



SHIFT



FIAP





# MASTERING C#

CODE EVERYTHING

SHIFT  FIAP

# BANCO DE **DADOS**

---



# BANCO DE DADOS

---

1

## AULA 1

CONTEÚDO – Introdução ao Contexto de Dados e à Linguagem SQL

2

## AULA 2

CONTEÚDO – Criação de Objetos (DDL)

3

## AULA 3

CONTEÚDO – Alteração e Exclusão de Objetos (DDL)

4

## AULA 4

CONTEÚDO – Inserção, Atualização e Exclusão de Dados (DML)

# BANCO DE DADOS

---

5

**AULA 5**

CONTEÚDO – Seleção e Manipulação de Dados (DML)

6

**AULA 6**

CONTEÚDO – Seleção e Junção de Dados (JOIN)

7

**AULA 7**

CONTEÚDO – Funções e Cálculos

8

**AULA 8**

CONTEÚDO – Funções e Agrupamentos

## AULA 2

# CRIAÇÃO DE OBJETOS DDL (DATA DEFINITION LANGUAGE)

---



Tipos de Objetos



---

Tipos de Dados

---

Criação de Tabelas e Constraints

---

Exercícios

# TIPOS DE **OBJETOS**





NOME	NASCIMENTO	SEXO	ÓBITO
STEPHEN HAWKING	08/01/1942	M	14/03/2018
ALBERT EINSTEIN	14/03/1879	M	18/04/1955
ISAAC NEWTON	04/01/1643	M	31/03/1727
GALILEU GALILEI	15/02/1564	M	08/01/1642
NICOLAU COPERNICO	19/02/1473	M	21/05/1543



# SCHEMA OBJECTS (“OBJETOS DE ESQUEMA”)

OBJETO	DESCRIÇÃO
TABLES (TABELAS)	Objetos que armazenam dados distribuídos em linhas e colunas.
INDEXES (ÍNDICES)	Estruturas opcionais e lógicas (fisicamente independentes dos dados aos quais estão associadas) que podem acelerar o acesso às linhas da tabela.
PARTITIONS (PARTIÇÕES)	Permitem decompor as tabelas e os índices que são muito grandes em partes menores, facilitando o gerenciamento dos objetos e melhorando a performance das consultas.
VIEWS (VISUALIZAÇÕES)	São representações lógicas de uma ou mais tabelas, em resumo, são consultas armazenadas.
SEQUENCE (SEQUÊNCIAS)	Objetos que geram números inteiros e sequenciais. Normalmente, são utilizados para gerar valores para campos de chave primária.
PL/SQL	PL (procedural language) é uma linguagem de extensão do SQL para o banco Oracle.

# INTEGRITY CONSTRAINTS (“RESTRIÇÕES DE INTEGRIDADE”)

OBJETO	DESCRIÇÃO
NOT NULL (NÃO NULO)	<b>Proíbe</b> inserções ou atualizações de linhas contendo um <b>valor nulo</b> (ausência de valor) em uma <b>coluna</b> especificada.
PRIMARY KEY (CHAVE PRIMÁRIA)	<b>Garante</b> a <b>unicidade das linhas</b> , em uma determinada coluna ou em uma combinação de colunas, e <b>restringe</b> a inserção de <b>valor nulo</b> .
PRIMARY KEY (CHAVE PRIMÁRIA)	<b>Impede</b> a ocorrência de <b>valores repetidos</b> , no entanto, <b>não proíbe</b> a inserção de valor <b>nulo</b> .
FOREIGN KEY (CHAVE ESTRANGEIRA)	<b>Garante</b> a <b>integridade referencial</b> dos <b>dados</b> , ou seja, obriga o relacionamento entre duas ou mais tabelas que possuem uma ou mais colunas em comum.
CHECK (VERIFICAÇÃO)	<b>Expressão</b> lógica para <b>verificação</b> de regra específica (pode ser comparada à condicional “IF”)

# TIPOS DE **DADOS**



# TIPOS BÁSICOS DE DADOS

Os **tipos de dados seguem** um **padrão** do Instituto Nacional Americano de Padrões (American National Standard Institute, **ANSI**), entretanto podem ocorrer variações na capacidade de armazenamento e precisão de um SGBD para outro.

TIPO	DESCRIÇÃO
CHARACTERS (ALFANUMÉRICOS)	Armazenam dados <b>alfanuméricos</b> (letras, números e símbolos especiais) em strings. Exemplo: <b>VARCHAR2</b> .
NUMERICS (NUMÉRICOS)	Armazenam dados <b>numéricos</b> (inteiros e reais). Exemplo: <b>NUMBER</b> .
DATES (DATAS E HORAS)	Armazenam dados <b>data</b> e <b>hora</b> . Exemplo: <b>DATE</b> .

# PRINCIPAIS TIPOS DE DADOS NO ORACLE

FAMÍLIA	TIPO	DESCRIÇÃO
CARACTER	CHAR (tamanho)	Dados de <b>caractere</b> de <b>comprimento fixo</b> . O tamanho default mínimo de caracteres é 1 e o máximo 2.000.
CARACTER	VARCHAR2 (tamanho)	Dados de <b>caractere</b> de <b>comprimento variável</b> . O tamanho deve ser especificado, sendo o dado mínimo de caractere 1 e o máximo 4.000.
NUMÉRICO	NUMBER (p,s)	Dados <b>numéricos</b> de comprimento variável, em que <b>p</b> representa a quantidade de números <b>inteiros</b> e <b>s</b> , a parte correspondente aos <b>decimais</b> . O tamanho (p) pode variar entre 1 a 38 dígitos e (s) -84 a 127.
DATA	DATE	O tipo DATE armazena informações de <b>data</b> e <b>hora</b> . Embora estes dados possam ser representados em tipos de caracteres e em tipos numéricos, o DATE possui propriedades adicionais: para cada valor DATE, o Oracle armazena toda a estrutura de data/hora ( <b>ano, mês, dia, hora, minuto e segundo</b> ).
DATA	TIMESTAMP	É uma extensão do tipo DATE que armazena dados de <b>tempo com frações de segundo</b> (podendo ser entre 0 e 9).

## CHAR X VARCHAR

---

Exemplo de um campo com tamanho **fixo**, do tipo **CHAR**, limitado a 20 posições, ou 20 bytes: nome\_materia char(20)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
B	A	N	C	O		D	E		D	A	D	O	S						



## CHAR X VARCHAR

---

Exemplo de um campo com tamanho *variável*, do tipo *VARCHAR*, limitado a 20 posições, ou 20 bytes: `nome_materia varchar(20)`

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
B	A	N	C	O		D	E		D	A	D	O	S						







# CRIAÇÃO DE **TABELAS E CONSTRAINTS**

# REGRAS PARA NOMES DE OBJETOS

---

Os nomes de objetos no SGBD Oracle devem respeitar algumas regras:

- No máximo 128 bytes para tabelas (bancos 30 bytes);
- Palavras reservadas (como SELECT, CREATE, UPDATE, TABLE, etc) não podem ser usadas como nomes de objetos;
- Todos os nomes devem começar com uma letra de A a Z;
- Os nomes podem conter letras, números, sublinhado (\_), cifrão (\$) ou o símbolo de hash (#);
- Letras minúsculas são convertidas para letras maiúsculas automaticamente durante a criação do objeto.



# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**SEM CONSTRAINT**



## O COMANDO CREATE

---

- Por meio do comando CREATE, é possível criar diversos objetos como: tabelas, views, usuários, procedures, triggers, entre outros.
- O comando CREATE deve ser sempre acompanhado do tipo de objeto seguido do seu nome.
- Para criar um objeto do tipo tabela, é preciso listar entre parênteses todas as suas colunas, seguidas pelos seus tipos de dados.

**CREATE TABLE**

**nome\_tabela (**

**nome\_campo TIPO,**

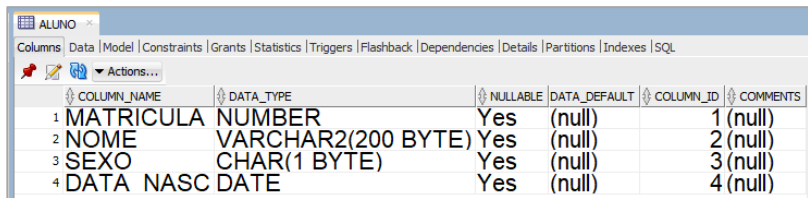
**nome\_campo TIPO,**

**nome\_campo TIPO**

**);**

FONTE: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/CREATE-TABLE.html#>

## EXEMPLO DE CRIAÇÃO DE TABELA



The screenshot shows the 'ALUNO' table structure in Oracle SQL Developer. The table has four columns: MATRICULA (NUMBER), NOME (VARCHAR2(200 BYTE)), SEXO (CHAR(1 BYTE)), and DATA\_NASC (DATE). All columns are nullable and have default values of null. The column IDs are 1, 2, 3, and 4 respectively.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MATRICULA	NUMBER	Yes	(null)	1	(null)
2	NOME	VARCHAR2(200 BYTE)	Yes	(null)	2	(null)
3	SEXO	CHAR(1 BYTE)	Yes	(null)	3	(null)
4	DATA_NASC	DATE	Yes	(null)	4	(null)

FONTE: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/CREATE-TABLE.html#>

```
CREATE TABLE  
ALUNO (  
  
    MATRICULA  
  
    NUMBER,  
  
    NOME  
  
    VARCHAR2(200),  
  
    SEXO CHAR(1),  
  
    DATA_NASC DATE  
  
);
```

# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**CONSTRAINT NOT NULL**



## RESTRIÇÃO: NOT NULL

---

- Nulo ou null é a ausência de valor e ocorre quando um campo não é preenchido, isto é, quando não há dado dentro dele.
- A restrição NOT NULL assegura que nulos não sejam permitidos na coluna, determinando o preenchimento obrigatório dos campos.
- Colunas sem uma restrição NOT NULL podem não conter valores.

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER NOT  
    NULL,  
    NOME VARCHAR2(200)  
    NOT NULL,  
    SEXO CHAR(1),  
    DATA_NASC DATE  
);
```

FONTE: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/constraint.html#>

## NOMEANDO UMA RESTRIÇÃO NOT NULL

---

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER CONSTRAINT NN_ALUNO_MATRICULA  
    NOT NULL,  
    NOME VARCHAR2(200) CONSTRAINT NN_ALUNO_NOME  
    NOT NULL,  
    SEXO CHAR(1),  
    DATA_NASC DATE  
);
```





## RESTRIÇÃO: NOT NULL

---

- A definição NOT NULL deve ser, obrigatoriamente, definida no nível da coluna.
- A alteração de uma coluna para NOT NULL, em uma tabela existente, só será realizada se a tabela não tiver nenhuma linha inserida ou se todas elas estiverem populadas (preenchidas).



# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**CONSTRAINT UNIQUE KEY**



## RESTRIÇÃO: UNIQUE KEY (NÍVEL DA COLUNA)

- Uma restrição de integridade unique key **impõe** que **cada valor seja exclusivo**, ou seja, duas linhas de uma tabela não podem ter valores duplicados em uma coluna específica ou em um conjunto de colunas.
- A coluna ou a chave incluída na definição de restrição unique key é chamada chave exclusiva. Se a chave unique contiver mais de uma coluna, tal grupo de colunas é considerado uma chave exclusiva composta. A restrição unique key **pode ser definida** no **nível da coluna** ou no da **tabela**.

```
CREATE TABLE MATERIA (  
  
    CODIGO NUMBER NOT NULL,  
  
    DESCRICAO VARCHAR2(200)  
  
    UNIQUE  
  
);
```

FONTE: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/constraint.html#>

## NOMEANDO UMA RESTRIÇÃO UNIQUE KEY

---

```
CREATE TABLE MATERIA (  
    CODIGO NUMBER CONSTRAINT NN_MATERIA_CODIGO  
    NOT NULL,  
    DESCRICAO VARCHAR2(200) CONSTRAINT  
    UK_MATERIA_DESCRICAO UNIQUE  
);
```



## RESTRIÇÃO UNIQUE KEY (NÍVEL DA TABELA)

---

```
CREATE TABLE MATERIA (  
    CODIGO NUMBER CONSTRAINT NN_MATERIA_CODIGO NOT NULL,  
    DESCRICAO VARCHAR2(200),  
    CONSTRAINT UK_MATERIA_DESCRICAO UNIQUE(DESCRICAO)  
);
```

Obs.: Ao criar uma restrição no nível da tabela, o nome da constraint se torna obrigatório.



# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**CONSTRAINT PRIMARY KEY**



## RESTRIÇÃO: PRIMARY KEY (NÍVEL DA COLUNA)

- A chave primária é uma **coluna**, ou um **conjunto** de **colunas**, que **identifica exclusivamente cada registro** de uma **tabela**.
- Essa **restrição** também assegura que nenhuma coluna que seja parte da chave primária seja **nula**.
- A PK pode ser definida tanto no **nível** da **tabela** como no da **coluna**.

```
CREATE TABLE MATERIA (  
  
    CODIGO NUMBER PRIMARY KEY,  
  
    DESCRICAO VARCHAR2(200)  
  
    UNIQUE  
  
);
```

## NOMEANDO UMA RESTRIÇÃO PRIMARY KEY

---

**CREATE TABLE MATERIA (**

**CODIGO** **NUMBER** **CONSTRAINT PK\_MATERIA\_CODIGO**  
**PRIMARY KEY,**

**DESCRICAO** **VARCHAR2(200)** **CONSTRAINT**  
**UK\_MATERIA\_DESCRICAO** **UNIQUE**

**);**





## RESTRIÇÃO UNIQUE KEY (NÍVEL DA TABELA)

---

```
CREATE TABLE MATERIA (  
  
    CODIGO NUMBER,  
  
    SIGLA CHAR(2),  
  
    DESCRICAO VARCHAR2(200) UNIQUE,  
  
    CONSTRAINT PK_MATERIA_CODIGO_SIGLA PRIMARY KEY  
(CODIGO, SIGLA)  
  
);
```

Obs.: Ao criar uma restrição no nível da tabela, o nome da constraint se torna obrigatório. Fora isso, toda chave primária composta deve ser criada no nível da tabela.



# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**CONSTRAINT CHECK**



## RESTRIÇÃO: CHECK (NÍVEL DA COLUNA)

- Define uma **condição** que deve ser atendida a **cada registro** de uma tabela, isto é, os dados que serão inseridos na coluna devem pertencer a um conjunto de valores predefinido.
- Podem ser utilizados **operadores de comparação** (>, <, >=, <=, IN, NOT IN, NOT BETWEEN, BETWEEN, = e <>).
- Essa restrição **pode ser definida** no **nível da coluna** ou no da **tabela**.

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER,  
    NOME VARCHAR2(30),  
    SEXO CHAR(1) CHECK (SEXO IN  
        ('M', 'F'))  
);
```

## NOMEANDO UMA RESTRIÇÃO CHECK

---

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER,  
    NOME VARCHAR2(30),  
    SEXO CHAR(1) CONSTRAINT CK_ALUNO_SEXO CHECK  
    (SEXO IN ('M', 'F'))  
);
```

Obs.: Embora o Oracle não seja case sensitive em sua sintaxe, os valores 'm' e 'f' não serão aceitos.



## RESTRIÇÃO CHECK (NÍVEL DA TABELA)

---

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER,  
    NOME VARCHAR2(30),  
    SEXO CHAR(1),  
    CONSTRAINT CK_ALUNO_SEXO CHECK  
    (SEXO IN ('M', 'F'))  
);
```

Obs.: Embora o Oracle não seja case sensitive em sua sintaxe, os valores 'm' e 'f' não serão aceitos.



# DATA DEFINITION **LANGUAGE**

---

SINTAXE DE CRIAÇÃO DE TABELA  
**CONSTRAINT FOREIGN KEY**



## RESTRIÇÃO: FOREIGN KEY (NÍVEL DA COLUNA)

- Restrição de **integridade referencial**, que designa uma **coluna** ou uma combinação de **colunas** como **chave estrangeira** e estabelece um **relacionamento** entre a **chave primária**, ou uma chave exclusiva, na mesma tabela ou em uma tabela diferente.
- O valor de chave estrangeira deve corresponder ao valor existente na tabela-mãe.
- As chaves estrangeiras são baseadas nos valores dos dados, sendo uma relação lógica e não física. Essa restrição pode ser definida no **nível** da **coluna** ou no da **tabela**.

```
CREATE TABLE ALUNO (  
    MATRICULA NUMBER PRIMARY KEY,  
    NOME VARCHAR2(30),  
    SEXO CHAR(1) CHECK (SEXO IN ('M', 'F'))  
);  
  
CREATE TABLE CELULAR (  
    COD_CELULAR NUMBER PRIMARY KEY,  
    NUMERO VARCHAR2(10) UNIQUE,  
    MATRICULA NUMBER(3) REFERENCES ALUNO  
    (MATRICULA)  
);
```

FONTE: <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/constraint.html#>

## NOMEANDO UMA RESTRIÇÃO FOREIGN KEY

---

```
CREATE TABLE CELULAR (  
    COD_CELULAR NUMBER CONSTRAINT PK_CELULAR_COD_CELULAR PRIMARY KEY,  
    NUMERO VARCHAR2(10) CONSTRAINT UK_CELULAR_NUMERO UNIQUE,  
    MATRICULA NUMBER(3)  
    CONSTRAINT FK_CELULAR_ALUNO_MATRICULA REFERENCES ALUNO  
    (MATRICULA)  
);
```

Obs.: Lembrando que a tabela ALUNO deve ser criada antes da CELULAR.





## RESTRIÇÃO: FOREIGN KEY (NÍVEL DA TABELA)

---

```
CREATE TABLE CELULAR (  
    COD_CELULAR NUMBER PRIMARY KEY,  
    NUMERO VARCHAR2(10) UNIQUE,  
    CPF VARCHAR2(14),  
    MATRICULA NUMBER(3),  
    CONSTRAINT FK_CELULAR_ALUNO_CPF_MATRICULA  
    FOREIGN KEY (CPF, MATRICULA)  
    REFERENCES ALUNO (CPF, MATRICULA)  
);
```

Obs.: Ao criar uma restrição no nível da tabela, o nome da constraint se torna obrigatório. Fora isso, toda chave estrangeira composta deve ser criada no nível da tabela.



# EXERCÍCIOS



# DDL: CRIAÇÃO DE CONSTRAINTS



## DDL: CRIAÇÃO DE CONSTRAINTS

---

Criar as estruturas das tabelas no SQL Developer.

1. Criar as chaves primárias e estrangeiras.
2. Os campos PRODUTO, CATEGORIA e MATERIA\_PRIMA devem ser únicos.
3. O campo QUANTIDADE não pode aceitar valor nulo.
4. A CATEGORIA deve ser criada para receber valores restritos a DOCE, BEBIDA e CONGELADO.
5. Trazer as regras de restrições da tabela CATEGORIA.



## BIBLIOGRAFIA BÁSICA

---

- **TAYLOR, A.** SQL para Leigos. 2016.
- **PUGA, S.; FRANÇA, E.; GOYA, M..** Banco de Dados: Implementação em SQL, PL/SQL e Oracle 11g. 2013.
- **ORACLE LIVE SQL.** Learn and share SQL: Running on Oracle Database 19c. Disponível em: <<https://livesql.oracle.com/apex/f?p=590:1000>>.
- **ORACLE HELP CENTER.** Oracle Database 19c Get Started. Disponível em: <<https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>>



# OBRIGADO



in /alexandrebarcelos

## FIAP

Copyright © 2021 | Professor MSc. Alexandre Barcelos

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



SHIFT



FIAP



00000