

INSTITUTO FEDERAL

Goiano

Campus Morrinhos

BANCO DE DADOS

SQL

PROFESSOR MARCEL MELO

MARCEL.MELO@IFGOIANO.EDU.BR

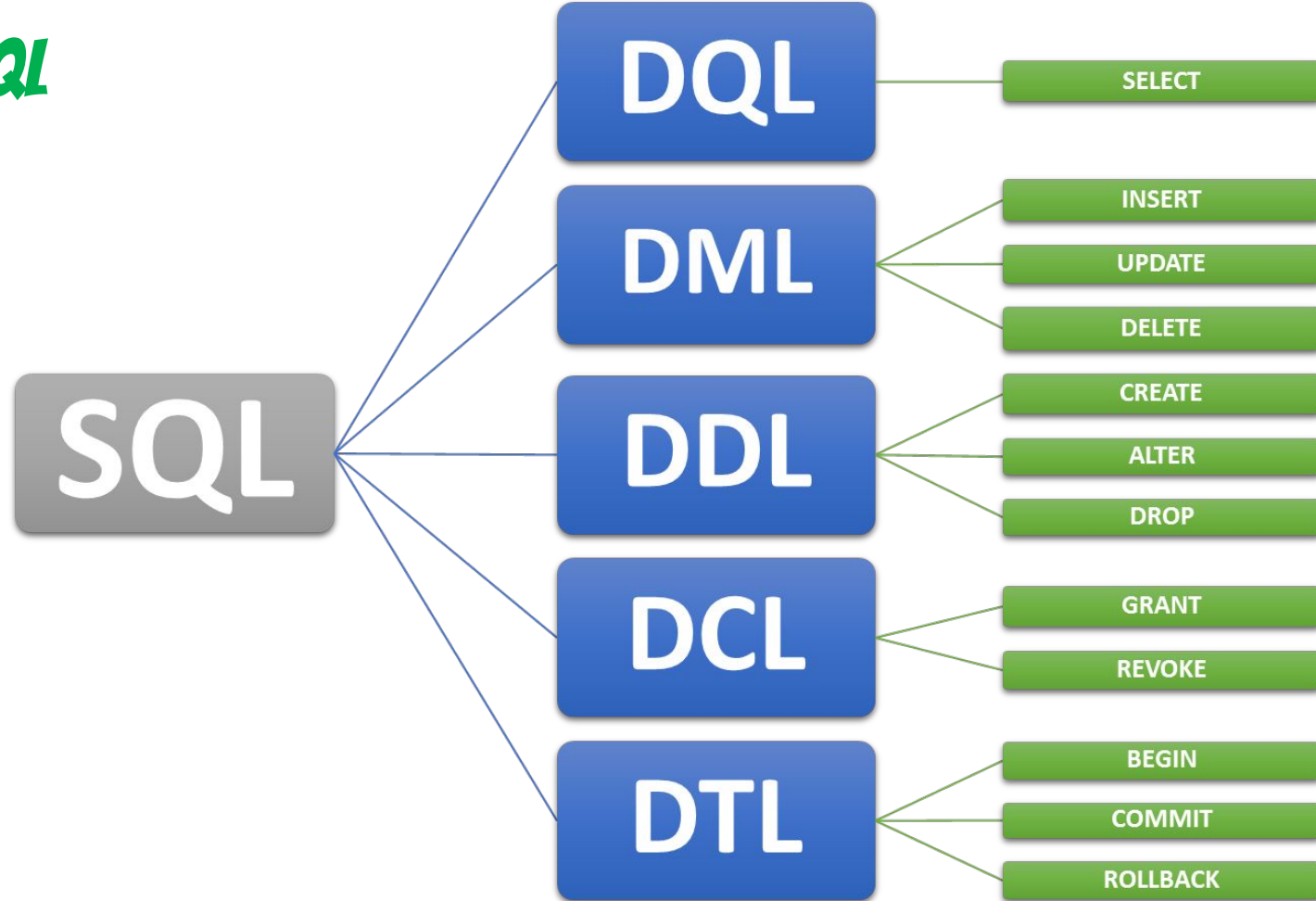
SQL - Structured Query Language

É uma linguagem para acesso e manipulação de banco de dados relacionais.

Apesar de ser padronizada pela ANSI (American National Standards Institute) e ISO (International Standards Organizations) possui diferenças entre versões da linguagem.

A linguagem SQL é formada pelo conjunto das seguintes linguagens:

- **Data Definition Language** - Linguagem de definição de dados
- **Data Manipulation Language** - Linguagem de manipulação de dados
- **Data Query Language** - Linguagem de consulta de dados
- **Data Control Language** - Linguagem de controle de dados
- **Data Transaction Language** - Linguagem de transação de dados



Fonte: <https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>

Data Definition Language () **DDL** é utilizada para a criação do esquema do banco de dados. Os principais comandos desta linguagem são:

- Criação de Banco de Dados (CREATE DATABASE)
- Criação de tabelas (CREATE TABLE)
- Alteração de Banco de Dados (ALTER DATABASE)
- Alteração de tabelas (ALTER TABLE)
- Remoção de Banco de Dados (DROP DATABASE)
- Remoção de tabelas (DROP TABLE)

CRIAÇÃO / ALTERAÇÃO / EXCLUSÃO
BANCO DE DADOS

CREATE DATABASE - Utilizado para criação de um banco de dados. Dentro da estrutura do Banco de Dados que são criadas as tabelas.

CREATE DATABASE <nome_database>

ALTER DATABASE - Permite alterar algumas propriedades do banco de dados.

ALTER DATABASE <nome_database> **SET** <propriedade>

DROP DATABASE - Permite apagar todo banco de dados do sistema

DROP DATABASE <nome_database>

```
CREATE DATABASE aula_Marcel
```


CRIAÇÃO / ALTERAÇÃO / EXCLUSÃO
TABELAS

CREATE TABLE

Criar a estrutura de uma tabela definindo as colunas, seus tipos e restrições, as chaves primárias e chaves estrangeiras existentes.

```
CREATE TABLE <nome-tabela>
(
    coluna1    tipo_dados    constraint,
    coluna2    tipo_dados    constraint,
    coluna3    tipo_dados    constraint,
    PRIMARY KEY (coluna)
)
```

CREATE TABLE - EXEMPLO

Usuarios(id, nome, cpf, idade)

```
CREATE TABLE Usuarios
(
    id int NOT NULL AUTO_INCREMENT,
    nome varchar(50) NOT NULL,
    cpf varchar(14) NOT NULL UNIQUE,
    idade int NULL,
    PRIMARY KEY (id)
)
```

ALTER TABLE <nome-tabela> **DROP COLUMN** colunaX

ALTER TABLE <nome-tabela> **ADD** colunaX tipo_dados
constraint

ALTER TABLE <nome-tabela> **RENAME COLUMN** colunaX **TO**
colunaY

ALTER TABLE <nome-tabela> **MODIFY** colunaX tipo_dados
constraint

```
ALTER TABLE <nome-tabela> ADD PRIMARY KEY  
<nome-coluna>
```

```
ALTER TABLE <nome-tabela> DROP PRIMARY KEY
```

```
ALTER TABLE <nome-tabela> ADD FOREIGN KEY  
<nome-coluna-chave-estrangeira> REFERENCES  
(nome-tabela-pai) ON DELETE [RESTRICT] [CASCADE] [SET  
NULL]
```

```
ALTER TABLE <nome-tabela> DROP CONSTRAINT  
nome-constraint
```

ALTER TABLE - EXEMPLO

```
ALTER TABLE Usuarios MODIFY nome varchar(100)
```

```
ALTER TABLE Usuarios DROP COLUMN idade;
```

```
ALTER TABLE Usuarios ADD endereco varchar(50) NULL;
```

Excluir a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão excluídos todos os dados, estrutura e índices de acessos que estejam a ela associados.

```
DROP TABLE <nome-tabela>
```

Exemplo:

```
DROP TABLE Usuarios
```

TIPO DE DADOS

TIPO DE DADOS NUMÉRICOS

TIPO DE DADOS	FAIXA DE VALORES	TAMANHO
TINYINT	SIGNED = -128 a 127 / UNSIGNED = 0 a 255	1 byte
SMALLINT	SIGNED = -32768 a 32768 / UNSIGNED = 0 a 65535	2 bytes
MEDIUMINT	SIGNED = -8388608 a 8388608 / UNSIGNED = 0 a 16777215	3 bytes
INT / INTEGER	SIGNED = -2147483648 a 2147483648 / UNSIGNED = 0 a 4294967295	4 bytes
BIGINT	SIGNED = -9223372036854775808 a 9223372036854775808 UNSIGNED = 0 a 18446744073709551615	8 bytes

TIPO DE DADOS NUMÉRICOS FRACIONÁRIOS

TIPO DE DADOS	DESCRIÇÃO
FLOAT(N)	O valor N define quando o campo utilizará 4 ou 8 bytes. float(24) utiliza 4 bytes enquanto float(53) 8 bytes.
REAL	Valor de ponto flutuante com precisão variável entre $-3.40E + 38$ to $3.40E + 38$ (4 bytes)
DECIMAL(P,S) NUMERIC(P,S)	Valor com precisão fixa. P indica o máximo de dígitos que o número utilizará (tanto antes do ponto quanto após). O valor de P deve ser entre 1 e 38 com valor padrão de 18. S indica a quantidade máxima de dígitos que será armazenada na casa decimal. O valor padrão é 0.
SMALLMONEY	Utilizado para valores monetários. Faixa: -214,748.3648 a 214,748.3647 (4 bytes)
MONEY	Utilizado para valores monetários. Faixa: -922,337,203,685,477.5808 a 922,337,203,685,477.5807 (8 bytes)

TIPO DE DADOS - DATA E HORA

TIPO DE DADOS	DESCRIÇÃO
DATE	Armazena uma data no formato AAAA-MM-DD. (3 bytes) Faixa suportada: '1000-01-01' e '9999-12-31'
DATETIME	Armazena uma data e hora no formato AAAA-MM-DD HH:MM:SS. A faixa suportada é entre '1000-01-01 00:00:00' e '9999-12-31 23:59:59'
TIMESTAMP	Armazena uma data e hora no formato AAAA-MM-DD HH:MM:SS. A faixa é entre '1970-01-01 00:00:00' e algum momento no ano 2037. (4 bytes)
TIME	Armazena uma hora específica. A faixa é entre '-838:59:59' e '838:59:59'
YEAR	Um ano no formato de 2 ou 4 dígitos (padrão são 4 dígitos). Os valores permitidos estão entre 1901 e 2155, 0000 no formato de 4 dígitos, e 1970-2069 se você estiver usando o formato de 2 dígitos (70-69).

TIPO DE DADOS - STRING

TIPO DE DADOS	DESCRIÇÃO
CHAR	Armazena um caractere (1 byte)
VARCHAR(N)	Armazena uma String de tamanho máximo variável, definido por N. O valor de N pode variar entre 1 e 255. Se N for definido acima de 255, a coluna é convertida para TEXT
TINYTEXT	Armazena um texto de até 255 caracteres.
TEXT	Armazena um texto de até 65535 caracteres.
MEDIUMTEXT	Armazena um texto de até 16777215 caracteres.
LONGTEXT	Armazena um texto de até 4294967295 caracteres (4 Gigas).

TIPO DE DADOS - BOOLEAN E BLOB (BINARY LARGE OBJECTS)

TIPO DE DADOS	DESCRIÇÃO
BOOLEAN	Armazena um valor booleano (TINYINT) 0 representa o valor FALSO Qualquer outro caractere é VERDADEIRO
TINYBLOB	Armazena um texto de até 255 caracteres.
BLOB	Armazena um texto de até 65535 caracteres.
MEDIUMBLOB	Armazena um texto de até 16777215 caracteres.
LOB	Armazena um texto de até 4294967295 caracteres (4 Gigas).

TIPO DE DADOS - SET E ENUM

TIPO DE DADOS	DESCRIÇÃO
ENUM('valor1', 'valor2',)	Uma enumeração. Um objeto string que só pode ter um valor, selecionado da lista de valores 'valor1', 'valor2', ..., NULL. Um ENUM pode ter um máximo de 65535 valores diferentes.
SET('valor1', 'valor2',)	Um objeto string que pode ter zero ou mais valores, cada um deve ser selecionado da lista de valores 'valor1', 'valor2', Um SET pode ter até 64 membros.

Documentação MySQL:

<http://ftp.tcrc.edu.tw/MySQL/doc/refman/4.1/pt/column-types.html>

W3Schools:

https://www.w3schools.com/sql/sql_datatypes.asp

EXEMPLO

Crie os scripts DDL para o modelo relacional apresentado a seguir. Na criação das tabelas é necessário criar as chaves-primárias para cada tabela.

