



**UFS - UNIVERSIDADE FEDERAL DE SERGIPE**  
**CCET - CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA**  
**DCOMP - DEPARTAMENTO DE COMPUTAÇÃO**  
**CIÊNCIA DA COMPUTAÇÃO**

**BANCO DE DADOS**

**ETAPA 2: SQL E TRABALHO PRÁTICO**

**PROF. ANDRÉ BRITTO**

**Ailton de Almeida Rangel Neto – 201500017845**  
**Aryella Machado Lacerda – 201500305484**  
**Raul Oliveira de Andrade - 201500307353**  
**Weslan Rezende Alves - 201500307630**

**São Cristóvão - SE, 2018**

# SUMARIO

<b>Aplicação Lemovie</b>	<b>3</b>
<b>Como executar</b>	<b>3</b>
<b>Consultas ao banco de dados</b>	<b>3</b>

## 1. Aplicação *Lemovie*

Nossa aplicação se chama *Lemovie*, uma espécie de 'rede social' onde os usuários avaliam e se recomendam séries, livros, e filmes.

Criamos uma interface simples para nossa aplicação, onde o usuário; neste caso o usuário precisa ser administrador; faz o login e, caso seu login seja encontrado no banco de dados, é redirecionado a uma tela inicial. Nessa tela, o usuário poderá escolher consultar para visualizar outros usuários, filmes, séries, livros, e indicações, e mais 10 consultas específicas (seguindo as especificações do professor).

As 10 consultas específicas estão em formato de lista, cada um com título referente ao(s) principal(ais) comando(s) SQL utilizado(s) na consulta. Para executar cada consulta, é só clicar no botão à esquerda.

Nosso esquema no banco de dados do professor se chama *lemovie*.

## 2. Como executar

Enviamos o arquivo *lemovie.jar*, ....

Para efetuar o login, use os seguintes dados:

**Usuário:** andrebritto

**Senha:** 12345

## 3. Consultas ao banco de dados

### 3.1. Listar nome e sobrenome de todos os usuário que não são administradores

Para essa consulta, utilizamos somente comandos básicos do SQL: SELECT, FROM, e WHERE. Ficando assim a seguinte consulta:

```
SELECT
    nome,
    sobrenome
FROM
    lemovie.usuario
WHERE nivelacesso = 'USER';
```

### 3.2. Listar nome e ano de todos os filmes

Para essa consulta, além dos comandos básicos do SQL, utilizamos o comando: JOIN ON;

```
SELECT
    nome,
    ano
FROM
    lemovie.midia
    JOIN lemovie.filme ON (idmidia = idfilme);
```

### 3.3. Listar todo comentário feito, o nome de quem o fez, a mídia a qual ele está ligado, e se o usuário que deixou o comentário já viu/leu a mídia

Para essa consulta, além dos comandos básicos do SQL, utilizamos o comando: JOIN ON, USING;

```
SELECT
    u.nome,
    c.texto,
    a.visto,
    m.nome
FROM
    lemovie.usuario u
    JOIN lemovie.comentario c ON(u.login =
c.usuario)
    JOIN lemovie.avaliacao a USING (midia)
    JOIN lemovie.midia m ON (a.midia =
m.idmidia);
```

### 3.4. Listar quantas séries tiveram notas idênticas

Para essa consulta, além dos comandos básicos do SQL, utilizamos os comandos: JOIN ON, GROUP BY, e a função de agregação COUNT();

```

SELECT
    count(*) as qtd_notas,
    n.valor
FROM
    lemovie.midia m
    JOIN lemovie.filme s ON (s.idfilme =
m.idmidia)
    LEFT JOIN lemovie.avaliacao a ON(m.idmidia =
a.midia)
    LEFT JOIN lemovie.nota n ON (a.idavaliacao =
n.idnota)
GROUP BY(valor);

```

### 3.5. Listar login, nome, e sobrenome dos usuários que fizeram mais de um comentário

Para essa consulta, além dos comandos básicos do SQL, utilizamos os comandos: JOIN ON, GROUP BY, HAVING, e a função de agregação COUNT();

```

SELECT
    user_cmm.usuario,
    u.nome,
    u.sobrenome
FROM
    lemovie.usuario u
    JOIN (
        SELECT
            usuario
        FROM
            lemovie.comentario c
        GROUP BY (c.usuario)
        HAVING count(c.usuario) > 1
    ) AS user_cmm ON(user_cmm.usuario =
u.login);

```

### 3.6. Mostrar quem fez um comentário e quem respondeu, caso eles sejam amigos

Para essa consulta, além dos comandos básicos do SQL, utilizamos consultas aninhadas e os comandos: DISTINCT, LEFT JOIN, JOIN ON;

```
SELECT
    respostas_user.user1 AS comentou,
    respostas_user.user2 AS respondeu
FROM
    lemovie.amizade am
    LEFT JOIN (
        SELECT
            DISTINCT c.usuario as USER1,
            c2.usuario as USER2
        FROM
            lemovie.resposta r
            JOIN lemovie.comentario c
            ON(c.idcomentario = r.comentario1)
            JOIN lemovie.comentario c2 ON
            (c2.idcomentario = r.comentario2)
        ) as respostas_user ON(respostas_user.user1
= usuario1 )
    WHERE am.usuario2 = respostas_user.user2;
```

**3.7. Lista o usuário, o nome da mídia, o comentário e se o usuário viu ou não a mídia. Esta lista apresenta somente os usuários que não viram a mídia mas contém a palavra 'adoro' em seu comentário.**

Para essa consulta, além dos comandos básicos do SQL, fazemos uso de consultas aninhadas e os comandos: LEFT JOIN, JOIN ON, AND, IS NOT NULL, ILIKE;

```
SELECT
    av.usuario,
    av.visto,
    comments_user.nome_midia,
    comments_user.texto
FROM
    lemovie.avaliacao av
    LEFT JOIN (
        SELECT
            u.midia,
```

```

        lower(u.texto) as texto,
        m.nome AS nome_midia,
        u.usuario
    FROM
        lemovie.comentario u
        JOIN lemovie.midia m ON (m.idmidia
            = u.midia)
    ) AS comments_user ON ( av.midia =
        comments_user.midia AND av.usuario =
        comments_user.usuario)
WHERE
    (comments_user.midia IS NOT NULL)
    AND (comments_user.texto ILIKE '%adoro%')
    AND visto IS FALSE;

```

### 3.8. Listar a mídia com a maior média, mostrar seu tipo, nome, e qual é a média

Para essa consulta, além dos comandos básicos do SQL, fazemos uso de consultas aninhadas e os comandos: WITH, JOIN ON, GROUP BY, e a função de agregação MAX();

```

WITH media_midia AS (
    SELECT
        av.midia,
        avg(valor) AS media
    FROM
        lemovie.avaliacao av
        JOIN lemovie.nota n ON(n.idnota =
            av.idavaliacao)
    GROUP BY (av.midia)
)

SELECT
    mi.nome,
    mi.tipo,
    mm.media
FROM
    media_midia mm

```

```

        JOIN lemovie.midia mi ON(mm.midia =
mi.idmidia )
WHERE mm.media = (
        SELECT
            max(mm.media)
        FROM media_midia mm
    );

```

### 3.9. Listar as mídias de 1970 até 1999 e quantas pessoas viram/leram cada uma das mídias

Para essa consulta, além dos comandos básicos do SQL, utilizamos os comandos: GROUP BY, ORDER BY, BETWEEN, JOIN ON;

```

SELECT
    m.ano,
    m.nome,
    count(av.visto) AS quantos_viram
FROM
    lemovie.midia m
    JOIN lemovie.avaliacao av ON(idmidia =
midia)
WHERE
    ano BETWEEN '1970' AND '1999'
GROUP BY(idmidia)
ORDER BY (ano);

```

### 3.10. Listar o nome dos usuários e a quantidade de horas de filme que eles assistiram

Para essa consulta, além dos comandos básicos do SQL, fazemos uso de consultas aninhadas, operações aritméticas, e os comandos: LEFT JOIN, JOIN ON, GROUP BY;

```

SELECT
    u.nome,
    sum(dur_mid.duracao) AS ASSISTIU
FROM

```



```

        lemovie.usuario u
    JOIN lemovie.avaliacao a ON(u.login =
a.usuario)
    LEFT JOIN (
        SELECT
            m.idmidia,
            f.duracao,
            nome
        FROM
            lemovie.midia m
            JOIN lemovie.filme f ON (m.idmidia
            = f.idfilme)
        ) AS dur_mid ON(dur_mid.idmidia = a.midia )
WHERE visto IS TRUE
GROUP BY (login);

```