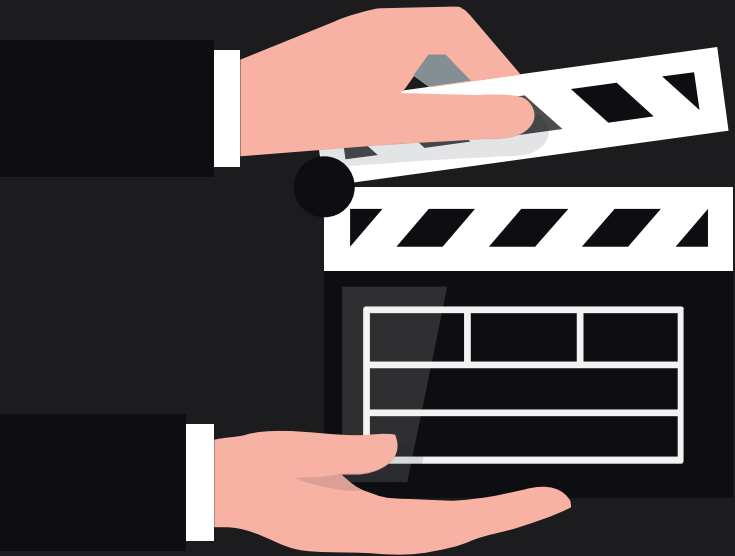


# Projeto Banco de Dados – 2022.2

**Discentes:** José Gervásio, Ricardo Morato, Victor Oliveira, Vitória Pinheiro, Wytoria Faustino

**Docente:** Robson Fidalgo





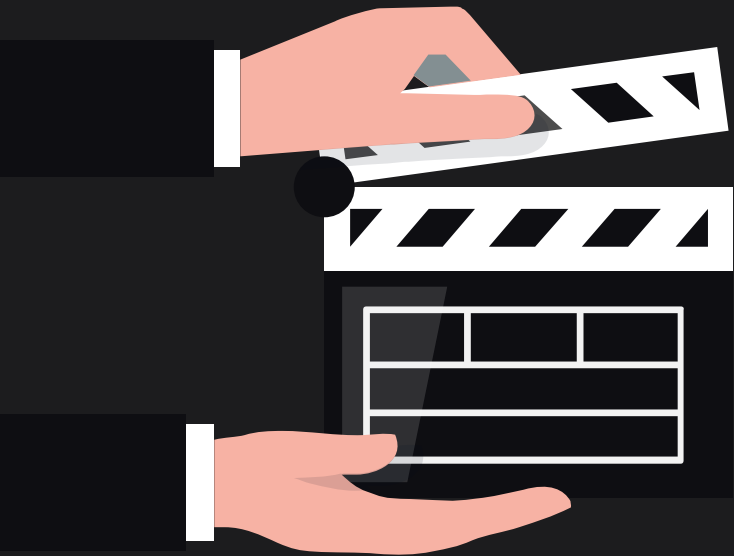
01

Contexto

# Agência de atores

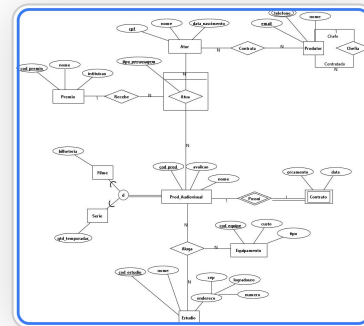
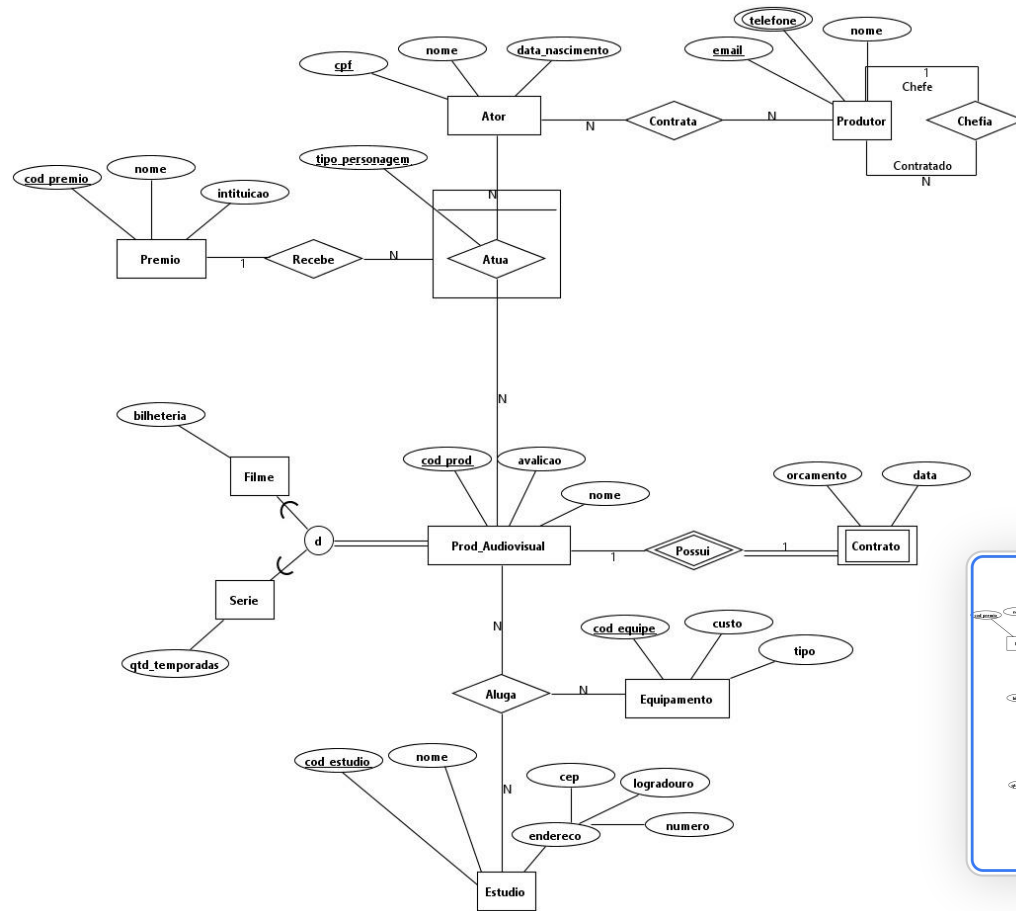
Atores, produções audiovisuais,  
estúdios e equipamentos, tudo em  
um só lugar!



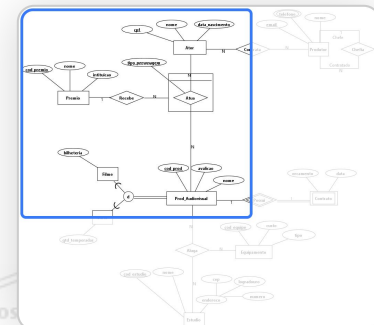
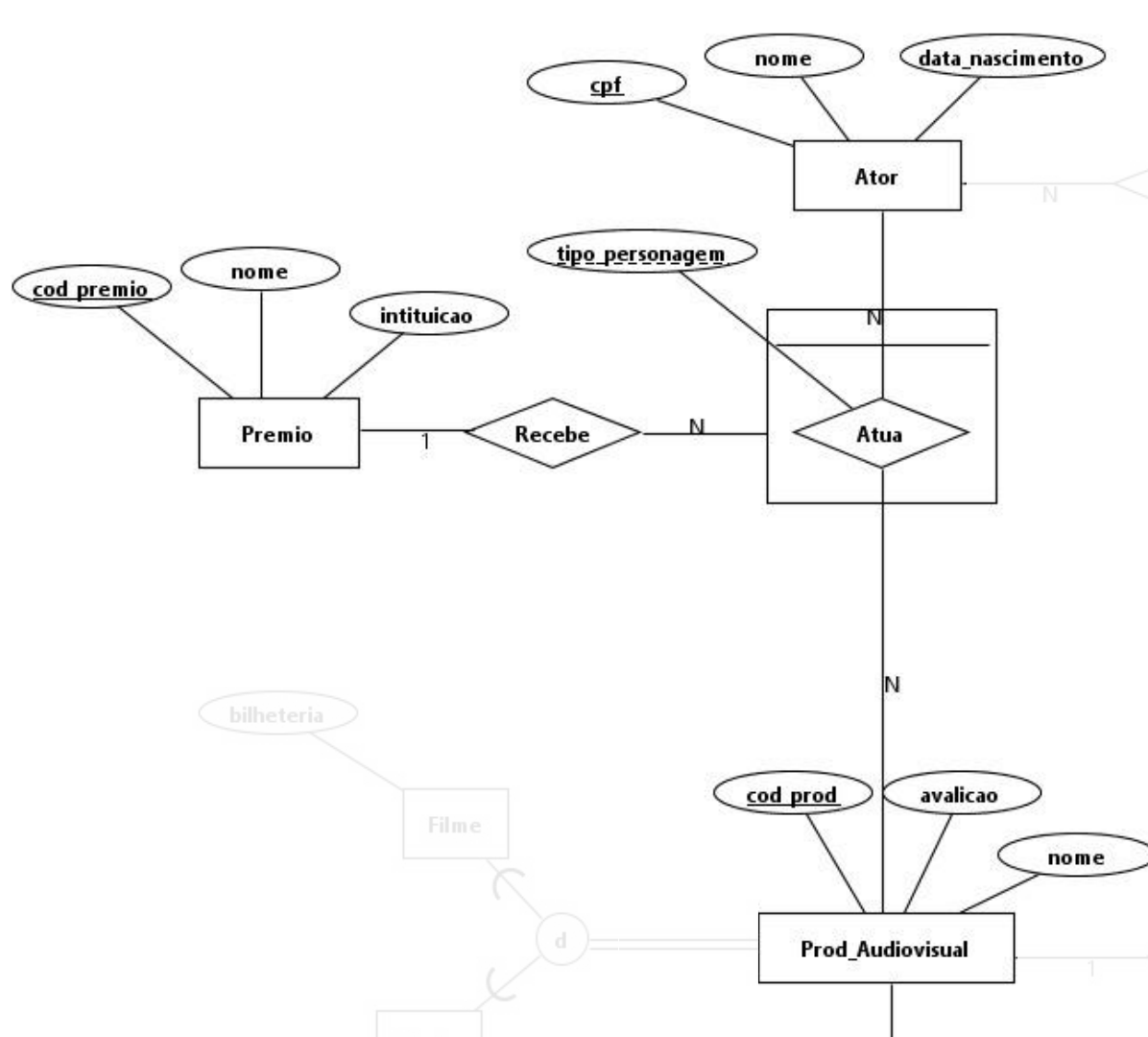


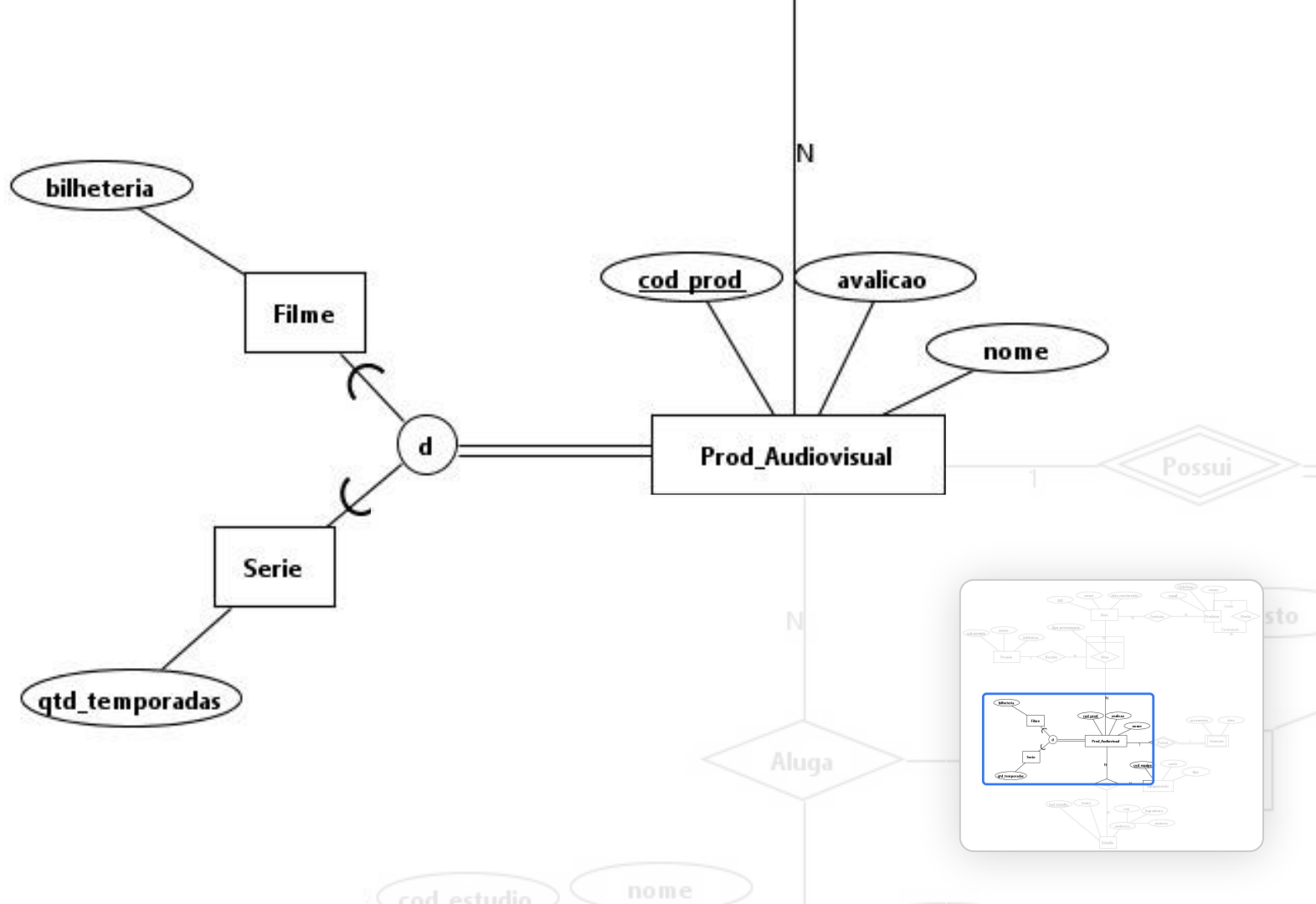
02

Projeto Conceitual

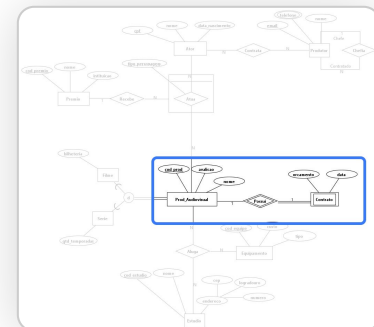
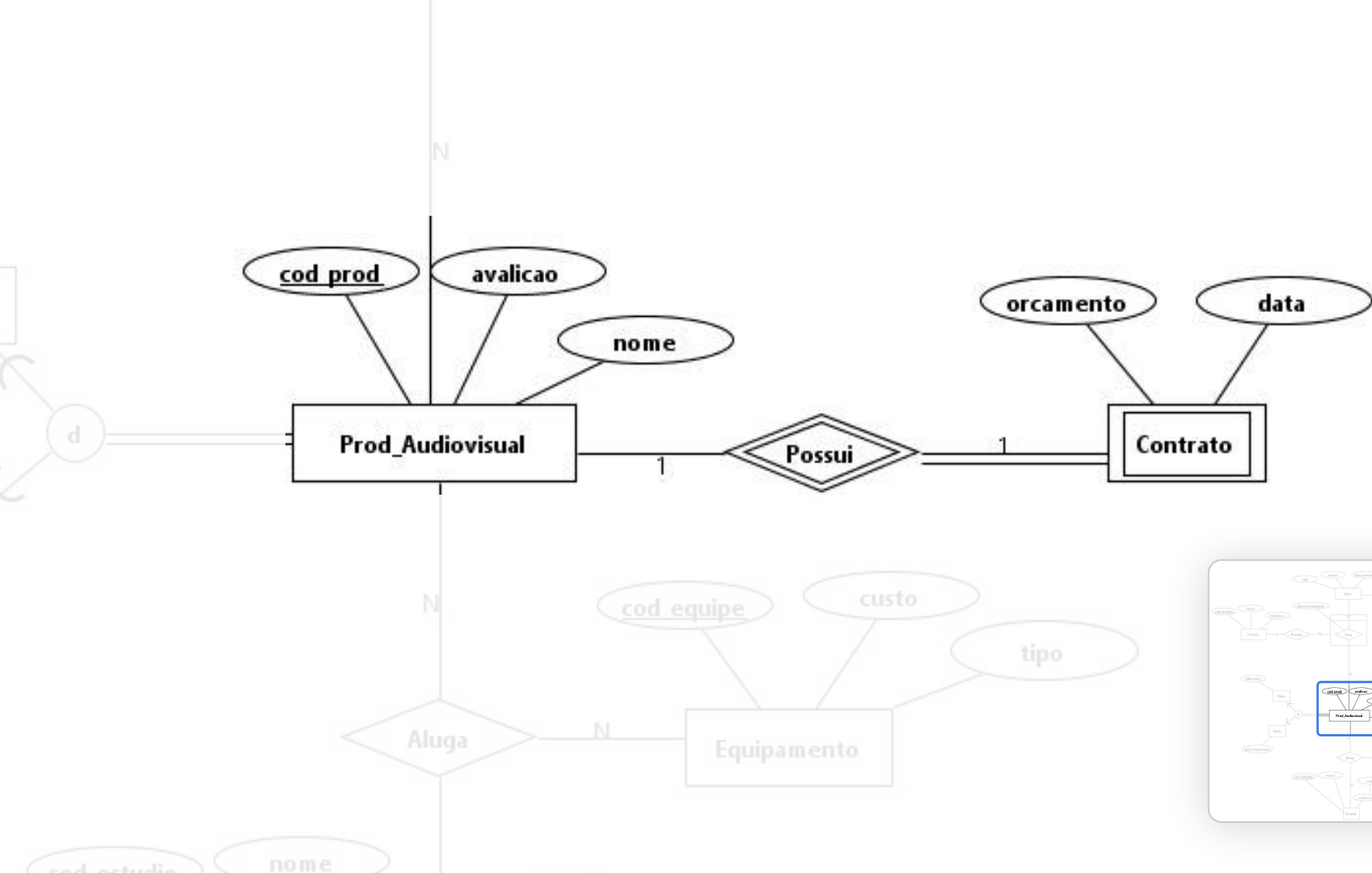


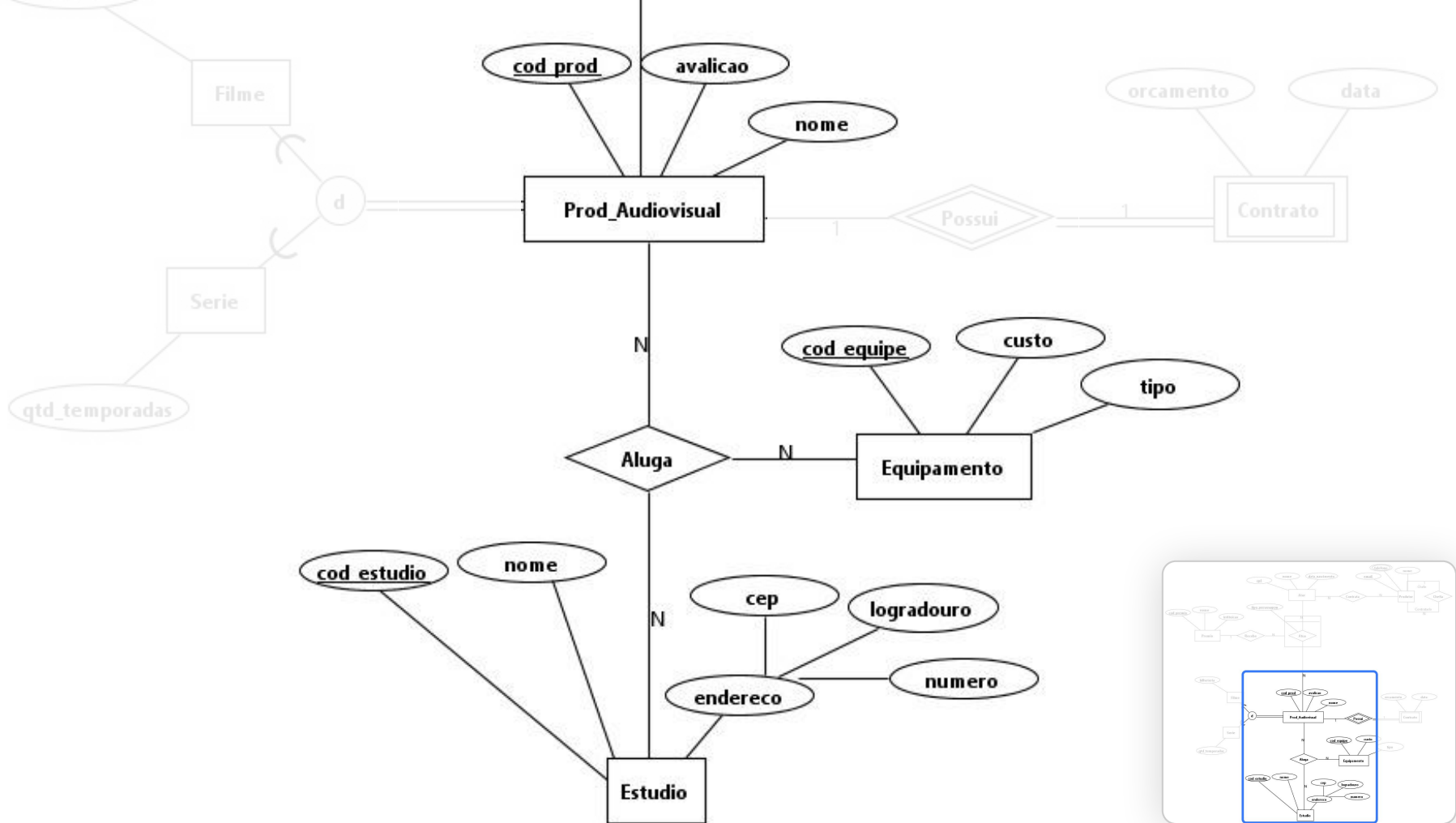


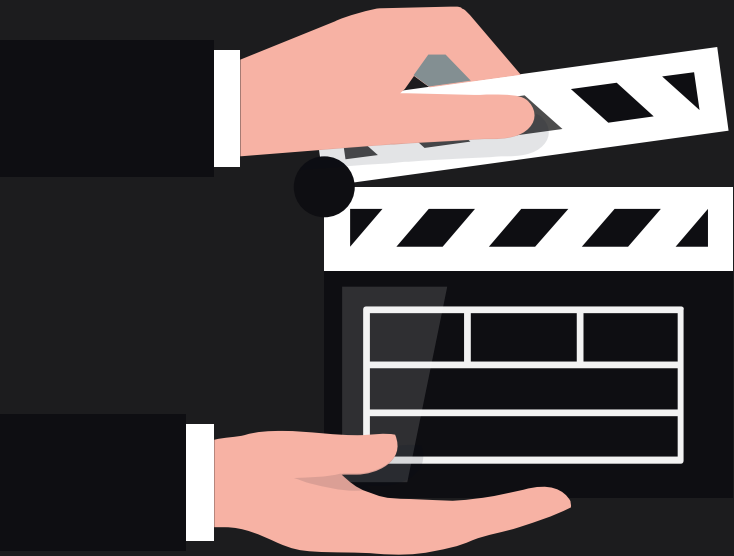






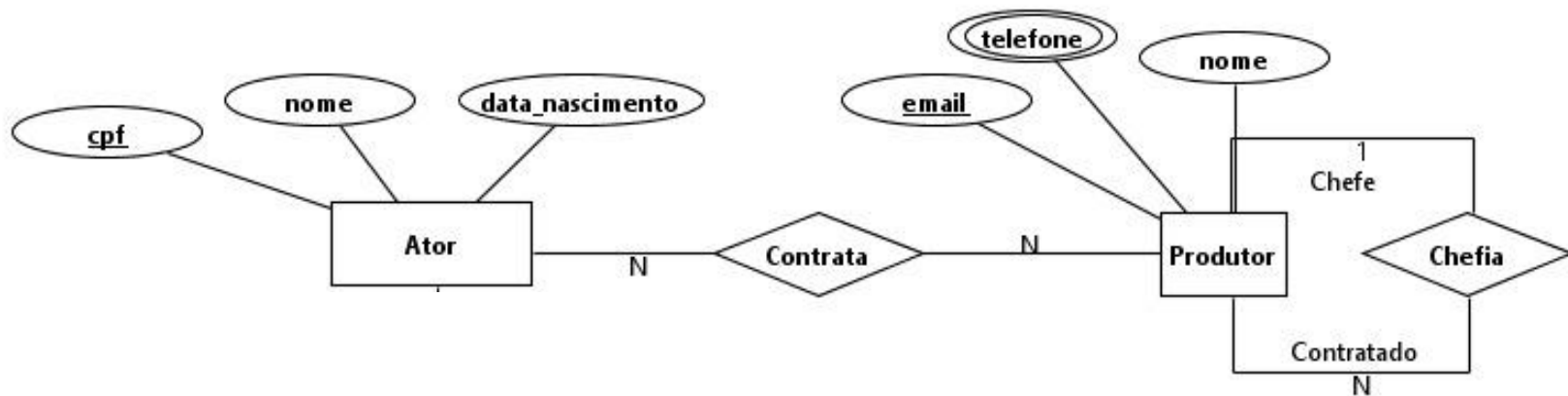






03

Projeto Lógico



**Ator (cpf, nome, data\_nascimento)**

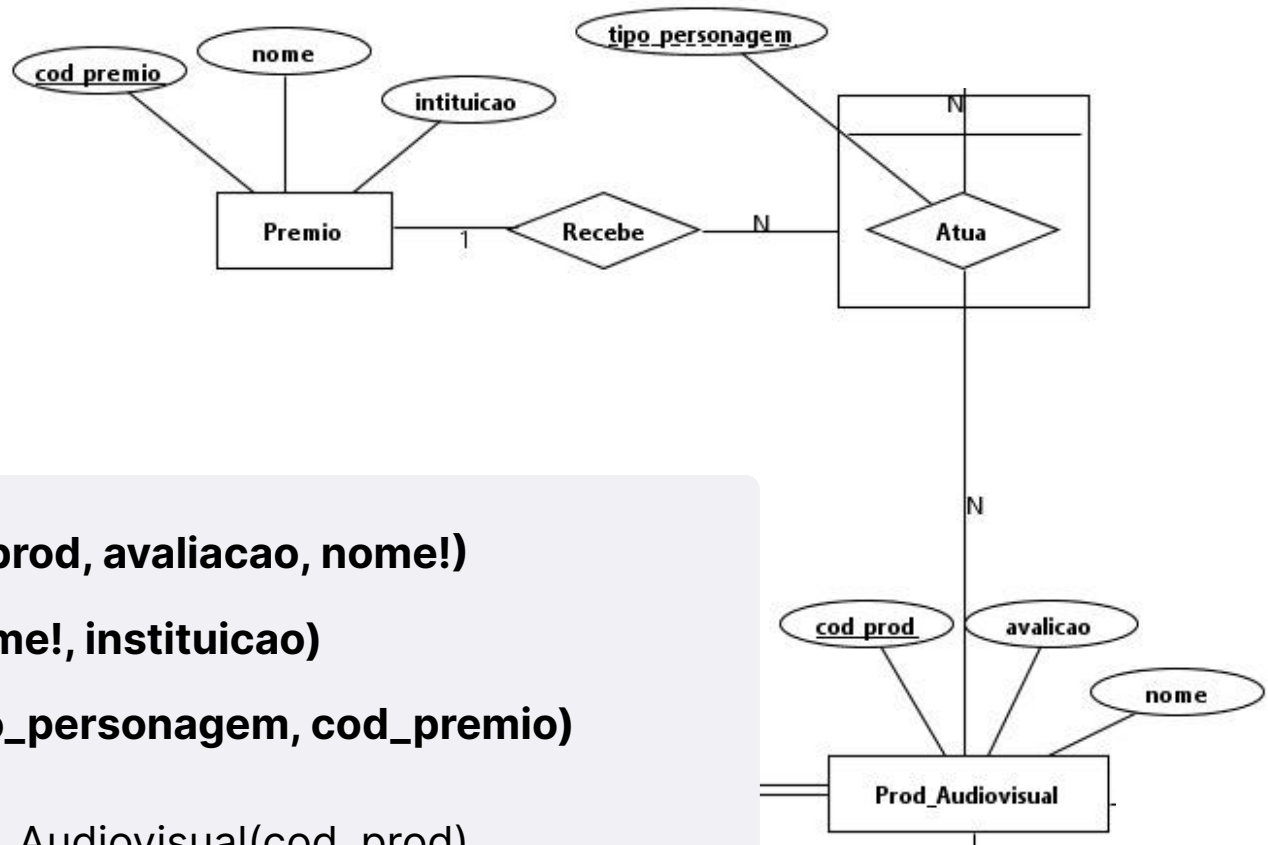
**Produtor (email, nome, chefe, tel\_1, tel\_2)**

chefe → Produtor (email)

**Contrata (email, cpf)**

email → Produtor (email)

cpf → Ator (cpf)



**Prod\_Audiovisual (cod\_prod, avaliacao, nome!)**

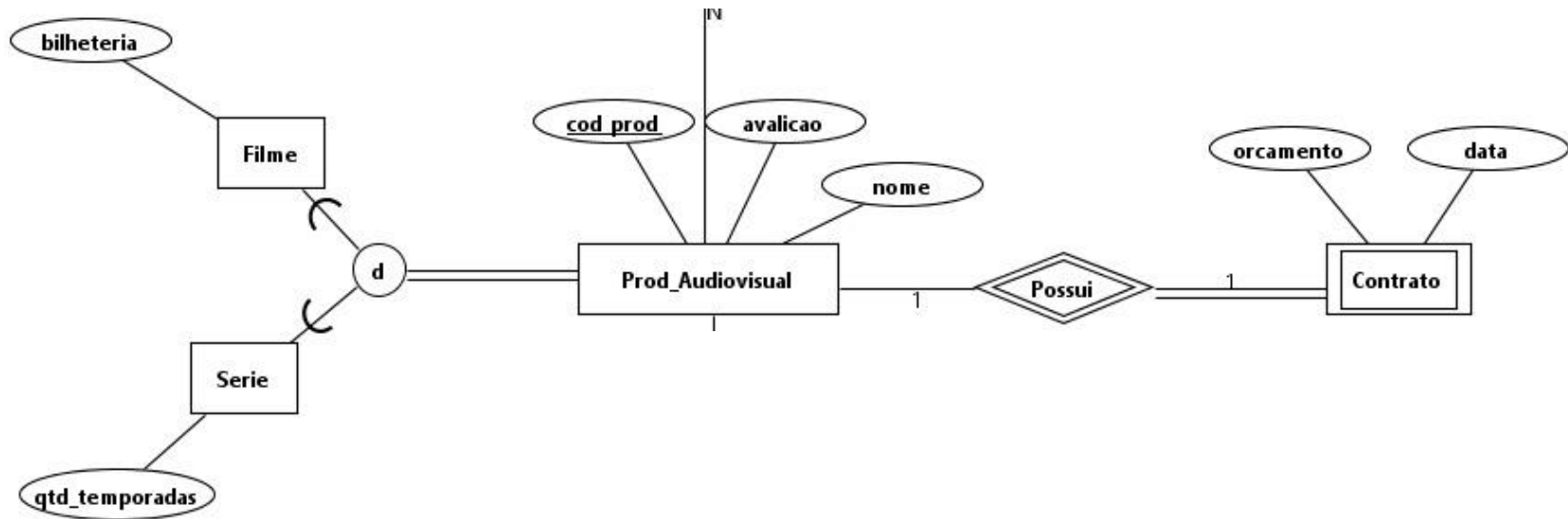
**Premio (cod\_premio, nome!, instituicao)**

**Atua (cpf, cod\_prod, tipo\_personagem, cod\_premio)**

cpf → Ator(cpf)

codigo\_prod → Prod\_Audiovisual(cod\_prod)

cod\_premio → Premio(cod\_premio)



### **Serie (cod\_prod, qtd\_temporadas!)**

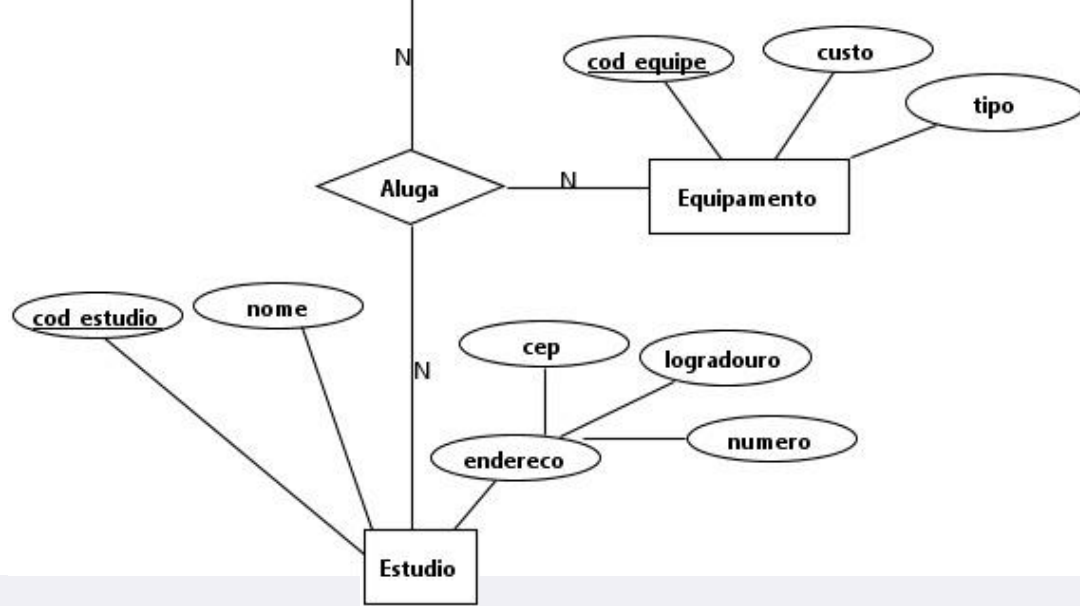
cod\_prod → Prod\_Audiovisual(cod\_prod)

### **Filme (cod\_prod, bilheteria!)**

cod\_prod → Prod\_Audiovisual(cod\_prod)

### **Contrato (cod\_prod, data, orcamento)**

cod\_prod → Prod\_Audiovisual(cod\_prod)



**Equipamento (cod Equip, custo!, tipo)**

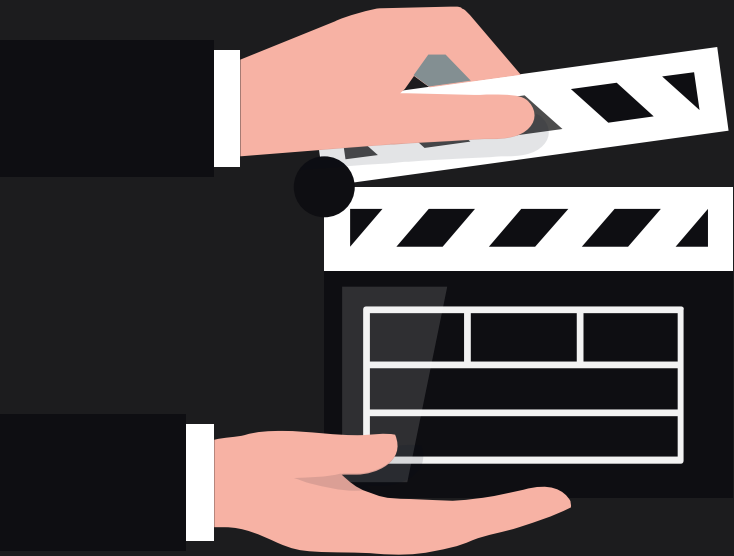
**Estudio (cod estudio, nome!, cep, logradouro, numero)**

**Aluga (cod\_prod, cod\_estudio, cod\_equipe)**

cod\_prod → ProdAudiovisual(cod\_prod)

cod\_estudio → Estudio(cod\_estudio)

cod\_equipe → Equipamento(cod\_equipe)



04

Projeto Físico



# Subconsulta Escalar

Projetar o nome do produtor e a quantidade de atores que ele contratou em ordem alfabética.

```
SELECT P.nome AS NOME_PRODUTOR,  
       (SELECT COUNT(*) FROM CONTRATA C  
        WHERE C.EMAIL = P.EMAIL) AS  
       QTD_ATORES_CONTRATADOS  
FROM PRODUTOR P  
ORDER BY P.NOME ASC;
```

NOME_PRODUTOR	QTD_ATORES_CONTRATADOS
Alexander Ramos	0
Amanda Wilson	0
Bob Smith	1
Chefe da Produção	2
David Choi	0
Jane Doe	2
John Doe	2

# Subconsulta Linha

Projetar o nome dos estúdios que estão localizados no mesmo endereço que o estúdio MGM

```
SELECT NOME
FROM ESTUDIO
WHERE (CEP, LOGRADOURO) = (
    SELECT CEP, LOGRADOURO
    FROM ESTUDIO
    WHERE NOME = 'MGM'
);
```

NOME
MGM
Miramax

# Subconsulta Tabela

Projetar os produtores já contrataram atores premiados em papéis principais.

```
SELECT *
FROM PRODUTOR
WHERE EMAIL IN (
  SELECT EMAIL
  FROM CONTRATA
  WHERE CPF IN (
    SELECT CPF
    FROM ATUA
    WHERE TIPO_PERSONAGEM = 'Protagonista' AND COD_PREMIO IS NOT NULL
  )
);
```

EMAIL	NOME	CHEFE	TEL_1	TEL_2
chefe@email.com	Chefe da Produção	-	11987654321	11987654322
produtor1@email.com	Produtor 1	chefe@email.com	11987654323	11987654324
produtor2@email.com	Produtor 2	chefe@email.com	11987654325	11987654326
johndoe@example.com	John Doe	-	5555555555	-
janedoe@example.com	Jane Doe	johndoe@example.com	5555555556	-
michaelbrown@example.com	Michael Brown	janedoe@example.com	5555555559	-

# Junção interna

Projetar nome do ator, do prêmio e quantidade de vezes que ganhou aquele prêmio.

```
SELECT A.NOME, PR.NOME AS PREMIO, COUNT(*) AS VEZES_QUE_GANHOU
FROM
  ATOR A INNER JOIN ATUA AU
    ON A.CPF = AU.CPF
  INNER JOIN PREMIO PR
    ON AU.COD_PREMIO = PR.COD_PREMIO
GROUP BY A.NOME, PR.NOME;
```

Nome	Prêmio	Qtd_Prêmios
Fulano Silva	Melhor Ator Principal de drama	1
Fulano Silva	Melhor Ator Principal de Terror	1
Beltrano Souza	Melhor Atriz Principal de comédia	1

# Junção interna

Projetar dados do filme, do ator e do prêmio que o ator ganhou no filme.

```
SELECT to_char(PA.AVALIACAO, '0.0') AS NOTA, PA.NOME, ATR.CPF, ATR.NOME, PR.NOME AS PREMIO, PR.INSTITUICAO
FROM Prod_Audiovisual PA
      INNER JOIN ATUA A ON (PA.COD_PROD = A.COD_PROD)
      INNER JOIN PREMIO PR ON (A.COD_PREMIO = PR.COD_PREMIO)
      INNER JOIN ATOR ATR ON (A.CPF = ATR.CPF)
ORDER BY NOTA DESC;
```

Avaliação	Filme	CPF	Nome	Prêmio	Instituição
9.9	Senhor dos anéis	12345678901	Fulano Silva	Melhor Ator Principal	Oscar: Academy of Motion Picture Arts and Sciences
9.2	O Poderoso Chefão	23456789012	Beltrano Souza	Melhor Atriz Principal de Terror	Tony Award: American Theatre Wing
9.2	O Poderoso Chefão	34567890123	Ciclano Santos	Melhor Atriz Principal de drama	Golden Globe: Hollywood Foreign Press Association
9.2	O Poderoso Chefão	00011122233	Paulo Henrique	Melhor Ator Principal	Oscar: Academy of Motion Picture Arts and Sciences

# Junção externa, interna e à esquerda

Projetar dados do ator, da atuação, da produção audiovisual e do produtor.

```
SELECT
  A.NOME,
  AU.TIPO_PERSONAGEM,
  to_char(PA.AVALIACAO, '0.0') AS NOTA,
  PA.NOME AS PRODUCAO,
  PR.EMAIL,
  PR.NOME AS PRODUTOR
FROM ATOR A FULL OUTER JOIN CONTRATA C ON A.CPF = C.CPF
  INNER JOIN PRODUTOR PR ON C.EMAIL = PR.EMAIL
  LEFT JOIN ATUA AU ON A.CPF = AU.CPF
  LEFT JOIN PROD_AUDIOVISUAL PA ON AU.COD_PROD = PA.COD_PROD
WHERE PA.COD_PROD IS NOT NULL;
```

NOME	TIPO_PERSONAGEM	NOTA	PRODUCAO	EMAIL	PRODUTOR
Fulano Silva	Protagonista	9.9	Senhor dos anéis	chefe@email.com	Chefe da Produção
Fulana Barbosa	Coadjuvante	7.3	Bob esponja: No fundo do mar	chefe@email.com	Chefe da Produção
Beltrano Souza	Antagonista	7.2	Amnésia	produtor1@email.com	Produtor 1

# Operação de conjunto

Projetar o nome dos atores que já participaram de filmes e de séries

```
SELECT ATOR.NOME
FROM ATOR
  JOIN ATUA ON ATOR.CPF = ATUA.CPF
  JOIN PROD_AUDIOVISUAL ON ATUA.COD_PROD =
PROD_AUDIOVISUAL.COD_PROD
  JOIN FILME ON PROD_AUDIOVISUAL.COD_PROD = FILME.COD_PROD
UNION
SELECT ATOR.NOME
FROM ATOR
  JOIN ATUA ON ATOR.CPF = ATUA.CPF
  JOIN PROD_AUDIOVISUAL ON ATUA.COD_PROD =
PROD_AUDIOVISUAL.COD_PROD
  JOIN SERIE ON PROD_AUDIOVISUAL.COD_PROD = SERIE.COD_PROD;
```

NOME_ATOR
Ana Paula Costa
Beltrano Souza
Beltrão de Souza
Carlos Roberto
Ciclano Santos
Fulana Barbosa

# Group By + Having

Projetar os atores que trabalharam em pelo menos duas produções com avaliação maior ou igual a 7.0.

```
SELECT A.NOME, COUNT(AU.COD_PROD) AS QTD_PRODUCOES
FROM ATOR A
      INNER JOIN ATUA AU ON A.CPF = AU.CPF
      INNER JOIN PROD_AUDIOVISUAL PA ON AU.COD_PROD = PA.COD_PROD
WHERE PA.AVALIACAO >= 7.0
GROUP BY A.NOME
HAVING COUNT(AU.COD_PROD) >= 2;
```

NOME	QTD_PRODUCOES
Ciclano Santos	5
Roberta Santos	2
Fulana Barbosa	3
Paulo Henrique	4
Beltrano Souza	4
Fulano Silva	3
Carlos Roberto	3
Sil Vesteres Ta Lone	5



# SEMI-JOIN - EXISTS

Projetar o código e nome da produção audiovisual que possui o ator com cpf “12345678901”.

```
SELECT p.cod_prod, p.nome
FROM Prod_Audiovisual p
WHERE EXISTS (
    SELECT *
    FROM Atua a
    WHERE a.cpf = '12345678901'
    AND a.cod_prod = p.cod_prod
);
```

COD_PROD	NOME
1	Senhor dos anéis
8	Pior série do mundo: Uma série inspirada no filme
10	A Lista de Schindler
11	Forrest Gump: O Contador de Histórias
18	The Emoji Movie

# ANTI-JOIN - NO EXISTS

Projetar o nome de todos os atores que não foram premiados

```
SELECT NOME
FROM ATOR AT
WHERE NOT EXISTS (
    SELECT *
    FROM ATUA AU
    WHERE AT.CPF = AU.CPF
    AND AU.COD_PREMIO IS NULL
);
```

NOME
Fulana Barbosa
Fulano Silva
João Santos

# Case

Projetar a opinião sobre a produção em excelente, bom, regular, ruim ou péssimo.

```
SELECT to_char(AVALIACAO, '0.0') AS NOTA, NOME,  
CASE  
  WHEN AVALIACAO >= 8.5 THEN 'EXCELENTE'  
  WHEN AVALIACAO >= 6.5 THEN 'BOM'  
  WHEN AVALIACAO >= 5 THEN 'REGULAR'  
  WHEN AVALIACAO >= 3 THEN 'RUIM'  
  ELSE 'PESSIMO'  
END AS OPINIAO  
FROM Prod_Audiovisual  
ORDER BY NOTA DESC;
```

NOTA	NOME	OPINIAO
9.9	Senhor dos anéis	EXCELENTE
8.9	A Lista de Schindler	EXCELENTE
8.0	Rocky Balboa	BOM
2.7	Cats	PESSIMO

# Limitar linhas

Projetar as 10 primeiras linhas de atores que atuaram em uma determinada produção audiovisual e o tipo de personagem que interpretaram

```
SELECT A.NOME, AU.TIPO_PERSONAGEM, PA.NOME AS PRODUCAO_AUDIOVISUAL
FROM ATOR A
  INNER JOIN ATUA AU ON A.CPF = AU.CPF
  INNER JOIN PROD_AUDIOVISUAL PA ON AU.COD_PROD = PA.COD_PROD
FETCH FIRST 10 ROWS ONLY;
```

Ou

```
SELECT A.NOME, AU.TIPO_PERSONAGEM, PA.NOME AS PRODUCAO_AUDIOVISUAL
FROM ATOR A
  INNER JOIN ATUA AU ON A.CPF = AU.CPF
  INNER JOIN PROD_AUDIOVISUAL PA ON AU.COD_PROD = PA.COD_PROD
WHERE ROWNUM <= 10;
```

NOME	TIPO_PERSONAGEM	PRODUCAO_AUDIOVISUAL
Fulano Silva	Protagonista	Senhor dos anéis
Ana Paula Costa	Coadjuvante	Senhor dos anéis
Carlos Roberto	Coadjuvante	Senhor dos anéis
Fulana Barbosa	Coadjuvante	Bob esponja: No fundo do mar
Ciclano Santos	Protagonista	Bob esponja: No fundo do mar
Carlos Roberto	Protagonista	Bob esponja: No fundo do mar
Beltrano Souza	Antagonista	Amnésia
João Santos	Protagonista	Amnésia
Roberta Santos	Protagonista	Amnésia
Ciclano Santos	Coadjuvante	Batman vs Superman

# SUM() e Junção externa

Projetar a produção audiovisual, o orçamento inicial, custo pago em aluguéis e o orçamento restante

```
SELECT P.NOME, C.ORCAMENTO, NVL(SUM(E.CUSTO), 0) AS CUSTO_ALUGUEL, (C.ORCAMENTO - NVL(SUM(E.CUSTO), 0)) AS ORCAMENTO_RESTANTE
FROM PROD_AUDIOVISUAL P
FULL OUTER JOIN ALUGA A ON A.COD_PROD = P.COD_PROD
FULL OUTER JOIN EQUIPAMENTO E ON E.COD_EQUIP = A.COD_EQUIP
FULL OUTER JOIN CONTRATO C ON C.COD_PROD = P.COD_PROD
WHERE C.COD_PROD IS NOT NULL
GROUP BY P.NOME, C.ORCAMENTO;
```

NOME	ORCAMENTO	CUSTO_ALUGUEL	ORCAMENTO_RESTANTE
Bob esponja: No fundo do mar	2000000	7700	1992300
Amnésia	3000000	15900	2984100
Forrest Gump: O Contador de Histórias	10000000000000	12100	9999999987900
A Origem	10000000000000	5200	9999999994800
The Emoji Movie	10000000000000	6500	9999999993500
Um Sonho de Liberdade	10000000000000	7700	9999999992300

# PL/SQL

Função para retornar a quantidade de atores em uma produção audiovisual

```
CREATE OR REPLACE FUNCTION count_atores_in_prod_audiovisual(  
    p_cod_prod INT  
) RETURN INT AS  
    v_count INT;  
BEGIN  
    SELECT COUNT(*)  
    INTO v_count  
    FROM Atua  
    WHERE cod_prod = p_cod_prod;  
  
    RETURN v_count;  
EXCEPTION  
    WHEN OTHERS THEN  
        ROLLBACK;  
        RAISE;  
END count_atores_in_prod_audiovisual;
```

```
SELECT cod_prod,  
count_atores_in_prod_audiovisual(cod_prod)  
AS qtd_atores  
FROM Prod_Audiovisual;
```

# PL/SQL

## Procedimento para adicionar um novo equipamento

```
CREATE OR REPLACE PROCEDURE add_equipamento(  
    p_cod_equip INT,  
    p_custo FLOAT,  
    p_tipo VARCHAR2  
) AS  
BEGIN  
    INSERT INTO Equipamento (cod_equip, custo, tipo)  
    VALUES (p_cod_equip, p_custo, p_tipo);  
    COMMIT;  
EXCEPTION  
    WHEN OTHERS THEN  
        ROLLBACK;  
        RAISE;  
END add_equipamento;
```

```
BEGIN  
    add_equipamento(  
        p_cod_equip => 100,  
        p_custo => 1500,  
        p_tipo => 'Câmera Tekpix'  
    );  
    DBMS_OUTPUT.PUT_LINE('Equipamento adicionado com sucesso.');
```

EXCEPTION

```
    WHEN OTHERS THEN  
        DBMS_OUTPUT.PUT_LINE('Erro ao adicionar o equipamento: ' ||  
SQLERRM);  
END;
```

# PL/SQL

## Procedimento para adicionar um novo estúdio

```
CREATE OR REPLACE PROCEDURE add_estudio(
    p_cod_estudio INT,
    p_nome VARCHAR2,
    p_cep VARCHAR2,
    p_logradouro VARCHAR2,
    p_numero VARCHAR2
) AS
BEGIN
    INSERT INTO Estudio (cod_estudio, nome, cep,
        logradouro, numero)
    VALUES (p_cod_estudio, p_nome, p_cep, p_logradouro,
        p_numero);
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END add_estudio;
```

```
BEGIN
    add_estudio(
        p_cod_estudio => 100,
        p_nome        => 'Sony Pictures Entertainment',
        p_cep          => '56789012',
        p_logradouro   => 'Avenida Hollywood',
        p_numero       => '200'
    );
    DBMS_OUTPUT.PUT_LINE('Estúdio adicionado com sucesso.');
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao adicionar o estúdio: ' || SQLERRM);
END;
```



# PL/SQL

## Procedimento para transferir um ator de um produtor para outro

```
CREATE OR REPLACE PROCEDURE transfer_ator(
    p_cpf_ator VARCHAR2,
    p_email_antigo_produtores VARCHAR2,
    p_email_novo_produtores VARCHAR2
) AS
    v_rows_affected INT;
BEGIN
    UPDATE Contrata
    SET email = p_email_novo_produtores
    WHERE cpf = p_cpf_ator AND email = p_email_antigo_produtores;

    -- Obter o número de linhas afetadas pela atualização
    v_rows_affected := SQL%ROWCOUNT;

    IF v_rows_affected = 0 THEN
        -- Lançar exceção genérica se nenhuma linha for afetada
        RAISE_APPLICATION_ERROR(-20001, 'Nenhuma linha afetada.
        Verifique os dados fornecidos.');
```

```
    END IF;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END transfer_ator;
```

```
BEGIN
    -- Chamar o procedimento passando o CPF do ator, o email do
    produtor atual e o email do novo produtor
    transfer_ator('66677788899', 'michaelbrown@example.com',
    'janedoe@example.com');
```

```
    -- Exibir mensagem de sucesso
    DBMS_OUTPUT.PUT_LINE('Ator transferido com sucesso.');
```

```
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao transferir o ator: ' ||
        SQLERRM);
END;
```

# PL/SQL

## Procedimento para atualizar custo de um equipamento

```
CREATE OR REPLACE PROCEDURE update_custo_equipamento(
    p_cod_equip INT,
    p_novo_custo FLOAT
) AS
    v_rows_affected INT;
BEGIN
    UPDATE Equipamento
    SET custo = p_novo_custo
    WHERE cod_equip = p_cod_equip;

    -- Obter o número de linhas afetadas pela atualização
    v_rows_affected := SQL%ROWCOUNT;

    IF v_rows_affected = 0 THEN
        -- Lançar exceção genérica se nenhuma linha for afetada
        RAISE_APPLICATION_ERROR(-20001, 'Nenhuma linha afetada. Verifique os dados fornecidos.');
```

END IF;

```
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE;
END update_custo_equipamento;
```

```
BEGIN
    -- Chamar o procedimento passando o código do equipamento e o seu novo custo
    update_custo_equipamento(1, 1500.00);

    -- Exibir mensagem de sucesso
    DBMS_OUTPUT.PUT_LINE('Equipamento atualizado com sucesso.');
```

EXCEPTION

```
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erro ao atualizar equipamento: ' || SQLERRM);
END;
```

Muito Obrigado!

