

# Linguagem SQL

## Data Definition Language (DDL)

Profa Dra Jeroniza Nunes Marchaukoski

# Linguagem SQL

- Combina construtores em Álgebra e Cálculo Relacional e possui as seguintes partes:
  - Linguagem de definição de dados (DDL)
  - Definição de Visões (DDL e Consulta)
  - Autorização (DDL)
  - Integridade (DDL)
  - Controle de Transações (DDL)
  - Linguagem interativa de manipulação de dados (DML - Query)
  - Incorporada DML (*Embedded SQL*)

# Linguagem SQL - DDL

- Objetos componentes do Banco de Dados:
  - Tabelas
  - Atributos
  - Domínios
  - Regras
  - Chaves
  - Relacionamentos
  - Gatilhos
  - Sinônimos
  - Índices
  - Visões
  - Funções e procedimentos etc

# Linguagem SQL - DDL

- Uso dos comandos DDL para manipular os componentes do banco de dados e para implementar as regras de integridade:
  - Create
  - Drop

# Linguagem SQL – DDL Tabela

- Create table: Cria a estrutura de uma tabela(arquivo) definido as colunas (campos), as chaves primárias, chaves estrangeiras e demais restrições de domínio.

Sintaxe:

```
CREATE TABLE <nome-tabela>  
(<nome-coluna> , <tipo-do-dado> [NOT NULL]  
(<nome-coluna> CHECK <restrição>  
PRIMARY KEY (nome(S)-coluna(S)-chave)  
UNIQUE KEY (nome(S)-coluna(S)-chave)  
FOREIGN KEY (nome(S)-coluna(S) - chave-estrangeira)  
REFERENCES (nome-tabela-pai) ON DELETE [CASCADE]
```

# Linguagem SQL – DDL Tabelas

- Create table:
  - a) nome-tabela - Representa o nome da tabela que será criada.
  - b) nome-coluna - Representa o nome da coluna que será criada. A definição das colunas de uma tabela é feita relacionando-as uma após a outra.
  - c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela. Os tipos de dados mais comuns serão definidos mais à frente.
  - d) NOT NULL - Exige o preenchimento do campo, ou seja, no momento da inclusão é obrigatório que possua um conteúdo.
  - e) CHECK – Restrição de domínio. Limita o conjunto de valores permitidos para o campo.

# Linguagem SQL – DDL Tabelas

- Create table:
  - f) PRIMARY KEY (nome-coluna(s)-chave) - Definir para o banco de dados a(s) coluna(s) que será a chave primária da tabela. Caso ela tenha mais de um coluna como chave, elas deverão ser relacionadas entre os parênteses.
  - g) UNIQUE (nome-coluna(s)) – Definir uma chave candidata. Valor que não se duplique nos registros. Ao contrário da PK pode ficar vazia.
  - h) FOREIGN KEY (nome-coluna(s)-chave-estrangeira) REFERENCES (nome-tabela-pai) - Definir para o banco de dados as colunas que são chaves estrangeiras, ou seja, os campos que são chaves primárias de outras tabelas. Na opção REFERENCES deve ser especificado a tabela na qual a coluna é a chave primária.

# Linguagem SQL – DDL Tabelas

- Create table:
  - i) ON DELETE - Esta opção especifica os procedimentos que devem ser feitos pelo SGBD quando houver uma exclusão de um registro na tabela pai quando existe um registro correspondente nas tabelas filhas. As opções disponíveis são:
  - CASCADE - Esta opção realiza a exclusão em todas as tabelas filhas que possuam o valor da chave que será excluída na tabela pai.

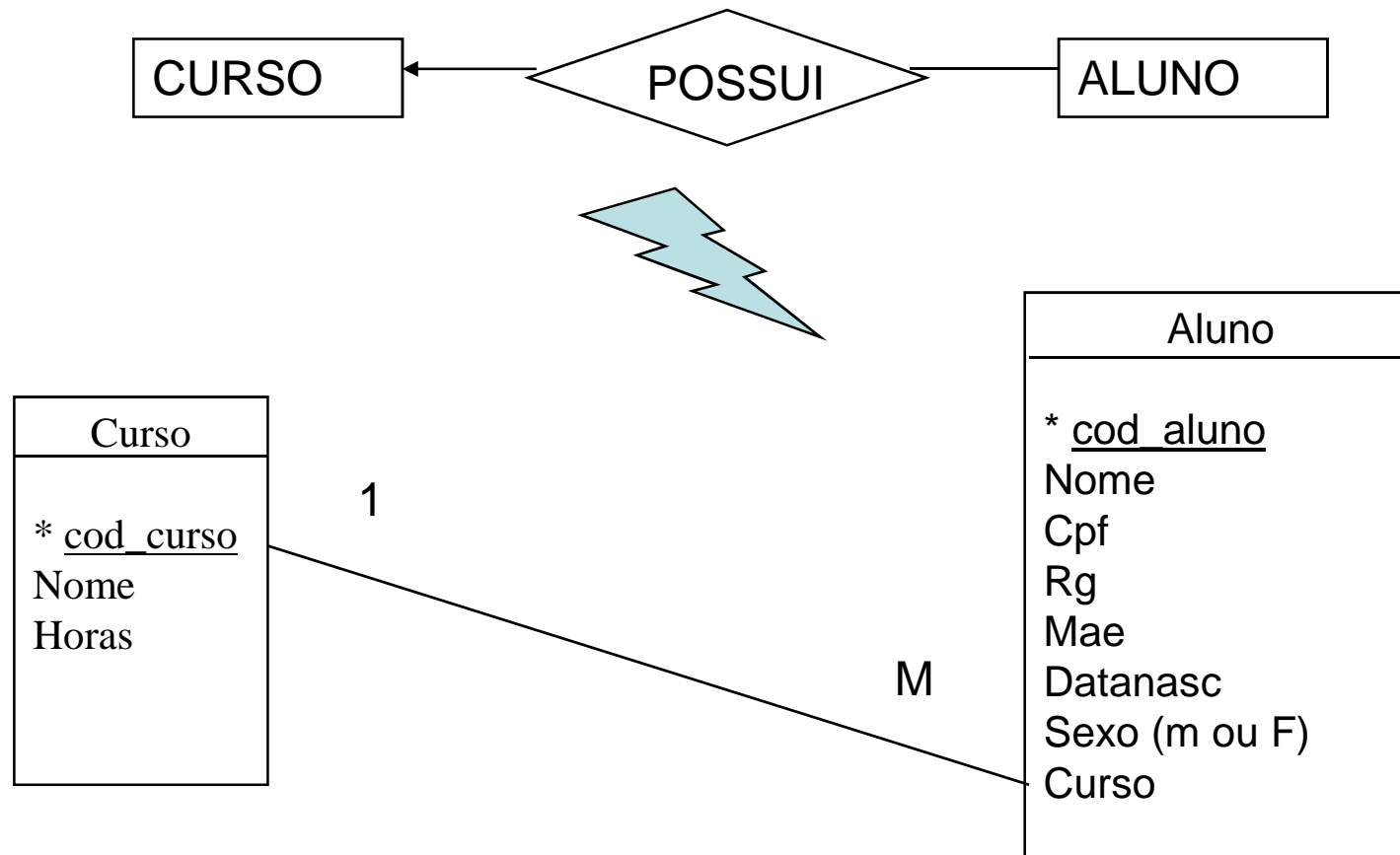


# Linguagem SQL – DDL Tabelas

- Create table:
  - Tipos de dados mais comuns:
  - 1) Numéricos:
    - Integer - Armazena valores numéricos, em quatro bytes binários, compreendidos entre o intervalo -2147483648 a +2147483647
    - Number
  - 2) Alfanuméricos:
    - Varchar (n) - Definir um campo alfanumérico de até n caracteres, onde n deve ser menor ou igual a 2000 caracteres (ou maior dependendo do SGBD).
    - Char (n) - Definir um campo alfanumérico de n caracteres, onde n deve ser menor ou igual a 254 caracteres.

# Linguagem SQL DDL Tabelas

- Exemplo 1: Seja um curso que possui muitos alunos a ele vinculados.



# Linguagem SQL DDL Tabelas

– Exemplo 1 (Oracle):

- **create table** curso (
  - cod\_curso **number(5)**,
  - nome **varchar2(50)**,
  - horas **number(5)**,
  - **constraint** pkcurso **primary key** (cod\_curso))
- **create table** aluno (
  - cod\_aluno **number(5)**, nome **varchar2(50)**, dt\_nasc **date**,
  - mae **varchar2(50)**, sexo **char(1)**, **curso number(5)**,
  - **constraint** pkaluno **primary key** (cod\_aluno),
  - **constraint** ukaluno **unique**(nome, dt\_nasc, mae),
  - **constraint** sexo **check** (sexo in('m','f')),
  - **constraint** fkcursoAluno **foreign key** (curso) **references** curso(cod\_curso))

# Linguagem SQL DDL Tabelas

- **Alter Table:** Altera a estrutura de uma tabela(arquivo) acrescentando, alterando, retirando e alterando nomes, formatos das colunas e a integridade referencial definidas em uma determinada tabela.
- Sintaxe:
  - **ALTER TABLE** <nome-tabela>
  - **DROP COLUMN** <nome-coluna>
  - **ADD** <nome-coluna> <tipo-do-dado> [NOT NULL]
  - [NOT NULL WITH  
DEFAULT]
  - **RENAME COLUMN** <nome-coluna> **TO** <novo-nome-coluna>
  - **RENAME TO** <novo-nome-tabela>
  - **MODIFY** <nome-coluna> <tipo-do-dado> [NULL]
  - [NOT NULL]
  - [NOT NULL WITH  
DEFAULT]
  -

# Linguagem SQL DDL Tabelas

- Alter Table onde:
- a) nome-tabela - Representa o nome da tabela que será atualizada.
- b) nome-coluna - Representa o nome da coluna que será criada.
- c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela.
- d) DROP <nome-coluna> - Realiza a retirada da coluna especificada na estrutura da tabela.

# Linguagem SQL DDL Tabelas

- Alter Table onde:
- e) ADD <nome-coluna> <tipo-do-dado> - Realiza a inclusão da coluna especificada na estrutura da tabela. Na coluna correspondente a este campo nos registros já existentes será preenchido o valor NULL (Nulo). As definições NOT NULL e NOT NULL WITH DEFAULT são semelhantes à do comando CREATE TABLE.
- f) RENAME <nome-coluna> <novo-nome-coluna> - Realiza a troca do nome da coluna especificada.
- g) RENAME TABLE <novo-nome-tabela> - Realiza a troca do nome da tabela especificada.

# Linguagem SQL DDL Tabelas

- Alter Table onde:
- h) MODIFY <nome-coluna> <tipo-do-dado> - Permite a alteração na característica da coluna especificada.
- Opções:
- Além das existentes na opção ADD (NOT NULL e NOT NULL WITH DEFAULT), temos a opção NULL que altera a característica do campo passando a permitir o preenchimento com o valor Nulo.
- i) ADD PRIMARY KEY <nome-coluna> - Esta opção é utilizada quando é acrescentado um novo campo como chave primária da tabela.

# Linguagem SQL DDL Tabelas

- Alter Table onde:
- j) DROP PRIMARY KEY <nome-coluna> - Esta opção é utilizada quando é retirado um campo como chave primária da tabela.
- l) ADD FOREIGN KEY <nome-coluna> - Esta opção é utilizada quando é acrescentado um novo campo sendo ele uma chave estrangeira.
- l) DROP FOREIGN KEY <nome-coluna> - Esta opção é utilizada quando é retirado uma chave estrangeira da estrutura da tabela.



# Linguagem SQL DDL Tabelas

- Alter table nome\_tabela
- Add ... adicionar campo, constraint
- Modify... modificar um campo
- Drop constraint... excluir uma constraint

# Linguagem SQL DDL Tabelas

- Exemplo (Oracle):
  - Alter table aluno add (endereço varchar2(60))
  - \*\*\*\* inserimos o campo endereço
  - alter table aluno modify (endereço varchar2(100))
  - \*\*\* modificamos o campo endereço aumentando o seu tamanho
  - alter table aluno drop constraint cksexo
  - \*\*\* eliminamos a constraint check do campo sexo

# Linguagem SQL DDL Tabelas

- **Drop Table**

- Objetivo:
- Deletar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão deletados todos dados, estrutura e índices de acessos que estejam a ela associados.
- Sintaxe:
- **DROP TABLE** <nome-tabela>
- onde:
- a) nome-tabela - Representa o nome da tabela que será deletada.

# Linguagem SQL DDL Tabelas

- **Drop Table**

- Exemplo (Oracle):
- Drop table curso;