

## Atividade de Aula – Trabalho Prático

<b>Disciplina</b>	<b>SQL – Aplicações com Linguagem SQL</b>
<b>Nome do Aluno</b>	<b>Ivan Madeira de Oliveira</b>

### Objetivos

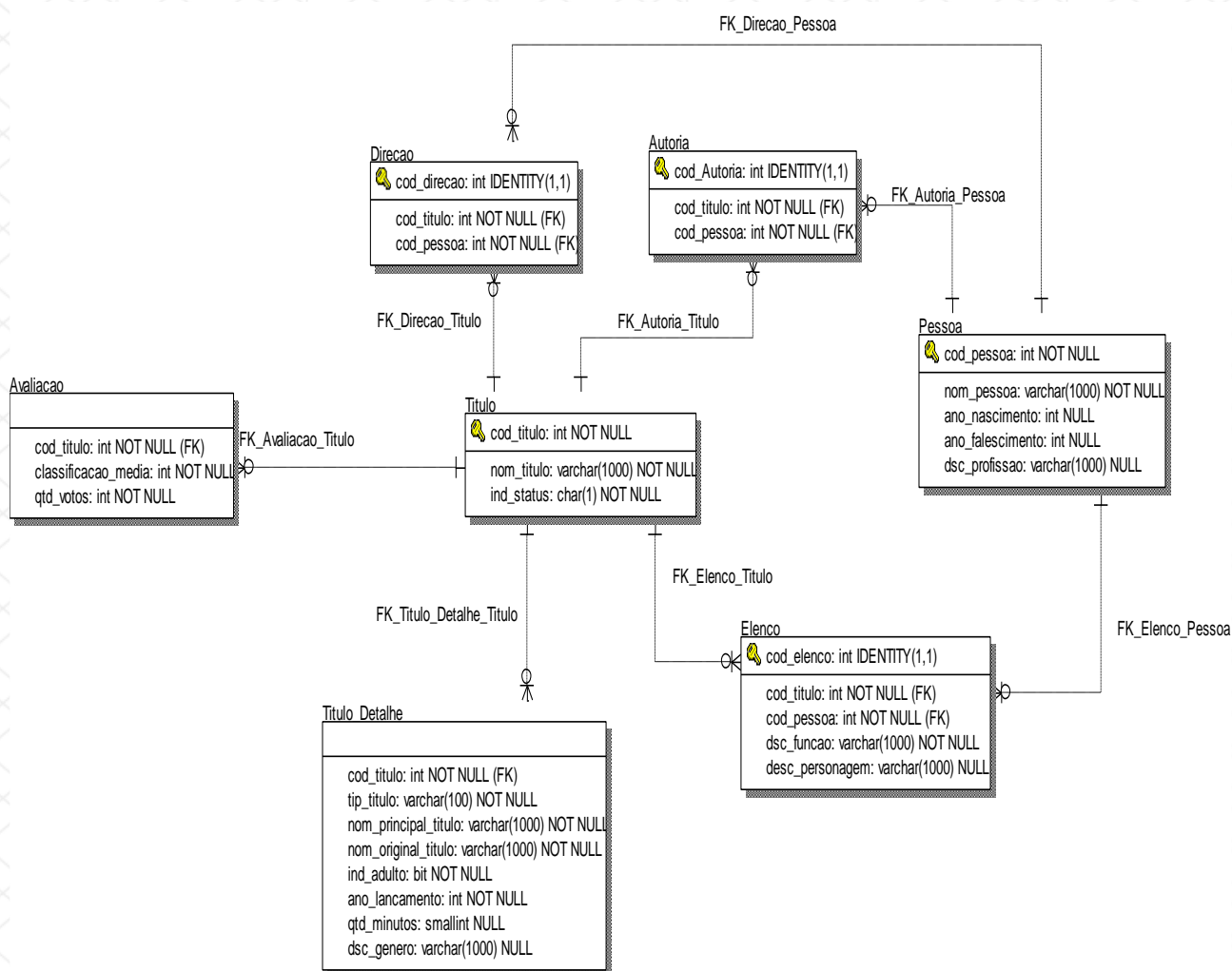
Exercitar os seguintes conceitos vistos em aula:

- ✓ Modelo de dados relacional.
- ✓ Banco de dados relacional.
- ✓ Comandos DDL da Linguagem SQL.
- ✓ Comandos DML da Linguagem SQL: UPDATE, DELETE e SELECT.
- ✓ Comandos DCL da Linguagem SQL: GRANT.
- ✓ Carga massiva de dados (operações de bulk load).

## Enunciado

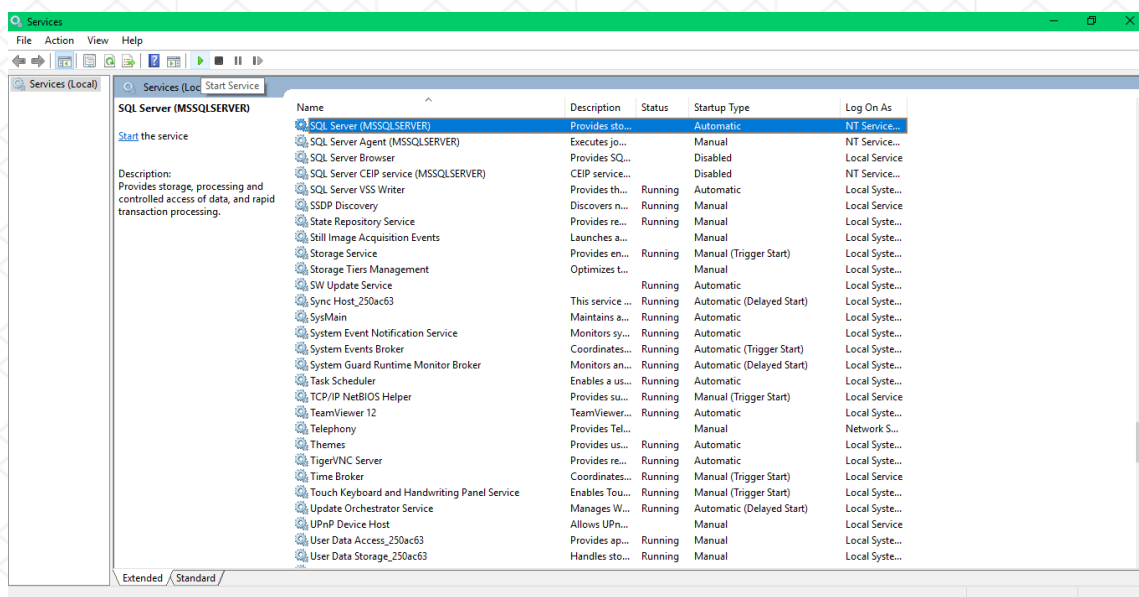
Como informado na Atividade Prévia, uma grande empresa do setor de comércio on-line, necessita construir um sistema para armazenar as informações acerca das produções que serão comercializadas em sua plataforma on-line. Com base no modelo de dados abaixo, que foi ajustado na Atividade Prévia, vocês foram contratados para realizarem as atividades a seguir, como parte da etapa de implantação do projeto. Elas devem ser realizadas no ambiente virtual disponibilizado para o trabalho prático, que já se encontra com todos os arquivos e softwares necessários instalados.

**Para cada atividade, deve-se dar print na tela de execução e colar a imagem no documento do trabalho prático que será entregue. Ao dar print nas telas, não se esqueçam de incluir as informações da sessão e o nome do grupo.**

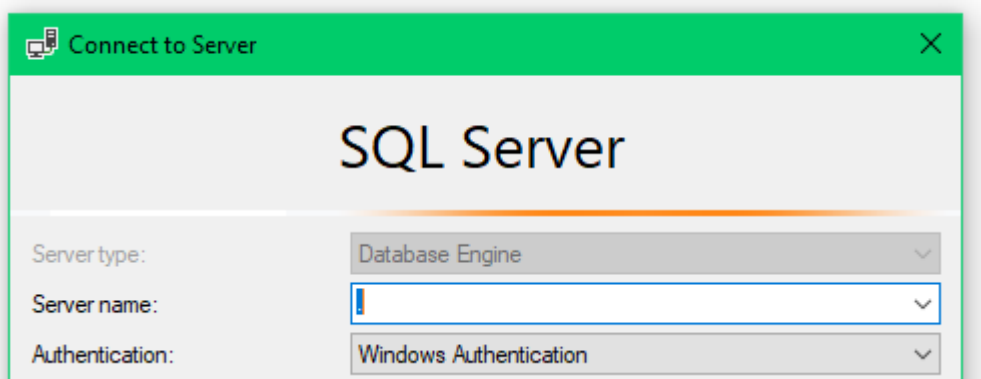


## Atividades

1. Criar os objetos no banco de dados de produção da empresa. **(1 ponto)**
  - a. Fazer o logon na máquina virtual com o IP, usuário e senha fornecidos no fórum do grupo;
  - b. Verificar na console de serviços (atalho → *services.msc*) se o serviço do *SQL Server* está executando e, em caso negativo, iniciá-lo;



- c. Abrir o *SQL Server Management Studio* e conectar à instância local ("."), como mostrado abaixo;



- d. Abrir e executar o script ***BDProducoes\_Trabalho\_Pratico.sql*** que se encontra na pasta ***C:\Scripts***;



- e. Conferir se os objetos foram criados corretamente e **anexar o print do resultado da execução do comando abaixo.**

```
USE BDProducoes
GO
```

```
SELECT name, type_desc, create_date, modify_date
FROM sys.tables
ORDER BY name
GO
```

SQLQuery2.sql - (lo...M. de Oliveira (53))\*

```
GO

SELECT name, type_desc, create_date, modify_date
FROM sys.tables
ORDER BY name
GO
```

100 %

Results Messages

	name	type_desc	create_date	modify_date
1	Autoria	USER_TABLE	2020-10-23 21:42:57.583	2020-10-23 21:44:11.970
2	Avaliacao	USER_TABLE	2020-10-23 21:42:57.590	2020-10-23 21:43:47.807
3	Direcao	USER_TABLE	2020-10-23 21:42:57.590	2020-10-23 21:44:12.070
4	Elenco	USER_TABLE	2020-10-23 21:42:57.593	2020-10-23 21:44:12.267
5	Pessoa	USER_TABLE	2020-10-23 21:42:57.600	2020-10-23 21:42:57.647
6	Titulo	USER_TABLE	2020-10-23 21:42:57.600	2020-10-23 21:43:21.980
7	Titulo_Detalhe	USER_TABLE	2020-10-23 21:42:57.603	2020-10-23 21:43:47.363

2. Realizar a carga dos dados que foram fornecidos pela empresa no formato de arquivo (*flat file*). **(1 ponto)**

- a. Conectado no banco de dados, executar o comando abaixo para carregar cada uma das **7 tabelas**, substituindo o nome do arquivo a ser carregado e o nome da tabela onde a carga deve ser feita. **Lembre-se que como o schema físico foi criado e já contém as restrições de integridade (foreign Keys), deve-se carregar as tabelas na ordem de precedência de existência dos dados (“pais antes dos filhos”);**

```
BULK INSERT BDProducoes.dbo.NOME_DA_TABELA
FROM 'C:\Scripts\NOME_DO_ARQUIVO.txt'
WITH
(
    FIELDTERMINATOR = '\t',
    ROWTERMINATOR = '\n',
    --Manter os valores nos arquivos para colunas com a prop. IDENTITY
    KEEPIDENTITY
);
```

- b. Verificar se todas as 7 tabelas foram carregadas com sucesso, usando o comando abaixo para gerar os comandos DML para serem executados e fazer essa checagem. **Anexar o print da tela com o resultado da execução dos comandos gerados pelo script abaixo;**

```
USE BDProducoes
GO
```

```
SELECT 'SELECT COUNT(*) AS Qtde_Linhas_Tabela_' + name + ' from ' +
name
FROM sys.tables
ORDER BY name
GO
```

SQLQuery2.sql - (lo...M. de Oliveira (53))\*

```
USE BDProducoes
GO
```

```
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Autoria from Autoria
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Avaliacao from Avaliacao
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Direcao from Direcao
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Elenco from Elenco
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Pessoa from Pessoa
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Titulo from Titulo
GO
SELECT COUNT(*) AS Qtde_Linhas_Tabela_Titulo_Detalhe from Titulo_Detalhe
GO
```

100 %

Results Messages

Qtde_Linhas_Tabela_Autoria	4454
Qtde_Linhas_Tabela_Avaliacao	11978
Qtde_Linhas_Tabela_Direcao	17760
Qtde_Linhas_Tabela_Elenco	152899
Qtde_Linhas_Tabela_Pessoa	4064704
Qtde_Linhas_Tabela_Titulo	19614
Qtde_Linhas_Tabela_Titulo_Detalhe	19614

Query executed successfully.

(local) (15.0 RTM)

DESKTOP-OU2CFBJ\Ivan M...

BDProducoes

00:00:00

7 rows



3. A empresa solicitou que você faça uma espécie de “auditoria da qualidade dos dados”. Para isso, você precisa **criar uma visão (views) no banco de dados** para cada uma das situações abaixo. **Anexe a evidência da contagem de linhas retornada por cada view. (2 pontos)**

- **Títulos sem avaliação → VW\_Titulos\_Sem\_Avaliacao**

SQLQuery2.sql - (lo...M. de Oliveira (53))\*

```
CREATE VIEW VW_Titulos_Sem_Avaliacao
AS
SELECT cod_titulo, nom_titulo
FROM Titulo
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Avaliacao
    WHERE classificacao_media = NULL
);

SELECT COUNT(*) AS Qntd_linhas_Titulo_Sem_Avaliacao
FROM VW_Titulos_Sem_Avaliacao;
```

100 %

Results Messages

	Qntd_linhas_Titulo_Sem_Avaliacao
1	0

- Títulos sem o detalhe da duração (qtd\_minutos) e/ou informação do gênero (dsc\_genero) → VW\_Titulos\_Sem\_Detalhes

SQLQuery2.sql - (lo...M. de Oliveira (53))\*

```
CREATE VIEW VW_Titulos_Sem_Detalhes
AS
SELECT cod_titulo, nom_titulo
FROM Titulo
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Titulo_Detalhe
    WHERE qtd_minutos = NULL
    OR dsc_genero = 'NULL'
);

SELECT COUNT(*) AS Qntd_linhas_Titulo_Sem_Detalhes
FROM VW_Titulos_Sem_Detalhes;
```

100 %

Results Messages

	Qntd_linhas_Titulo_Sem_Detalhes
1	0



- Títulos sem autor → VW\_Titulos\_Sem\_Autor

```

CREATE VIEW VW_Titulos_Sem_Autor
AS
SELECT cod_titulo, nom_titulo
FROM Titulo
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Aatoria
    WHERE cod_pessoa = NULL
);

SELECT COUNT(*) AS Qntd_linhas_Titulo_Sem_Autor
FROM VW_Titulos_Sem_Autor;

```

.00 %

Results Messages

	Qntd_linhas_Titulo_Sem_Autor
1	0

- Títulos sem diretor → VW\_Titulos\_Sem\_Diretor

```

CREATE VIEW VW_Titulos_Sem_Diretor
AS
SELECT cod_titulo, nom_titulo
FROM Titulo
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Direcao
    WHERE cod_pessoa = NULL
);

SELECT COUNT(*) AS Qntd_linhas_Titulo_Sem_Diretor
FROM VW_Titulos_Sem_Diretor;

```

100 %

Results Messages

Qntd_linhas_Titulo_Sem_Diretor
0

- Títulos sem elenco → VW\_Titulos\_Sem\_Elenco

SQLQuery2.sql - (lo...M. de Oliveira (53))\*

```
CREATE VIEW VW_Titulos_Sem_Elenco
AS
SELECT cod_titulo, nom_titulo
FROM Titulo
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Elenco
    WHERE cod_pessoa = NULL
);

SELECT COUNT(*) AS Qntd_linhas_Titulo_Sem_Elenco
FROM VW_Titulos_Sem_Elenco;
```

100 %

Results Messages

	Qntd_linhas_Titulo_Sem_Elenco
1	0



4. Com base no trabalho feito no item anterior, a empresa solicitou que você **altere o status dos títulos que estiverem sem a informação do elenco** para o status inativo (`ind_status= 'I'`). **Anexe a evidência da quantidade de títulos ativos (`ind_status= 'A'`) após essa atualização. (1 ponto)**

SQLQuery2.sql - (10...M. de Oliveira (33))

```
UPDATE Titulo
SET ind_status='I'
WHERE cod_titulo IN(
    SELECT cod_titulo FROM Elenco
    WHERE cod_pessoa = NULL
);

SELECT COUNT(*) FROM Titulo
WHERE ind_status = 'A';
```

100 %

Results Messages

	(No column name)
1	19614

5. Um desenvolvedor da empresa está reclamando que a consulta abaixo, para listar as pessoas da base de dados que se encontram vivas, está demorando demais. **Informe qual o motivo dessa demora e uma possível solução para acelerar essa consulta. (2 pontos)**

```
SELECT nom_pessoa, ano_nascimento, ano_falecimento  
FROM Pessoa  
WHERE ano_falecimento IS NULL;
```

Deve ser feita uma análise prévia da tabela em questão e a utilização de índices para reduzir futuros delays no plano de execução.

```
CREATE NONCLUSTERED INDEX [Nome] ON Pessoa.nom_pessoa  
CREATE NONCLUSTERED INDEX [Ano_nsc] ON Pessoa.ano_nascimento  
CREATE NONCLUSTERED INDEX [Ano_fsc] ON Pessoa.ano_falecimento
```

A empresa resolveu fazer uma limpeza na base de dados de pessoas, e solicitou que você **forneça o script DML para excluir da tabela Pessoas, todos aqueles que não forem autor ou diretor de algum título, e que não façam parte do elenco de nenhum título. (2 pontos)**

```
DELETE FROM Pessoa  
WHERE cod_pessoa IN(  
    SELECT cod_pessoa FROM Autoria  
    WHERE cod_pessoa = NULL  
    AND cod_pessoa IN(  
        SELECT cod_pessoa FROM Elenco  
        WHERE cod_pessoa = NULL  
    )  
);  
  
DELETE FROM Pessoa  
WHERE cod_pessoa IN(  
    SELECT cod_pessoa FROM Direcao  
    WHERE cod_pessoa = NULL  
    AND cod_pessoa IN(  
        SELECT cod_pessoa FROM Elenco  
        WHERE cod_pessoa = NULL  
    )  
);
```

A empresa informou que fará uma nova carga das avaliações dos títulos, e pediu que você apagasse todos os registros da tabela Avaliacao (mantendo sua estrutura) antes da carga. **Qual comando DDL** você precisa usar para fazer essa operação da forma mais rápida possível? (1 ponto)

```
DELETE * FROM Avaliacao
```

6. Foi solicitada a você a criação de um objeto como código compilado (uma procedure), para fazer a inserção de dados na tabela Titulos, de forma que os dados sejam passados como parâmetros. **Forneça o código DDL para criação dessa procedure e o exemplo de execução dela. (2 pontos)**

```
CREATE PROCEDURE Insere_em_Titulo  
  
    @cod_titulo,  
    @nom_titulo varchar(1000),  
    @ind_status char(a)  
  
AS  
BEGIN  
  
    INSERT INTO Titulo(cod_titulo, nom_titulo,  
ind_status)  
VALUES (@cod_titulo, @nom_titulo, @ind_status)  
  
END;
```



7. Com base no trabalho feito no item anterior, a empresa solicitou que você **construa o script DCL**, de forma que o usuário da aplicação, de nome **UserApp**, tenha permissão para selecionar, inserir, atualizar e excluir dados de todas as tabelas, mas não tenha permissão de inserção direta na tabela Títulos. **Quais os comandos que devem estar nesse script DCL? (1 ponto)**

```
GRANT ALL ON BDProducoes TO UserApp;
```

```
REVOKE INSERT ON BDProducoes.Titulo TO UserApp;
```

8. Por fim, a empresa solicitou que você desenvolva um relatório (script DML), que **retorne as seguintes colunas** acerca dos títulos **ativos, ordenados alfabeticamente pelo nome** do título. **(2 pontos)**

- **Nome do Título: coluna nom\_titulo da tabela Titulo**

```
SELECT nom_titulo FROM Titulo
```

```
WHERE Titulo.ind_status = 'A'
```

```
ORDER BY nom_titulo ASC;
```

- **Tipo do Título: coluna tip\_titulo da tabela Titulo\_Detalhe (em maiúsculo)**

```
SELECT UPPER(tip_titulo) FROM Titulo_Detalhe
```

```
WHERE Titulo_Detalhe.cod_titulo IN(
```

```
    SELECT cod_titulo FROM Titulo
```

```
    WHERE Titulo.ind_status = 'A'
```

```
    ORDER BY Titulo.nom_titulo ASC
```

```
);
```

- **Ano de Lançamento: coluna ano\_lancamento da tabela Titulo\_Detalhe**

```
SELECT ano_lancamento FROM Titulo_Detalhe  
WHERE Titulo_Detalhe.cod_titulo IN(  
  
    SELECT cod_titulo FROM Titulo  
  
    WHERE Titulo.ind_status = 'A'  
  
    ORDER BY Titulo.nom_titulo ASC  
  
);
```

- **Duração: coluna qtd\_minutos da tabela Titulo\_Detalhe (se existir ou não)**

```
SELECT qtd_minutos FROM Titulo_Detalhe  
WHERE Titulo_Detalhe.cod_titulo IN(  
  
    SELECT cod_titulo FROM Titulo  
  
    WHERE Titulo.ind_status = 'A'  
  
    ORDER BY Titulo.nom_titulo ASC  
  
);
```

- **Gênero(s): coluna dsc\_genero da tabela Titulo\_Detalhe (se existir ou não)**

```
SELECT dsc_genero FROM Titulo_Detalhe  
WHERE Titulo_Detalhe.cod_titulo IN(  
  
    SELECT cod_titulo FROM Titulo  
  
    WHERE Titulo.ind_status = 'A'  
  
    ORDER BY Titulo.nom_titulo ASC  
  
);
```

- **Nota: coluna classificacao\_media da tabela Avaliacao (se existir ou não)**

```
SELECT classificacao_media FROM Avaliacao
WHERE Avaliacao.cod_titulo IN(

    SELECT cod_titulo FROM Titulo

    WHERE Titulo.ind_status = 'A'

    ORDER BY Titulo.nom_titulo ASC

);
```

- **Autor: nom\_pessoa da tabela Pessoa (se existir ou não)**

```
SELECT nom_pessoa FROM Pessoa
WHERE Pessoa.cod_pessoa IN(

    SELECT cod_pessoa FROM Autoria

    INNER JOIN Titulo

    ON Autoria.cod_pessoa = Titulo.cod_pessoa

    WHERE Titulo.ind_status = 'A'

    ORDER BY Titulo.nom_titulo ASC

);
```



- **Diretor: nom\_pessoa da tabela Pessoa (se existir ou não)**

```
SELECT nom_pessoa FROM Pessoa
WHERE Pessoa.cod_pessoa IN(
    SELECT cod_pessoa FROM Direcao
    INNER JOIN Titulo
    ON Direcao.cod_pessoa = Titulo.cod_pessoa
    WHERE Titulo.ind_status = 'A'
    ORDER BY Titulo.nom_titulo ASC
);
```