



BACK-END

Desenvolvimento de Software para Internet



LINGUAGENS DO SQL

Diferentes formas de manipular o banco de dados



```
c:\wamp64\bin\mysql\mysql5.7.36\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.36 MySQL Community Server (GPL)

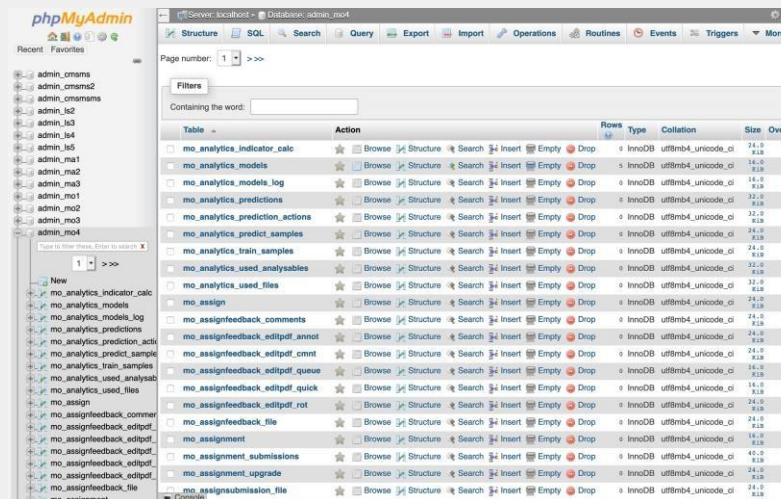
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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

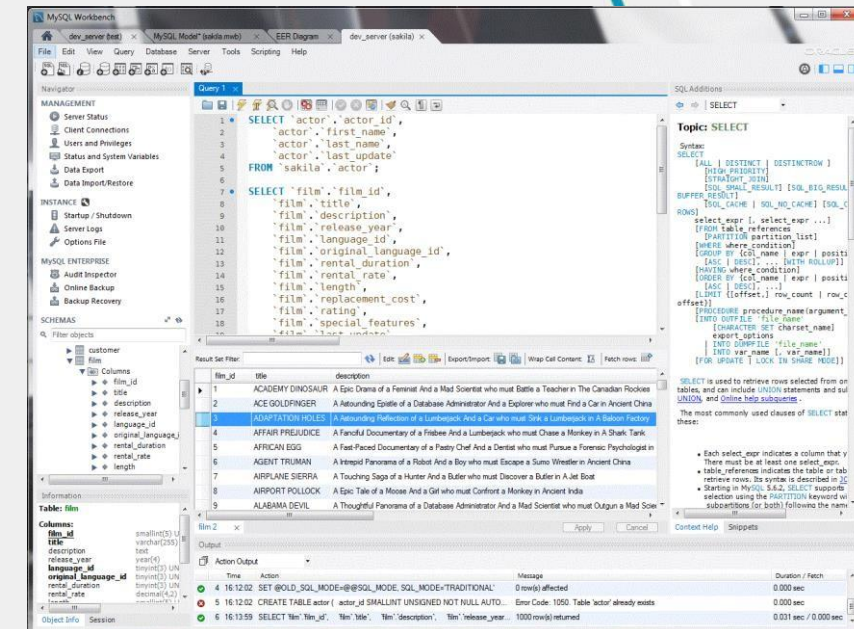
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

MySQL “na unha”

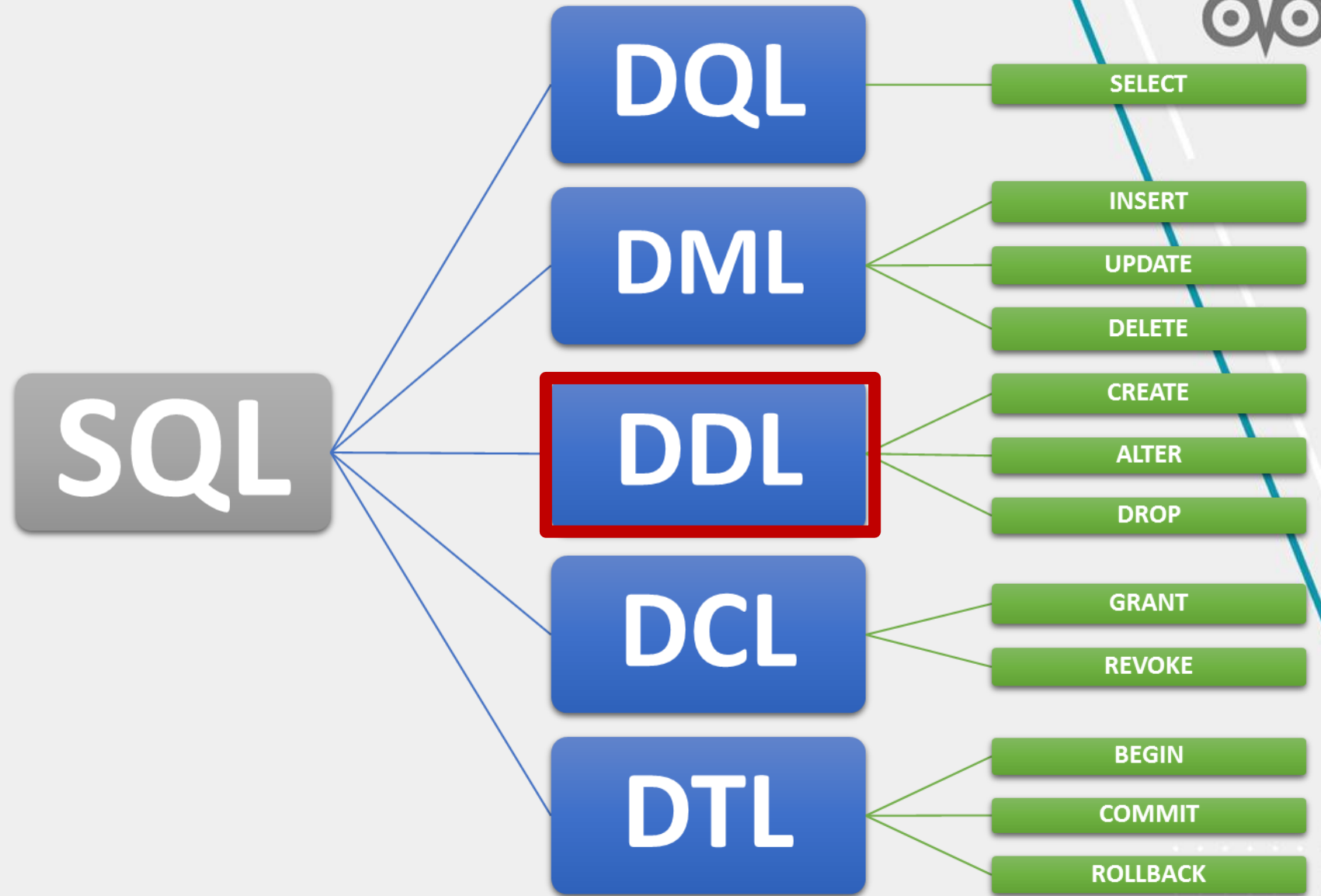


PHP MyAdmin



MySQL Workbench

Structured Query Language



Crédito: [DevMedia](#)





DDL – Data Definition Language

Linguagem de Definição de Dados


```
-- -----  
-- Table `projeto`.`pessoa`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`pessoa` (  
  `idpessoa` INT NOT NULL,  
  `nome` VARCHAR(50) NULL,  
  `cpf` VARCHAR(45) NOT NULL,  
  `num_armario` INT NULL,  
  PRIMARY KEY (`idpessoa`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `projeto`.`avaliacao`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `projeto`.`avaliacao` (  
  `idavaliacao` INT NOT NULL,  
  `texto` LONGTEXT NULL,
```



SHOW DATABASES

```
SHOW { DATABASES | SCHEMAS }  
  
[LIKE 'pattern' | WHERE expr]
```

Objetivo: Listar os bancos de dados presentes no servidor
MySQL

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/show-databases.html>



CREATE DATABASE | USE DATABASE

```
CREATE DATABASE nome_do_banco_de_dados;
```

```
USE nome_do_banco_de_dados
```

```
Database changed
```

Objetivo: Criar e selecionar um banco de dados

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/creating-database.html>



SHOW TABLES

```
SHOW TABLES;
```

Objetivo: Exibir todas as tabelas associadas ao banco de dados que está selecionado (USE deve ter sido acionado anteriormente)

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/show-tables.html>



CREATE TABLE

```
CREATE TABLE nome_tabela (  
    campo1 TIPO_DE_DADO REST,  
    campo2 TIPO_DE_DADO REST,  
    campo3 TIPO_DE_DADO REST  
);
```

Objetivo: Criar tabela (e campos)

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/creating-tables.html>

TIPO DE DADO	DESCRIÇÃO
INTEGER	Números inteiros
FLOAT	Números reais
CHAR (tamanho)	String de comprimento fixo (0 a 255)
VARCHAR(tamanho)	String de tamanho variável: 0 a 65.535
BINARY(size)	Igual a CHAR(), mas armazena binário. Padrão: 1
TEXT(size)	String com comprimento máximo de 65.535
LONGTEXT	String com comprimento máx. 4,294,967,295
DATETIME	Padrão: 'YYYY-MM-DD hh:mm:ss'

[Veja outros tipos de dados](#)





ALTER TABLE

```
ALTER TABLE nome_da_tabela
```

```
ADD nome_da_coluna TIPO_DE_DADO;
```

Objetivo: Alterar estrutura de uma tabela

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/alter-table.html>



DROP TABLE

```
DROP TABLE nome_da_tabela;
```

Objetivo: Remover uma tabela

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/drop-table.html>





DROP DATABASE

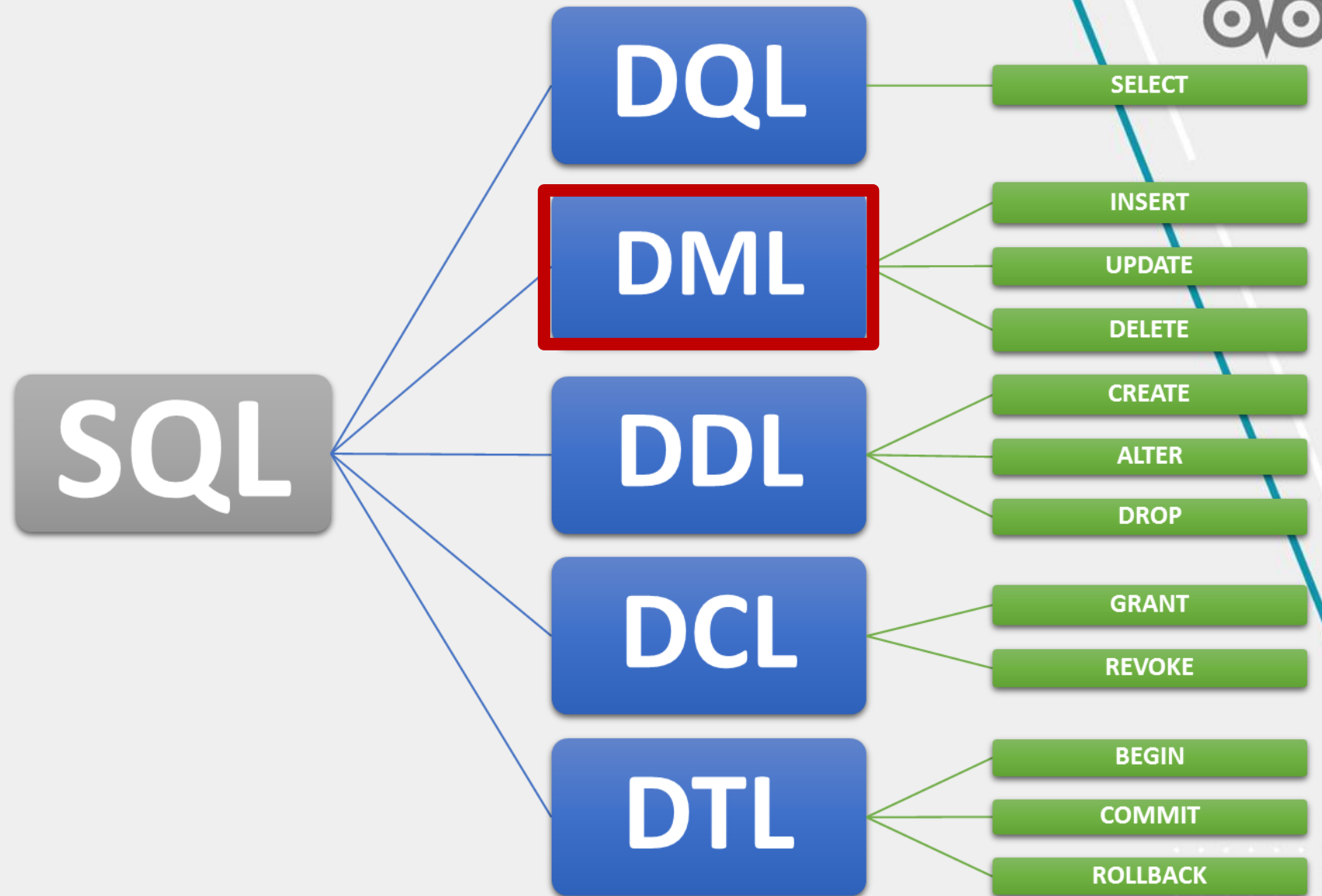
```
DROP DATABASE nome_do_banco_de_dados;
```

Objetivo: Remover um banco de dados

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/drop-database.html>



Structured Query Language



Crédito: [DevMedia](#)





DML – Data Manipulation Language

Linguagem de Manipulação de Dados





INSERT INTO

```
INSERT INTO nome_da_tabela (coluna1, coluna2 ...)  
VALUES (valor1, valor2 ...)
```

Objetivo: Inserir novos registros na tabela

OBS: Caso vá inserir valores em todas as colunas, não é necessário indicar o nome da coluna! Mas precisa inserir dados na ordem em que as colunas estão definidas no SGBD

Doc: <https://dev.mysql.com/doc/refman/8.0/en/insert.html>



UPDATE

```
UPDATE nome_da_tabela  
SET coluna1=valor1, coluna2=valor2,  
WHERE condição
```

Objetivo: Modificar um registro existente na tabela

OBS: Estabelecer a condição de alteração é essencial. Caso contrário, modifica todos os valores da coluna.

Doc: https://www.w3schools.com/mysql/mysql_update.asp



DELETE FROM

`DELETE FROM nome_da_tabela`

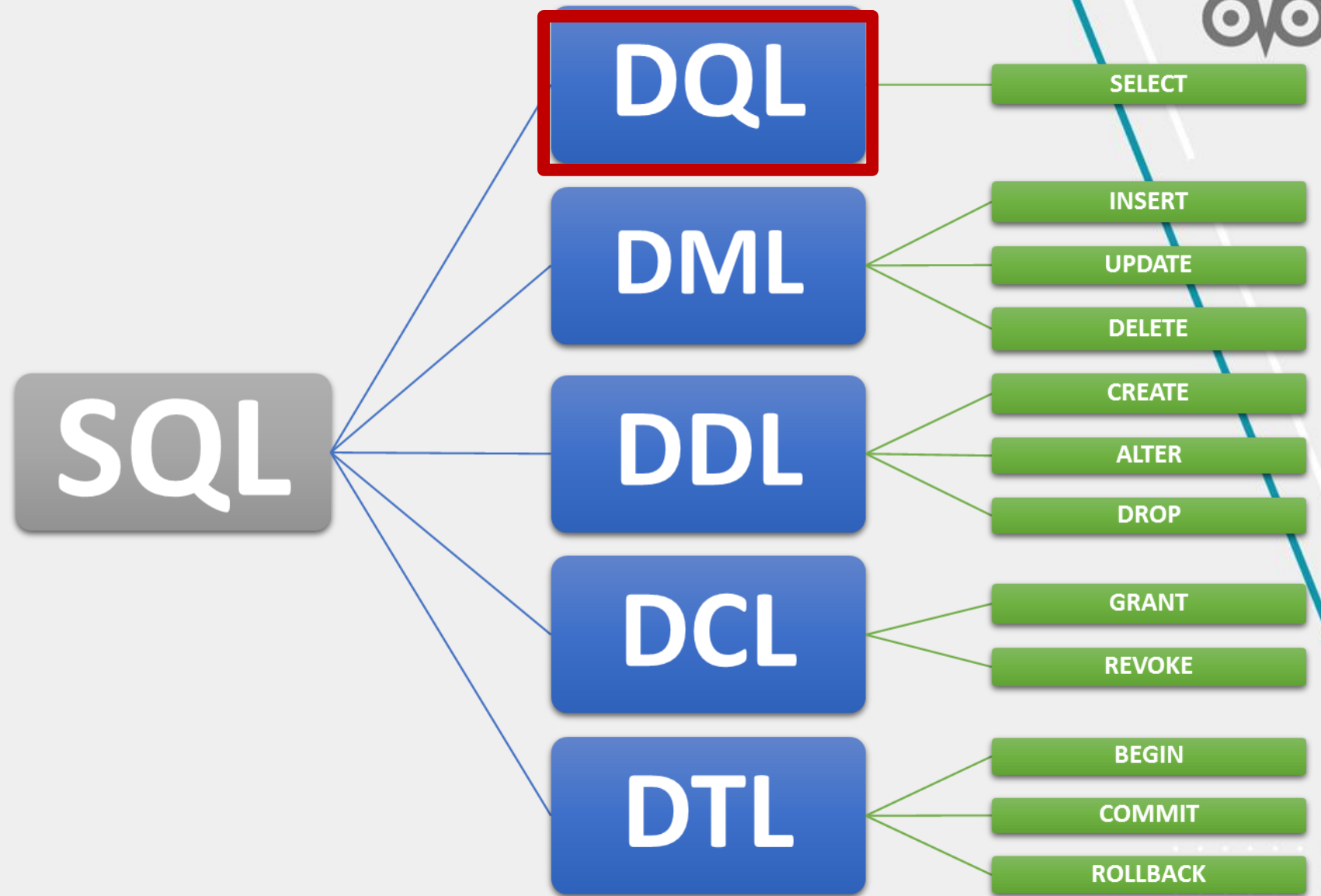
`WHERE` condição

Objetivo: Remover um registro da tabela

OBS: Aqui a condição também é essencial!

Doc: <https://dev.mysql.com/doc/refman/8.0/en/delete.html>

Structured Query Language



Crédito: [DevMedia](#)





DQL – Data Query Language

Linguagem de Consulta de Dados





SELECT

```
SELECT coluna1, coluna2, coluna3, ...
```

```
FROM nome_da_tabela
```

Objetivo: Selecionar/Consultar dados do banco de dados

Doc: <https://dev.mysql.com/doc/refman/8.0/en/select.html>





SELECT ... WHERE

```
SELECT coluna1, coluna2, coluna3, ...
```

```
FROM nome_da_tabela
```

```
WHERE condição
```

Objetivo: Selecionar/Consultar dados do banco de dados, atendendo a determinada condição

Doc: <https://dev.mysql.com/doc/refman/8.0/en/select.html>



SELECT ... WHERE ... AND, OR, NOT

```
SELECT coluna1, coluna2, coluna3, ...
```

```
FROM nome_da_tabela
```

```
WHERE condição AND/OR/NOT condição
```

Objetivo: Selecionar/Consultar dados do banco de dados, atendendo a uma ou mais condições

Doc: <https://dev.mysql.com/doc/refman/8.0/en/select.html>





SELECT ... WHERE ... ORDER BY

```
SELECT coluna1, coluna2, coluna3, ...  
FROM nome_da_tabela  
WHERE condição AND|OR|NOT condição  
ORDER BY coluna1, coluna2, ... ASC|DESC
```

Objetivo: Selecionar/Consultar dados do banco de dados, atendendo a uma ou mais condições e apresentando resultados ordenado por determinada coluna (de forma crescente ou decrescente)

Doc: <https://dev.mysql.com/doc/refman/8.0/en/select.html>





E como consultar dados de tabelas relacionadas?



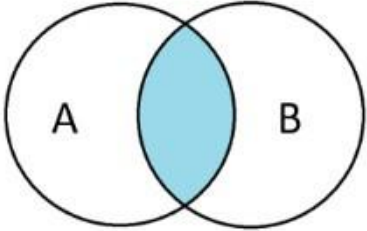
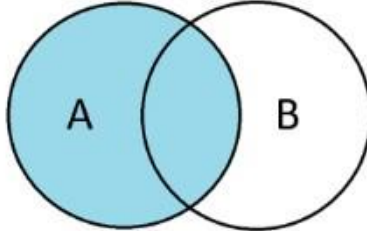
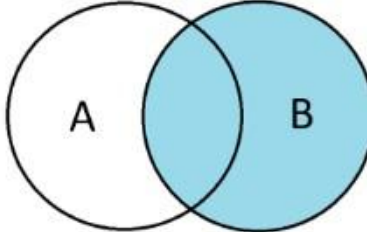
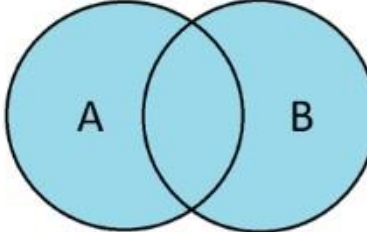


JOIN

Objetivo: combinar linhas de duas ou mais tabelas, com base em uma coluna relacionada entre elas.

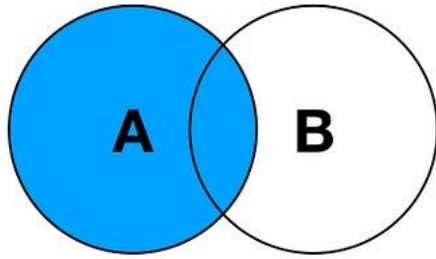




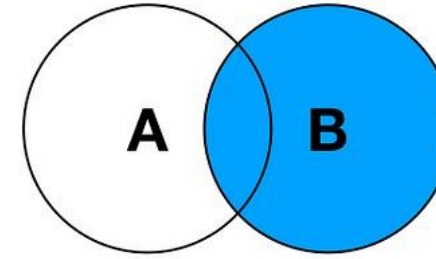
TIPO	DESCRIÇÃO	DIAGRAMA
INNER JOIN	Registros comuns entre as tabelas A e B	
LEFT JOIN	Todos os registros da tabela A (esquerda) e registros correspondentes na tabela B (direita)	
RIGHT JOIN	Todos os registros da tabela B (direita) e registros correspondentes na tabela A (esquerda)	
CROSS JOIN	Todos os registros de ambas as tabelas A e B (pode-se usar JOIN também)	



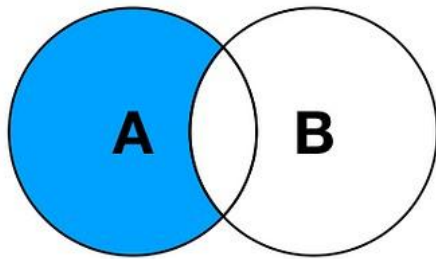
SQL JOINS



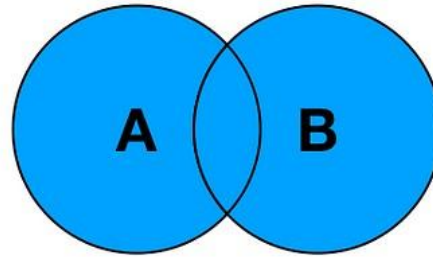
LEFT JOIN



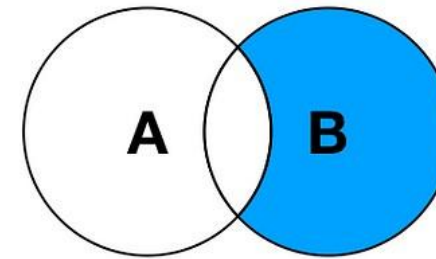
RIGHT JOIN



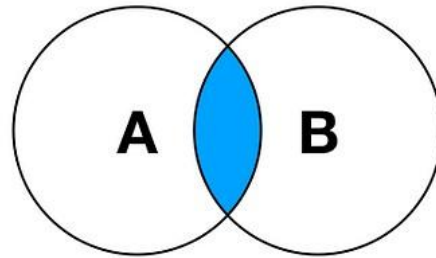
**LEFT JOIN EXCLUDING
INNER JOIN**



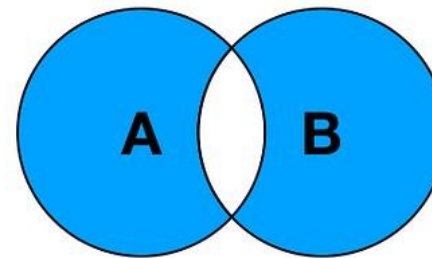
FULL OUTER JOIN



**RIGHT JOIN EXCLUDING
INNER JOIN**



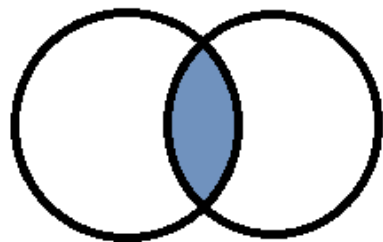
INNER JOIN



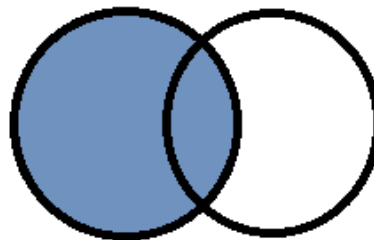
**FULL OUTER JOIN EXCLUDING
INNER JOIN**



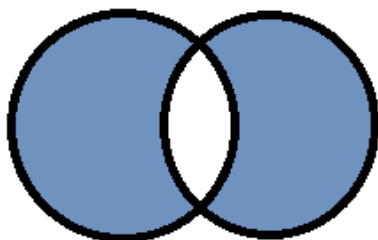
JOIN



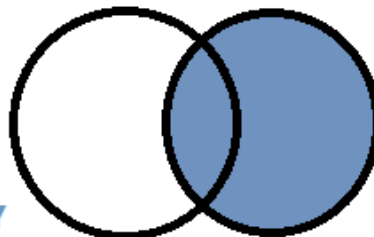
SELECT *
FROM T_1 A
INNER JOIN T_2 B
ON A.KEY = B.KEY;



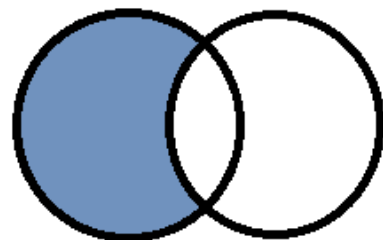
SELECT *
FROM T_1 A
LEFT JOIN T_2 B
ON A.KEY = B.KEY;



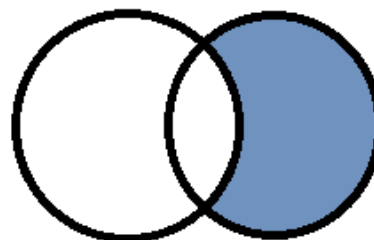
SELECT * FROM T_1 A
LEFT JOIN T_2 B
UNION
SELECT * FROM T_1 A
LEFT JOIN T_2 B
WHERE A.KEY IS NULL OR B.KEY
IS NULL OR A.KEY IS NULL;



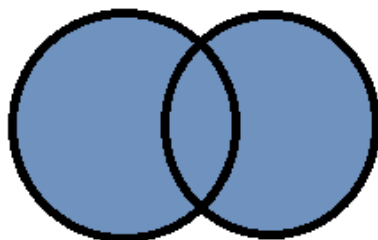
SELECT *
FROM T_1 A
RIGHT JOIN T_2 B
ON A.KEY = B.KEY;



SELECT *
FROM T_1 A
LEFT JOIN T_2 B
ON A.KEY = B.KEY
WHERE B.KEY IS NULL;



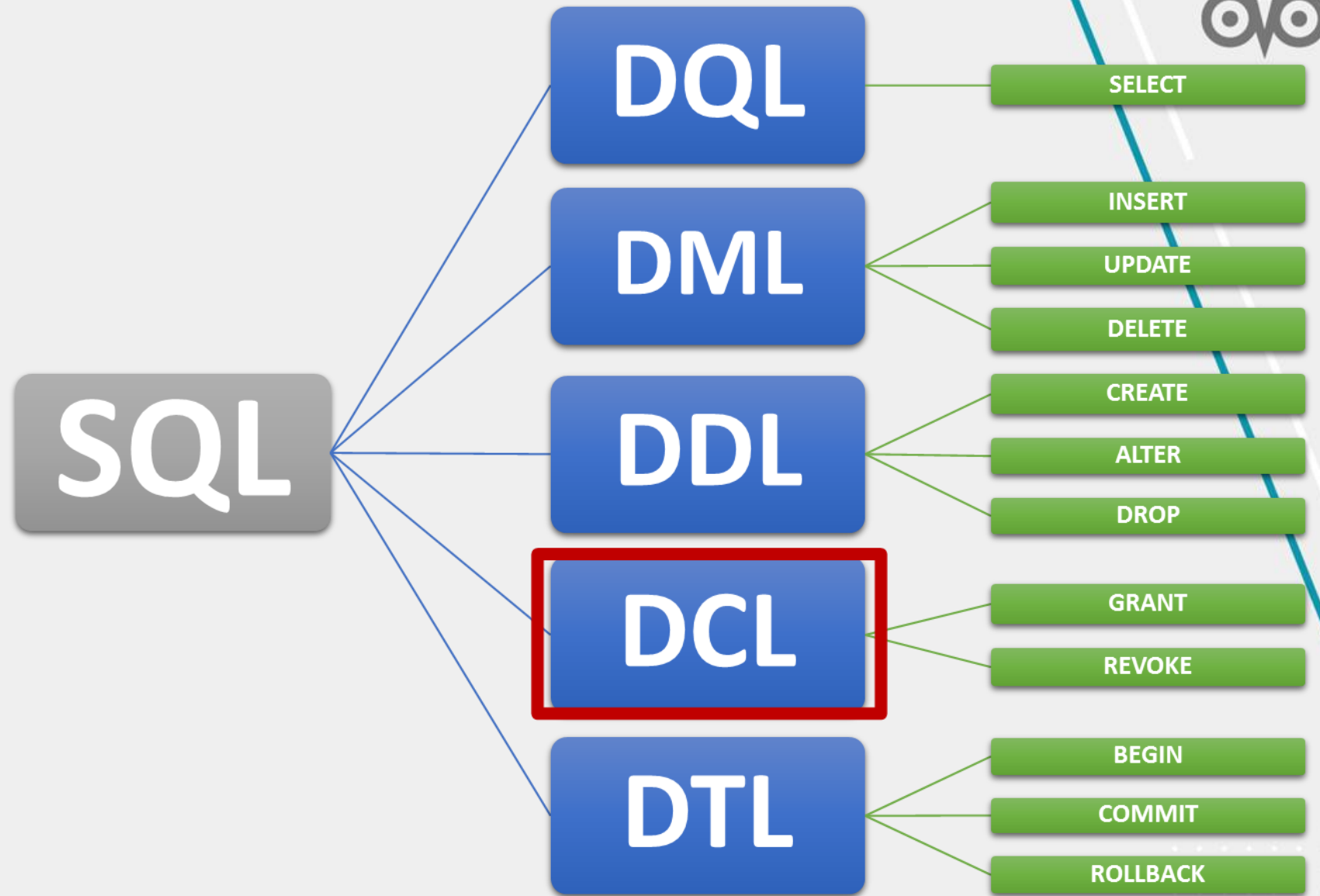
SELECT *
FROM T_1 A
RIGHT JOIN T_2 B
ON A.KEY = B.KEY
WHERE A.KEY IS NULL;



SELECT * FROM T_1 A LEFT JOIN T_2 B
UNION
SELECT * FROM T_1 A RIGHT JOIN T_2 B;



Structured Query Language



Crédito: [DevMedia](#)



DCL: Data Control Language

Linguagem de Controle de Dados



GRANT

```
GRANT ALL PRIVILEGES ON banco_de_dados.coluna.*  
TO 'username'@'server'
```

ou `GRANT SELECT, INSERT, UPDATE, DELETE` (*privilégios específicos*) ...

Objetivo: Permitir o acesso de um banco de dados para um usuário (ideal por questões de segurança)

Doc: <https://dev.mysql.com/doc/refman/8.0/en/grant.html>





GRANT ... ANTES PORÉM ... CREATE USER

```
CREATE USER 'nome'@'localhost' IDENTIFIED BY 'senha';
```

É necessário criar o usuário que terá acesso ao banco de dados!

Doc: <https://dev.mysql.com/doc/refman/8.0/en/create-user.html>





REVOKE

```
REVOKE lista_de_privilégios ON objeto FROM  
usuário1,usuário2 ...
```

Objetivo: Remover permissões de acesso!

Doc: <https://dev.mysql.com/doc/refman/8.0/en/revoke.html>





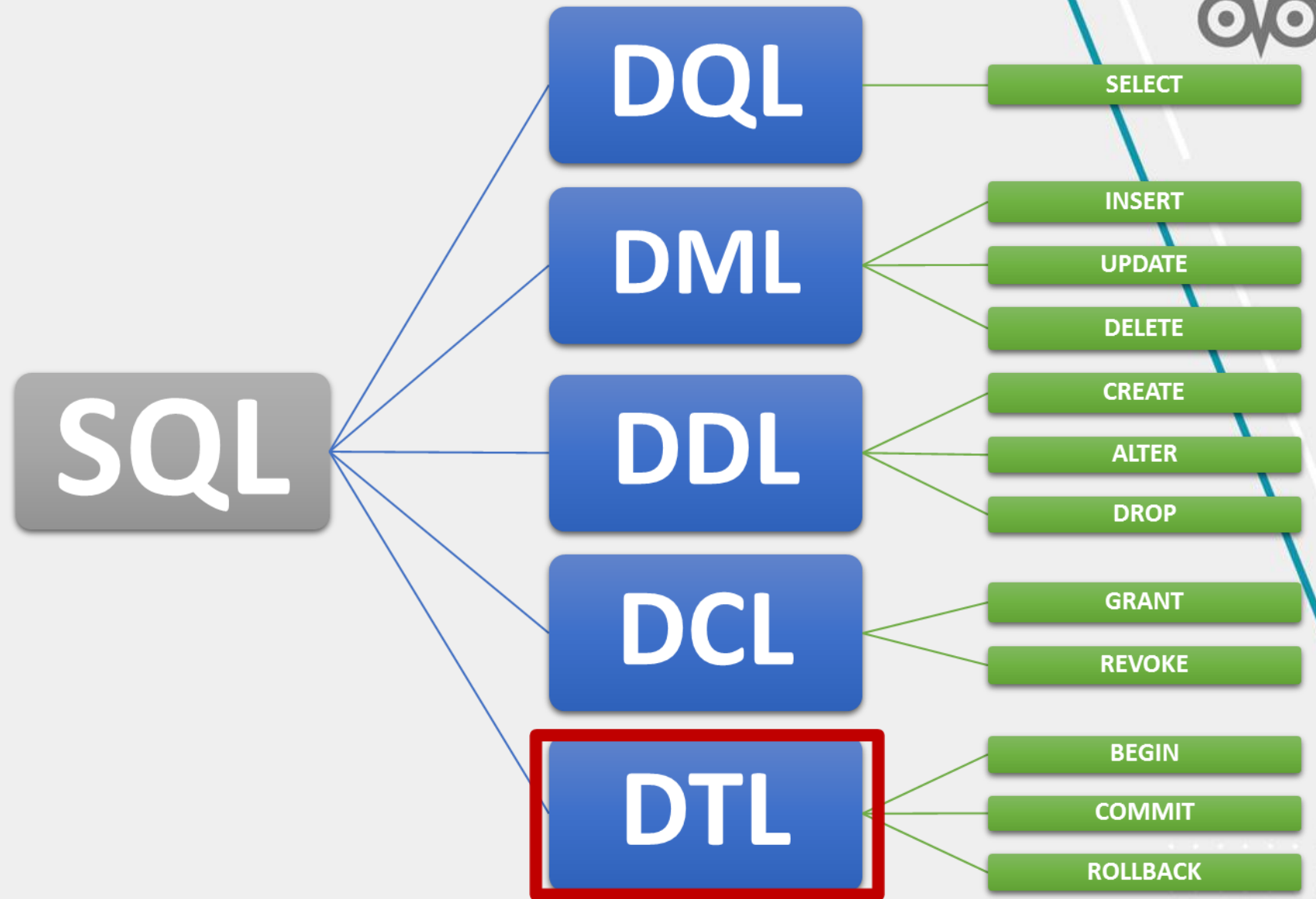
Mais sobre GRANT e REVOKE

- Curso de MySQL – GRANT e REVOKE – Definindo privilégios de acesso aos bancos de dados (Boson Treinamentos):

<http://www.bosontreinamentos.com.br/mysql/curso-de-mysql-grant-e-revoke-definindo-privilegios-de-acesso-aos-bancos-de-dados/>



Structured Query Language



Crédito: [DevMedia](#)



DTL: Data Transaction Language

Linguagem de Transação de Dados





TRANSAÇÕES EM SQL: PROPÓSITO

- Manter o banco de dados consistente em caso de falha do sistema ou de operações
- Fornecer isolamento entre os programas que acessam o banco de forma concorrente.



Crédito: [Usuário do Reddit](#)



COMMIT e ROLLBACK

COMMIT

Quando uma transação é efetuada com sucesso, o banco de dados é alterado de forma permanente, com os dados envolvidos na transação persistidos

Ativado por padrão na engine InnoDB();

ROLLBACK

Caso haja falha em qualquer uma das operações da transação, o banco de dados deverá ser retornado ao estado em que se encontrava anteriormente à execução da transação




```
CREATE TABLE cliente (  
    idcliente INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(255) NOT NULL,  
    datanasc DATE  
) Engine=InnoDB;
```

```
CREATE TABLE telefone (  
    idtelefone INT AUTO_INCREMENT PRIMARY KEY,  
    numero VARCHAR(255) NOT NULL,  
    tipo VARCHAR(255),  
    cliente_idcliente INT NOT NULL,  
    CONSTRAINT fk_telefone_cliente  
    FOREIGN KEY (cliente_idcliente)  
    REFERENCES cliente (idcliente)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION  
) Engine=InnoDB;
```



Como verificar se o commit está ativado por padrão?



1: **ATIVADO**

0: **DESATIVADO**

```
mysql> SELECT @@autocommit;
+-----+
| @@autocommit |
+-----+
|          1   |
+-----+
1 row in set (0.00 sec)
```





Desabilitar e habilitar commit

```
/*DESATIVAR O COMMIT*/  
SET autocommit = 0;  
/*ou*/  
SET autocommit = OFF  
  
/*ATIVAR NOVAMENTE */  
SET autocommit = 1;  
/*ou*/  
SET autocommit = ON;
```

Por que isso é relevante?

Para conseguirmos aplicar ROLLBACK!





Lembra daquela situação?

UPDATE cliente **SET** nome = 'Carolina';

Se o **COMMIT** estiver desativado, o **ROLLBACK** funciona!

Então ... Se for fazer alguma transação importante: desative
o **COMMIT**





Upload de arquivos no PHP





Upload de arquivos no PHP [1]

1. Criar um formulário compatível com envio de arquivos

```
<form action="cadastrar.php" enctype="multipart/form-data" method="post">  
  ...  
  <input id="foto" name="foto" type="file">
```





Upload de arquivos no PHP [2]

1. Criar um formulário compatível com envio de arquivos
2. Receber os dados do arquivo na variável **`$_FILES`**

```
array (size=1)
  'foto' =>
    array (size=5)
      'name' => string 'carinha.jpg' (length=11)
      'type' => string 'image/jpeg' (length=10)
      'tmp_name' => string 'C:\wamp64\tmp\php857D.tmp' (length=25)
      'error' => int 0
      'size' => int 496060
```





Sobre a \$_FILES[]

Parâmetro da \$_FILES	O que faz?
<code>\$_FILES ['arquivo'] ['name']</code>	Contém o nome original do arquivo, dado pelo usuário
<code>\$_FILES ['arquivo'] ['type']</code>	Tipo MIME (multipurpose internet mail extensions), caso o navegador informe
<code>\$_FILES ['arquivo'] ['size']</code>	Tamanho do arquivo em bytes
<code>\$_FILES ['arquivo'] ['tmp_name']</code>	Endereço e nome do arquivo temporários, usados para armazenamento no servidor
<code>\$_FILES ['arquivo'] ['error']</code>	Código de erro associado, se houver





Códigos de erro da \$_FILE[]

Código de erro	Valor	Significado
UPLOAD_ERR_OK	0	Sem erros
UPLOAD_ERR_INI_SIZE	1	Arquivo enviado excede limite definido no servidor (diretiva UPLOAD_MAX_FILESIZE no php.ini)
UPLOAD_ERR_FORM_SIZE	2	Arquivo enviado excede limite definido no formulário HTML (MAX_FILE_SIZE)
UPLOAD_ERR_PARTIAL	3	Upload executado parcialmente
UPLOAD_ERR_NO_FILE	4	Não foi enviado arquivo no upload
UPLOAD_ERR_NO_TMP_DIR	5	Pasta temporária de upload não existe
UPLOAD_ERR_CANT_WRITE	7	Falha ao escrever o arquivo no servidor
UPLOAD_ERR_EXTENSION	8	Uma extensão qualquer do PHP interrompeu o upload do arquivo



Upload de arquivos no PHP [3]

1. Criar um formulário compatível com envio de arquivos
2. Receber os dados do arquivo na variável `$_FILES`
3. Movimentar arquivo no servidor: `move_upload_file()`





Código para movimentar o arquivo no servidor

```
<?php
$diretorio_upload = 'imagens/';
$arquivo_upload = $diretorio_upload . $_FILES['foto']['name'];
if (move_uploaded_file($_FILES['foto']['tmp_name'],$arquivo_upload))
|     echo "Upload realizado com sucesso";
else
|     echo "Erro no upload";
?>
```

- Importante criar um diretório com permissões de acesso ao usuário do Apache (ou outro servidor web)
- Avaliar se vale usar o nome de arquivo fornecido pelo usuário





Mais referências sobre o tema

- [Boson Treinamentos \(YouTube\). Curso de Modelagem de Dados](#)
- [Miriam TechCode \(YouTube\). Modelagem de dados - modelo conceitual, lógico e físico.](#)
- [Manual de Referência do MySQL \(documentação oficial\)](#)
- [Tutorial de MySQL do W3Schools](#)





DÚVIDAS?



Créditos:

Profª: Carolina Sacramento
Carolina.sacramento@fiocruz.br

