NORMA DE PADRONIZAÇÃO

DA NOMENCLATURA DE OBJETOS

DO BANCO DE DADOS

1. Introdução

O presente documento tem por objetivo propor uma forma de padronização para os nomes de objetos gerenciados pelo Sybase, levando em consideração a importância que ela tem para a administração dos dados da empresa no que tange à facilidade de identificação e localização proporcionada.

O documento apresenta os objetos de banco de dados com três itens: regras, sintaxe, e exemplo, isto para facilitar o entendimento dos desenvolvedores, analistas e quem se utiliza deste manual.

Esta proposta está aberta a sugestões que possam melhorá-la tanto no aspecto funcional como no operacional. Funcionalmente, o objetivo é facilitar a recuperação de objetos de um mesmo tipo, com identificação imediata do aplicativo ao qual pertencem. Operacionalmente, devemos nos preocupar com os meios possíveis para controlar a padronização de forma automática.

SUMÁRIO DE INFORMAÇÕES DO DOCUMENTO			
Documento: Norma de padronização da nomenclatura de objetos do banco de dados			
Versão	**Data**	**Mudanças**	**Autor**
0.24	10/07/2017	Alteração da versão 0.23	Luciene Santos
0.24	13/07/2017	Inclusão das regras das siglas.	Luciene Santos
0.25	16/08/2018	Ajuste na regra da Trigger	Luciene Santos
0.26	11/11/2019	Colocado o prefixo AutoOperacao_	Luciene Santos
0.27	05/10/2020	Colocação do tipo BigDateTime e BigTime que são DateTime	Luciene Santos
0.28	03/03/2021	Inclusão da regra de índice com mesmo ano de pesquisa	Luciene Santos
0.29	10/11/2025	Atualização nas regras gerais, inclusão do manual de bolso, inclusão da regra de proxy table, inclusão da regra da equipe de infrações, inclusão do tipo Booleano.	Draomiro Aguiar

2. Regras Gerais

2.1 Caracteres permitidos

✅ Usar apenas letras (A–Z, a–z), números (0–9) e _ (underline).

❌ Não usar acentos, cedilha (ç), espaços.

❌ Não usar caracteres especiais (#, @, %, $, !, *, +, -, /, =).

2.2 Forma dos nomes

✅ Primeira letra de cada palavra maiúscula (Ex.: `ParcelaDebito`).

✅ Usar termos em português e no singular.

✅ Usar nomes curtos, claros e sem ambiguidade.

❌ Evitar preposições (Ex.: “de”, “da”, “do”).

2.3 Limite de tamanho

✅ Máximo de 30 caracteres (se ultrapassar, usar abreviações coerentes).

2.4 Restrições

❌ Não usar palavras reservadas (INSERT, DELETE, SELECT...).

❌ Não usar apenas números, verbos ou nomes próprios.

2.5 Siglas

✅ Siglas oficiais: primeira letra maiúscula e demais minúsculas (Ex.: `Ipva`, `Cnh`).

2.6 Padrão de nomes (manual de bolso)

Objeto	Padrão	Exemplo
Banco	**db** + área (até 5 posições) + seq	**dbhbio01**
Tabela	Singular, claro sem abreviação	**Veiculo**
Tabela Log	**Log** + nome da tabela	**LogParcelaDebito**
Tabela Temp	**temp** + nome tabela	**tmpVeiculoI**
Tabela "z"	**z** + login + objetivo da procedure	**zkrmlduplicidadedebitoLimpar**
Proxy Table	**px** + Origem + NomeObjeto	**pxProtLaudoToxicologico**
Coluna	tipo + NomeColuna	**nValorPagar, nCpf**
PK (Primary Key)	**pk** + NomeTabela	**pkVeiculo**
FK (Foreign Key)	**fk** + TabelaFilha + TabelaPai	**fkProcessoUsuario**
Unique	**u** + NomeTabela + Coluna	**uUsuarioEmail**
Check	**chk** + NomeTabela + Coluna	**chkUsuarioSexo**
View comum	**vw** + NomeTabela	**vwUsuarioProcesso**
View materializada	**vm** + Objetivo[Complemento]	**vmProcessoUsuario**
Índice	tabela + _ + Coluna	**Usuario_nCpf**
Procedure	Objetivo + [Complemento] + Operação	**RegistroCfcCategoriaA, AtendimentoE**
Trigger	**tg** + Tabela + Operação	**tgTabelaI**

3. Padronização dos nomes dos objetos

3.1 Banco de Dados:

O nome do banco de dados deverá identificar o negócio que está sendo automatizado ou deverá refletir a sigla da aplicação.

Nome do banco: dbaaaaann,

Onde: **db** define que é uma database;

aaaaa define a área, o negócio (até 5 posições);

nn define um seqüencial.

Exemplo: dbhbio01, dbhpop01, dbimp01.

3.2 Tabelas:

Utilização dos nomes completos dos termos que compõem o nome da tabela, excetuando-se quando isto não for possível devido à limitação da quantidade de caracteres que o SGBD impõe (30 caracteres). Neste caso, os termos devem ser abreviados sem perder a coerência do entendimento;

Exemplo: ControleLicencaBiometriaCredenciado poderia ficar:

ControleLicencaBioCredenciado

ControleLicencaBiometriaCreden

O nome de uma tabela deverá ser sugestivo. Deve-se fazer o uso de nomenclatura orientado a objeto, por exemplo: se no departamento financeiro for necessário manter uma tabela de feriados, esta tabela deve ser nomeada identificando claramente seu conteúdo, isto significa que seu nome então deverá ser Feriado; O nome da tabela deve estar sempre no singular;

Exemplo: “Veiculo” no lugar de “Veiculos”

Quando a tabela for derivada de relacionamento N:N, o seu nome deverá ser composto pelos nomes das entidades relacionadas;

Exemplo: ServicoVeiculo

O campo identificador / Chave-Primária deve ser referente ao nome da tabela;

Obs.: Não utilizar id, cod ou semelhante na chave identificadora.

Exemplo: pkFeriado

Utilização do 1º caracter de cada termo em maiúsculo, ou seja, primeira letra em maiúscula, demais em minúsculas. Para cada palavra interna, primeira letra em maiúscula, notação húngara;

Exemplo: ParcelaDebito, MarcaVeiculo, Veiculo.

Siglas como IPVA, CNPJ, CNH, FEBRABAN obedecerá a mesma regra sendo a primeira letra maiúscula e as demais minúsculas.

Exemplo: Ipva, Cnpj, Cnh, Febraban e etc.

Obs.: Notação Húngara é a diferenciação dos elementos de uma dada nomenclatura através do uso de letras maiúsculas e letras minúsculas. Tem a sua origem na codificação de programas. Para o nosso padrão de nomenclatura vamos utilizar uma versão adaptada desta notação que é o uso da letra maiúscula no início de cada elemento do nome.

Quando a tabela for de Log, deve-se usar o nome da tabela antecedida do termo “Log”.

Obs.: Tabelas de Log **não terão** chave estrangeira (fk) no entanto terão chave primária(pk).

Exemplo: LogParcelaDebito

Tabela temporária(tmp): É uma tabela que é usada de forma auxiliar, cujo conteúdo será excluído pela própria aplicação.

Exemplo: tmpVeiculoI

Tabela do tipo “z”: São tabelas que serão excluídas do banco de dados. Sua nomenclatura deverá ser composta com a letra “z” no início + login do criador + nome da tabela.

O nome da procedure deve ser composto da seguinte forma: z + login + objetivo da

procedure (não podemos colocar o nome da tabela, pois normalmente esse tipo de procedure não afeta apenas uma tabela)

Exemplos: zkrmlduplicidadedebitoLimpar

Tabela do tipo Proxy: As Proxy tables são tabelas “espelho” ou de referência que permite acessar dados que estão em outro banco de dados ou servidor, como se estivessem no banco local. Sua nomenclatura deverá ser composta por px<Origem><NomeDoObjeto>

Exemplo: pxProtLaudoToxicologico

3.3 Colunas:

Seguindo o mesmo padrão utilizado para tabelas, deve-se identificar a coluna da tabela de maneira clara e descritiva.

Na primeira posição deve-se utilizar, em letra minúscula, o tipo do dado, seguido pelo nome identificador da coluna com o 1º caracter de cada termo em maiúsculo;

Exemplo: nValorPagar

O nome da coluna deve ser o mais próximo possível do seu objetivo / significado.

Siglas como IPVA, CNPJ, CNH, FEBRABAN obedecerá a mesma regra sendo a primeira letra maiúscula e as demais minúsculas.

Exemplo: Ipva, Cnpj, Cnh, Febraban e etc.

Os sistemas de Veículos e os sistemas da equipe de Infrações referentes a veículos **não utilizarão** o tipo do dado na frente do nome da coluna (**exceto se houver a necessidade de desenvolver um sistema que use um novo banco de dados)**, no entanto, todas as outras regras de padronização devem ser seguidas conforme o manual.

Exemplo: Chassi (Sistemas já existentes Bancos Legados)

sChassi (Sistemas em Bancos Novos)

No caso das tabelas que possuem procedures ligadas a elas (external procedure) as colunas que serão usadas como passagem de parâmetros deverão iniciar seu nome com underline.

Exemplo: _sNome(Habilitação), _Nome(Veículo).

As tabelas de Log que são uma réplica de uma tabela do sistema já existente terão a constituição dos nomes das suas colunas idênticas aos da tabela já existente. No caso das colunas que não existam na tabela do sistema a nomenclatura dessas colunas deverá seguir o padrão já existente.

Exemplo: Veiculo(Chassi) ➡ LogVeiculo(Chassi)

Usuario(nCpf) ➡ LogUsuario(nCpf)

3.3.1 Tipo de Dado:

Tipo do Dado	Sigla do Tipo
bit	b
datetime, smalldatetime,bigdatetime, bigtime	d
text, image, binary, long	I
money, smallmoney	m
numeric, int, smallint, tinyint float, real, double precision	n
char, varchar	s
Time	T
Booleano	bo

Exemplo: nCpf, sChassi.

Se o campo for um identificador, deve-se colocar na segunda posição a letra “i” em minúsculo;

Exemplo: n**i**Pagamento (Indicativo do Pagamento, onde: 1 = On-Line, 2 = Manual).

3.3.2 Chaves Primárias (pk):

Utilização dos atributos considerados identificadores naturais das tabelas, acrescidos do nome da tabela;

Exemplos:

Tabela ➡ Profissional

Chave Primária ➡ nCpfProfissional

Tabela ➡ Oficio

Chave Primária composta ➡ AnoOficio / NumOficio

Se a chave primária for um número seqüencial, deve-se utilizar o nome da tabela

Exemplo: Tabela ➡ Processo então Chave Primária ➡ nProcesso;

Quando a tabela for de relacionamento, os nomes dos campos que compõem a chave primária devem ser mantidos dos campos que foram trazidos das tabelas de origem. Esses campos serão ao mesmo tempo chaves primárias e estrangeiras;

Exemplo:

sServico (pk) **sServico (pk),(fk)** nTaxa (pk)

nTaxa (pk),(fk)

3.3.3 Chaves Estrangeiras (fk):

Utilização do nome do campo da Tabela Pai (nome de origem);

Exemplo:

nVeiculo (pk) nProcesso (pk)

nVeiculo (fk)

Se o campo não for chave primária na tabela de origem, deve-se acrescentar o nome da tabela-pai ao nome do campo.

Exemplo:

nVeiculo (pk) nProcesso (pk)

sPlaca (ak) (*ak - alternate key) **sPlacaVeiculo (fk)**

Quando existir mais de uma “fk” para a mesma tabela pai, deve-se usar um número sequencial no final do nome da “fk”.

Exemplo: fkVeiculoCategoriaVeiculo01

3.4 Nomes de Constraints:

Primary key: Usar o prefixo “pk” mais o nome da tabela.

Exemplo: pkNomeTabela

pkVeiculo

Foreign key: Usar o prefixo “fk” mais os nomes das tabelas filha e pai.

Exemplo: fkTabelaFilhaTabelaPai

fkProcessoUsuario

Unique: Usar o prefixo “u” mais o nome da tabela mais o nome da coluna.

Exemplo: uNomeTabela + Coluna

uUsuarioEmail

Check: Usar o prefixo “chk” mais o nome da tabela mais o nome da coluna.

Exemplo: NomeTabela + Coluna

chkUsuarioSexo

3.5 Nome de View:

Views podem ser adotadas para queries complexas, envolvendo mais de uma tabela, cujo formato de saída dos dados selecionados seja considerado padrão para uma ou mais transações do sistema. O uso de view pode melhorar o tempo de resposta da

consulta, pois o comando SELECT é interpretado no momento de criação da view, e não na sua execução.

Views também podem ser adotadas por questões de segurança, como um instrumento para conceder aos usuários permissão de acesso somente a determinados atributos e/ou a informações já consolidadas.

Views devem ser utilizadas somente para consultas. As tabelas envolvidas em uma view não devem ser atualizadas indiretamente, via comandos de INSERT, UPDATE e DELETE efetuados na view.

Deve-se usar a mesma semântica utilizada para as tabelas. Para views não materializadas deve ser prefixada a palavra “vw” seguido do nome da tabela.

Exemplo: vwUsuarioProcesso

View Materializada também deve utilizar a semântica das tabelas.

Onde:

vm: prefixo que identifica uma view materializada.

Objetivo: Usar substantivo que responda a pergunta: O que selecionar? Muitas vezes coincidirá com o nome da tabela.

Complemento:

Opcional. Caso o objetivo não seja suficiente para traduzir a função da view materializada acrescenta-se um complemento / descrição para melhor definição da mesma.

Exemplo: vmObjetivo[Complemento]

vmProcessoUsuario

3.6 Índice:

Nome da tabela mais underline seguido do nome da primeira coluna que compõe o índice. Usar abreviações se o nome do campo for muito grande.

Exemplo: Tabela_PrimeiroCampo

Usuario_nCpf

Quando existir mais de um “índice” com o mesmo campo, deve-se usar um número sequencial no final do nome do “índice”.

Exemplo: TipoObjeto_nObjeto01

3.7 Procedures:

Nome de procedure: Objetivo[Complemento]Operacao,

Onde:

Objetivo: Usar substantivo que responda às perguntas: O que selecionar? O que inserir? Neste caso, muitas vezes coincidirá com o nome da tabela.

Complemento: Opcional. Caso o objetivo não seja suficiente para traduzir a função da procedure, acrescenta-se um complemento / descrição para melhor definição da mesma.

Operação: Usar sigla, em letra maiúscula, das operações básicas: Selecionar (S), Inserir (I), Excluir (E), Alterar (A), Relatório (R).

Exemplo: RegistroCfcCategoriaA

AtendimentoE

Se a operação efetuada não for uma das citadas anteriormente, deve-se utilizar o nome do verbo (operação) completo no infinitivo no fim do Objetivo[Complemento]**Operação**.

Exemplo: TurmaCandidato**Vincular**

ApreensaoVeiculo**Gerar**

Siglas como IPVA, CNPJ, CNH, FEBRABAM, AutoOperacao_, obedecerá a mesma regra sendo a primeira letra maiúscula e as demais minúsculas.

Exemplo: Ipva, Cnpj, Cnh, Febrabam, AutoS_Acessologado e etc.

No sistema de Habilitação as procedures de envio do RENACH ficarão iguais às procedures já existentes do RENACH.

Exemplo: RenachProcessoAtualizar

As procedures usadas pela FISEPE (SEFAZ) irão iniciar com as letras “FI” (maiúsculo).

Exemplo: FITipoDebitoS

No sistema de Veículos, os nomes das procedures RENAVAM e RENAINF ficarão iguais aos já existentes. Se for no banco RENAVAM todos os padrões serão mantidos, mas se for em outro banco, só se houver algum termo que indique que a procedure faz parte de um desses projetos.

Exemplo:

Banco RENAVAM: DetranResp206, PartResp206, rnvDetranResp206...

Outros Bancos: dbvcen..**bin**ConsultaVeiculoS_EV02,

dbvcen..Transacao**bin**920, dbinfracao..**rnf**CodRetornoS…

Quando a procedure for executada via processamento batch, deve-se colocar no início do nome da procedure a palavra “Batch”.

Exemplo: BatchNomeprocedure

Quando a procedure for de Habilitação e for executada em Veículo ou o contrário, deve-se colocar no início do nome da procedure a palavra “Rpc”.

Exemplo: RpcNomeprocedure

Quando a procedure for para acesso via internet, incluir a letra i, em minúsculo, como prefixo no nome da procedure, ou seja, só é usada pelas aplicações web.

Exemplo: iNomeprocedure

Nas procedures de acesso à Internet os nomes das variáveis que são nomes de colunas nas suas respectivas tabelas, deve ser igual ao nome da coluna antecedido do @.

Exemplo: @nTaxa

Para variáveis que não são nomes de colunas, deve-se colocar o @ e seguir as mesmas regras utilizadas para nomeação de colunas.

Exemplo: @nValorAux

Na Procedure definir uma área de identificação geral, onde deverão existir informações seguir o padrão abaixo:
Fazer uma descrição clara e objetiva da função da procedure;
Colocar a data de criação e o responsável pela criação da procedure (quem alterar a procedure deve acrescentar o nome do responsável pela alteração, a data e o objetivo da alteração);

Exemplo:

```
CREATE PROCEDURE ProcessoS
/******************************************************************************
** CRIADA POR: Nome do desenvolvedor
** DATA: 01/08/03
** OBJETIVO: Seleciona processos cadastrados a partir de uma determinada data.
**
******************************************************************************/
@dAbertura    smalldatetime,        // Cadastramento do Processo
@nProcesso    numeric(13,0) = 0     //  Número do Processo
...
```

Identificar as alterações efetuadas na área apropriada, para permitir fácil localização de problemas no código;

Indentar o código de modo a tornar a codificação clara e facilitar o trabalho de manutenção, pode-se usar para indentar a ferramenta SQL Expert;

Ao longo da procedure, inserir comentários sempre que necessário. Comentários adicionais que auxiliem a compreensão de problemas complexos. Não poluir o código com comentários desnecessários, que descrevem procedimentos óbvios;

Utilização das palavras reservadas SQL em letras maiúsculas, inclusive os tipos de dados (WHERE, FROM, NUMERIC, etc...);

Quando houver join, as colunas devem ser prefixadas de preferência com a primeira letra do nome da tabela;

Tratando-se nome de tabela composto, usar as primeiras letras das palavras;

Caso seja necessário, usar duas, três ou mais letras para diferenciar os alias.

Evitar o aninhamento excessivo de comandos, o que costuma dificultar a manutenção do código. Dar preferência à codificação mais longa, porém mais clara, desde que não prejudique a performance.

3.8 Triggers:

Usar o prefixo “tg” seguido do nome da tabela onde está sendo disparada ação e da sigla do evento (I = Inserir, A = Alterar, E = Excluir).
Usar uma trigger para cada evento.

Exemplo: tgTabelaI

tgTabelaA

4. Tipos de Dados:

Tipos de dados	Faixa de valores	Espaço ocupado
Bit	0 ou 1	1 byte
Int	- 2.147.483.648 a 2.147.483.647	4 bytes
Smallint	- 32.768 a 32.767	2 bytes
Tinyint	0 a 255	1 byte
Text		Até 2.147.483.647 bytes
Image		Até 2.147.483.647 bytes
Char(n)	Range igual à página do servidor 2K = 1948 caracteres (lock allpage) e 1954 (lock datarow)	N bytes
Varchar(n)	Range igual à página do servidor 2K = 1948 caracteres (lock allpage) e 1954 (lock datarow)	N bytes
Numeric(p,s)	- 10E38 a 10E38 - 1	Depende da precisão (P)
Datetime	1/1/1753 a 31/12/9999 precisão: milissegundos	8 bytes

Smalldatetime	1/1/1900 a 6/6/2079 precisão: minutos	4 bytes
Float	Depende da máquina	4 ou 8 bytes
Double precision	Depende da máquina	8 bytes
Real	Depende da máquina	4 bytes
Binary(n)	Range igual à página do servidor 2K = 1948 caracteres (lock allpage) e 1954 (lock datarow)	N bytes
Money	922.337.203.685.477,5807 a 922.337.203.685.477,5807	8 bytes
Smallmoney	214.748.3648 a 214.748.3648	4 bytes

Atributos de Uso Comum:

Atributos	Tipos de Dados
Pessoas	Varchar(50)
E-mail	Varchar(60)
Telefone	Varchar(10)
Fax	Varchar(10)
Logradouro	Varchar(60)
Complemento	Varchar(65)
CEP	Numeric(8)
Bairro	Varchar(60)
Município	Varchar(60)
País	Varchar(60)
CGC	Char(14)
CPF	Char(11)
Login	Varchar(30)

4.1 Função de Usuário:

Função tipo 1:

sp_f_Objetivo[Complemento] ➡ Função Global, ou seja roda de qualquer banco.

Prefixo sp_f identifica uma função para manipular strings e/ou números ou objetos de sistema, que são comuns a todos os bancos.

Função tipo 2:

sp_f_Nomedobanco_Objetivo[Complemento] ➡ Roda apenas de um banco de dados específico.

Prefixo p_f_nomedobanco: identifica uma função que referencia objeto de banco (tabela, procedure, view, etc) pertencente ao respectivo banco (sugestão: retirar o prefixo db do nome do banco, ex: sp_f_arrec)

Objetivo: usar um substantivo que indique a finalidade a função

Complemento: Opcional. Caso o objetivo não seja suficiente para traduzir a finalidade da função, acrescenta-se um complemento / descrição para melhor definição da mesma.

Obs.: Todas as funções ficarão armazenadas no banco sybsystemprocs, que é o banco padrão de armazenamento de procedure de sistema, como por exemplo: sp_help, sp_who, etc.
O prefixo sp é necessário para que seja possível a pesquisa da mesma, sem a indicação do banco

5. Recomendações e boas práticas

Para programação nos objetos de dados (procedures, triggers, ...):

Todos os comandos SQL utilizados por programas devem ser realizados, obrigatoriamente, através de Stored Procedure, exceto updates e inserts para colunas dos tipos texto e imagem;

Integridade referencial deve ser implementada através de constraints (Primary Key, Unique e Foreign);

Preferencialmente não utilizar a estrutura de programação cursor;

Preservar a cláusula abaixo no script, antes do cabeçalho, a fim de agilizar a criação de procedures e triggers armazenadas no repositório.

```
IF EXISTS (SELECT 1 FROM sysobjects WHERE name = ‘NomeProcedure')
BEGIN
   DROP PROCEDURE NomeProcedure
END
GO

CREATE PROCEDURE NomeProcedure

IF EXISTS (SELECT 1 FROM sysobjects WHERE name = 'NomeTrigger')
BEGIN
   DROP TRIGGER NomeTrigger
END
GO

CREATE TRIGGER NomeTrigger
```

Devem ser evitados tipos de dados definidos pelos usuários;

Atributos de quantidade escolher entre os tipos a seguir, dependendo da faixa de valor: TINYINT, SMALLINT, INT;

Atributos de data / hora usar SMALLDATETIME ou DATETIME dependendo da precisão necessária para o cálculo de horas, bem como da faixa de valores de cada;

Atributos com conteúdo alfanumérico com mais de 255 caracteres, usar tipo Varchar, preferencialmente;

Atributos de imagem usar tipo IMAGE;

Evitar join com mais de 4 tabelas, quando for necessário, utilizar tabelas temporárias;

Evitar a utilização de convert na cláusula where;

Dar preferência a construções do tipo: SELECT iCol, iCol2 from Table WHERE iCol >= 10 a construções do tipo: SELECT iCol, iCol2 from Table WHERE iCol > 9;

Evitar cláusula NOT EXISTS, NOT IN e NOT LIKE. Usar se possível EXISTS, IN e LIKE;

Varchar x char: Deve ser aplicado caso a caso. Sendo o dado de tamanho fixo e not null, deve-se usar o tipo char. Outra característica que devemos analisar é se o dado sofrerá muitas atualizações. Caso seja verdade, ele é um forte candidato a ser do tipo char, para se ter ganho de performance. O tipo varchar deve ser utilizado visando economizar em área de armazenamento;

Not null x null: Deve ser aplicado caso a caso também, já que o campo null é tratado como tipo variável. Lembrando que em muitas situações podemos usar um valor default, para aplicação do campo not null;

Algumas vezes o sybase não consegue usar as estatísticas para uma variável inicializada em tempo de execução, implicando em perda de performance. Então, a forma de resolver esse problema é fazer um split de procedure, que significa dividir a procedure em duas, como o exemplo que segue:

```
DECLARE @sCity varchar(25)
SELECT @sCity = sCity
FROM Publishers WHERE sName = @sName
SELECT sAuName
FROM Authors  WHERE sCity = @sCity
```

Split da procedure:

```
CREATE PROCEDURE AuNamesS
   @sName varchar(30)
AS
```

```
    DECLARE @sCity VARCHAR(25)
    SELECT @sCity = sCity
    FROM Publishers
    WHERE sName = @sName
    EXEC ProcS @sCity

CREATE PROCEDURE ProcS
    @sCity VARCHAR(25)
AS
    SELECT sAuName
    FROM Authors
    WHERE sCity = @sCity
```

Obs.: Qualquer dúvida em relação a estas dicas de programação no banco de dados conversar com o DBA.

6. Dicionário de Termos:

O dicionário de termos tem por objetivo oferecer um catálogo contendo os termos a serem utilizados para se dar nomes aos objetos de dados do ambiente do Detran-PE contêm palavras, suas abreviaturas e remissões para o uso de sinônimos.

A inclusão dos termos deve ser feita pelos coordenadores das equipes de desenvolvimento de sistemas, apenas eles e a equipe de suporte ao desenvolvimento terão acesso à manutenção deste catálogo.

Antes de incluir um termo, deve ser efetuada uma pesquisa, de forma a certificar-se de que dentre os termos já cadastrados não haja outro com o mesmo sentido que possa ser utilizado.

Os termos homógrafos (que tem a mesma grafia) não deverão ser utilizados, portanto cada termo será sempre utilizado com um, e somente um, sentido.

A seção de suporte ao desenvolvimento é responsável pela validação dos termos incluídos, cabendo a ela a última palavra sobre a utilização ou não de determinado termo ou abreviatura.

Exemplo:

Termo	Abreviação	Sinônimo
Data	Data	Datas
Descricao	Desc	Descricoes
		Descritivos
		Descritiva
		Descritivas
		Descritivo
Dia	Dia	Dias
Quantidade	Quant	Quantidades
		Quantid
		Qtd
		Qtde
Selecionar	S	Buscar
		Ler
		Pesquisar
		Consultar
		Mostrar

Obs.: Por enquanto não está se utilizando o dicionário de termos para os objetos de dados.