



MANUAL DE PADRÕES E NORMAS

PARA ADMINISTRAÇÃO DE DADOS

Versão 1.0

HISTÓRICO

Data	Versão	Descrição	Autor
18/11/2013	1.0	Criação do Documento.	Janaina Rabello

Sumário

1. INTRODUÇÃO	5
2. CONVENÇÃO BÁSICA PARA NOMENCLATURA DE OBJETOS DE BANCOS DE DADOS	6
3. NOMENCLATURA DE OBJETOS DE BANCO DE DADOS	10
4. NOMENCLATURA DE COLUNAS DE TABELAS E <i>VIEWS</i> DE BANCO DE DADOS	12
5. NORMAS GERAIS PARA MODELAGEM E CRIAÇÃO DE OBJETOS DE BANCO DE DADOS	13
5.1 Regras gerais	14
5.2 Restrição de Integridade	14
5.2.1 Restrições de chaves	14
5.2.2 Restrições de Integridade Referencial	15
5.2.3 Restrições de Domínio	16
5.3 Atributos multivalorados	18
5.4 Obrigatoriedade de colunas	18

5.5	Definição de tamanho de campos do tipo texto.....	18
5.6	Tabelas existentes em outros sistemas (schemas)	19
5.7	Padrão de nomenclatura para dados de endereço definido pelos Correios	19
5.8	Armazenamento de arquivos.....	20
5.9	Tabelas temporárias.....	21
5.10	Padrão de cabeçalho para <i>packages, body packages, stored procedures, functions e triggers</i>	21
5.11	Padrão de cabeçalho <i>views e materializeds views</i>	22
5.12	Indexes.....	23
6.	USUÁRIOS.....	23
7	DICIONÁRIO DE DADOS	24
7.1	Regras Gerais	25
7.2	Modelo de Dicionário de Dados	29
8.	AVALIAÇÃO DO MODELO DE DADOS.....	32

1. INTRODUÇÃO

A atividade de Administração de Dados trás em seu cerne a preocupação em tornar seguro o entendimento das informações que estão sendo armazenadas, objetivando o bom entendimento do negócio da organização, a projeção adequada das bases de dados, o compartilhamento de informações e a integração entre os sistemas de informação.

A implantação de uma política de Administração de Dados (AD) visa minimizar riscos quanto à complexidade da informação armazenada – levando em consideração a relativa diversidade de sistemas de informação disponíveis no Ibram -, além de prover qualidade no controle e manutenabilidade da informação a ser gerida, de forma a tornar transparente o universo heterogêneo de sistemas existentes no Instituto. Apresenta-se este MANUAL DE PADRÕES E NORMAS PARA ADMINISTRAÇÃO DE DADOS –

resultado de análise do atual cenário, benchmarking e boas práticas – cujo objetivo é suprir a necessidade de gerir a informação proveniente dos sistemas de informação.

As informações descritas visam estabelecer uma convenção de nomenclatura e modelagem de dados a ser seguida por todos os envolvidos no processo de desenvolvimento de novas aplicações que utilizam o armazenamento de informações em banco de dados.

Os objetos legados não serão afetados, entretanto novos objetos criados e/ou alterados, mesmo que derivados dos objetos legados, devem se enquadrar aos padrões e normas de nomenclatura estabelecidos neste MANUAL DE PADRÕES E NORMAS PARA ADMINISTRAÇÃO DE DADOS.

A adoção de um padrão e normatização de processos de criação de objetos nas bases de dados tem por base a uniformização do entendimento acerca das informações que o Ibram produz – diminuindo a complexidade dos sistemas -, garantindo a integridade de suas informações, mitigando riscos de desentendimento, garantindo a manutenabilidade dos objetos existentes e/ou objetos a serem construídos e disponibilizados na base de dados.

2. CONVENÇÃO BÁSICA PARA NOMENCLATURA DE OBJETOS DE BANCOS DE DADOS

Os objetos de banco de dados devem obedecer aos seguintes critérios básicos no que diz respeito à nomeação de objetos:

- Uso de caracteres alfabéticos apenas;
- Letras maiúsculas;
- Para determinado nome que é composto por mais de uma palavra, deve-se utilizar o caractere *underscore* ('_') como separador entre cada palavra;
- Palavras no singular;
- Não uso de acentuação gráfica e nem de caracteres especiais (tais como @, #, \$, %, &, etc.);

Abaixo segue quadrante contendo informações relacionadas ao tamanho máximo de nomes de objetos permitidos para cada tecnologia de banco de dados:

OBJETO	Quantidade máxima de caracteres permitida por SGBD	
	SQLServer	MySQL
<i>Column</i>	30	30
<i>Constraints</i>	30	30
<i>Cursor</i>	30	30
<i>Function</i>	30	30
<i>Indexes</i>	30	30
<i>Materialized View</i>	30	30

Padrões e Normas para Administração de Dados

<i>Package</i>	30	30
<i>Package Body</i>	30	30
<i>Password</i>	-	-
<i>Savepoint</i>	30	30
<i>Schema</i>	30	30
<i>Sequence</i>	30	30
<i>SQL variable name</i>	30	30
<i>Stored Procedures</i>	30	30
<i>Synonyms</i>	30	30
<i>Table</i>	30	30
<i>Trigger</i>	30	30
<i>View</i>	30	30

Tabela 1 Quantidade máxima de caracteres permitida por tecnologia SGBD para nomeação de objetos de banco de dados.

O uso de acrônimos e/ou abreviações é permitido quando o tamanho do nome do objeto ultrapassar o tamanho máximo de caracteres permitido – de acordo com a tabela acima. Os acrônimos devem seguir a convenção nacional, como por exemplo, CEP, CPF, etc. Já as abreviaturas, estas, devem seguir as regras básicas abaixo:

Regra geral para abreviação de palavras:

- Mantém-se a primeira sílaba e a letra seguinte da próxima sílaba da palavra a ser abreviada;
- Caso a primeira letra da segunda sílaba seja uma vogal, escreve-se até a consoante.
- Caso a segunda sílaba inicie-se por duas consoantes, as duas consoantes farão parte da abreviatura.
- Exemplos:

Palavras	Abreviatura
SITUACAO	SIT
ENTIDADE	ENT
PESSOA	PESS
SECRETARIA	SECR

Tabela 2 Tabela de exemplo de abreviatura de palavras.

- A palavra abreviada deve ter metade ou menos da metade da palavra original;
- As palavras abreviadas não devem conter comprimento menor do que três caracteres, salvo em casos de a palavra ser autoexplicativa, como 'kg', 'GB', 'MB';

3. NOMENCLATURA DE OBJETOS DE BANCO DE DADOS

Abaixo estão apresentados os padrões de nomenclatura de objetos de banco de dados. Todos os objetos são identificados por intermédio do prefixo indicador do tipo de objeto, seguido do separador universal ('_'), somado ao nome do objeto propriamente dito. O quadrante apresenta exemplos de nomes dos objetos listados para facilitar o entendimento.

Objeto	Tipo	Padrão de Nomenclatura de objetos de banco de dados		Exemplos
		Prefixo	Complemento	
Constraints	Check Constraint	CK_	<nome da tabela> + <nome do campo>	CK_NUCLEOAT_STATUS
	Foreing Key	FK_	<nome da tabela pai> + <nome da tabela filha>	FK_AREA_RESPONSAVEL
	Primary Key	PK_	<nome da tabela>	PK_MUNICIPIO
	Unique Key	UK_	<numero cardinal> + <nome da tabela>	UK_01_BENEFICIARIO
Function	-	FC_	<nome da função que indique objetivo da mesma>	FC_CALCULA_DV
Index	-	IN_	<numero cardinal> + <nome da tabela>	IN_01_PARQUE

Padrões e Normas para Administração de Dados

Materialized View	-	MV_	<nome da <i>view materialized</i> >	MV_STATUS_PLANO
Package	-	P_	<nome da package que indique objetivo da mesma>	P_ACERTO_DADOS_METAS
Package Body	-	P_	<nome do package body que indique objetivo do mesmo>	P_ACERTO_DADOS_METAS P_AMETAS
Schema	N/A	N/A	N/A	UAG;DILOG;GEREC
Sequence	-	SQ_	<nome da tabela>	SQ_RESPONSABEL_LEGAL
Stored Procedures	-	SP_	<nome da procedure que indique o objetivo da mesma>	SP_APLICA_MULTA
Table	Sistema	TB_	<nome da tabela>	TB_MUNICIPIO
		RL_	<nome da tabela 1><nome da tabela 2>	RL_SERVIDOR_PROJETO
	Temporária	TMP_	<nome da tabela>	TMP_CARGA
	Histórico	TH_	<nome da tabela>	TH_PAGAMENTO
Trigger	Sistema	TG_	<nome da trigger que indique o objetivo da mesma>	TG_ATUALIZASTATUS_SERVIDOR
	Auditoria	TGA_	<nome da tabela auditada>	TGA_CONVENIO

View	-	VW_	<nome da view>	VW_LICENCA
------	---	-----	----------------	------------

Tabela 3 Tabela de especificação de nomenclatura de objetos de banco de dados

4. NOMENCLATURA DE COLUNAS DE TABELAS E VIEWS DE BANCO DE DADOS

No quadro a seguir, é apresentado o padrão de nomenclatura de colunas de tabelas de banco de dados. Todos os nomes de colunas são identificados por intermédio do prefixo indicador do tipo de coluna, seguido do separador universal ('_'), somado ao nome do campo. O quadrante apresenta ainda, os tipos de dados para as tecnologias SQLServer e MySQL, juntamente com exemplos de nomes de colunas para facilitar o entendimento.

Tipos de colunas	Padrão de nomenclatura de objetos		Tipo de dados padrão		Exemplos
	Prefixo	Complemento	SQLServer	MySQL	
Código numérico sequencial ou não, único e identificador - que represente a chave primária da tabela.	CO_	<nome do atributo identificador>	INT	INT	CO_PESSOA CO_IBGE
Data.	DT_	<nome do atributo>	DATETIME	DATETIME	DT_INICIO DT_INICIO_CONTRATO
Descrição ou descritivo.	DS_	<nome do atributo>	VARCHAR	VARCHAR	DS_OBSERVACAO

Padrões e Normas para Administração de Dados

Hora.	HR_	<nome do atributo>	TIME	TIME	HR_EMISSAO_TITULO
Nome.	NO_	<nome do atributo>	CHAR VARCHAR	CHAR VARCHAR	NO_SECRETARIA NO_PESSOA
Número.	NU_	<nome do atributo>	INT CHAR	INT CHAR	NU_TELEFONE_PESSOA; NU_CPF
Quantidade.	QT_	<nome do atributo>	INT	INT	QT_INTEGRANTE_FAMILIA
Sigla.	SG_	<nome do atributo>	CHAR	CHAR	SG_UF
Status - representa um campo da tabela que possua duas qualificações distintas de elementos conhecidos.	ST_	<nome do atributo>	CHAR	CHAR	ST_STATUS_ATIVO
Tipo - representa um campo da tabela que possua um conjunto específico de elementos conhecido.	TP_	<nome do atributo>	CHAR	CHAR	TP_DOCUMENTO TP_ESTADO_CIVIL
Valor.	VL_	<nome do atributo>	NUMERIC	NUMERIC DECIMAL	VL_SALARIO_BRUTO

Tabela 4 Tabela de especificação de nomenclatura de colunas de tabelas de banco de dados.

5. NORMAS GERAIS PARA MODELAGEM E CRIAÇÃO DE OBJETOS DE BANCO DE DADOS

Neste tópico são apresentadas as normas e padrões para modelagem e criação de objetos de banco de dados. Os itens a seguir tratam de requisitos mínimos que devem ser apresentados em um modelo de dados. O objetivo é reduzir a complexidade e minimizar os riscos operacionais na gestão dos bancos de dados do Ibram, garantindo qualidade e uniformidade dos objetos a serem construídos e disponibilizados nos ambientes de banco de dados.

5.1 Regras gerais

- As tabelas devem possuir uma ou mais colunas:
 - Caso a tabela possua uma única coluna, esta coluna deve ser chave primária da tabela.
 - Caso a tabela não possua uma coluna chave, deve-se implementar uma coluna chave para a tabela. Comumente, faz-se o uso de *sequences* nestas situações para se garantir unicidade na geração de registros.
- Nenhum registro pré-formatado (uso de máscaras) deve ser armazenado no banco de dados.

5.2 Restrição de Integridade

Regras de integridade referem-se à *precisão* ou *correção* dos dados no banco de dados. Dessa forma, as restrições de integridade garantem que mudanças realizadas na base de dados, por usuários autorizados, não resultem em perda de consistência dos dados, protegendo assim, o banco de dados de danos acidentais. Os próximos tópicos deste item tratam os tipos mais comuns de restrições de integridade que devem ser mantidos no modelo de dados.

5.2.1 Restrições de chaves

- Chaves primárias e chaves únicas (candidatas ou alternativas) devem obter as propriedades de unicidade e irredutibilidade.
- Chaves estrangeiras devem possuir um valor correspondente ao de uma chave primária. Caso contrário, deve-se incluir uma cláusula no dicionário de dados do modelo de dados, explicitando a possibilidade de existência de registros “filhos” sem “pai”, juntamente com o motivo que levou a escolha deste tipo de opção. Para este tipo de escolha, a Geti irá avaliar se cabe o uso deste tipo de implementação, e ainda, se existem outras formas de construção do modelo que atenda o objetivo desejado.
- Todas as chaves primárias, estrangeiras e únicas (candidatas ou alternativas) devem possuir um nome seguindo os padrões de nomenclatura de objetos.

5.2.2 Restrições de Integridade Referencial

- Por via de regra, o banco de dados não deve conter quaisquer valores de chaves estrangeiras não associadas a uma chave primária. Se **B** faz referencia a **X**, então **X** deve existir.
- Existem casos em que não se é possível eliminar registros “filhos” sem pai. Para estes casos, deve-se incluir no dicionário de dados do modelo em questão, de forma clara e objetiva, os motivos que levaram a escolha de se manter no banco de dados registros sem referência. Os motivos estarão sujeitos à análise da Geti que irá avaliar se a escolha foi a melhor alternativa, e ainda, se existe outra forma de implementação possível para evitar este tipo de situação.
- Uma chave estrangeira é simples ou composta se a chave primária é simples ou composta.
- Cada coluna da chave estrangeira deve possuir a mesma formatação do campo correspondente da chave primária associada.
- Para cada relacionamento regular de cardinalidade N: M, uma tabela associativa Tⁱ deverá ser implementada contendo todos os atributos característicos do relacionamento e as chaves primárias das tabelas T1 e T2 envolvidas no relacionamento.

Dessa forma, a tabela associativa ou de controle, tem como chave primária as duas chaves estrangeiras provenientes de T1 e T2. O uso da chave primária composta pode ser substituído por uma chave artificial quando a tabela T1 participar de outros relacionamentos para evitar a transposição de chaves estrangeiras. No dicionário de dados, devem-se especificar quais são as tabelas envolvidas no relacionamento, e, nos casos de a tabela fazer parte de outros relacionamentos - incluindo agregação, o dicionário de dados também deve ser alimentado destas informações. Lembrando que o padrão de nomenclatura para este tipo de tabela é RL_<nome tabela T1><nome tabela T2>, conforme especificado no item 3. Nomenclatura de objetos de banco de dados.

5.2.3 Restrições de Domínio

- Restrições de domínio devem ser implementadas no banco de dados para garantir a integridade dos dados através de *constraints* de *check*, preferencialmente. É extremamente recomendável que estas restrições (regras de consistência) residam também na camada de aplicação.
- O uso de *procedures* e *triggers* implicará ao analista desenvolvedor a descrição em detalhes das regras de integridade envolvidas, a codificação e o teste de sua primeira versão.
- O uso de *triggers* será avaliado pela Geti quanto à sua real necessidade, à complexidade da lógica e ao volume de eventos acionadores (transações) na tabela, visto que o uso de tal recurso pode trazer impactos de performance e se mostrar ineficaz em situações de desabilitação ou invalidade;
- Restrições de atributo são obrigatórias para colunas de *status* e *tipo*. Além disso, esta regra aplica-se para quaisquer outras colunas que possuam domínio finito e conhecido. Restrições de transição de estado devem ser também implementadas para garantir que a transição de um estado correto de um atributo para outro estado seja válida.
- Todas as *constraints* de *check* implementadas devem possuir um nome seguindo os padrões de nomenclatura de objetos.

Padrões e Normas para Administração de Dados

- O dicionário de dados do modelo em questão deve explicitar de forma clara e objetiva quais são as restrições de domínio do modelo.

Exemplos de uso de restrições de Domínio			
Tipo de restrição de Domínio	Coluna	Domínio	Observações
Restrição de atributo	ST_ATIVO	S para Ativo N para Inativo	Este tipo de restrição é implementada no banco de dados através de uma <i>constraint</i> de <i>check</i> .
Restrição de atributo	TP_ESTADO_CIVIL	S para Solteiro C para Casado E para Separado D para Divorciado V para Viúvo	Este tipo de restrição é implementada no banco de dados através de uma <i>constraint</i> de <i>check</i> .
Restrição de tupla	DT_INICIAL DT_FIM	DT_INICIAL < DT_FIM	Este tipo de restrição pode ser implementado no banco de dados através de <i>constraint</i> de <i>check</i> ou através de <i>trigger</i> .
Restrição de transição de estado	TP_ESTADO_CIVIL	S → C válido S → V inválido	Se o estado inicial correto for Solteiro (S), a transição para o estado de Casado (C) é válida. Entretanto, caso o estado inicial correto seja Solteiro (S), a transição para o estado de Viúvo (V) é inválida. Este tipo de restrição pode ser implementada através de <i>trigger</i> .

Tabela 5 Tabela contendo exemplos de restrições de domínio de atributo, tupla e transição de estado.

5.3 Atributos multivalorados

- Para atributos multivalorados, uma tabela T' deve ser criada, contendo a chave primária da tabela T e o atributo multivalorado AM. A chave primária de T' será a chave primaria da tabela T e o atributo multivalorado AM. Se a chave primária de T for composta, então a tabela T' conterá todos os atributos que compõem a chave de T. Exemplo: telefone.

5.4 Obrigatoriedade de colunas

- O número de campos que aceitem *null* em uma tabela específica deve ser mínimo ou inexistente. As tabelas não devem possuir campos nulos. Caso exista algum campo que permita valores do tipo *null*, uma justificativa deve ser incluída no dicionário de dados e a Geti irá avaliar a proposição, podendo ou não aceitá-la, após análise.

5.5 Definição de tamanho de campos do tipo texto

- Para colunas que possuam um tamanho previsível e/ou definido (finito), o tipo de dados usado na definição do campo deve ser *char*. Isto se aplica para campos que armazenem siglas, status ou tipo de registros. Exemplos: telefone, CPF, etc.
- Todos os campos do tipo varchar devem ter seu tamanho justificado. Estas informações são de extrema importância para a equipe de Administração de Banco de Dados para implementação física dos modelos de dados.
- A contagem de caracteres envolve apenas caracteres alfanuméricos.

5.6 Tabelas existentes em outros sistemas (schemas)

- Replicação de tabelas e/ou dados existentes em outros sistemas devem ser evitadas. Para todos os casos deve-se apresentar uma justificativa e exposição da necessidade de replicação. A equipe da Geti irá avaliar a melhor e mais adequada alternativa para este tipo de ação.
- O acesso às tabelas de outros sistemas, inclusas no modelo de dados (explicitamente), será realizado através de *views* ou concessão de privilégio de consulta às tabelas necessárias, sendo que para fins de integridade referencial (tabelas corporativas ou residentes em outros ambientes), a replicação deve ser implementada em ambiente corporativo através de *Materialized Views* com apenas permissão de consulta, cujo conteúdo (uma ou mais tabelas, colunas e filtros), frequência (tempo de refresh), forma (completa ou parcial) e meios (softwares) devem ser claramente definidos.

5.7 Padrão de nomenclatura para dados de endereço definido pelos Correios

- Para tabelas de endereço, as colunas que irão formatar os dados devem seguir o padrão de nomenclatura e formato definido pelos Correios, conforme remessa de correspondências postais utilizando os parâmetros abaixo:

Atributos	Tipos de Dados		Descrição
	SQLServer	MySQL	
Localidade	<i>varchar (72)</i>	<i>varchar(72)</i>	Localidade pode adotar os seguintes valores: cidade, município, região administrativa ou Distrito Federal.
Tipo do	<i>int(9)</i>	<i>int(9)</i>	É o tipo do Logradouro, ex.: RUA, AVENIDA, PRAÇA. A tabela de domínio pode ser

Padrões e Normas para Administração de Dados

Logradouro			encontrada na tabela CORPORATIVO.TB_TIPO_LOGRADOURO.
Logradouro	<i>varchar(100)</i>	<i>varchar(100)</i>	O logradouro é o endereço propriamente dito, contendo rua, número, entre outras informações de identificação. Através do CEP é possível identificar o logradouro.
Complemento Logradouro	<i>varchar(50)</i>	<i>varchar(50)</i>	Complementos dos dados de logradouro, que identificam os dados específicos do endereço.
Bairro	<i>varchar (72)</i>	<i>varchar(72)</i>	Bairro é a subdivisão da localidade.
Cidade	<i>int(9)</i>	<i>int(9)</i>	É a cidade do endereço normalizada. A tabela padrão com os códigos das cidades do país encontra-se em CORPORATIVO.TB_CIDADE.
UF	<i>char(2)</i>	<i>char(2)</i>	Unidade da Federação.
CEP	<i>char(8)</i>	<i>char(8)</i>	Código de endereçamento postal.

Tabela 6 Tabela de nomenclatura para dados de endereço definido pelos Correios.

5.8 Armazenamento de arquivos

- Nenhum arquivo será armazenado no banco de dados. Para estes casos, uma pasta no servidor de arquivos deve ser criada para armazenamento dos arquivos. O banco de dados deve apenas armazenar os endereços (caminho) dos arquivos. Exemplos: imagens, arquivos texto, etc.
- O dicionário de dados deve explicitar o uso de arquivos externos, e ainda o tipo (extensão) de arquivo utilizado pela aplicação.

5.9 Tabelas temporárias

- Tabelas temporárias tem vida útil. O tempo máximo de existência de uma tabela temporária é o período de 60 dias (dois meses). Após este período, a tabela será removida.
- No dicionário de dados deve ser informado o período de utilização da tabela. A data de início e de fim da existência da tabela deve ser registrada no formato DD/MM/YYYY – (dia/mês/ano).

5.10 Padrão de cabeçalho para *packages*, *body packages*, *stored procedures*, *functions* e *triggers*

- *Packages*, *body packages*, *stored procedures*, *functions* e *triggers* devem possuir um cabeçalho. O padrão segue abaixo:

```
/*
* SCHEMA.NOME_DO_OBJETO.SQL
*
* Copyright:
*
* Este programa de computador é de propriedade intelectual do Instituto Brasília Ambiental.
*
* =====
* Autor: Fulano de Tal.      Empresa: Empresa Beta      Contato: XXXX-XXXX
* =====
*
* Objetivo: Importar dados do IbramWeb.
* =====
*
* Histórico de Mudanças:
* =====
* Data      Autor      Descrição da Alteração
* =====
```

```
* 01/JAN/2014    Fulano de Tal    Criação do objeto.
* 01/FEV/2014    Sicrano de Tal    Alteração nos parâmetros de entrada.
* =====
*
*/
```

5.11 Padrão de cabeçalho *views* e *materialized views*

- *Views* e *Materialized Views* devem possuir o seguinte cabeçalho:

```
/* SCHEMA.NOME_DO_OBJETO.SQL
*
* Copyright:
*
* Este programa de computador é de propriedade intelectual do Instituto Brasília Ambiental.
*
* =====
* Autor: Fulano de Tal.                Empresa: Empresa Beta                Contato: XXXX-XXXX
* =====
*
* Objetivo: Selecciona os servidores que ficaram afastados em determinado ano por motivo de doença.
* =====
* Schema(s) envolvido(s): IBRAMWEB_ADM
* =====
* Tabelas envolvidas: TB_SERVIDOR, TB_LICENCA ,TB_TIPOLICENCA
  Colunas: TB.SERVIDOR.NO_NOME, TB_LICENCA.NO_LICENCA, TB_TIPOLICENCA.TP_TIPO
  Filtros: TB.LICENCA.DATA= '2012' AND TB_TIPOLICENCA.TP_TIPO = '2'
* =====
* Atualização: ( ) Sob Demanda  (X) Periodicamente
  Se periodicamente, frequência de atualização: até o dia 05 de cada mês.
* =====
* Forma de atualização (para as materialized views): ( ) Parcial  (X) Completa
* =====
```

* Histórico de Mudanças:

=====		
* Data	Autor	Descrição da Alteração

* 01/JAN/2014	Fulano de Tal	Criação do objeto.
* 01/FEV/2014	Sicrano de Tal	Inclusão do campo: TB_LICENCA.DATA
=====		
* /		

5.12 Indexes

- A criação de índices deve estar diretamente ligada às consultas efetuadas pelo sistema. Índices obsoletos serão eliminados.
- No dicionário de dados deve ser informado o período de utilização da tabela. A data de início e de fim da existência da tabela deve ser registrada no formato DD/MM/YYYY – (dia/mês/ano).
- O número máximo de índices permitido por tabela é de seis, contando com o índice da chave primária. Caso mais índices sejam necessários, deve justificar necessidade no dicionário de dados e a Geti irá avaliar a proposição.

6. USUÁRIOS

- Os tipos de usuários disponíveis seguem abaixo:
 - Usuário dono do *schema* - sistema;
 - Usuário de aplicação;
 - Usuário de consulta.

- O nome do usuário dono do schema não deve ser separado por nenhum operador.
- Nenhum usuário terá permissão de DBA no banco de dados.
- Abaixo segue tabela de privilégios de usuários:

TIPO DE USUÁRIO	PREFIXO	SUFIXO	EXEMPLO	PERMISSÕES
Sistema	<nome do schema>	-	RH	Operações DDL, DML e DTL nos objetos pertencentes ao <i>schema</i> .
Aplicação	<nome do schema>	_INTER	RH_TREINAMENTO_INTER	<u>Tabelas</u> : insert, select, update e delete. <u>Views e sequences</u> : select. <u>Procedures/functions</u> : execute.
Consulta	<nome do schema>	_CONS	RH_CONS	<u>Tabelas</u> : select. <u>Views</u> : select.

Tabela 7 Tabela de apresentação dos tipos de usuários existentes e a permissão de cada um deles no banco de dados.

7 DICIONÁRIO DE DADOS

A representação gráfica e denominação dos elementos que compõem um modelo de dados não são suficientes para traduzir todos os conceitos do negócio. Além disso, a manutenabilidade dos objetos de banco de dados também é comprometida pela ausência de dicionarização dos objetos.

O dicionário de dados tem por objetivo registrar e padronizar conceitos e aumentar a fidelidade ao negócio na interpretação dos conceitos. Faz-se possível, dessa forma, diminuir a complexidade dos modelos de dados e aumentar a uniformidade dos objetos a serem construídos e mantidos no banco de dados.

7.1 Regras Gerais

- O dicionário de dados deve seguir aos seguintes critérios básicos:
 - Clareza, objetividade, detalhamento e precisão;
 - Justificativa de escolha de tipo de dados do tipo texto;
 - Especificação de restrições de domínio;
 - Citação de exemplos ilustrativos - sempre que possível para facilitar entendimento;
 - Respeito às regras vigentes no Instituto Brasília Ambiental (Normas);
 - Respeito às normas da língua portuguesa;
 - Especificação das regras nas quais os elementos conceituados se encaixam;
 - Especificação de exceções às regras, quando aplicáveis;
 - Correlação entre conceitos descritos;
 - Acréscimo de informações que contribuam para compreensão.
- Os seguintes fatores devem ser evitados na descrição dos objetos do modelo de dados:
 - Ambiguidade;
 - Sintetização exagerada;
 - Permissão de inferência;

- Distorção do sentido dos termos;
- Assumir que um conceito é trivial e não precisa ser detalhado;
- Uso de siglas e abreviações na descrição dos elementos.
- O conjunto mínimo de informações que um dicionário de dados deve conter, considerando os itens descritos anteriormente, segue abaixo:
 - Descrição dos elementos que compõem o modelo de dados: todos os elementos devem ter seu significado descrito, contendo informações sobre as quais os dados serão armazenados na mesma;
 - Tabelas: considerando que este é o objeto mais significativo do banco de dados, deve-se documentá-lo com bastante atenção. Sua descrição deve conter o conceito da entidade por ela representada, incluindo a importância e a relação com outras tabelas e objetos do banco de dados. Abaixo estão dispostas algumas perguntas que facilitam a descrição de uma tabela:
 - ✓ O que é a entidade?
 - ✓ O que faz a entidade?
 - ✓ Para que serve?
 - ✓ O que engloba a categoria de elementos que compõem a entidade?
 - ✓ O que está excluído dessa categoria?
 - ✓ Quando alguém ou algo passa a ser, ou deixa de ser, um elemento dessa entidade?
 - ✓ Sua permanência nesta categoria é imutável?

Como exemplo, vamos considerar uma tabela que contenha informações de esportistas, sendo seu nome TB_ESPORTISTA. De acordo com as regras aqui definidas uma descrição incorreta seria: “contém os dados relativos aos esportistas da empresa que participam de um torneio”.

Porque esta descrição não estaria correta? O que se deseja conhecer não é a finalidade do objeto, mas sim:

- ✓ O que é um esportista?
- ✓ O que faz um esportista?
- ✓ Para que serve um esportista?
- ✓ O que engloba a categoria dos esportistas no nosso caso?
- ✓ O que exclui a categoria dos esportistas no nosso caso?
- ✓ Quando alguém passa a ser, ou deixa de ser um esportista?

Então, uma descrição correta para esta tabela seria: “é todo funcionário da empresa, que para fins desportivos, representa-a em uma ou mais modalidades individuais ou coletivas. São considerados esportistas todos aqueles que participaram oficialmente do grêmio desportivo da empresa, por adesão voluntária ou indicação de terceiros. Não é permitida a participação de pessoas externas à empresa como esportistas”.

- Sequences: deve conter a descrição, regra de incremento e o nome da coluna alimentada pela *sequence*;
- Restrições de domínio/Constraints: devem estar descritas, relacionando os atributos que compõem estes objetos e o conjunto de elementos que formam este domínio – quando aplicável;

- Descrição das colunas: deve ser detalhada. Os dados que os campos irão armazenar devem ser descritos. Correlações entre atributos que envolvam regras de negócios devem ser relatados. A opção por tipo de dados específicos deve ser documentada e em caso de dados do tipo texto, o tamanho do campo deve ser justificado. A descrição da coluna, em conjunto com a descrição de sua tabela, deve ser suficiente para a compreensão e interpretação da informação nela registrada;
- Relacionamentos/Constraints: os relacionamentos devem estar documentados quanto a sua cardinalidade e obrigatoriedade, explicando questões sensíveis à regra de negócio definida;
Ex.: Suponha a FK_STATUS_RESERVALEGAL
Uma descrição apropriada para a *constraint* FK_STATUS_RESERVALEGAL estabelece a seguinte relação entre as tabelas TB_RESERVALEGAL e TB_STATUS: uma reserva legal **necessariamente** possui um status (ATIVA, EXCLUIDA, TRANSFERIDA, MUDOU NOME) e um status **poderá estar ou não** relacionado a uma ou a várias reservas legais.
- Views/Materialized Views/Package Body/Packages: devem estar descritas, principalmente no que diz respeito a sua usabilidade.
No caso de *Materialized Views* deverá ser informado:
 - Se ela será atualizada periodicamente ou sob demanda;
 - Caso seja sob demanda, deverá ser informado quem será o responsável pelo “refresh” da *view* e também se essa atualização será incremental ou por completo. *Materialized Views* sob demanda só serão aceitas quando tecnicamente justificadas;

- Caso seja periodicamente, informar a data limite para sua atualização. Ex: Esta *Materialized View* deverá ser atualizada até o dia 05 de cada mês.

As *Materialized Views* que fazem referência à(s) tabelas de outro(s) *schema(s)* que não o do usuário, serão implementadas em um *schema* corporativo.

As *Materialized Views* somente serão utilizadas para fins de consulta, não sendo permitido ao usuário inserir, deletar ou modificar dados destas.

As *Materialized Views* deverão ser desenvolvidas seguindo o mesmo padrão de nomenclatura de tabelas físicas.

- Triggers/Functions: devem estar descritas, suas ações devem ser explicadas e também a sua correlação com a implementação de regras de negócios devem ser descritas (em casos de implementação de restrições de domínio);
- Store Procedures/Package Body/ Packages: os procedimentos armazenados devem estar descritos quanto a sua funcionalidade e em caso de implementação de restrições de domínio, as regras devem ser especificadas.

7.2 Modelo de Dicionário de Dados

Um modelo de dicionário de dados, considerando todos os itens descritos no tópico de normas para administração de dados, segue abaixo:

- **Tabela TB_SERVIDOR**

▪ Visão Geral da tabela TB_SERVIDOR

Owner	RH
Name	TB_SERVIDOR
Comment	Tabela que contém o cadastro de todos os servidores efetivos, comissionados e estagiários que desenvolvem atividades na sede e nos parques do Ibram. Os agentes públicos que forem exonerados, imediatamente deixam de pertencer à entidade TB_SERVIDOR e passam a fazer parte da entidade TH_SERVIDOR.

▪ Colunas da Tabela TB_SERVIDOR

Name	Data Type	Comment	Mandatory	Default	Primary	Foreign Key
CD_SERVIDOR	INTEGER	Código sequencial da tabela, chave primária alimentada pela <i>sequence</i> sq_servidor	X		X	
CD_MATRICULA	INTEGER	Matricula do servidor	X			
CD_SETOR	INTEGER	Código do setor em que o servidor está lotado				
DS_ENDERECO	VARCHAR(400)	Endereço residencial do servidor	X			

Padrões e Normas para Administração de Dados

NU_NUMERO	INTEGER	Número do endereço residencial do servidor				
TP_ESTADO CIVIL	CHAR	Estado Civil do servidor				
DT_POSSE	DATETIME	Data da posse do servidor				
...
...

▪ Chaves da tabela TB_SERVIDOR

Constraint Name	Primary	Comment
PK_SERVIDOR	X	

▪ Chave PK_SERVIDOR da tabela TB_SERVIDOR

Colunas da Chave PK_SERVIDOR

Name	Primary	Comment
CD_SERVIDOR	X	Código sequencial da tabela, chave primária alimentada pela <i>sequence</i> sq_servidor

- Chaves Estrangeiras

Foreign Key Constraint Name	Parent Table	Child Table	Foreign Key Columns	Comment
FK_SETOR_SERVIDOR	TB_SETOR	TB_SERVIDOR	CD_SETOR	Chave estrangeira que referencia a tabela TB_SETOR, o qual indica o setor em que o servidor está lotado.

Tabela 8 Tabelas contendo pequena parte do modelo de dicionário de dados apresentado à GETI.

- Restrições de domínio que envolvam *TRIGGERS* ou *PROCEDURES* deverão conter especificação das regras de integridade:

- Descrição da funcionalidade da trigger/procedure;
- Detalhamento das regras de integridade envolvidas.

8. AVALIAÇÃO DO MODELO DE DADOS

- Todos os modelos de dados de novos sistemas do Instituto Brasília Ambiental devem ser submetidos à apreciação da Geti a qual irá avaliar o modelo antes da implementação do mesmo no banco de dados. O Modelo Entidade Relacionamento (MER) deve ser enviado no mínimo duas semanas antes da data desejada para a criação deste nos ambientes de homologação e produção.

- Todos os modelos de dados disponíveis para avaliação devem ser acompanhados do **DICIONÁRIO DE DADOS ATUALIZADO** e do modelo físico necessariamente.

- A avaliação do modelo de dados será efetuado mediante a apresentação dos seguintes documentos:

- Documento de visão;
 - Documento de arquitetura, caso exista. Este documento tem por objetivo informar a integração do modelo que está sendo proposto com outros *schemas*, e até mesmo com outros bancos de dados, bem como outras informações consideradas relevantes que não foram especificadas nos demais documentos;
 - Documento de especificação suplementar, caso exista. Este descreve alterações dos documentos anteriores que foram realizados durante a execução do projeto, alterando o escopo do mesmo;
 - Dicionário de dados atualizado;
 - Modelo físico;
 - Scripts de criação do modelo físico;
-
- Em caso de alteração de um modelo de dados já implementado, deverá ser apresentado os seguintes documentos:
 - Dicionário de dados atualizado;
 - Modelo físico;
 - Scripts que geraram a alteração;
 - Descrição das modificações realizadas;
 - Após a avaliação do modelo de dados, a Geti irá gerar um relatório de avaliação do modelo referido. O relatório irá relatar todos os itens avaliados e o resultado da avaliação.