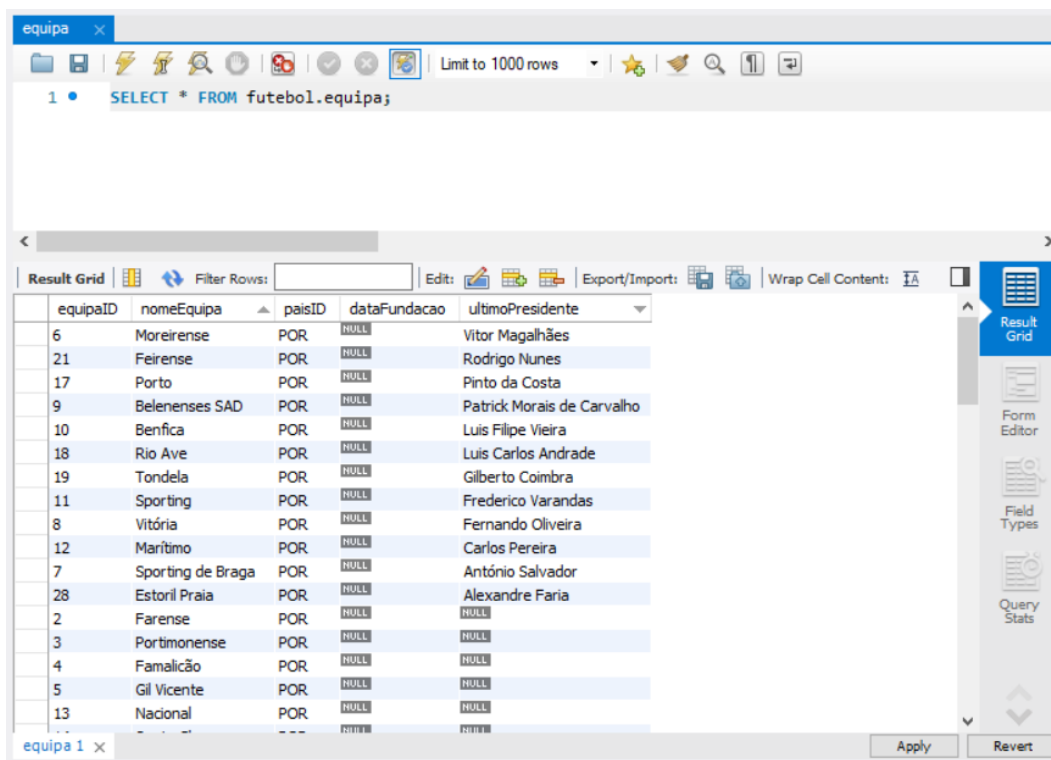
ADD COLUMN 'ultimoPresidente' VARCHAR(100) NULL

AFTER 'dataFundacao';
```

Ao explorar e ao fazer testes e alterações nos diferentes separadores da janela, poderá introduzir todas as alterações que considere adequadas nas tabelas e na sua estrutura.

Para fazer alterações nos dados, basta clicar com o botão direito do rato na tabela em que deseja alterar os seus dados e seleccionar a opção **Edit Table Data** ou passar o cursor sobre o nome da tabela e clicar no ícone da tabela com um pequeno raio, dependendo da versão instalada. Neste caso, fará o teste com a tabela “equipa”, mais especificamente com a coluna “ultimoPresidente”. Irá atualizar as seguintes equipas com os respetivos presidentes:

- Pinto da Costa (FC Porto).
- Luís Filipe Vieira (Benfica).
- Frederico Varandas (Sporting).
- Carlos Pereira (Marítimo).
- António Salvador (Sp. Braga).
- Vítor Magalhães (Mor).
- João Loureiro (Boa).
- Rodrigo Nunes (Fei).
- Gilberto Coimbra (Ton).
- Fernando Oliveira (VFC).
- Alexandre Faria (Estoril).
- Luiz Carlos Andrade (Aves).
- Patrick Morais de Carvalho (Belenenses).



1 • `SELECT * FROM futebol.equipa;`

Result Grid

equipaID	nomeEquipa	paisID	dataFundacao	ultimoPresidente
6	Moreirense	POR	NULL	Vitor Magalhães
21	Feirense	POR	NULL	Rodrigo Nunes
17	Porto	POR	NULL	Pinto da Costa
9	Belenenses SAD	POR	NULL	Patrick Morais de Carvalho
10	Benfica	POR	NULL	Luis Filipe Vieira
18	Rio Ave	POR	NULL	Luis Carlos Andrade
19	Tondela	POR	NULL	Gilberto Coimbra
11	Sporting	POR	NULL	Frederico Varandas
8	Vitória	POR	NULL	Fernando Oliveira
12	Marítimo	POR	NULL	Carlos Pereira
7	Sporting de Braga	POR	NULL	António Salvador
28	Estoril Praia	POR	NULL	Alexandre Faria
2	Farense	POR	NULL	NULL
3	Portimonense	POR	NULL	NULL
4	Famalicão	POR	NULL	NULL
5	Gil Vicente	POR	NULL	NULL
13	Nacional	POR	NULL	NULL

Apply Revert

Uma vez realizadas estas alterações na tabela, deverá clicar no botão **Apply** para guardar os dados inseridos: aparecerá uma janela com o código correspondente às consultas de atualização que serão executadas para que as alterações tenham efeito. Aplique as alterações e, se tudo correr bem, estas serão introduzidas sem problemas.

Falta, então, atualizar os dados sobre as datas de algumas finais da Taça de Portugal. Seguindo a mesma lógica utilizada na unidade didática, use o editor fornecido pelo MySQL Workbench para executar as seguintes consultas, uma após a outra. As datas utilizadas são meramente indicativas e não correspondem a datas reais.

```
UPDATE campeonato c SET c.data = '1997-06-28' WHERE c.campeonatoID = 140;
```

```
UPDATE campeonato c SET c.data = '1998-04-29' WHERE c.campeonatoID = 141;
```

```
UPDATE campeonato c SET c.data = '1999-06-26' WHERE c.campeonatoID = 142;
```

```
UPDATE campeonato c SET c.data = '2000-05-27' WHERE c.campeonatoID = 143;
```

```

UPDATE campeonato c SET c.data = '2001-06-30' WHERE c.campeonatoID
= 144;

UPDATE campeonato c SET c.data = '2002-03-06' WHERE c.campeonatoID
= 145;

UPDATE campeonato c SET c.data = '2003-06-28' WHERE c.campeonatoID
= 146;

UPDATE campeonato c SET c.data = '2004-03-17' WHERE c.campeonatoID
= 147;

UPDATE campeonato c SET c.data = '2005-06-11' WHERE c.campeonatoID
= 148;

UPDATE campeonato c SET c.data = '2006-04-12' WHERE c.campeonatoID
= 149;

UPDATE campeonato c SET c.data = '2007-06-23' WHERE c.campeonatoID
= 150;

UPDATE campeonato c SET c.data = '2008-04-16' WHERE c.campeonatoID
= 151;

UPDATE campeonato c SET c.data = '2009-05-13' WHERE c.campeonatoID
= 152;

UPDATE campeonato c SET c.data = '2010-05-19' WHERE c.campeonatoID
= 153;

UPDATE campeonato c SET c.data = '2011-06-29' WHERE c.campeonatoID
= 154;

UPDATE campeonato c SET c.data = '2012-05-19' WHERE c.campeonatoID
= 155;

UPDATE campeonato c SET c.data = '2013-04-23' WHERE c.campeonatoID
= 156;

UPDATE campeonato c SET c.data = '2014-06-25' WHERE c.campeonatoID
= 157;

UPDATE campeonato c SET c.data = '2015-05-21' WHERE c.campeonatoID
= 158;

UPDATE campeonato c SET c.data = '2016-06-10' WHERE c.campeonatoID
= 159;

UPDATE campeonato c SET c.data = '2017-04-11' WHERE c.campeonatoID
= 160;

UPDATE campeonato c SET c.data = '2018-05-07' WHERE c.campeonatoID
= 161;

```

```
UPDATE campeonato c SET c.data = '2019-04-30' WHERE c.campeonatoID  
= 162;
```

```
UPDATE campeonato c SET c.data = '2020-05-19' WHERE c.campeonatoID  
= 163;
```

Para verificar se os dados foram atualizados corretamente, pode utilizar a consulta que se segue. Com esta consulta, verá os registos cujo nome da competição começa com a palavra “Taça” e cuja data não seja nula.

```
SELECT  
  
CONCAT(CAST(temp.anoInicio AS CHAR(4)), ' - ', CAST(temp.anoFim AS  
CHAR(4))) AS Temporada,  
  
comp.competicaoNome AS Competição,  
  
eq1.nomeEquipa AS Campeão,  
  
eq2.nomeEquipa AS SegundoClassificado,  
  
DATE_FORMAT(camp.data, '%d/%m/%Y') AS Data  
  
FROM  
  
campeonato camp  
  
INNER JOIN competicao comp ON  
  
(camp.competicaoID = comp.competicaoID)  
  
INNER JOIN temporada temp ON  
  
(camp.temporadaID = temp.temporadaID)  
  
INNER JOIN equipa eq1 ON  
  
(camp.campeao = eq1.equipaID)  
  
LEFT JOIN equipa eq2 ON  
  
(camp.subcampeao = eq2.equipaID)  
  
WHERE  
  
comp.competicaoNome LIKE 'Taça%'  
  
AND camp.data IS NOT NULL  
  
ORDER BY
```



```
Temporada;
```

Após aprender a introduzir todas estas mudanças, é possível fazer qualquer outra alteração, pois será sempre uma questão de repetir continuamente algumas das operações mostradas durante a unidade didática e esta prática.

Por isso, recomenda-se que tente modificar a coluna “data” da tabela “campeonato” para poder definir a data e a hora em que ocorreram as finais dos campeonatos. Depois, pode atualizar os registos correspondentes na tabela “campeonato” para que, além da data, também seja registada a hora a que ocorreram as finais (os já atualizados na prática).

Depois de tentar introduzir estas alterações, continue a prática, pois estas mudanças também serão explicadas.

Primeiro, para modificar a coluna “data” da tabela “campeonato”, para que esta também aceite horas, basta alterar o tipo de dado de **DATE** para **DATETIME**.

Para fazê-lo, vamos executar a seguinte instrução do MySQL Workbench:

```
ALTER TABLE campeonato MODIFY COLUMN data DATETIME;
```

competicao x

Limit to 1000 rows

- 1 • ALTER TABLE campeonato MODIFY COLUMN data DATETIME;
- 2 • SELECT \* FROM campeonato;

Result Grid

campeonatoID	competicaoID	temporadaID	campeao	subcampeao	data
142	1	110	11	31	1999-06-26 00:00:00
143	1	111	17	31	2000-05-27 00:00:00
144	1	112	10	17	2001-06-30 00:00:00
145	1	113	8	10	2002-03-06 00:00:00
146	1	114	17	8	2003-06-28 00:00:00
147	1	115	11	9	2004-03-17 00:00:00
148	1	116	11	17	2005-06-11 00:00:00
149	1	117	17	16	2006-04-12 00:00:00
150	1	118	17	36	2007-06-23 00:00:00
151	1	119	17	77	2008-04-16 00:00:00
152	1	120	25	11	2009-05-13 00:00:00
153	1	121	77	10	2010-05-19 00:00:00
154	1	122	10	18	2011-06-29 00:00:00
155	1	123	11	7	2012-05-19 00:00:00
156	1	124	7	17	2013-04-23 00:00:00
157	1	125	10	77	2014-06-25 00:00:00
158	1	126	18	11	2015-05-21 00:00:00

campeonato 6 x

Apply Revert

Como pode ver, os registos que já tinham uma data foram mantidos e, além disso, também foi adicionada a hora 00:00:00. Este é o teste que confirma que já é possível armazenar a data e a hora.

Por fim, para atualizar as datas e colocar as horas das finais, basta copiar os comandos utilizados anteriormente e adicionar as horas da seguinte forma (referimos, novamente, que as horas e as datas são meramente indicativas):

```
UPDATE campeonato c SET c.data = '1997-06-28 22:00:00' WHERE
c.campeonatoID = 140;
```

```
UPDATE campeonato c SET c.data = '1998-04-29 21:30:00' WHERE
c.campeonatoID = 141;
```

```
UPDATE campeonato c SET c.data = '1999-06-26 21:30:00' WHERE
c.campeonatoID = 142;
```

```
UPDATE campeonato c SET c.data = '2000-05-27 20:00:00' WHERE
c.campeonatoID = 143;
```

```
UPDATE campeonato c SET c.data = '2001-06-30 22:30:00' WHERE
c.campeonatoID = 144;
```

```
UPDATE campeonato c SET c.data = '2002-03-06 21:00:00' WHERE
c.campeonatoID = 145;
```

```

UPDATE campeonato c SET c.data = '2003-06-28 21:30:00' WHERE
c.campeonatoID = 146;

UPDATE campeonato c SET c.data = '2004-03-17 20:00:00' WHERE
c.campeonatoID = 147;

UPDATE campeonato c SET c.data = '2005-06-11 20:30:00' WHERE
c.campeonatoID = 148;

UPDATE campeonato c SET c.data = '2006-04-12 20:00:00' WHERE
c.campeonatoID = 149;

UPDATE campeonato c SET c.data = '2007-06-23 21:30:00' WHERE
c.campeonatoID = 150;

UPDATE campeonato c SET c.data = '2008-04-16 22:00:00' WHERE
c.campeonatoID = 151;

UPDATE campeonato c SET c.data = '2009-05-13 19:30:00' WHERE
c.campeonatoID = 152;

UPDATE campeonato c SET c.data = '2010-05-19 20:30:00' WHERE
c.campeonatoID = 153;

UPDATE campeonato c SET c.data = '2011-06-29 21:00:00' WHERE
c.campeonatoID = 154;

UPDATE campeonato c SET c.data = '2012-05-19 20:30:00' WHERE
c.campeonatoID = 155;

UPDATE campeonato c SET c.data = '2013-04-23 19:30:00' WHERE
c.campeonatoID = 156;

UPDATE campeonato c SET c.data = '2014-06-25 20:00:00' WHERE
c.campeonatoID = 157;

UPDATE campeonato c SET c.data = '2015-05-21 20:30:00' WHERE
c.campeonatoID = 158;

UPDATE campeonato c SET c.data = '2016-06-10 21:30:00' WHERE
c.campeonatoID = 159;

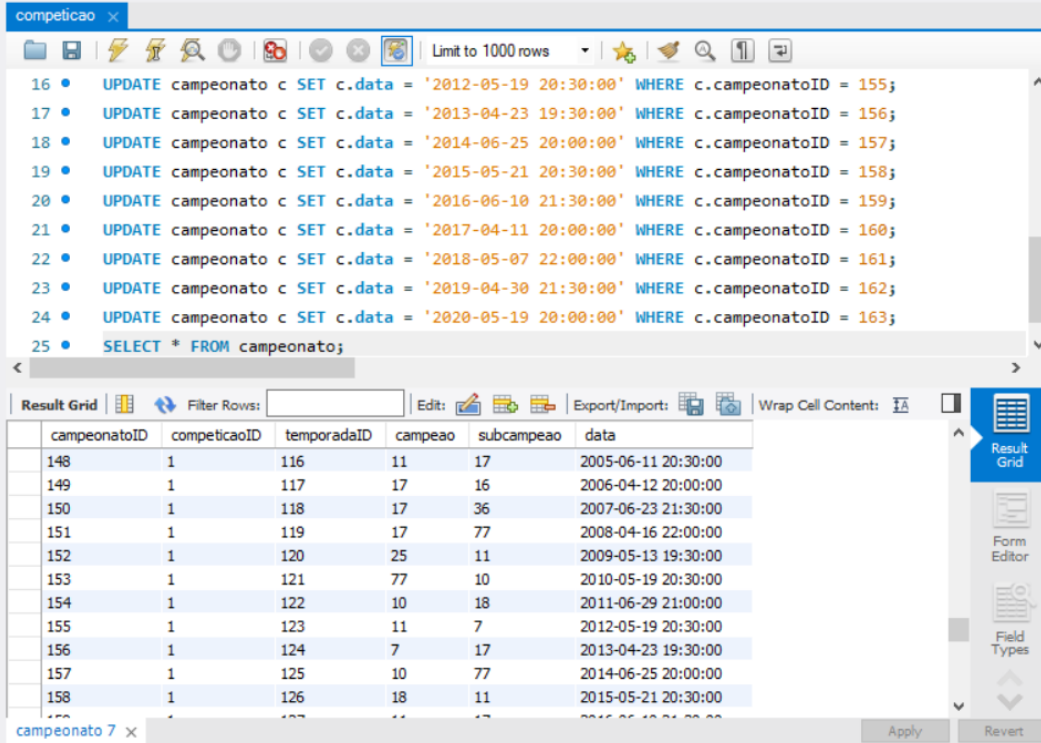
UPDATE campeonato c SET c.data = '2017-04-11 20:00:00' WHERE
c.campeonatoID = 160;

UPDATE campeonato c SET c.data = '2018-05-07 22:00:00' WHERE
c.campeonatoID = 161;

UPDATE campeonato c SET c.data = '2019-04-30 21:30:00' WHERE
c.campeonatoID = 162;

```

```
UPDATE campeonato c SET c.data = '2020-05-19 20:00:00' WHERE
c.campeonatoID = 163;
```



The screenshot shows a MySQL database management tool interface. The top panel displays a list of SQL statements, including 14 UPDATE statements for the 'campeonato' table, each setting a specific date and time for a given 'campeonatoID'. The bottom panel shows a table view of the 'campeonato' table with columns: campeonatoID, competicaoID, temporadaID, campeao, subcampeao, and data. The table contains 14 rows of data, corresponding to the IDs 148 through 162.

campeonatoID	competicaoID	temporadaID	campeao	subcampeao	data
148	1	116	11	17	2005-06-11 20:30:00
149	1	117	17	16	2006-04-12 20:00:00
150	1	118	17	36	2007-06-23 21:30:00
151	1	119	17	77	2008-04-16 22:00:00
152	1	120	25	11	2009-05-13 19:30:00
153	1	121	77	10	2010-05-19 20:30:00
154	1	122	10	18	2011-06-29 21:00:00
155	1	123	11	7	2012-05-19 20:30:00
156	1	124	7	17	2013-04-23 19:30:00
157	1	125	10	77	2014-06-25 20:00:00
158	1	126	18	11	2015-05-21 20:30:00

Como alternativa à execução das consultas anteriores, poderia também ter realizado as modificações diretamente no modo gráfico, na coluna “data” da tabela “campeonato”.