

# Aplicação de Base de Dados: Hollywood

Parte 2

**Projeto realizado por:**

**Alexandre Pereira, nº30003630**

**Bruno Silva, nº30003696**

**Diogo Mendes, nº30003865**

**Reinaldo Coelho, nº30003670**

**Unidade Curricular: Aplicação de Base de Dados**

**Curso: Engenharia Informática**

**Turno: Diurno**

**Professora: Valéria Pequeno**

## Introdução

Com este trabalho pretendemos solucionar um problema que nos foi apresentado por Hollywood, para realizarmos o modelo de base de dados e a sua respetiva aplicação de forma a armazenar informações relativas aos estúdios nesta cidade.

Na primeira parte do trabalho, com base nas informações que nos foram dadas e nos requisitos que são pretendidos da base de dados, passámos por elaborar um Diagrama de Entidade-Relacionamento, com o draw.io, uma ferramenta que permite realizar diagramas deste tipo.

Em seguida foi-nos proposto implementar o modelo lógico, para isso, criámos um modelo Relacional (MR) na primeira forma normal através do DER existente que colocámos na terceira forma normal. Implementámos o MR no Apex do Oracle, utilizando a linguagem SQL. Por fim com todas as tabelas e respetivos atributos inseridos, adicionámos dados meramente fictícios, não correspondentes á realidade, para que seja possível verificar o desempenho relativo á consulta da base de dados e, assim, terminar a primeira fase do trabalho proposto por Hollywood.

Para a segunda parte do projeto, Hollywood pediu alterar o esquema da base de dados para cumprir novas restrições de integridade, onde para cumprimos essas restrições, temos de usar comandos SQL ou PL/SQL, dependendo da situação. O pedido mais importante por Hollywood para a segunda parte é criar aplicações através da linguagem PL/SQL, que é uma variação do SQL/PSM que o Oracle usa, que permite criar stored procedures e funções, triggers, entre outras funcionalidade que o SQL não suporta. Iremos usar o Oracle APEX para criar as aplicações de Hollywood em PL/SQL.

## Modelação

### MR - Primeira Forma Normal

Em relação ao Diagrama Entidade Relacionamento, não realizamos qualquer alteração, pois não verificamos que houvesse algum erro no Diagrama e achamos que está correto com todos os requisitos que foram pedidos na primeira parte.

Já para o Modelo Relacional, decidimos realizar uma pequena alteração na terceira forma normal na qual iremos explicar em baixo na secção da terceira forma normal.

**ATOR** (Número\_ID, Nome\_Artístico, Número\_SS, Telefone, Telemóvel, E-mail, Sexo, Data\_Nascimento, Idade, Nacionalidade)

**FILME** (Nome\_Original, Nome\_Português, Duração\_Filmagens, Custo\_Total, Ano\_Lançamento, Nome\_Realizador)

**FK:** Nome\_Realizador refere-se a Nome (Realizador);

**REALIZADOR** (Nome, Telefone)

**ESTÚDIO** (Nome, Morada, Cod\_Postal, Número\_Porta, Dono, Data\_Fundação, Lucro)

**CINEMA** (Localidade, Nome)

**ATOR\_É\_REALIZADOR** (Número\_ID, Nome)

**FK:** Número\_ID refere-se a Número\_ID (Ator); Nome refere-se a Nome (Realizador);

**PARTICIPA\_EM** (Número\_ID, Nome\_Original, Cachê, Nome\_Personagem)

**FK:** Número\_ID refere-se a Número\_ID (Ator); Nome\_Original refere-se a Nome\_Original(Filme);

**PRODUZIDO\_POR** (Nome\_Original, Nome\_Estúdio)

**FK:** Nome\_Original refere-se a Nome\_Original(Filme); Nome\_Estúdio refere-se a Nome\_Estúdio (Estúdio)

**DÍSTRIBUIDO\_POR** (Nome\_Original, Nome\_Cinema, Localização)

**FK:** Nome\_Original refere-se a Nome\_Original (Filme);

**FK:** Nome\_Cinema refere-se a Nome (Cinema);

**FK:** Localização refere-se a Localização (Cinema);

**ATOR\_APTIDÃO** (Número\_ID, Aptidão)

**FK:** Número\_ID refere-se a Número\_ID (Ator);

**FILME\_GÉNERO** (Nome\_Original, Género)

**FK:** Nome\_Original refere-se a Nome\_Original(Filme);

### **MR - Terceira Forma Normal**

**ATOR** (Número\_ID, Telefone, Telemóvel, E-mail, Sexo, Data\_Nascimento, Nacionalidade)

**ATOR\_NOME\_SS** (Número\_ID, Nome\_Artístico, Número\_SS)

**FK:** Número\_ID refere-se a Número\_ID (Ator);

**FILME** (Nome\_Original, Nome\_Português, Duração\_Filagens, Custo\_Total, Ano\_Lançamento, Nome\_Realizador)

**FK:** Nome\_Realizador refere-se a Nome (Realizador);

**REALIZADOR** (Nome, Telefone)

**ESTÚDIO** (Nome, Morada, Número\_Porta, Dono, Data\_Fundação, Lucro)

**ESTÚDIO\_CODPOSTAL** (Nome\_Estúdio, Cod\_Postal)

**FK:** Nome\_Estúdio refere-se a Nome (Estúdio);

**CINEMA** (Localidade, Nome)

**ATOR\_REALIZADOR** (Número\_ID, Nome)

**FK:** Número\_ID refere-se a Número\_ID (Ator); Nome refere-se a Nome (Realizador);

**PARTICIPA\_EM** (Número\_ID, Nome\_Original, Cachê, Nome\_Personagem)

**FK:** Número\_ID refere-se a Número\_ID (Ator); Nome\_Original refere-se a Nome\_Original (Filme);

**PRODUZIDO\_POR** (Nome\_Original, Nome\_Estúdio)

**FK:** Nome\_Original refere-se a Nome\_Original (Filme); Nome\_Estúdio refere-se a Nome\_Estúdio (Estúdio);

**DÍSTRIBUIDO\_POR** (Nome\_Original, Nome\_Cinema, Localização)

**FK:** Nome\_Original refere-se a Nome\_Original (Filme);

**FK:** Nome\_Cinema refere-se a Nome (Cinema);

**FK:** Localização refere-se a Localização (Cinema);

**ATOR\_APTIDÃO** (Número\_ID, Aptidão)

**FK:** Número\_ID refere-se a Número\_ID (Ator);

**FILME\_GÊNERO** (Nome\_Original, Gênero)

**FK:** Nome\_Original refere-se a Nome\_Original(Filme);

**ATOR\_IDADE**(Número\_ID, Idade)

**FK:** Número\_ID refere-se a Número\_ID(Ator);

Para a criação da Terceira Forma Normal (3FN), realizámos algumas alterações no nosso modelo relacional de forma a cumprir com as normas da 3FN.

Para a segunda parte do projeto, criámos uma relação nova por motivos de retificação de um erro na primeira parte, ao transformarmos o MR na 3FN não considerámos que a idade depende funcionalmente da data de nascimento, erro que corrigimos nesta segunda parte. Para tal efeito retiramos o atributo idade da relação ator e criamos uma nova relação chamada Ator\_Idade que terá como atributos o Numero\_ID do Ator, que é a chave primária e também chave estrangeira já que se referencia ao Numero\_ID da tabela Ator e o atributo Idade.

## Script de criação de dados para a parte 2 do projeto

```
create table Realizador(  
    Nome VARCHAR2(30),  
    Telefone numeric(9,0),  
    Primary key (Nome)  
);
```

```
create table Filme(  
    nome_original VARCHAR2(20),  
    nome_português VARCHAR2(20),  
    duração_filmagens numeric(9,0),  
    custo_total numeric(12,0),  
    Nome_Realizador VARCHAR2(20),  
    ano_lançamento date,  
  
    Primary Key (nome_original),  
    Foreign KEY (Nome_Realizador) REFERENCES Realizador(Nome)  
);
```

```
create table Filme_Genero(  
    nome_original VARCHAR2(20),  
    genero VARCHAR2(20),  
    PRIMARY key (nome_original, genero),  
    Foreign Key (nome_original) references Filme(nome_original)  
);
```

```
create table Cinema(  
    Nome VARCHAR2(30),  
    Localidade VARCHAR2(20),  
    Primary key (Nome, Localidade)  
);
```

```
create table Ator(  
    Numero_ID numeric(4,0),  
    Nacionalidade VARCHAR2(20),  
    data_nascimento date,  
    sexo CHAR(1),  
    email VARCHAR2(50),  
    telefone numeric(9,0),
```

```

    telemovel numeric(9,0),

    Primary key (Numero_ID)
);

CREATE TABLE ATOR_NOME_SS(
    Nome_artistico VARCHAR2(35) not null UNIQUE,
    Numero_ID numeric(4,0),
    Numero_SS VARCHAR2(12) not null UNIQUE,

    Primary key (Numero_ID) ,
    Foreign Key (Numero_ID) references Ator(Numero_ID)
);

Create table Ator_Idade(
    Numero_ID numeric(4,0) primary key,
    Idade numeric(3,0),
    Foreign Key (Numero_ID) references Ator(Numero_ID)

create table Ator_É_Realizador(
    Numero_ID numeric(4,0),
    Nome VARCHAR2(30),

    PRIMARY KEY (Numero_ID),
    Foreign Key (Numero_ID) references Ator(Numero_ID),
    Foreign Key (Nome) references Realizador(Nome)
);

create table Ator_Aptidao (
    Aptidao varchar2(100),
    Numero_ID numeric(4,0),
    PRIMARY KEY (Aptidao, Numero_ID),
    Foreign key (Numero_ID) references Ator(Numero_ID)
);

create table Participa_Em(
    Numero_ID numeric(4,0),
    Nome_Original varchar(20),
    nome_Personagem varchar2(20),
    Cache numeric(10,0),

    PRIMARY KEY (Numero_ID, Nome_Original),

```

Foreign key (Numero\_ID) references Ator(Numero\_ID),  
Foreign key (Nome\_Original) references Filme(Nome\_Original)

);

```
create table Estudio(  
  Nome VARCHAR2(30),  
  Morada VARCHAR2(50),  
  Dono VARCHAR2(20),  
  Numero_Porta numeric(2,0),  
  data_fundação date,  
  lucro numeric(16,0),  
  PRIMARY KEY (nome)
```

);

```
CREATE TABLE Estudio_CodPostal(  
  Nome_estudio VARCHAR2(20),  
  cod_postal VARCHAR2(7),  
  PRIMARY KEY (Nome_estudio),  
  Foreign key (Nome_estudio) references Estudio(Nome)
```

);

```
CREATE TABLE Produzido_Por(  
  Nome_Original VARCHAR2(20),  
  Nome_estudio VARCHAR2(20),  
  PRIMARY KEY (Nome_Original, Nome_estudio),
```

Foreign key (Nome\_estudio) references Estudio(nome),  
Foreign key (Nome\_Original) references Filme(Nome\_Original)

);

```
create table Distribuido_Por(  
  nome_original VARCHAR2(20),  
  Nome_Cinema VARCHAR2(20),  
  Localidade varchar2(20),
```

PRIMARY KEY (nome\_original, Nome\_Cinema, Localidade),

Foreign key (Nome\_Original) references Filme(nome\_original),  
Foreign key (Nome\_Cinema, Localidade) references Cinema(Nome,  
Localidade);



);

-- Parte 2

Create table filme\_log(

nome\_original VARCHAR2(20),

custo\_anterior numeric(12,0),

custo\_atual numeric(12,0),

data\_mod date,

PRIMARY KEY (nome\_original, data\_mod),

Foreign key (nome\_original) references Filme(nome\_original)

);

-- 8.2.2.2

Alter table Ator ADD CONSTRAINT Sexo\_Ator CHECK(sexo = 'M' or sexo = 'F');

-- 8.2.2.4

Alter table Filme MODIFY custo\_total numeric(12,0) NOT NULL;

Alter table Filme ADD CONSTRAINT custo\_total CHECK (custo\_total > 0);

## Dicionário de Dados

Realizador: Contém todos os realizadores de Hollywood				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Nome	VARCHAR2	30 caracteres	PK	Nome do realizador
Telefone	Numeric	9caracteres/0decimas		Telefone do realizador

Filme: Contém todos os filmes de Hollywood				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
nome_original	VARCHAR2	20 caracteres	PK	Nome do filme
Nome_português	VARCHAR2	20 caracteres		Nome em português do filme
Duração_filmagens	NUMERIC	9caracteres/0decimas		Duração das filmagens do filme
Custo_total	NUMERIC	10caracteres/0decimas		Custo total do filme
Nome_realizador	VARCHAR	20caracteres	FK REFERENCE Realizador(nome)	Nome do realizador do filme
Ano_lançamento	DATE	MM/DD/AAAA		Ano de lançamento do filme

Filme_Genero: Contém todos os géneros de todos os filmes de Hollywood				
Atributos	Tipos de Dados	Comprimento	Restrições	Descrição
Nome_original	VARCHAR2	30 caracteres	PK, FK reference Filme(nome_original)	Nome do filme
Genero	VARCHAR2	20 caracteres	PK	Genero do filme

Cinema: Contém todos os cinemas				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Nome	VARCHAR2	30 caracteres	PK	Nome do cinema
Localidade	VARCHAR2	20 caracteres	PK	Localidade do cinema

Ator: Contém todos os atores de Hollywood				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Numero_ID	Numeric	4 caracteres/0 decimas	PK	Número de identificação (interno) do ator
Nacionalidade	Varchar2	20 caracteres		Nacionalidade do ator
Data_nascimento	DATE	MM/DD/AAAA		Data de Nascimento do ator
Sexo	Char	1 caractere		Sexo do Ator
Email	Varchar2	50 caracteres		Email do ator
Telefone	Numeric	9 caracteres/0 decimas		Telefone do Ator
Telemovel	Numeric	9 caracteres/0 decimas		Telemovel do Ator

Ator_Nome_SS: Contém o nome artístico e número de segurança social dos atores				
Atributos	Tipos de Dados	Comprimento	Restrições	Descrição
Nome_artistico	VARCHAR2	35 caracteres	NOT NULL, Unique	Nome artístico do ator
Numero_ID	NUMERIC	4 caracteres/0 decimas	PK, FK reference to Ator(Numero_ID), NOT NULL	Numero interno do ator
Numero_SS	VARCHAR2	12 caracteres	NOT NULL, Unique	Número de segurança social do ator

Ator_É_realizador: Contém todos os atores que são realizadores e vice-versa				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Numero_ID	Numeric	4caracteres/0 decimas	PK, FK reference Ator(Numero_ID),	Número interno do ator
Nome	VARCHAR2	30 caracteres	FK reference Realizador(Nome),	Nome do realizador

Ator_Aptidão: Contem as capacidades de atuação de todos os atores				
Atributos	Tipos de Dados	Comprimento	Restrições	Descrição
Aptidão	VARCHAR2	100 caracteres	PK	Tipos de personagem que o ator está apto a empenhar
Numero_ID	NUMERIC	4 carecteres/0 decimais	PK, FK Reference Ator(Numero_ID)	Numero interno do ator

Participa_Em: Contem todos os Atores que participam num dado filme				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Numero_ID	Numeric	4 caracteres/0 decimas	PK, FK reference Ator(Numero_ID)	Número interno do ator
Nome_Original	VARCAR2	20 caracteres	PK, FK reference Filme(Nome_Original)	Nome original do filme
Nome_Personagem	VARCHAR2	20 caracteres		Nome do personagem
Cache	Numeric	10 caracteres/0 decimas		Valor monetário obtido pelo ator pela participação no filme

Estudio: Contém todos os estúdios de Hollywood				
Atributos	Tipos de Dados	Comprimento	Restrições	Descrição
Nome	VARCHAR2	30	PK	Nome do estúdio
Morada	VARCHAR2	50		Morada do estúdio
Dono	VARCHAR2	20		Dono do estúdio
Numero_Porta	Numeric	2/0		Número de porta do estúdio
Data_Fundação	DATE	MM/DD/AAAA		Data de fundação do estúdio
Lucro	Numeric	16/0		Lucro do estúdio

Estudio_CodPostal: Contém o código postal de todos os estúdios				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Nome_estudio	VARCHAR2	20 caracteres	PK, FK reference Estudio(Nome)	Nome do estúdio
Cod_postal	VARCHAR2	7 caracteres		Código Postal do estúdio

Produzido_Por: Contém todos os filmes produzidos por cada estúdio				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Nome_Original	VARCHAR2	20 caracteres	PK, FK reference Estudio(nome)	Nome original do filme
Nome_Estudio	VARCHAR2	20caracteres	PK, FK reference Filme(Nome_Original)	Nome do estúdio

Distribuido_Por: Contém todos os filmes em exibição em cada cinema				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Nome_original	VARCHAR2	20 caracteres	PK, FK reference Filme(Nome_original)	Nome original do filme
Nome_Cinema	VARCHAR2	20 caracteres	PK, FK reference Cinema(Nome)	Nome do cinema
Localidade	VARCHAR2	20 caracteres	PK, FK reference Filme(Localidade)	Localidade do cinema

Ator_Idade: Contém a idade de todos os atores				
Atributo	Tipos de Dados	Comprimento	Restrições	Descrição
Numero_ID	Numeric	4 caracteres/0 decimas	PK, FK reference Ator(Numero_ID)	Número de Identificação do Ator
Idade	Numeric	3 caracteres/0 decimas		Idade do Ator

## Aplicação

### Consultas a executar

#### 8.2.2.1 Não existe gêneros diferentes com o mesmo nome

```
create or replace trigger genero_igual
before insert or update of genero on Filme_Genero
for each row
BEGIN
    :new.genero := UPPER(:new.genero);
END;
```

Para esta secção nós consideramos que a inclusão do atributo género na chave primária da relação já é suficiente para garantir que não há géneros diferentes com o mesmo nome. Mesmo assim aplicamos um trigger de modo a que os géneros estejam todos escritos (no ponto de vista do tipo de dado string) da mesma maneira (todos os caracteres em maiúsculas) garantindo assim que no caso de uma pesquisa não haverá conflitos inesperados quando se trata do atributo género.

De acordo com o cenário, é necessário saber a idade de um ator. Esta idade deve ser calculada com base nas informações disponíveis. Apenas deve-se poder aceder a idade do ator (não a do realizador).

Para determinar o cálculo da idade do ator realizamos um bloco PL/SQL utilizando o Número ID do Ator e um trigger antes de ser inserido ou atualizado a data de nascimento do Ator, dentro do bloco o cálculo da idade será feito por uma divisão que consiste entre a subtração entre a data atual que esteja no Oracle e a data de nascimento do Ator a dividir por 365.25 que é a média de dias que existe por ano, para assim contabilizamos quem nasceu num ano bissexto.

```
create or replace trigger ator_idade
before insert or update of data_nascimento on Ator
for each row
when (new.numero_id > 0)
begin
    :new.idade := trunc((Sysdate - :new.data_nascimento)/365.25);
end;
```

### 8.3.1. Criar um bloco PL/SQL anônimo para:

- a. Listar o nome de todos os estúdios (sem repetições) que produziram filmes do gênero "fantasia" tendo como ator/atriz "Ruan Carlos".

```
DECLARE
    valor Produzido_por.Nome_Estudio%TYPE;
    CURSOR cur_produzidoPor is select distinct Produzido_por.Nome_Estudio
    from (((Produzido_por join Filme_Genero on Produzido_por.Nome_Original = Filme_Genero.Nome_Original)
    JOIN Participa_em on Produzido_por.Nome_Original = Participa_em.Nome_Original) join ATOR_NOME_SS
    on Participa_em.Numero_ID = ATOR_NOME_SS.Numero_ID)
    where Filme_Genero.genero = 'Fantasia' and ATOR_NOME_SS.Nome_Artistico = 'Ruan Carlos';

BEGIN
    OPEN cur_produzidoPor;
    LOOP
        FETCH cur_produzidoPor INTO valor;

        IF cur_produzidoPor%NOTFOUND THEN
            EXIT;
        END IF;

        Dbms_output.put_line(valor);
    END LOOP;
    CLOSE cur_produzidoPor;
END;
```

Results	Explain	Desc
20th Century Fox		
Marvel Studios		
Statement processed.		

- b. Listar o filme de aventura com maior custo de produção tendo como realizador XPTO.

```
DECLARE
    nome Filme.nome_original%type;
BEGIN
    SELECT nome_original into nome
    FROM (SELECT Filme.nome_original, max(Filme.custo_total) as maximo
    FROM (Filme inner join Filme_Genero on Filme.nome_original = Filme_Genero.nome_original)
    where Filme_Genero.genero = 'Aventura' and Filme.Nome_Realizador = 'XPTO'
    group by Filme.nome_original
    order by maximo desc)
    where rownum < 2;
    dbms_output.put_line(nome);
END;
```

```
The Adventure 3
Statement processed.
0.02 seconds
```


8.3.2. Crie um bloco PL/SQL (procedimento, função, trigger, etc., o que achar mais conveniente) para atualizar automaticamente uma tabela chamada filme\_log sempre que uma inserção ou atualização no campo custo ocorrer na tabela filme. A tabela filme\_log deve ser colocada no script criado e deve poder armazenar os seguintes

valores: nome do filme, custo anterior (se houver), custo atual, data em que esta alteração ocorreu na base de dados.

```
create or replace trigger custo_log
before update of custo_total on Filme
for each row
begin
    insert into filme_log VALUES(:old.nome_original, :old.custo_total, :new.custo_total, sysdate());
end;
```

### 8.3.3. Criar um bloco PL/SQL para testar o que foi feito no item anterior.

```
update filme set custo_total=12 where nome_original='The godmother'
```

Data				
EDIT	NOME_ORIGINAL	CUSTO_ANTERIOR	CUSTO_ATUAL	DATA_MOD
	The godmother	1200000000	12	12/17/2019

8.3.4. Crie o código necessário para que todo dia 01 de cada mês seja exibida uma mensagem indicando os atores/realizadores que aniversariam neste mês. (**Dica:** veja o funcionamento de `dbms_scheduler.create_job`).

```
create or replace procedure aniversarios_mes
AS
nome ator_nome_SS.nome_artístico%TYPE;
data ator.data_nascimento%TYPE;
dia int;
CURSOR cur_pessoas is select ator_nome_SS.nome_artístico, ator.data_nascimento
from ator join ator_nome_SS on ator.numero_ID = ator_nome_SS.numero_ID
where EXTRACT(MONTH FROM data_nascimento) = EXTRACT(MONTH FROM SYSDATE);
BEGIN
    OPEN cur_pessoas;
    LOOP
        FETCH cur_pessoas into nome, data;
        IF cur_pessoas%NOTFOUND THEN
            EXIT;
        END IF;
        dia:=EXTRACT(DAY FROM DATA);
        dbms_output.put_line(nome || ' faz anos dia ' || dia);
    END LOOP;
    CLOSE cur_pessoas;
END;
```

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
    job_name      => 'Aniversarios_schedule',
    job_type      => 'PLSQL_BLOCK',
    job_action    => 'BEGIN aniversarios_mes; END;'
```



```
repeat_interval => 'FREQ=MONTHLY; BYMONTHDAY=1',
enabled         => TRUE,
comments        => 'Aponta os aniversários do mês');
END;
```

```
Reinaldo faz anos dia: 20
```

## Scripts de Remoção de dados

```
DROP TABLE Filme_Genero;
DROP TABLE Distribuido_Por;
DROP TABLE Produzido_Por;
DROP TABLE Cinema;
DROP TABLE ATOR_NOME_SS;
DROP TABLE Ator_Aptidao;
DROP TABLE Participa_Em;
DROP TABLE Ator_É_Realizador;
DROP TABLE Ator_Idade;
DROP TABLE Ator;
DROP TABLE Filme;
DROP TABLE Realizador;
DROP TABLE Estudio_CodPostal;
DROP TABLE Estudio;
```

## Scripts de remoção de tabelas

```
DELETE FROM Filme_Genero;
DELETE FROM Distribuido_Por;
DELETE FROM Produzido_Por;
DELETE FROM Cinema;
DELETE FROM ATOR_NOME_SS;
DELETE FROM Ator_Aptidao;
DELETE FROM Participa_Em;
DELETE FROM Ator_É_Realizador;
DELETE FROM Ator_Idade;
DELETE FROM Ator;
DELETE FROM Filme;
DELETE FROM Realizador;
DELETE FROM Estudio_CodPostal;
DELETE FROM Estudio;
```

## CONCLUSÃO

Para a primeira parte deste projeto, conseguimos realizar todas as tarefas propostas por Hollywood que constituí construir uma base de dados para os estúdios de Hollywood. Conseguimos elaborar um MR que inclui todas as exigências feitas por Hollywood, um script SQL para criação de base de dados em Oracle e respetivo dicionário de dados.

Para a segunda parte deste projeto adicionámos todas as restrições adicionais após a criação da base de dados em si, e também conseguimos realizar todas as outras tarefas relativamente ao uso do PL/SQL sem muitas dificuldades.

Sendo assim, não houve nenhum objetivo que Hollywood que não conseguimos realizar, tendo em conta que realizamos todos os objetivos propostos na primeira parte e também para a segunda parte.