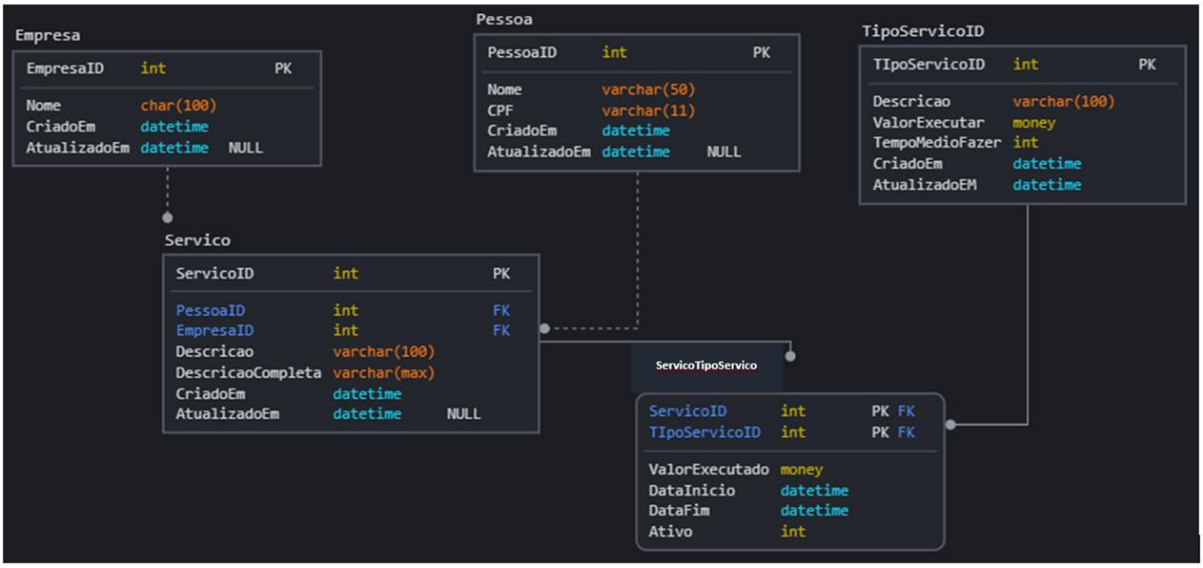
Regras para Realização da Avaliação/Projeto

- Qualquer indício de cópia será descontado da nota final.

- Envie a resolução em um arquivo **PDF**, organize o conteúdo para auxiliar na compreensão/correção



1 - Com base no diagrama, responda e/ou desenvolva:

A – Explique onde este modelo ‘de banco de dados seria aplicado: tipo de negócio, etc. 20

B – Crie um Banco de Dados no SQL SERVER. O banco não pode ser criado na pasta sugerida pelo SQL Server. Ou seja, crie o banco de dados em uma pasta específica.

C - Crie um Esquema no banco de dados (seu nome + RA) e crie as tabelas do diagrama neste esquema.

D - Você alteraria algum tipo de dado das tabelas acima? Faça a alteração e justifique sua resposta!

E – Faça a inserção de pelo menos três registros em cada tabela.

F – Pesquise todas as Pessoas e o tipos de serviços que elas contrataram. Não apresente serviços repetidos.

G - O que é necessário definir no início de um projeto desse banco de dados para que auxilie na performance?

H - Teremos dois tipos de usuários que devem ser adicionados no banco: o primeiro terá acesso apenas para leitura de dados, pois será utilizado em aplicativo de consulta, o segundo terá acesso para escrita e leitura no banco, mas não poderá apagar(drop) objetos do banco. Qual seria sua estratégia para o banco de dados. Crie esses usuários.

2 – Responda como funciona a paginação (de armazenamento) de dados no SQL Server?

3 – Descreva o que é collation e qual a sua importância ao configurar um banco de dados SQL Server. Altere a collation do Banco de Dados que você criou e mostre na tela.

4 – Você está configurando um banco de dados que não será case sensitive. Qual seria o collation ideal?

5 – O que são indices, quais os principais tipos de index do SQL Server?

6 – Com base no banco de dados e tabelas que você criou no exercício 1, responda:

A – Existe uma busca constante pelo CPF da Pessoa, qual estratégia você utilizaria para melhorar a performance da pesquisa.

B – A pesquisa “select * from ServicoTipoServico Where year(DataInicio)=2020” é executada constantemente. Para melhorar a performance, seria útil a utilização de índice? Justifique sua resposta.

C – Crie pelo menos 3 índices (non-clustered) para as tabelas do banco criado no exercício 1.
Explique qual o motivo da criação de cada um deles.

7 – Descreva quais os principais tipos de backup e como eles funcionam. Além disso, monte uma estratégia de backup para o banco criado no exercício 1, tanto relacionado com tipo e local de armazenamento. Apresente o código e/ou imagem e a estratégia da criação dos backups.

8 – Pesquise como funciona a otimização de consulta em banco de dados orientado a documento, por exemplo, MongoDB. Explique com suas palavras o que compreendeu.

9 – Crie um login e um usuário no SQL Server que terá acesso apenas de leitura no Banco de Dados que você criou

10 – O que são roles no Banco de Dados e na instância. Crie exemplos e cite exemplo de aplicação.

11 – Crie um usuário que terá acesso de leitura na tabela TipoServico, contudo não terá acesso para ver a coluna ValorExecutar.

12 – Crie 5 exemplos de GRANT e REVOKE no banco de dados e explique cada um deles.

13 – Apresente a similaridade desse projeto com o conteúdo abordado no Semestre.

“A sorte chega para quem está em movimento.”

Respostas

1- a) Na imagem aparentemente seria a representação de uma pessoa que esteja se associando a um serviço em uma empresa x que para realizar um tipo de serviço. Nisto eu imagino que a estrutura possa servir para uma empresa que forneça trabalhos terceirizados e queira gerenciar onde e com o que seu funcionário está trabalhando, uma empresa que apresenta ofertas de vagas tipo o *linkedin* para associar pessoas as vagas e tipo de serviços.

b)

```
CREATE DATABASE MeuBancoDeDados
ON PRIMARY (
    NAME = MeuBancoDeDados_Data,
    FILENAME = 'D:\MeusBancosDeDados\MeuBancoDeDados_Data.mdf',
    SIZE = 10MB,
    MAXSIZE = 100MB,
    FILEGROWTH = 10MB
)
LOG ON (
    NAME = MeuBancoDeDados_Log,
    FILENAME = 'D:\MeusBancosDeDados\MeuBancoDeDados_Log.ldf',
    SIZE = 10MB,
    MAXSIZE = 50MB,
    FILEGROWTH = 10MB
);
```

Result:

```
9:35:56 AM     Started executing query at Line 1
               Commands completed successfully.
9:35:56 AM     Started executing query at Line 3
               Msg 262, Level 14, State 1, Line 4
               CREATE DATABASE permission denied in database 'master'.
               Total execution time: 00:00:00.028
```

c)

```
CREATE SCHEMA Thieres_044987;
```

```
    PessoaID INT PRIMARY KEY,
    Nome VARCHAR(50) NOT NULL,
    CPF VARCHAR(11) NOT NULL,
    CriadoEm DATE NOT NULL,
    AtualizadoEm DATE NULL
);

CREATE TABLE Thieres_044987.Empresa (
    EmpresaID INT PRIMARY KEY,
    Nome CHAR(100) NOT NULL,
    CriadoEm DATE NOT NULL,
    AtualizadoEm DATE NULL
);

CREATE TABLE Thieres_044987.Servico (
    ServicoID INT PRIMARY KEY,
    EmpresaID INT,
    PessoaID INT,
    Descricao VARCHAR(100) NOT NULL,
    DescricaoCompleta VARCHAR(MAX) NOT NULL,
    CriadoEm DATE NOT NULL,
    AtualizadoEm DATE NULL,
    FOREIGN KEY (PessoaID) REFERENCES Thieres_044987.Pessoa(PessoaID),
    FOREIGN KEY (EmpresaID) REFERENCES Thieres_044987.Empresa(EmpresaID),
);

CREATE TABLE Thieres_044987.TipoServico (
    TipoServicoID INT PRIMARY KEY,
    Descricao VARCHAR(100) NOT NULL,
    ValorExecutar MONEY NOT NULL,
    TempoMedioFazer INT NOT NULL,
    CriadoEm DATE NOT NULL,
    AtualizadoEm DATE NULL
);

CREATE TABLE Thieres_044987.ServicoTipoServico (
    ServicoID INT,
    TipoServicoID INT,
    ValorExecutar MONEY NOT NULL,
    DataInicio DATE NOT NULL,
    DataFim DATE NULL,
    Ativo INT NOT NULL,
    PRIMARY KEY (ServicoID, TipoServicoID),
    FOREIGN KEY (ServicoID) REFERENCES Thieres_044987.Servico(ServicoID),
    FOREIGN KEY (TipoServicoID) REFERENCES Thieres_044987.TipoServico(TipoServicoID),
);
```

d) A primeiro momento não mudaria pensando em aproveitar como se fosse a estrutura da empresa, porém conforme a necessidade faria futuras recomendações.

e)

```
INSERT INTO Thieres_044987.Pessoa
(
    PessoaID,
    Nome,
    CPF,
    CriadoEm
) VALUES
(1,'User1','12345678912',GETDATE()),
(2,'User2','12345678913',GETDATE()),
(3,'User3','12345678912',GETDATE())
```

```
INSERT INTO Thieres_044987.Empresa
(
    EmpresaID,
    Nome,
    CriadoEm
) VALUES
(1,'Company1',GETDATE()),
(2,'Company2',GETDATE()),
(3,'Company3',GETDATE())
```

```
INSERT INTO Thieres_044987.Servico
(
    ServicoID,
    EmpresaID,
    PessoaID,
    Descricao,
    DescricaoCompleta,
    CriadoEm
) VALUES
(1,1,1,'service1','desc of service',GETDATE()),
(2,2,2,'service2','desc of service',GETDATE()),
(3,3,3,'service3','desc of service',GETDATE())
```

```
INSERT INTO Thieres_044987.TipoServico
(
    TipoServicoID,
    Descricao,
    ValorExecutar,
    TempoMedioFazer,
    CriadoEm
) VALUES
(1,'typeService1',200.00,8,GETDATE()),
(2,'typeService2',200.00,8,GETDATE()),
(3,'typeService3',200.00,8,GETDATE())
```

```
INSERT INTO Thieres_044987.ServicoTipoServico
(
    ServicoID,
    TipoServicoID,
    ValorExecutar,
    Ativo,
    DataInicio
) VALUES
(1,1,200.00,1,GETDATE()),
(2,2,200.00,1,GETDATE()),
(3,3,200.00,1,GETDATE())
```

f)

```sql
SELECT
    p.PessoaID,
    p.Nome AS NomePessoa,
    s.ServicoID,
    s.Descricao AS DescricaoServico,
    e.Nome AS NomeEmpresa,
    ts.Descricao AS TipoServicoDescricao,
    sts.ValorExecutar,
    sts.DataInicio,
    sts.DataFim,
    sts.Ativo
FROM
    Thieres_044987.Pessoa p
JOIN
    Thieres_044987.Servico s ON p.PessoaID = s.PessoaID
JOIN
    Thieres_044987.Empresa e ON s.EmpresaID = e.EmpresaID
JOIN
    Thieres_044987.ServicoTipoServico sts ON s.ServicoID = sts.ServicoID
JOIN
    Thieres_044987.TipoServico ts ON sts.TipoServicoID = ts.TipoServicoID
GROUP BY
    p.PessoaID,
    p.Nome,
    s.ServicoID,
    s.Descricao,
    e.Nome,
    ts.Descricao,
    sts.ValorExecutar,
    sts.DataInicio,
    sts.DataFim,
    sts.Ativo
ORDER BY
    p.PessoaID;
```

g) Definir esses aspectos no início do projeto de um banco de dados SQL Server ajuda a garantir uma performance adequada e escalabilidade futura. Investir tempo na modelagem de dados, configuração de collation, criação de índices, particionamento, otimização de armazenamento e configuração do servidor são passos essenciais para construir um sistema eficiente e confiável.

h)

```
-- Criar login para usuário de leitura
CREATE LOGIN ReadOnlyUser WITH PASSWORD = 'StrongPassword123!';
-- Criar usuário de banco de dados para leitura
CREATE USER ReadOnlyUser FOR LOGIN ReadOnlyUser;
-- Conceder permissões de leitura ao ReadOnlyUser
ALTER ROLE db_datareader ADD MEMBER ReadOnlyUser;

-- Criar login para usuário de leitura e escrita
CREATE LOGIN ReadWriteUser WITH PASSWORD = 'AnotherStrongPassword123!';
-- Criar usuário de banco de dados para leitura e escrita
CREATE USER ReadWriteUser FOR LOGIN ReadWriteUser;
-- Conceder permissões de leitura e escrita ao ReadWriteUser
ALTER ROLE db_datareader ADD MEMBER ReadWriteUser;
ALTER ROLE db_datawriter ADD MEMBER ReadWriteUser;
-- Remover a permissão de deletar objetos do ReadWriteUser
DENY ALTER, CONTROL, DELETE, DROP, REFERENCES ON SCHEMA::dbo TO ReadWriteUser;
```

2) No SQL Server, a paginação de armazenamento de dados refere-se ao método pelo qual o SQL Server organiza e gerencia fisicamente os dados em disco. Isso é crucial para o desempenho e eficiência das operações de leitura e gravação no banco de dados.

3) Collation é essencial para garantir que os dados de texto sejam armazenados, comparados e ordenados corretamente de acordo com as regras de idioma e localidade específicas. Uma configuração adequada de collation impacta a compatibilidade, desempenho, consistência e interoperabilidade dos dados no SQL Server.

4) Se você está configurando um banco de dados que não deve ser sensível a maiúsculas e minúsculas (case insensitive), o collation ideal deve incluir CI (Case Insensitive) na sua definição. Além disso, você também pode considerar outras sensibilidades, como acentuação (Accent Sensitivity) dependendo dos requisitos do seu banco de dados.

Um collation comum para este tipo de configuração é Latin1_General_CI_AS, que é case insensitive e accent sensitive. Isso significa que ele trata letras maiúsculas e minúsculas como iguais, mas distingue caracteres acentuados.

5) Os índices são ferramentas essenciais para melhorar a performance de consultas e operações de leitura em um banco de dados SQL Server. Os principais tipos incluem índices clusterizados, não clusterizados, únicos, filtrados, columnstore e de texto completo. A escolha e o gerenciamento adequados dos índices são cruciais para garantir um desempenho eficiente e a integridade dos dados no banco de dados.

6 a)Para melhorar a performance da pesquisa constante pelo CPF na tabela Pessoa, pode-se utilizar um índice no campo CPF.

```sql
CREATE NONCLUSTERED INDEX IX_Pessoa_CPF
ON Pessoa (CPF);
```

b) Sim, a utilização de um índice pode melhorar significativamente a performance da consulta.

```sql
CREATE NONCLUSTERED INDEX IX_ServicoTipoServico_DataInicio
ON ServicoTipoServico (DataInicio);
```

c)

```sql
-- CPF de pessoa (Busca constante)
CREATE NONCLUSTERED INDEX IX_Pessoa_CPF
ON Thieres_044987.Pessoa (CPF);

-- EmpresaID de Serviço (Filtrar por empresa já que é apenas uma foreigh key)
CREATE NONCLUSTERED INDEX IX_Servico_EmpresaID
ON Thieres_044987.Servico (EmpresaID);

-- Chave composta em ServicoTipoServico(facilitar a busca)
CREATE NONCLUSTERED INDEX IX_ServicoTipoServico_ServicoID_TipoServicoID
ON Thieres_044987.ServicoTipoServico (ServicoID, TipoServicoID);
```

7) Existem diferentes tipos de backups que podem ser implementados em um sistema de banco de dados para garantir a segurança dos dados e a capacidade de recuperação em caso de falhas.

- **Backup Completo (Full Backup):**
  - Cópia de todos os dados do banco de dados, incluindo objetos e estruturas.
  - Usado como ponto de partida para restaurações completas.
- **Backup Diferencial (Differential Backup):**
  - Captura alterações desde o último backup completo.
  - Mais rápido e ocupa menos espaço que o backup completo.
- **Backup de Log de Transações (Transaction Log Backup):**
  - Registra todas as transações desde o último backup de log.
  - Essencial para a recuperação até um ponto específico no tempo.
- **Backup de Arquivo ou Grupo de Arquivos (File/Filegroup Backup):**
  - Backup de partes específicas do banco de dados.
  - Útil para gerenciar grandes bancos de dados de forma independente.
- **Backup de Cópia (Copy-Only Backup):**

- Backup independente dos backups regulares.
- Usado para fins especiais sem afetar a sequência de backup regular.

Imagino que o backup completo, com uma periodicidade semanal e armazenamento remoto ou local seria uma ótima forma de manter estes dados.

8) A otimização de consultas em bancos de dados orientados a documentos, como MongoDB, envolve estratégias como modelagem eficiente de dados, uso de índices (simples, compostos, de texto, geoespaciais), análise de consultas para escolha de planos de execução eficientes, e técnicas avançadas como o Aggregation Framework e sharding para escalabilidade. Monitoramento contínuo e ajustes são essenciais para garantir alto desempenho das consultas.

9)

```
USE ThieresDB;
GO

CREATE USER MeuUsuarioLogin FOR LOGIN MeuUsuarioLogin;
GO

GRANT SELECT ON SCHEMA :: dbo TO MeuUsuarioLogin;
GO
```

10) As roles no banco de dados são usadas para agrupar permissões que são aplicadas a usuários ou outros roles dentro do contexto de um banco de dados específico.

```
USE ThieresDB;
GO
-- Cria a role
CREATE ROLE LeituraSomente;
-- permissão de leitura (SELECT) no schema padrão (dbo)
GRANT SELECT ON SCHEMA :: dbo TO LeituraSomente;
-- usuarios associado a esta role terá as permissões de leitura definidas.
ALTER ROLE LeituraSomente ADD MEMBER MeuUsuarioLogin;
```

```
USE Master;
GO
-- Cria a role
CREATE SERVER ROLE AdminDBA;
-- logins associados à role AdminDBA possam alterar quaisquer logins na instância do SQL Server.
GRANT ALTER ANY LOGIN TO AdminDBA;
-- MeuUsuarioLogin agora terá a capacidade de alterar quaisquer logins na instância do SQL Server.
ALTER SERVER ROLE AdminDBA ADD MEMBER MeuUsuarioLogin;
```

11)

```sql
USE ThieresDB;
GO
CREATE USER UsuarioTeste FOR LOGIN UsuarioTeste;
GO
GRANT SELECT ON dbo.TipoServico TO UsuarioTeste;
GO
DENY SELECT (ValorExecutar) ON dbo.TipoServico TO UsuarioTeste;
GO
```

```sql
USE Thieres_044987;
GO
CREATE USER UsuarioTeste FOR LOGIN UsuarioTeste;
GO
GRANT SELECT ON dbo.TipoServico TO UsuarioTeste;
GO
DENY SELECT (ValorExecutar) ON dbo.TipoServico TO UsuarioTeste;
GO
```

12)

```sql
-- GRANT: Conceder permissão SELECT em uma tabela específica para um usuário
GRANT SELECT ON dbo.MinhaTabela TO MeuUsuario;

-- GRANT: Conceder permissão EXECUTE em um procedimento armazenado para uma função
GRANT EXECUTE ON dbo.Meuprocedimento TO MinhaFuncao;

-- GRANT: Conceder permissão CONTROL SERVER para um login de servidor
GRANT CONTROL SERVER TO MeuLogin;

-- REVOKE: Revogar permissão SELECT de uma tabela específica de um usuário
REVOKE SELECT ON dbo.MinhaTabela FROM MeuUsuario;

-- REVOKE: Revogar permissão EXECUTE de um procedimento armazenado de uma função
REVOKE EXECUTE ON dbo.Meuprocedimento FROM MinhaFuncao;

-- REVOKE: Revogar permissão CONTROL SERVER de um login de servidor
REVOKE CONTROL SERVER TO MeuLogin;
```

13) As atividades em relação ao conteudo abordado neste bimestre conseguiram de certa forma conectar todas os ensinamentos e reunir todo o conteudo que eu admito não ter entendido em sala mais compleendido melhor executando(O que podia) com estes exercicios. Então obrigado e Boa ferias Robinho!