

**INSTITUTO  
FEDERAL**  
Santa Catarina

# Banco de Dados 2

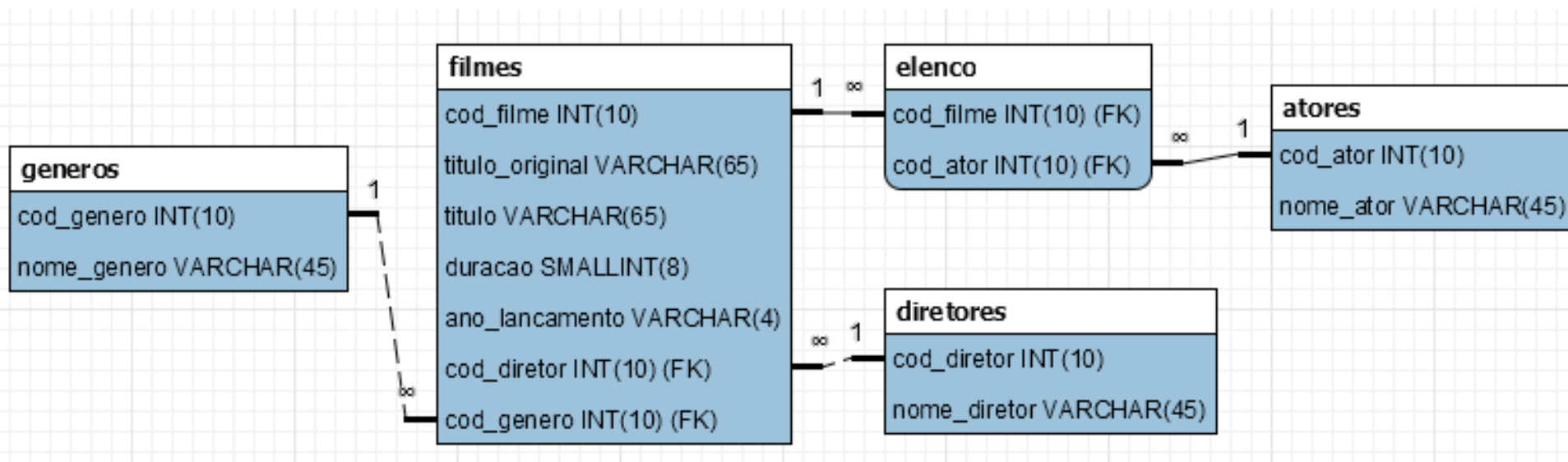


- 1) Revisão SQL DML (insert, update delete);
- 2) Revisão SQL DQL
- 3) Operadores de comparação e lógicos;
- 4) Operações com Strings



# Materiais desta aula

- Para esta aula precisaremos do BD “**filmes**”, cuja estrutura está representada abaixo.



# Materiais desta aula

- Baixe o BD filmes no SIGAA
- No shell do Mysql:  
# mysql -u root -p  
# create database filmes;  
# exit;  
# mysql -u root -p filmes < filmes.sql

1/4

# Revisão SQL DML

Data Manipulation Language



# O comando INSERT

**INSERT INTO** **tabela** (lista atributos)

**VALUES** (lista de valores)

Exemplo:

**INSERT INTO filmes** (cod\_filme, titulo\_original, titulo, duracao, ano\_lancamento, cod\_diretor, cod\_genero)

**VALUES** (143, 'Pretty Woman', 'Uma Linda Mulher', NULL, '1990', 76, 6);

# O comando UPDATE

**UPDATE** *tabela*

**SET** *atributo*=valor1, *atributo2*=valor2

**WHERE** *condicao*;

Exemplo:

**UPDATE** *filmes*

**SET** *cod\_genero*=10, *duracao*=119

**WHERE** *cod\_filme*=143;

# O comando DELETE

**DELETE FROM** *tabela* **WHERE** condicao

Exemplo:

**DELETE FROM** *filmes* **WHERE** *cod\_filme*=143



# Revisando conceitos

- Operações de SQL DML podem violar alguma(s) das restrições de integridade abaixo:
  - **Integridade de domínio:** Especificam que o valor de um campo deve obedecer a definição de valores admitidos para a coluna (tipos).
  - **Integridade de vazio:** Especifica se um campo pode ou não ser vazio.
  - **Integridade de Chave:** Os valores da chave primária devem ser únicos.
  - **Integridade Referencial:** É a restrição que define que os valores dos campos que aparecem em uma chave estrangeira devem aparecer na chave primária da tabela referenciada.

# Revisando conceitos

- Considerando os comandos de SQL: **INSERT**, **UPDATE** e **DELETE**, quais restrições cada comando pode violar?
  - **Integridade de domínio**
  - **Integridade de vazio**
  - **Integridade de Chave**
  - **Integridade Referencial**



As vezes você verá nomes de tabelas e campos com **backticks**.  
Qual o motivo?

```
INSERT INTO `diretores` (`cod_diretor`, `nome_diretor`)  
VALUES (73, 'Christopher Nolan');
```



## Você já ouviu falar de Sql INJECTION?

Leia mais em: [https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp)



## Outros comandos para pesquisar...

- **REPLACE:** O equivalente a DELETE + INSERT, ou só INSERT se não encontrar a linha
- **TRUNCATE TABLE:** apaga (DROP) e recria a tabela



Cadastre algum filme que você goste que tenha sido lançado nos últimos 5 anos.

## Dicas

- Verifique se os atores, diretor e gênero já estão cadastrados
- Não esqueça de vincular o elenco com o filme
- Caso não tenha todas informações, verifique quais campos aceitam NULL na tabela filmes

2/4

# Revisão SQL DQL

Data Query Language



# SQL e a cláusula **SELECT**

O comando **SELECT** realiza uma pesquisa no banco de dados cujo propósito é recuperar as linhas (registros) que atendam as condições apresentadas.

Um **SELECT** é capaz de fazer uma consulta simples, composta, com junções (diferentes tabelas), com operações que alteram valores para a exibição, com funções, com agrupamentos ou até mesmo sobre outras consultas.

**SELECT \* FROM** *tabela*;

**SELECT** *campo1*, *campo2*, *campo3* **FROM** *tabela*;



## Consultas utilizando mais de uma tabela

- - exibe o título do filme e o nome do respectivo diretor

```
SELECT titulo, nome_diretor  
FROM filmes, diretores  
WHERE filmes.cod_diretor = diretores.cod_diretor
```

- - ou com apelidos para tabelas

```
SELECT titulo, nome_diretor  
FROM filmes f, diretores d  
WHERE f.cod_diretor = d.cod_diretor
```

3/4

# Operadores de comparação e lógicos



# Principais Operadores de comparação

>	Maior
>=	Maior ou igual
<	Menor
<=	Menor ou igual
!=	Não igual
=	Igual

# Principais Operadores de comparação

<b>BETWEEN</b>	Entre um valor e outro
<b>NOT BETWEEN</b>	Não está entre um valor e outro
<b>IS</b>	Valor booleano
<b>IS NOT</b>	Valor booleano
<b>IN</b>	interseção
<b>NOT IN</b>	diferença
<b>IS NULL</b>	É nulo
<b>IS NOT NULL</b>	Não é nulo

# Operadores lógicos

AND	&&	e
OR		ou
XOR		Ou exclusivo
NOT	!	Não

# Algumas cláusulas e operadores

- - elimina linhas iguais

```
SELECT DISTINCT ano_lancamento FROM filmes
```

- - nome temporário para coluna

```
SELECT titulo, ano_lancamento AS 'Ano de lançamento' FROM filmes
```

- - atende um critério

```
SELECT titulo, ano_lancamento  
FROM filmes  
WHERE ano_lancamento >= 2000
```

# Algumas cláusulas e operadores

-- filmes que não são do ano 2000

```
SELECT titulo, ano_lancamento  
FROM filmes  
WHERE ano_lancamento <> 2000
```

-- ou

```
SELECT titulo, ano_lancamento  
FROM filmes  
WHERE ano_lancamento != 2000
```

# Algumas cláusulas e operadores

- - filmes que foram lançados entre 2000 e 2005

**SELECT** titulo, ano\_lancamento

**FROM** filmes

**WHERE** ano\_lancamento **BETWEEN** 2000 **AND** 2005



# Algumas cláusulas e operadores

- - filmes que **NÃO** foram lançados entre 2000 e 2005

**SELECT** titulo, ano\_lancamento

**FROM** filmes

**WHERE** ano\_lancamento **NOT BETWEEN** 2000 **AND** 2005

## Consultas utilizando mais de uma tabela

Exibe o título do filme e o nome do respectivo diretor e respectivo gênero

```
SELECT f.titulo, d.nome_diretor, g.nome_genero  
FROM filmes f, diretores d, generos g  
WHERE f.cod_diretor = d.cod_diretor  
AND f.cod_genero=g.cod_genero
```

## Consultas utilizando mais de uma tabela

Exibe o título do filme e o nome do respectivo diretor somente dos filmes de Suspense

```
SELECT f.titulo, d.nome_diretor
FROM filmes f, diretores d, generos g
WHERE f.cod_diretor = d.cod_diretor
      AND f.cod_genero=g.cod_genero
      AND g.nome_genero='Suspense'
```

# Ordenação de resultados

Exibe os filmes ordenados pelo ano de lançamento (mais antigo para mais novo)

```
SELECT titulo, ano_lancamento  
FROM filmes  
ORDER BY ano_lancamento
```

## Ordenação de resultados

Exibe os filmes ordenados pelo ano de lançamento (mais antigo para mais novo)

Removendo campos nulos ou vazios

```
SELECT titulo, ano_lancamento
```

```
FROM filmes
```

```
WHERE ano_lancamento IS NOT NULL AND ano_lancamento != ""
```

```
ORDER BY ano_lancamento
```


# Ordenação de resultados

Exibe os filmes ordenados pelo ano de lançamento (mais novos primeiros)

ASC ascendente

DESC decendente

```
SELECT titulo, ano_lancamento  
FROM filmes  
ORDER BY ano_lancamento DESC
```



# Limitando o número de resultados

Exibe os 3 maiores filmes do catálogo

```
SELECT titulo, duracao, ano_lancamento  
FROM filmes  
ORDER BY duracao DESC  
LIMIT 3
```



## Limitando o número de resultados

Exibe os 3º filme com maior duração

```
SELECT titulo, duracao, ano_lancamento  
FROM filmes  
ORDER BY duracao DESC  
LIMIT 2,1
```

A partir do segundo resultado, busque uma linha



## Limitando o número de resultados

Exibe os 3º, 4º e 5º filme com maior duração

```
SELECT titulo, duracao, ano_lancamento  
FROM filmes  
ORDER BY duracao DESC  
LIMIT 2,3
```

A partir do segundo resultado, busque três linhas

Curiosidade: esse modo é muito utilizado para fazer sistemas de paginação

# Operadores lógicos

Exibe os filmes de suspense ou drama

```
SELECT f.titulo, g.nome_genero  
FROM filmes f, generos g  
WHERE f.cod_genero=g.cod_genero  
        AND (g.nome_genero='Suspense' OR g.nome_genero='Drama')
```

Atenção com a precedência dos operadores

4/4

# Operações com Strings



# Funções para Strings

- Existem dezenas de funções para Strings nativas nos SGBDs
- A maioria simples e auto explicativos como:
  - TRIM
  - LTRIM
  - RTRIM
  - CONCAT
  - SUBSTRING
- Fonte: <https://mariadb.com/kb/en/string-functions/>

# Operações com Strings

- Além das funções, um operador de comparação de strings muito útil é o **LIKE**, que é utilizado para procurar um padrão em uma coluna.
- Utiliza-se com % ou \_
  - % combina com qualquer substring, independente do tamanho.
  - \_ combina com qualquer caractere.

# Operações com Strings

Exibe os filmes cujo título original começam com “the”

```
SELECT titulo_original  
FROM filmes  
WHERE titulo_original LIKE 'The%'
```

# Operações com Strings

Exibe os filmes cujo título terminam com “vivo”

**SELECT** titulo

**FROM** filmes

**WHERE** titulo **LIKE** '%vivo'

# Operações com Strings

Exibe os filmes cujo título contenha a palavra “rei”

**SELECT** titulo

**FROM** filmes

**WHERE** titulo **LIKE** '%rei%'



# Testando seus conhecimentos

Lista 1

