

FIAP



AULA 03

INTRODUÇÃO A LINGUAGEM DE
DEFINIÇÃO DE DADOS (DDL) - CREATE

Welcome to the next evolution in higher education.

☐ Linguagem de Definição de Dados (DDL)

☐ Sobre a DDL

☐ Criar Objetos (SQL Worksheet)

☐ Criar tabelas

☐ O que é CONSTRAINT?

☐ Criar CONSTRAINT NOT NULL

☐ Criar CONSTRAINT UNIQUE KEY

☐ Criar CONSTRAINT PRIMARY KEY

☐ Criar CONSTRAINT FOREIGN KEY

☐ Criar CONSTRAINT CHECK

☐ Criar Objetos (Programação Visual)

☐ Exercício Prático

| OBJETIVO

Introduzir conceitos iniciais do SQL/DDL

Aplicar os conceitos no Oracle SQL Developer

CRIAR OBJETOS

SQL Worksheet

- ***Data Definition Language (DDL)*** – Linguagem de Definição de Dados , expressa um conjunto de comandos que especificam a:
 - **CRIAR** estruturas de dados (CREATE)
 - **ALTERAR** estruturas de dados (ALTER)
 - **REMOVER** estruturas de dados (DROP)
- Os comandos da DDL viabiliza criar as estruturas de armazenamento que foram projetados na Modelagem Conceitual/Lógica.
- A DDL poderá ainda prover as restrições de integridade, domínio e referencial

I COMANDO: CREATE

- O comando **CREATE** é utilizado para criar diversos objetos em um banco de dados, como **tabelas, views, usuários, procedures, triggers**, entre outros;
- Para usar o comando CREATE, é necessário especificar o tipo de objeto que se deseja criar, seguido do seu nome e dos seus atributos;
- Veja o exemplo da Sintaxe para Criar uma Tabela

SINTAXE

```
CREATE TABLE  
    nome_tabela (  
        nome_campo TIPO(tamanho),  
        nome_campo TIPO(tamanho)  
    );
```

EXEMPLO CREATE TABLE

CRIAÇÃO DE TABELA

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1	(null)
2	NOME	VARCHAR2(60 BYTE)	Yes	(null)	2	(null)
3	DT_NASCIMENTO	DATE	Yes	(null)	3	(null)

COMANDO

```
CREATE TABLE  
  
    ALUNO (  
  
        ra int,  
  
        nome varchar(60),  
  
        dt_nascimento date  
  
    );
```


I O QUE É **CONSTRAINT**?

- Uma **CONSTRAINT** é um tipo de restrição (ou regras) que podem ser aplicadas nas colunas de uma determinada tabela.
- Podem ser especificadas no momento de criação da tabela (CREATE) ou após a tabela ter sido criada (ALTER)
- As principais **CONSTRAINTs** são as seguintes:
 - **NOT NULL**
 - **UNIQUE**
 - **PRIMARY KEY**
 - **FOREIGN KEY**

- Em alguns SGBDR não há necessidade em definir a palavra reservada **CONSTRAINT** antes da restrição imposta. Por exemplo:
 - O MySQL ou PostgreSQL, pode definir as restrições diretamente na declaração da tabela;
 - O ORACLE já solicita a palavra reservada **CONSTRAINT** seguida por um nome opcional – principalmente em chaves primárias e estrangeiras compostas;
- É importante consultar a documentação específica de cada SGDBR para entender a sintaxe correta para utilizar uma **CONSTRAINT** ou não

I RESTRIÇÃO: **NOT NULL**

- Uma coluna que possui a restrição **NOT NULL** não pode ter **valores nulos**, ou seja, valores que indicam a ausência de dados.
- **Essa restrição garante** que todos os registros da coluna tenham **algum valor preenchido**, evitando inconsistências ou ambiguidades na base de dados.
- **Colunas que não** têm a restrição **NOT NULL** podem aceitar valores nulos, o que significa que **o dado pode ser omitido** ou desconhecido.

SINTAXE

```
CREATE TABLE  
    NOME_TABELA (  
        nome_campo TIPO(tamanho) NOTNULL,  
        nome_campo TIPO(tamanho)  
    );
```

I RESTRIÇÃO: **NOT NULL**

- A definição **NOT NULL** só pode ser definida em colunas
- **A alteração** de uma coluna para NOT NULL, em uma tabela criada, é possível quando a tabela não tiver nenhuma linha inserida OU se todas as colunas estiverem preenchidas (ou populadas)

SINTAXE

```
CREATE TABLE  
    NOME_TABELA (  
        nome_campo TIPO(tamanho) NOTNULL,  
        nome_campo TIPO(tamanho)  
    );
```

EXEMPLO NOT NULL

NÃO VAZIO

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1	(null)
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2	(null)
3	DT_NASCIMENTO	DATE	Yes	(null)	3	(null)

The screenshot shows the Oracle SQL Developer interface for the 'ALUNO' table. The 'Constraints' tab is selected, displaying a list of constraints. The first constraint, 'SYS_C008315', is a 'Check' constraint with the condition '"NOME" IS NOT NULL'. The second constraint, 'SYS_C008316', is a 'Primary_Key' constraint. Below the constraints, the 'Colunas' (Columns) tab is visible, showing the table's structure with columns 'RA', 'NOME', and 'DT_NASCIMENTO'.

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_C
1 SYS_C008315	Check	"NOME" IS NOT NULL	(nul
2 SYS_C008316	Primary_Key	(null)	(nul

COLUMN_NAME	COLUMN_POSITION
1 NOME	(null)

```
CREATE TABLE ALUNO(  
    ra int,  
    nome varchar2(60) NOT NULL,  
    data_nascimento date,  
);
```

EXEMPLO NOT NULL

NÃO VAZIO

- A definição **NOT NULL** só pode ser definida em **colunas**
- **A alteração** de uma coluna para NOT NUL, em uma tabela criada, é possível quando a tabela não tiver nenhuma linha inserida OU se todas as colunas estiverem preenchidas (ou populadas)

FIAP

COMANDO

```
CREATE TABLE  
    ALUNO (  
        ra int,  
        nome varchar(60) CONSTRAINT nn_nome NOTNULL,  
        data_nascimento date  
    );
```

I RESTRIÇÃO: **UNIQUE KEY**

- Uma restrição de integridade **UNIQUE KEY** garante que cada valor em uma coluna ou em um conjunto de colunas seja único, ou seja, não existam duas linhas na mesma tabela com valores iguais nessas colunas;
- A coluna ou o conjunto de colunas que faz parte da definição da restrição **UNIQUE KEY** é chamado de chave exclusiva. Se a chave exclusiva tiver mais de uma coluna, ela é chamada de chave exclusiva composta;
- A restrição **UNIQUE KEY** pode ser aplicada tanto no nível da coluna quanto no da tabela.

SINTAXE

```
CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(TAMANHO) UNIQUE
    );

-- OU

CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(TAMANHO),
        UNIQUE (nome_campo)
    );
```

EXEMPLO UNIQUE KEY

CHAVE ÚNICA

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1	(null)
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2	(null)
3	CPF	NUMBER(11,0)	Yes	(null)	3	(null)
4	DT_NASCIMENTO	DATE	Yes	(null)	4	(null)

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OV
1	SYS_C008319	Check	"NOME" IS NOT NULL	(null)
2	SYS_C008320	Unique	(null)	(null)

Colunas

Atualizar: 0

	COLUMN_NAME	COLUMN_POSITION
1	CPF	1

```
CREATE TABLE ALUNO(  
    ra int,  
    nome varchar2(60) NOT NULL,  
    cpf int(11) UNIQUE,  
    data_nascimento date,  
);
```


EXEMPLO UNIQUE KEY

CHAVE ÚNICA

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER (38, 0)	Yes	(null)	1	(null)
2	NOME	VARCHAR2 (60 BYTE)	No	(null)	2	(null)
3	CPF	NUMBER (11, 0)	Yes	(null)	3	(null)
4	DT_NASCIMENTO	DATE	Yes	(null)	4	(null)

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER
1	SYS_C008337	Check	"NOME" IS NOT NULL	(null)
2	UK_CPF	Unique	(null)	(null)

Colunas

Atualizar: 0

	COLUMN_NAME	COLUMN_POSITION
1	CPF	1

```
CREATE TABLE ALUNO(  
    ra int,  
    nome varchar2(60) NOT NULL,  
    cpf int(11),  
    data_nascimento date,  
    CONSTRAINT uk_cpf UNIQUE (cpf)  
);
```

I RESTRIÇÃO: **PRIMARY KEY**

- Uma **PRIMARY KEY (PK)** é um atributo, ou um conjunto de atributos, **que distingue cada linha de uma tabela**.
- Essa restrição também garante que nenhum atributo que faça parte da **PK** seja nulo.
- A **PK** pode ser especificada tanto no nível da tabela quanto no do atributo.
- Cada tabela deve ter uma chave primária e apenas uma chave primária.

SINTAXE

```
CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(tamanho) PRIMARY KEY
    );

-- OU

CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(tamanho),
        PRIMARY KEY (nome_campo)
    );
```

EXEMPLO PRIMARY KEY

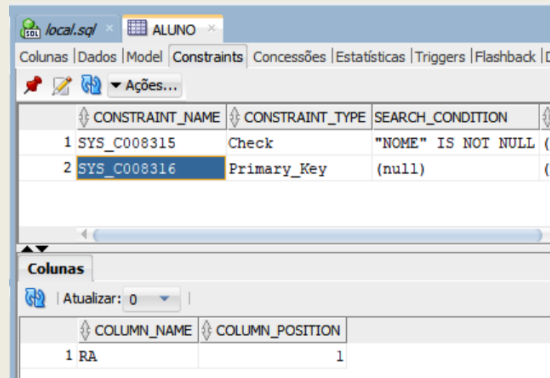
CHAVE PRIMÁRIA

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	No	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	DT_NASCIMENTO	DATE	Yes	(null)	3 (null)	



CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 SYS_C008315	Check	"NOME" IS NOT NULL (
2 SYS_C008316	Primary_Key	(null) (

COLUMN_NAME	COLUMN_POSITION
1 RA	1

```
CREATE TABLE ALUNO(  
    ra int,  
    nome varchar2(60) NOT NULL,  
    cpf int(11) UNIQUE,  
    data_nascimento date,  
    PRIMARY KEY (ra)  
);
```

OU

```
CREATE TABLE ALUNO (  
    ra int PRIMARY KEY,  
    nome varchar2(60) NOT NULL,  
    cpf int (11) UNIQUE,  
    data_nascimento date  
)
```

EXEMPLO PRIMARY KEY

CHAVE PRIMÁRIA

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	No	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	DT_NASCIMENTO	DATE	Yes	(null)	3 (null)	

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER
1	PK_RA	Primary_Key	(null)	(null)
2	SYS_C008317	Check	"NOME" IS NOT NULL	(null)

Colunas	
Atualizar: 0	
COLUMN_NAME	COLUMN_POSITION
1 RA	1

```
CREATE TABLE ALUNO (  
    ra int,  
    nome varchar2(60) NOT NULL,  
    cpf int(11) UNIQUE,  
    data_nascimento date,  
    CONSTRAINT pk_ra PRIMARY KEY (ra)  
)
```

- Ao criar uma restrição no nível da tabela, o nome da **CONSTRAINT** se torna obrigatório.

I RESTRIÇÃO: **FOREIGN KEY**

- Uma **FOREIGN KEY (FK)** é uma **restrição de integridade referencial**, que permiti garantir a consistência dos dados em um banco de dados relacional.
- A **FK** especifica que uma coluna ou um conjunto de colunas que fazem parte de **uma chave estrangeira devem ter valores que correspondam aos valores da chave primária** ou de uma chave exclusiva **na tabela referenciada**.
- Assim, evita-se que haja registros órfãos ou inválidos nas tabelas relacionadas. As restrições de integridade referencial são baseadas nos valores dos dados e não na estrutura física do banco de dados. Elas podem ser definidas no nível da coluna ou no nível da tabela.

SINTAXE

```
CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(TAMANHO)
        REFERENCES nome_tab_referenciada(coluna_pk)
    );
-- OU
CREATE TABLE
    NOME_TABELA (
        nome_campo TIPO(TAMANHO),
        FOREIGN KEY (nome_campo)
        REFERENCES nome_tab_referenciada(coluna_pk)
    );
```

EXEMPLO FOREIGN KEY

CHAVE ESTRANGEIRA

FIAP

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CEP	NUMBER(38,0)	No	(null)	1 (null)	
2	RUA	VARCHAR2(100 BYTE)	Yes	(null)	2 (null)	
3	CIDADE	VARCHAR2(100 BYTE)	Yes	(null)	3 (null)	
4	UF	CHAR(2 BYTE)	Yes	(null)	4 (null)	

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	CPF	NUMBER(11,0)	Yes	(null)	3 (null)	
4	DT_NASCIMENTO	DATE	Yes	(null)	4 (null)	
5	CEP	NUMBER(38,0)	Yes	(null)	5 (null)	

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT
1	SYS_C008324	Check	"NOME" IS NOT NULL	(null)	(null)	(null)
2	SYS_C008325	Unique	(null)	(null)	(null)	(null)
3	SYS_C008326	Foreign_Key	(null)	SYS	ENDereco	SYS_C008323

Colunas

Atualizar: 0

	COLUMN_NAME	COLUMN_POSITION
1	CEP	1

COMANDO

```
CREATE TABLE
    ENDereco(
        cep int PRIMARY KEY,
        rua varchar(100),
        cidade varchar(100),
        uf char(2)
    );
```

```
CREATE TABLE
    ALUNO (
        ra int PRIMARY KEY,
        nome varchar(60) NOTNULL,
        data_nascimento date,
        cep int
        REFERENCES ENDereco(cep)
    );
```

OU

```
CREATE TABLE
    ENDereco(
        cep int PRIMARY KEY,
        rua varchar(100),
        cidade varchar(100),
        uf char(2)
    );
```

```
CREATE TABLE
    ALUNO (
        ra int PRIMARY KEY,
        nome varchar(60) NOTNULL,
        data_nascimento date,
        cep int,
        FOREIGN KEY (cep) REFERENCES ENDereco(cep)
    );
```

EXEMPLO FOREIGN KEY

CHAVE ESTRANGEIRA

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CEP	NUMBER(38,0)	No	(null)	1 (null)	
2	RUA	VARCHAR2(100 BYTE)	Yes	(null)	2 (null)	
3	CIDADE	VARCHAR2(100 BYTE)	Yes	(null)	3 (null)	
4	UF	CHAR(2 BYTE)	Yes	(null)	4 (null)	

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	CPF	NUMBER(11,0)	Yes	(null)	3 (null)	
4	DT_NASCIMENTO	DATE	Yes	(null)	4 (null)	
5	CEP	NUMBER(38,0)	Yes	(null)	5 (null)	

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE
1	FK_ALUNO_ENDERECO	Foreign_Key	(null)	SYS	ENDERECO	SYS_C008323	NO ACTION
2	SYS_C008327	Check	"NOME" IS NOT NULL	(null)	(null)	(null)	(null)
3	SYS_C008328	Unique	(null)	(null)	(null)	(null)	(null)

Colunas

Atualizar: 0

	COLUMN_NAME	COLUMN_POSITION
1	CEP	1

```
CREATE TABLE
    ENDERECO(
        cep int PRIMARY KEY,
        rua varchar(100),
        cidade varchar(100),
        uf char(2)
    );

CREATE TABLE
    ALUNO (
        ra int PRIMARY KEY,
        nome varchar(60) NOTNULL,
        cpf int (11) UNIQUE,
        data_nascimento date,
        cep int
        CONSTRAINT fk_aluno_endereco
            REFERENCES ENDERECO(cep)
    );
```

OU

```
CREATE TABLE
    ENDERECO(
        cep int PRIMARY KEY,
        rua varchar(100),
        cidade varchar(100),
        uf char(2)
    );

CREATE TABLE
    ALUNO (
        ra int PRIMARY KEY,
        nome varchar(60) NOTNULL,
        cpf int (11) UNIQUE,
        data_nascimento date,
        cep int,
        CONSTRAINT fk_aluno_endereco
            FOREIGN KEY (cep) REFERENCES ENDERECO(cep)
    );
```

- Ao criar uma restrição no nível da tabela, o nome da **CONSTRAINT** se torna obrigatório.

I RESTRIÇÃO: CHECK

- Uma restrição **CHECK** é uma regra que determina quais valores são válidos para uma coluna ou uma tabela em um banco de dados.
- Ela **permite especificar uma expressão lógica que deve ser verdadeira para cada registro inserido** ou atualizado na coluna ou na tabela.
- A expressão lógica pode usar operadores de comparação como **>, <, >=, <=, IN, NOT IN, NOT BETWEEN, BETWEEN, = e <>**.
- Uma restrição de verificação pode ser definida no momento da criação da coluna ou da tabela, ou posteriormente com o comando ALTER TABLE.

SINTAXE

```
CREATE TABLE  
    NOME_TABELA (  
        nome_campo TIPO(TAMANHO)  
        CHECK (operacao_logica)  
    );
```


EXEMPLO CHECK

VERIFICAR

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	DT_NASCIMENTO	DATE	Yes	(null)	3 (null)	
4	SEXO	CHAR(2 BYTE)	Yes	(null)	4 (null)	

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_T
1 SYS_C008333	Check	"NOME" IS NOT NULL (null)	(null)	(null)
2 SYS_C008334	Check	sexo in('m','f')	(null)	(null)

COLUMN_NAME	COLUMN_POSITION
1 SEXO	(null)

```
CREATE TABLE ALUNO (  
    ra int,  
    nome varchar2(60) NOT NULL,  
    cpf int(11) UNIQUE,  
    data_nascimento date,  
    cep int,  
    sexo char(1) CHECK (sexo IN('M','F'))  
)
```

EXEMPLO CHECK

VERIFICAR

FIAP

COMANDO

RESULTADO

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1 (null)	
2	NOME	VARCHAR2(60 BYTE)	No	(null)	2 (null)	
3	DT_NASCIMENTO	DATE	Yes	(null)	3 (null)	
4	SEXO	CHAR(2 BYTE)	Yes	(null)	4 (null)	

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	
1	CK_SEXO	Check	sexo in('m','f')	(null)	(null)	(n
2	SYS_C008335	Check	"NOME" IS NOT NULL	(null)	(null)	(n

Colunas



Atualizar: 0

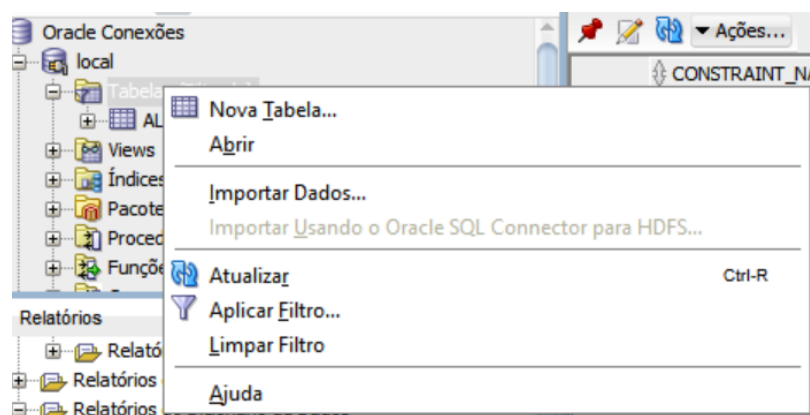
	COLUMN_NAME	COLUMN_POSITION
1	SEXO	(null)

```
CREATE TABLE
    ALUNO (
        ra int PRIMARY KEY,
        nome varchar(60) NOTNULL,
        data_nascimento date,
        cep int,
        sexo char(2),
        CONSTRAINT ck_sexo CHECK (sexo IN ('M','F'))
    );
```

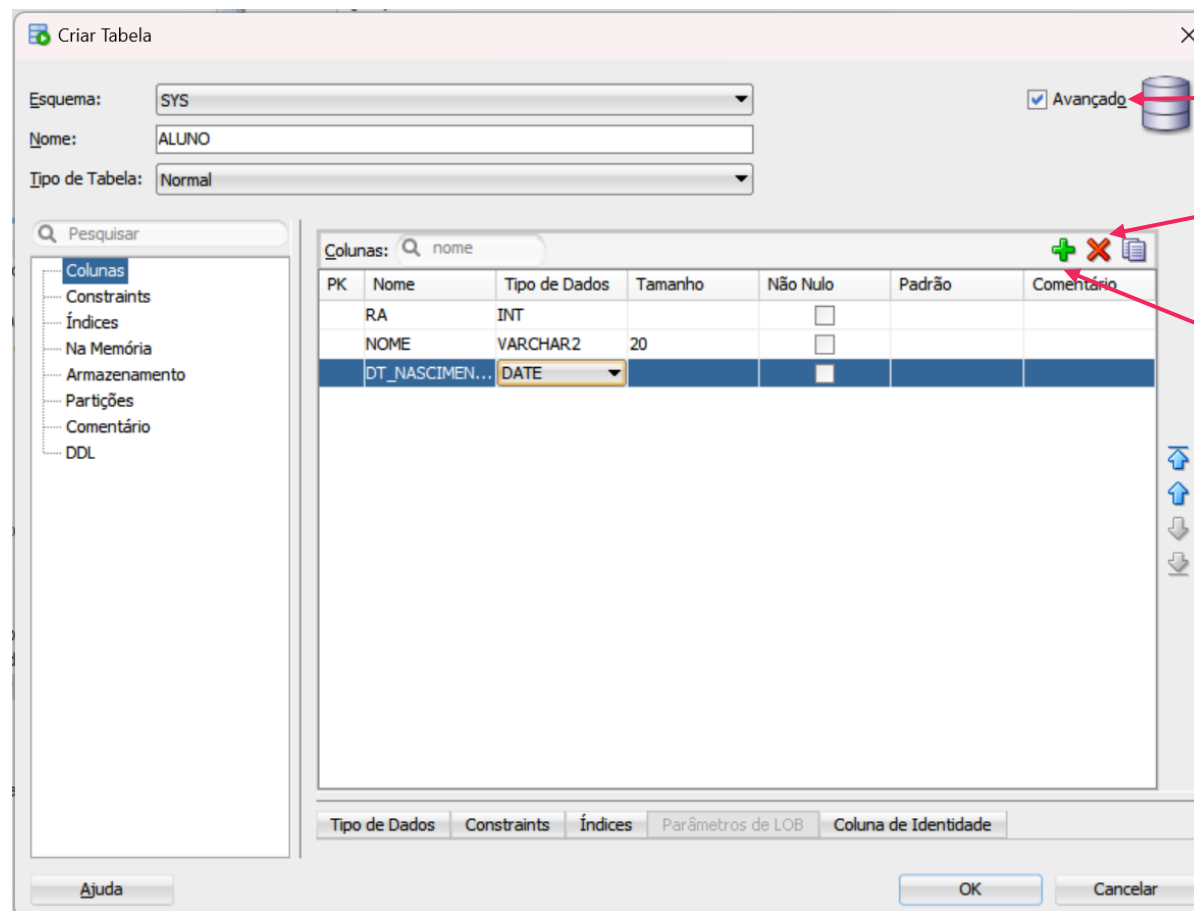
DATA DEFINITION LANGUAGE (DDL)

Programação Visual

1. Clique-direito no objeto da ligação -> Nova Tabela



2. Preencha as colunas com os seus tipos



Habilita

Remove
Colunas

Adicionar
Colunas

3. Habilita Chave Primária

Esquema: SYS
Nome: ALUNO
Tipo de Tabela: Normal

Avançado

Colunas: nome

PK	Nome	Tipo de Dados	Tamanho	Não Nulo	Padrão	Comentário
<input checked="" type="checkbox"/>	RA	INT		<input checked="" type="checkbox"/>		
	NOME	VARCHAR2	20	<input type="checkbox"/>		
	DT_NASCIMEN...	DATE		<input type="checkbox"/>		

Tipo de Dados Constraints Índices Parâmetros de LOB Coluna de Identidade

Ajuda OK Cancelar

4. Visualiza o SQL em DDL

Esquema: SYS
Nome: ALUNO
Tipo de Tabela: Normal

Avançado

Colunas Constraints Índices Na Memória Armazenamento Partições Comentário DDL

Instrução(ões) SQL:



☒ Criar ☐ Atualizar (para edição atual)

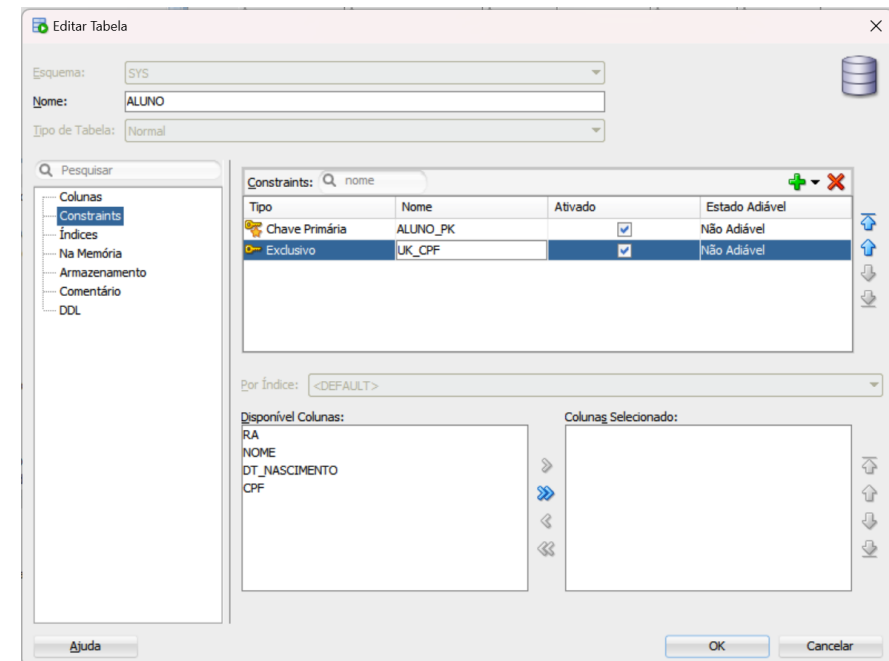
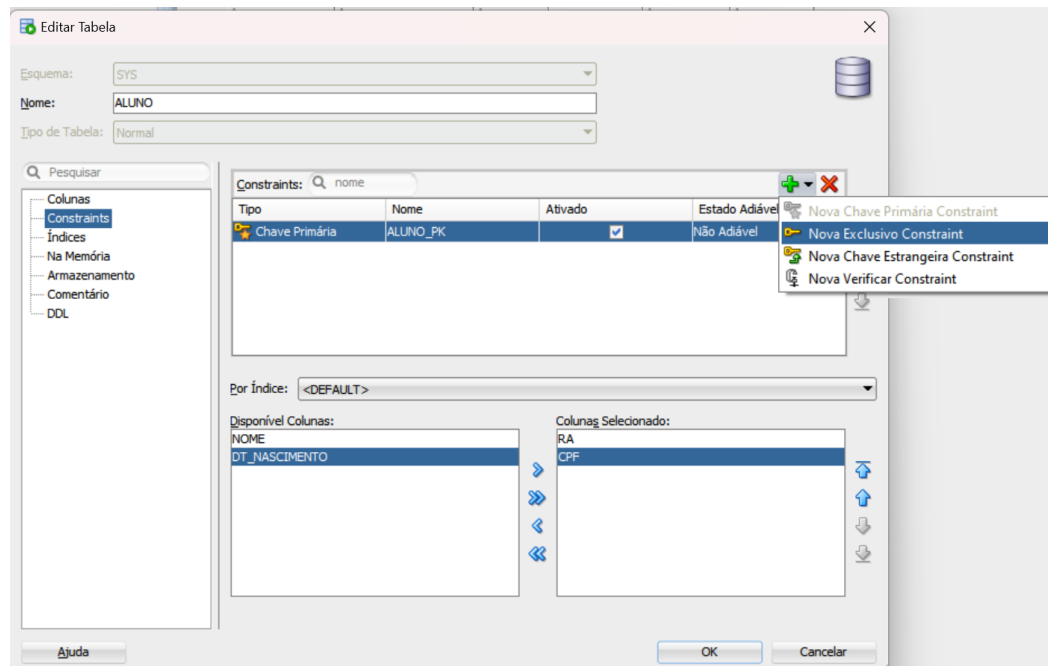
```
CREATE TABLE ALUNO
(
  RA INT
, NOME VARCHAR2 (20)
, DT_NASCIMENTO DATE
, CONSTRAINT ALUNO_PK PRIMARY KEY
(
  RA
)
ENABLE
);
```

Salvar...

Ajuda OK Cancelar


4. Adicionar uma **CONSTRAINT UNIQUE**:

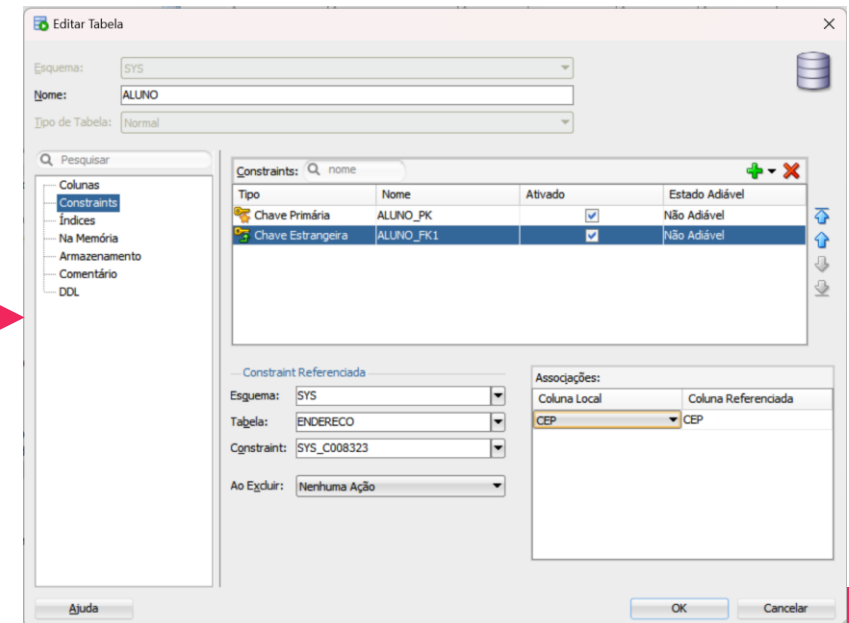
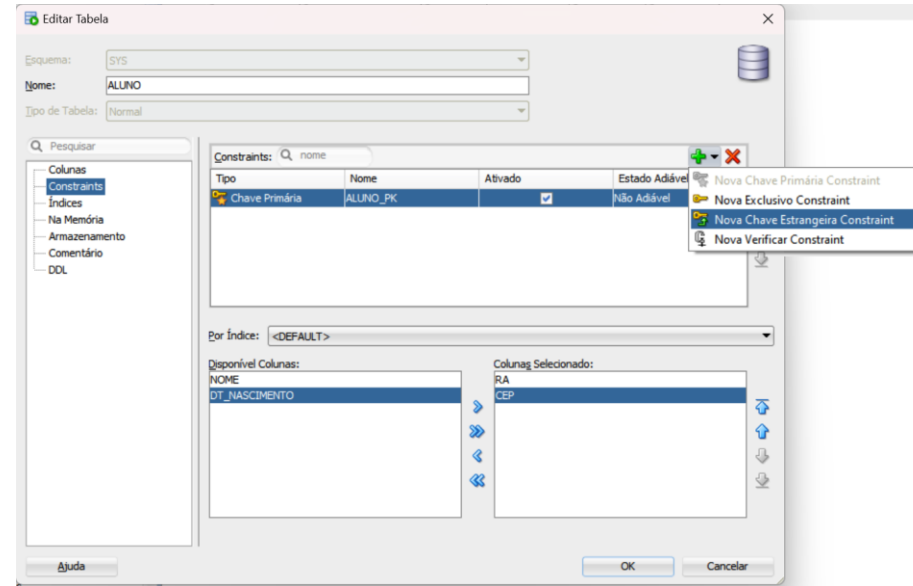
- Selecione uma coluna disponível;
- Clique em   Nova Exclusivo Constraint
- Denomine o nome para a CONSTRAINT



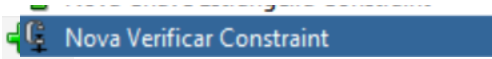
I CREATE TABLE

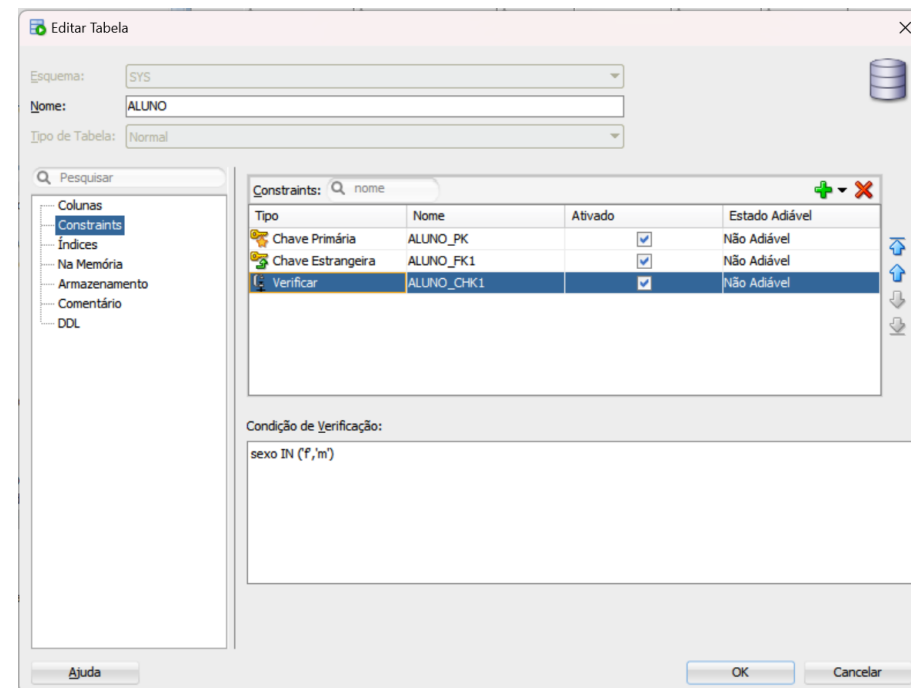
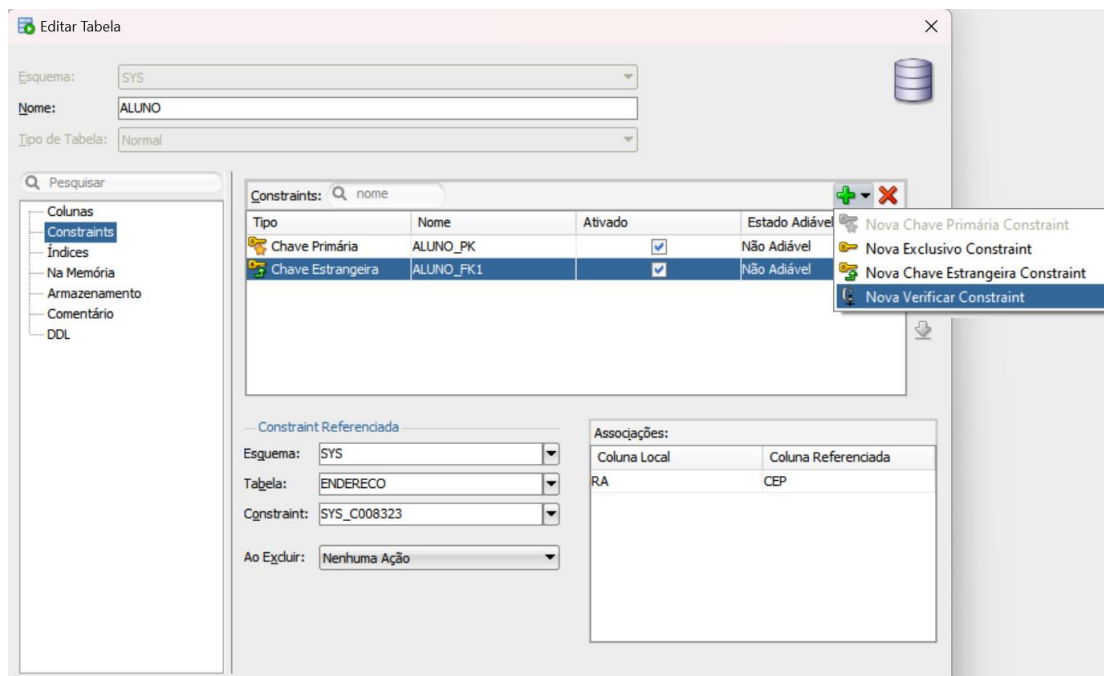
4. Adicionar uma **CONSTRAINT FOREIGN KEY**:

- Selecione uma coluna disponível;
- Clique em  Nova Chave Estrangeira Constraint
- Selecione a Tabela
- Selecione a CONSTRAINT da Chave Primária da Tabela a ser Referenciada;
- Denomine o nome para a CONSTRAINT



4. Adicionar uma **CONSTRAINT CHECK**:

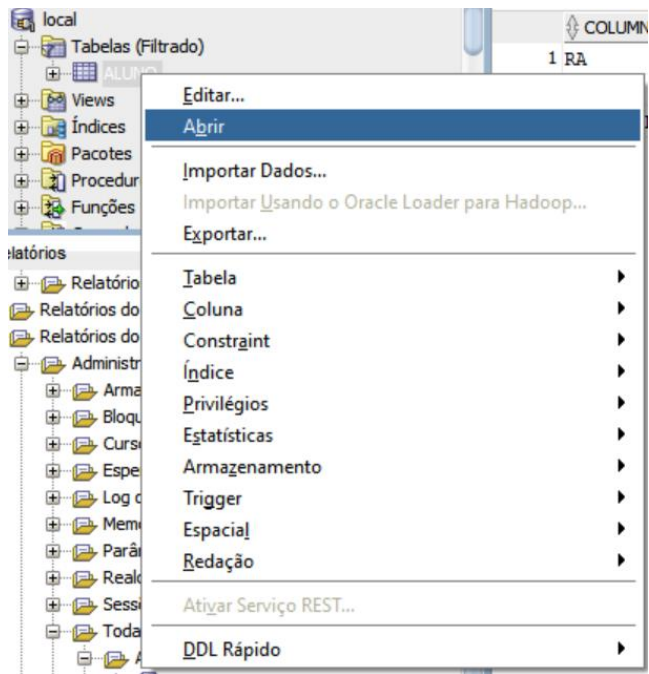
- Selecione uma coluna disponível;
- Clique em  Nova Verificar Constraint
- Defina uma condição lógica
- Denomine o nome para a CONSTRAINT



Resultado: Tabela Aluno Criada

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT	↕ COLUMN_ID	↕ COMMENTS
1	RA	NUMBER(38,0)	Yes	(null)	1	(null)
2	NOME	VARCHAR2(60 BYTE)	Yes	(null)	2	(null)
3	DT_NASCIMENTO	DATE	Yes	(null)	3	(null)

1. Clique-direito no objeto da tabela-> Abrir



Estrutura da Tabela

Dados da Tabela

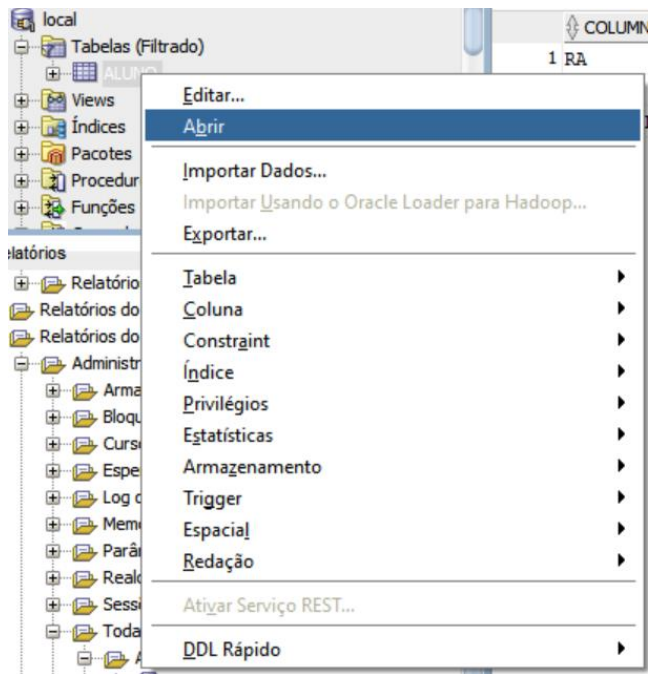
Diagrama da Tabela

Restrições da Tabela

SQL da Tabela

Colunas	Dados	Model	Constraints	Concessões	Estatísticas	Triggers	Flashback	Dependências	Detalhes	Partições	Índices	SQL
Ações...												
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS							
1 RA	NUMBER(38,0)	No	(null)	1	(null)							
2 NOME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)							
3 DT_NASCIMENTO	DATE	Yes	(null)	3	(null)							

1. Clique-direito no objeto da tabela-> Abrir



Estrutura da Tabela
Dados da Tabela
Diagrama da Tabela
Restrições da Tabela
SQL da Tabela

Colunas

Dados

Model

Constraints

Concessões

Estatísticas

Triggers

Flashback

Dependências

Detalhes

Partições

Índices

SQL

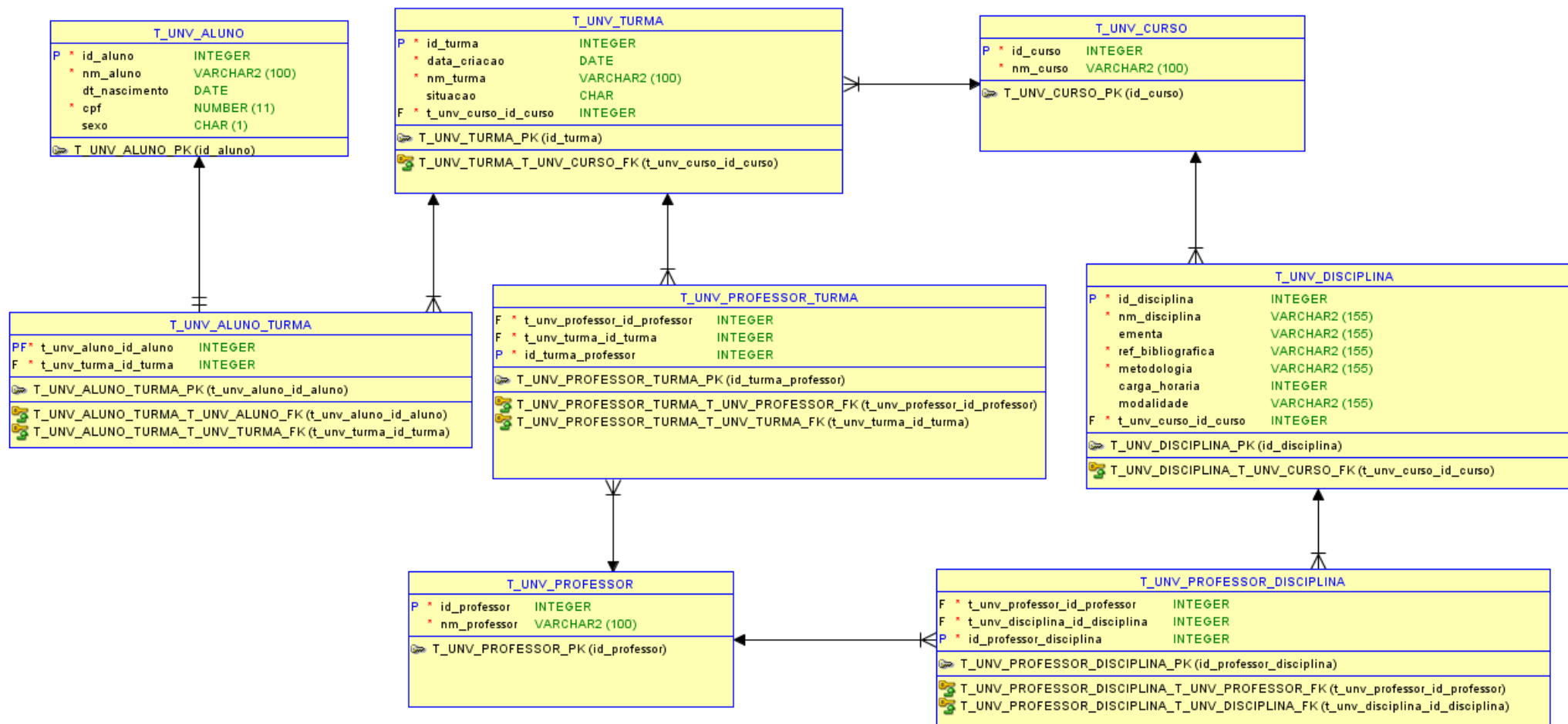
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RA	NUMBER(38,0)	No	(null)	1	(null)
2	NOME	VARCHAR2(20 BYTE)	Yes	(null)	2	(null)
3	DT_NASCIMENTO	DATE	Yes	(null)	3	(null)

EXERCÍCIO PRÁTICO

- A universidade UNV necessita ter um controle dos ALUNOS, PROFESSORES, CURSOS e DISCIPLINAS;
- As etapa de Modelagem Conceitual e Lógica foi realizada por uma equipe específica e Você está sendo convidado a continuar na implementação desse Sistema de Informação;
- Nesse momento, você irá construir a Modelagem Física (o script SQL) no



EXERCÍCIO PRÁTICO



- Criar as estruturas das tabelas no SQL Developer, além de:
 - Criar as chaves primárias e estrangeiras;
 - Os campos cpf (ALUNO), nm_curso (CURSO) e nm_turma (TURMA) são chave única;
 - O campo sexo (ALUNO) deve ser criado para receber três valores: (I) Indefinido, (F) Feminino e (M) Masculino;
 - O campo modalidade (disciplina) deve ser criado para receber dois valores: PRESENCIAL ou REMOTO;
 - O campo situação (TURMA) deve ser criado para receber dois valores: (I) Inativo ou (A) Ativo;
 - Os campos * não podem aceitar valores nulos;

Copyright © 2024 Profº Drº Francisco Douglas Lima Abreu

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito ao autor

FIAP

THE WAY WE ARE