

MINISTÉRIO DA SAÚDE
SECRETARIA DE INFORMAÇÃO E SAÚDE DIGITAL
DEPARTAMENTO DE INFORMAÇÃO E INFORMÁTICA DO SUS
COORDENAÇÃO-GERAL DE INFRAESTRUTURA E SEGURANÇA DA INFORMAÇÃO

**ORIENTAÇÕES PARA CONSTRUÇÃO DE
MODELO DE DADOS
COM QUALIDADE**



Apresentação

A Administração de Dados em uma instituição é uma área que deve ter uma ampla atuação e andar em conjunto com a área de Desenvolvimento de Sistemas.

A Administração de Dados tem como objetivo principal fazer a gestão do modelo de dados corporativo da instituição, promovendo sua conceituação, segurança, integridade, compartilhamento e qualidade.

Para isso a administração de Dados deve atuar de forma à:

- Conhecer o negócio da empresa de forma, a saber, as áreas que utilizam os dados, considerando o nível de acesso que cada um terá do dado;
- Modelar os dados corporativamente;
- Respeitar as múltiplas visões e utilizações de um mesmo dado;
- Disponibilizar os dados de forma organizada;
- Compartilhar os dados evitando redundância;
- Servir como base para a construção de sistemas de informações flexíveis, integrados e eficientes.

O objetivo deste documento é apresentar diretrizes para a construção de um modelo de dados com qualidade, e que devem obrigatoriamente ser seguidas, visando assim a melhoria da qualidade da informação do Ministério da Saúde.



Figura 1 - Administrador de Dados

Histórico de Revisão

Data	Demanda	Autor	Descrição	Versão
15/01/2016	N/A	Eliete Colucci Sousa	Criação da 1ª versão da Metodologia de Administração de Dados (MAD).	1.0
27/11/2015	N/A	Eliete Colucci Sousa	Revisão de itens de acordo com observações sobre o uso da MAD.	2.1
06/01/2016	N/A	Eliete Colucci Sousa	Revisão de itens de acordo com observações sobre o uso da MAD	2.2
20/07/2016	N/A	Eliete Colucci Sousa	Alterações nos seguintes itens da Cartilha do formato da MAD de forma a deixar de acordo com o padrão de publicação no Portal do DATASUS.	2.2.1
21/11/2016	N/A	Eliete Colucci Sousa	Alterações na formatação do texto diagramado, de forma a facilitar a sua revisão. Além disso, os seguintes itens foram incluídos / revisados: 1) Inclusão de classes de dados; 2) Alteração de procedimento de criação de AUDITORIA; 3) Alteração de procedimento de criação de concessão de permissões de obkets em banco de dados; 4) Alteração de procedimento de criação de texto do uso de palavras em outro idioma; 5) Inclusão de regra para datatype padrão para campos do tipo caracter com tamanho máximo delimitado e numérico com precisão; 6) Incluir resposta padrão para campos BLOB.	2.2.2
31/07/2018	N/A	Eliete Colucci Sousa	Pequenas alterações para tornar mais claro o texto de alguns itens. Inclusão do item “Privilégios Padrão”, que define quais são os privilégios concedidos, por padrão, na criação de objetos.	2.3
20/01/2023		Gislaine Mabel da N. Trindade	Revisão Técnica do documento e atualização do documento para contemplar o cenário atual do DATASUS	2.4

Sumário

1	Introdução	8
2	Definição de Padrões.....	9
2.1	Padrões de Nomes de Objetos de Banco de Dados	9
3	Qualidade.....	10
3.1	Diretrizes	10
3.2	Regras Técnicas.....	12
3.2.1	Primary Key	12
3.2.2	Forcing Key (FK).....	13
3.2.3	Constraint para Coluna NOT NULL	13
3.2.4	Constraint Default de Coluna	13
3.2.5	Lista de Valores	14
3.2.6	Date x Timestamp	17
3.2.7	Tabelas de Auditoria.....	18
3.2.8	Tabelas de Histórico	23
3.2.9	Tabelas Associativas.....	23
3.2.10	Exclusão Lógica.....	25
3.2.11	Index.....	25
3.2.12	Triggers	26
3.2.13	Views.....	26

3.2.14 View Materializada	27
3.2.15 Campo BLOB.....	27
3.3 Privilégios Padrão.....	28
3.4 Normas para Modelagem Relacional.....	29
3.4.1 Tabelas.....	29
3.4.2 Colunas.....	30
3.4.3 Outros Objetos.....	30
3.4.4 Qualidade Negocial.....	31
4 Dicionário de Dados.....	32
4.1 Diretrizes para Documentação do Modelo de Dados	32
4.2 Descrições Específicas.....	34
4.3 Armazenamento de Dicionário de Dados.....	35
4.3.1 Modelo de Dados.....	36
4.3.2 Tabela	37
4.3.3 Coluna.....	38
5 Outros Tipos de Demandas	39
5.1 Concessão de Permissões e Auditoria.....	39
5.1.1 Concessão para tabelas de próprio esquema	40
5.1.2 Concessão em tabelas de esquema diferente.....	41
5.2 Solicitação de DROP / TRUNCATE / DELETE.....	42
6 Conclusão.....	43
7 ANEXO I.....	45
7.1 Indicadores de Qualidade de Modelo de Dados.....	45

Figuras

Figura 1 - Administrador de Dados.....	2
Figura 2 - Fique Ligado – Padrões e Diretrizes.....	8
Figura 3 - Fique Ligado - Padrões de Nomenclatura de BD.....	9
Figura 4 - Fique Lidado Qualidade do Modelo de Dados.....	10
Figura 5- AD Chamando a atenção para as dicas do uso de objetos de BD.....	12
Figura 6 – Colunas com domínio – CK ou Tabela?	16
Figura 7 - Date x Timestamp.....	17
Figura 8 - Fique Ligado – Execução de procedimentos na aplicação para trilha de auditoria.....	18
Figura 9 - Orientações da GAAD sobre auditoria	19
Figura 10 - Nunca deixe mais de uma trigger de auditoria habilitada	19
Figura 11 – Quando gerar AUDOTORIA?	20
Figura 12 - Atenção todos os AD's - Geração e de Auditoria	21
Figura 13 - AD chamando a atenção para as normas negociais.....	29
Figura 14 – Modelo de Dados com Qualidade	31
Figura 15 - Dicionário de Dados não bem compreendido.....	32
Figura 16 - Dicionário de Dados com informação relevante.....	33
Figura 17 - Fique Ligado – Gere um bom dicionário de dados	34
Figura 18 - Informações gerais do modelo de dados	36
Figura 19 - Informacionais adicionais do modelo de dados	36
Figura 20 - Informações gerais de tabela	37
Figura 21 - Informações adicionais de tabela	37
Figura 22 - Informações gerais de coluna	38
Figura 23 - Dúvida sobre os novos formulários no PowerDesigner.....	38
Figura 24 - Concessão de Acesso.....	39
Figura 25 - - DROP e TRUNCATE de tabelas.....	42
Figura 26 – Em caso de dúvida, fale com a GAAD	43
Figura 27 – Sou um AD da GAAD	44

Tabelas

Tabela 2 - Colunas de Controle de Tabela de Auditoria	20
Tabela 3 – Concessão de acesso em tabelas do próprio esquema e Auditoria.....	41
Tabela 4 – Regras para concessão de permissões em Tabelas de esquemas diferentes e Auditoria	42

1 Introdução

A Administração de Dados possui vários processos, mas o foco deste documento é a modelagem de dados.

Sendo assim, nele são tratados os padrões e introduzidas as diretrizes para uma modelagem de dados de qualidade, servindo de orientação para os técnicos da os usuários.

Como a qualidade de um modelo é imprescindível para que o Ministério da Saúde possa prestar seus serviços ao cidadão, estas diretrizes devem obrigatoriamente ser seguidas.

Fique ligado

Os padrões e diretrizes aqui indicadas promovem modelos de dados de qualidade.

Figura 2 - Fique Ligado – Padrões e Diretrizes



2 Definição de Padrões

2.1 Padrões de Nomes de Objetos de Banco de Dados

A nomenclatura para objetos de banco de dados está definida na norma constante no link <https://ced.saude.gov.br/index.php?r=site%2Fmad> – Nomenclatura.

O padrão de nomes deve seguir a norma Padrão ISO/IEC 11179-5, que é um padrão internacional, assim como o Catálogo de Padrões do Governo (e-ping – Padrões de Interoperabilidade do Governo Brasileiro).

Fique Ligado

É de fundamental importância, que o modelo de dados corporativo da instituição esteja de acordo com o padrão de nomenclatura definido para objetivos de banco de dados, pois isso facilita o entendimento de todos que manipulam e necessitam de modelo de dados para o desenvolvimento de seus trabalhos. Isso é um dos itens de garantia de qualidade do modelo!



Figura 3 - Fique Ligado - Padrões de Nomenclatura de BD

1 Introdução

3.1 Diretrizes

O assunto qualidade de modelo de dados está diretamente ligado com garantia da qualidade da informação, pois um bom modelo de dados é peça fundamental para a garantia de dados confiáveis e seguros.

Um modelo de dados representa a definição, caracterização e relacionamento dos dados em um determinado ambiente, e por isso, para que ele possua qualidade deve produzir dados corretos, precisos, consistentes, completos, integrados, aderentes às regras de negócio e aos domínios estabelecidos.

Além disso, um modelo de dados deve sempre que possível levar em consideração a possibilidade de futuras implementações, considerando aspectos de aquisição, arquivamento e padronização dos dados, bem como no armazenamento, recuperação e reutilização das informações.

Este item tem por objetivo descrever procedimentos que devem ser adotados na elaboração / manutenção de modelos de dados, visando nortear a Administração de Dados na construção de modelos de dados com qualidade.

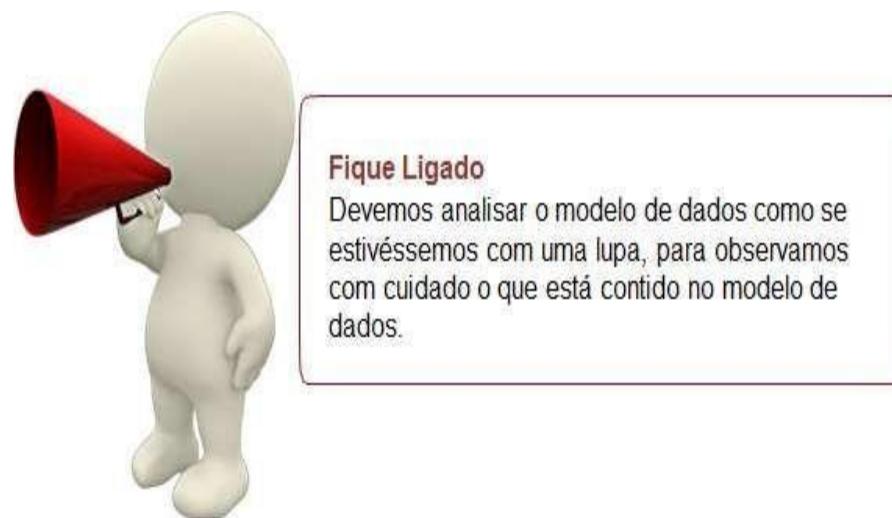


Figura 4 - Fique Ligado Qualidade do Modelo de Dados

Portanto, na elaboração e manutenção de um modelo de dados temos que trabalhar de forma a garantirmos a qualidade do modelo. Para isso, as regras técnicas, normas para modelagem de dados e documentação descritas neste documento devem ser seguidas, pois são boas práticas de modelagem de dados. Veja no Anexo I os indicadores de qualidade de um modelo de dados.

Devemos todos trabalharmos juntos para a melhoria da qualidade da informação.

Assim estaremos construindo para a melhoria dos serviços prestados pelo Ministério da Saúde ao cidadão.



Figura 5 - Melhoria dos Serviços Prestados pelo Ministério da Saúde

3.2 Regras Técnicas

Com relação às regras técnicas, são descritas a seguir orientações quanto ao bom uso de objetos de banco de dados.

Preste atenção!

As diretrizes a seguir dão boas dicas!

3.2.1 Primary Key

Uma primary key permite a identificação única do registro, podendo ser um sequencial (controlada por uma sequência do banco de dados), que não tem significado negocial, ou uma chave negocial, simples ou composta. No caso da PK ser um sequencial, é interessante que se defina uma UK como chave negocial da tabela, por exemplo, em uma tabela de Pessoas Físicas a chave negocial seria o nº do CPF.

Conforme o documento de padrão de nomenclatura (tem 10.2 – Tipos de Tabelas) está definido quais tipos de tabelas são classificadas como negociais e quais não são. Toda tabela definida como negocial é obrigatória a definição de uma PK, para as que não são classificadas como negocial, a PK é opcional.



Figura 5- AD Chamando a atenção para as dicas do uso de objetos de BD

3.2.2 Foreign Key (FK)

Uma ou mais colunas que são geradas por um relacionamento com a tabela detentora destas colunas e mais outra tabela, que vai determinar os valores possíveis para estas colunas. Em uma FK podemos ter uma coluna ou mais de uma (chave composta). No caso de chave composta é interessante que as colunas tenham preenchimento obrigatório, pois o SGBD somente realiza a verificação de integridade quando todas as colunas estiverem preenchidas.

Conforme Anexo I, o não atendimento deste item contraria os indicadores de qualidade acessibilidade e credibilidade.

3.2.3 Constraint para Coluna NOT NULL

Para essas constraints não há necessidade de se definir um nome, pois estas são criadas automaticamente pelo banco de dados.

Conforme Anexo I, o não atendimento deste item contraria o indicador de qualidade documentação.

3.2.4 Constraint Default de Coluna

Para colunas com preenchimento obrigatório, o uso do valor default é obrigatório somente nos casos de criação de coluna com preenchimento obrigatório em tabela já existente.

Para essas constraints não há necessidade de se definir um nome, pois estas são criadas automaticamente pelo banco de dados.

Conforme Anexo I, o não atendimento deste item contraria os indicadores de qualidade acessibilidade e documentação.

3.2.5 Lista de Valores

Esta situação se refere a colunas que possuem um **domínio definido**, por exemplo, estado civil, tipo sanguíneo, tipo de pessoa, situação de contrato, etc.

É importante ficar claro que nestas situações é imperativo que:

- Haja a implementação de controles como valores válidos constantes na lista, que pode ser por check constraint ou tabela de domínio, conforme regras definidas neste item;
- Todos os valores devem ter o significado de cada um definido e documentado, pois isso diminui os problemas com informações sem credibilidade por falta de entendimento dos dados armazenados na base de dados.

Então para a implementação de lista de valores devem ser seguidas obrigatoriamente as 5 regras descritas a seguir.

i. Devem ter preenchimento obrigatório.

O preenchimento obrigatório é importante, pois NULL não é um valor pré-definido e sim ausência de valor.

Acrescentando-se que registros com NULL prejudicam a performance de consulta, pois estes não são considerados em index.

Além disso, na obtenção de indicadores que envolvam essas colunas de domínio com valores NULL, esses registros também não são considerados.

Nos casos onde é necessário NULL, deve-se definir como valor no domínio, a coluna preenchida totalmente com o dígito 9, seja para datatype numérico ou de caractere. Por exemplo:

- Coluna numérica de tamanho 1, o valor correspondente deve ser 9;
- Coluna numérica de tamanho 2, o valor correspondente deve ser 99;
- Coluna numérica de tamanho 3, o valor correspondente deve ser 999;
- Coluna caractere de tamanho 1, o valor correspondente deve ser ‘9’;
- Coluna caractere de tamanho 2, o valor correspondente deve ser ‘99’;
- Coluna caractere de tamanho 3, o valor correspondente deve ser ‘999’.

A descrição do valor deve ser definida da forma mais adequada ao negócio que está atendendo.

ii. Utilizar CK com a definição dos valores do domínio:

- Domínio possui no máximo dois valores e mais o valor para NULL conforme descrito acima;

Os valores contidos na lista devem indicar que são estáveis, isto é, que a lista não vai aumentar;

Para o caso da CK depender do conteúdo de uma outra coluna da tabela, deve ser criada uma CK da tabela;

No comentário do campo, onde é descrita a sua finalidade, é obrigatório que contenha cada valor e a descrição do significado de cada um.

Na situação aqui indicada é obrigatório o uso de uma CK.

iii. Domínio Sim ou Não

Para colunas cujo conteúdo deve expressar Sim ou Não, os valores do domínio devem ser S ou N, respectivamente.

iv. Utilizar uma tabela de domínio, quando o domínio não se encaixa nas situações relacionadas com indicação de uso de CK.

v. Como incluir dados em tabela de domínio?

- Para tabelas do DBGERAL, a responsabilidade é da coordenação de banco de dados;
- Para tabelas de negócio temos as seguintes situações:

- Manutenção de dados em tabela que não tem impacto no negócio e pode ter muitas alterações: o indicado é que o sistema tenha telas para manutenção das tabelas;
- Manutenção de dados em tabela que não tem impacto no negócio e que o previsto é que praticamente não haja alterações em seu conteúdo: o indicado é que o sistema não tenha telas para manutenção das tabelas (atualização por script);
- Manutenção de dados em tabela que tem impacto no negócio: o indicado é que o sistema não tenha telas para manutenção das tabelas, até porque em caso de alterações no conteúdo, a equipe responsável precisa analisar o impacto no sistema.

Conforme Anexo I, o não atendimento deste item contraria os indicadores de qualidade acessibilidade, credibilidade e flexibilidade.



Regras para colunas com **DOMÍNIO DEFINIDO**:

- Preenchimento Obrigatório
- Para o controle dos valores do domínio, devemos utilizar CK ou tabela?
 - Check Constraint (CK) se possuir no máximo 2 valores e mais um para quando o NULL representar o NULL, quando necessário (normalmente colunas com prefixo ST ou TP);
 - Quando possuir mais valores, deve-se ter uma tabela para armazenar os valores de domínio (normalmente coluna com prefixo CO recebendo FK da tabela de domínio).
 - Se uma coluna for criada como CK (3 valores no máximo), e devido essa lista não ser estável e os valores do domínio começarem a crescer, o controle deverá ser alterado para tabela com tais valores, exportando FK → por isso, somente utilize CK quando tiver certeza que a lista de valores é estável.

Figura 6 – Colunas com domínio – CK ou Tabela?

3.2.6 Date x Timestamp

A utilização desses datatypes as vezes causa um pouco de confusão. Quando devo usar cada um?



O que é melhor Date ou Timestamp?

Utilize o DATE quando é necessário apenas a DATA ou DATA e HORA com precisão de segundos.

Utilize o TIMESTAMP quando é necessário DATA e HORA com precisão de milissegundos ou para utilização de timezone. O timezone é apropriado quando é necessário exibir informações de data e horários usando o fuso horário do sistema cliente.

Figura 7 - Date x Timestamp

3.2.7 Tabelas de Auditoria

Tabelas que permitem efetuar um **rastreamento de qualquer operação que ocorre nos dados de uma tabela**. Para que uma tabela de auditoria seja eficaz, deve responder quem fez a operação, quando foi feita e o que foi feito.

Para a auditoria funcionar, devem ser inseridos na aplicação comandos para se obter o usuário final e o IP da máquina onde o usuário está logado:

```
-- Login/Identificador do usuário como parâmetro  
EXEC DBGERAL.PKG_INFO_LOG.PC_INFORMA_USUARIO  
(‘USERNAME’);  
  
-- IP do usuário como parâmetro  
EXEC DBGERAL.PKG_INFO_LOG.PC_INFORMA_IP(‘10.0.0.1’);
```

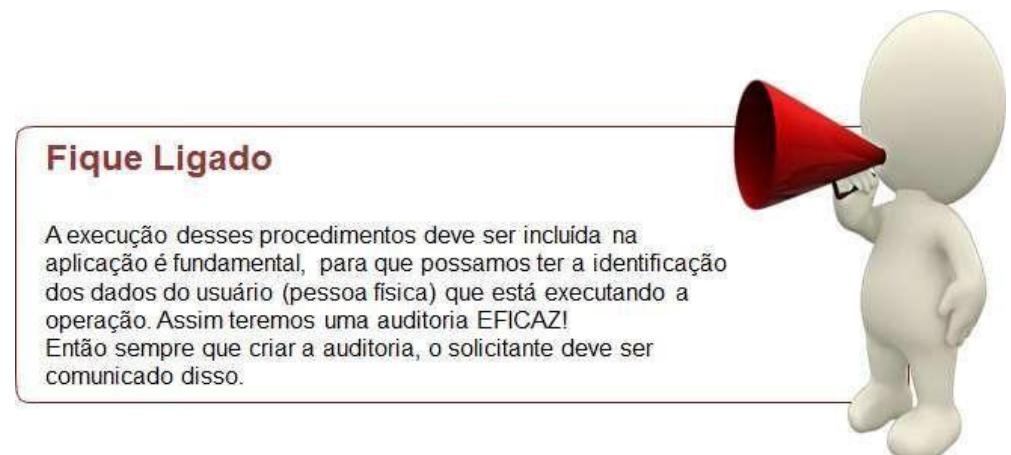


Figura 8 - Fique Ligado – Execução de procedimentos na aplicação para trilha de auditoria

Quanto aos privilégios de acesso de tabelas de AUDITORIA:

- A alimentação de tabela de auditoria pode ser feita por trigger ou pela aplicação. É sugerido a utilização de trigger, pois julga ser um método mais efetivo.
- Para garantir a confiabilidade e a rastreabilidade das informações, tabelas de auditoria não podem ter seu dado alterado. Por isso os privilégios de acesso a essas tabelas seguem as seguintes regras:
 - Alimentada por trigger: **grant de SELECT;**
 - Alimentada pelo aplicativo: **grant de SELECT e INSERT.**



Portanto preste muita atenção, isso é muito é muito importante!

Investigue antes. Nunca deixe mais de uma trigger de auditoria habilitada.

Mas então como proceder para sistemas legados em que a equipe decide por utilizar a nova estrutura:

- Se não há auditoria, auditoria será criada no padrão da metodologia MAD.
- Se há auditoria em outro formato, é sugerido a utilização do padrão da equipe, mas a decisão é da equipe responsável pelo aplicativo. Nunca se esquecer, que na criação da uma nova auditoria para uma tabela, a trigger da antiga deve ser desabilitada.
- Para projetos novos, deve ser adotado o padrão da MAD.

Figura 9 - Nunca deixe mais de uma trigger de auditoria habilitada

A estrutura da tabela de auditoria no padrão da MAD possui as mesmas colunas da tabela origem e mais as de controle, conforme definido na tabela 2.

Coluna	Tipo/Tamanho	Descrição
AU_DT_OPERACAO	TIMESTAMP(6)	Data da operação, armazenada no formato TIMESTAMP “DD/MM/YYYY HH24:MI:SS.FF”.
AU_TP_OPERACAO	VARCHAR2(1)	Indica a operação efetuada pelo usuário na origem. Podendo ser: (I: para inclusão de registro (comando insert)); (A: para alteração de registro (comando update)); (E: para exclusão de registro (comando delete)).
AU_DS_USUARIO_SESSAO	VARCHAR2(30)	Usuário de banco da sessão ORACLE.
AU_NU_IP_SESSAO	VARCHAR2(15)	Endereço IP da máquina que originou a sessão ORACLE.
AU_DS_USUARIO_REDE	VARCHAR2(40)	Usuário de rede logado na máquina.
AU_DS_HOST_REDE	VARCHAR2(40)	Nome da máquina na rede.

Coluna	Tipo/Tamanho	Descrição
AU_DS_USUARIO_APPLICACAO	VARCHAR2(100)	usuário final, informado pela aplicação. Informado e recuperado através da package DBGERAL.PKG_INFO_LOG.
AU_NU_IP_APPLICACAO	VARCHAR2(15)	Endereço IP do usuário final, informado pela aplicação. Informado e recuperado através da package DBGERAL.PKG_INFO_LOG.
AU_CO_TRANSACAO	VARCHAR2(20)	Código identificador da transação no banco de dados.
AU_DS_USUARIO_COMPLEMENTO	VARCHAR2(500)	Informação complementar do usuário final, informado pela aplicação. Informado e recuperado através da package DBGERAL.PKG_INFO_LOG.

Tabela 1 - Colunas de Controle de Tabela de Auditoria

Quando gerar AUDITORIA para uma tabela?



A decisão de gerar ou não AUDITORIA para uma tabela é do Gestor de Negócio e / ou Coordenador da Equipe de Sustentação responsável pelo aplicativo.

É importante ressaltar que uma **AUDITORIA cresce de forma bem mais rápida que a tabela original**, pois a cada operação de INSERT / UPDATE / DELETE é gerado um novo registro na AUDITORIA.

Sendo assim, ao se analisar a necessidade da AUDITORIA, os seguintes itens deverão ser considerados:

- Os dados da tabela são críticos para o negócio.
- Qual é o crescimento da tabela? Para isso, deve-se verificar o nº inicial de linhas, o % de crescimento e a periodicidade para essa %.
- Qual é a justificativa para a criação da AUDITORIA.

Figura 10 – Quando gerar AUDITORIA?

O questionamento sobre a necessidade de AUDITORIA ou não, deve ser feito no momento de sua criação no DESENVOLVIMENTO, seja a pedido da equipe ou por concessão de privilégios (item 5.1).

Para gerar e atualizar a estrutura de tabelas de auditoria devem ser observados os itens descritos a seguir.

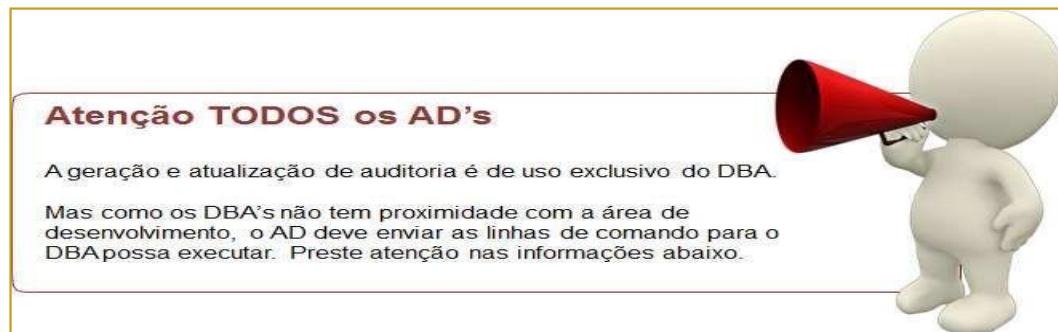


Figura 11 - Atenção todos os AD's - Geração e de Auditoria

1. Gerar Auditoria

O AD deve enviar a linha de comando a seguir para que o DBA a execute e gere a auditoria para os casos em que há necessidade de que uma tabela tenha auditoria.

```
EXEC DBDBA.PKG_AUDITORIA.PC_GERA_AUDITORIA('OWNER','TABLE_NAME');
```

Ao responder o atendimento da demanda, o solicitante deve ser comunicado que:

2. Atualizar Auditoria

Sempre que a estrutura de uma tabela for alterada, o AD deve verificar se essa tabela possui auditoria no padrão MAD e enviar a linha de comando a seguir para que o DBA a execute e atualize a auditoria.

Mas atenção nos casos de RENAME de coluna, se a package for executada diretamente, será incluída uma nova coluna com o novo nome. Então, nessas situações, o correto é executar o comando para renomear a coluna na tabela de auditoria: é bom que o AD passe esse comando para o DBA.

```
EXEC DBDBA.PKG_AUDITORIA.PC_ATUALIZA_AUDITORIA('OWNER','TABLE_NAME')
```

3. Gerar Script com Comandos de Atualização de Auditoria de Todas as Tabelas de um Esquema

```
DECLARE CURSOR TBS IS
```

```
SELECT OWNER, TABLE_NAME  
FROM DBA_TABLES  
WHERE OWNER = 'DBGERAL'  
AND (TABLE_NAME LIKE 'TB\_%' ESCAPE '\'  
OR TABLE_NAME LIKE 'RL\_%' ESCAPE '\'  
OR TABLE_NAME LIKE 'RT\_%' ESCAPE '\');  
BEGIN  
FOR R IN TBS  
LOOP  
DBGERAL.PKG_INFO_LOG.PC_ATUALIZA_ESTRUTURA(R.OWNER, R.TABLE_NAME);  
END LOOP;  
END;
```

4. Verificar de Auditoria Desatualizadas em um Esquema

```
SELECT TB.OWNER, TB.TABLE_NAME  
FROM DBA_TABLES AU, DBA_TABLES TB  
WHERE AU.OWNER = TB.OWNER AND AU.OWNER = 'OWNER'  
AND SUBSTR(AU.TABLE_NAME,3,30) = SUBSTR(TB.TABLE_NAME,3,30) AND AU.TABLE_NAME LIKE 'AU\_%' ESCAPE '\'  
AND (TB.TABLE_NAME LIKE 'TB\_%' ESCAPE '\'  
OR TB.TABLE_NAME LIKE 'RL\_%' ESCAPE '\'  
OR TB.TABLE_NAME LIKE 'RT\_%' ESCAPE '\')
```

AND DBGERAL.PKG_INFO_LOG.FC_VERIFICA_ESTRUTURA(TB.OWNER,TB.TABLE_NAME) > 0

5. Observações Importantes

- Tabela de auditoria somente deve possuir privilégio de acesso de SELECT. Para os casos em que a tabela é atualizada pelo aplicativo também deve ter o INSERT.
- Campos BLOB não são auditados, para o caso de tabelas que ainda possuem campos desse tipo.
- Na exclusão de colunas da tabela origem, estas não são excluídas da tabela de auditoria, pois esta visa manter o controle das informações existentes e as que já existiram na tabela de origem.
- Tabela de auditoria não deve possuir constraint, pois as restrições de integridade devem ser garantidas pela tabela origem.
- Quando uma tabela de auditoria nos padrões da MAD é criada, é importante comunicar a necessidade dos comandos que devem ser inseridos na aplicação.

Conforme Anexo I, para tabelas que devem ter auditoria, o não atendimento deste item contraria o indicador de qualidade credibilidade.

3.2.8 Tabelas de Histórico

Tabela utilizada para armazenar os dados históricos para atender de uma determinada funcionalidade necessária para o negócio.

Diferentemente de tabelas de auditoria, não há necessidade de que as tabelas com dados históricos possuam todas as colunas da tabela de origem.

A definição de quais colunas devem fazer parte da tabela de histórico deve constar nos requisitos no aplicativo, assim como a forma de manutenção de seus dados, que pode ser por trigger ou pelo próprio aplicativo.

Conforme Anexo I, o não atendimento deste item não contraria nenhum indicador de qualidade, mas é importante observar bem a diferença entre histórico e auditoria.

3.2.9 Tabelas Associativas

Uma tabela associativa representa uma entidade que não existe por si só e sua existência está condicionada à existência de duas ou mais entidades com relacionamento do tipo N:N. Além disso, o identificador negocial da tabela é formado exclusivamente pelas colunas que são geradas pela FK dessas tabelas relacionadas.

Vale ressaltar que mesmo que estas tabelas possuam atributos próprios devem ser tratadas como associativas (tabelas com prefixo RL_ ou RT_). Conforme Anexo I, o não atendimento deste item contraria o indicador de qualidade documentação.

3.2.10 Exclusão Lógica

Para o caso de tabelas com exclusão lógica, é importante verificar qual será o mecanismo de controle quanto ao uso de registros que estejam marcados como “Excluídos”, isto porque estes registros não podem ser utilizados. Sendo assim, ou a aplicação deve tratar isso ou então devem ser implementadas triggers nas tabelas que recebem FK dessas tabelas com exclusão lógica, de forma, que os registros marcados como excluídos não sejam mais utilizados.

Mas o que é importante de observar é que a exclusão lógica de uma tabela deve ser tratada por uma coluna, onde o nome obrigatoriamente é ST_REGISTRO_ATIVO com datatype VARCHAR2(1) e preenchimento obrigatório. Além disso, deve possuir uma check constraint com os valores ‘S’ ou ‘N’ e a descrição pode ser “Indica se o registro está ativo ou não(excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação.”.

Uma observação importante é que quando a tabela é criada já com essa coluna, não é obrigatório a definição de um valor DEFAULT, mas quando a coluna é adicionada em uma tabela já existente, deve-se definir o valor DEFAULT, pois assim a coluna pode ser criada como NOT NULL e o valor DEFAULT será preenchido para todos os registros existentes na tabela.

Conforme Anexo I, o não atendimento deste item contraria os indicadores de qualidade acessibilidade e documentação.

3.2.11 Index

A criação de index deve ser feita visando uma melhoria de performance em consultas ao banco de dados, sendo que isso é possível na criação de index adequados. Outra forma de termos performance melhor é ter uma melhoria de hábitos de forma a produzirmos os comandos SQL mais eficientes que implicarão em respostas mais rápidas para os usuários dos sistemas e uma melhoria no funcionamento dos bancos de dados, mas este será um assunto para ser tratado posteriormente.

A criação de index deve ser bem planejada, poiseles ajudam nas consultas, mas em inclusões, atualizações e deleções de registros, eles também são atualizados e com isso há uma perda de performance nessas operações.

Então a criação de index, deve ser feita de forma que seja bem utilizado na consulta e traga bom ganho de performance, que valha a pena o tempo perdido nas atualizações.

Sendo assim a criação de index é importante quando:

- Uma coluna é referenciada em um comando SQL de consulta na cláusula de restrição (filtro), através de:

- Uma igualdade: sg_uf = 'DF';
- Comparação com intervalo ilimitado: nu_vaga > 100;
- Comparação com intervalo limitado: nu_vaga between 100 AND 200;
- Para ORDER BY ou GROUP BY é bom que tenhamos um index considerando todas as colunas envolvidas e na ordem em que são especificadas nessas cláusulas (é bom que estas colunas tenham valores preenchidos, pois coluna com NULL não são consideradas em index). No caso do ORDER BY é melhor que as colunas sejam NOT NULL;
- Utilizando-se MAX ou MIN, desde que a função faça parte da seleção e esteja sozinha, pois em caso contrário o index não é utilizado;
- Para colunas que são FK, desde que estas não tenham pouca variação de valores, por exemplo, colunas de domínio;
- Na utilização de OR na cláusula de junção / restrição, deve-se ter index em todas as colunas envolvidas no OR;
- Quando se fizer uso de função na cláusula de junção / restrição, definir um index baseado em função;
- O uso de consultas envolvendo na cláusula de restrições campos textuais devem ser evitados. Caso seja inevitável e o SGBD for o Oracle, utilizar index Intermedia Text (para campos VARCHAR2, CHAR e CLOB).

Conforme Anexo I, o não atendimento deste item contraria o indicador de qualidade acessibilidade.

3.2.12 Triggers

O uso de triggers para implementação de regras de negócio é aconselhável e evitar, sendo interessante a utilização para auditoria, histórico e garantia de integridade quando não é possível a implementação por constraints.

Nos casos de uso de Trigger em implementação de regra de negócio do aplicativo, a responsabilidade é da equipe de sustentação.

Conforme Anexo I, o não atendimento deste item contraria os indicadores de qualidade acessibilidade e credibilidade.

3.2.13 Views

O uso de view tem como objetivo facilitar a visualização dos dados, sendo que com elas é possível simplificar comandos com colunas resultantes de agrupamentos de dados de mais de uma tabela, cálculos, quando há necessidade de inibir colunas ou linhas de tabelas, por questões de segurança, etc.

Sendo assim, o uso de views é interessante quando se tem uma consulta de uso frequente, consultas complexas, para restringir o acesso a visualização de dados.

Conforme Anexo I, o não atendimento deste item não contraria nenhum indicador de qualidade, mas deve-se ter parcimônia no uso de views e observar a real necessidade do uso de view.

3.2.14 View Materializada

Tipo especial de view com característica com armazenamento de dados próprio, originados de uma ou mais tabelas. Bastante utilizadas para facilitar consultas sem impactar na performance das tabelas transacionais. Também permite tratamento de regras negociais e criação de index, diferentemente das views convencionais.

Conforme Anexo I, o não atendimento deste item contraria o indicador de qualidade.

3.2.15 Campo BLOB

Quanto ao uso de campos com tipo BLOB, o armazenamento não deve ocorrer em base de dados, mas sim utilizando a tecnologia NFS (Network Fife System).

Havendo a necessidade de armazenar o nome do arquivo e a descrição do local de armazenamento na base de dados, devem ser utilizados campos com prefixo NO (nome) e DS (descrição, respectivamente) do tipo texto.

Para o armazenamento em filesystem, a equipe de responsável pelo aplicativo deverá procurar a equipe de Infraestrutura para obter as devidas orientações.

Para maiores informações consultar o memorando nº 32/2016/CGAM/DATASUS/SE/MS, SIPAR 2500.083574/2016-10 e Nota Técnica nº. 063/2016/Infraestrutura DATASUS.

Conforme Anexo I, o não atendimento deste item não contraria nenhum indicador de qualidade.

3.3 Privilégios Padrão

Para a concessão de privilégios, temos as seguintes observações:

- Na criação de tabelas os privilégios que são concedidos, por padrão, são **SELECT / INSERT / UPDATE**. Havendo necessidade de **DELETE**, deve ser informado na criação do objeto ou sob demanda após sua criação, informando a necessidade ou não de criação de auditoria no padrão DATASUS (tabelas com prefixo **AU_**).
- As exceções para concessão de privilégios para tabelas são descritas a seguir:
 - Tabela de Auditoria Padrão DATASUS (**AU_**): somente privilégio de **SELECT**.
 - Tabela de LOG cuja manutenção é de responsabilidade da equipe responsável pelo aplicativo (**TL_**): somente privilégio de **SELECT / INSERT**.
 - Tabela de Histórico (**TH_**): somente privilégio de **SELECT / INSERT**. Havendo necessidade de qualquer outro privilégio este deve ser solicitado.
- Para o caso de DROP e recriação de objetos, os privilégios concedidos são os que estão especificados como padrão neste item. Caso o objeto tenha privilégios que não sigam o padrão e que devem ser mantidos, estes devem ser especificados na demanda.
- Os privilégios são concedidos para as roles padrão do esquema e estas para o usuário de manutenção (menos ambiente de Produção) e aplicação (todos os ambientes).
- Para o caso de alteração em View Materializada, esta é recriada, portanto deve seguir as orientações para o caso de DROP e recriação.

3.4 Normas para Modelagem Relacional

Resumidamente, as normas para modelagem de dados devem estar de acordo com o definido nos subitens de 3.3.1 à 3.3.4.

Este item apresenta boas práticas a serem observadas no processo de modelagem de dados.

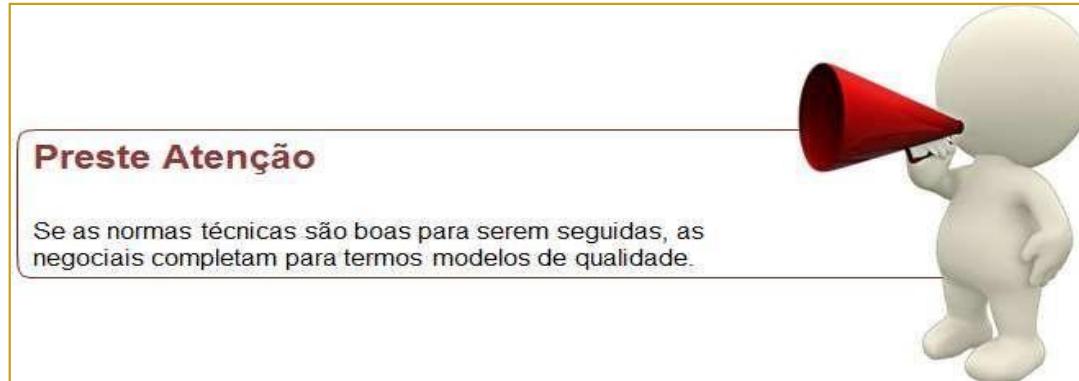


Figura 12 - AD chamando a atenção para as normas negociais

3.4.1 Tabelas

- a) Sempre que for identificada a existência de tabelas distintas com muitas propriedades, atributos e relacionamentos em comum, recomenda-se avaliar a possibilidade de aplicar o conceito de generalização / especialização.
- b) Em princípio, as tabelas deverão ser modeladas respeitando-se as regras de normalização. Desnormalizações poderão ser feitas desde que haja justificativa técnica da sua necessidade, como por exemplo, para facilitar o acesso às informações e/ou para melhoria de performance. No entanto, para qualquer desnormalização deverá haver um procedimento implementado no banco de dados que garanta a integridade da informação. As justificativas para tais procedimentos deverão ser documentadas na ferramenta case.
- c) Em princípio todas as tabelas deverão possuir PK. Para os casos de tabelas em que a Primary Key é uma coluna sequencial, não temos a garantia de unicidade de registros de forma negocial e para essas situações deverá ser verificada a existência de pelo menos uma Unique Key (UK) composta ou simples. Nessa verificação não são levadas em consideração tabelas de apoio, log e temporárias.

- d) O relacionamento entre as tabelas deverá ser feito através da criação de chave estrangeira (FK). Excepcionalmente, quando isso não for possível, deverá ficar documentado na tabela o motivo e a forma que será utilizada para garantir a integridade dos dados, devendo ser observado o item que trata de Primary Key no documento.
- e) As tabelas mais volumosas deverão ser estudadas de forma a se decidir a criação de índices, bem como outras questões físicas como por exemplo, particionamento.
- f) Tabelas com muitas colunas sem preenchimento obrigatório deve-se avaliar real necessidade de tais colunas ou até da tabela. Havendo a necessidade de se manter essa ocorrência, deve haver uma justificativa técnica e documenta-la.

3.4.2 Colunas

- a) Toda coluna cujo conteúdo é uma lista de valores deve ter seu domínio definido.
- b) Colunas cujo conteúdo é uma lista de valores implementada por FK com tabela de domínio ou CK devem possuir preenchimento obrigatório, pois as colunas desse tipo não se devem admitir ocorrência nula, visto que o nulo não é um valor pré-definido e sim ausência de valor.
- c) Colunas cujo conteúdo representam a mesma informação em modelos diferentes devem possuir o mesmo tipo/tamanho e lista de valores quando for o caso.

3.4.3 Outros Objetos

- a) As tabelas e colunas devem estar documentadas conforme definido neste documento.
- b) No caso de ausência de constraint onde deveria existir, deve ser verificado motivo desse tipo de ocorrência e caso seja necessário a não alteração, isso deve possuir uma justificativa técnica e ser documentada.
- c) O desenho gráfico do modelo de dados deve possuir estética agradável com utilização de cores para diferenciar módulos funcionais e tabelas compartilhadas de outros modelos, além do que deve ser evitado o cruzamento de linhas.

3.4.4 Qualidade Negocial

- a) Nome de qualquer objeto deve estar de acordo com a sua finalidade, sendo que para tabelas e colunas isso deve ser sempre verificado e para outros tipos de objetos, somente aqueles que devem ter o nome representativo de acordo com o negócio, conforme especificado no documento de normas de nomenclatura de objetos de banco de dados.
- b) O modelo de dados deve ser aderente aos documentos de especificação do projeto. No caso de situações onde esta regra não é seguida, deve ser documentada a justificativa para tal.



Figura 13 – Modelo de Dados com Qualidade

4 Dicionário de Dados

4.1 Diretrizes para Documentação do Modelo de Dados

Neste item vamos descrever algumas regras básicas que devem ser seguidas para que possamos ter um bom dicionário de dados que é o “Registro detalhado dos conceitos que compõem um universo pré-definido”.

Portanto é importante entender que “A representação gráfica e denominação dos elementos que compõem um modelo de dados não são suficientes para traduzir todos os conceitos do negócio”.

Para que tenhamos uma boa descrição dos elementos de dados devemos considerar os seguintes fatores: clareza, objetividade, respeito à legislação e normativos, respeito à terminologia da área negocial que está sendo tratada, respeito às normas da língua portuguesa, citação de exemplos.

Então não corra o risco de interpretações equivocadas e não claras, ou você prefere isso?



Figura 14 - Dicionário de Dados não bem compreendido

E para gerar um dicionário de dados que seja bem compreendido e que gere um metadados de qualidade, o que deve ser feito?

Em primeiro lugar na descrição de uma tabela não deve constar que “armazena dados ...”, mas que representa uma entidade negocial, então na descrição da tabela deve ser bem-conceituada essa entidade.

No caso das colunas, se a entidade que a tabela representa estiver bem-conceituada, a descrição destas fica simplificada, mas deve ficar claro o que a coluna representa dentro do contexto da entidade.

Para facilitar, a descrição da tabela é interessante responder as seguintes perguntas sobre a entidade que representa.

- O que é a entidade?
- O que faz a entidade?
- Para que serve?
- Quando alguém ou algo passa a ser, ou deixa de ser, um elemento dessa entidade?
 - Em caso de tabela de domínio, devem constar exemplos.
 - A exclusão é lógica?



Figura 15 - Dicionário de Dados com informação relevante

Assim a descrição da tabela é formada pelo conjunto dessas respostas. Outro ponto a ser observado é que nem sempre é conseguido responder a todas as perguntas, mas é preciso tentar.

Para que um modelo de dados seja considerado de qualidade, ele tem que ser bem documentado, ou seja, o dicionário de dados deve conter informação relevante e é obrigatório!

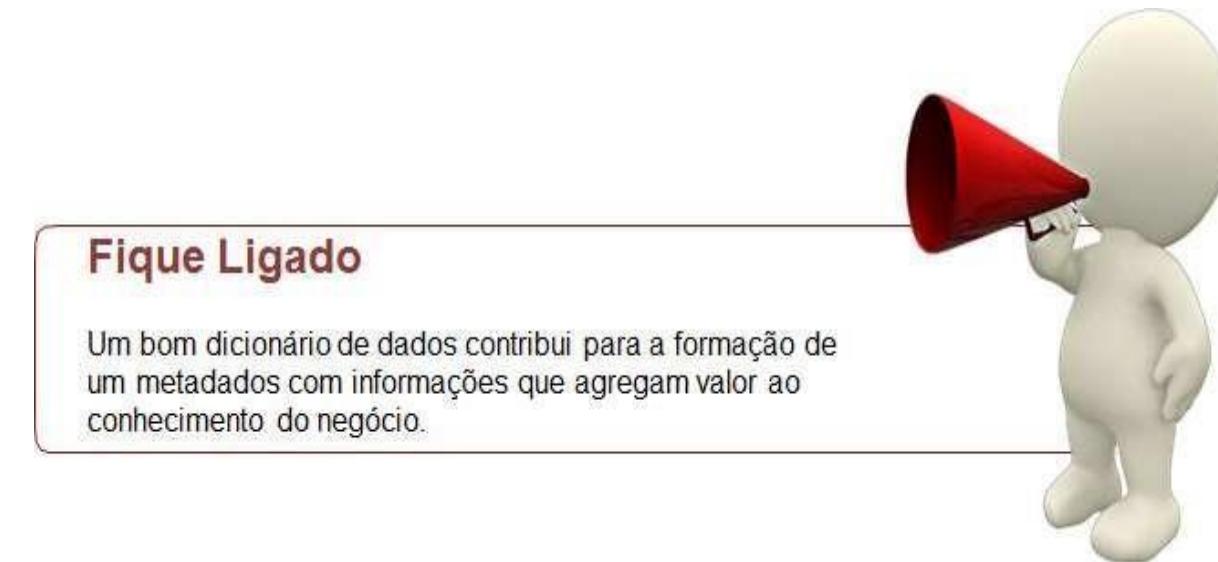


Figura 16 - Fique Ligado – Gere um bom dicionário de dados

4.2 Descrições Específicas

A seguir temos sugestões de algumas descrições específicas de determinadas categorias de colunas que são muito utilizadas. É importante ficar claro que as descrições utilizadas no modelo não precisam ser exatamente iguais as aqui sugeridas, mas o significado deve ser o mesmo.

- Colunas que são PK e alimentadas por sequence:

ORACLE - “Representa a chave primária sequencial da tabela, que é controlada pela sequence [NomeSequence] do banco de dados específica para a tabela.”

POSTGRE ou MySQL - “Representa a chave primária sequencial da tabela, que é controlada pelo banco de dados através do datatype desta coluna SERIAL.” Este caso é para quando não é definida uma sequence específica.

- Colunas que são FK em uma tabela

[Descrição da finalidade da coluna dentro de contexto da entidade negocial representada pela tabela.] + [A coluna é FK da tabela [NomeTabela].]

- Coluna ST_REGISTRO_ATIVO:

“Indica se o registro está ativo ou não (excluído logicamente). O seu domínio é: S – Sim (está ativo) ou N – Não (não está ativo). O controle no uso de registros excluídos deve ser feito pela aplicação.”.

- Coluna que possui CK:

[Descrição da finalidade da coluna] + [Domínio] Observação: Entender como domínio, a indicação de todos os valores, e a descrição do significado de cada um (semelhante ao ST_REGISTRO_ATIVO).

- Tabela de Domínio:

O comentário da tabela de domínio, além da descrição de sua finalidade, deve conter a indicação de todos os valores e a descrição do significado de cada um. Caso o domínio seja composto por mais de 10 valores, indicar exemplos.

4.3 Armazenamento de Dicionário de Dados

As informações expostas neste anexo devem ficar armazenadas na ferramenta case adotada pelo Ministério da Saúde, sendo que o que estiver no Comment deve ser gerado script para armazenamento no Banco de Dados. Os formulários com tais informações do dicionário de dados ficam são descritas a partir da figura 18.

É importante salientar que os formulários são definidos no driver do banco de dados que devem ser fornecidos para todos que utilizam a ferramenta case e para cada SGBD devemos ter essas definições.

4.3.1 Modelo de Dados

A informação para dicionário de dados para o “Modelo de Dados” na ferramenta case, podem ser visualizadas nas figuras 18 e 19.

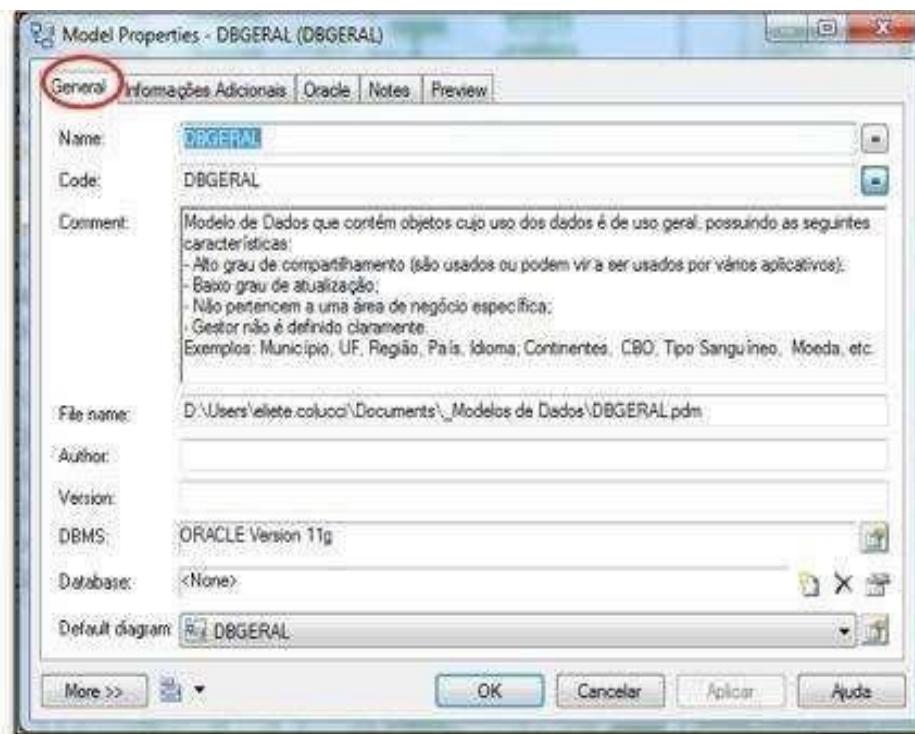


Figura 17 - Informações gerais do modelo de dados

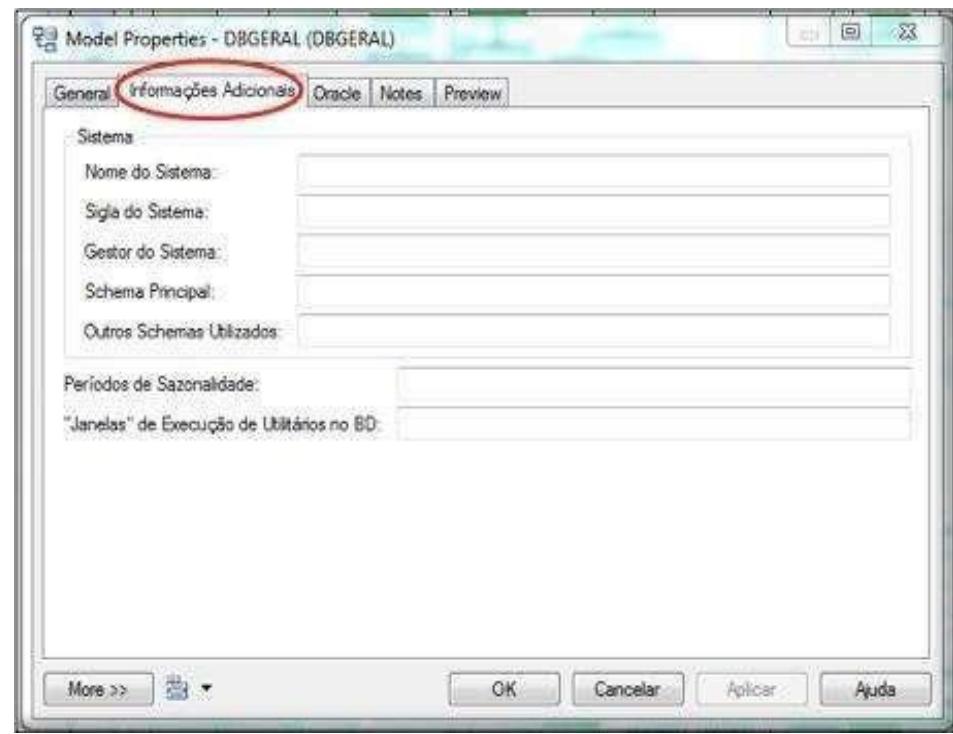


Figura 18 - Informacionais adicionais do modelo de dados

4.3.2 Tabela

A informação para dicionário de dados para a “Tabela” na ferramenta case, podem ser visualizadas nas figuras 20 e 21.

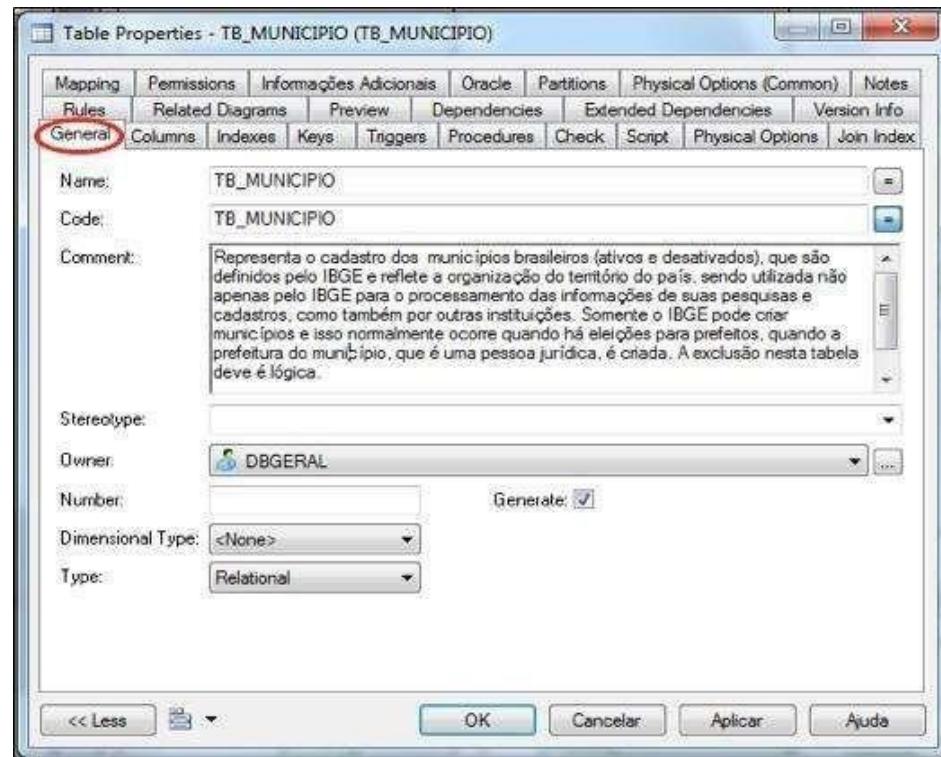


Figura 19 - Informações gerais de tabela

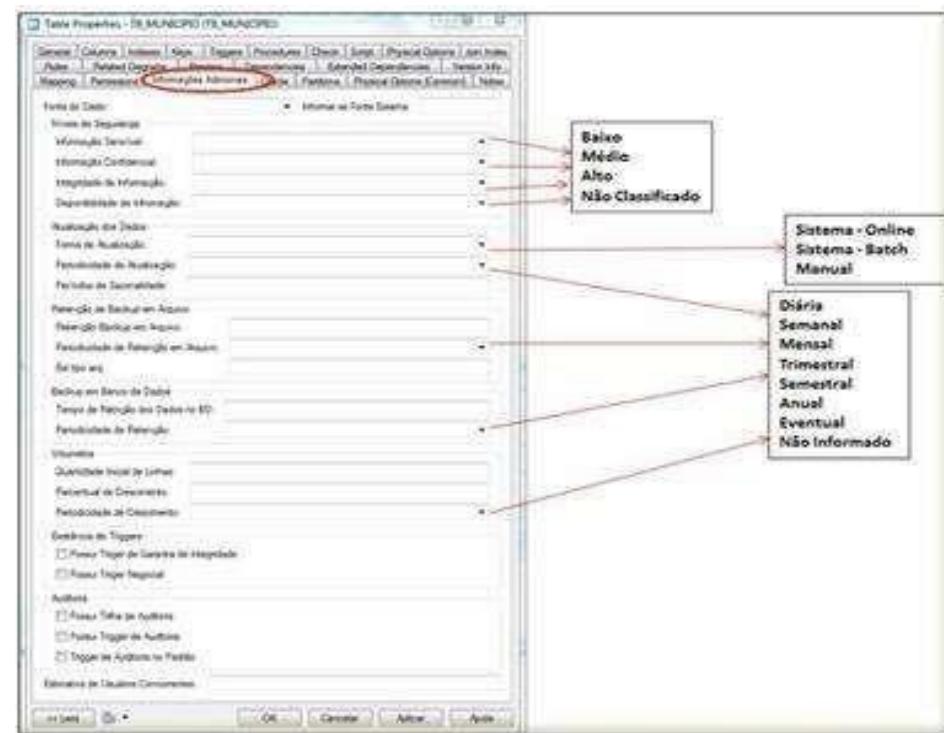


Figura 20 - Informações adicionais de tabela

4.3.3 Coluna

As informações para dicionário de dados para a “Coluna” na ferramenta case podem ser visualizadas nas figuras 22.

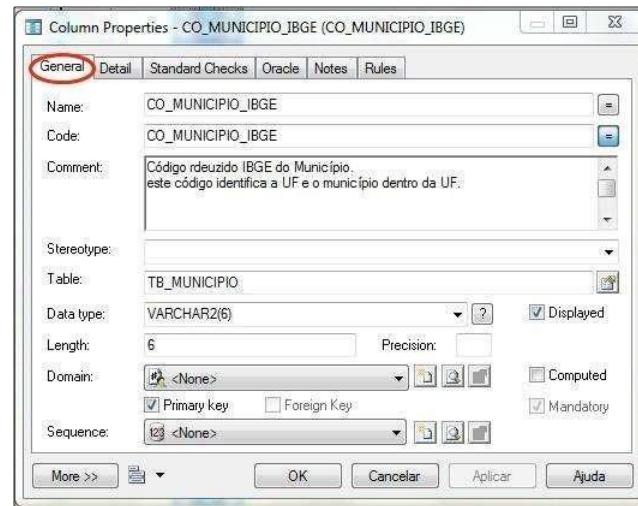


Figura 21 - Informações gerais de coluna



E como faço para esse novo formulários na ferramenta case PowerDesigner?

Já temos um drive pronto na ferramenta case PowerDesigner para o SGBD Oracle. É só procurar a equipe de banco que forneceremos o novo drive ora11g.xdb. Depois na pasta C:\Program Files\Sybase\PowerDesigner 15\Resource Files\DBMS renomear o arquivo ora11g.xdb para ora11g_Original.xdb e copiar o novo drive.

Figura 22 - Dúvida sobre os novos formulários no PowerDesigner

5 Outros Tipos de Demandas

5.1 Concessão de Permissões e Auditoria

A segurança de dados é um requisito de gerenciamento de sistemas de banco de dados para proteger os dados contra usuários não autorizados, devendo ser implementada por meio de mecanismos de concessão de permissões de usuários e regras para tais concessões de forma que haja garantia de a proteção dos dados.

É importante salientar que pelos procedimentos que temos a concessão de permissões e a geração de AUDITORIA estão ligadas conforme descrito nos itens 5.1.1 e 5.1.2.

Observe que nos itens 5.1.1. e 5.1.2, quando é dito “Questionar Auditoria”, deve-se considerar as questões citas no item 3.2.7 – figura 12.



Figura 23 - Concessão de Acesso

5.1.1 Concessão para tabelas de próprio esquema

As regras definidas na tabela 3 devem, ser consideradas para a concessão de privilégio de **DELETE** em **tabelas no próprio esquema**.

Ambiente	Tipo do Usuário	Auditória	Procedimento	Autorização
Produção	Aplicação	Possui Auditoria	Conceder Acesso	Desnecessária
		Não Possui Auditoria	Conceder Acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela “Não Criação da Auditoria”, formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Não se Aplica	Não se Aplica	Não se Aplica
Homologação	Aplicação	Possui Auditoria	Conceder Acesso	Desnecessária
		Não Possui Auditoria	Conceder Acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela “Não Criação da Auditoria”, há necessidade de formalização por e-mail junto à área gestora / coordenação da área de sustentação responsável da não necessidade de criação de auditoria
	Manutenção	Possui Auditoria	Conceder Acesso	Desnecessária
		Não Possui Auditoria	Conceder Acesso	
Treinamento	Aplicação	Possui Auditoria	Conceder Acesso	Desnecessária
		Não Possui Auditoria	Conceder Acesso + Auditoria (questionar somente se não existe no desenvolvimento)	Caso seja decidido pela “Não Criação da Auditoria”, formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Possui Auditoria - Não Precisa	Conceder Acesso	Desnecessária
		Não Possui Auditoria - Não Precisa	Conceder Acesso	
Desenvolvimento	Aplicação	Possui Auditoria	Conceder Acesso	Desnecessária - Informar ao solicitante a necessidade de criação de auditoria em todos os ambientes
		Não Possui Auditoria	Conceder Acesso + Questionar Auditoria	1) Caso seja decidido pela “Não Criação da Auditoria”, formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável Informar ao solicitante necessidade de criação de auditoria em todos os ambientes
	Manutenção	Possui Auditoria	Conceder Acesso	

		Não Possui Auditoria	Conceder Acesso	Desnecessária - Informar ao solicitante necessidade de criação de auditoria em todos os ambientes
--	--	----------------------	-----------------	---

Tabela 2 – Concessão de acesso em tabelas do próprio esquema e Auditoria

5.1.2 Concessão em tabelas de esquema diferente

As regras definidas na tabela 4 devem ser consideradas para a concessão de privilégio de **SELECT / INSERT / UPDATE / DELETE** em esquema diferente, sendo que a auditoria não é necessária para SELECT.

Ambiente	Tipo do Usuário	Auditoria	Procedimento	Autorização
Produção	Aplicação	Possui Auditoria	Termo de Responsabilidade	SIPAR
		Não Possui Auditoria	Termo de Responsabilidade + Auditoria (questionar somente do desenvolvimento)	1) SIPAR 2) Caso seja decidido pela “Não Criação da Auditoria”, formalizar por e-mail junto à área gestora / coordenação da área de sustentação responsável
	Manutenção	Não se Aplica	Não se Aplica	Não se Aplica
Homologação	Aplicação	Possui Auditoria	Conceder Acesso	Formalização por e-mail de área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
			Conceder Acesso + Auditoria (questionar somente do desenvolvimento)	1)
	Manutenção	Possui Auditoria	Conceder Acesso	Formalização por e-mail de área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
Treinamento	Aplicação	Não Possui Auditoria	Conceder Acesso	Formalização por e-mail de área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
		Não Possui Auditoria	Conceder Acesso Auditoria (questionar somente do desenvolvimento)	
	Manutenção	Possui Auditoria - Não Precisa	Conceder Acesso	Formalização por e-mail de área gestora responsável pelo objeto envolvido autorizando a concessão de acesso
		Não Possui Auditoria - Não Precisa	Conceder Acesso	
Desenvolvimento	Aplicação	Possui Auditoria	Conceder Acesso	

		Não Possui Auditoria	Conceder Acesso + Questionar Auditoria	
Manutenção	Possui Auditoria		Conceder Acesso	
	Não Possui Auditoria		Conceder Acesso	

Tabela 3 – Regras para concessão de permissões em Tabelas de esquemas diferentes e Auditoria

5.2 Solicitação de DROP / TRUNCATE / DELETE

O DROP de um objeto em banco de dados elimina-o, não sendo mais possível recuperá-lo caso não haja uma cópia de segurança.

No caso do comando TRUNCATE em uma tabela, todas as linhas são removidas e não pode ser revertida, como no caso de DELETE é preciso executar comando COMMIT para confirmar exclusão ou o comando ROLLBACK para reverter a transação de efetuar a mudança.

Devido a essas características os comandos: DROP e TRUNCATE, são necessários alguns cuidados adicionais são necessários.

- O comando TRUNCATE remove todas as linhas de uma tabela.
- Para a solicitação de DROP ou TRUNCATE de tabelas as regras são as seguintes:
 - Ambiente de desenvolvimento, treinamento e homologação: será realizada sem necessidade de formalização da área gestora responsável pelo objeto. Nesta situação deverá ser informada a necessidade de formalização para produção (conforme item abaixo), bem como uma análise de impacto (baseada em FK's e roles), caso a tabela seja utilizada por outros esquemas.
 - Para o ambiente de produção, somente será necessária a formalização da área gestora havendo linhas na tabela.

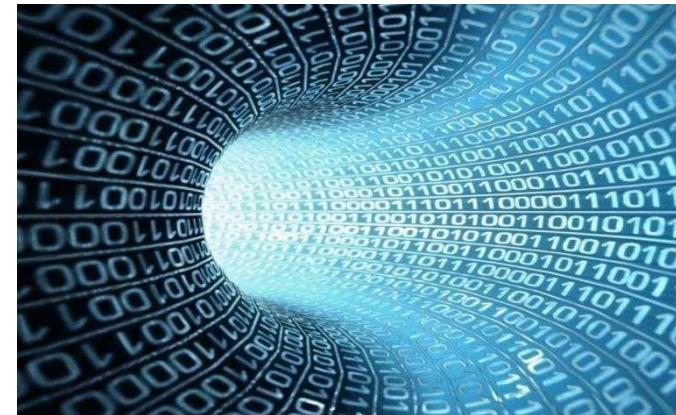


Figura 24 - - DROP e TRUNCATE de tabelas

6 Conclusão

Nesta cartilha temos os principais aspectos que devemos nos preocupar para que tenhamos modelos de dados com qualidade. Essas orientações merecem uma reflexão e uma mudança na construção de modelos de dados, de forma que possamos garantir a qualidade técnica destes, além de que sejam aderentes ao negócio que devem atender.

Estou com uma dúvida. A quem devo recorrer?

E quando há uma dúvida, procure se informar nesta cartilha, ou então consulte quem realmente entende do assunto!

Entre em contato com a equipe de banco. Podemos ser contatados por:

- e-mail: datasus.ad@saude.gov.br;
-



Figur
de banco

equipe

7 ANEXO I

7.1 Indicadores de Qualidade de Modelo de Dados

Para que um modelo de dados seja considerado de boa qualidade, deve atender os requisitos funcionais do projeto a que se propõe atender e seguir as boas práticas de modelagem de dados.

Para que um modelo de dados seja avaliado como seguindo as boas práticas, os itens descritos a seguir devem ser considerados.

- a) **Acessibilidade:** o modelo disponibiliza as informações de forma clara, fácil e rápida manipulação.
- b) **Reutilização:** o modelo apresenta facilidade de reutilização de seus objetos em outros negócios da instituição e com isso facilita a existência de um modelo corporativo.
- c) **Amplitude:** o modelo deve ser completo, profundo e objetivo de forma a atender todas as necessidades de informações do negócio, sendo que estas deverão ser de fácil compreensão de todos e possuírem condições de serem atualizadas.
- d) **Representação Concisa:** no modelo devem estar representadas a quantidade correta de informações que o negócio necessita, isto é, deverá conter as informações que são importantes e de interesse ao negócio, que são aplicáveis à tarefa a que se propõe, sendo que normalmente informação que não se tem condição de mantê-la atualizada normalmente não é importante (excesso de informações pode não auxiliar na análise, além de se gastar mais tempo na escolha de informações relevantes, por exemplo tabelas com muitas colunas NULL).
- e) **Valor Agregado:** no modelo devem estar contidas informações que trazem benefício ou provêm vantagens pelo seu uso;
- f) **Credibilidade:** no modelo deve ter tratamento para as informações de forma que o armazenamento das mesmas seja dentro da realidade, correto e confiável e que por isso seja capaz de ser considerada verdadeira e crível. Além disso, as informações devem ser bem-conceituadas quanto a sua origem, conteúdo e legislação vigente, além de apresentar uma representação consistente, isto é, informações do mesmo tipo deverão ser apresentadas sempre no mesmo formato em todos os modelos (por exemplo, não é interessante ter a informação de sexo de uma pessoa definida em um modelo com F e M e em outro como 1 (Feminino) e 2 (Masculino)).
- g) **Flexibilidade:** o modelo de dados deve ser flexível, de forma a suportar as mudanças necessárias sem grandes impactos.
- h) **Documentação:** Todos os elementos contidos no modelo de dados devem ser nomeados de acordo com os padrões definidos para este órgão, bem como estar documentados com definições completas e inequívocas.
- i) **Legibilidade:** o gráfico do modelo de dados deve ser legível, ou seja, respeita critérios de estética que tornam o diagrama agradável para a leitura. Por exemplo: evita cruzamento de ligações dos relacionamentos; evita curvas; sempre que possível respeita a hierarquia, ou seja, coloca os “pais” acima dos “filhos”.

