



**Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios**  
mysql>

# Arquitetura Cliente Servidor e Tipos de Dados e seus Domínios

O objetivo deste tutorial é demonstrar como se configura e implementa a arquitetura cliente-servidor no SGBD MySQL. Além disso, iremos também abordar os seus tipos de dados e seus respectivos domínios.

Carlos Henrique Reis - 30415

Mateus Henrique Toledo - 34849

Victor Rodrigues da Silva - 31054



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

### Instalação

Para o funcionamento da Arquitetura Cliente-Servidor é necessária uma rede local configurada e funcionando corretamente, sendo que o servidor tem que possuir IP fixo. Desta forma iniciamos a configuração do servidor e posteriormente do(s) clientes(s).

### Servidor

Ambiente utilizado: Ubuntu 14.04 LTS

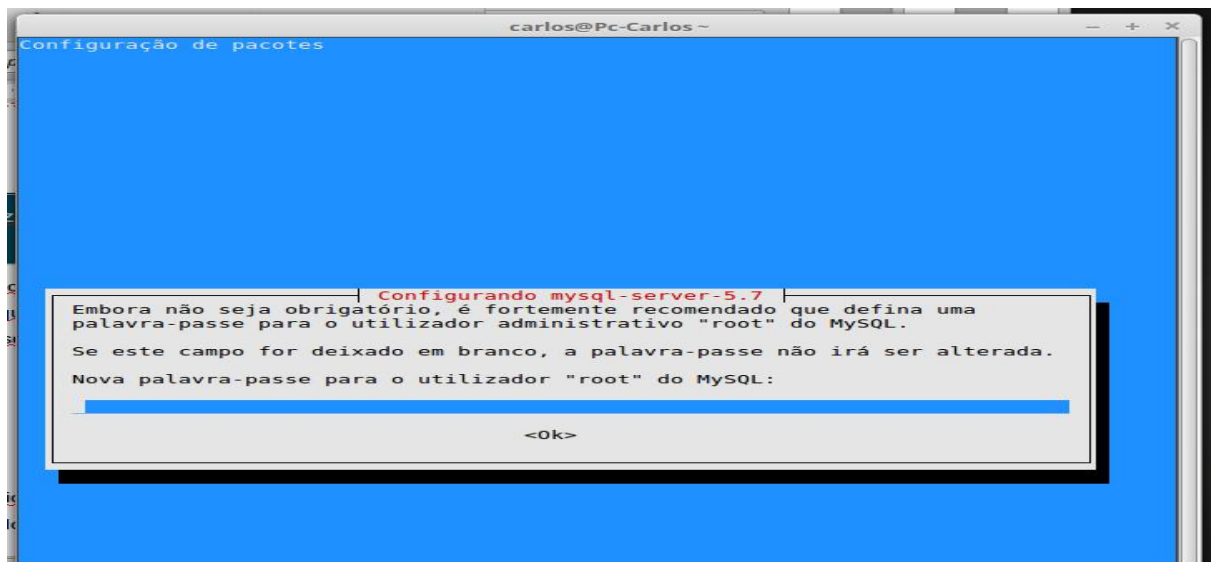
#### **1.1 Instalação do MySQL-Server:**

Assim como a instalação de qualquer pacote é necessário atualizar os índices dos repositórios dos pacotes instalados, para que a instalação de um novo pacote ocorra sem problemas. Execute os comandos abaixo, para a instalação do pacote mysql-server e execute os *scripts* de inicialização de segurança e de banco de dados.

```
sudo apt-get update
```

```
sudo apt-get install mysql-server
```

Após o comando entre com a senha do root:





## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

### 1.2. Configuração do MySQL-Server:

O comando abaixo iniciará um questionário com algumas configurações básicas que devem ser realizadas (solicitará senha *root* do MySQL):

```
sudo mysql_secure_installation
```

Pressione "**Enter**" e, em seguida, realize as seguintes configurações:

```
Set root password? [Y/n] y
```

Após selecionar "**y**", o MySQL configurará uma nova senha para o usuário *root* do serviço. Caso não seja necessário, essa etapa pode ser ignorada (digitando N ao invés de Y). É possível optar por configurar os seguintes parâmetros: remover usuários anônimos, não permitir o *login* de *root* remotamente, remover banco de dados de teste e seu acesso e recarregar tabelas de privilégios agora.

```
Remove anonymous users? [Y/n] y
```

```
Disallow root login remotely? [Y/n] y
```

```
Remove test database and access to it? [Y/n] y
```

```
Reload privilege tables now? [Y/n] y
```

Se você estiver usando uma versão do MySQL anterior à 5.7.6, você deve inicializar o diretório de dados executando:

```
sudo mysql_install_db
```

Se você instalou a versão 5.7 em alguma distribuição Debian (Ubuntu, Mint, etc), como na primeira etapa, o diretório de dados foi inicializado automaticamente, então você não precisa realizar esta operação. Se você tentar executar o comando anterior de qualquer maneira, você verá algum erro.

### 1.3. Testando o MySQL:

Caso o MySQL esteja instalado, o mesmo, está executando automaticamente. Para testar a execução verifique seu status.

```
service mysql status
```

Você verá a saída a seguir:



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

```
carlos@Pc-Carlos ~  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)  
   Active: active (running) since Qua 2017-03-15 13:29:22 BRT; 33min ago  
     Process: 1203 ExecStartPost=/usr/share/mysql/mysql-systemd-start post (code=exited, status=0/SUCCESS)  
     Process: 1192 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)  
    Main PID: 1202 (mysqld)  
      CGroup: /system.slice/mysql.service  
              └─1202 /usr/sbin/mysqld  
  
Mar 15 13:29:08 Pc-Carlos systemd[1]: Starting MySQL Community Server...  
Mar 15 13:29:22 Pc-Carlos systemd[1]: Started MySQL Community Server.  
~  
~  
~  
~  
~  
~  
~  
~  
~  
lines 1-11/11 (END)
```

Se o MySQL não estiver em execução, você pode iniciá-lo com:

```
sudo service mysql start
```

A execução pode ser encerrada com:

```
sudo service mysql stop
```

Você pode tentar se conectar ao banco de dados usando a ferramenta `mysqladmin`, que permite executar comandos administrativos. Por exemplo, este comando conecta ao MySQL como `root` (`-u root`), solicitando uma senha (`-p`) e retornar a versão.

```
mysqladmin -p -u root version
```

Você deve ver uma saída semelhante a esta:



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

```
carlos@Pc-Carlos ~ $ mysqladmin -p -u root version
Enter password:
mysqladmin Ver 8.42 Distrib 5.7.17, for Linux on x86_64
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version          5.7.17-0ubuntu0.16.04.1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/run/mysqld/mysqld.sock
Uptime:                 49 min 6 sec

Threads: 1  Questions: 5  Slow queries: 0  Opens: 107  Flush tables: 1  Open tables: 26  Queries per second avg: 0.001
carlos@Pc-Carlos ~ $
```

### 1.4. Configuração para acesso remoto:

Como comentado os requisitos básicos para fazer qualquer máquina funcionar como servidor de banco de dados MySQL é que a mesma tenha acesso a rede e ter um IP fixo. Além de realizar as seguintes configurações:

1. Altere o arquivo de configuração do MySQL. Para isso, execute o seguinte comando, como *root* para abrir o arquivo de configuração:

```
nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

2. Mude o IP da seguinte linha, para 0.0.0.0:

```
bind-address = 127.0.0.1
```

3. Ficando assim:

```
bind-address= 0.0.0.0
```

4. Reinicie o serviço do MySQL:

```
sudo /etc/init.d/mysql restart
```

5. Entre no MySQL como usuário *root*:

```
mysql -u root -p
```

### 1.5. Concessão de privilégios:

1. Conceda os devidos privilégios para o usuário, caso ele já exista:

```
GRANT <privilégios> ON <itens> TO '<usuario>'@<ip da máquina cliente>;
```

2. Caso o usuário não exista é necessário criá-lo e delegar os privilégios do item



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

anterior:

```
create user '<nome>' with password '<senha>';
```

Note que o usuário pode conceder seus privilégios a outros usuários com adição do WITH GRANT OPTION ao comando (**não é remendado**).

### Cliente

Para ter acesso ao MySQL é necessário que a máquina cliente possua o MySQL instalado. Para a instalação execute o seguinte comando:

```
sudo apt-get update  
sudo apt-get install mysql-client
```

Para o acesso remoto ao servidor execute:

```
mysqlq -u <usuário> -p 'senha' -h[IP do servidor]
```

Nota: Para ter acesso ao MySQL é necessário que a máquina cliente possua o MySQL instalado.

## Tipos de dados e seus domínios

### Dados Numéricos

O sistema gerenciador de banco de dados MySQL usa os tipos de dados numéricos apresentados na tabela abaixo. A definição dos campos como UNSIGNED (sem sinal) determina que serão aceitos apenas valores positivos. A definição dos campos como ZEROFILL (preenchimento com 0) preenche os campos vazios da coluna com 0, além da definição automática de UNSIGNED à coluna.



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

TIPO DE DADO	VALORES ACEITOS	USO DE MEMÓRIA (bytes)
BIT(N)	0,1	N+7/8
TINYINT(N)	-128 a 127	1
[UNSIGNED] [ZEROFILL]	0 a 255	
BOOL, BOOLEAN	0,1	1
SMALLINT(N)	-32768 a 32767	2
[UNSIGNED] [ZEROFILL]	0 a 65535.	
MEDIUMINT(N)	-8388608 a 8388607	3
[UNSIGNED] [ZEROFILL]	0 a 16777215	
INT(N),INTEGER(N)	-2147483648 a 2147483647	4
[UNSIGNED] [ZEROFILL]	0 a 4294967295	
BIGINT(N)	-9223372036854775807 a	8
[UNSIGNED] [ZEROFILL]	9223372036854775807	
DECIMAL(N,P)	N com máximo de 65 casas e P 30	Máx. 8 bytes
[UNSIGNED] [ZEROFILL]	*Sinal não incluso no número de casas.	
DEC[(M[,D])], NUMERIC[(M[,D])] ,	Seguem o mesmo padrão que os campos do tipo DECIMAL	Máx. 8 bytes



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

FIXED[(M[,D])] [UNSIGNED] [ZEROFILL]		
FLOAT(M,D)  [UNSIGNED] [ZEROFILL]	-3.402823466E+38 a  -1.175494351E-38  1.175494351E-38 a  3.402823466E+38	4
DOUBLE, DOUBLE PRECISION[(M,D)], REAL[(M,D)]  [UNSIGNED] [ZEROFILL]	-1,7976931348623157E+308 a  -2,2250738585072014E-308, 0  2,2250738585072014E-308 a  1,7976931348623157E + 308, 0	8
FLOAT(N) [UNSIGNED] [ZEROFILL]	Com valores de 0 a 24 é convertido para FLOAT(M,D), valores acima para DOUBLE.	4 se 0>=N<=24  8 se 25>=N<=53

### Armazenamento de data e tempo

Datas e tempos são representados pelos tipos de dados presentes na tabela abaixo. Os campos com **fsp** entre colchetes aceitam valores fracionários de





## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

segundos com no máximo 6 casas decimais.

TIPO DE DADO	VALORES ACEITOS	USO DE MEMÓRIA (bytes)
DATE	'1000-01-01' a '999-12-31'	3
DATETIME[(fsp)]	'1000-01-01 00:00:00.000000' a '9999-12-31 23:59:59.999999'.	5 + armazenamento da parte fracionária.
TIMESTAMP[(fsp)]	'1970-01-01 00:00:01.000000' UTC a '2038-01-19 03:14:07.999999'	4 + armazenamento da parte fracionária.
TIME[(fsp)]	'-838:59:59.000000' a '838:59:59.000000'	3
YEAR[(fsp)]	1901 a 2155 e 0000	1

### Armazenamento de caracteres e valores binários

O armazenamento de caracteres e sequência de bits no SGBD MySQL pode ser realizado com os tipos de dados apresentados na tabela abaixo.



## Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios

mysql>

TIPO DE DADO	QUANTIDADE MÁXIMA DE CARACTERES	USO DE MEMÓRIA (bytes)
CHAR(N),  BINARY(N)	255	N x W onde W representa o uso de memória da codificação de caracteres.
VARCHAR(N), VARBINARY(N)	65535 , Máx. 21844 por padrão.	Valor em bytes da String +2 (máximo)
TINYBLOB,TINYTEXT	Não especificado	257
BLOB[(N)]	Não especificado	65538
TEXT[(N)]	65535	65538
MEDIUMBLOB	Não especificado	16777219
MEDIUMTEXT	16777215	16777219
LOBLOB	Não especificado	4,294,967,300 (4 GB)
LONGTEXT	Não especificado	4,294,967,300 (4 GB)

Existe a possibilidade de armazenar dados do tipo caractere em formato de lista: ENUM ('Valor1','Valor2', ...)

Os atributos do tipo String armazenam somente um dos valores listados. É recomendado colocar menos de 3000 valores possíveis como parâmetros no



## **Arquitetura Cliente-Servidor e Tipos de Dados e seus Domínios**

mysql>

ENUM.

SET ('Valor1','Valor2',...)

Os atributos do tipo String declarados com SET armazenam um ou mais dos valores listados do SET.

### **Conclusão**

Durante a nossa pesquisa sobre os tipos de dados aceitos pelo MySQL e seus domínios, conhecemos alguns tipos ainda não explorados por nós em nenhum outro sistema gerenciador de banco de dados ou linguagem de programação. Os tipos DATETIME e TIMESTAMP além de desconhecidos foram agradáveis surpresas devido à possibilidade de armazenar frações de segundos, muito úteis em certos tipos de aplicações. Os tipos BINARY e VARBINARY também surpreenderam, já que são cadeias de caracteres binários e que se comportam como CHAR e VARCHAR, respectivamente. Outro tipo de dado utilizado para armazenar caracteres bastante interessante é o BLOB, que consegue armazenar cadeias de caracteres muito grandes.