Universidade Federal de Minas Gerais
Colégio Técnico - COLTEC
Prof. Márcio Fantini
Estudo Orientado 1- Mysql: Resumo da Teoria e Exemplos
Operando, Ordenando e Formatando Resultados

1 Introdução

Nesse estudo orientado vamos rever algumas instruções básicas e formas de gerar saídas formatas.
As tabelas apresentadas abaixo serão usadas nos exemplos para as instruções, funções, predicados e cláusulas aqui apresentadas. (tabelas 1, 1, 3). Elas (e os textos apresentandos nessa nota) foram retiradas/adaptados do livro [1].

Código	Nome
1	Mirandela Editora
2	Editora Via-Norte
3	Editora Ilhas Tijucas
4	Maria José Editora

Tabela 1: Banco de Dados Livraria01 - Tabela Editora.

Sigla	Descrição
B	Banco de Dados
P	Programação
R	Redes
S	Sistemas Operacionais

Tabela 2: Banco de Dados Livraria01 -Tabela Assunto.

Código	Título	Preço	Lançamento	Assunto	Editora
1	Banco de Dados para Web	31,20	10/01/1999	B	1
2	Programando em Linguagem C	30,00	01/10/1997	P	1
3	Programando em Linguagem C++	111,50	01/11/1998	P	3
4	Banco de Dados na Bioinformática	48,00		B	2
5	Redes de Computadores	42,00	01/09/1996	R	2

Tabela 3: Banco de Dados Livraria01 -Tabela Livro.

2 Revendo Exemplos com Select

2.1 Iniciando o uso do Banco Livraria

USE e SHOW TABLES

```
      mysql> use livraria
Database changed
mysql> show tables;
+--------------------+
| Tables_in_livraria |
+--------------------+
| assunto            |
| editora            |
| livro              |
+--------------------+
3 rows in set (0.02 sec)

mysql> select * from livro;
+--------+-----------------------------------+--------+------------+---------+---------+
| codigo | titulo                            | preco  | lancamento | assunto | editora |
+--------+-----------------------------------+--------+------------+---------+---------+
|      1 | Banco de Dados para Web           |  31.20 | 1999-01-10 | B       |       1 |
|      2 | Programando em Linguagem C        |  30.00 | 1997-10-01 | P       |       1 |
|      3 | Programando em Linguagem C++      | 111.50 | 1998-11-01 | P       |       3 |
|      4 | Banco de Dados na Bioinformática  |  48.00 |            | B       |       2 |
|      5 | Redes de Computadores             |  42.00 | 1996-09-01 | R       |       2 |
+--------+-----------------------------------+--------+------------+---------+---------+
5 rows in set (0.00 sec)
```

2.1.1 SELECT com WHERE

```
      mysql> select * from livro where preco > 30;
+--------+-----------------------------------+--------+------------+---------+---------+
| codigo | titulo                            | preco  | lancamento | assunto | editora |
+--------+-----------------------------------+--------+------------+---------+---------+
|      1 | Banco de Dados para Web           |  31.20 | 1999-01-10 | B       |       1 |
|      3 | Programando em Linguagem C++      | 111.50 | 1998-11-01 | P       |       3 |
|      4 | Banco de Dados na Bioinformática  |  48.00 |            | B       |       2 |
|      5 | Redes de Computadores             |  42.00 | 1996-09-01 | R       |       2 |
+--------+-----------------------------------+--------+------------+---------+---------+
4 rows in set (0.00 sec)
```

```
mysql> select * from livro where preco > 30
    -> and assunto = 'B';
+--------+-----------------------------------+-------+------------+---------+---------+
| codigo | titulo                            | preco | lancamento | assunto | editora |
+--------+-----------------------------------+-------+------------+---------+---------+
|      1 | Banco de Dados para Web           | 31.20 | 1999-01-10 | B       |       1 |
|      4 | Banco de Dados na Bioinformática  | 48.00 |            | B       |       2 |
+--------+-----------------------------------+-------+------------+---------+---------+
2 rows in set (0.00 sec)


mysql> select * from livro
    -> where lancamento is NULL;
+--------+-----------------------------------+-------+------------+---------+---------+
| codigo | titulo                            | preco | lancamento | assunto | editora |
+--------+-----------------------------------+-------+------------+---------+---------+
|      4 | Banco de Dados na Bioinformática  | 48.00 | NULL       | B       |       2 |
+--------+-----------------------------------+-------+------------+---------+---------+
1 row in set (0.00 sec)

mysql> select * from livro where lancamento is NOT NULL;
+--------+-------------------------------+--------+------------+---------+---------+
| codigo | titulo                        | preco  | lancamento | assunto | editora |
+--------+-------------------------------+--------+------------+---------+---------+
|      1 | Banco de Dados para Web       |  31.20 | 1999-01-10 | B       |       1 |
|      2 | Programando em Linguagem C    |  30.00 | 1997-10-01 | P       |       1 |
|      3 | Programando em Linguagem C++  | 111.50 | 1998-11-01 | P       |       3 |
|      5 | Redes de Computadores         |  42.00 | 1996-09-01 | R       |       2 |
+--------+-------------------------------+--------+------------+---------+---------+
4 rows in set (0.00 sec)

mysql>
```

3 Usando LIKE

```
      mysql> select * from livro where titulo LIKE "BANCO%";
+--------+-----------------------------------+-------+------------+---------+---------+
| codigo | titulo                            | preco | lancamento | assunto | editora |
+--------+-----------------------------------+-------+------------+---------+---------+
|      1 | Banco de Dados para Web           | 31.20 | 1999-01-10 | B       |       1 |
|      4 | Banco de Dados na Bioinformática  | 48.00 | NULL       | B       |       2 |
+--------+-----------------------------------+-------+------------+---------+---------+
2 rows in set (0.00 sec)

mysql> select * from editora where nome LIKE "edit%";
+---------+----------------------+
| codigo1 | nome                 |
+---------+----------------------+
|       2 | Editora via-norte    |
|       3 | Editora Ilhas Tijucas |
|       5 | Editora Bom Livro    |
+---------+----------------------+
3 rows in set (0.00 sec)


mysql> select codigo,titulo,preco
    -> from livro
    -> where titulo like '%LINGUAGEM C%';
+--------+-------------------------------+--------+
| codigo | titulo                        | preco  |
+--------+-------------------------------+--------+
|      2 | Programando em Linguagem C    |  30.00 |
|      3 | Programando em Linguagem C++ | 111.50 |
+--------+-------------------------------+--------+
2 rows in set (0.01 sec)
```

4 Agrupando Dados

Na linguagem SQL são definidas várias funções que operam sobe grupos de dados. Tais funções, usualmente, realizam operações ou comparações sobre um conjunto de dados e retornam, como resultado, uma relação de apenas uma linha e uma coluna. São chamadas **funções agregadas**. Comumente, recebem apenas um parâmetro. Nessa seção veremos alguns exemplos dessas funções.

4.1 Contagem

Muitas vezes é necessário contar a quantidade de linhas que satisfazem determinada condição. Para isso usamos a função **COUNT**.
Como parâmetro para a função **COUNT** podemos utilizar o nome de uma coluna ou o caractere $\star$, para contar todas as linhas da tabela. No caso da utilização de uma coluna como parâmetro, o resultado obtido é o número de ocorrências não-nulas desta coluna na tabela.

4.1.1 Exemplos

1. Contar a quantidade de linhas da tabela de livros:

```
SELECT COUNT (*)
FROM LIVRO
```

2. Contar a quantidade de linhas da tabela de livros com a coluna de código preenchida:

```
SELECT COUNT (CODIGO)
FROM LIVRO
```

3. Contar a quantidade de linhas da tabela de livros com a coluna de data de lançamento preenchida.

```
SELECT COUNT (LANCAMENTO)
FROM LIVRO
```

4.2 SOMA

Para somar os valores de uma coluna para um grupo de dados usamos a função SUM. Por exemplo:

- Somatório dos preços da tabela de livros

```
SELECT SUM (PRECO)
FROM LIVRO
```

4.3 Média

Para calcular a média aritmética dos valores de uma coluna, utilizams a função **AVG**. Essa função retornará a média considerando apenas os valores não nulos da coluna especificada. Exemplo:

- Média dos preços de livros

```
SELECT AVG (PRECO)
FROM LIVRO
```

4.4 Valores Máximo e Mínimo

Exemplo: obtenha o maior preço da tabela de livros, para livros cujo assunto seja 'P'.

```
SELECT MAX(PRECO)
FROM LIVRO
WHERE ASSUNTO = 'P'
```

Agora encontre o menor preço para assunto 'B'.

```
SELECT MIN(PRECO)
FROM LIVRO
WHERE ASSUNTO = 'B'
```

5 Cláusula GROUP BY

As funções apresentadas na seção anterior realizam operações sobre conjuntos de dados, transformando vários valores de entrada em um único de saída. Com essas funções foi possível, por exemplo, calcular uma média para um conjunto de livros. Como faríamos para calcular o preço médio dos livros **para cada assunto?**
A consulta a seguir NÃO poderia ser usada para isso:

```
SELECT ASSUNTO, AVG(PRECO)
from LIVRO
```

pois ela resulta na média total, sem separar por assunto. Para "separar por assunto"usamos a cláusula GROUP BY, que faz com que os dados sejam organizados pelas colunas que são especificadas por ela. Assim somentes valores distintos aparecerão no resultado, sendo que os semelhantes foram agrupados.
Acompanhe os exemplos.

5.0.1 Exemplos

1. Qual o preço médio dos livros paa cada assunto?

   ```
   SELECT ASSUNTO, AVG(PRECO)
   from LIVRO
   GROUP BY ASSUNTO
   ```

2. Quantos livros existem para cada assunto?

   ```
   SELECT ASSUNTO, COUNT(*)
   from LIVRO
   GROUP BY ASSUNTO
   ```

3. Qual o preço do livro mais caro de cada assunto, dentre os que já foram lançados?

```
SELECT ASSUNTO, MAX(PRECO)
from LIVRO
where lancamento is not null
GROUP BY ASSUNTO
```

4. Quantos livros já foram lançados por cada editora?

```
SELECT EDITORA, COUNT(*)
from LIVRO
WHERE LANCAMENTO IS NOT NULL
GROUP BY EDITORA
```

5. Qual a média de preço para cada código e assunto?

```
SELECT CODIGO, ASSUNTO, AVG(PRECO)
from LIVRO
GROUP BY CODIGO, ASSUNTO
```

Repare que para cada valor distinto da combinação das colunas especificadas na cláusula GROUP BY, será gerada uma linha no resultado da consulta. Repare o último exemplo acima.

6 Cláusula HAVING

Nas seções anteriores vimos como montar grupos de dados e como utilizar funções agregadas sobre esse grupos. Foi mostrado também que podemos utilizar a cláusula WHERE para fazer restrições quanto aos dados que serão considerados para a montagem dos grupos de dados. Porém, cláusulas WHERE não permitem realizar restrições com base nos resultados da funções agregadas. Para isso deve-se utilizar a cláusula HAVING.
A cláusula HAVING será seguida de uma expressão lógica (que poderá ser composta ou "simples"), de forma idêntica ao que foi apresentado para a cláusula WHERE.
A cláusula HAVING serve como um filtro (assim como WHERE). No caso da WHERE, o filtro é aplicado quando as linhas são recuperadas do banco de dados, fazendo que essas nem cheguem a ser consideradas quando da realização de agrupamentos ou na execução de funções agregadas.
As restrições HAVING serão aplicadas somente após a recuperação das linhas do banco de dados, da montagem dos grupos e da execução de funções agregadas. Por isso é possível utilizar funções agregadas em expressões lógicas da cláusula HAVING.

6.1 Sintaxe Básica

```
SELECT COL1, COL2,...,COLN, FUNCAO1,...FUNCAON
FROM NOME_TABELA
WHERE EXPRESSAO_LOGICA_WHERE
GROUP BY COL1,COL2, ...,COLN
HAVING EXPRESSAO_LOGICA_WHERE
```

6.2 Exemplos

1. Quais os assuntos cujo preço médio dos livros ultrapassa $R\$50,00$?

```
select assunto
from livro
group by assunto
having avg(preco)>50 (aqui nao se aplica o where)
```

2. Quais os assuntos que possuem pelo menos dois livros?

```
select assunto, count(*)
from livro
group by assunto
having count(*)>1
```

3. Quais os assuntos que possuem pelo menos dois livros já lançados?

```
select assunto, count(*)
from livro
where lancamento is not null
group by assunto
having count(*)>1
```

7 Apelidos

A SQL possibilita atribuir apelidos para colunas e tabelas. Essa prática é usada para simplificar expressões mais complexas, ou referenciar tabelas ou colunas de forma mais simples.

7.1 Apelidos para coluna

Para atribuir apelido para coluna resultante de um consulta, deve-se fazer como indicado abaixo:

```
SELECT col1 AS apelido
FROM tabela1
```

Sendo

```
SELECT e FROM: comandos da seleção
tabela1 e col1: nomes da tabela e coluna
apelido: é o apelido dado para a coluna
AS usada para indicar o novo nome (apelido)
```

Veja o exemplo para nosso banco de dados livraria.

- Obtenha o maior preço da tabela de livros para livros cujo assunto seja 'P'.

No caso abaixo, a consulta é feita para que o resultado se apresente com um coluna com o nome de $PRECO_MAXIMO$.

```
SELECT MAX(PRECO) AS PRECO_MAXIMO
FROM LIVRO
WHERE ASSUNTO ="P"
```

Repare que a saída fara referência a tabela "preco_maximo".

7.2 Apelidos para tabelas

Em consultas do Mysql é possível fazer referência à coluna em associação com sua tabela, a partir da **notação de ponto**. Por exemplo, pode-se referenciar uma coluna com o comando:

nome-coluna.nome-tabela

Em algumas situações, o nome da tabela pode ser simplificado por apelidos (alias). Em alguns casos é obrigatória a utilização de apelidos (em junções, por exemplo). Para usar apelido para tabela, faça como indicado abaixo:

```
SELECT col1
FROM tabela1 AS apelido
```

Sendo

```
SELECT e FROM: comandos da seleção
tabela1 e col1 : nomes da tabela e da coluna
apelido: é o apelido dado para a tabela
AS: usado para indicar o apelido (AS é opcional para tabelas).
```

Veja o exemplo.

```
SELECT MAX(L.PRECO) AS PRECO_MAXIMO
FROM LIVRO AS L
WHERE ASSUNTO ="P"
```

Esse exemplo poderia ser rodado sem a palavra AS

```
SELECT MAX(L.PRECO) AS PRECO_MAXIMO
FROM LIVRO L
WHERE ASSUNTO ="P"
```

8 Concatenação: Função CONCAT()

É possível usar a função CONCAT() da cláusula select ou concatenar colunas e textos com os resultados de uma consulta.
Veja o exemplo.

```
mysql> select 'LIVRO: ' as texto,titulo
    -> from livro;
+---------+-----------------------------------+
| texto   | titulo                            |
+---------+-----------------------------------+
| LIVRO:  | Banco de Dados para Web           |
| LIVRO:  | Programando em Linguagem C        |
| LIVRO:  | Programando em Linguagem C++      |
| LIVRO:  | Banco de Dados na Bioinformática  |
| LIVRO:  | Redes de Computadores             |
+---------+-----------------------------------+
5 rows in set (0.00 sec)


mysql> select concat('LIVRO: ',titulo) as texto from livro;
+----------------------------------------+
| texto                                  |
+----------------------------------------+
| LIVRO: Banco de Dados para Web         |
| LIVRO: Programando em Linguagem C      |
| LIVRO: Programando em Linguagem C++    |
| LIVRO: Banco de Dados na Bioinformática|
| LIVRO: Redes de Computadores           |
+----------------------------------------+
5 rows in set (0.00 sec)
```

Na segunda consula realizada acima, concatenamos a string "LIVRO: "com o resultado da busca da coluna "titulo".

9 Operações Aritméticas

É possível usar operadores aritméticos e funções matemáticas nas consultas.

1. Listar novos precos dos livros se os valores forem reajustados em 10%.

```
SELECT TITULO, PRECO*1.1 as NOVO_PRECO
FROM LIVRO
```

10 DISTINCT

Quando realizamos consultas em tabelas podemos obter linhas repetidas. Para eliminar repetições, em relações resultantes de consultas, foi definido o predicado DISTINCT.
Exemplo: recuperar os assuntos distintos da tabela livros.

```
SELECT DISTINCT ASSUNTO as ASSUNTO
from livro
```

11 Ordenando Resultados com ORDER BY

A cláusula ORDER BY é sempre posicionada como a última em uma consulta. Sua sintaxe é:

```
SELECT COL1, COL2, ..., COLN
from NOME_TABELA
WHERE CONDICAO
GROUP by COL1, COL2, ...,COLN
HAVING ExPRESSAO_LOGICA_HAVING
ORDER BY Col1 [DESC,ASC], COL2 [DESC,ASC]
```

11.1 Exemplos

1. Gerar listagem dos livros contendo assunto, título e preço. A listagem deve estar ordenada em ordem crescente de assunto e decrescente de preço.

   ```
   SELECT ASSUNTO, TITULO, PRECO
   from livro
   order by ASSUNTO, PRECO DESC
   ```

2. Repita a consulta acima, agora em ordem crescente de título e decrescente de preço.

   ```
   SELECT ASSUNTO, TITULO, PRECO
   from livro
   order by 2, PRECO DESC
   ```

12 Predicado IN

Quando desejarmos testar se um valor está contido em uma lista de valores, podemos usar o predicado IN. Por exemplo, selecione os livros cujo assunto seja 'S' ou 'P' ou 'B'.

```
... WHERE ASSUNTO IN ('S', 'P', 'B')
```

13 Strings

Podemos formatar a saída com funções específicas que manipulam os resultados de consultas SQL, formatando-os ou alterando-os
Algumas dessas funções são:

- UPPER: retorna todos caracteres em maiúsculo
- LOWER: retorna todos caracteres em minúsculo
- TRIM: retira os espaços da string
- SUBSTRING: retorna um trecho da string (recebe além da string, o início e comprimento do trecho)
- LENGTH: retorna o comprimento da string

Veja algums exemplos

```
mysql> select concat(trim('      livro:     '),titulo) as Livro
    -> from livro
    -> where assunto = 'B';
+----------------------------------------+
| Livro                                  |
+----------------------------------------+
| livro:Banco de Dados para Web          |
| livro:Banco de Dados na Bioinformática |
+----------------------------------------+
2 rows in set (0.00 sec)



mysql> select substring(titulo,1,10) as titulo
    -> from livro;
+------------+
| titulo     |
+------------+
| Banco de D |
| Programand |
| Programand |
| Banco de D |
| Redes de C |
+------------+
5 rows in set (0.00 sec)
```

```
mysql> select length(titulo) as comprimento
    -> from livro
    -> where assunto = 'B';
+-------------+
| comprimento |
+-------------+
|          23 |
|          33 |
+-------------+
2 rows in set (0.00 sec)


mysql> select UPPER(titulo) from livro
    -> where assunto = 'R';
+-----------------------+
| UPPER(titulo)         |
+-----------------------+
| REDES DE COMPUTADORES |
+-----------------------+
1 row in set (0.00 sec)
```

14 Funções de Data

Podemos formatar campos do tipo DATE usando funções específicas, dentre elas: DAY, MONTH e YEAR. Veja alguns exemplos.

```
mysql> select titulo,day(lancamento) as 'Dia Lançamento' from livro;
+----------------------------------+-----------------+
| titulo                           | Dia Lançamento  |
+----------------------------------+-----------------+
| Banco de Dados para Web          |              10 |
| Programando em Linguagem C       |               1 |
| Programando em Linguagem C++     |               1 |
| Banco de Dados na Bioinformática |            NULL |
| Redes de Computadores            |               1 |
+----------------------------------+-----------------+
5 rows in set (0.00 sec)


mysql> select titulo,year(lancamento) as 'Ano Lançamento' from livro;
```

```
+----------------------------------+-----------------+
| titulo                           | Ano Lançamento  |
+----------------------------------+-----------------+
| Banco de Dados para Web          |            1999 |
| Programando em Linguagem C       |            1997 |
| Programando em Linguagem C++     |            1998 |
| Banco de Dados na Bioinformática |            NULL |
| Redes de Computadores            |            1996 |
+----------------------------------+-----------------+
5 rows in set (0.02 sec)


mysql> select titulo, month(lancamento) as Mes,
    -> year(lancamento) as Ano
    -> from livro;
+----------------------------------+------+------+
| titulo                           | Mes  | Ano  |
+----------------------------------+------+------+
| Banco de Dados para Web          |    1 | 1999 |
| Programando em Linguagem C       |   10 | 1997 |
| Programando em Linguagem C++     |   11 | 1998 |
| Banco de Dados na Bioinformática | NULL | NULL |
| Redes de Computadores            |    9 | 1996 |
+----------------------------------+------+------+
5 rows in set (0.00 sec)
```

Outras Funções de Data e Tempo

Algumas funções comumente usadas:

- ADDDATE() - adiciona data
- ADDTIME() - adiciona tempo (hora:minutos:segundos)
- CONVERT_TZ(); -converte ”time-zone”(fuso horário).
- CURDATE(), CURRENT_DATE() - retorna a data atual
- CURRENT_TIME();
- CURRENT_TIME_STAMP();
- DATE_FORMAT();

Veja mais em `https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html`

15 Funções Aritméticas e Resultados Numéricos

15.1 FLOOR() e CEIL()

As funções CEIL() e FLOOR arredondam, respectivamente, para o inteiro imediatamente superior e imediatamente inferior. Veja os exemplos:

```
mysql> select codigo,preco from livro where codigo < 4;
+--------+--------+
| codigo | preco  |
+--------+--------+
|      1 |  31.20 |
|      2 |  30.00 |
|      3 | 111.50 |
+--------+--------+
3 rows in set (0.00 sec)


mysql> select codigo,CEIL(preco) from livro;
+--------+-------------+
| codigo | CEIL(preco) |
+--------+-------------+
|      1 |          32 |
|      2 |          30 |
|      3 |         112 |
|      4 |          48 |
|      5 |          42 |
+--------+-------------+
5 rows in set (0.03 sec)

mysql> select codigo,FLOOR(preco) from livro;
+--------+--------------+
| codigo | FLOOR(preco) |
+--------+--------------+
|      1 |           31 |
|      2 |           30 |
|      3 |          111 |
|      4 |           48 |
|      5 |           42 |
+--------+--------------+
5 rows in set (0.00 sec)
```

15.2 POWER(), ROUND()

A função POWER eleva o número a n-ésima potência. A função ROUND arredonda. Veja o exemplo:

```
mysql> select codigo,ROUND(POWER(preco,2)) from livro;
+--------+-----------------------+
| codigo | ROUND(POWER(preco,2)) |
+--------+-----------------------+
|      1 |                   973 |
|      2 |                   900 |
|      3 |                 12432 |
|      4 |                  2304 |
|      5 |                  1764 |
+--------+-----------------------+
5 rows in set (0.00 sec)
```

Veja a lista completa de funções matemáticas no link
`https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html`

16 Exercícios

Baseados e adaptados de [1]. Faça as consultas pedidas para o banco de dados **livraria**.

1. Liste os nomes das editoras, ordenadas em ordem crescente. A coluna dos nomes deve ter o label "Editora".

2. Mostre os títulos e os preços dos livros, ordenados em ordem DECRESCENTE.

3. Mostre os códigos e os preços dos livros e mais três colunas, com valores dos preços acrescidos de 10%, 20% e 30%.

4. Mostre o código da editora, a sigla do assunto, o mês e o ano de lancamento (como mes:ano), em uma única coluna, e o titulo dos livros que foram lancados depois de 1997. Ordene em ordem decrescente de preço e ascendente em relação ao código da editora.

5. Liste os livros (titulo) que possuem titulo superior a 20 caracteres;

17 Algumas Soluções

```
mysql> select codigo,preco,
   -> preco*1.1 as Preco10,
   -> preco*1.2 as Preco20, preco*1.3 as Preco30 from livro;
+--------+--------+---------+---------+---------+
| codigo | preco  | Preco10 | Preco20 | Preco30 |
+--------+--------+---------+---------+---------+
|      1 |  31.20 |  34.320 |  37.440 |  40.560 |
|      2 |  30.00 |  33.000 |  36.000 |  39.000 |
|      3 | 111.50 | 122.650 | 133.800 | 144.950 |
|      4 |  48.00 |  52.800 |  57.600 |  62.400 |
|      5 |  42.00 |  46.200 |  50.400 |  54.600 |
+--------+--------+---------+---------+---------+
5 rows in set (0.00 sec)
```

```
mysql> select editora,assunto,concat(month(lancamento),':',year(lancamento)) as "mes:ano",
    ->titulo from livro
    -> where year(lancamento)>1997 order by preco DESC, editora ASC;
+---------+---------+---------+-----------------------------+
| editora | assunto | mes:ano | titulo                      |
+---------+---------+---------+-----------------------------+
|       3 | P       | 11:1998 | Programando em Linguagem C++ |
|       1 | B       | 1:1999  | Banco de Dados para Web      |
+---------+---------+---------+-----------------------------+
2 rows in set (0.00 sec)
```

```
mysql> select titulo from livro where length(titulo) > 20;
+----------------------------------+
| titulo                           |
+----------------------------------+
| Banco de Dados para Web          |
| Programando em Linguagem C       |
| Programando em Linguagem C++     |
| Banco de Dados na Bioinformática |
| Redes de Computadores            |
+----------------------------------+
5 rows in set (0.00 sec)
```

Referências

[1] R.L. DE CARVALHO COSTA. *SQL Guia Prático - 2a edição*. BRASPORT.

Compilado 27 de agosto de 2022 - 04:55:57