

Bootcamp IGTi: Analista de Bando de Dados**Desafio**

Módulo 4	Performance e Otimização
-----------------	---------------------------------

Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Análise e otimização de plano de queries no SQL Server e MongoDB.
- ✓ Cargas e expurgos massivos de dados no SQL Server e MongoDB.

Enunciado

O projeto para o qual você foi contratado no Módulo 2, entra agora em sua segunda fase, onde será feita a otimização das queries que foram implementadas no SQL Server.

Atividades

1. Criar o banco **BDProducoes** e seu schema físico usando o script abaixo;

```
CREATE DATABASE BDProducoes
GO
```

```
USE [BDProducoes]
GO
```

```
CREATE TABLE [dbo].[Autoria](
    [cod_Autoria] [int] IDENTITY(1,1) NOT NULL,
    [cod_titulo] [int] NOT NULL,
    [cod_pessoa] [int] NOT NULL,
    CONSTRAINT [PK_Autoria] PRIMARY KEY CLUSTERED
(
    [cod_Autoria] ASC
)
)
GO
```

```
CREATE TABLE [dbo].[Avaliacao](
    [cod_titulo] [int] NOT NULL,
    [classificacao_media] [int] NOT NULL,
    [qtd_votos] [int] NOT NULL
)
GO
```

```
CREATE TABLE [dbo].[Direcao](
    [cod_direcao] [int] IDENTITY(1,1) NOT NULL,
    [cod_titulo] [int] NOT NULL,
    [cod_pessoa] [int] NOT NULL,
    CONSTRAINT [PK_Direcao] PRIMARY KEY CLUSTERED
(
    [cod_direcao] ASC
)
)
GO
```

```
CREATE TABLE [dbo].[Elenco](
    [cod_elenco] [int] IDENTITY(1,1) NOT NULL,
    [cod_titulo] [int] NOT NULL,
    [cod_pessoa] [int] NOT NULL,
    [dsc_funcao] [varchar](1000) NOT NULL,
    [desc_personagem] [varchar](1000) NULL,
    CONSTRAINT [PK_Elenco_1] PRIMARY KEY CLUSTERED
(
    [cod_elenco] ASC
)
)
GO
```

```
CREATE TABLE [dbo].[Pessoa](
    [cod_pessoa] [int] NOT NULL,
    [nom_pessoa] [varchar](500) NOT NULL,
    [ano_nascimento] [int] NULL,
    [ano_falecimento] [int] NULL,
    [dsc_profissao] [varchar](1000) NULL,
    CONSTRAINT [PK_Pessoa] PRIMARY KEY CLUSTERED
(
    [cod_pessoa] ASC
)
)
GO
```

```
CREATE TABLE [dbo].[Titulo](
    [cod_titulo] [int] NOT NULL,
    [nom_titulo] [varchar](1000) NULL,
    CONSTRAINT [PK_Titulo] PRIMARY KEY CLUSTERED
(
    [cod_titulo] ASC
)
)
GO
```

```
CREATE TABLE [dbo].[Titulo_Detalhe](
    [cod_titulo] [int] NOT NULL,
    [tip_titulo] [varchar](100) NOT NULL,
    [nom_principal_titulo] [varchar](1000) NOT NULL,
    [nom_original_titulo] [varchar](1000) NOT NULL,
    [ind_adulto] [bit] NOT NULL,
    [ano_lancamento] [int] NOT NULL,
    [qtd_minutos] [smallint] NULL,
    [dsc_genero] [varchar](1000) NULL
)
GO
```

```
ALTER TABLE [dbo].[Autoria] WITH CHECK ADD CONSTRAINT [FK_Autoria_Pessoa] FOREIGN
KEY([cod_pessoa])
REFERENCES [dbo].[Pessoa] ([cod_pessoa])
GO
```

```
ALTER TABLE [dbo].[Autoria] CHECK CONSTRAINT [FK_Autoria_Pessoa]
GO
```

```
ALTER TABLE [dbo].[Autoria] WITH CHECK ADD CONSTRAINT [FK_Autoria_Titulo] FOREIGN
KEY([cod_titulo])
REFERENCES [dbo].[Titulo] ([cod_titulo])
GO
```

```
ALTER TABLE [dbo].[Autoria] CHECK CONSTRAINT [FK_Autoria_Titulo]
GO
```

```
ALTER TABLE [dbo].[Avaliacao] WITH CHECK ADD CONSTRAINT [FK_Avaliacao_Titulo]
FOREIGN KEY([cod_titulo])
REFERENCES [dbo].[Titulo] ([cod_titulo])
GO
```

```
ALTER TABLE [dbo].[Avaliacao] CHECK CONSTRAINT [FK_Avaliacao_Titulo]
GO
```

```
ALTER TABLE [dbo].[Direcao] WITH CHECK ADD CONSTRAINT [FK_Direcao_Pessoa] FOREIGN
KEY([cod_pessoa])
```



```
REFERENCES [dbo].[Pessoa] ([cod_pessoa])
GO

ALTER TABLE [dbo].[Direcao] CHECK CONSTRAINT [FK_Direcao_Pessoa]
GO

ALTER TABLE [dbo].[Direcao] WITH CHECK ADD CONSTRAINT [FK_Direcao_Titulo] FOREIGN
KEY([cod_titulo])
REFERENCES [dbo].[Titulo] ([cod_titulo])
GO

ALTER TABLE [dbo].[Direcao] CHECK CONSTRAINT [FK_Direcao_Titulo]
GO

ALTER TABLE [dbo].[Elenco] WITH CHECK ADD CONSTRAINT [FK_Elenco_Pessoa] FOREIGN
KEY([cod_pessoa])
REFERENCES [dbo].[Pessoa] ([cod_pessoa])
GO

ALTER TABLE [dbo].[Elenco] CHECK CONSTRAINT [FK_Elenco_Pessoa]
GO

ALTER TABLE [dbo].[Elenco] WITH CHECK ADD CONSTRAINT [FK_Elenco_Titulo] FOREIGN
KEY([cod_titulo])
REFERENCES [dbo].[Titulo] ([cod_titulo])
GO

ALTER TABLE [dbo].[Elenco] CHECK CONSTRAINT [FK_Elenco_Titulo]
GO

ALTER TABLE [dbo].[Titulo_Detalhe] WITH CHECK ADD CONSTRAINT
[FK_Titulo_Detalhe_Titulo] FOREIGN KEY([cod_titulo])
REFERENCES [dbo].[Titulo] ([cod_titulo])
GO

ALTER TABLE [dbo].[Titulo_Detalhe] CHECK CONSTRAINT [FK_Titulo_Detalhe_Titulo]
GO

ALTER TABLE dbo.Titulo
ALTER COLUMN nom_titulo VARCHAR(1000) NOT NULL
GO

ALTER TABLE Pessoa ALTER COLUMN nom_pessoa varchar(1000) NOT NULL
GO

ALTER TABLE Titulo ADD ind_status char(1) NOT NULL CONSTRAINT DF_Titulo_ind_status2
DEFAULT 'A'
GO
```

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

a. Baixar os arquivos disponíveis em:

<https://1drv.ms/u/s!AnQw47iLLy1dxc1CHhuVvilRvkJheA?e=1eocGv;>

- b. Conectado no banco de dados criado no item 1, substituir, nos comandos abaixo, o caminho no qual os arquivos foram salvos e executar o script para carregar os dados, usando o comando BULK INSERT.

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

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

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

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

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

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

```
BULK INSERT BDProducoes.dbo.Direcao
FROM 'F:\Dataset\Direcao.txt'
WITH ( FIELDTERMINATOR = '\t',
       ROWTERMINATOR = '\n',

       --Manter os valores nos arquivos para colunas com a prop. IDENTITY
       KEEPIDENTITY
     );
```

3. Analise o plano de execução da query abaixo, que retorna a quantidade de títulos ativos, e verifique como ela pode ser melhorada;

```
SELECT count(*) AS "Total de Títulos Ativos"
FROM Titulo
WHERE ind_status = 'A'
GO
```

4. Analise o plano de execução da query abaixo, que retorna a relação dos 100 títulos mais bem avaliados, suas avaliações e total de votos, e verifique como ela pode ser melhorada:

```
SELECT TOP (100) T.nom_titulo, A.classificacao_media, A.qtd_votos
FROM Titulo T JOIN Avaliacao A
ON T.cod_titulo = A.cod_titulo
ORDER BY A.classificacao_media DESC
```

5. Analise o plano de execução da query abaixo, Títulos sem o detalhe da duração (qtd_minutos) ou informação do gênero (dsc_genero), e verifique como ela pode ser melhorada:

```
SELECT *
FROM Titulo T JOIN Titulo_Detalhe D
ON T.cod_titulo = D.cod_titulo
WHERE D.qtd_minutos IS NULL OR D.dsc_genero IS NULL
```


6. Analise o plano de execução da query abaixo, que retorna informações acerca dos títulos ativos, ordenados alfabeticamente pelo nome do título, e verifique como ela pode ser melhorada.

```
SELECT T.nom_titulo AS "Nome do Título", UPPER (TD.tip_titulo) AS "Tipo do  
Título", TD.ano_lancamento AS "Ano de Lançamento", TD.qtd_minutos AS "Duração",  
TD.dsc_genero AS "Gênero(s)", A.classificacao_media AS "Nota", P.nom_pessoa AS  
"Autor", P2.nom_pessoa AS "Diretor"  
FROM Titulo T LEFT JOIN Titulo_Detalhe TD  
ON T.cod_titulo = TD.cod_titulo  
LEFT JOIN Avaliacao A  
ON T.cod_titulo = A.cod_titulo  
LEFT JOIN Autoria E  
ON T.cod_titulo = E.cod_titulo  
LEFT JOIN Pessoa P  
ON E.cod_pessoa = P.cod_pessoa  
LEFT JOIN Direcao D  
ON T.cod_titulo = D.cod_titulo  
LEFT JOIN Pessoa P2  
ON D.cod_pessoa = P2.cod_pessoa  
WHERE T.ind_status = 'A'  
ORDER BY T.nom_titulo
```

7. A tabela **dbo.Pessoa** é a maior tabela do sistema. Constantemente é necessário fazer uma limpeza das pessoas cadastradas que não exercem nenhuma função, ou seja, não é do elenco, não é diretor e não é autor. Sem incluir, alterar ou excluir as colunas das tabelas existentes, pense em uma estratégia para esse expurgo ser feito da forma mais performática possível.
8. O diretor da empresa que te contratou, ficou sabendo que você aprendeu, no Bootcamp de Analista de Banco de Dados do IGTI, como melhorar performance de queries no MongoDB. Devido a isso, te contratou também para trabalhar em uma força tarefa para melhorar a performance de um sistema de outra empresa do grupo, que controla informações dos restaurantes de todo o Estados Unidos.

Para isso, foi disponibilizado para você, no drive <https://1drv.ms/u/s!AnQw47iLLy1dxoppaSVsA1AUibBQpA?e=T8QVgl>, o dump das duas collections do sistema e o script para fazer a carga das informações no MongoDB.

Com os dados carregados, você deve ajudar a melhorar a performance das seguintes queries:

```
db.people.find( { last_name : "Acevedo" } )
```

```
db.people.find({"address.zip" : "10914-3394"})
```

```
db.people.find({"address.zip" : { $in: ["10914-3394", "53666"]}})
```

```
db.people.find({}, { _id : 0, first_name: 1, job: 1 }).sort({ first_name: 1 })
```

```
db.restaurants.find({'cuisine': 'Sichuan'})
```

```
db.restaurants.find ({ cuisine: 'Sushi' }).sort({ stars: -1 })
```

```
db.restaurants.find({ "stars": { $gte: 4 }, cuisine: 'Italian' }).sort({ name: 1 })
```


Respostas Finais

Os alunos deverão desenvolver a prática e, depois, responder às seguintes questões objetivas: