



PROGRAMAÇÃO SQL

DDL | DML | DQL

OBJECTIVOS

- DDL
- DML
- DQL

SGBD

- Sistema Gestão de Bases de Dados
- Um sistema de software que permite que os utilizadores definam, criem e mantenham a base de dados e providencia um acesso controlado a essa base de dados.

SGBD



CREATE



READ



UPDATE



DELETE

C

R

U

D

SGBD

- EXEMPLOS

- Microsoft Access,
- Microsoft SQL Server,
- MySQL,
- Oracle DB,
- etc.



SQL

- A Linguagem SQL (Structured Query Language) é essencialmente uma linguagem de pesquisa (query) associada aos SGBDr.
- É também uma linguagem declarativa que permite definir, questionar e manipular os dados de uma base de dados.

SQL

- A Linguagem SQL (Structured Query Language) incorpora várias facilidades que permitem:
 - Definir restrições de integridade (CHAVES);
 - Definir vistas sobre os dados (VIEWS);
 - Especificar procedimentos (STORED PROCEDURES);
 - Especificar permissões de segurança e de acesso (ROLES);
 - Interligar a outras linguagens de programação.

ÂMBITO DA LINGUAGEM SQL

- DDL (Data Definition Language) – Definição de Dados
 - Create
 - Alter
 - Drop
- DML (Data Manipulation Language) – Linguagem de manipulação de dados
 - Insert
 - Update
 - Delete
- DQL (Data Query Language) – Linguagem de Pesquisa de Dados
 - Select

MySQL

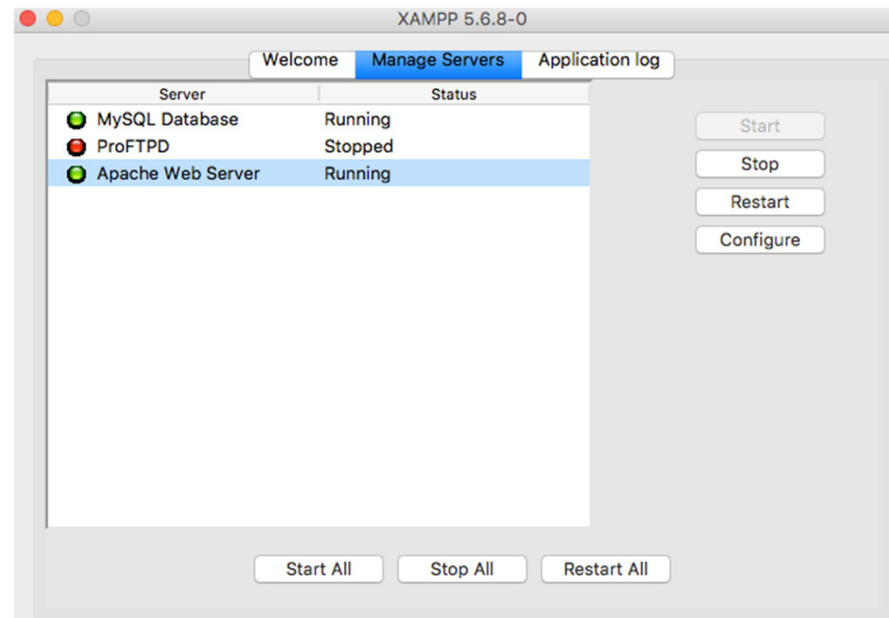
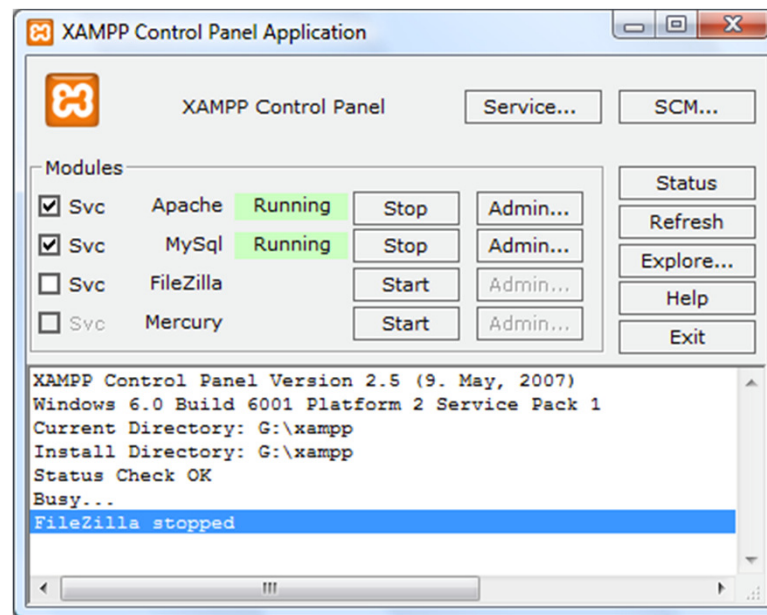
- Sistema de Gestor de Bases de Dados gratuito.
- É necessário instalar o software MySQL.
 - Em alternativa, pode ser instalado o XAMPP que além do MySQL ainda incorpora o Apache e suporta interpretação de PHP e Pearl.
 - <https://www.apachefriends.org/index.html>

XAMPP

- XAMPP é o ambiente de desenvolvimento PHP mais popular
- Completamente gratuito
- Fácil de instalar
- Código aberto
- Inclui Apache + (MariaDB) MySQL + PHP + Perl
- O X representa qualquer Sistema Operativo

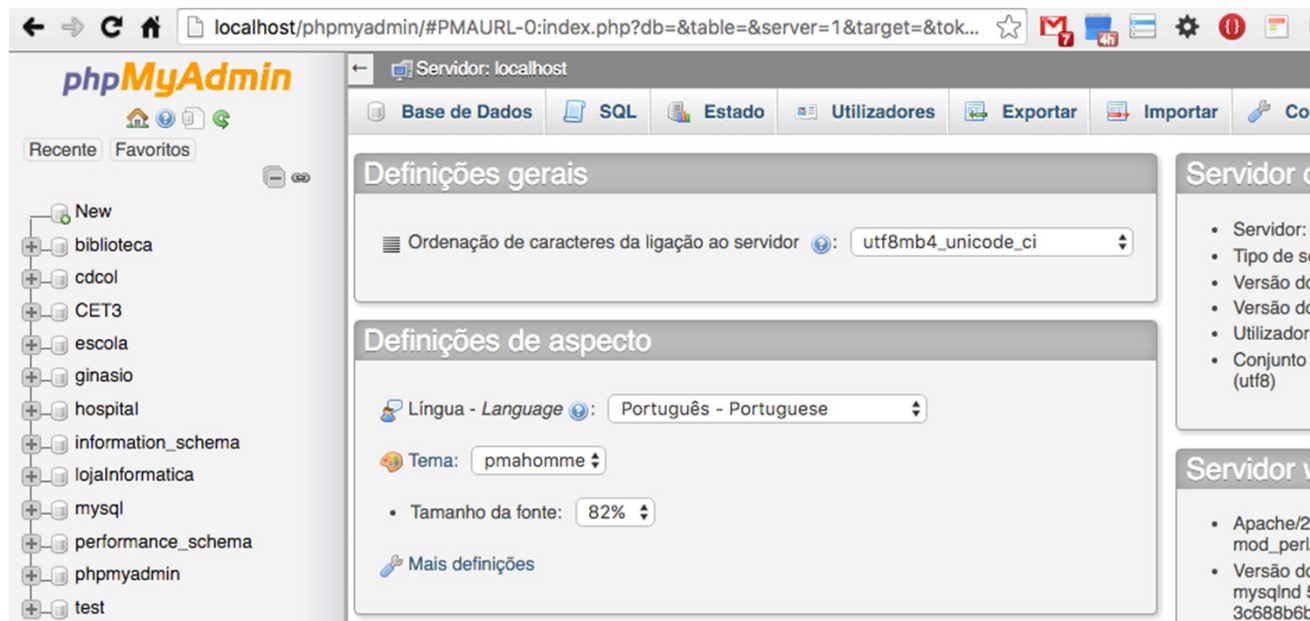
XAMPP

- Iniciar os serviços Apache e MySQL (Windows e MacOS)



XAMMP

- Aceder ao gestor do MySQL através do browser:
 - localhost/phpmyadmin



OBJECTIVOS

- DDL
- DML
- DQL

DDL

- DDL ou Data Definition Language (Definição de dados) consistem em comandos SQL que são usados para criar/eliminar/alterar dados:
 - **CREATE:** cria um objeto (DATABASE, TABLE, INDEX, VIEW, ...)
 - **DROP:** apaga um objeto da base de dados
 - **ALTER:** permite alterar um objeto (ex.: adicionar atributo a uma tabela)
 - ADD
 - DROP
 - MODIFY
 - **RENAME:** permite renomear um objeto existente

CREATE DATABASE

- Criar uma Base de Dados

```
CREATE DATABASE <Nome>;
```

DROP DATABASE

- Eliminar uma Base de Dados

```
DROP DATABASE <Nome>;
```


CREATE TABLE

- Criar uma tabela

```
CREATE TABLE <Nome> (  
    <Atributo1> <Domínio1> <Restrições1>,  
    <Atributo2> <Domínio2> <Restrições2>,  
    <Atributo3> <Domínio3> <Restrições3>,  
    ...  
);
```

EXEMPLO

- Criar uma tabela ALUNO

Aluno	
PK	<u>Num_Aluno</u>
	Nome
	Morada
	CodPostal
	Localidade
	DataInscricao
	Idade

Até ao momento só definíamos o nome dos atributos

EXEMPLO

- Criar uma tabela ALUNO

```
CREATE TABLE Aluno (  
    Num_Aluno <Domínio1> <Restrições1>,  
    Nome <Domínio2> <Restrições2>,  
    Morada <Domínio3> <Restrições3>,  
    CodPostal <Domínio4> <Restrições4>,  
    Localidade <Domínio5> <Restrições5>,  
    DataInscricao <Domínio6> <Restrições6>,  
    Idade <Domínio7> <Restrições7>  
);
```

DOMÍNIO?

- Criar uma tabela ALUNO

```
CREATE TABLE Aluno (  
    Num_Aluno <Domínio1> <Restrições1>,  
    Nome <Domínio2> <Restrições2>,  
    Morada <Domínio3> <Restrições3>,  
    CodPostal <Domínio4> <Restrições4>,  
    Localidade <Domínio5> <Restrições5>,  
    DataInscricao <Domínio6> <Restrições6>,  
    Idade <Domínio7> <Restrições7>  
);
```

DOMÍNIO DOS ATRIBUTOS

- O domínio dos atributos é o conjunto de valores possíveis para um dado atributo.
- Em SQL é o que permite restringir a entrada de dados num atributo a um tipo de dados que é definido na criação da tabela.

DOMÍNIO O DOS ATRIBUTOS

- **Valores Numéricos:**

- **TINYINT:** número inteiro desde -128 até 127 (1 byte)
- **SMALLINT:** número inteiro desde -32768 até 32767 (2 bytes)
- **MEDIUMINT:** número inteiro desde -8388608 até 8388607 (3 bytes)
- **INT:** valores inteiros desde -2147483648 até 2147483647 (4bytes)
- **BIGINT:** número inteiro com desde -9.223.372.036.854.775.808 até 9.223.372.036.854.775.807 (8bytes)
- **FLOAT:** número fracionário (4 bytes)
- **DOUBLE:** número fracionário (8 bytes)
- **DECIMAL:** Número fracionário desempacotado: DECIMAL(M,D)
 - M dígitos com D dígitos à direita do ponto decimal

DOMÍNIO DOS ATRIBUTOS

- **Sequências de texto:**

- **CHAR(*n*):** texto de comprimento fixo entre 0 e 255 caracteres
- **VARCHAR(*n*):** texto de comprimento variável até 65.535 caracteres
- **TINYTEXT:** texto de comprimento variável até 255 caracteres
- **TEXT:** texto de comprimento variável até 65.535 caracteres
- **MEDIUMTEXT:** texto de comprimento variável até 16.777.215 caracteres
- **LONGTEXT:** texto de comprimento variável até 4.294.967.295 caracteres

Aqueles com *n* implica indicar o número máximo de caracteres que terá esse atributo

- **BLOB(*n*):** Binary Large Objects, tipicamente usado para imagens, vídeos, etc.

DOMÍNIO DOS ATRIBUTOS

- **Valores Temporais:**

- **DATE:** Data, 'AAAA-MM-DD' (Ex: '2022-03-16')
- **TIME:** Hora, 'HH:MM:SS' (Ex: '12:53:47')
- **DATETIME:** Data e Hora, 'AAAA-MM-DD HH:MM:SS' (Ex: '2022-03-16 12:53:47')
- **TIMESTAMP:** Data e Hora, 'AAAAMMDDHHMMSS' (Ex: '20220316125347')

- **Valores Lógicos:**

- **BOOL:** Falso (0) ou Verdadeiro (tudo que for diferente de 0)
- **BOOLEAN:** Falso (FALSE) ou Verdadeiro (TRUE)

EXEMPLO

- Definir domínio dos atributos da tabela ALUNO

Aluno	
PK	<u>Num_Aluno</u> INT
	Nome VARCHAR(50)
	Morada VARCHAR(70)
	CodPostal CHAR(8)
	Localidade VARCHAR(25)
	DataInscricao DATE
	Idade SMALLINT

Por norma, o domínio dos atributos é identificado no Modelo de Dados

EXEMPLO

- Criar uma tabela ALUNO

```
CREATE TABLE Aluno (  
    Num_Aluno INT <Restrições1>,  
    Nome VARCHAR(50) <Restrições2>,  
    Morada VARCHAR(70) <Restrições3>,  
    CodPostal CHAR(8) <Restrições4>,  
    Localidade VARCHAR(25) <Restrições5>,  
    DataInscricao DATE <Restrições6>,  
    Idade SMALLINT <Restrições7>  
);
```

RESTRIÇÕES?

- Criar uma tabela ALUNO

```
CREATE TABLE Aluno (  
    Num_Aluno INT <Restrições1>,  
    Nome VARCHAR(50) <Restrições2>,  
    Morada VARCHAR(70) <Restrições3>,  
    CodPostal CHAR(8) <Restrições4>,  
    Localidade VARCHAR(25) <Restrições5>,  
    DataInscricao DATE <Restrições6>,  
    Idade SMALLINT <Restrições7>  
);
```

RESTRIÇÕES ATRIBUTOS

- Servem para aplicar regras específicas dentro do domínio dos atributos.
- Em SQL é o que permite limitar a entrada de dados num atributo.

Se existir alguma violação entre a restrição imposta e a ação dada, a mesma é abortada.

RESTRIÇÕES ATRIBUTOS

- **NOT NULL** – Não permite que o valor de um campo seja nulo ou vazio
 - `<Atributo> <Dominio> NOT NULL`
- **DEFAULT** *valor* – Define um valor por omissão que é adicionado mas que pode ser alterado
 - `<Atributo> <Dominio> DEFAULT <VAL>`
- **CHECK** *condição* – Faz validação aos valores de um campo de acordo com uma expressão
 - `<Atributo> <Dominio> CHECK (<COND>)`
- **UNIQUE** – Não permite que o valor de um campo seja repetido
 - `UNIQUE (<Atributo>)`

RESTRIÇÕES ATRIBUTOS

- **PRIMARY KEY** – Identifica um campo único como chave primária.
 - PRIMARY KEY (<Atributo(s)>)
- **FOREIGN KEY** – Identifica um campo como chave estrangeira – tem de referenciar a tabela mãe.
 - FOREIGN KEY (<Atributo(s)>)
REFERENCES <TABELA> (<Chave(s)>)
- **AUTO INCREMENT** – Incremento automático do MySQL (começa em 1 e é acrescentado sempre +1)

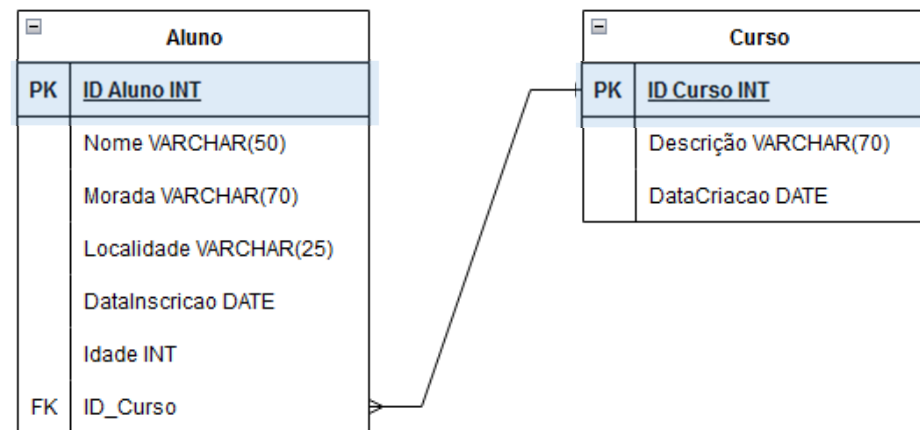
RESTRIÇÕES?

- Criar uma tabela ALUNO

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18)  
);
```

CHAVE PRIMÁRIA

- Exemplo



CHAVE PRIMÁRIA

- Criar uma tabela ALUNO com CHAVE PRIMÁRIA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18)  
);
```

CHAVE PRIMÁRIA

- Criar uma tabela ALUNO com CHAVE PRIMÁRIA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18)  
);
```

APENAS PODE SER USADO
PARA CASOS DE TABELAS
COM CHAVE PRIMÁRIA
CONSTITUIDA POR UM SÓ
CAMPO

CHAVE PRIMÁRIA

- Criar uma tabela ALUNO com CHAVE PRIMÁRIA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    PRIMARY KEY (Num_Aluno)  
);
```

PARA CASOS COM VÁRIOS
ATRIBUTOS COMO CHAVE
EX: PRIMARY KEY
(ATRIBUTO1, ATRIBUTO2,
ETC..)

CHAVE PRIMÁRIA

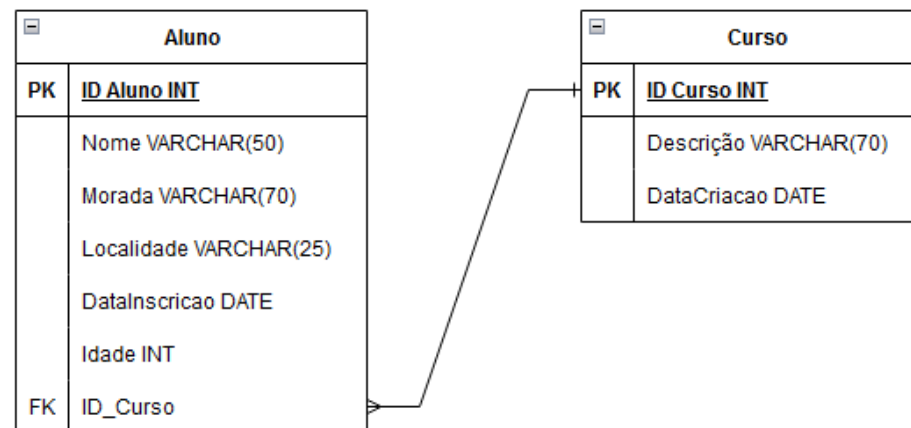
- Criar uma tabela ALUNO com CHAVE PRIMÁRIA definindo nome para a chave

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    CONSTRAINT PK_Aluno PRIMARY KEY (Num_Aluno)  
);
```

SERVE PARA DEFINIR UM
NOME À RESTRIÇÃO

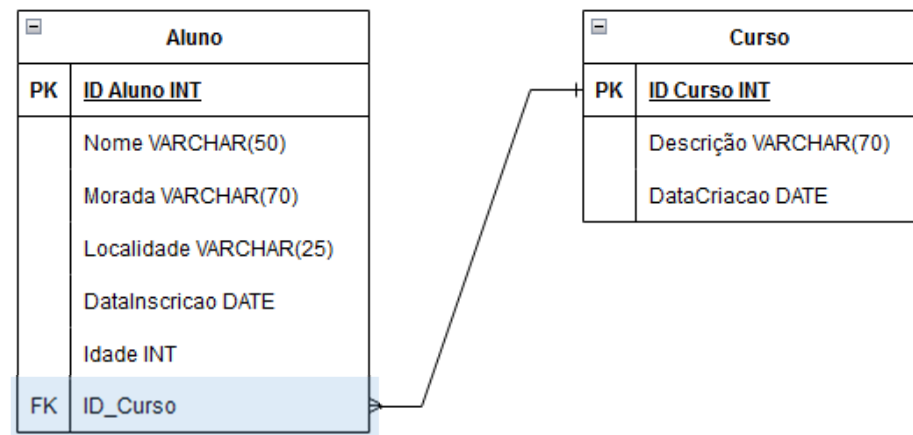
CHAVE ESTRANGEIRA

- Exemplo



CHAVE ESTRANGEIRA

- Exemplo



O atributo chave estrangeira deverá ser adicionada na tabela onde está presente

CHAVE ESTRANGEIRA

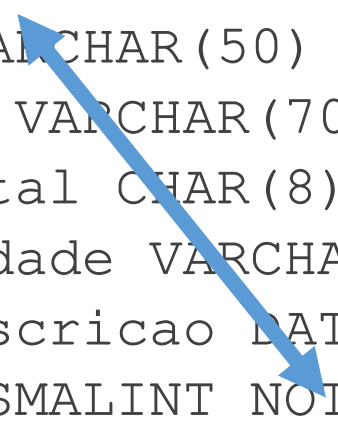
- Criar uma tabela ALUNO com CHAVE ESTRANGEIRA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    FOREIGN KEY (Num_Aluno) REFERENCES Disciplinas (idAluno)  
);
```

CHAVE ESTRANGEIRA

- Criar uma tabela ALUNO com CHAVE ESTRANGEIRA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    FOREIGN KEY (Num_Aluno) REFERENCES Disciplinas (idAluno)  
);
```



CHAVE ESTRANGEIRA

- Criar uma tabela ALUNO com CHAVE ESTRANGEIRA

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    FOREIGN KEY (Num_Aluno) REFERENCES Disciplinas (idAluno)  
);
```

DEVE SER REFERENCIADO
PRIMEIRO A TABELA E DE
SEGUIDA O ATRIBUTO,
ONDE ESTE É CHAVE
ESTRANGEIRA

CHAVE ESTRANGEIRA

- Criar uma tabela ALUNO com CHAVE ESTRANGEIRA definindo nome para a chave

```
CREATE TABLE Aluno (  
    Num_Aluno INT AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Morada VARCHAR(70) NOT NULL,  
    CodPostal CHAR(8) NOT NULL,  
    Localidade VARCHAR(25),  
    DataInscricao DATE DEFAULT '2022-01-01',  
    Idade SMALLINT NOT NULL CHECK (Idade > 18),  
    CONSTRAINT FK_Aluno FOREIGN KEY (Num_Aluno)  
    REFERENCES Disciplinas(idAluno)
```

SERVE PARA DEFINIR UM
NOME À RESTRIÇÃO

DROP TABLE

- Eliminar uma tabela

```
DROP TABLE <Nome>;
```

ALTER (ADD)

- Adicionar novo atributo uma tabela

```
ALTER TABLE <Nome>  
ADD <Atributo> <Domínio> <Restrições>;
```

EXEMPLO

- Adicionar novo atributo uma tabela

```
ALTER TABLE Aluno  
ADD Idade SMALLINT;
```

EXEMPLO

- Alterar um atributo de uma tabela (adicionar uma chave primária posteriormente)

```
ALTER TABLE  Aluno  
ADD CONSTRAINT PK_Aluno  
PRIMARY KEY  (Num_Aluno);
```

EXEMPLO

- Alterar um atributo de uma tabela (adicionar uma chave estrangeira posteriormente)

```
ALTER TABLE  Aluno  
ADD CONSTRAINT FK_Aluno  
    FOREIGN KEY  (Num_Aluno)  
    REFERENCES  Disciplina(idAluno) ;
```

ALTER (MODIFY)

- Alterar atributos existentes de uma tabela

```
ALTER TABLE <Nome>  
MODIFY COLUMN <Atributo> <Domínio> <Restrições>;
```


EXEMPLO

- Adicionar novos atributos uma tabela

```
ALTER TABLE Aluno  
MODIFY COLUMN Idade SMALLINT CHECK (Idade>18) ;
```

ALTER (DROP)

- Eliminar um atributo de uma tabela

```
ALTER TABLE  <Nome>  
DROP COLUMN <Atributo>;
```

EXEMPLO

- Eliminar um atributo de uma tabela

```
ALTER TABLE Aluno  
DROP COLUMN DataInscricao;
```

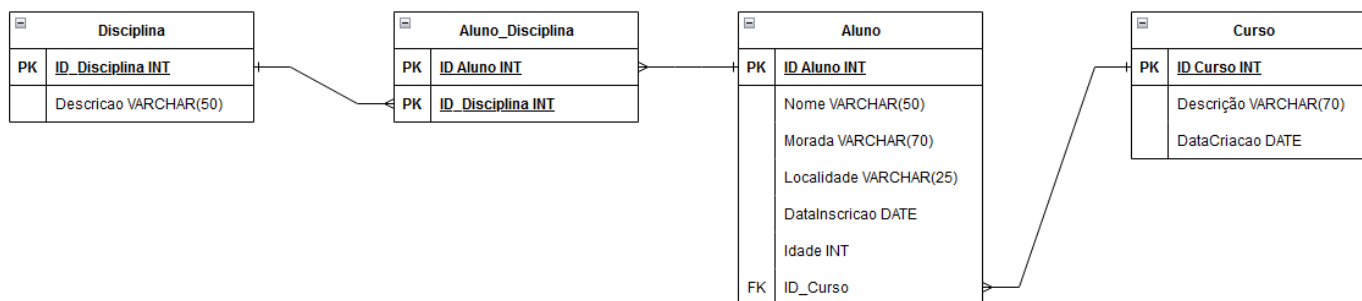
EXEMPLO

- Alterar um atributo de uma tabela (apagar chave primária posteriormente)

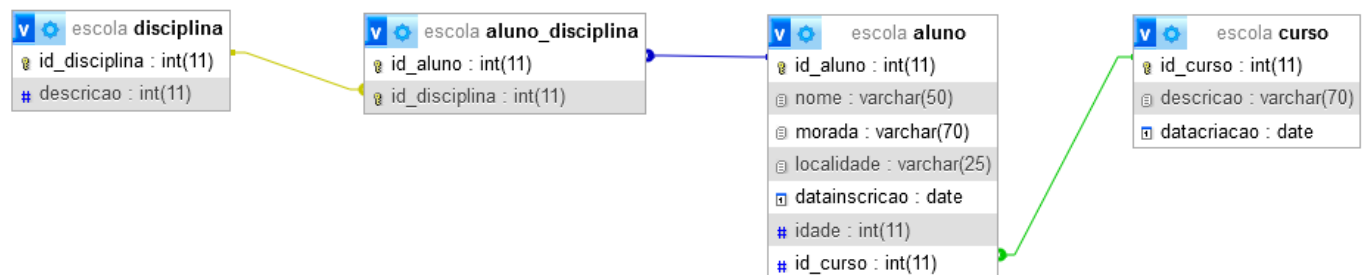
```
ALTER TABLE Aluno  
DROP PRIMARY KEY;
```

DDL

Modelo de Dados

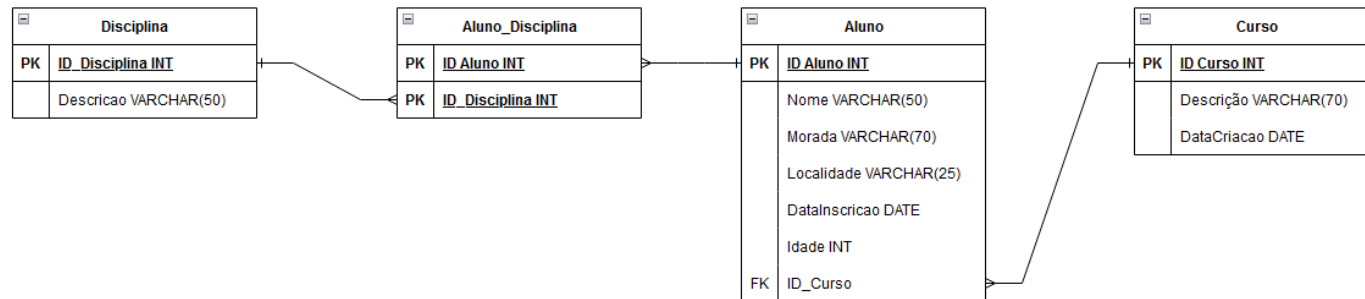


Modelo Físico



EXERCICIO

Modelo de Dados



Modelo Físico

