

Banco de Dados – C07

Felipe Tagawa Reis

Comandos DCL

Linguagem SQL

SQL COMMANDS

1

DATA DEFINITION LANGUAGE (DDL)

- CREATE
- DROP
- ALTER
- TRUNCATE

2

DATA MANIPULATION LANGUAGE (DML)

- INSERT
- UPDATE
- DELETE

3

DATA CONTROL LANGUAGE (DCL)

- GRANT
- REVOKE

4

TRANSACTION CONTROL LANGUAGE (TCL)

- COMMIT
- ROLLBACK
- SAVEPOINT

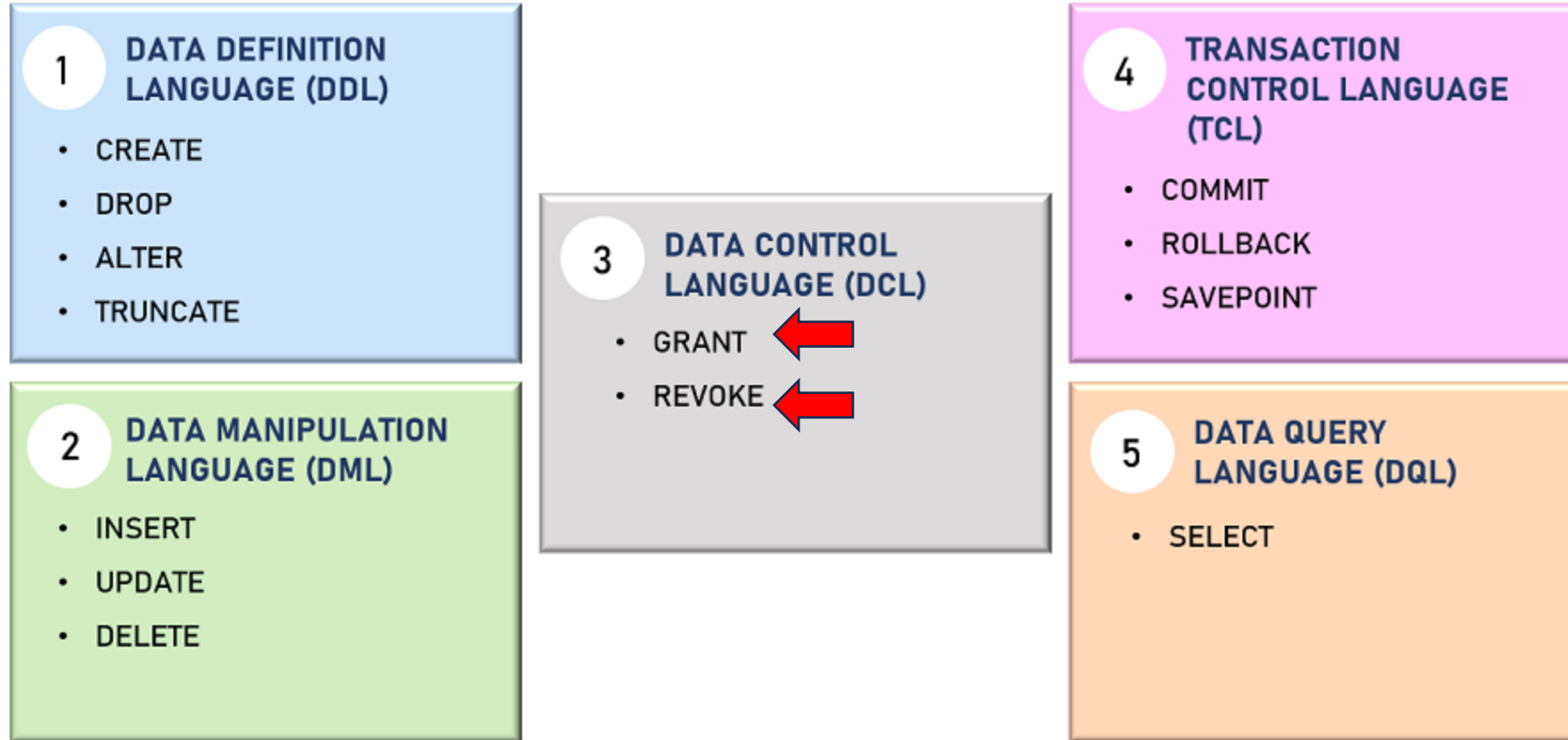
5

DATA QUERY LANGUAGE (DQL)

- SELECT

Aula de Hoje

SQL COMMANDS



Comandos DCL

DCL (Data Control Language) é o conjunto de comandos responsáveis por garantir o controle de acesso ao banco de dados

Comandos:

➤ GRANT

➤ REVOKE

Comando GRANT

Concede certos privilégios a um usuário especificado, é necessária a existência desse usuário previamente para o comando funcionar corretamente. É **necessário** que informemos qual o endereço IP poderá acessar ('@%', '@localhost' ou '@EndereçoIP' após o nome).

Privilégios:

ALL PRIVILEGES – todos os privilégios, menos o GRANT;

CREATE, ALTER e DROP – DDL

SELECT, INSERT, UPDATE E DELETE – DQL e DML

```
1  -- Considerando que o database se chame 'BD'
2  -- Criar o usuário Professor
3  CREATE USER 'Professor'@'%' IDENTIFIED BY '1234';
4  -- Criar o usuário Monitor
5  CREATE USER 'Monitor'@'localhost' IDENTIFIED BY '5678';
6  -- Apagar o Monitor
7  DROP USER 'Monitor';
8
9  -- Garantir privilégios específicos apenas para a tabela alunos ao Professor
10 GRANT SELECT, INSERT, UPDATE, DELETE ON BD.alunos TO 'Professor'@'%';
11
12 -- Garantir todos os privilégios de todas as tabelas para o Professor
13 GRANT ALL PRIVILEGES ON BD.* TO 'Professor'; -- Omitir o endereço (Considera %)
```

Hierarquia de Privilégios

Os privilégios podem ser aplicados em diferentes níveis.

Nível Global: GRANT ALL ON *.* (Todo o servidor);

Nível Database: GRANT ALL ON bd_exemplo.*;

Nível de Tabela: GRANT SELECT ON bd_exemplo.Musica.

FLUSH PRIVILEGES

shutterstock.com · 2323114893

- Atualiza a tabela de acessos imediatamente na memória, garantindo que as alterações entrem em vigor na hora.

Comando REVOKE

Revoga/remove certos privilégios de um usuário especificado, é claro que é necessária a existência desse usuário previamente para o comando funcionar corretamente.

```
-- Remover o comando DELETE do Monitor.
```

```
REVOKE DELETE ON BD.alunos FROM 'Monitor';
```

```
-- Remover todos os privilégios de todas as tabelas para o professor.
```

```
REVOKE ALL PRIVILEGES ON BD.* FROM 'Professor';
```

Visualizando

```
SELECT * FROM mysql.user;
```




```
-- Para visualizar todos os usuários do MySQL
```

```
SHOW GRANTS FOR 'Monitor';
```

```
SHOW GRANTS FOR 'Professor';
```

```
-- Para visualizar as permissões de um usuário específico
```


MANAGEMENT

-  Server Status
-  Client Connections
-  **Users and Privileges**
-  Status and System Variables
-  Data Export
-  Data Import/Restore



Local instance MySQL80

Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

Details for account root@localhost

Login Account Limits Administrative Roles **Schema Privileges**

Login Name: root

You may create multiple accounts with the same name to connect from different hosts.

Authentication Type: caching_sha2_password

For the standard password and/or host based authentication, select 'Standard'.

Limit to Hosts Matching: localhost

% and _ wildcards may be used

Password: *****

Type a password to reset it.

Consider using a password with 8 or more characters with mixed case letters, numbers and punctuation marks.

Confirm Password: *****

Enter password again to confirm.

Expire Password

Authentication String: \$A\$005\$6□UNWG□!P_D~[□HtF +

Authentication plugin specific parameters.

See the plugin documentation for valid values and details.

Details for account root@localhost

[Login](#)[Account Limits](#)[Administrative Roles](#)[Schema Privileges](#)

Schema	Privileges
%	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

[Revoke All Privileges](#)[Delete Entry](#)[Add Entry...](#)

The user 'root'@'localhost' will have the following access rights to any schema:

Object Rights

- ☒ SELECT
- ☒ INSERT
- ☒ UPDATE
- ☒ DELETE
- ☒ EXECUTE
- ☒ SHOW VIEW

DDL Rights

- ☒ CREATE
- ☒ ALTER
- ☒ REFERENCES
- ☒ INDEX
- ☒ CREATE VIEW
- ☒ CREATE ROUTINE
- ☒ ALTER ROUTINE
- ☒ EVENT
- ☒ DROP
- ☒ TRIGGER

Other Rights

- ☐ GRANT OPTION
- ☒ CREATE TEMPORARY TABLES
- ☒ LOCK TABLES

[Unselect All](#)[Select "ALL"](#)

Exercício

```
DROP DATABASE IF EXISTS BD;
CREATE DATABASE BD;
USE BD;

CREATE TABLE Musica(
    id INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(30) NOT NULL,
    dataLancamento DATE,
    genero VARCHAR(20),
    primary key(id)
);
```

Para o esquema ao lado, faça o que se pede:

- Crie dois usuários com suas senhas;
- Para o primeiro, conceda INSERT, DELETE;
- Para o segundo, conceda todos os privilégios em todas as tabelas;
- Retire todos os privilégios do primeiro;
- Retire os privilégios UPDATE e DELETE do Segundo em todas as tabelas.

AULA
CONCLUÍDA!