


Segurança em SQL

- 
- ▶ Um usuário no mysql é uma combinação de nome de usuário e uma string do host.
 - ▶ A string do host pode ser um endereço IP, um hostname, DNS ou máscara de rede.
 - ▶ Ou seja, mesmo que dois usuários compartilhem o mesmo nome de usuário:
 - ▶ admin@192.168.2.10 é diferente de admin@'192.168.2.%'
 - ▶ os usuários podem ter diferente senhas e permissões de acesso.

- ▶ Ex: 2 usuários com o mesmo nome de usuário e diferentes senhas e permissões
- ▶ `mysql -u root -p senha`
- ▶ `GRANT USAGE ON *.* TO admin@'192.168.2.10' IDENTIFIED BY 'senha';`
- ▶ `GRANT ALL ON turma3i.* TO admin@'192.168.2.20' IDENTIFIED BY 'outrasenha';`

```
mysql> select user,host,password from mysql.user where user='admin';
```

user	host	password
admin	192.168.2.10	*2F9A309FBEA7337E61AA2953EB48179BF9300B7C
admin	192.168.2.20	*4CBC947A0D5CF017233C027F4597C92A92D02F92

```
2 rows in set (0.05 sec)
```

Access Control List

- ▶ Uma ACL (Access Control List) é uma lista de permissões que está associada com um objeto. Esta lista é a base do modelo de segurança no MySQL server.
- ▶ O MySQL mantém as ACLs (também chamadas grant tables) armazenadas em memória. Quando um usuário tenta se autenticar ou executar um comando, o MySQL checa as informações de autenticação e as permissões nas ACLs, em uma ordem pré-determinada.
- ▶ Se dois usuários tentarem acesso, `admin@'192.168.2.%'` e `admin@192.168.2.10`, o usuário `admin@'192.168.2.%'` vem depois de `admin@192.168.2.10` no teste de controle de acessos. Quando o MySQL checa a autenticação, o `admin@'192.168.2.%'` é o último usuário que as credenciais batem com as credenciais fornecidas.

- ▶ Lembrando o caso de dois usuários, com o mesmo nome de usuário mas com diferentes strings de host, podem ter diferentes senhas. Nesse exemplo, o computador usado pelo usuário tem o IP 192.168.2.20:

```
shell> mysql -u admin -senha -h 192.168.1.5
```

```
ERROR 1045 (28000): Access denied for user 'admin @'192.168.2.20'  
(using password: YES)
```

- ▶ Senhas
 - ▶ 'admin @'192.168.2.20' – outrasenha
 - ▶ 'admin @'192.168.2.10' – senha
- ▶ Os dois nomes de usuário podem ter a mesma senha, e possuírem permissões de acessos diferentes.
- ▶ Para saber o usuário atualmente autenticado
 - ▶ CURRENT_USER()

Coringas (Wildcard)

- ▶ Representados por
 - ▶ % - qualquer coisa
 - ▶ Ex: '192.168.2.%' – qualquer IP que comece por 192.168.2.
 - ▶ _ um caractere
 - ▶ Ex: '192.168.2.1_0' – Qualquer IP que possua 192.168.2. e termine com múltiplo de 10, entre 100 e 190.
- ▶ Quando o MySQL verifica a ACL, organiza de forma que as strings mais genéricas sejam testadas por último, ou seja, endereços IPs sem coringas são testados primeiro que os endereços com coringas.

- ▶ Por exemplo, admin@192.168.2.10 possui permissão total de leitura/escrita no banco de dados turma3i, e admin@'19.168.2.%' possui permissão somente de leitura.

```
mysql> GRANT SELECT ON turma3i.* TO admin@'192.168.2.%' identified  
by 'senha';
```

```
Query OK, 0 rows affected (0.39 sec)
```

```
mysql> GRANT ALL ON turma3i.* TO admin@'192.168.2.10' identified by  
'senha';
```

```
Query OK, 0 rows affected (0.00 sec)
```




```
mysql> SHOW GRANTS\G
```

```
***** 1. row *****
```

```
Grants for admin@192.168.2.10: GRANT USAGE ON *.* TO 'admin'@'192.168.2.10' IDENTIFIED BY PASSWORD '*2C6396ADEEF1AF865672D48735C0E3EC8B1A9CEC'
```

```
***** 2. row *****
```

```
Grants for admin@192.168.2.10: GRANT ALL PRIVILEGES ON `turma3i`.* TO 'admin'@'192.168.2.10'
```


```
2 rows in set (0.00 sec)
```


- Todas as informações de usuário e permissões ficam armazenadas no banco de dados mysql em tabelas conhecidas como grant tables. Se executar o comando SHOW DATABASES, aparecerá o seguinte:

```
mysql> SHOW DATABASES;
```

```
+-----+  
| Database           |  
+-----+  
| information_schema |  
| mysql              |  
| test               |  
+-----+
```

```
3 rows in set (0.02 sec)
```


- 
- ▶ O BD information_schema na verdade não é um BD, mas uma interface para vários dados do Sistema.
 - ▶ O BD test é um BD vazio usado para realizar testes.
 - ▶ O BD mysql armazena as informações dos usuários. Além das grant tables, o mysql possui tabelas contendo outras informações de sistema, como, por exemplo, a tabela event, com as informações do agendador de eventos.
 - ▶ As tabelas que são de interesse de controle de usuários são as tabelas columns_priv, db, host, procs_priv, tables_priv e user. É possível manipular diretamente as tabelas usando comandos SQL, como select, update, insert, delete, etc.

- Um dos maiores problemas em BD ocorre quando o usuário possui acesso irrestrito ao BD mysql, pois permite, por exemplo, manipular os dados dos usuários cadastrados.

```
mysql> SELECT user, host, password FROM mysql.user;
```

+-----+-----+-----+		
user	host	password
+-----+-----+-----+		
root	%	*ACC4836009D0D7911EFE143E154D3E7C32AB8EEB
root	localhost	*ACC4836009D0D7911EFE143E154D3E7C32AB8EEB
developer	localhost	*50C0E8BEE396F2367258EC80901409C4BE300238
production_slave	slave.company.com	*891A44E50A5E8286F04BC1EFB0292BE3AFE74D5E
production_slave	192.168.2.191	*891A44E50A5E8286F04BC1EFB0292BE3AFE74D5E
ops	localhost	*99FFA08BDD2C5D80552F52F441AA632DFA1DE9E3
cto	192.%	*B81134DE91B9BE86259180DC8446A254008A1D9E
+-----+-----+-----+		

```
7 rows in set (0.00 sec)
```

- 
- ▶ Para criar um usuário:
 - ▶ CREATE USER.
 - ▶ Para renomear um usuário
 - ▶ RENAME USER
 - ▶ Para apagar um usuário:
 - ▶ DROP USER.



► Ex: Criar o usuário ops, dar privilégios, renomear para over_lords e deletar.

```
mysql> CREATE USER 'ops'@'192.168.%' IDENTIFIED BY 'password';
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> GRANT ALL PRIVILEGES ON test.* TO 'ops'@'192.168.%;
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> RENAME USER 'ops'@'192.168.%' TO 'over_lords'@'192.168.%;
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> DROP USER 'over_lords'@'192.168.%;
```

Query OK, 0 rows affected (0.00 sec)

- ▶ Dar privilégios
 - ▶ GRANT privilégio ON banco.tabela TO usuário
- ▶ Remover privilégios
 - ▶ REVOKE privilégio ON banco.tabela FROM usuário
- ▶ Permissão global – Fica armazenado na tabela mysql.user
 - ▶ GRANT | REVOKE privilégio ON *.* TO | FROM usuário
 - ▶ Ex: GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'ops'@'192.168.%';
 - ▶ Ex: GRANT ALL ON *.* TO 'ops'@'192.168.%';
- ▶ Permissão em DB específico – Fica armazenado nas tabelas mysql.db e mysql.host
 - ▶ GRANT | REVOKE privilégio ON banco.* TO | FROM usuário
 - ▶ Ex: GRANT ALL ON turma3i.* TO 'ops'@'192.168.%';

- ▶ Permissão em tabela específica – Fica armazenado em `mysql.tables_priv`
 - ▶ `GRANT | REVOKE privilegio ON db_name.table_name TO | FROM usuário;`
 - ▶ Ex: `GRANT SELECT, INSERT, UPDATE, DELETE ON turma3i.pessoa TO 'ops'@'192.168.%';`
- ▶ Permissão em coluna específica – Fica armazenado em `mysql.columns_priv`
 - ▶ `GRANT | REVOKE privilegio (coluna1, coluna2,...) ON db_name.table_name TO | FROM usuário;`
 - ▶ Ex: `GRANT SELECT (col1,col2) ON turma3i.pessoa TO 'ops'@'192.168.%';`
- ▶ Permissão em rotinas armazenadas – Fica armazenado em `mysql.procs_priv`. Pode ser definida em nível global ou de BD.
 - ▶ `GRANT | REVOKE privilegio ROUTINE ON db_name.* TO | FROM usuário;`
 - ▶ `GRANT | REVOKE privilegio ON PROCEDURE | FUNCTION db_name.(proc | func)_name TO | FROM usuário;`
 - ▶ Ex: `GRANT CREATE ROUTINE ON turma3i.* TO 'someuser'@'somehost';`
 - ▶ Ex: `GRANT EXECUTE ON PROCEDURE turma3i.ver_saldo TO 'someuser'@'somehost';`