**Planejamento de Normas de Padronização de Banco de dados**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 18/03/2023 | 1.0 | Elaboração do Documento | Germano Manente Neto |
| 10/04/2023 | 2.0 | Revisão e Implementação de Novas Praticas de Gerenciamento do SGDB | Germano Manente Neto |

**Integrantes da Equipe**

|  |  |
| --- | --- |
| **Nome** | **Cargo Ocupado** |
| Flávio Gabriel Ruas Gimenes | Diretor da Organização |
| Clayton W Brugnoli | Administrador de Banco de Dados |
| Dhiogo Nascimento | Administrador de Banco de Dados |
| Arlen Augusto | Coordenador de Desenvolvimento |

**Sumário**

**1** **Visão Geral do Artefato 5**

1.1 Propósito do Documento 5

1.2 Público Alvo 5

1.3 Finalidade 5

1.4 Organograma Funcional 5

1.5 Contatos dos Integrantes 6

1.6 Papéis e Responsabilidades 6

**2** **Regra Geral de Desenvolvimento 7**

2.1 Padrão de nomenclatura 7

2.2 Minimizar o uso de procedures, functions e views 7

2.3 Minimizar o uso Jobs 7

2.4 Padrão de codificação do script 7

2.5 Campos da tabela 8

2.6 Diagrama 8

2.7 Nomes de Views 9

2.8 Nomes de procedures 9

2.9 Nomes de Funções 9

2.10 Documentação de Código 9

2.11 Endentação do código 10

**3** **Exemplo 10**

3.1 Assinatura da Schema 10

3.2 Assinatura da Tabela 11

3.3 Assinatura da Chave Primária 11

3.4 Assinatura da Chave Estrangeira 11

3.5 Assinatura da Índice 12

3.6 Assinatura do Procedimento 12

3.7 Assinatura da Função 13

3.8 Assinatura da Trigger 14

3.9 Sugestão de padrões para criar restrições no modelo físico. 15

**4** **Plano de Gerenciamento de 16**

4.1 Check List 16

4.2 Espaço em Disco 17

* **Visão Geral do Artefato**
* **Propósito do Documento**

O propósito deste documento é informar ao analista o padrão de desenvolvimento de todos os objetos do banco de dados, bem como Gerenciar de forma eficiente o Sistema Gerenciador de Banco de Dados SQL Server.

* **Público Alvo**

Este documento se destina ao Setor de TI da Organização, garantindo assim uma melhor qualidade e entendimento de todos os analistas responsáveis do setor.

* **Finalidade**

O objetivo deste tópico é informar como o sistema gerenciador de banco de dados está se comportando, bem como a sua disponibilidade e informar a todos os envolvidos de sua situação.

Este tópico aborda as seguintes informações:

* Organograma Funcional: descrever uma visão geral dos principais integrantes que fazem parte do setor da Organização.
* Contatos dos Integrantes: descreve as informações dos contatos contidos no organograma funcional.
* Papeis e Responsabilidades: descrever o papel a ser realizado e as responsabilidades dos envolvidos para o funcionamento do SGDB e bancos de dados da instituição.
* **Organograma Funcional**
* **Contatos dos Integrantes**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Cargo / Função | Telefone (s) | E-mail |
| Flavio Gabriel | Diretor de Organização | 65 9952-7412 | gflavio581@gmail.com |
| Clayton W Brugnoli | Administrador de Banco de Dados | 65 9218-4810 | brugnol1@gmail.com |
| Dhiogo Nascimento | Administrador de Banco de Dados | 65 8160-0011 | dhgnsc@gmail.com |
| Arlen Augusto | Coordenador de Desenvolvimento | 62 9967-0021 | arlen242@gmail.com |

* **Papéis e Responsabilidades**

*Descrever detalhadamente o papel e responsabilidade de cada uma das funções abaixo relacionadas.*

**Administrador de Banco de Dados:**

**Papel**: O Administrador de Banco de Dados (DBA) é responsável por gerenciar, manter e garantir a disponibilidade, segurança e desempenho dos bancos de dados de uma organização. Ele é essencial para assegurar que os dados corporativos sejam armazenados, acessados e utilizados de maneira eficiente.

**Responsabilidades**:

**Gerenciamento de Banco de Dados**: Criar, configurar, monitorar e manter bancos de dados.

**Segurança**: Implementar políticas de segurança para proteger os dados contra acessos não autorizados e falhas.

**Backup e Recuperação**: Garantir que backups sejam realizados regularmente e que estratégias de recuperação estejam disponíveis para minimizar perda de dados.

**Otimização de Desempenho**: Monitorar o desempenho dos bancos de dados e ajustar configurações para evitar gargalos.

**Atualizações**: Planejar e aplicar atualizações e patches de segurança.

Suporte a Desenvolvedores: Colaborar com equipes de desenvolvimento para criar e ajustar estruturas de banco de dados que atendam às necessidades dos aplicativos.

**Auditoria**: Gerenciar logs de acesso e auditorias para garantir conformidade com regulamentos.

**Documentação**: Documentar mudanças, configurações e melhores práticas.

**Diretor de TI:**

**Papel**: O Diretor de TI é um executivo sênior responsável por definir e supervisionar a estratégia tecnológica da organização. Ele alinha a tecnologia com os objetivos de negócios para impulsionar o crescimento e a inovação.

**Responsabilidades:**

**Definição de Estratégia:** Desenvolver e implementar a visão e a estratégia de TI alinhadas aos objetivos da empresa.

**Gestão de Orçamento**: Gerenciar o orçamento de TI, garantindo o uso eficiente dos recursos.

**Governança de TI**: Criar políticas, padrões e regulamentos para assegurar a conformidade com normas e legislações.

**Supervisão de Projetos**: Acompanhar e priorizar iniciativas tecnológicas para atender às demandas do negócio.

**Liderança de Equipe**: Gerenciar líderes e equipes de TI, promovendo um ambiente colaborativo.

**Avaliação de Tecnologia**: Identificar e implementar novas tecnologias que aumentem a eficiência e competitividade.

**Gerenciamento de Riscos**: Avaliar riscos relacionados à TI, incluindo cibersegurança, e implementar medidas de mitigação.

**Relacionamento com Stakeholders**: Comunicar-se com executivos e investidores para mostrar o valor da tecnologia no crescimento organizacional.

**Coordenador de Desenvolvimento:**  
**Papel**: O Coordenador de Desenvolvimento é responsável por gerenciar equipes de desenvolvimento de software, garantindo que os projetos sejam entregues no prazo, dentro do orçamento e com a qualidade esperada.

**Responsabilidades:**

**Gestão de Equipe**: Liderar desenvolvedores, alocar tarefas e monitorar o progresso.

**Planejamento de Projetos**: Criar cronogramas detalhados, definir metas e estimar recursos necessários.

**Garantia de Qualidade**: Assegurar que os produtos finais estejam de acordo com os requisitos e padrões técnicos.

**Comunicação**: Facilitar a comunicação entre a equipe de desenvolvimento e outros departamentos.

**Treinamento e Capacitação**: Identificar necessidades de treinamento e promover o crescimento técnico da equipe.

**Implementação de Metodologias**: Adotar metodologias de desenvolvimento ágil ou tradicional conforme o projeto.

**Resolução de Problemas**: Diagnosticar e solucionar problemas técnicos ou de equipe que possam afetar o projeto.

**Monitoramento de Tendências**: Manter-se atualizado com as melhores práticas e tecnologias emergentes no mercado.

**Coordenador de Infraestrutura:**

**Papel**: O Coordenador de Infraestrutura gerencia as operações relacionadas aos recursos tecnológicos físicos e virtuais da organização, garantindo que estejam disponíveis e funcionais para os usuários.

**Responsabilidades:**

**Gestão de Redes e Servidores**: Supervisionar a instalação, configuração e manutenção de redes, servidores e sistemas.

**Suporte Técnico**: Garantir suporte para usuários internos e externos em questões relacionadas à infraestrutura.

**Segurança:** Implementar soluções de segurança para proteger sistemas contra ameaças cibernéticas.

**Gerenciamento de Data Center**: Coordenar o funcionamento de data centers, incluindo a alocação de espaço, energia e refrigeração.

**Planejamento de Capacidade:** Prever necessidades futuras e implementar upgrades de hardware e software.

**Backup e Recuperação**: Garantir sistemas de backup e estratégias de recuperação para continuidade dos negócios.

**Monitoramento e Documentação**: Acompanhar métricas de desempenho e manter registros detalhados de configuração e mudanças.

**Gerenciamento de Fornecedores**: Trabalhar com fornecedores para adquirir equipamentos e serviços, além de gerenciar contratos de manutenção.

* **Regra Geral de Desenvolvimento**
* **Padrão de nomenclatura**
* O nome dos objetos do banco de dados deve começar sempre com letra maiúscula **(Pascal Case)** e caso for um nome composto o próximo nome deverá começar com a letra maiúscula também, nunca separar por “**underline”** (\_).
* O nome dos objetos deve ficar sempre no singular.
* Sempre inserir o comentário para qualquer objeto.
* Fácil entendimento, intuitiva.
* Todas as palavras em português.
* Evitar a utilização de siglas nos objetos ou alias nas consultas do SGDB.
* Procurar utilizar nomes que identifiquem sua utilidade e aplicação.
* Variáveis, campos e tabelas devem ser utilizados com o mesmo padrão que foram criados.
* Triggers: Deve ser criada apenas uma trigger para a entidade e ser tratada as diversas regras de negócio dentro desta. Triggers sempre devem realizar o tratamento de operações em lote.
* Toda Tabela deve ser obrigatoriamente cadastrado no dicionário de dados.
* **Minimizar o uso de procedures, functions e views**
* Avaliar a necessidade da criação de “procedures” e “*views”.*
* Tentar agrupar procedures, para que não sejam “*select's”* simples
* Procedures não podem ser apenas de “*select's”* simples
* **Minimizar o uso Jobs**
* Avaliar real necessidade da criação de uma job a ser programada no SQL Server.
* Verificar a possibilidade de se criar um software para gerenciar serviços a serem executados em tempo de execução, assim dividiremos o processamento destes processos em outro servidor, uma vez que a regra de negócio deve ficar apenas na aplicação e não no SGDB.
* **Padrão de codificação do script**

Todas as palavras reservadas deverão ser escritas com letra maiúsculas.

Após um SELECT a vírgula (,) deverá ser colocada no início de cada campo.

* **Campos da tabela**

Toda tabela deverá conter uma chave única auto incremento, o nome da “PRIMARY KEY” deverá ser obrigatoriamente Id + o Nome Tabela. Exemplo:

* IdPessoa
* IdPessoaFisica
* IdAlunoDisciplinaCurso
* Não utilizar chave composta nas tabelas.
* Nunca abreviar um nome de campo, como dtAtual, flgStatus. Utilize sempre DataAtual, Status.
* Para campos que poderão conter mais de um valor utilizar o campo SMALLINT
* Exemplo:

TipoDemanda SMALLINT

* 1: Pessoalmente
* 2:Fax
* 3: Email
* 4: Telefone
* **Diagrama**

Toda entidade a ser criada no banco de dados, antes deve ser criada um DER (Diagrama de Entidade de Relacionamento) para as mesmas para que fique de bem claro o relacionamento e de fácil entendimento.

O nome dos diagramas deve conter o prefixo com **“Der”** seguido pelo nome do diagrama. Exemplo:

**DerUsuario**



* **Nomes de Views**

O nome da “*view”* deverá conter o prefixo **“Uvw”** seguido pelo nome da funcionalidade. Exemplo:

* UvwAluno
* UvwAlunoDisciplina
* UvwAlunoDisciplinaCurso
* **Nomes de procedures**

Os nomes das procedures NUNCA deverão começar com **“sp”** pois o SQL Server utiliza este prefixo é usado pelo SQL Server para designar procedimentos armazenados de sistema. O nome do procedimento deverá conter o prefixo **“Usp”** seguido pelo nome da funcionalidade. Exemplo:

**CREATE PROCEDURE UspAlunoDisciplina001**

* **Nomes de Funções**

O nome das “**functions”** deverá conter o mesmo padrão adotado pela criação das procedures. O nome da função deverá conter o prefixo **“Ufc”** seguido pelo nome da funcionalidade Exemplo:

CREATE FUNCTION UfcDataAtual()

* **Documentação de Código**

Cabeçalho, deve ser criado sempre abaixo da linha de definição do objeto, no padrão abaixo:

----------------------------------------------------------------------------------------

-- Autor/Criação.......: Nome do Autor

-- Data de Criação.....: 00/00/000 00:00

-- Funcionalidade/Uso..: Breve resumo da funcionalidade

-----------------------------------------------------------------------------------------

-- Histórico : Data | Hora | Autor | Histórico

-- de : ---------------+-------------+----------------+--------------

-- Alterações 01) : 21/07/2008 | 13:30:00 | Germano |

-----------------------------------------------------------------------------------------

Linhas de processamento, ex.: *“Insert's*” e “*Update's*”, devem ser comentadas de acordo com o seu negócio, para fornecer entendimento sem a necessidade de interpretação de código.

* **Endentação do código**

Variáveis que forem utilizadas como constantes devem ser declaradas no início do objeto. Abreviaturas

As abreviaturas serão constituídas sempre por 3 caracteres. Caso a abreviatura já tenha sido usada, encontrar uma solução que torne a identificação intuitiva, preferencialmente alterando o 3º caractere da abreviatura utilizando a próxima consoante no nome correspondente conforme exemplo abaixo.

|  |  |  |
| --- | --- | --- |
| **Número de Palavras** | **Norma** | **Exemplo** |
| 01 Palavra | Utilizar as 03 primeiras letras da palavra | Nome: Pessoa Abreviatura: Pes |
| 02 Palavras | Utilizar as 02 primeiras letras da primeira palavra e a 1ª letra da segunda palavra | Nome: Pessoa Física  Abreviatura: Pes |
| 03 ou mais Palavra | Utilizar a 1ª letra das duas primeiras palavras e a 1ª letra da última palavra | Nome: Tipo Contrato Prestador Abreviatura: Tcp |

* **Exemplo**
* **Assinatura da Schema**

A utilização do “*schema”* é permitir um maior controle e segurança para o DBA. A ideia é descentralizar o sistema construindo-o por módulos, onde cada módulo terá o seu “*schema”*.

--

-- CRIAÇÃO O SCHEMA

--

CREATE SCHEMA Endereco

Abaixo uma lista de alguns exemplos.

* Endereco;
* Compra;
* Financeiro;
* Contabilidade;
* **Assinatura da Tabela**

--

-- CRIAÇÃO DA TABELA

--

IF EXISTS(SELECT 1 FROM sysobjects WHERE name = 'Pais')

DROP TABLE Endereco.Pais

CREATE TABLE Endereco.Pais

(

IdPais INT IDENTITY(1,1) NOT NULL

,NomePais VARCHAR(80) NOT NULL

,SiglaPais CHAR(3) NOT NULL

)

* **Assinatura da Chave Primária**

O nome da chave primaria será composto pela sigla Pk + Nome da Tabela, Para realizar o suporte, o desenvolvedor conseguirá identificar facilmente o problema somente pela mensagem de erro exibida e resolve-lo em um menor tempo do que o exemplo do modelo com falta de padronização. Abaixo o exemplo:

--

-- CRIAÇÃO DA CHAVE PRIMARIA

--

ALTER TABLE Endereco.Pais

ADD CONSTRAINT PkPais PRIMARY KEY (IdPais)

* **Assinatura da Chave Estrangeira**

O nome da chave estrangeira será composto pela sigla **Fk** + Nome da Tabela de Origem **+ X +** Nome da Tabela de Destino. Abaixo o exemplo:

--

-- CRIAÇÃO DA TABELA

--

CREATE TABLE Endereco.Estado

(

IdEstado INT IDENTITY(1,1) NOT NULL

,IdPais INT NOT NULL

,Nome VARCHAR(80) NOT NULL

,Sigla CHAR(3) NOT NULL

)

--

-- CRIAÇÃO DA CHAVE PRIMARIA

--

ALTER TABLE Endereco.Estado

ADD CONSTRAINT PkEstado PRIMARY KEY (IdEstado)

--

-- CRIAÇÃO DA CHAVE ESTRANGEIRA

--

ALTER TABLE Endereco.Estado

ADD CONSTRAINT FkEstadoXPais FOREIGN KEY (IdPais)

REFERENCES Endereco.Pais (IdPais)

* **Assinatura da Índice**

O nome do índice será composto pela sigla Ndx + Nome do Índice. Exemplo:

CREATE INDEX NdxEstadoSigla

* **Assinatura do Procedimento**

CREATE PROCEDURE Schema.UspNomeProcedimento

(

@pParam1 TIPO

,@pParam2 TIPO

,@pParam3 TIPO

,@pParam3 TIPO

,@pParam4 TIPO

,@pParam5 TIPO OUTPUT

)

----------------------------------------------------------------------------------------

-- Autor/Criação.......: Nome do Autor

-- Data de Criação.....: 00/00/000 00:00

-- Funcionalidade/Uso..: Breve resumo da funcionalidade

-----------------------------------------------------------------------------------------

-- Histórico : Data | Hora | Autor | Histórico

-- de : ---------------+-------------+----------------+--------------

-- Alterações 01) : 21/07/2008 | 13:30:00 | Germano |

-----------------------------------------------------------------------------------------

AS

BEGIN

BEGIN TRANSACTION

BEGIN TRY

--

-- Descrição do Procedimento

--

COMMIT

SET pParam5 = 0

END TRY

BEGIN CATCH

ROLLBACK

SET @pParam5 = 1

END CATCH

END

END

* **Assinatura da Função**

CREATE FUNCTION Comum.UfcDataExtenso

(

@pData DATETIME

)

RETURNS VARCHAR(70)

AS

BEGIN

------------------------------------------------------------------------------------------------------------------------------------------

-- Autor/Criação.............: Nome do Autor

-- Data de Criação..........: 05/05/2013 00:00

-- Funcionalidade/Uso..: Breve resumo da funcionalidade

------------------------------------------------------------------------------------------------------------------------------------------

-- Histórico : Data | Hora | Autor | Histórico

-- de : -----------------+-------------+--------------+----------------------------------------------------------------

-- Alterações 01) : 21/07/2008 | 13:30:00 | Germano | Converter a data para formato extenso

------------------------------------------------------------------------------------------------------------------------------------------

DECLARE @vDia CHAR(2)

DECLARE @vMes CHAR(2)

DECLARE @vAno CHAR(4)

DECLARE @vRetorno VARCHAR(80)

SET @vDia = CONVERT(CHAR(2),(SELECT RIGHT('0000'+ CONVERT(CHAR(2),DAY(@pData)),2)))

SET @vMes = CONVERT(CHAR(2),(SELECT RIGHT('0000'+ CONVERT(CHAR(2),MONTH(@pData)),2)))

SET @vAno = CONVERT(CHAR(4),(SELECT RIGHT('0000'+ CONVERT(CHAR(4),YEAR(@pData)),4)))

SET @vRetorno = CONVERT(CHAR(2), @vDia) +

' de ' + (SELECT Descricao

FROM UvwMes

WHERE codigo = @vMes) + ' de ' + CONVERT(CHAR(4), @vAno)

RETURN (@vRetorno)

END

* **Assinatura da Trigger**

CREATE TRIGGER Schema.TrgNomeTrigger

ON Schema.Tabela FOR UPDATE

AS

BEGIN

-----------------------------------------------------------------------------------------------------------------------------------

-- Autor/Criação............: Nome do Autor

-- Data de Criação.........: 05/05/2013 00:00

-- Funcionalidade/Uso..: Breve resumo da funcionalidade

-----------------------------------------------------------------------------------------------------------------------------------

-- Histórico : Data | Hora | Autor | Histórico

-- de : -----------------+-------------+--------------+-----------------------------------------------------

-- Alterações 01) : 21/07/2008 | 13:30:00 | Germano | Converter a data para formato extenso

-----------------------------------------------------------------------------------------------------------------------------------

UPDATE Tabela

SET Tabela.Campo2 = Inserted.campo2

FROM Tabela

JOIN Inserted

ON Tabela.Campo1 = Inserted.Campo1

END

* **Sugestão de padrões para criar restrições no modelo físico.**

|  |  |
| --- | --- |
| **Identificação** | **Descrição do Atributo** |
| PK | Nome da chave primaria de uma determinada tabela. |
| FK | Nome da chave estrangeira associada a uma determinada tabela. |
| NDX | Nome de índices associados a uma determinada tabela. |
| MN | Nome de uma restrição do tipo not null, coluna obrigatória. |
| CK | Nome de uma restrição do tipo check, ou seja, valores pré-definidos. Exemplo: coluna sexo, valores possíveis ou de armazenamento. |
| USP | Nome do Procedimento. |
| UFC | Nome da função. |
| TRG | Nome da trigger associada a uma determinada tabela. |

* **Plano de Gerenciamento de**
* **Check List**

CheckList diário do ambiente de banco de dados, onde periodicamente deve ser enviado um e-mail para todos os responsáveis pelo monitoramento do banco de dados com todas as informações abaixo:

* Tempo de disponibilidade do SQL Server.
* Informações sobre espaço em disco das bases.
* Informações sobre os arquivos MDF. É possível ver se ele já está no limite de crescimento e pode causar uma parada do banco de dados.
* Informações sobre os arquivos LDF também.
* Monitoramento do crescimento das databases nos últimos 1, 15, 30 dias.
* Monitoramento do crescimento das tabelas nos últimos 1, 15, 30 dias.
* Informações sobre a Escrita nos arquivos do banco de dados.
* Informações sobre a Leitura nos arquivos do banco de dados.
* Monitoramento das bases que não possuem backup nas últimas 16 horas.
* Monitoramento dos backups realizados nas últimas 24 horas.
* Monitoramento das queries que estão em execução a mais de 2 horas.
* Monitoramento dos jobs que estão em execução a mais de 10 minutos.
* Informações de todos os Jobs que foram alterados nas últimas 24 horas.
* Informações sobre todos os jobs que falharam nas últimas 24 horas.
* Tempo de execução dos jobs das últimas 24 horas.
* Acompanhamento das queries mais lentas do dia anterior.
* Resumo dos valores dos principais contadores do dia anterior.
* Informações sobre a quantidade de conexões abertas no banco de dados.
* Acompanhamento de como está a Fragmentação de índices das tabelas.
* Acompanhamento dos maiores Waits do Banco de Dados no dia anterior.
* Informações sobre os alertas que estão sem CLEAR.
* Informações sobre os alertas do dia anterior.
* Informações sobre erro de falha de login no SQL Server.
* Informações do Log do SQL Server para análise.