

Definição de Banco de Dados com SQL

Prof. Msc. Luã Marcelo Muriana

Engenharia de Software – 2024/01

Experimentos Práticos com Banco de Dados



Na aula anterior

- Estrutura elementar de uma tabela (relação)
 - Atributos (colunas)
 - Tuplas (cada linha)
- Esquema de relação (cabeçalhos das colunas da tabela)
- Restrições
 - Domínio
 - Chave (Chave Primária e Chave Estrangeira)
 - Integridade de Entidade (PK not null)
 - Integridade Referencial



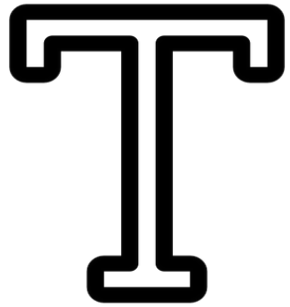
Dados

VS



Informação

Tipos de Dados



TEXTO

São as strings para armazenar caracteres.

Exemplo: CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, NLOB, LONG



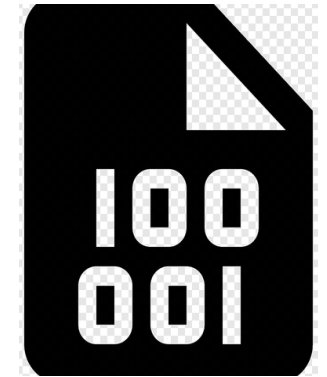
NUMÉRICOS

Onde é possível gravar tipos de dados inteiros ou decimais.
Exemplos: NUMBER e INTEGER



DATAS

São os temporais.
Exemplos: DATE, TIMESTAMP, TIMESTAMP with Time Zone e TIMESTAMP with Local Time Zone



BINÁRIOS

Armazenados no BD e em arquivos do S.O
Exemplos: BLOB e BFILE

Tipos de Dados

TEXTO

CHAR	Armazena cadeias de caracteres de comprimento fixo (1 a 2000 caracteres).
VARCHAR VARCHAR2	São sinônimos. Armazena caracteres de comprimento variável (1 a 4000 caracteres).
NCHAR NCHAR2	Armazena cadeias de caracteres de comprimento fixo ou variável. São caracteres Unicode.
CLOB NLOB	São campos longos que podem armazenar até 8 Terabytes de dados, por campo.

Tipos de Dados

NUMÉRICO

NUMBER	Declaração	Ex. Entrada	Ex. Armazenamento
	number (*,1)	7,456,123.89	7,456,123.9
	number (9,2)	7,456,123.89	7,456,123.89
	number (9,1)	7,456,123.89	7,456,123.9
	number (6)	7,456,123.89	ERRO. Excede a precisão

INT INTEGER	Ex. Entrada	Ex. Armazenamento
	123	123
	123456789	123456789
	123.2	ERRO.

FLOAT	Ex. Entrada	Ex. Armazenamento
	123.4	123.4
	123.45	123.45
	123	123.0

Tipos de Dados

DATA

	FORMATO	Ex. Entrada
DATE	yyyy-mm-dd mm/dd/yyyy dd.mm.yyyy	'2024-02-28'
		'12/25/2022'
		'08.03.2024'
		'2024-08-06'

HORA

	FORMATO	Ex. Entrada
TIME	hh:mm[:ss]	'15:09:02'
	hh.mm[.ss]	'15.09.02'

DATA + HORA

	FORMATO	Ex. Entrada
TIMESTAMP	yyyy-mm-dd hh:mm:ss[.nnnnnnnnnn] yyyy-mm-dd-hh.mm.ss[.nnnnnnnnnn]	'1960-01-01 23:03:20'
		'1962-09-23 03:23:34.234'
		'1962-09-23-03:23:34.987654321'



Structured Query Language

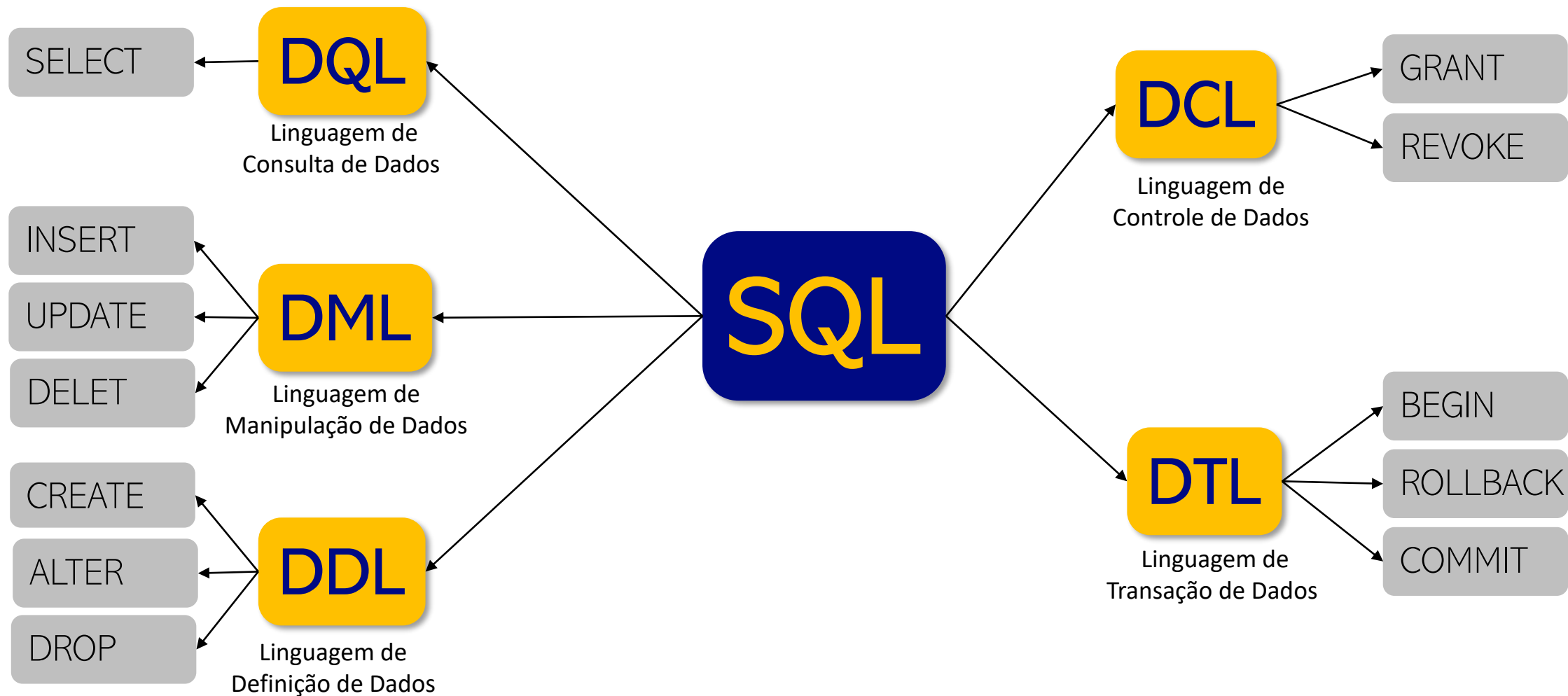


Structured Query Language

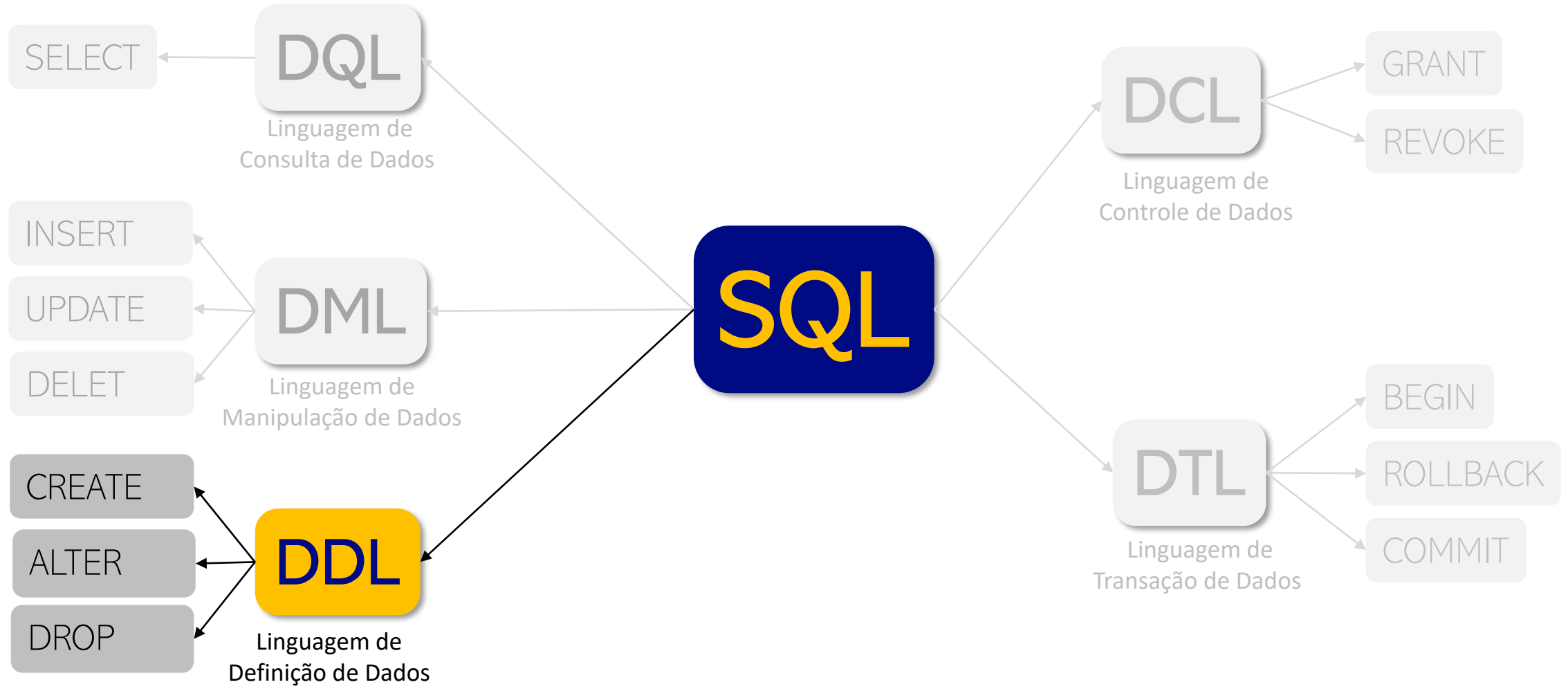
Linguagem de Consulta Estruturada

O SQL é uma linguagem padrão para trabalhar com bancos de dados relacionais, amplamente utilizada por profissionais em diversas áreas, desde cientistas de dados até pessoas usuárias de Excel.

Organização SQL



Na aula de hoje...



Criação de Tabela – CREATE TABLE

Sintaxe básica

```
CREATE TABLE table_name  
(  
    coluna1 datatype,  
    coluna2 datatype,  
    ...  
    coluna $n$  datatype  
);
```

Criação Tabela – Regras de Nomenclatura

Nomes de tabelas e colunas

- Devem começar com uma letra
- Devem ter de 1 a 30 caracteres é tamanho
- Devem conter somente A-Z, a-z, 0-9,_, \$, e #
- Não pode ser um nome duplicado de outro objeto de propriedade do mesmo usuário (mesmo schema)
- Não pode ser uma palavra reservada do SGBD

ATIVIDADE

Nome da Conexão: < um nome a sua escolha >
Nome do usuário: <usuário> Senha: *****
Tipo de conexão: Básica Atribuição: Padrão
Nome do host ou String do Host: BD-ACD
Porta: 1521
SID: XE

No SQL Developer
(para quem usar computador da PUC)

No SQL Developer:

Crie uma tabela **ALUNOS**
com os campos **RA,**
nome, idade e cidade.

Não se esqueça de colocar o tipo de
dado adequado para cada atributo.



PROBLEMA

Imagine que você esqueceu de adicionar um ou mais campos na tabela ALUNOS.

Por exemplo, esqueceu de adicionar o campo "Estado"

Alteração de Tabela – ALTER TABLE

Comando utilizado para adicionar, modificar, deletar informações de um banco de dados.

Utilize o comando ALTER TABLE para:

- Adicionar uma nova coluna (ADD)
- Modificar uma nova coluna (MODIFY)
- Definir um valor default para uma nova coluna (DEFAULT)
- Remover uma coluna (DROP [*column*])
- Renomear uma coluna (RENAME COLUMN)

Alteração de Tabela – ALTER TABLE

Comando utilizado para adicionar, modificar, deletar informações de um banco de dados.

Utilize o comando ALTER TABLE para	Sintaxe
Adicionar uma nova coluna	ALTER TABLE table_name ADD column datatype
Modificar uma nova coluna	ALTER TABLE table_name MODIFY column [datatype]
Definir um valor default para uma nova coluna	ALTER TABLE table_name ALTER column SET DEFAULT
Remover uma coluna	ALTER TABLE table_name DROP (column)
Renomear uma coluna	ALTER TABLE table_name RENAME COLUMN old_name to new_name;

ATIVIDADE



No SQL Developer:

Adicione o campo UF na
tabela ALUNOS.

No SQL Developer:

Na tabela **ALUNO**, altere o nome do campo **UF** para **ESTADO** e faça outras modificações para que agora ele possa ser preenchido por extenso.

No SQL Developer:

Exclua o campo **IDADE**
da tabela **ALUNOS**.

No SQL Developer:

Exclua a tabela **ALUNO**.

DROP

- Remove a tabela e coloca a tabela na lixeira (recycle bin)
- Se você desejar remover a tabela e seus dados definitivamente (sem colocar na lixeira) utilize a clausula **PURGE** → **PURGE TABLE table_name;**





PROBLEMA

Imagine que você precise que todos os campos sejam preenchidos obrigatoriamente no momento da inserção dos dados no BD e que o campo RA seja único para cada registro.

Como ter essa garantia?

Restrições - SQL Constraints

- São regras aplicadas nas colunas de uma tabela
- São usadas para limitar os tipos de dados que são inseridos
- Podem ser especificadas no momento da criação de uma tabela ou após

- As principais:
 - NOT NULL
 - PRIMARY KEY (PK)
 - FOREIGN KEY (FK)

Restrição – NOT NULL

- A constraint NOT NULL impõe a uma coluna a NÃO aceitar valores nulos.
- Obriga um campo a sempre possuir um valor.
- Não é possível inserir um registro (ou atualizar) sem entrar com um valor neste campo.

Opção 1: no momento da criação da tabela, especificar qual ou quais campos terão a restrição.

```
CREATE TABLE table_name (  
    coluna1 datatype NOT NULL  
);
```

Opção 2: se a tabela já existe, utilize a instrução ALTER TABLE com MODIFY

```
ALTER TABLE table_nome  
MODIFY column NOT NULL;
```

Restrição – PRIMARY KEY (PK)

- Identifica de forma única cada registro em uma tabela de banco de dados.
- Chaves Primárias devem conter valores únicos.
- Uma coluna de chave primária não pode conter valores NULL.
- Cada tabela deve ter uma chave primária.

CPF	Nome
12345678999	Harry Potter
23456789111	Hermione Granger
34567891222	Rony Weasley

Id_jogo	Nome
1	Hollow knight
2	Ori and the Will of the wisps
3	Prince of Persia: The lost Crown
4	Mario Kart 8 Deluxe

Restrição – PRIMARY KEY (PK)

Opção 1: no momento da criação da tabela, especificar qual ou quais campos terão a restrição.

```
CREATE TABLE table_name (  
    coluna1 datatype,  
    coluna2 datatype,  
    CONSTRAINT nome_constraint PRIMARY KEY (coluna)  
);
```

Exemplo

```
CREATE TABLE teste(  
    cod_teste number(3),  
    nome varchar2 (10),  
    CONSTRAINT pk_teste PRIMARY KEY (cod_teste)  
);
```

Restrição – PRIMARY KEY (PK)

Opção 2: use a instrução ALTER TABLE.

```
ALTER TABLE table_name  
ADD CONSTRAINT nome_constraint PRIMARY KEY (coluna)
```

Exemplo

```
ALTER TABLE teste  
ADD CONSTRAINT pk_teste PRIMARY KEY (cod_teste);
```

Restrição – FOREIGN KEY (FK)

- Utilizada para estabelecer o relacionamento entre tabelas.
- Chaves Estrangeiras são chaves primárias em outras tabelas.
- Uma coluna de chave estrangeira não pode conter valores NULL.

Id_jogo	Nome	Gender	Id_gender	Nome
1	Hollow knight	1	1	Metroidvania
2	Ori and the Will of the wisps	1	2	Race
3	Prince of Persina: The lost Crown	1	3	Roguelike
4	Mario Kart 8 Deluxe	2		

Uma chave estrangeira só pode ser utilizada se ela já existe como chave primária – Restrição de Integridade Referencial



Restrição – FOREIGN KEY (FK)

Opção 1: no momento da criação da tabela, especificar qual ou quais campos terão a restrição.

```
CREATE TABLE table_name (  
    coluna1 datatype,  
    coluna2 datatype,  
    CONSTRAINT nome_constraint FOREIGN KEY (coluna) REFERENCES  
table_origem (coluna_origem)  
);
```

Exemplo

```
CREATE TABLE jogos(  
    id_jogo number(3),  
    nome varchar2 (30),  
    gender number (3)  
    CONSTRAINT fk_gender FOREIGN KEY (Gender) REFERENCES  
GENDER (id_gender)  
);
```


Restrição – FOREIGN KEY (FK)

Opção 1: no momento da criação da tabela, especificar qual ou quais campos terão a restrição.

```
CREATE TABLE table_name (  
    coluna1 datatype,  
    coluna2 datatype,  
    CONSTRAINT nome_constraint FOREIGN KEY (coluna) REFERENCES table_origem (coluna_origem)  
);
```

JOGOS

Id_jogo	Nome	Gender
1	Hollow knight	1
2	Ori and the Will of the wisps	1
3	Prince of Persina: The lost Crown	1
4	Mario Kart 8 Deluxe	2

GENERO

Id_gender	Nome
1	Metroidvania
2	Race
3	Roguelike

```
CREATE TABLE jogos(  
    id_jogo number(3),  
    nome varchar2 (30),  
    gender number (3)  
    CONSTRAINT fk_gender FOREIGN KEY (Gender) REFERENCES  
    GENERO (id_gender)  
);
```

Restrição – FOREIGN KEY (PK)

Opção 2: use a instrução ALTER TABLE.

ALTER TABLE table_name

ADD CONSTRAINT nome_constraint FOREIGN KEY (coluna) REFERENCES
table_origem (coluna_origem)

JOGOS

Id_jogo	Nome	Gender
1	Hollow knight	1
2	Ori and the Will of the wisps	1
3	Prince of Persina: The lost Crown	1
4	Mario Kart 8 Deluxe	2

GENERO

Id_gender	Nome
1	Metroidvania
2	Race
3	Roguelike

```
CREATE TABLE jogos(  
    id_jogo number(3),  
    nome varchar2 (30),  
    gender number (3)  
);
```



```
ALTER TABLE jogos  
ADD CONSTRAINT fk_gender FOREIGN KEY (Gender) REFERENCES  
GENERO (Id_gender);
```

ATIVIDADE



No SQL Developer:

Adicione a restrição de
chave primária ao
atributo RA da tabela
ALUNOS e a de *not null*
no atributo nome da
mesma tabela.